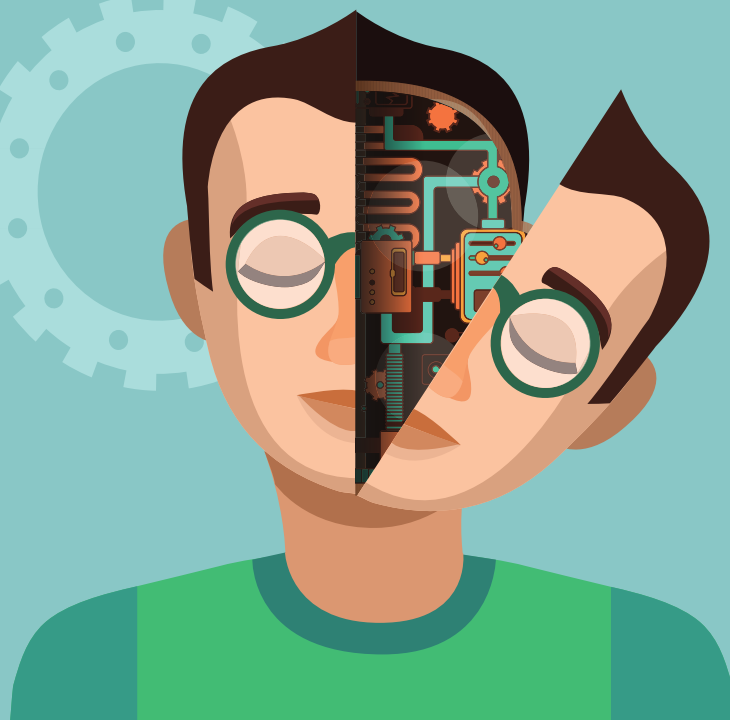
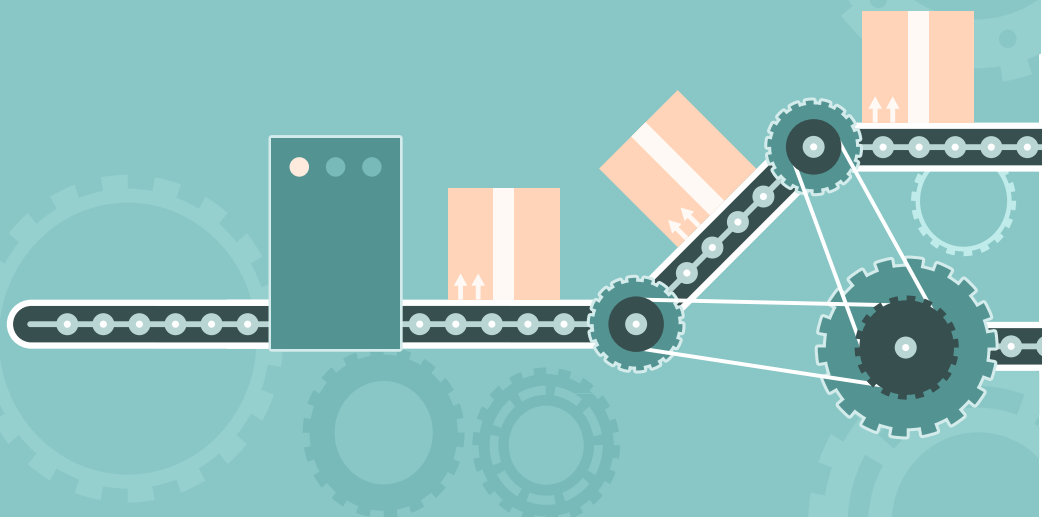


Marcelo R. Petry
Janaina G. Guimarães (Eds.)

Marcelo R. Petry
Janaina Gonçalves Guimarães (Eds.)



I Workshop de Engenharia de Controle e Automação



WECA 2016 - UFSC - Blumenau - Brasil

Marcelo R. Petry
Janaina Gonçalves Guimarães(Eds.)

Anais do I Workshop de Engenharia de Controle e Automação

Universidade Federal de Santa Catarina, Blumenau, Brasil
14 e 15 de julho de 2016

Editores

Marcelo R. Petry

Universidade Federal de Santa Catarina - UFSC

Rua Pomerode, 710, Bairro Salto do Norte

Blumenau, SC, Brasil, 89065-300

E-mail: marcelo.petry@ufsc.br

Janaina Gonçalves Guimarães

Universidade Federal de Santa Catarina - UFSC

Rua Pomerode, 710, Bairro Salto do Norte

Blumenau, SC, Brasil, 89065-300

E-mail: janaina.guimaraes@ufsc.br

ISBN xxxxxxxx-xxxx

Esta obra é distribuída por meio da Licença
Creative Commons-Atribuição-NãoComercial 4.0 Internacional



Prefácio

Este volume contém 43 artigos aceitos e apresentados no I Workshop de Engenharia de Controle e Automação realizado nos dias 14 e 15 de julho de 2016 no Centro de Blumenau da Universidade Federal de Santa Catarina. O evento foi concebido e preparado com o objetivo de provocar entre os alunos do curso de Engenharia de Controle e Automação e demais participantes debates e críticas construtivas a respeito dos conceitos apresentados em sala de aula. Ademais, também procurou-se incentivar a busca por novos conceitos técnicos e tecnológicos que agregassem valores à formação profissional dos graduandos.

O curso de Engenharia de Controle e Automação da UFSC em Blumenau foi criado em 2014. A infraestrutura acadêmica e física ainda está em fase de implementação. Nesse contexto, a realização do Workshop visa também contribuir para a fundamentação dos conceitos de pesquisa e inovação tecnológica no mais novo Campus da Universidade.

Os trabalhos apresentados nessa primeira edição foram, em sua grande maioria, desenvolvidos no decorrer das disciplinas oferecidas durante o 1o semestre de 2016. Dessa forma, houve a oportunidade dos alunos aplicarem os conhecimentos teóricos adquiridos durante as aulas já vislumbrando o desenvolvimento de projetos de engenharia.

Gostaríamos de expressar nosso agradecimento ao trabalho dos membros do Comitê de Programa, que foram responsáveis por avaliar e revisar os artigos submetidos em um tempo substancialmente curto. Também gostaríamos de agradecer ao Campus de Blumenau pelo suporte financeiro, pela disponibilização do espaço físico, bem como pela disponibilização de dois Técnico-Administrativos em Educação. Este evento não teria se tornado realidade sem a contribuição dos Técnico-Administrativos em Educação Eduardo José Borba de Amorim e Marcel Luis Agostini, que trabalharam na organização local, design gráfico, e gestão do sistema multimídia. Por fim, gostaríamos de agradecer ao EasyChair por disponibilizar de forma gratuita sua plataforma de gestão de conferências.

3 de agosto de 2016
Blumenau

Marcelo R. Petry
Janaina Guimaraes

Comitê de Programa

Adão Boava	Universidade Federal de Santa Catarina - UFSC
Tiago Busarello	Universidade Federal de Santa Catarina - UFSC
Mauri Ferrandin	Universidade Federal de Santa Catarina - UFSC
Janaina Guimarães	Universidade Federal de Santa Catarina - UFSC
Daniel Lima	Universidade Federal de Santa Catarina - UFSC
Marcelo Petry	Universidade Federal de Santa Catarina - UFSC
	INESC P&D Brasil
Alex Sandro Roschildt Pinto	Universidade Federal de Santa Catarina - UFSC
Ebrahim Youssef	Universidade Federal de Santa Catarina - UFSC

Índice

Detector de Acidentes	1
<i>Henrique Lange, Herman Schwank, Erick Bergmann, Marco Akerman and Carlos Novelletto</i>	
CNC	6
<i>Rodolfo Floriani</i>	
Projeto Arduino - Porta automatizada: fechadura sonora	13
<i>Tiago Battiston, Arthur Milheiro, Guillermo Dilly, Gustavo Luz and Maria Martins</i>	
Sistema de fechadura inteligente, baseado em RFID	18
<i>Nicolas Antero, Lukas Belck, Alvaro Rodrigo, João Victor Ferreira and Gabriel Ferraz</i>	
Sistema Above de Segurança	24
<i>Nathã Vieira de Azevedo, Allyson Freitag Binsfeld, Felipe Alfredo Nack, Gustavo Paloschi and Lucas Azzolini Vieira</i>	
Projeto Armaduino	30
<i>Allan Heil, Guilherme Geske, Arthur Wippel, Carlos Ribeiro and Bruno Bubeck</i>	
Nursy Project	36
<i>Elton Silva, Gabriel Davanço and Rafael Lobato</i>	
HAPPY PET	41
<i>Bruna Pinto, Douglas Liao, Gabriel Mafra, Gabriel Zimmer and Martha Bloemer</i>	
Gerenciamento Integrado de Parâmetros Agronômicos para Cultivo Indoor	47
<i>Christian Mailer, André Filipe Obenaus and Bruno Leone Hasse</i>	
Sistema Automatizado de Irrigação por Gotejamento	53
<i>Cecili Duwe Fernandes, Jeane Almeida Santos and Letícia Terezinha Farias Schiochet</i>	
Piano Eletrônico com CI555	58
<i>Gabriel A. Silvestre</i>	
Utilização de Circuitos Eletrônicos para Reproduzir Distorções de Som	64
<i>Brunno Vanelli and Guilherme Moser Manerichi</i>	
Reguladores conversores de Tensão	69
<i>Andre Luiz De Sousa Marcondes Reuter and Juliano E. N. Masson</i>	
Amplificador de Áudio usando TDA 2050	75
<i>Rômulo Pacher and Stephan August Roehrig Domingues Dos Santos</i>	
Stun Gun	79
<i>Caique Sanches, Enrique Geske and Diogo Luiz Demarchi</i>	

Implementação Física de um neurônio artificial eletrônico.....	82
<i>Christian Correa de Borba, Bianca Menon and Janaina Gonçalves Guimarães</i>	
Problema de um Cruzamento Controlado: Modelagem e Análise por Redes de Petri.....	86
<i>Brunno Vanelli, André Luiz De Sousa Marcondes Reuter and Rômulo Pacher Júnior</i>	
Sistema Controlado de Trens.....	90
<i>Guilherme Moser Manerichi, Juliano Emir Nunes Masson and Stephan August R. D. Dos Santos</i>	
Modelagem do Problema de Sincronização Passeio dos Filósofos Através da Rede de Petri	94
<i>Pedro Arthur Cogliatti, Ciro Ota Umezawa and Caique Sanches</i>	
Misturador Automático.....	98
<i>Martin Hermes Anschau, Christian Correa de Borba and André Luiz Granemann</i>	
Misturador de tintas.....	106
<i>Artur Henrique Lauth, Nikolas Sbaraini Hammes and Arthur Phillip B. Siegel</i>	
Fabricação de Refrigerantes.....	112
<i>Eduardo Sonohara, Leonardo Almeida and João Gabriel Godoy</i>	
Controle de Processo Industrial Genérico com Interface entre CLP e Microcontrolador....	119
<i>André Filipe Obenaus, Júlio Joel Da Costa Neto and Pedro Arthur Cogliatti</i>	
Modelo de Sistema de controle para motor dc.....	125
<i>Nikolas Hammes and Rodrigo Weber</i>	
Uma breve descrição da rede PROFINET.....	130
<i>Juliano E. N. Masson</i>	
Protocolo de Comunicação Sem Fio Zigbee.....	136
<i>Brunno Vanelli</i>	
Bluetooth.....	137
<i>Jorge Santana</i>	
Redes de Comunicação Automotiva: Apresentação e perspectivas.....	144
<i>Gabriel A. Silvestre</i>	
CAN - Control Area Network.....	151
<i>André Luiz De Sousa Marcondes Reuter</i>	
Redes Veiculares: Princípios e aplicações.....	157
<i>Gabriela Kriech</i>	
AS-INTERFACE.....	158
<i>Ronaldo Baggio</i>	
P-NET.....	164
<i>Guilherme Moser Manerichi</i>	

HART e WirelessHART	169
<i>Fellipe E G Silva</i>	
Rede Sinec	170
<i>Diogo Luiz Demarchi</i>	
Protocolo de Redes Industriais PROFIBUS	175
<i>Rômulo Pacher</i>	
Protocolo BITBUS	181
<i>Caique Sanches</i>	
Interbus	186
<i>Enrique Geske</i>	

Detector de Acidentes

Trabalho Arduino

Henrique Marchi Lange, Herman Schwank Silva Queiroz, Marco Antonio Akerman, Erick Bergmann, Carlos Eduardo Novelletto Ricardo

UFSC: Universidade Federal de Santa Catarina
Engenharia de Controle e Automação
Blumenau, SC
Henrique.marchi.lange@grad.ufsc.br

Resumo— Atualmente se tornou muito decorrente em nossas rotinas nos depararmos com acidentes entre veículos. O Projeto que segue tem como objetivo reduzir o intervalo de tempo entre um acidente grave de um automóvel e o socorro fornecido por equipes de resgate. Dessa forma, a partir dos componentes Arduino Uno, Piezo eletrônico e Shield GPRS foi desenvolvido um protótipo capaz de captar batidas e classificar sua gravidade. Assim, quando ocorrer um acidente não haverá o risco de que ninguém alerte o corpo de bombeiros ou a polícia. Portanto, esse projeto consegue substituir a necessidade de pessoas para alertarem sobre acidentes.

Palavras-chave— Prototipagem; Segurança; Acidentes; Arduino; Tráfego;

I. INTRODUÇÃO

Atualmente se tornou muito decorrente em nossas rotinas nos depararmos com acidentes entre veículos. Muitas vezes estes causam a morte de várias pessoas, o que faz com que a sociedade, cada vez mais, tenha medo de dirigir em estradas muito movimentadas. Para piorar a situação, acidentes são inevitáveis, a chance de que uma pessoa sofra um acidente ao ir para o trabalho é maior do que as chances de um avião cair.

Porém, muitas vezes as mortes ligadas a acidentes de trânsito estão relacionadas à demora para que um serviço de resgate chegue no local, tudo isso se deve ao fato de que ainda não existe um serviço instantâneo que envie um alerta para o corpo de bombeiros, polícia, etc. No momento em que ocorreu a batida.

O Projeto que segue tem como objetivo reduzir o intervalo de tempo entre um acidente grave de um automóvel e o socorro fornecido por equipes de resgate. Além disso, também se tem como objetivo fornecer os detalhes do acidente, classificando-o em categorias de acordo com a sua gravidade.

II. PROBLEMA

A. DESCRIÇÃO

Devido ao crescente número de veículos em nossas estradas a ocorrência de acidentes os envolvendo também se tornou elevada. Em muitos países, como o Brasil, o principal meio de transporte é o rodoviário e não existem alternativas realmente

eficientes para substituí-lo, fazendo com que a maior parte da população seja obrigada a ter um veículo. Tendo isso em mente, pode-se concluir que acidentes envolvendo automóveis são inevitáveis.

Contudo, grande parte das mortes que ocorrem em acidentes se devem à demora para que as pessoas envolvidas nestes sejam socorridas. As equipes de emergência muitas vezes são incapazes de chegar a tempo de socorrer a(s) vítima(s), isso porque estas equipes são alertadas por pessoas que veem o acidente, mas o grande problema surge quando ninguém passa pelo local onde ocorreu a batida por um longo período de tempo.

Só no Brasil, segundo o departamento de informática do Sistema Único de Saúde do Brasil (SUS), de 2004 a 2014, o número de mortes por acidentes de trânsito registradas chega a ser maior que 400 mil. O gráfico abaixo ilustra o número de mortes ocorridas em acidentes de trânsito no período de 2004 a 2014.

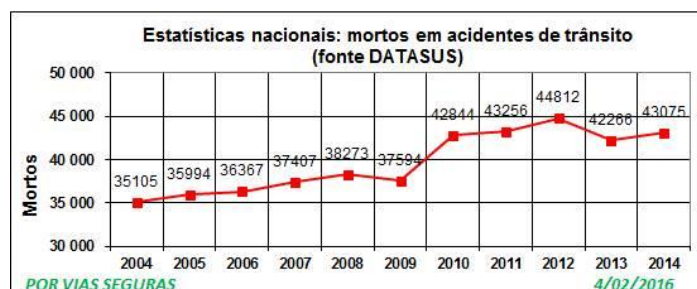


Fig. 1. Gráfico de fatalidades no trânsito.

B. POSSÍVEIS OCORRÊNCIAS DE ACIDENTES

O impacto da batida seja forte o suficiente, e assim, deixe a vítima presa dentro do veículo ou que ele presencie vazamentos de fluidos do veículo, dessa forma, será preciso chamar o corpo de bombeiros;

Um acidente no trânsito em que a vítima fica inconsciente e/ou tenha ferimentos graves e precise de primeiros socorros, será necessário chamar ambos corpo de bombeiros e ambulâncias;

Omissão de socorro no trânsito: Após uma batida entre dois veículos distintos ou até mesmo entre um veículo e um pedestre, sendo que, uma das pessoas fica incapacitada para pedir o socorro própria e a outra ainda consciente do ocorrido, vai embora do local sem prestar socorro imediato.

Ou ainda que seja uma emergência em uma rodoviária, é imprescindível alertar à Polícia Rodoviária ou Estadual, Serviço de Atendimento ao Usuário(SAU) e Serviços Rodoviários Federais ou Estaduais mais próximos e o mais rápidos possível.

Logo a ideia deste projeto é desenvolver um produto capaz de substituir a necessidade de pessoas para alertarem sobre acidentes, que esteja embarcado nos veículos e que avise instantaneamente alguma equipe de socorro sobre o ocorrido.

III. SOLUÇÃO

Com o intuito de reduzir o número de mortes relacionadas a acidentes entre veículos foi desenvolvido um protótipo capaz de captar batidas. Este será implementado diretamente nos veículos, assim quando ocorrer um acidente não haverá mais o risco de que ninguém alerte o corpo de bombeiros ou a polícia.

O intuito do projeto é de que o protótipo final consiga captar batidas frontais que ocorrem quando um dos veículos estiver andando a uma velocidade superior à 60 km/h ou quando o veículo capotar e assim alertar o corpo de bombeiros através de mensagens de texto. Deste modo serão evitados alarmes falsos, uma vez que batidas que ocorrem abaixo desta velocidade não apresentam grande risco para os envolvidos.

Como pretende-se que o protótipo consiga enviar alertas do maior número de locais possíveis no Brasil (onde primeiramente será implementado) será usado um GPRS, devido ao fato de que a cobertura do sinal de telefone ainda supera a de outras redes de comunicação como a internet e é mais viável economicamente do que o uso de satélites. A imagem abaixo ilustra a cobertura do sinal de telefone no Brasil. Obs: Escala de 0 a 4 ou 5 onde pontos mais claros (Branco são menores quantidades de operadoras e Azul Marinho maiores quantidades).

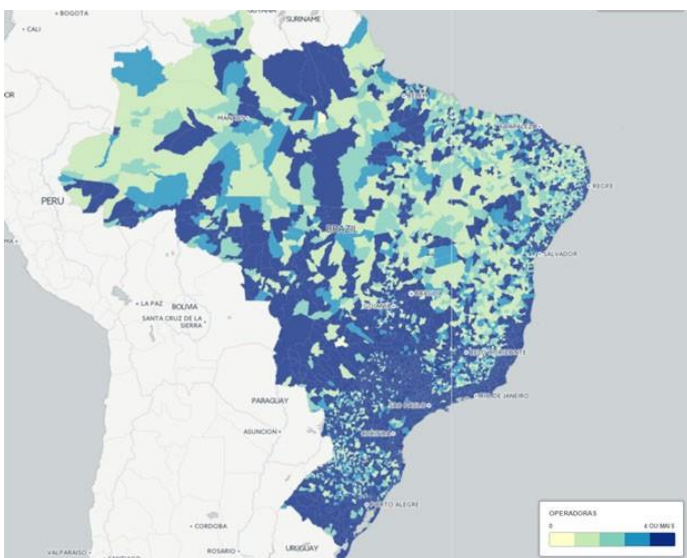


Fig. 2. Mapa da cobertura de sinal de telefone no Brasil.

Nos próximos tópicos serão especificadas a montagem do projeto, bem como o hardware necessário para a criação do mesmo, o algoritmo empregado e os custos envolvidos na implementação e produção dos protótipos.

A. ESPECIFICAÇÕES DE HARDWARE

Com o intuito de produzir um protótipo compacto o suficiente para poder ser instalado em qualquer veículo, foi criado um hardware que usa o mínimo possível nos quesitos de sensores e Shields e ao mesmo tempo consiga atender às exigências da proposta. Os componentes utilizados, bem como os seus preços estão descritos na tabela abaixo:

Arduino Uno R3	R\$ 119,00
Piezo Elétrico	R\$ 0,60
Shield GPRS	R\$ 210,00
Jumpers	R\$ 9,00
Compartimento para o hardware	R\$ 10,00

^a componentes e seus respectivos preços

Somando os valores acima temos um custo final de R\$ 348,60 para a produção de um protótipo. Tal custo poderá ser reduzido ao produzir um maior número de protótipos. Além deste custo de produção haverá os custos de instalação e manutenção, estes serão especificados na seção de montagem e implementação.

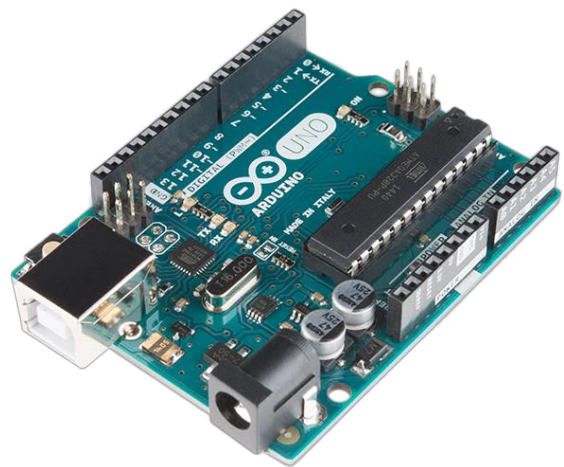
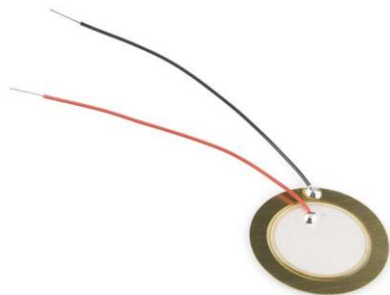


Fig. 3. Microcontrolador utilizado no projeto.

Fig. 4. *Shield GPRS.*Fig. 5. *Piezo eletrônico.*Fig. 6. *Compartimento.*

B. MONTAGEM E IMPLEMENTAÇÃO

O equipamento irá ser posto dentro de um compartimento, este será instalado no interior do carro em um local onde não ocorram perturbações externas capazes de afetar o funcionamento do dispositivo, uma vez que este possui uma função muito importante e não pode ser alterado pelo usuário. A antena do GPRS ficará na parte externa do compartimento, é recomendável que o equipamento seja instalado no teto do carro na parte externa do veículo para que a antena capte melhor o sinal.

A fonte de alimentação será a própria bateria do carro, uma vez que o projeto não requer uma alta taxa de energia para funcionar. O recomendável é que o protótipo seja ligado em uma

bateria que possa fornecê-lo 12 Volts e 1 Ampère. Para testes recomenda-se ligar o dispositivo em uma fonte de 12 V e 1 A.

Pensando em reduzir custos e facilitar a instalação do equipamento nos veículos o mesmo será disponibilizado para que as próprias montadoras os instalem em seus veículos da maneira que lhes for mais convenientes, porém esta deverá seguir os requisitos mínimos para o bom funcionamento do produto, tais requisitos estarão especificados na seção de algoritmo e lógica de funcionamento.

Como, com o decorrer do tempo, o funcionamento do produto pode ser afetado é essencial que seja feita uma revisão do equipamento juntamente com a revisão do veículo, esta revisão deve ser feita pelo grupo, tendo em vista que este não pode apresentar falhas que poderiam ser ocasionalmente geradas por mal calibragem de sensores ou mal avaliação do funcionamento do produto. Esta revisão terá um custo de R\$ 50,00 mais o custo de possíveis peças defeituosas que precisarão ser trocadas.

Fig. 7. *Protótipo final.*



Fig. 8. Protótipo final.

C. ALGORITMO E LÓGICA DE FUNCIONAMENTO

O algoritmo presente no produto possui uma lógica simples: através da leitura dos Piezos elétricos é possível detectar quando ocorreu de fato uma batida e assim mandar um comando para que a Shield GPRS envie um alerta para a emergência. Abaixo pode-se conferir um fluxograma explicando a lógica de programação.

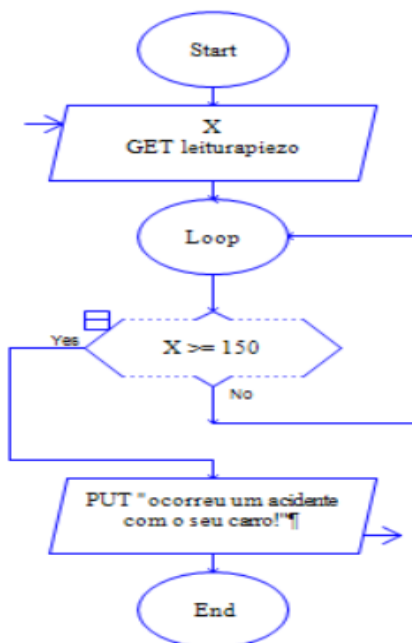


Fig. 9. Fluxograma.

O componente Piezo possibilita a detecção de batidas pelo fato de que ele capta vibrações, assim através da leitura das vibrações é possível detectar padrões para quando o carro está andando normalmente e quando ocorreu de fato um acidente. Abaixo é ilustrada a leitura de vibrações pelo componente através de um gráfico.

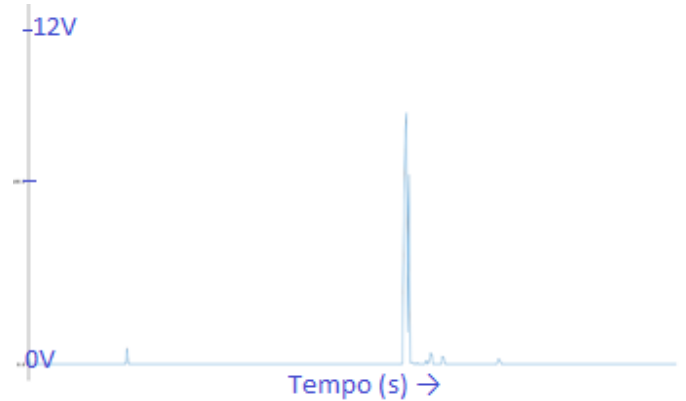


Fig. 10. gráfico da leitura do componente piezo.

Pelo gráfico é possível notar que quando ocorre algum impacto com o veículo a leitura assume valores muito maiores do que o habitual. Neste caso, os picos que aparecem no gráfico foram obtidos quando ocorreu algum impacto na superfície em que o Piezo estava.

Tendo em mente o funcionamento do algoritmo é aconselhável que os sensores sejam posicionados em locais do veículo que estejam expostos à impactos, como no para-choque, caso contrário o funcionamento do equipamento será afetado. Também é necessário que a antena do Shield GPRS não esteja coberta, pois isso poderá dificultar a comunicação da placa com a rede telefônica.

D. TESTES

Os testes realizados no decorrer do desenvolvimento foram feitos em ambiente real, pelo fato de que ainda não existem ferramentas virtuais eficientes para testar o comportamento de sensores de impacto nas situações em que o projeto irá ser implementado.

Para assegurar que o protótipo final estivesse funcionando com uma margem de erros mínima foram realizados testes com diversos tipos de sensores durante uma semana e após ter sido escolhido o componente piezoelétrico para ser embarcado no hardware final passou-se mais uma semana testando o comportamento de tal sensor.

Além dos testes de impacto também foram feitos testes com o Shield GPRS em diversos locais e com diversas fontes de alimentação, devido ao fato de que este demanda uma fonte de 9V e 1A para o seu funcionamento. Como nesta primeira fase de desenvolvimento o produto não será implementado em carros reais foi adotada uma fonte de 12V e 1A para a alimentação.

E. Restrições

Como o produto depende de redes de comunicação para o seu funcionamento é importante ressaltar que este não irá conseguir enviar alertas em locais onde a rede de telefones não tem alcance. Também é necessário que a antena do GPRs não esteja coberta, pois isso poderá prejudicar o seu funcionamento.

Quanto aos sensores é recomendável que estes estejam em áreas que irão sofrer impactos durante a batida, caso contrário estes não irão captar o impacto durante um acidente. Isso se deve ao método de funcionamento dos sensores, tal método está explicado na sessão de algoritmo e lógica de funcionamento.

IV. RESULTADOS

Após um mês de desenvolvimento e alterações nos protótipos iniciais chegou-se no resultado esperado, o grupo conseguiu desenvolver uma tecnologia que de fato é capaz de detectar acidentes e enviar mensagens de alerta para familiares ou centros de socorro.

Tentou-se implementar a função de enviar a localização do carro junto com os alertas, contudo devido ao tempo não foi possível fazer tal implementação, pelo fato de que esta demandaria mais pesquisa e estudo na área de Shields e um replanejamento para a alimentação do protótipo final, que iria requerer uma fonte diferente da que foi adotada no produto final.

Também foram estudadas outras possibilidades de sensores que seriam mais ideais para a implementação do projeto em veículos reais como acelerômetros, sensores gravitacionais, giroscópios, sincronização com o sistema de Airbags, etc. Contudo, devido ao tempo disponível para desenvolvimento e dificuldades com tais sensores preferiu-se a utilização de Piezos eletrônicos por serem mais simples e se mostrarem uma opção viável para o produto inicial.

V. CONCLUSÕES E PERSPECTIVAS FUTURAS

Tendo em vista os resultados obtidos no tempo disponibilizado para desenvolvimento pode-se concluir que o produto final consegue cumprir com o escopo do projeto, podendo ser implementado em situações reais e apresentando uma média de erros menor que 5%, de modo que já pode ser considerado como uma solução viável para o problema.

Futuramente poderão ser inclusos mais recursos que irão garantir maior segurança para os motoristas, tais recursos já foram pensados pela equipe no início do desenvolvimento e envolvem: Sincronização com o sistema de AirBags dos veículos, criação de aplicativos móveis que possam entregar uma interface de usuário mais intuitiva para o consumidor final, disponibilização de localização do veículo, sincronização com sistemas de mapa como o Google Maps, emissão de alertas para serviços de guincho e de bombeiros mais próximos do acidente, criação de um algoritmo que classifique diversos tipos de batidas conforme o risco que estas representam para as pessoas envolvidas, etc.

Nota-se que o projeto de fato é algo muito útil e importante para a sociedade, de modo que qualquer pessoa futuramente poderá sentir-se mais segura caso seja vítima de um acidente automotivo. As possíveis melhorias que podem ser implementadas

em versões futuras são apenas uma amostra da capacidade que este produto tem para melhorar a vida das pessoas.

Por fim chega-se à conclusão que além de ter um desempenho favorável, a primeira versão do produto já tem capacidade de ser comercializada e poderá beneficiar toda a sociedade e uma indústria, principalmente se analisarmos que no futuro tecnologias como esta irão poder ser embarcadas também em carros autônomos, deixando-os assim ainda mais autônomos.

AGRADECIMENTOS

A conclusão do projeto atual não teria sido possível sem o apoio e auxílio das seguintes pessoas:

Professor de Introdução à Automação Marcelo Roberto Peetry do departamento de Engenharias UFSC Blumenau, por avaliar o relatório, fornecer hardware e aconselhar em função do avanço das ideias, isso tudo, além de dar a oportunidade de realizá-las;

Professor de Introdução à Informática para Automação Mauri Ferrandin do departamento de Engenharias UFSC Blumenau, por auxiliar na parte de programação;

Paulo Lange Engenheiro Eletrônico que também aconselhou durante a construção do protótipo;

Colega de classe, Gabriel Alves Ferraz por auxiliar na produção do vídeo;

Por fim, agradecemos a todos que participaram diretamente e indiretamente no desenvolvimento deste projeto.

REFERÊNCIAS

- [1] FILIPEFLOP. Enviando SMS e Fazendo Chamadas com o Arduino GSM Shield. 2016. Disponível em: <<http://blog.filipeflop.com/wireless/tutorial-arduino-gsm-shield.html>>. Acesso em: 15 jun. 2016;
- [2] ARDUINO. Toc-toc. 2016. Disponível em: <<http://playground.arduino.cc/Portugues/LearningKnock>>. Acesso em: 10 jun. 2016;
- [3] AUTODESK CIRCUITS. Electronics from beginner to pro. 2016. Disponível em: <<https://circuits.io/>>. Acesso em: 14 jun. 2016;
- [4] TOLEDO, Benedito Lucas. Tutorial Arduino : GSM / GPRS SHIELD (SIM900) Enviar e receber SMS Arduino Uno Tutorial. Disponível em: <<http://urbuzz.club/trends/watch/vid88EP2R-qiCg2k>>. Acesso em: 10 jun. 2016;
- [5] VASSALLO, Francisco Ruiz. Formulário de eletrônica: todas as leis fundamentais da eletricidade e da eletrônica. São Paulo: Hemus, 1978;
- [6] VIBRODATA. Sensor piezo eletrônico. Disponível em: <<http://www.vibrodata.com.br/imagens/informacoes/sensor-piezoelétrico-01.jpg>>. Acesso em: 25 jun. 2016;
- [7] HALILCAKAL. Gprs shield. Disponível em: <<http://www.halilcakal.com/wordpress/wp-content/uploads/2014/12/sim900-gprs-shield.jpg>>. Acesso em: 25 jun. 2016;
- [8] ROBOCORE. Arduino uno. Disponível em: <https://www.roboCORE.net/upload/lojavirtual/120_1_H.png>. Acesso em: 25 jun. 2016;
- [9] GLOBO, G1. Banda larga deve ser novo foco das telecomunicações no Brasil. Disponível em: <<http://g1.globo.com/tecnologia/noticia/2016/04/banda-larga-deve-ser-novo-foco-das-telecomunicacoes-no-brasil.html>>. Acesso em: 25 jun. 2016;
- [10] VIASSEGURAS. Os acidentes: estatísticas. Disponível em: <http://www.viasseguras.com/os_acidentes/estatisticas/estatisticas_nacionais>. Acesso em: 25 jun. 2016;

CNC

DENER KRAUS, KEIVE ALEXANDRE TIRONI, MATHEUS JONES SEBERINO DA SILVA, RODOLFO FLORIANI MACHADO

UNIVERSIDADE FEDERAL DE SANTA CATARINA – CAMPUS BLUENAU
ENGENHARIA DE CONTROLE E AUTOMAÇÃO
Blumenau, Brasil

Resumo—Nas indústrias, ou em qualquer outro lugar que necessite da produção de algum material, os benefícios da automação industrial podem ser considerados ilimitados. É possível aumentar significativamente o seu desempenho na empresa, automatizando seus processos. Este é um dos motivos responsáveis por fazer com que a automação industrial seja cada vez mais popular em todo o mundo. Algumas empresas vêm demonstrando dificuldades em obter um produto de forma rápida e eficiente, visando sempre na qualidade de seus produtos.

Com este propósito nosso trabalho será baseado na construção de um CNC utilizando da plataforma Arduino para sanar a dificuldade na confecção de placas de circuitos eletrônicos, mostrando suas vantagens perante a confecção à mão livre, sistemas correlatos, metodologia utilizada para criação, ferramentas e especificações técnicas.

Seu funcionamento eletrônico consiste na programação utilizando o programa Grbl Controller que é um software que foi projetado para enviar G-CODE para máquinas CNC, que inclui um poderoso gerenciamento de aceleração, fazendo assim com que se possa planejar suas velocidades evitando curvas acentuadas malfeitas.

Para construção física foi utilizado de EasyDrivers que são controladores de motor de passo de baixo custo, e estruturas de CD/DVD que foram reaproveitadas. Com o projeto implantado podemos obter benefícios como agilidade no processo de fabricação, devido ao aumento na precisão e a qualidade do produto, tendo seu acabamento final com uma qualidade superior ao de um circuito feito à mão.

I. INTRODUÇÃO

Automação é um termo que vem do latim “Automatus” que tem como tradução algo que se movimenta sozinho. Sistema autônomo é por definição um mecanismo que não necessita da interferência humana para funcionar, ou seja, um mecanismo que pode ser utilizado para substituir a manufatura.

Com o aumento exponencial de seres humanos no planeta, evidentemente temos um crescimento da necessidade de adquirir materiais de fácil consumo, portanto, este processo cultural está intimamente relacionado com a evolução das tecnologias utilizadas para a fabricação desses materiais mencionados.

Com a demanda crescente, as empresas tendem a agilizar a fabricação, utilizando novas tecnologias, assim modernizando

os meios de produção, que até à algumas décadas eram mecânicos não computadorizadas.

II. OBJETIVOS

O objetivo deste projeto é desenvolver um robô cartesiano CNC (Computer Numeric Control em tradução livre Controle de Código Numérico Computadorizado) para plotar desenhos de circuitos em placas de fenolite.

III. JUSTIFICATIVA

A crescente busca por agilidade na produção das empresas, gera a necessidade de evoluir as tecnologias utilizadas na fabricação. E em casos onde a precisão e a agilidade são os requisitos básicos para fabricação de circuitos elétricos, a melhor solução são robôs. Como várias empresas utilizam deste artifício para se manterem como grandes produtoras e ainda competitivas, isso torna o mercado de CNCs um mercado rentável, no entanto o custo da aquisição desses equipamentos é muito alto.

Sabendo disto, este projeto traz a solução de baixo custo para uma CNC que agiliza o processo de fabricação, aumenta a precisão e ainda o acabamento final ficara com uma qualidade superior ao de um circuito feito à mão.

IV. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão abordados os conceitos e a utilização da plataforma Arduino, para o desenvolvimento do robô cartesiano CNC.

A. Vantagens da utilização de CNCs.

Com o surgimento dos equipamentos CNCs, os trabalhos passaram a ter benefícios que auxiliaram muito os trabalhadores do ramo. A intervenção do operador relacionada a peças ou produtos é drasticamente reduzida ou eliminada. Muitos CNCs podem fazer um ciclo completo de plotagem, sem nenhum acompanhamento humano, fazendo com que o operador possa desempenhar outras tarefas enquanto o processo é executado. Isto permite ao usuário, vários benefícios como diminuição da manufatura, menores erros humanos, plotagem consistente e em tempo previsível para cada produto.

O nível de habilidade requerido do operador desse sistema, também é reduzido quando comparado a um operador de máquinas com ferramentas convencionais. As máquinas de

hoje se destacam na precisão incrível das especificações e também quanto a repetitividade. Isto significa que uma vez que um programa esteja testado e aprovado, podem ser produzidos outros vários produtos idênticos facilmente com precisão e consistência adequadas. Os CNCs reduziram também o número de erros humanos, fazendo com que se aumente a qualidade dos produtos e diminua o retrabalho e o desperdício.

Os processos nas linhas de montagem aumentaram devido a agilidade e a flexibilidade trazidas pelo equipamento. A mesma linha poderia ser modificada de uma forma com que se produzisse outro produto em um tempo mais curto, quando comparado com os processos tradicionais de produção.

B. Ferramentas

Nesta seção será apresentado as principais ferramentas utilizadas para facilitar a implementação deste projeto incluindo a plataforma Arduino, EasyDrivers e os softwares na sequência correta de trabalho, como demonstrado abaixo.

1) *Arduino*: O Arduino é um microcontrolador, pode ser considerado por definição um computador dedicado, que pode ser utilizado para controlar e dar inteligência para sensores e atuadores mais simples, tornando assim desnecessário a utilização de um computador convencional para coordenar hardwares e componentes mais simples, tornando todo o conjunto da solução mais barato. O Arduino tem sua licença de software e hardware livre e está aberto para uso e contribuição de toda sociedade. O conceito Arduino surgiu na Itália, em 2005, com o objetivo de criar um dispositivo que fosse utilizado em projetos/protótipos construídos de uma forma menos dispendiosa do que outros sistemas disponíveis no mercado.

a) Propriedades:

- Fornecer segurança - mesmo que a Arduino parasse de fornecer placas amanhã, outros fabricantes estariam oferecendo clones funcionais, e o hardware é de design livre. Mesmo se mudarem sua concepção física, não seria muito difícil fazer um conversor para adaptar novo shields (são placas que encaixamos em cima do Arduino para aumentar sua funcionalidade);
- Baixo Custo - Pode ser encontrado no mercado por até R\$50.00;
- Software tem suporte para várias plataformas - Microsoft Windows, Mac OS X e Linux.
- Linguagem simples - Os desenvolvedores do Arduino fizeram-no de maneira que o desenvolvimento seja de fácil para usuários não avançados, porém flexível o bastante para usuários avançados poderem utilizá-lo
- Software livre - O Arduino é completamente um software livre. Se quiser desenvolver seu próprio software ou modificar o original, você tem a total liberdade.

b) *Tabela com propriedades*: A tabela I apresenta as especificações técnicas do Arduino Uno.

Tabela I - Tabela de especificações técnicas do Arduino Uno

<i>Propriedades técnicas</i>	
Microcontrolador	ATmega328P
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limite)	6-20V
Pinos E/S digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entradas analógicas	6
Corrente CC por pinos E/S	40 mA
Corrente CC para o pino 3,3V	50 mA
Memoria Flash	32 KB (ATmega328) dos quais 0,5KB são utilizados pelo bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade de Clock	16 MHz

2) EasyDrivers

O EasyDriver é um controlador destinado para motores de passo, utilizando o G-CODE transmitido pelo Arduino para enviar pulsos elétricos para motores. Compatível com qualquer sistema que possa dar saída de um pulso digital de 0 a 5V. Necessita uma alimentação de 7 a 30V para alimentar os motores, que podem ter qualquer voltagem. Ele possui um regulador de voltagem embarcado para a interface digital que pode ser ajustado para 5 ou 3,3V. A figura 1 apresenta uma visão geral do EasyDriver.

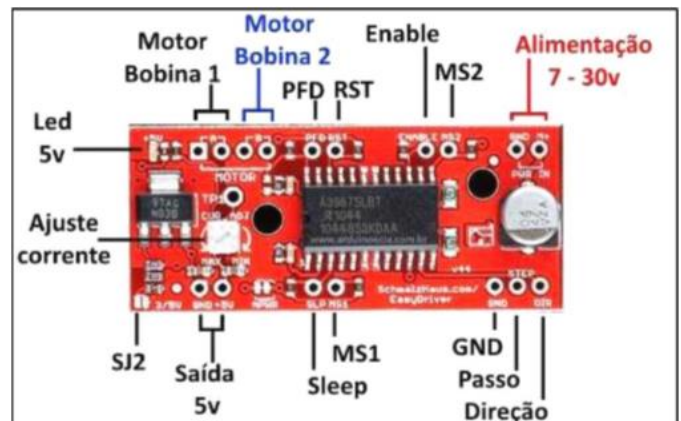


Fig. 01 – EasyDriver.

3) Softwares utilizados

Os softwares são utilizados para poupar trabalho na implementação de códigos de programação, tornando ainda mais barata e simples a resolução deste projeto. Seguindo este

preceito, utilizamos os softwares descritos abaixo na ordem do fluxograma de execução descritos na figura 02.

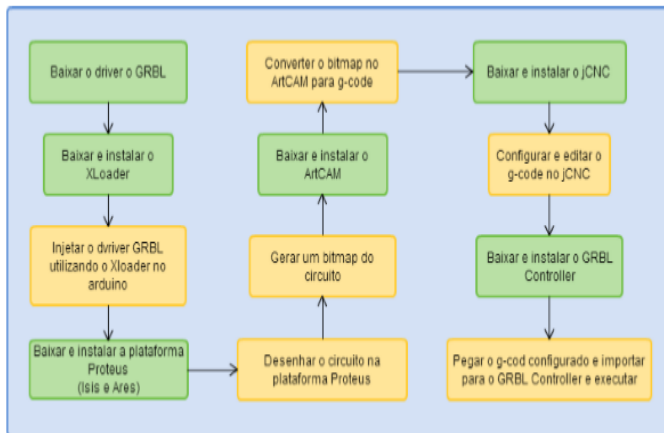


Fig. 02- Fluxograma Softwares

a) Plataforma Proteus

O programa Proteus da empresa Labcenter Electronics, é uma suíte de desenvolvimento que agrega um programa para criação de circuitos, o ISIS, é um programa para desenho de placa de circuito impresso, o ARES. Com a plataforma Proteus é possível a simulação SPICE de circuitos, além da simulação de microcontroladores diversos, podendo executar o seu firmware com opções de Debug e acesso aos registradores do microcontrolador. Pode-se desenvolver o hardware Arduino baseado em Atmega168, Atmega328 ou Arduino Mega 2560. O ponto forte está no compilador integrado que permite o desenvolvimento do firmware e a simulação passo a passo, que não é possível dentro da IDE do Arduino. Além disso é possível ver os valores das variáveis e registradores do micro controlador durante a simulação. Possui diversos recursos que agilizam o processo de criação e testes durante o desenvolvimento de um projeto. Com a integração de um editor de código e a compilação do firmware, tornase ainda mais poderoso. Além da criação e simulação, o Proteus possibilita realizar a prototipagem da placa de circuito impresso, gerando visualização tridimensional e o layout físico do projeto. Para realizar todas essas tarefas o Proteus contempla duas interfaces: ISIS e ARES.

b) Isis

O ISIS (Intelligent Schematic Input System) permite desenhar o esquema elétrico do circuito a ser realizado com componentes muito diferentes, a partir de resistências simples, a um ou outro microprocessador ou microcontrolador, incluindo as fontes de alimentação dos geradores de sinais, e muitos outros componentes com desempenho diferente. Com ISIS os projetos podem ser simulados em tempo real pelo módulo VSM diretamente associado com ISIS, como demonstrado na figura 3. Através deste ambiente é possível

inserir componentes animados ao circuito, permitindo assim uma simulação mais real e amigável.

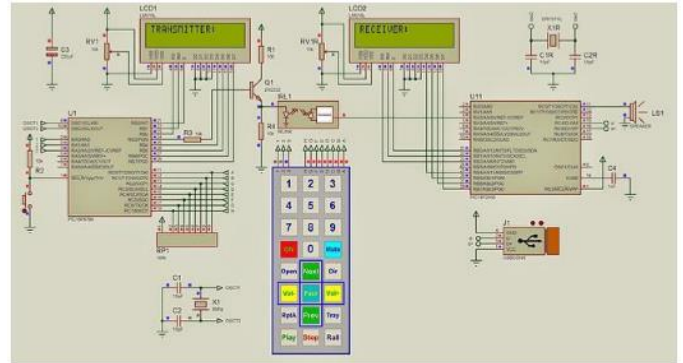


Fig. 03 – Circuito eletrônico ISIS

c) Ares

Na interface Ares (Advanced Routing and Editing Software) é possível realizar o layout de sua placa, determinar a largura das trilhas, o diâmetro das ilhas, o número de layers, inserir planos de terra, legenda, determinar furações, tamanho físico, tipos de encapsulamento, gerar arquivo gerber(obtém todas as informações das coordenadas para a geração de fotolitos) e lista de componentes. Além disso tudo ainda é possível gerar uma visualização tridimensional de seu circuito antes da fabricação, demonstrado na figura 04.

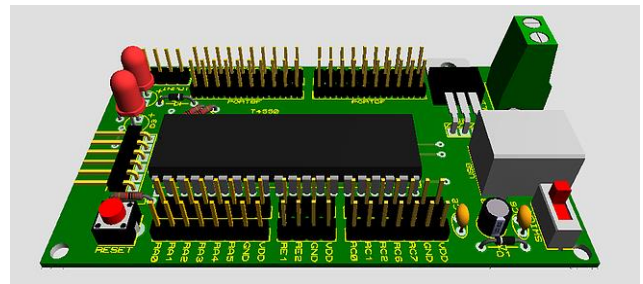


Fig. 04 – Visualização tridimensional no ARES

d) Artcam

Após realizarmos o layout na interface Ares, o próximo objetivo é transformar o PCB (Placa de circuito impresso) no G-código, que é a linguagem de programação que irá comandar nossa máquina operatriz de Controle Numérico Computadorizado.

Para isso iremos utilizar o Artcam que é um software exclusivo que permite produzir produtos de alta qualidade de forma livre 3D a partir de esboços ou imagens, demonstrado na figura 5. Tudo o que você desenha, está pronto para a fabricação. Este software suporta um número de processos de fabricação, incluindo maquinação CNC, corte a laser e prototipagem rápida.

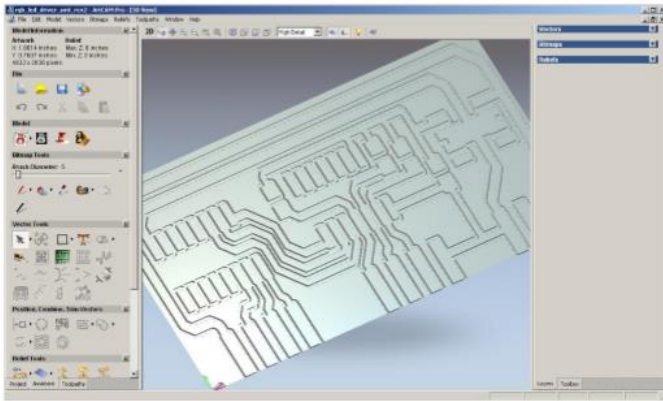


Fig. 05 – Modelo 3D da placa de circuito impresso.

Com o surgimento do Comando Numérico foi necessário se desenvolver uma linguagem entendível pelos controles das máquinas. Gcode é uma linguagem de programação criada para comandar este sistema. O processo é definido por instruções sobre para onde ir, a velocidade do movimento, e qual é o caminho para se mover. Também são chamados de códigos de preparação, e também é qualquer palavra em um programa CNC que começa com a letra G. Geralmente, ele é um código dizendo para a máquina que tipo de ação a ser executada, tais como:

- Movimento rápido (transportar a ferramenta o mais rápido possível através do espaço para o local onde ela vai cortar);
- Alimentação controlada em uma linha reta ou arco;
- Série de movimentos de alimentação controlados, que resultariam em uma peça a ser furada, um corte da peça de trabalho encaminhados a uma dimensão específica, ou um perfil ou contorno, que seria a forma adicionada à borda de uma peça de trabalho;
- Definir informações da ferramenta como no deslocamento;
- Mudar sistemas de coordenadas.

e) XLoader

É um injetor usado para inserir um driver/software no Arduino, sendo este driver o responsável por fazer com que o microcontrolador reconheça G-CODE como coordenadas, assim, transformando o conjunto em um CNC.

f) Grbl Controller

Grbl Controller é um software que foi projetado para enviar Gcode para máquinas CNC, como fresadoras 3D. Não é um software com interface muito sofisticado, afinal ele só precisa disponibilizar ao usuário uma maneira fácil de comandar os motores de passo e seus drivers. É escrito utilizando as bibliotecas cross-platform Qt (estrutura multiplataforma para desenvolvimento de interfaces gráficas). Ele também recebe alguma ajuda do QextSerialPort, que é

uma biblioteca que simplifica a escolha da porta serial USB correta.

Grbl inclui um poderoso gerenciamento de aceleração. Isso significa que o controlador vai estar até 18 movimentos a frente, assim podendo planejar suas velocidades evitando curvas acentuadas malfeitas.

g) jCNC

Este software é semelhante ao GRBL Controller porem tem algumas funcionalidades extras. A comunicação com o controlador da máquina é feita via USB. O software trabalha com G-CODE padrão, fazendo com que o código possa ser criado em um editor com destaque de sintaxe, podendo ser editado e visualizado simultaneamente em 3D. Com esta funcionalidade de poder visualizar e editar podemos ver os erros com antecedência assim permitindo a correção imediata e facilitada evitando desperdícios gerados por erros.

V. METODOLOGIA

Nessa seção será apresentado a descrição do problema, proposta do projeto, solução implementada e o funcionamento do sistema CNC. Será abordado também as ferramentas que serão utilizadas e análise do projeto com requisitos físicos.

A. Descrição do problema

O motivo da elaboração e construção de um sistema CNC para confeccionar placas de circuito impressos, surgiu da dificuldade em produzi-las manualmente, devido não ter a precisão necessária na hora de desenhar as trilhas de circuitos e o tempo gasto para sua confecção. No caso de a placa ser usada para circuitos de alta frequência, se as linhas de circuitos não estiverem bem alinhadas, ou estiverem muito largas podem ocorrer interações e interferência eletromagnética, que poderá causar consequentemente alguma realimentação, ou oscilação indesejável do circuito eletrônico. Podemos verificar na figura abaixo.



Fig. 06. - Placa de circuito plotado manualmente.

B. Descrição da proposta

O projeto tem como objetivo geral construir um equipamento com sistema CNC capaz de confeccionar placas de circuito impresso de uma forma mais precisa e confiável.

Para a execução da tarefa, iremos utilizar os seguintes componentes para montagem do projeto:

- Uno Arduíno: Placa de micro controlador baseado ATmega328p;
- EasyDriver: Controlador de motor de passo submetido ao G-Code;
- Placa de ensaio: Placa de ensaio ou matriz de contato, é uma placa com orifícios e conexões condutoras para montagem de circuitos elétricos experimentais, esposta na figura 07.

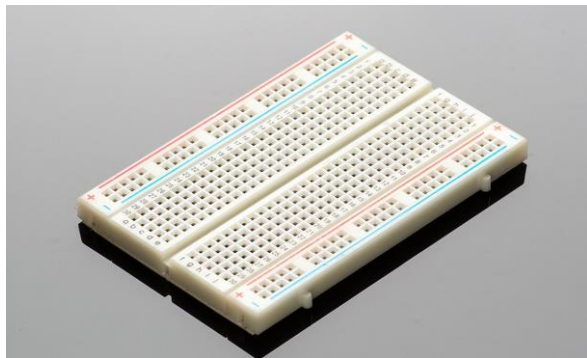


Fig. 07. – Placa de ensaio.

- Motor de Passo com fuso, exposto na figura 8:

1) *Motor de passo*: Dispositivos de malha fechada, ou seja, recebem um sinal de controle, que verifica a posição atual para controlar sua movimentação, indo para a posição desejada com velocidade monitorada;

2) *Fuso*: Peça mecânica feita em material resistente em formato de espiral cilíndrica, que serve como guia para movimentar objetos utilizando atuadores ou motores, para isso ocorrer com êxito, o fuso precisa ter sido forjado com muita precisão.



Fig. 08 – Motor de passo com fuso.

C. *3x leitor DVD/CD, exposto foto 09: Utilizado na parte estrutural para montagem das mesas do projeto é constituído por:*

- 1) *Corpo de Metal;*

- 2) *Motor de passo;*
- 3) *Fuso;*
- 4) *Guia de metal;*



Fig. 09- Estrutura DVD/CD.

D. *Fio de Jumper, exposto na Figura 10:*



Fig. 10 – Jumper para ligação.

E. *Descrição da solução*

Predefinida a solução para a implementação do projeto abaixo temos o orçamento com o custo das peças básicas previamente imaginadas com o preço de três fornecedores diferente (tabela II), fazendo assim com que possamos selecionar os locais de compra com maior custo benefício.

Tabela II – Tabela de custos: A tabela II apresenta o orçamento do projeto.

Quantidade	Componente	Custo 01	Custo 02	Custo 03
1	Arduino UNO (ATmega328)	R\$ 69,00 Lab de Garagem	R\$79,00 RoboCore	R\$ 59,00 FelipeFlop
3	Easydriver V4.4.	R\$ 15,90 Mercado Livre	R\$ 66,80 Lab de Garagem	R\$ 24,90 FelipeFlop
3	Estruturas CD/DVD	R\$ 59,90 Kabum	R\$ 51,35 Balão da Informática	R\$ 24,99 Mercado Livre
1	Placa de Ensaio 830 furos.	R\$ 19,90 Lab de Garagem	R\$ 29,00 RoboCore	R\$ 24,90 FilipeFlop
1	Kit Fios de Jumper 20 cm	10 fios R\$ 15,00 Lab de Garagem	10 fios R\$ 10,00 RoboCore	40 fios R\$ 23,80 Mamute Eletronica
1	Kit Leds	Kit 15 leds R\$3,50 Lab de Garagem	Kit 10 leds R\$ 01,90 FilipeFlop	Kit 5 leds R\$ 00,90 RoboCore
2	Barra de Vergalhão Roscado 1 m.	R\$ 04,50 1M Lutec Parafuso e Ferramenta	R\$ 01,84 1M A. Camargo	R\$ 02,30 1M Mercado Livre
75	Porca M6	100 pçs. R\$ 06,00 Lutec Parafusos e Ferramenta		

VI. FUNCIONAMENTO

Nesta seção tendo o projeto devidamente definido e estruturado, será demonstrado abaixo de forma breve mas compreensível o funcionamento do nosso protótipo.

A. Descrição Detalhada

Como já mencionado, o “cérebro” do projeto é o micro controlador Arduino Uno, que é responsável por coordenar o funcionamento dos componentes como por exemplo os motores, os leds e os EasyDrives.

O conjunto mecânico será montado de maneira estratégica, semelhante as impressoras 3D atuais, baseados em planos cartesianos com eixo X (movimento ara frente e para traz), Y (movimento para esquerda e para direita) e Z (movimento para cima e para baixo).

Para funcionar e desenhar:

- Os motores têm que estar conectados perpendicularmente nos fusos (equivalentes aos eixos X, Y e Z);
- Os drivers ligados eletronicamente nos motores e no arduino (como no diagrama elétrico na figura 11);

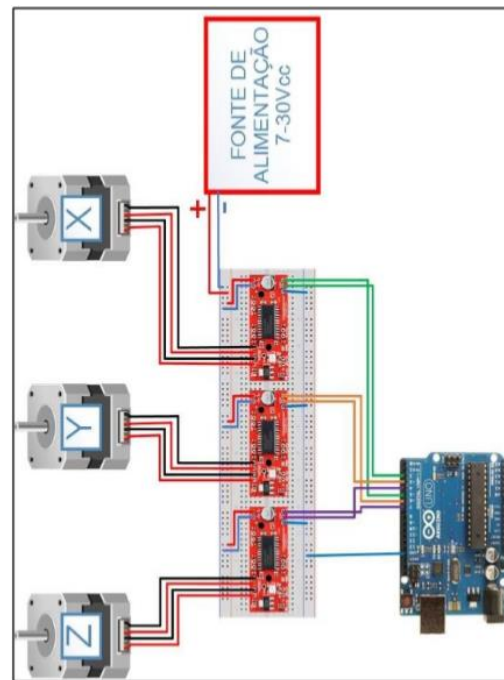


Fig. 11. – Diagrama elétrico.

Funciona com o Arduino conectado a um computador que porta as ferramentas de desenhos virtuais. O computador converte os desenhos em GCODE e envia isso para o Arduino, que por sua vez distribui as funções para cada EasyDriver. Ele manda para cada um em forma de sinais elétricos as coordenadas, e por fim cada EasyDriver manda para seu motor de passo a energia necessária para girar para cada posição, além disso o Driver detecta a posição atual, assim, evitando que o desenho fuja do seu porte ideal, demonstrado na figura 12.

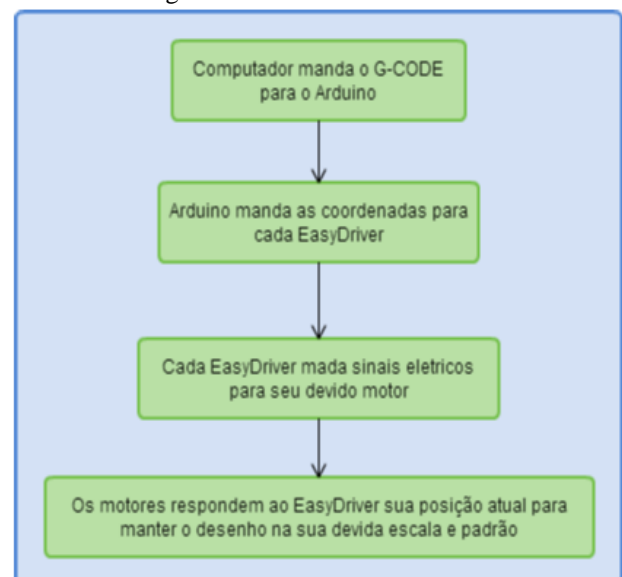


Fig. 12. – Diagrama eletrônico.

CONCLUSÃO

A utilização do robô cnc para a produção de placas de circuitos se torna lucrativa para o empresário. Tendo em vista a importância de se ter um sistema que agilize o processo de fabricação nas empresas, esse robô poderá confeccionar as placas de circuitos com rapidez e precisão, fazendo com que o programador encaminhe o andamento da tarefa para que o robô siga as coordenadas correspondentes para a fabricação das placas. Com a utilização deste equipamento, o empresário irá obter um ganho no tempo e dinheiro, deixando o processo organizado. Cada vez mais procurados para sanar as necessidades da empresa nos métodos de produção, devido a sua alta performance, este tipo de robô consegue cumprir a tarefa que será designado, como desenhar as trilhas na placa com êxito, sem falhas, possuindo grande importância para a obtenção do produto final. Em uma sociedade onde cada vez mais é necessário a produção e entrega rápida de produtos, tem a necessidade do auxílio de robôs para agilizar o processo de transporte. A tendência é que as indústrias utilizem cada vez mais robôs em suas linhas de produção, devido à grande importância no processo de fabricação e pelos seus benefícios.

REFERENCES

- [1] LABORATORIODEGARAGEM. Site Lab de Garagem. 2015. Disponível em: <<http://www.labdegaragem.org/loja/fresadora-cnc/kit-fresadora-cnc.html>>, acesso em: 13 dezembro 2015. .
- [2] EMBARCADOS. Site Embarcados 2015. Disponível em: <<http://www.embarcados.com.br/arduino-proteus-8-1/>>, acesso em: 8 dezembro 2015.
- [3] MODOZERO. Modo Zero. 2015. Disponível em: <<http://www.modozero.com/#!Proteus-VSMAPresenta%C3%A7%C3%A3o/c1kt3/55e45b440cf28ffc7eeced25>>, acesso em: 8 dezembros 2015.
- [4] LABORATORIODEGARAGEM. Site Lab de Garagem. 2015. Disponível em: <<http://www.labdegaragem.org/loja/45-equipamentos/maquina-para-gravac-o-alaser.html>>, acesso em: 8 dezembro 2015.
- [5] ZAPMAKER. Site Zap Maker. 2015. Disponível em: <<http://zapmaker.org/projects/grbl-controller-3-0/>>, acesso em: 9 dezembro 2015.

Projeto Arduino

Porta Automática

Arthur Milheiro, Guillermo Dilly, Gustavo Luz, Maria Eugênia Martins e Tiago Battiston, of Engenharia de Controle e Automação: dept. UFSC

Blumenau, Brasil
arthur.milheiro@grad.ufsc.br
guidilly@hotmail.com
mariaeugenia7675@gmail.com
tiago.battiston@gmail.com

Resumo—A tecnologia está cada dia mais presente em todos os lugares, seja desde as empresas que ela vem tanto para novos produtos como controle de processos, quanto coisas mais do dia-dia como nos domicílios, ou seja, a domótica, ou seja, a automação da casa, onde se busca uma maior comodidade e eficiência das partes da casa, tornando-a toda integrada.

O Arduino é uma ótima tecnologia para realizar esses propósitos. Uma das partes da domótica trata-se do acesso a lugares, fazendo com que não exista mais necessidade de carregar chaves ou outros objetos físicos. Neste trabalho vamos explorar essa área da automação, trabalhando com a abertura de portas a partir de sequências de batidas na mesma.

Palavras chave - Arduino; porta; automação; fechadura; domotica; batidas.

I. INTRODUÇÃO

Neste trabalho será apresentado um projeto realizado com o dispositivo eletrônico Arduino Uno. Desse modo, o grupo teve que pensar em um problema para selecioná-lo utilizando o Arduino, assim sendo necessário explorar conhecimentos de eletrônica, programação além de sua criatividade para decidir a melhor solução para o problema escolhido.

Para isso foi preciso pensar nos vários ramos que a automação e controle possui e levar em conta o que seria possível fazer no tempo determinado. Depois de uma análise em problemas e soluções para o dia-dia o grupo escolheu o ramo da domótica (automação residencial), para achar um problema envolvendo-a, assim teve-se que pensar em maneiras que deixaria os domicílios melhores, mais práticos e conectados.

Então dentre vários problemas cotidianos que precisam ser resolvidos escolhemos resolver problemas envolvendo portas, assim pensando em jeitos de melhorar seu uso e praticidade, para isso desenvolvemos modos inovadores de abri-las. Desse modo facilitando e diferenciando a entrada e

saída das pessoas de um ambiente para o outro, visando a praticidade e a segurança.

II. DESCRIÇÃO DO PROBLEMA

Embora muito presente na sociedade atual a chave já está muito ultrapassada comparada com a evolução tecnológica. Então, com a intenção de substituir tal mecanismo, nós da X desenvolvemos um dispositivo com muito mais usabilidade na hora de abrir a porta.

Problemas com chaves:

- Possibilidades de serem quebradas na fechadura;
- Esquecendo elas nos lugares;
- Carência de várias cópias da mesma chave.
- Necessidade de ter muitas chaves no dia a dia para cada ocasião

Problemas com o acesso a porta:

- Poder permitir a entrada de alguém quando você não está apto a abrir a porta, seja por estar longe do local ou ocupado.
- Uma das mãos ocupadas (como o caso de estar voltando do mercado).

III. DESCRIÇÃO DA SOLUÇÃO

Desse modo se faz necessário ter mais opções, além da chave, para destrancar a porta estando perto ou longe dela. Por isso, nós a empresa X fizemos um projeto de domotica para solucionar os problemas descritos anteriormente.

No nosso projeto, se dá uma solução a partir de um mecanismo que abre e fecha a porta automaticamente, através de uma sequência específica de batidas na porta, podendo colocar mais de uma senha, assim podendo ter senhas

diferentes para cada pessoa da casa, além de um alarme próprio.

A. Parte Física

Uma das principais partes do projeto é o Arduino Uno, Fig. 1, que realizará a parte lógica do mecanismo, assim sendo programado ele atuará como dispositivo que interpretará as informações coletadas e tomará decisões que fará abrir a fechadura e realizar toda a parte lógica da interface homem-máquina.

A escolha pelo arduino Uno no lugar de outros modelos de arduino pois, mesmo sendo com um processador mais baixo e menos portas que outros modelos como o Mega ou Due, ele é de baixo custo e suficiente para esse projeto.

O Arduino Uno é um microcontrolador baseado no ATmega328P, ele possui 14 portas digitais sendo 6 que podem ser usadas como saídas PWM, conexão USB. Ele é utilizado principalmente para criar protótipos, indo desde trabalhos simples a trabalhos realmente simples, até projetos complexos, desde que as portas disponíveis sejam suficientes e não ultrapasse sua corrente máxima.

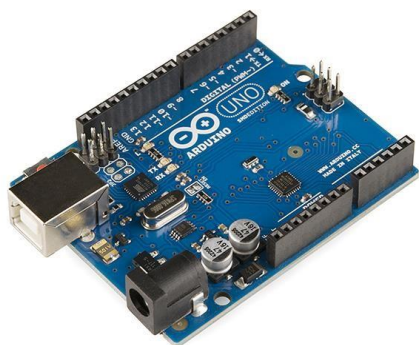


Fig. 1. Arduino Uno.

Para fazer a recepção da intensidade sonora das batidas da porta utiliza-se um sensor de Som KY-038 Microfone, Fig. 2, que irá fornecer os dados sonoros para o Arduino processa-los.

O sensor tem um funcionamento simples, quando detecta o som o microfone varia a tensão na saída analógica dele e aciona também a saída digital, assim medindo se há ou não som e sua intensidade, a saída digital pode ser ajustada a partir da regulagem do potenciômetro presente no módulo



Fig. 2. Sensor de som KY-038 microfone.

Para fazer a conexão do Arduino com a internet, assim fazendo com que possa ser operado por computador ou celular, o Módulo WiFi ESP8266 ESP-01 para Arduino, Fig. 3, assim possibilitando que como segunda opção para abrir portas, possibilitando fazê-lo mesmo de longe.

O módulo pode trabalhar tanto como um ponto de acesso quanto com uma estação mandando e recebendo dados, sua conexão com o arduino se dá via serial utilizando tanto os pinos TX quanto RX, podendo ser configurado por comandos AT.

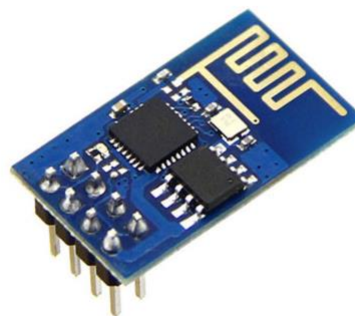


Fig. 3. Módulo WiFi ESP8266 ESP-01.

Para a melhor implantação do projeto, utilizou-se de um fecho Elétrico HDL, Fig. 4, que basicamente é uma tranca de porta que funciona ao ser alimentado com um pulso elétrico que abrirá o fecho. Para interligar o arduino com o fecho assim fazendo só mandar um pulso por vez e não um constante o é utilizado um módulo relé, Fig. 5, que ao ser chaveado libera a corrente para o fecho HDL.



Fig. 4. Fecho elétrico HDL



Fig. 6. Sinalização e Push-Button

Além disso o projeto possui uma caixa para realizar o suporte para o Arduino e o sensor de som de modo que fique com um bom aspecto estético e simplificado,

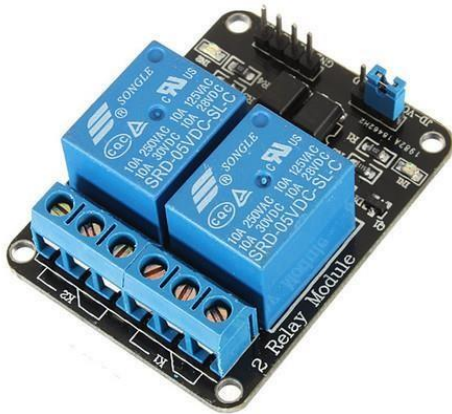


Fig. 5. Modulo relé

Neste projeto utiliza-se LEDs, os quais sinalizam sobre se o comando foi colocado certo, se está preparado para receber a senha. A utilização de um Buzzer, tanto serve como alarme após o número de batidas erradas e também para o sistema de lembrar toques já gravados. Desse modo fazendo com que o uso deste projeto seja de fácil entendimento. Para realizar procedimentos como redefinir senhas e ouvir a senha a interface com chaves Táctil Push-Button (Fig. 6), são utilizadas.

B. Parte Lógica

Neste projeto foi necessária uma programação que faça que o arduino interaja com os sensores, e com isso fazer medição e tomando decisão além de acionar sinalizadores para alertar o comportamento dele. A baixo tem-se de modo simples a lógica do programa feito na linguagem do arduino para que ele realize suas ações.

Logo que ligado, o programa entra no modo de espera, acendendo o led azul, que irá ficar aceso demonstrando que está esperando alguém fazer a sequência de batidas na caixa que a partir do tempo entre essas, que serão detectadas pelo sensor de som, irá grava-la. Assim o programa vai entrar em modo de espera pela senha para abrir a porta.

Se for decidido por abrir a porta, será necessário inserir novamente a senha já salva e caso esteja correta com a gravada, irá sinalizar que a porta abriu pelo led verde e acionará o relé para abrir o fecho. Caso tenha sido usado uma combinação diferente da salva irá sinalizar a partir do led vermelho que a senha está errada além que o fecho não irá se abrir. Se acaso for feito 5 tentativas consecutivas erradas, será acionado o alarme no qual só pode ser desligado por dentro da caixa.

Também há possibilidade de criar mais de uma combinação, na verdade até 3, nas quais as 3 serão possíveis poderão abrir o fecho. Para adicionar outra combinação, e preciso estar no modo espera de uma senha, aí ao apertar no botão fora da caixa, acenderá o Led azul assim como na primeira vez que foi inserido e seguirá o mesmo processo.

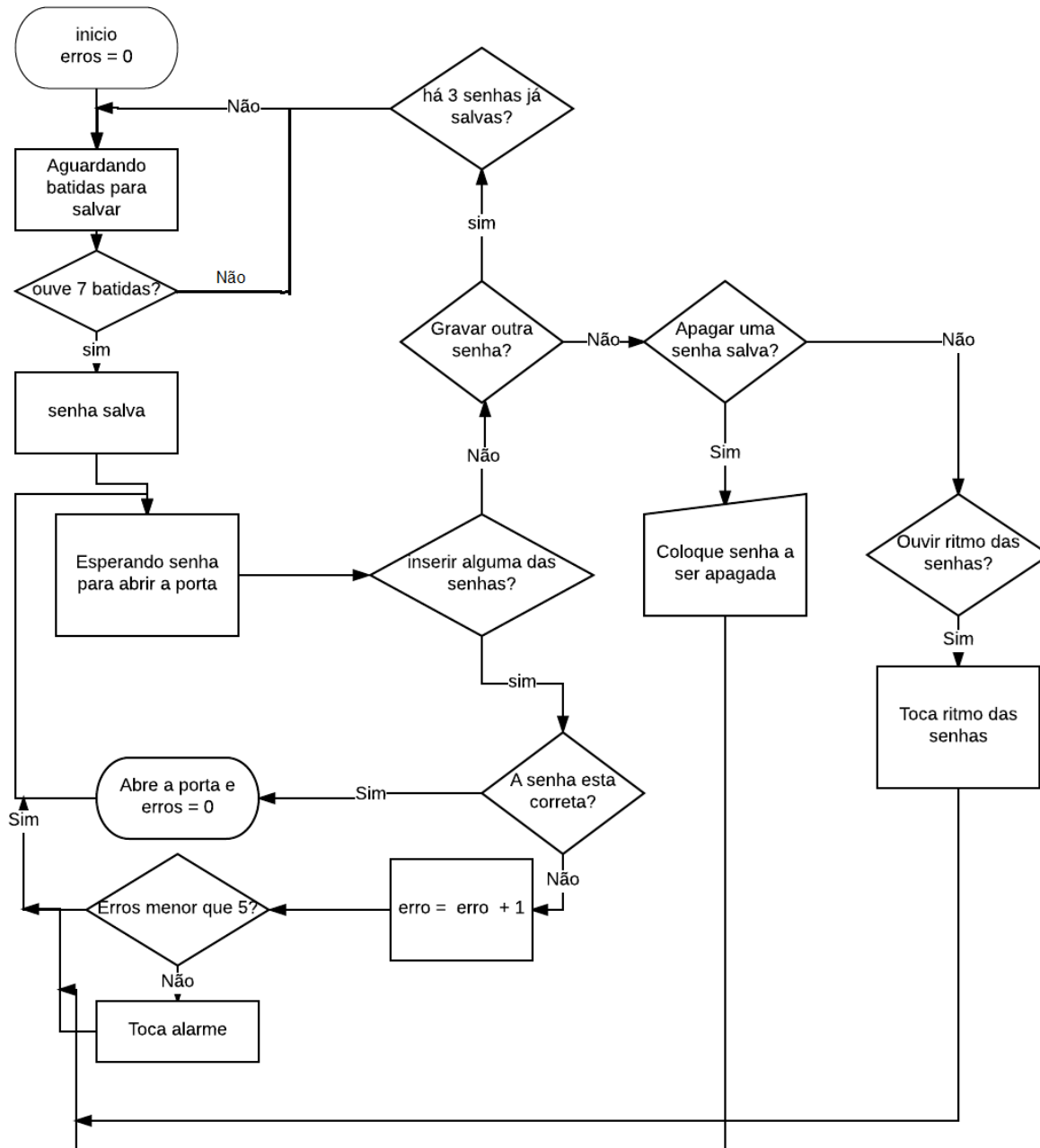
Caso seja necessário a redefinição de alguma das senhas gravadas, ao abrir a caixa e apertar o botão dentro dela ela irá aguardar que seja colocada a senha que se quer se apagada

assim depois de feito esse processo, se alguém tentar usar a senha apagada para abrir a porta, não irá conseguir pois a senha foi apagada ou substituída.

Em caso que as senhas já gravadas sejam esquecidas por algum acaso e assim não tendo como colocar uma nova ou apagar elas, é possível relembrá-las clicando simultaneamente em ambos botões (o de dentro e de fora da caixa), assim fazendo com que o arduino envie o comando que faça o buzzer funcionar no mesmo intervalo de tempo que as batidas, fazendo com que ele toque todas as senhas já gravadas, e com

isso possa usa-las para substituir por outra ou destrancar a porta.

Na imagem a baixo se encontra o fluxograma mostrando mais especificamente o funcionamento do programa, mostrando a toma de decisões do arduino, contagem de variáveis.



C. Custos

Para realizar este projeto, foi necessário a utilização de todos os componentes já citados antes na parte física do projeto e os custos deles estão presente na tabela de custos (Tabela I.) onde demonstra o preço de cada peça além da quantidade usada e o preço total., isso inclui os custos com o projeto, pela sua parte física, com o arduino, sensores etc.

TABELA I- TABELA DE CUSTOS

Peça	Tabela de Custos		
	Quantidade	Preço Unitário	Preço Total
Arduino Uno	1	R\$20,90	R\$20,90
Fecho Elétrico HDL	1	R\$60,00	R\$60,00
Caixa	1	R\$7,00	R\$7,00
Relé	1	R\$16,00	R\$16,00
Módulo WiFi ESP8266 ESP-01	1	R\$34,90	R\$34,90
Buzzer	1	R\$2,00	R\$2,00
Sensor de Som KY-038 Microfone	1	R\$19,90	R\$19,90
Push-Button	2	R\$0,40	R\$0,80
Total			R\$161,70

REFERENCIAS

- [1] ELEMENT14. **Figura 1:** Disponível em: <https://www.element14.com/community/servlet/JiveServlet/downloadImage/38-15926-197962/309-309/Arduino_Uno.jpg>. Acesso em: 23 jun. 2016.
- [2] ARDUINOECIA. **Figura 2** .: Disponível em: <<http://3.bp.blogspot.com/-tWhOleixewE/Var5xDOMtel/AAAAAAAAAD-0/cO2Dpud4mJM/s400/Sensor-de-som-KY-038-detalhe.png>>. Acesso em: 04 jun. 2016
- [3] FABRICADIGITAL.ORG. **Fig. 3:** Disponível em: <<http://fabricadigital.org/wp-content/uploads/COM-011-011.jpg>> Acesso em: 23 jun. 2016.

IV. CONCLUSÃO

Na sociedade contemporânea, com a ajuda da automação, busca facilidades e rapidez para executar os trabalhos do dia a dia, com isso, vemos inovações tecnológicas surgindo todos os dias, e para esse motivo criamos algo inovador que vai melhorar e facilitar a vida das pessoas, uma porta automática, assim não necessitando de objetos para abri-la, mas ainda assim trancando-a garantindo conforto e segurança.

Com sua característica básica de criar maior praticidade para acessar locais, nosso projeto possui grande utilidade para os domicílios, no qual se tornam cada dia mais interligados e automáticos. O ramo da domótica ainda possui muito a crescer, assim como esse projeto, como uma perspectiva futura tem-se em mente a criação de um aplicativo em que receberá notificações quando alguém utiliza a senha errada, assim avisando que tem alguém na sua porta, também enviar quando a porta está aberta por mais de 5 minutos, além de poder destravar a porta com ele, também adicionar um display onde faz a interface para realizar trocas de senhas e demonstrar se a senha usada está certa ou errada.

Este trabalho agregou grandes resultados positivos, pois fez o grupo trabalhar a mente afim de pensar problemas e maneiras de resolve-los. Além de ter feito irmos atrás de o que era necessário para fazer o projeto funcionar e relacionar-se com pessoas que já trabalharam ou trabalho no ramo, assim agregando tanto no conhecimento quanto no nosso social.

- [4] HDL. **Figura 4.** Fecho elétrico HDL. Disponível em: <<http://www.hdl.com.br/sites/default/files/styles/produto-fixo/public/images/produtos/produtos/90.01.03.082.jpg>>. Acesso em: 04 jun. 2016.
- [5] FILIPEFLOP. **Figura 5:** Modulo relé. Disponível em: <<http://s3.amazonaws.com/img.iluria.com/product/6B84A/2B4182/450xN.jpg>>. Acesso em: 20 jun. 2016
- [6] GREENPROPHET. **Figura 6** Disponível em: <<http://i2.wp.com/www.greenprophet.com/wp-content/uploads/2012/09/LED-lights-health-hazard.jpeg?w=560>>. Acesso em: 20 jun. 2016.
- [7] ARDUINO. **Arduino Uno.** Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardUno>>. Acesso em: 23 jun. 2016.

Sistema de fechadura inteligente, baseado em RFID

Álvaro Rodrigo, João Victor Ferreira, Gabriel Alves, Lukas Belck, Nicolas Antero Nunes,
 Universidade Federal de Santa Catarina – Campus de Blumenau
 lukas.belck@grad.ufsc.br
 nicolas.antero@grad.ufsc.br
 gabrield.ferraz@grad.ufsc.br
 joao.victor.ferreira@grad.ufsc.br
 alvaro.rscp@grad.ufsc.br

RESUMO

Objetivo: Criar um protótipo para substituir o atual sistema de concessão de chaves para a utilização da salas e laboratório da UFSC Blumenau. Utilizando a já existente tecnologia de RFID em conjunto com um Arduino que pode se comunicar com a internet.

Métodos: O Arduino utilizará o sensor de RFID/NFC para armazenar TAG's autorizadas em um banco de dados na internet. Opcionalmente é possível colocar um LCD para melhor estética e customização. A partir do banco de dados o professor poderá realizar a chamada de aula e ter um monitoramento de quem está utilizando a sala.

Resultados e Conclusão: O Sensor de RFID teve uma boa performance, lendo os cartões RFID sem problema e de uma distância razoável, constatamos também que é possível armazenar informações dentro dos cartões RFID não apenas no banco de dados, a conexão com a internet pelo wifi não foi possível por isso decidimos usar o ethernet shield (que utilizada a cabo) e atualmente estamos trabalhando para comunicar com o banco de dados.

Palavras-chave: Porta; RFID; Fechaduras; inteligentes; Arduino; Banco; dados; Mysql.

I. INTRODUÇÃO

A Universidade Federal do Estado de Santa Catarina é uma instituição federal, portanto tanto os colaboradores como os alunos têm o dever de manter e preservar tudo que esteja disponível, já que estes são de domínio público. A atual realidade do campus de Blumenau é algo ainda muito limitada estruturalmente, por se tratar de um espaço novo e temporário. Desta forma, as áreas de estudo para melhoria da condição atual da universidade são ilimitadas, partindo destas falhas elaboramos um estudo na nossa área de pesquisa que forneça maior proteção e segurança aos patrimônios da instituição já existentes e aos que ainda estão por ser adquiridos.

O projeto tem por objetivo desenvolver uma forma de acesso as salas e laboratórios do campus da UFSC Blumenau com base em um Arduino e um sensor RFID (do inglês

"Radio-Frequency IDentification"). RFID é um método de identificação automática que usa sinais de radiofrequência para envio e recebimento de dados onde que a partir de um cartão com autorização será possível ter acesso a sala e aos laboratórios para a realização de trabalhos e também de estudos. O projeto proposto se difere do sistema de acesso a sala atualmente usado, onde o aluno com autorização do profissional responsável da sala/laboratório se dirige aos guardas da instituição, onde assina seu nome em um protocolo, para retirar as chaves. Um exemplo dessa tecnologia, é o uso no "Sem Parar" em pedágios, onde a leitura do RFID instalado no carro geralmente como um adesivo, é quase que instantânea. Este é um exemplo dos vários existentes com uso do RFID em nossa sociedade.

O sistema proposto pelo nosso projeto já é utilizado de maneira similar em diversa áreas da sociedade garantindo aprovação e aceitação, pois além de ser um sistema de fácil implementação, também é seguro, prático e com poucas falhas.

A. O problema

O processo para concessão das chaves de salas e laboratórios é feita por intermédio dos seguranças que se localizam na entrada do Bloco A, sendo que a liberação das chaves só é feita se um responsável pela sala em questão tenha autorizado os alunos para utilização da sala/laboratório com antecedência. Os alunos só podem retirar as devidas chaves assim que assinarem o controle de chaves, mesmo que esse sistema venha sendo utilizado desde o início do campus, percebe-se que existem falhas e limitações e que é possível ocorrer uma melhoria como a automatização do sistema de acesso as salas.

B. Descrição de alguns problemas

Eis algumas falhas do modelo anterior que podemos citar e empecilhos a ser solucionados durante o desenvolvimento do projeto:

- A acessibilidade, pois para ter acesso a sala toda vez é necessária que algum aluno com autorização do professor responsável se desloque até o Bloco A para retirar a chave com os seguranças.

- A retirada da chave com os seguranças nem todas as vezes é rápida e eficaz, pois o segurança pode não estar no momento já que essa não é sua atividade principal.
- Quando é devolvido a chave da sala muitas vezes não é necessário assinar que foi devolvido, podendo futuramente causar problemas.
- É necessário a troca da fechadura caso se perca as chaves da sala.
- Falta de informações sobre quem está na sala/laboratório, além da pessoa que retirou a chave com os seguranças.
- Não é feita a conferência todas as vezes que a chave é retirada com os seguranças, podendo alguém sem autorização ter acesso a sala.
- A chave pode já ter sido retirada por um outro aluno fazendo com que a viagem até o bloco A tenha sido em vão.
- Atualmente como a anotação é feita em papeis, os mesmos devem ser mantidos guardados, necessitando de um espaço para arquivar esses documentos. Mesmo assim pode acontecer desses papeis serem perdidos nesse processo.
- Por falta de atenção, algum aluno pode levar a chave para casa.
- O armazenamento de dados coletados, e como será a classificação de entrada e saída no sistema.
- Como a autorização de acesso será fornecida e como ela será retirada.
- Caso o cartão de edição (do professor responsável) seja perdido como será o procedimento para resolução disso.

II. DESCRIÇÃO DA SOLUÇÃO

A solução do problema proposto, as especificações das peças, mostrando os componentes que o protótipo utilizará, a demonstração da montagem indicando onde cada peça se posicionará no projeto, os testes feitos e também a lógica do algoritmo a ser utilizado no arduino.

Em muitos prédios residenciais e comerciais a tecnologia RFID, já é difundida para o acesso dos mesmos. Por isso nossa equipe decidiu utilizar essa tecnologia junto com a acessibilidade do arduino para criar um sistema de acionamento de fechadura por RFID, sendo uma das motivações o fato os cartões de estudantes da UFSC já contêm um sistema de RFID passivo, facilitando assim a utilização desse sistema.

A. O uso do RFID em nossa sociedade

O RFID como dito antes, é um método de identificação automático, esse sistema é amplamente usado em vários

pontos em meio a nossa sociedade, como por exemplo, o sistema de transporte coletivo, e o cartão do “passe” em uma sua grande maioria usa o sistema de identificação do RFID, como cada cartão tem um código diferente e criptografado a esse sistema é bastante seguro e evita que o sistema seja burlado. No caso da utilização no sistema de transporte coletivo basta a pessoa inserir crédito no seu cartão e logo após, o cartão pode ser usado passando o nos leitores dentro dos ônibus desta forma o sistema descontará o valor da passagem do cartão, esse sistema agilizou o uso do transporte coletivo, pois é mais rápido, evitando filas e otimizando o tempo de percurso.

B. Componentes e suas especificações

Os cartões de estudantes da UFSC contêm etiquetas RFID passivas, que utilizando a rádio frequência do leitor para transmitir o seu sinal que contém as informações gravadas no cartão. O projeto conta com outros componentes além do sensor RFID, sendo estes três LEDs (azul, vermelho e verde), uma fechadura eletromagnética, um Arduino com o sensor RFID para a leitura, um rele, uma protoboard para ligações, um ethernet shield wireless e 2 duas fontes uma para o Arduino e outra para a fechadura. É possível ainda adicionar um LCD para identificar as operações. Abaixo na Figura 1 mostra todos componentes utilizados na construção do protótipo:



Figura 1- Principais componentes do projeto.

C. Especificação das peças:

1) Sensor RFID:

- Suporte IC2, SPI e HSU (High Speed UART)
- Suporte a leitura e escrita RFID
- Cartões Mifare, 1k, 4k, Ultralight e DesFire
- Suporte a comunicação P2P (Peer to Peer)
- Distância de leitura entre 5cm~7cm

2) LCD 16x2:

- Comunicação 12c – Utiliza apenas dois pinos do processador
- Não sobrecarrega controlador do Arduino
- Backlight controlado por Software
- Potenciômetro para ajuste de contraste da tela
- Led de ON/OFF

3) Arduino UNO:

- Micro controlador: ATmega328P
- Tensão de operação: 5V e 3.3V
- 14 Pinos Digitais I/O
- Memória Flash: 32kb
- Clock Speed: 16 MHz
- Comprimento: 68.6 mm
- Largura: 53.4 mm
- Peso: 25g

4) Rele 5V:

- Tensão de operação 5VDC
- Corrente típica de operação: 15-20 mA
- LED indicador de status
- Tempo de resposta: 5~10 ms
- Tensão de saída: 30 VDC a 10A ou 250VAC a 10A

5) Esp8266 wifi:

- Chip: ESP8266
- Modelo: ESP-01
- Tensão de operação: 3,3V
- Suporte á redes: 802.11 b/g/n
- Alcance: 90m aprox.
- Comunicação: Serial (TX/RX)
- Suporta Comunicação TCP e UDP
- Dimensões: 25 x 14x 1 mm
- Peso: 7g

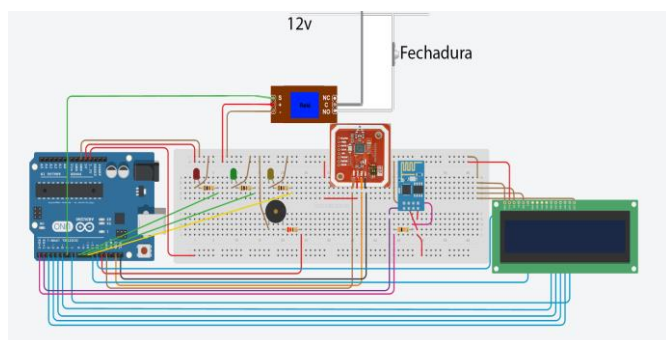


Figura 2- Esquema de ligações do projeto

D. Uso dos componentes

Cada componente terá a sua função junto com a programação do código, basicamente os LEDs serão

necessários para informar o modo e a função que o Arduino está exercendo, o LED vermelho notifica que o cartão não tem autorização no banco de dados, já o verde identifica que o cartão está autorizado e ocorreu liberação da porta, e o azul tem por finalidade informar que o sistema entrou em modo de edição desta forma poderá ser adicionado ou excluído outros cartões no banco de dados, sendo o cartão de edição o único que não poderá ser modificado.

A fechadura eletromagnética manterá a porta trancada, entretanto a partir do momento em que um cartão autorizado é identificado, o arduino libera um pulso elétrico que faz com que a fechadura seja destrancada permitindo acesso a porta que tiver o dispositivo instalado.

O protótipo contará com um buzzer que emitirá sons diferentes dependendo do status do cartão identificado pelo RFID, sendo estes para não autorizado, liberado e ativação do modo de edição. Foram usadas duas fontes um para alimentar o Arduino com tensão de 5v e outra para a fechadura com tensão de 12v.

Outro componente é o ESP8266 para a conexão com a internet, onde serão armazenados em um banco de dados os dados de entrada e saída de acordo com a data e horário que a tag fora identificada, onde a pessoa responsável, poderá monitorar, se a sala está sendo usada ou quem esteve nela, garantindo maior controle e segurança.

III. CUSTOS

A. Introdução

Neste artigo já foi apresentado vários pontos como, componentes, benefício e o problema em si, mas afinal para o que serve um projeto sendo que o custo não é compensatório.

B. Os custos do projeto.

Para visualizar o custo total do projeto está presente a Tabela 1 com o custo unitário das peças assim como o custo total.

Tabela 1: Relação de Custo

Peças	Quantidade	Custo
Arduino UNO	1	R\$ 20,90
LCD	1	R\$ 18,90
Sensor RFID\NFC	1	R\$ 80,00
Fecho elétrico AGL 12V	1	R\$ 54,90
ESP8266 WiFi	1	R\$ 29,90
LED	3	R\$ 0,40

Buzzer	1	R\$ 2,20
Modulo rele 5V	1	R\$ 7,80
Total		R\$ 215,00

Como podemos notar os custos do projeto condizem com a realidade da instituição em que estudamos, com poucos recursos para investimento e grande burocratização. A praticidade do sistema de fechadura inteligente com base no RFID é um dos vários motivos para a adesão do mesmo, outro motivo é a segurança pois com o sistema será possível fazer um monitoramento muito mais eficaz do que o existente atualmente, onde o professor, ou profissional responsável poderá ver em qualquer lugar que tenha conexão com a internet acessando ao site [UFSC RFID¹](http://ufscrfid.pe.hu). Será possível verificar se a sala, ou o laboratório está em uso e também quem está presente na sala, com ordem de quem entrou primeiramente de acordo com o horário de chegada.

IV. LOGICA DE PROGRAMAÇÃO

A. Modo de comunicação e leitura

Para o fácil entendimento e compreensão geral tem-se nessa secção uma explicação textual e um fluxograma para a programação e funções que nosso projeto será capaz de realizar.

O Arduino ligado à um ESP8266 Wifi fará conexão com um servidor MySQL na nuvem construindo um sistema informatizado que irá gerenciar todos os dados coletados pelo sensor RFID.

Os chips RFID contêm um código único que será utilizado para fazer a identificação pessoal do usuário através de uma associação em base de dados. Será criada uma tabela na qual irá constar as informações de cada usuário classificados em "Chave de edição", "Chave mestra", "Autorizados" e "Não autorizados". Um usuário pode estar autorizado em uma sala e não autorizado em outra, o que exigiria modificações no banco de dados, porém o nosso protótipo levará em consideração apenas uma porta (uma única sala). Nota-se que a classificação "Não autorizados" é atribuída automaticamente a todo ID não cadastrado no sistema.

B. Armazenamento de dados e classificações

Uma segunda tabela irá armazenar dados de utilização dos cartões como horário de entrada, sala além das informações pessoais do usuário que serão importadas da primeira tabela.

O usuário com classificação "Autorizados" terá permissão para abrir a porta, enquanto o usuário com classificação "Não autorizados", não terá permissão para abrir a porta.

O chip com classificação "Chave de edição" será único e haverá a possibilidade de alteração pela internet através de senha de acesso pré-definida caso o mesmo seja perdido. Ele servirá para alterar a classificação de um usuário pré-cadastrado de "Não autorizado" para "Autorizado", e vice-versa.

O usuário com classificação "Chave mestra" sempre terá acesso à porta e não poderá sofrer alteração da chave e edição, essa chave estará salva na memória do Arduino para que ela possa abrir a porta mesmo sem acesso à internet e normalmente pertencerá ao professor. Em uma eventual necessidade de alteração, a transferência da chave mestra deverá ser feita pela internet.

O software principal instalado no Arduino será subdividido em funções, cada uma responsável pelas ações que serão executadas a depender da classificação atribuída a chave detectada.

O sensor RFID/NFC estará sempre buscando um cartão e assim que o identificar o programa irá verificar a classificação inicialmente checando à memória interna para ver se é a chave mestra, caso contrário fará uma busca no banco de dados. Se ainda assim o programa não identificar a classificação da chave a mesma receberá a classificação de "Não autorizados". Em seguida, o software entrará em uma estrutura condicional que fará a chamada à função associada a classificação anteriormente selecionada.

C. Acionamento da fechadura

A seguir será feito o detalhamento de cada função. Para o caso dos cartões "Autorizados" ou "Chave mestra", será registrado no banco de dados o horário e as informações do usuário, caso obtenha sucesso então o arduino mandará para o relê a instrução para que este abra o contato por 10 segundos, não deixando então corrente elétrica passar para a fechadura eletromagnética assim destravando a porta, acionando um buzzer e LED's para notificar que a porta foi aberta, além de uma mensagem de confirmação no LCD. Caso não obtenha sucesso no registro dos dados o relê não será acionado, serão então enviadas instruções para o acionamento do LED e do buzzer além de uma mensagem no LCD, notificando o usuário.

Para o caso dos cartões "Chave de edição" o programa irá aguardar a próxima ação em um período de 10 segundos, caso a próxima ação seja passar um cartão "Não autorizado" no sensor RFID o arduino fará alteração no banco de dados que esse cartão passe a ser um cartão do tipo "Autorizados", caso a ação seja passar um cartão do tipo "Autorizados", será feita a alteração para "Não autorizados". Caso a ação seja passar novamente o cartão de edição o programa notificará no LCD que passe novamente o cartão para confirmar a exclusão de

¹ <http://ufscrfid.pe.hu>

todos os cartões do tipo "Autorizados", caso não seja realizada nenhuma ação por 10 segundos ou seja passado o cartão do tipo "Chave mestra", a função de edição será interrompida.

Na situação em que é passado um cartão do tipo "Não autorizado" o programa irá acionar o buzzer, os LED's e enviará um comando para mostrar no LCD que o mesmo não possui autorização. É possível ver o fluxograma da programação na figura 4.

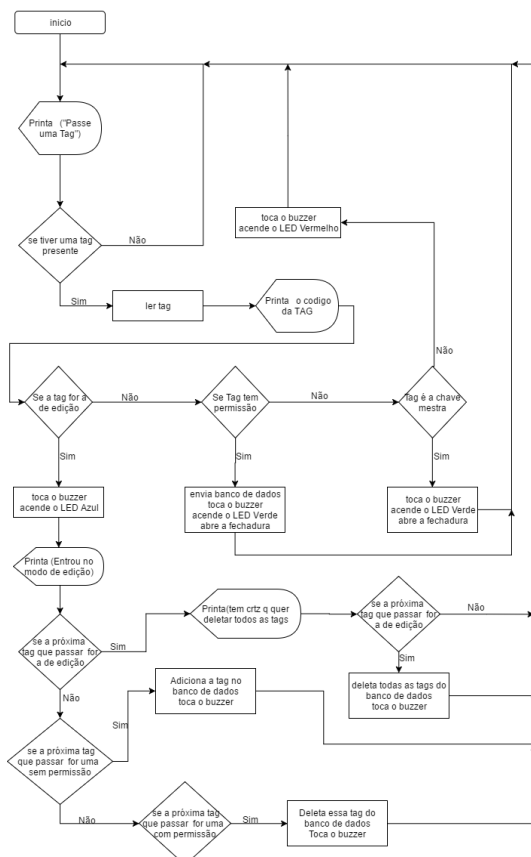


Figura 3 Fluxograma da Programação

V. RESULTADOS

No teste inicial com o equipamento foi obtido bons resultados, tudo ocorreu como o esperado. Os usuários não tiveram dificuldade em fazer uso do protótipo já que o mesmo foi planejado para funcionar de forma intuitiva, além de possuir manual de operações. O reconhecimento de tags, inserção de dados na web e acionamento da fechadura também tiveram êxito, funcionando sem problemas.

VI. CONCLUSÃO

A UFSC do Campus Blumenau utiliza atualmente um sistema de concessão de chaves ultrapassado, então a proposta do projeto de fechadura inteligente, baseado em RFID apresenta vantagens, comparada ao atual, pois garantem mais praticidade, velocidade, e um sistema de fácil configuração para todos, menos propenso a falhas e que principalmente garante mais segurança devido ao monitoramento. Ao longo

do projeto foram abordados vários pontos como, custos, componentes, funcionamento entre outros.

O projeto como apresentado na Figura 5 apresenta, uma forma de abertura de portas para laboratórios e salas de aula a princípio, onde que por meio de um Arduino e um leitor RFID, os alunos que apresentarem o cartão estudantil da UFSC com autorização de acesso a sala dada pelo professor, poderão acessar a sala, somente aproximando o cartão perto do leitor, onde as informações de entrada serão armazenadas em um banco de dados podendo ser monitorado a distância.

Ao longo do projeto foram feitos teste de funcionamento, onde os resultados foram bastantes satisfatório, devido ao código bem otimizado, não apresenta dificuldade para adicionar ou excluir autorizações dos cartões, o que torna o sistema bastante receptivo pela sua fácil utilização por pessoas sem conhecimento técnico.

Para aperfeiçoamento do projeto futuramente, uma entre inúmeras implementações que podem ser feitas, por exemplo é a criação de um aplicativo para Smartphones, também a instalação de um teclado, onde que por meio de senha seria possível ter acesso a sala e também fazer a recuperação do cartão de edição no caso de extravio do mesmo, sendo aconselhável somente a professores ou colaboradores da instituição possuírem a senha.

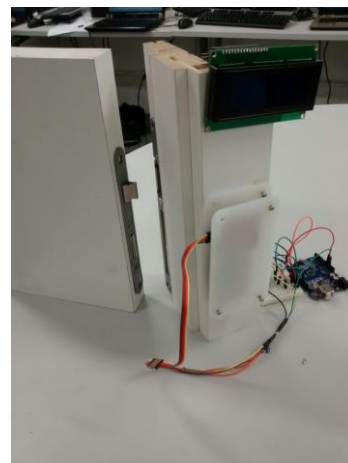


Figura 4- Protótipo do projeto.

REFERÊNCIAS

- [1] RFID & NFC, Emartee. Disponível em: <<http://www.emartee.com/product/42291/PN532%20NFC%20RFID%20Reader/Writer%20Module%20V3>>. Acesso em: 5 de jul. 2016;
- [2] Fecho Elétrico, HDL. Disponível em: <<http://www.hdl.com.br/produtos/fechaduras/fecho-eletrico/fecho-eletrico-mod-fec-91-ca-espelho-curto-trinco-ajustavel>>. Acesso em: 5 de jul. 2016;
- [3] Arduino & Genuino, Arduino. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 5 de jul. 2016;

- [4] Arduino, Wikipédia. Disponível em:
<<https://pt.wikipedia.org/wiki/Arduino/>>. Acesso em: 5 de jul. 2016;
- [5] RFID, Wikipédia. Disponível em:
<https://pt.wikipedia.org/wiki/Identifica%C3%A7%C3%A3o_por_radiofrequ%C3%Aancia>. Acesso em: 5 de jul. 2016;
- [6] NFC, Wikipédia. Disponível em:
<https://pt.wikipedia.org/wiki/Near_Field_Communication/>. Acesso em: 5 de jul. 2016;
- [7] Filipeflop. Disponível em:
<<http://www.filipeflop.com/>>. Acesso em: 5 de jul. 2016;
- [8] MySQL, Wikipédia. Disponível em:
<<https://pt.wikipedia.org/wiki/MySQL/>> Acesso em: 5 de jul. 2016;
- [9] NFC Computer Unlocker, joeatx247. Disponível em: <
<http://www.joeatx247.de/?p=11>> Acesso em: 5 de jul. 2016;

Sistema Above de Segurança

AboveSecurity S/A – Projeto Arduino

Allyson Freitag Binsfeld, Felipe Alfredo Nack,

Gustavo Paloschi, Lucas Azzolini Vieira e

Nathã Vieira de Azevedo

Universidade Federal de Santa Catarina – Campus Blumenau

Departamento de Engenharias

Blumenau, Brasil – Rua João Pessoa, 1750.

allysonfreitag99@hotmail.com, felipe.a.nack@gmail.com,

paloschigustavo@gmail.com, lucasazzollinivieira@hotmail.com,

nathaaezevedo@hotmail.com

Resumo — Este artigo tem como objetivo a documentação do projeto de um Sistema de Segurança desenvolvido pela empresa Above Security S/A utilizando uma placa de prototipagem Arduino. Ademais, o documento foi criado para auxiliar no entendimento do projeto e do sistema em si, tanto no seu desenvolvimento quanto na instalação deste. O sistema foi projetado para manter a integridade de áreas designadas, sejam pequenos negócios ou residências domiciliares.

Palavras-chave — microcontrolador, arduino, automação, domótica, segurança, sensores.

I. INTRODUÇÃO

Os problemas relacionados com o aumento das taxas de criminalidade, o aumento da sensação de insegurança, sobretudo nos grandes centros urbanos e de comércio, além da intensificação recorrente de assaltos à casas, apartamentos e estabelecimentos, representam um número extraordinário de desafios para combater a falta de segurança do cidadão, que, depois de tanto tempo, acabou se tornando comum no dia a dia deste.

A amplitude de tais problemas ligados à segurança é enorme e, que muitas vezes, acaba saindo do controle. Tal situação acaba ficando marcada na vida do cidadão, tanto que é extremamente comum um indivíduo não tirar o celular do bolso, bolsa ou mochila em uma rua ou beco, ficando receoso e inseguro, podendo ser assaltado a qualquer momento. Além de tudo, esse fator se torna ainda mais preocupante quando a pessoa passa a ter “medo” até mesmo em seus domicílios, fazendo com que os moradores sintam-se inseguros, mesmo no aconchego do seu próprio lar.

Depois do estudo realizado pela Coordenadoria de Análise e Planejamento da Secretaria de Estado da Segurança Pública [10], com base nos boletins de ocorrência de assaltos residências em Curitiba no ano de 2011, percebe-se a necessidade do aumento de esforços na área de segurança residencial e, segundo o estudo:

A cada 50 minutos, uma casa é furtada ou roubada na capital (do Paraná). Ao todo foram 859 casos – 697 furtos (quando ninguém está em

casa) e 162 roubos (assaltos à mão armada). Em comparação com o ano passado (2010), houve um aumento de 16,5% nas ocorrências de roubo e queda de 5% nas de furtos. Mais do que as estatísticas, o trabalho revela detalhes importantes sobre a ação dos bandidos, como dias da semana, horários e bairros em que os ataques a casas mais acontecem. Quase 50% dos roubos (76 assaltos) foram cometidos à noite, entre 20 e 23 horas. No caso dos furtos, 37% dos casos ocorreram na terça e na quarta-feira, principalmente à tarde. “O furto tem uma característica diferente: o ladrão escolhe um momento em que não haja ninguém em casa. Por isso, a maior parte dos casos ocorre à tarde e no meio da semana”, explica o delegado Guilherme Rangel, da Delegacia de Furtos e Roubos de Curitiba. [10]

Ainda de acordo com a notícia, durante os períodos vespertino e noturno, quando as residências estão vazias ou com índices de observação baixos, tais crimes se tornam mais frequentes, indicando que sistemas de segurança que independem do horário devem se tornar uma opção para o residente e, dessa forma, proteger o que for necessário.

Existem diversos meios para contornar problemas desse tipo que, dependendo da aplicação, podem variar o preço e qualidade. Há casos onde os domicílios e microempresas não necessitam de um sistema de segurança complexo, fazendo com que o leque de possibilidades de soluções se torne mais abrangente. Em situações como esta, onde a segurança pode ser mais simples, soluções que envolvam placas, como o *Arduino*, se tornam extremamente viáveis e acessíveis ao cidadão.

II. DESCRIÇÃO DO PROBLEMA

Uma das consequências da urbanização e do crescimento exagerado da densidade demográfica é a falta de segurança. É extremamente comum que casas, apartamentos e estabelecimentos comerciais mais humildes não possuam sistemas de segurança por conta do seu preço, acessibilidade e praticidade.

Um problema recorrente é, de fato, ter a certeza de que o espaço se mantenha com a sua integridade preservada durante viagens ou até mesmo durante a noite, período no qual as residências tendem a ser assaltadas com mais frequência.

Com isso, um grupo de empreendedores, em parceria com moradores, buscou os serviços da empresa *AboveSecurity S/A* para discutir os processos necessários para o início de um projeto específico que a empresa instale e otimize um sistema automatizado para as mais diversas das possibilidades de invasão. Ademais, o sistema será instalado individualmente em cada residência e estabelecimento para maior segurança e monitoramento dos residentes e seus bens.

A. Um problema, de fato

A primeira barreira a se enfrentar ao propor uma solução para a segurança de casas e apartamentos comuns é o custo, tal qual pode ser facilmente superada por conta dos preços acessíveis encontrados em equipamentos compatíveis com a placa *Arduino*.

A segunda barreira é, por outro lado, o espaço necessário e discrição dos equipamentos, mas que acaba se tornando de simples resolução se utilizado pequenos módulos.

Por fim, a terceira barreira é a praticidade, fácil instalação e manutenção que – por conta da simplicidade da placa *Arduino* – é resolvida com facilidade.

Mesmo com três principais barreiras definidas, se tratando do sistema, há muitos outros problemas e acasos decorrentes em qualquer sistema que seja desenvolvido podendo vir a ocorrer durante as partes que dividem a criação do projeto, como: durante o processo de desenvolvimento: a falta de fundos para arcar com as despesas do projeto, a falta de tempo para a implementação do sistema e a possíveis problemas técnicos em dispositivos da placa *Arduino* e os sensores utilizados no projeto; na instalação: local inadequado e, caso ocorra, a falta de consideração do residente se necessária a modificação do ambiente para melhor implantação do sistema, etc; uso: é possível, também, verificar-se problemas que podem ocorrer durante o funcionamento do deste, como o uso inadequado e prejudicial do sistema, etc.

Mesmo que diversas precauções sejam tomadas, o sistema não é completamente seguro, assim como muitos outros. Durante assaltos, por exemplo, cortes de energia podem vir-se a acontecer, dessa forma, precauções devem ser tomadas. Dito isto, projeções futuras são pensadas a fim de manter o sistema atualizado conforme a necessidade estipulada seja na proteção domicilia ou na proteção de estabelecimentos comerciais.

III. SOLUÇÃO DO PROBLEMA

A segurança pessoal é algo que todo ser humano gostaria de possuir, seja na própria casa, trabalho ou nas ruas da cidade deste.

Pensando nisto, voltados para a área de segurança residencial e comercial, a empresa *AboveSecurity S/A*, traz um sistema de segurança complementar de baixo custo e acessível. Através de diversos sensores, estes sendo: de presença/movimento, de som e de luminosidade. Tendo como objetivo a detecção de qualquer interferência em suas

designadas áreas de monitoramento e, acontecendo isto, é ativado o alarme que alertará o residente, comerciante, tanto quanto a vizinhança de que algo está sendo detectado pelos sensores, podendo ser um intruso no local onde o sistema foi instalado.

Já a ativação e desativação do sistema de segurança se dará por uma senha, esta sendo escolhida pelo usuário e tendo um máximo e mínimo de quatro números. Após a entrada da senha, o sistema demorará 30 segundos para ser ativado e, caso o usuário cometa um erro ao digitar a senha, o sistema lhe avisará, por meio de um sinal sonoro de alta intensidade, que a senha digitada não corresponde à previamente cadastrada.

Aqui segue alguns dos componentes do sistema e suas respectivas funções, tal como especificações técnicas:

A. Placa *Arduino*

O microcontrolador utilizado será o de modelo MEGA 2560, este possuindo uma quantidade maior de entradas e saídas para melhor elaboração do projeto. Nele, também, serão conectados todos os componentes do projeto.

Finalmente, o programa será importado para o microcontrolador, este sendo o cérebro de todo o sistema, monitorando todo o funcionamento dos componentes que serão utilizados neste. Especificações:

- Microcontrolador: ATmega2560;
- Tensão de operação: 5V;
- Portas digitais: 54;
- Portas analógicas: 16;
- Corrente dos pinos I/O: 50mA;
- Corrente dos pinos 3,3V: 50mA;
- Velocidade do *clock*: 16MHz.

B. Sensor de Presença/Movimento

Como sensor de presença, utilizamos o módulo PIR DYP-ME003, que detecta movimentos que estejam em uma área de até 7 metros e funcionando com um sensor infravermelho. Ele será posicionado na mais elevada da parede do recinto com o intuito de cobrir uma área maior.

O módulo deste sensor é o PIR DYP-ME003, que une numa mesma estrutura o sensor PIR e também os circuitos necessários para ajuste e controle do sinal de saída. O módulo contém o sensor PIR propriamente dito, composto, internamente, por duas faixas com material sensível à ondas infravermelhas.

Na parte externa, existe uma espécie de “capa/tampa” que, na verdade, é uma lente fresnel. Quando há variação na detecção de ondas infravermelhas entre essas duas faixas de material sensível, a saída é acionada por um determinado tempo. A lente fresnel tem a função de ampliar o campo de visão do sensor, fazendo com que a luz se concentre em um único ponto. Especificações:

- Sensor infravermelho com controle na placa;
- Sensibilidade e tempo ajustável;
- Tensão de operação: 4,5–20V;
- Tensão de dados: 3,3V (Alto) – 0V (Baixo);
- Distância detectável: 3–7m (Ajustável);
- Tempo de *delay*: 5 – 200seg (*Default*: 5seg);
- Tempo de bloqueio: 2,5seg (*Default*);
- *Trigger*: (L) – Não repetível, (H) – Repetível. (*Default*: H);
- Temperatura de operação: - 20 a +80 °C;
- Dimensões: 3,2 x 2,4 x 1,8cm;
- Peso: 7g.

C. Sensor de Luminosidade

O sensor utilizado é o LDR (*Light Dependent Resistor*) que será posicionado no hall de entrada do recinto, com um *laser* direcionado ao sensor. O sensor detectará sempre uma luminosidade alta, então, assim que algo cruzar em frente ao *laser*, interrompendo a chegada luminosa ao sensor, o alarme será acionado. Especificações:

- Diâmetro: 5mm;
- Temperatura de operação: -30 a +70 °C;
- Espectro: 540nm (“Cabeça” do sensor);
- Comprimento com terminais: 32mm;
- Resistência no escuro: 1M Ω (Lux 0);
- Resistência na luz: 10–20 K Ω (Lux 10).

D. Sensor de Som

Para medir a intensidade do som, dispositivo será composto por um módulo e microfone embutido neste. Quando o sensor detectar um som de alta intensidade como, por exemplo, um vidro sendo quebrado ou até mesmo sons de disparos de armas de fogo, o alarme será acionado. O sensor será posicionado na parte central do local, favorecendo a detecção de sons de toda área do recinto. Especificações:

- Tensão de operação: 4-6V DC;
- Sensibilidade ajustável via potenciômetro;
- Saída digital e analógica.

E. Teclado Matricial

Para a digitação da senha será utilizado um teclado matricial 4x4, este será localizado na parte interna do local e, quando acionado, o sistema aguardará 30 segundos para ativar todos os sensores, proporcionando algum tempo para que o usuário saia do local, evitando o disparo do alarme antes mesmo do indivíduo deixar a área. Especificações:

- Teclas: 16;
- Conector: 8 pinos (2,54mm);
- Montagem: Auto-adesivo;
- Tempo de resposta: 5ms;
- Durabilidade: um milhão de ciclos por tecla;
- Temperatura de funcionamento: 0 a 70 °C;
- Tamanho: 69 x 77 x 0,8mm;
- Comprimento do cabo: 86mm;
- Peso: 10g.

F. Tela de LCD

Para mostrar ao usuário as informações sobre o sistema, será utilizada uma tela de LCD dividida em quatro linhas e vinte colunas (20x4). Nela será mostrada a senha digitada, se a senha está ou não correta, se o alarme foi ativado e se o alarme foi desativado. A tela será posicionada junto ao teclado, para melhor utilização do sistema. Especificações:

- *Backlight*: Azul;
- Cor dos caracteres: Branco;
- Potenciômetro para ajuste do contraste;
- Tensão de operação: 5V;
- Dimensão total: 98,0 x 60,0 x 14,0mm;
- Dimensão da área visível: 76,0 x 26,0mm;
- Dimensão dos caracteres: 2,94 x 4,74mm;
- Dimensão do ponto: 0,54 x 0,54mm.

G. Alarme (*Buzzer*)

Para a função de alarme será utilizado um *buzzer*, este possuindo um oscilador interno e, mesmo possuindo um tamanho reduzido, é extremamente eficaz. Especificações:

- Tensão de operação: 5V;
- Ressonância: 2400Hz;
- Decibéis: 85dB;

- Temperatura de funcionamento: -20 a +70 °C;
- Altura: 10mm;
- Diâmetro: 12mm;
- Peso: 2g.

IV. CUSTOS

A seguir se encontra a tabela de custos. Esta consta os nomes de cada componente, quantidade, tanto quando seus preços e preço total. Com isso, é possível calcular os custos de desenvolvimento do projeto e ver se este é viável dependendo das condições de cada usuário interessado.

TABLE I. TABELA DE CUSTOS

A quantidade dos sensores a seguir foi cotada para um pequeno espaço de monitoramento (5m²), com o aumento da área, a quantidade de sensores aumenta da mesma forma, o que acaba por elevar o preço final do projeto.

Componentes	Quantidade (unidade)	Preço (reais)
Sensor de Movimento / Presença PIR	1	R\$ 16,90
Buzzer 5V Contínuo Ativo	1	R\$ 12,90
Sensor de Luminosidade LDR 5mm	1	R\$ 1,40
Sensor de Som KY-038 Microfone	1	R\$ 19,90
Placa Mega 2560 R3 + Cabo USB para Arduino	1	R\$ 89,90
Led de Alto Brilho	1	R\$ 0,20
TOTAL	6	R\$ 141,20

^a. Tabela de custos dos componentes usados no desenvolvimento do projeto.

^b. Preços cotados com distribuidores no mercado nacional.

V. LÓGICA

A princípio, uma senha já estará configurada como padrão direto de fábrica e recomenda-se que o usuário altere-a a seu

gosto logo no primeiro uso do sistema. A senha padrão será 0000 e para alterar a senha, o usuário deverá pressionar a tecla correspondente à letra *A*, sendo redirecionado para a interface de definição de senha. Então, seguindo as instruções presentes na tela de LCD, deverá ser digitada a senha atual para que a permissão de alteração seja liberada e a nova senha definida. Após a definição da senha realizada pelo usuário o sistema estará pronto para inicialização.

Logo ao ser ligado o sistema estará em modo de espera, aguardando que a senha correta seja digitada. No caso de o usuário digitar a senha incorretamente, uma mensagem será apresentada a este para que tenha consciência que a senha digitada não corresponde a senha correta e, então, o sistema retornará para a fase inicial, no qual o sistema aguarda o usuário digitar a senha corretamente. Quando tal senha for digitada, o sistema, inativamente, aguardará por trinta segundos para que o usuário possa sair do local e então iniciará o monitoramento do local.

O sensor de presença utilizado tem funcionamento bem simples de forma a retornar o valor *zero* sempre que não detectar movimento e, ao perceber, retorna valor *um*.

Na parte de detecção do sensor de luminosidade será direcionado um laser para que o valor de retorno seja alto e constante até que algum objeto interfira na passagem do laser e o valor diminua drasticamente, acionando o alarme.

Já as mensagens passadas para o usuário serão transmitidas por uma tela de LCD, que é conectada a um potenciômetro com o objetivo de regular o brilho da tela. Esse potenciômetro é necessário para o funcionamento correto do módulo, pois, sem ele, o brilho pode ter problemas e a tela pode não ficar visível de determinados ângulos de visão.

O teclado utilizado para comunicação direta entre o sistema e o usuário possui todos os algarismos árabes que utilizamos e mais dois caracteres especiais e as quatro primeiras letras do alfabeto, totalizando dezesseis teclas alinhadas em quatro linhas e quatro colunas. Para a manipulação deste teclado foi utilizada uma programação com as teclas formando uma matriz de dimensões 4x4. Os algarismos estão organizados em ordem de um a nove nas três primeiras linhas e colunas e o número zero se encontra na segunda posição da última linha da matriz. A última coluna possui as quatro primeiras letras do alfabeto (*A*, *B*, *C* e *D*), mas é utilizada apenas para duas funções específicas que são: ao pressionar a letra *A* o usuário será direcionado para a escolha de uma nova senha; e, ao pressionar a letra *B*, o usuário pode cancelar a contagem regressiva de trinta segundos para a ativação do alarme.

VI. TESTES REALIZADOS

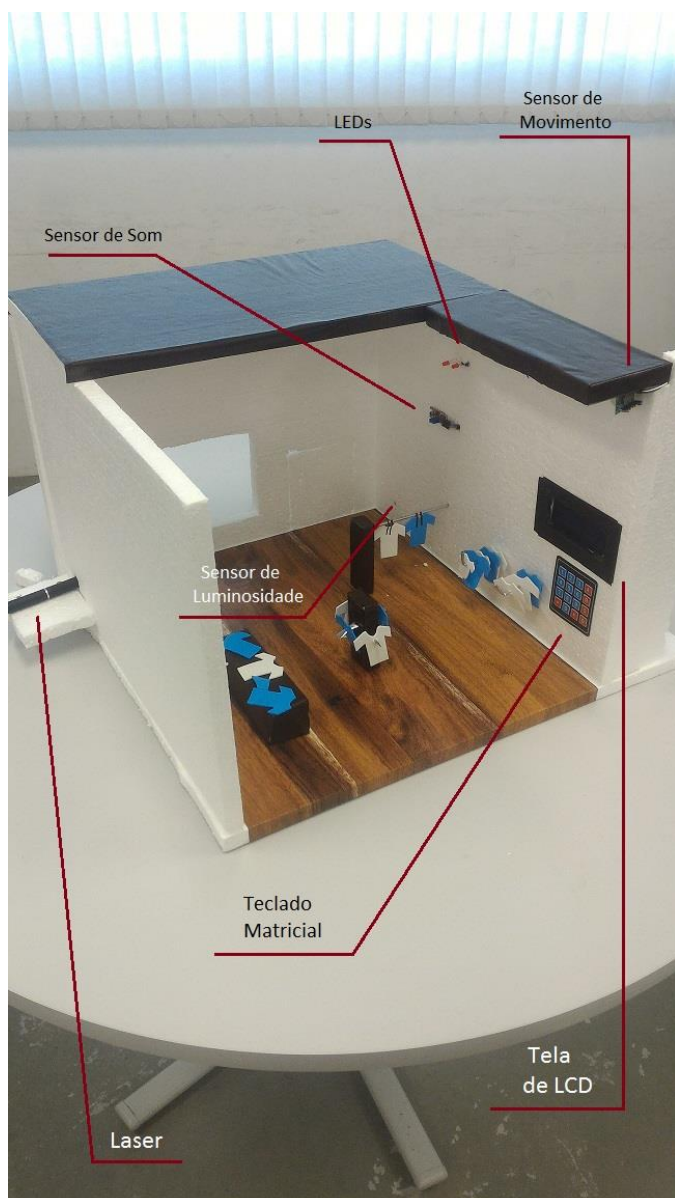


Figura: Sensores e componentes utilizados no desenvolvimento do projeto e suas devidas localizações na representação em pequena escala de uma loja.

Os primeiros testes realizados tiveram como objetivo a verificação da funcionalidade de todos os sensores presentes no projeto. Além disso, também foram testados valores que estes retornam, sendo testados em diversos locais e condições para maior precisão da coleta de dados, já que os sensores retornam valores distintos a cada segundo e, também, bastante específicos, o que nos deu uma quantidade maior de trabalho, mas, também, acabou por deixar as configurações do projeto mais precisas.

Após isso, a programação e implementação desta nos parâmetros do projeto foi realizada, mas, devido a facilidade de uso do microcontrolador *Arduino* e seus componentes, este passo não foi tão cansativo e complicado quanto o esperado, já

que tal projeto era algo novo para boa parte do grupo, deixando-nos um pouco receosos.

Por fim, uma versão reduzida de uma loja, uma maquete, foi criada, visando o teste de toda a amplitude do projeto. A maquete dispunha de um sensor de luminosidade, posicionado junto a entrada frontal, um sensor de movimento localizado em um dos cantos superiores e um sensor de som em uma das paredes. Ao decorrer dos testes, foi verificado que a integridade dos sensores se mantém, bem como seu funcionamento e o funcionamento de todas as funções propostas.

VII. PROJEÇÕES FUTURAS

Devido à falta de tempo ou até mesmo de fundos, diversas ideias que poderiam ser implementadas no projeto acabaram sendo deixadas de lado, mas, para projeções futuras, nós acrescentaríamos as seguintes funcionalidades ao sistema:

- Recuperação de senha via SMS;
- Monitoramento via câmeras de segurança;
- Aplicativo Android/iOS para melhor comunicação do usuário com o sistema. Além de complementar a implementação das câmeras de segurança, assim, ficando remotamente monitorável;
- Ativação do alarme remotamente via bluetooth ou wi-fi;
- Permitir que o usuário informe um horário para que o sistema se ative automaticamente neste momento;
- Autenticação do usuário através de um sensor biométrico para acionamento ou desligamento do sistema.

VIII. CONSIDERAÇÕES FINAIS

Um sistema de segurança, mesmo que simples, pode ter um custo extremamente elevado, levando pessoas a optarem por deixar seus bens imóveis desprotegidos. Esse problema foi de fato resolvido aplicando um sistema de segurança de baixo custo, feito a partir de uma placa *Arduino*. Apesar de seus custos baixos, os sensores empregados nesse projeto possuem uma precisão mais do que necessária para seus fins.

A placa possui, de fato, algumas limitações, como a sua quantidade de entradas e saídas, entretanto, é suficiente para suprir a demanda de um sistema de segurança simples, contando com sensores de presença, luminosidade e som. Por outro lado, desconsiderando as limitações, a placa possui um preço extremamente abaixo do comum que, desta forma, consumidores que antes deixariam a segurança de lado por

questão de preço, agora, poderão desfrutar de uma versão de baixo custo.

Essa parte do estudo visa principalmente a domótica que, em outras palavras, é a automação residencial. Entretanto, os mesmos equipamentos utilizados nesse projeto podem ser reutilizados para outros fins, sejam eles a automação industrial ou a manufatura de pequenos robôs. Por outro lado, também deve ser ressaltada a ideia de que, utilizando corretamente os recursos oferecidos pela plataforma *Arduino*, diversos projetos que custariam grandes quantias de dinheiro podem ser fabricados, conciliando preço e qualidade. Por fim, os testes são favoráveis. Os sensores empregados, junto com a programação aplicada formam um sistema de precisão aceitável e de baixo custo.

REFERÊNCIAS

- [1] Arduino, “Arduino Website”. Disponível em: <https://www.arduino.cc/>. Acesso em 26 de maio de 2016.
- [2] Wikipedia, The Free Encyclopedia, “Arduino”. Disponível em: <https://en.wikipedia.org/wiki/Arduino>. Acesso em 9 de junho de 2016.
- [3] Arduino e Cia, “Display LCD 20x4 Arduino”. Disponível em: <http://www.arduinoocia.com.br/2014/06/arduino-display-lcd-20x4.html>. Acesso em 20 de junho de 2016.
- [4] Arduino, “Language Reference”. Disponível em: <https://www.arduino.cc/en/Reference/HomePage>. Acesso em 23 de junho de 2016.
- [5] Blog FilipeFlop, “Como usar o teclado matricial 4x4 com arduino”. Disponível em: <http://blog.filipeflop.com/arduino/teclado-matricial-4x4-arduino.html>. Acesso em 26 de junho de 2016.
- [6] Arduino e Cia, “Sensor de presença com módulo PIR DYP-ME003”. Disponível em: <http://www.arduinoocia.com.br/2014/06/sensor-presenca-modulo-pir-dyp-me003.html>. Acesso em 26 de junho de 2016.
- [7] Arduino e Cia, “Controle de luz utilizando LDR”. Disponível em: <http://www.arduinoocia.com.br/2013/09/controle-de-luz-utilizando-ldr.html>. Acesso em 26 de junho de 2016.
- [8] Arduino e Cia, “Son no Arduino”. Disponível em: <http://www.arduinoocia.com.br/2013/06/sons-no-arduino.html>. Acesso em 27 de junho de 2016.
- [9] Arduino e Cia, “Medindo a intensidade do som com o sensor KY-038”. Disponível em: <http://www.arduinoocia.com.br/2015/07/sensor-de-som-ky-038-microfone-arduino.html>. Acesso em 5 de julho de 2016.
- [10] Gazeta do Povo, “A cada hora, uma casa é furtada”. Disponível em: <http://www.gazetadopovo.com.br/vida-e-cidadania/a-cada-hora-uma-casa-e-furtada-4bdkq1espw4pt7pu5ruim5k5q>. Acesso em 5 de julho de 2016.

Projeto: Armaduino

Allan de Melo Heil, Arthur Wegner Wippel, Bruno Bubeck, Carlos Eduardo Ribeiro e Guilherme Geske.
UFSC – Universidade Federal de Santa Catarina
Blumenau, Brasil

Resumo—A segurança em nosso país está precária. Tendo isso em vista, nosso grupo desenvolveu um produto, fazendo uso da plataforma arduino, que pode ser usado por grandes empresas, como bancos e multinacionais que precisam de um sistema de vigilância adequado. As especificações do produto, tal como sua montagem, construção e lógica estão descritas neste documento, incluindo dados, imagens e os motivos para a criação deste projeto.

Palavras-Chave: *Arduino, segurança, vigilância.*

I. INTRODUÇÃO

O projeto arduino trata-se do desafio dos alunos de analisar um problema, ou tema do cotidiano e solucioná-lo através de um produto que faz uso da plataforma arduino. Nossa equipe, membros da empresa Armaduino LTDA, observou o problema de segurança que afeta diversas empresas, ocasionado pela alta criminalidade vigente em nosso país e no mundo. Para solucioná-lo, a Armaduino LTDA desenvolveu o projeto de uma arma não letal de vigilância, controlada pelo computador, visando a melhor qualidade do sistema de segurança dessas empresas.

II. O PROBLEMA

A. **Segurança:** Em nosso país, a criminalidade e os assaltos vem crescendo assustadoramente. Os criminosos se sentem a vontade para assaltar pois a maioria das lojas e empresas não possuem um sistema de vigilância, ou quando possuem, não é adequado. De acordo com a Secretaria de Defesa Social, apenas em 2016, a taxa de assaltos cresceu 31,54% no estado de Pernambuco. Em São Paulo, houve um aumento de 42,8% no número de roubos a banco entre abril de 2014 e abril de 2015 (Figura 1), dados da Secretaria de Segurança Pública [5]. Tendo isso em vista, a empresa Armaduino LTDA vem ao mercado oferecer um novo tipo de produto neste setor, visando melhorar a qualidade da vigilância e o ampliamiento da segurança no país.



Fig. 1. Gráfico de crimes.

- B. **A ideia:** A empresa Armaduino LTDA tem como objetivo manter a segurança de seus clientes de forma simples e monetariamente acessível. Para isso, desenvolvemos uma sentinela munida de uma arma, que dispara balas de borracha (porém há a possibilidade da inserção de outros tipos de munição) controlada por computador. O objetivo é aumentar a segurança tanto para a empresa quanto para os próprio profissional que cuida dessa área, pois o nosso produto pode ser controlado à distância, evitando confrontos diretos entre criminosos e seguranças. Além disso, as empresas podem passar a gastar menos com esses profissionais, tendo em vista que com este produto, o número deles pode ser reduzido. Nosso projeto possibilita o monitoramento de empresas, condomínios, bancos e outros tipos de estabelecimentos que requerem um serviço de segurança e vigilância mais avançado.
- C. **Desafios:** Nossos desafios quanto aos problemas técnicos e estruturais começam pelos motores. Precisamos utilizar um motor, que além de forte e resistente, seja preciso, a fim de ter uma grande durabilidade e que seja de fácil manejo e manutenção. Além disso, precisaremos de uma estrutura que suporte o peso dos motores e da arma, e que ofereça firmeza na hora dos disparos e movimentos. Por fim, a programação deve ser projetada para que a pessoa que faça o uso do software tenha seu trabalho facilitado.

D. Possíveis Falhas e imprevistos: Durante o projeto, observamos que há a possibilidade de ocorrência de falhas. Dentre elas está o desgaste dos rolamentos, pois o protótipo exige muito uso dos mesmos, ao fazer a movimentação da arma. Outro fator de risco são os cabos, pois a disposição dos mesmos pode ser alterada e danificada pela movimentação da arma, o que pode ocasionar rompimento, desligamento ou mal funcionamento deles. Além disso, com o recuo da arma ao atirar pode fazer com que o suporte se mova, ocasionando imprecisão nos disparos, o que deve ser impedido com a boa fixação do mesmo no solo ou na superfície de disparo.

III. DESCRIÇÃO DA SOLUÇÃO QUANTO A CONSTRUÇÃO.

- A. Câmera:** O primeiro passo, é processar a imagem da câmera que ficará na arma. Como a capacidade de processamento do arduino é baixa, a imagem será enviada diretamente para o computador, que processará a mesma, mostrando em tempo real a imagem captada pela câmera. Por se tratar de um protótipo, optamos por esse método pois não temos acesso à uma plataforma arduino que pode ser utilizada para o processamento das imagens juntamente com os outros processos lógicos, o que acaba ocasionando uma limitação no projeto, tratada na sessão de possíveis futuras melhorias, no final deste documento.
- B. Estrutura e suporte:** Será construído um suporte (figura 2) com flexibilidade móvel para a otimização da mira da arma. Para isso, construiremos um suporte Pan/Tilt auxiliado de rolamentos para não causar uma sobrecarga nos motores.

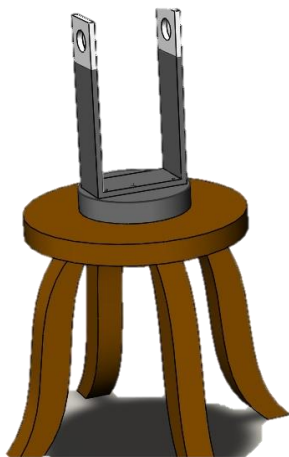


Fig. 2. Suporte e eixo.

Assim, a arma ficará estável e com uma boa mobilidade angular, tanto vertical quanto horizontalmente. Para a construção do suporte, a equipe fez uso das peças de MDF, mostrados na figura 3.



Fig. 3. Peças de MDF

- C. Motores:** Utilizaremos dois motores de passo (figura 4), pois oferecem uma grande precisão angular para o protótipo, além de bom torque para a realização do movimento da estrutura de suporte da arma. Um deles será localizado abaixo da superfície de suporte da arma, fazendo a rotação em 360 graus da mesma, ou seja, responsável pela movimentação PAN (horizontal). O outro motor de passo é localizado próximo ao encaixe da arma, e será responsável pela movimentação TILT (vertical). Além disso, faremos uso de um servo motor (Fig. 5) de pouca potência para efetuar os disparos, que ficará próximo ao gatilho (Fig. 12)



Fig. 4. Motor de passo.



Fig. 5. Servo motor.

D. Rolamentos: Faremos uso de três rolamentos para não sobrecarregar os motores, visando a otimização do protótipo. Um deles para a rotação da base e outros dois para o ajuste da inclinação da arma. O rolamento da base é do tipo axial, e os outros dois são do tipo rígidos de esfera, conforme mostra a Figura 6.

E. Drivers: Para aprimorar a comunicação arduino-motor e para o controle da corrente elétrica e tensão, utilizaremos dois drivers modelo Dvr8825 (Fig. 6), que será responsável por tal função, não deixando que uma corrente muito alta circule no motor, impedindo danos no mesmo.



Fig. 6. Rolamentos e drivers.

F. Arduino: O modelo de arduino que utilizamos no projeto foi o Mega 2560 (Figura 7). Ele é um microcontrolador baseado no ATmega2560. Possui 54 entradas / saídas digitais, 16 entradas analógicas, 4 UARTs (portas seriais de hardware), uma conexão USB, um conector de alimentação, um cabeçalho ICSP e um botão de reset. Sua tensão de funcionamento é de 5V e sua memória flash é de 256KB. Com isso, o arduino é equipado com tudo que é necessário para suportar o microcontrolador, basta conectá-lo ao computador com um cabo USB. O Mega 2560 é uma atualização do Arduino mega e pode ser programado com o software Arduino [2].

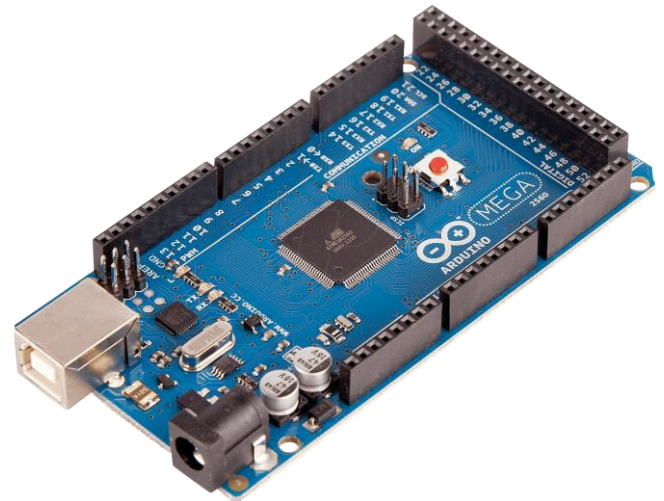


Fig. 7. Arduino MEGA 2560.

IV. DESCRIÇÃO DA SOLUÇÃO QUANTO A LÓGICA.

A. Comunicação com os motores: O Arduino faz o controle dos motores de passo através do driver Dvr8825, indicando a direção e a quantidade de passos que o mesmo deverá rotacionar enquanto o usuário estiver pressionando a tecla do respectivo comando, isto é, para esquerda, para direita, para cima e para baixo. Também se encarrega de controlar um servo motor, que faz o movimento para acionar o gatilho da arma e disparar os projéteis. Para que o Arduino dê a ordem de movimento ao driver de motor de passo, será utilizado a comunicação serial presente no microcontrolador através do protocolo RS-232¹, no qual um programa implementado no computador fará o envio dos caracteres ao microcontrolador que por sua vez fará o reconhecimento dos caracteres e dará o comando indicado aos drivers. Com isso, os seguintes caracteres serão enviados ao microcontrolador para dar as ordens de controle do equipamento: 1) 'w', fará com que a arma se movimente para cima no eixo vertical. 2) 's', fará com que a arma se movimente para baixo no eixo vertical. 3) 'a', fará com que a arma se movimente para a esquerda no eixo horizontal. 4) 'd', fará com que a arma se movimente para direita no eixo vertical. 5) 'f', fará com que acione o gatilho para dispara o projétil.

1. Protocolo RS-232: compilação de diretrizes que regulam a comunicação em série durante a troca de dados realizadas por dois dispositivos. Além disso, essa interface é utilizada para fixação dos patamares de tensão e a definição do posicionamento físico dos conectores a serem utilizados durante o processo [1].

B. Controle da corrente elétrica: O driver de motor de passo Dvr8825 possui um potenciômetro pelo qual pode-se ajustar a tensão de referência (VREF) (Figura 8), que pode ser medida no ponto destacado em vermelho na imagem acima junto com o pino GND (Ground) com um voltímetro. A tensão de referência serve para ajustar o limite de corrente “Im” que será liberado para o motor, assim a fórmula para calcular a corrente do motor é:

$$I_m = V_{ref} * 2$$

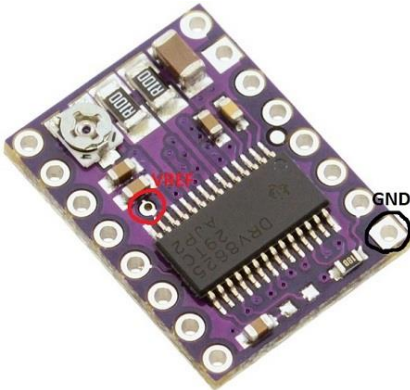


Fig. 8. Driver, posição VREF e GND.

A Figura 9 ilustra o esquema elétrico do circuito dos motores de passo acompanhado do driver e o microcontrolador. Os pinos STEP e DIR representam a quantidade de passos que o motor deverá dar e a direção no qual o motor deverá rotacionar os passos, esses passos são definidos pela biblioteca stepper do arduino.

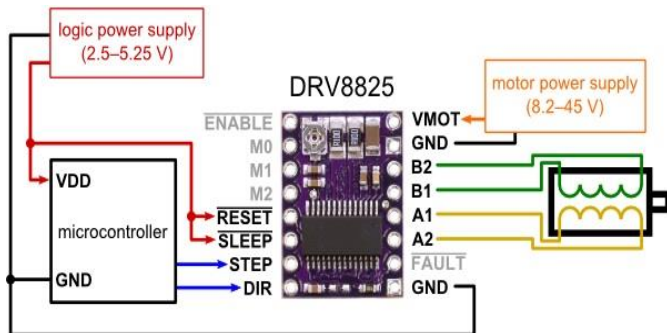
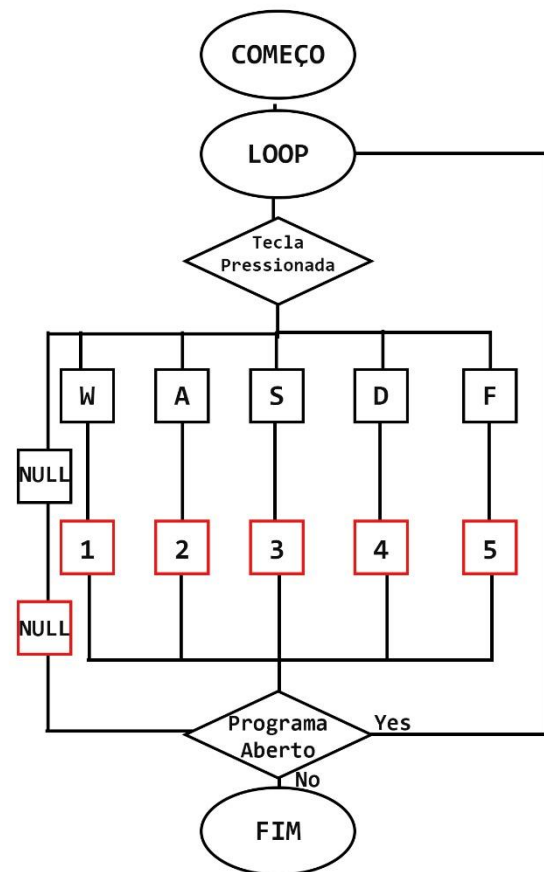


Fig. 9. Esquema elétrico. [4]

C. Programação: Para que o usuário controle o equipamento com mais facilidade, um programa em C, será desenvolvido para enviar os caracteres do computador para o utilizando o protocolo RS-232. A seguir, é mostrado um fluxograma com a lógica básica de programação para a movimentação dos eixos da arma.



- 1) Representa o pressionamento da tecla W pelo usuário, movimentará os eixos da arma no sentido vertical para cima.
- 2) Representa o pressionamento da tecla A pelo usuário, movimentará os eixos da arma no sentido horizontal para a esquerda.
- 3) Representa o pressionamento da tecla S pelo usuário, movimentará os eixos das armas no sentido vertical para baixo.
- 4) Representa o pressionamento da tecla D pelo usuário, movimentará os eixos das armas no sentido horizontal para a direita.
- 5) Representa o pressionamento da tecla F pelo usuário, efetuará os disparos da arma.

Observação: Caso a tecla pressionada não seja uma das cinco acima citadas, não haverá retorno (NULL).

V. TABELA DE CUSTOS

Abaixo encontra-se tabela de custos do projeto. Por ser composto por uma complexa montagem, o protótipo conta com alguns itens que foram adquiridos paralelamente, sempre visando a otimização dos preços para evitar gastos excessivos.

ITEM	PREÇO	QUANTIDADE	TOTAL
<i>Motor de passo</i>	120,00	2	240,00
<i>Drivers Dvr8825</i>	19,00	2	39,00
<i>Rolamentos</i>	10,00	3	30,00
<i>Chapa de alumínio</i>	45,00	1	45,00
<i>Peças MDF</i>	20,00	1	20,00
<i>Arma</i>	1500,00	1	1500,00
<i>Diversos</i>	15,00	-	15,00
TOTAL	-	-	1889,00

VI. MONTAGEM

Seguem abaixo imagens que ilustram o posicionamento das peças utilizadas para a montagem do projeto, incluindo uma montagem em CAD (Fig.10) que consiste numa versão “base” do projeto. As figuras 11, 12, 13 e 14 mostram o posicionamento de diversas peças. Dentre elas estão o rolamento axial, motor de passo, servo motor, câmera, suporte metálico e a peça de encaixe.

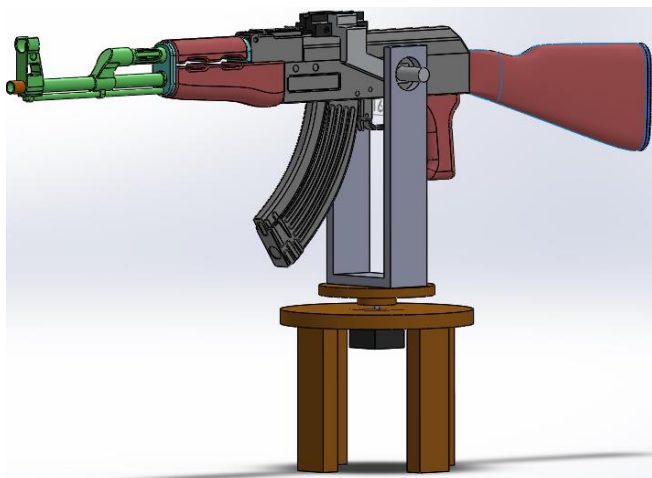


Fig. 10. Versão base do produto.



Fig. 11. Suporte, rolamento axial e motor de passo.

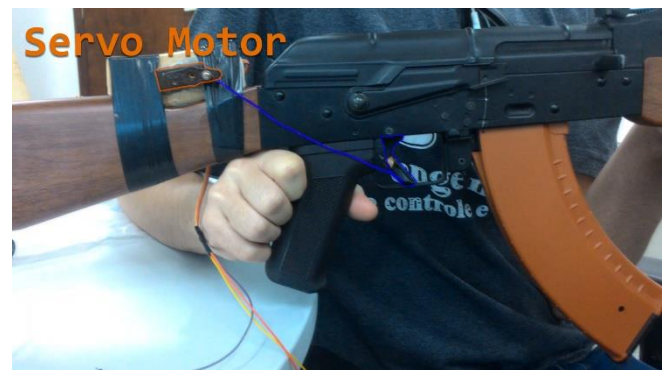


Fig. 12. Servo motor e gatilho.



Fig. 13. Câmera.

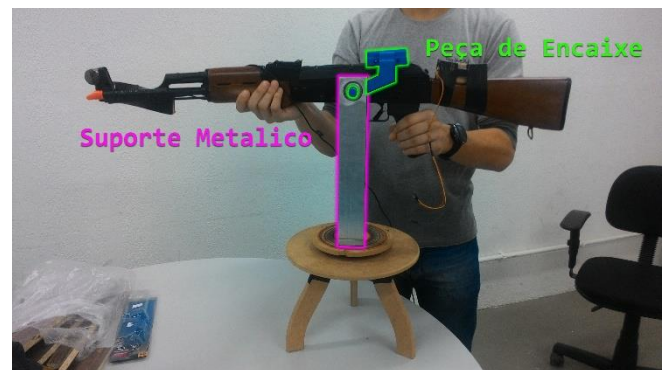


Fig. 14. Encaixe.

VII. CONCLUSÃO E POSSÍVEIS FUTURAS MELHORIAS

Podemos afirmar que este projeto aborda uma importante questão na sociedade. No futuro, o protótipo pode prestar serviço para diversas empresas e bancos e aprimorar o sistema de vigilância das mesmas, trazendo mais segurança para elas e para a própria população. A versão final do protótipo conta com um custo benefício otimizado, tendo em vista a complexidade do produto. Analisando seu desempenho e funcionamento, estipulamos algumas melhorias e aprimoramentos que podem ser implementados por alguma empresa de acordo com sua necessidade, visando a otimização do protótipo em diferentes tipos de situação.

Segue abaixo a lista de futuras possíveis melhorias

- A. **Câmera e arduino:** como já citado neste trabalho, a câmera é uma limitação do protótipo, tendo em vista que a mesma deve ser ligada diretamente no computador para o bom processamento da imagem. Com a obtenção de uma plataforma arduino ou outra plataforma mais eficiente, podemos obter uma câmara infravermelha para a otimização da visão noturna, período em que muitos assaltos acontecem, e a imagem captada pela mesma poderia ser processada diretamente na plataforma, o que melhoraria a disposição de cabos e outros dispositivos necessários no vigente produto.
- B. **Wi-fi e monitoramento à distância:** Analogamente às câmeras de vigilância residencial, na qual os proprietários conseguem monitorar a atividade em suas casa quando estão viajando, o mesmo pode ser aplicado a este projeto. O monitoramento a distância de empresas e bancos pode ser terceirizado e toda a parte de controle sobre as armas pode ser de responsabilidade de uma empresa de segurança, isso torna o trabalho mais profissional e feito de forma mais segura, sem a necessidade da presença de pessoas na empresa para a vigilância.
- C. **Sistema PAN/TILT e outras funcionalidades:** Em geral, o nosso projeto é baseado no sistema pan/tilt de movimentação, que neste caso, é da arma. Por ser muito versátil, este sistema pode ser utilizado para outros objetos, por exemplo câmeras de vigilância ou holofotes por exemplo. Portanto, em perspectivas futuras podemos desenvolver outros tipos de produtos utilizando outros objetos como peça central, além da arma. Para isso, é essencial a implementação da movimentação do dispositivo em 360 graus, que demandará melhor disposição de cabos e novas mecânicas.

- D. **Forças armadas:** Este protótipo pode ser modificado em relação a diversos aspectos. Um deles é quanto à munição. Há a possibilidade de usarmos armas de fogo no protótipo, visando um uso militar, tanto para o exército ou marinha. Para isso, pode ser implementado uma proteção para a arma na parte anterior, visando sua proteção contra outros disparos. Além disso, neste caso, é essencial o recarregamento autônomo da arma, ou até mesmo a remoção do pente de munição e a implementação de um pente estendido com alta capacidade de armazenamento de balas, pois a quantidade de disparos, em situações enfrentadas pelas forças armadas pode ser muito grande.

REFERÊNCIAS

- [1] EXPLICA FÁCIL. **O que é RS-232**. 2015. Disponível em: <<http://www.explicafacil.com.br/televisao/o-que-e-rs-232-protocolo-comunicacao/61>>. Acesso em: 7 jul. 2016.
- [2] ARDUINO (Estados Unidos). **Arduino Mega 2560**. 2016. Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>>. Acesso em: 07 jul. 2016.
- [3] WIKIPEDIA. **RS-232**. 2014. Disponível em: <<https://pt.wikipedia.org/wiki/RS-232>>. Acesso em: 05 jul. 2016.
- [4] ELETRONICS, Pololu Robotics &. **DRV8825 Stepper Motor Driver Carrier, High Current**. Disponível em: <<https://www.pololu.com/product/2133>>. Acesso em: 1 jul. 2016.
- [5] G1 SÃO PAULO. **Roubo de carga e a banco crescem na capital e no estado de SP**. 2015. Disponível em: <<http://glo.bo/1HFLeqc>>. Acesso em: 4 jul. 2016.

Nursy Project

Smart Hospital

Elton da Silva

Engenharia de Controle e
Automação
Universidade Federal de Santa
Catarina
Blumenau, Brasil

Gabriel Davanço

Engenharia de Controle e
Automação
Universidade Federal de Santa
Catarina
Blumenau, Brasil

Rafael Lobato

Engenharia de Controle e
Automação
Universidade Federal de Santa
Catarina
Blumenau, Brasil

Resumo—Este projeto será capaz de monitorar a frequência cardíaca de um indivíduo e exibir a leitura na tela de um computador por intermédio de um sistema embarcado para processamento e com comunicação com um computador pessoal (PC), emitindo uma mensagem de alerta em caso de anormalidades. Esse projeto será uma alternativa viável e de baixo custo para hospitais, de maneira a disponibilizar um sistema de monitoramento simples que auxilie o trabalho de enfermeiros (as) e profissionais da saúde.

Palavras-chave — *Monitoramento Cardíaco; Coração; Saúde; Arduino; Embarcados; ky039; programação; UNO; MEGA; nRF24l01+;*

I. INTRODUÇÃO

Vivemos em uma era conectada e automatizada no sentido de simplificar procedimentos, tornando assim o dia-a-dia mais dinâmico. Por intermédio de softwares e sistemas embarcados podemos fazer com que problemas rotineiros, com alta complexidade, possam ser simplificados e de fácil assimilação.

A automação desses processos tem se tornado algo indispensável para diminuir a quantidade de trabalho manual e aumentar o aproveitamento do tempo em atividades onde a máquina ainda não pode ser introduzida.

Esse tipo de implementação é cada vez mais necessário. Um grande exemplo de aplicação disso é o atual modelo do Sistema Único de Saúde (SUS) de gerenciamento e controle do estado vital dos pacientes não atende um modelo que garanta a eficácia na agilidade de atendimento. Números mostram que o índice médio de mortes no Brasil cresceu 28% somente de 2008 a 2015; no Rio de Janeiro estes números são mais alarmantes, crescendo mais de 45% [2].

Pensando nisso, é apresentada a proposta de desenvolvimento de um protótipo que tem como principal objetivo o de gerenciar, analisar e dinamizar informações relevantes para os profissionais de saúde em uma sala de

internação, simplesmente monitorando o estado de pacientes e emitindo mensagens de alerta em caso de anormalidades.

Como principal diferencial este sistema tem como objetivo o de monitorar uma gama de componentes e relacioná-los com os seus respectivos usuários, trazendo assim, um novo conceito de hospital automatizado, onde a necessidade de preenchimentos de diversos formulários aleatórios seria dispensado, interligando todos os setores do hospital, trazendo um atendimento mais eficaz e sucinto.

II. DESENVOLVIMENTO

A. Descrição do problema

Visa-se, através de um sistema embarcado Arduino, realizar, de forma mais eficiente, barata e ágil, uma significativa melhora em relação ao atual sistema de monitoramento e observação de pacientes em hospitais e clínicas para tratamentos de saúde, feita, quase que em sua totalidade, de maneira manual e não-automatizada.

O projeto envolve softwares de programação e interação entre o homem e a máquina para, constantemente, verificar e monitorar os sinais vitais do indivíduo internado e, através desse fluxo de dados, gerar relatórios, tabelas comparativas e eventuais anormalidades que podem vir a acontecer com o paciente em questão.

Monitoramento este que se torna demasiadamente complexo de ser realizado de forma manual através dos profissionais que atuam no local.

Através da web, os funcionários poderão acessar de forma ágil e simples informações de qualquer paciente e todos os sinais de anormalidade serão automaticamente enviados para o centro de enfermaria do hospital ou clínica médica em questão, além de constantemente monitorar os pacientes deve informar, não somente a frequência do batimento cardíaco do indivíduo, mas também, especificamente, a localização desse

paciente no complexo de enfermagem para facilitar o deslocamento da equipe médica.

B. Proposta de projeto

Analisando o atual modelo de gerenciamento em hospitais, há uma grande dificuldade por parte de enfermeiros e enfermeiras de manter a atenção em todas as salas de atendimento. Isso porque é muito complicado para os agentes suprir a demanda gerada pela quantidade de pessoas por hospital. Muitas vezes há alguma emergência ou algum imprevisto com pacientes e a informação chega nos responsáveis em um tempo maior que o esperado, reduzindo a eficácia com a qual o problema será solucionado, podendo comprometer a saúde de certos pacientes.

É nesse contexto que introduzimos nosso projeto de Arduino, uma maneira de manter os profissionais responsáveis inteirados de todo imprevisto que possa ocorrer com equipamentos e pacientes, manter um canal de comunicação e monitoramento ativo e possibilitar diminuir o atraso do período em que acontece uma anormalidade com o paciente até o momento da ação pelo corpo médico do hospital.

Uma das consequências diretas dessa adversidade é a desproporção existente entre profissionais de saúde responsáveis por determinado hospital e a quantidade de cidadãos esperando assistência.

Dessa forma, para criar alguma que pudesse auxiliar esse processo, seria necessária uma alternativa de baixo custo e que pudesse ser amplamente difundido sem dificuldade.

A saída por nós encontrada foi a utilização de sistemas embarcados para criar um equipamento de monitoramento, gerenciamento, processamento e apresentação de todos os dados obtidos a partir da leitura das máquinas que monitoram o estado vital do paciente encontradas nas salas de internação.

Apresentamos o Nursy, sistema embarcado que irá realizar todo esse processo.

A partir de um sistema embarcado que faz-se, por meio de um microcontrolador Arduino UNO para gerenciar todas as informações, dois sensores para a comunicação sem fio (nRF24L01) e um sensor de leitura de batimentos cardíacos (ky039) que fornece dados do pulso do paciente que são mostrados, através de um software que converte tais dados em gráficos, o eletrocardiograma da pessoa avaliada pelo sensor.

A implementação desse software não denota o fim do monitoramento feito pelos agentes, tampouco a realização de periódicos exames pelos mesmos, apenas o auxilia neste processo tão complexo

C. Especificações técnicas

Para que tal projeto seja implementado, foi necessário orçar alguns componentes para sua implementação. Segue abaixo uma relação dos componentes orçados e especificados:

TABLE I. TABELA DE PREÇOS

Tabela de Preços		
Componente	Quantidade	Preço (R\$)
Arduino Uno	1	20,00
Arduino Mega	1	89,90
Módulo NRF24L01	2	7,44
Sensor ky039	1	14,89
Total		139,67

nRF24l01+:

Na transmissão e leitura de dados, foi selecionado o módulo nRF24L01, este, por sua vez, foi escolhido devido seu alcance de transmissão de até 50m na versão padrão e na versão com antena seu alcance pode chegar até 200m [9] (esse alcance depende de vários fatores tais como o caminho entre os sensores, pois o sinal perde força quando existem obstáculos à sua frente.

Outro fator interessante que é o seu diferencial é o baixo consumo de energia para operação (que possibilita uma maior duração de pilhas por exemplo, pois seu consumo será mínimo), isso também é útil para alocar esses módulos em locais onde não demanda muita manutenção (em locais de difícil acesso) ou intervenções constantes.

O módulo NRF24L01 pode transmitir ou receber dados via rádio para outro módulo similar, possibilitando a troca de informações dentro de um sistema de maneira rápida e eficiente (sem o uso de cabos). Seguem abaixo algumas das especificações técnicas mais importantes para esta aplicação:

- 250kbps, 1Mbps and 2Mbps on air data rates
- 11,3mA TX at 0dBm output power
- 13,5mA RX at 2Mbps air data rate
- 26 μ A in standby-I
- 6 data piper Multiciever™

Fig. 1. nRF24L01:



Fonte: Sunrom.com

ky039:

Para a detecção dos batimentos cardíacos foi escolhido o sensor ky039 devido à sua fácil aquisição e, por consequência, barateamento dos custos do projeto.

Outro fator determinante na decisão desse equipamento foi sua precisão e sensibilidade a mínimas variações de luminosidade, tornando assim o circuito muito mais preciso e eficaz nas menores alterações do ambiente, levando à uma maior certificação de que o equipamento funcione como o esperado.

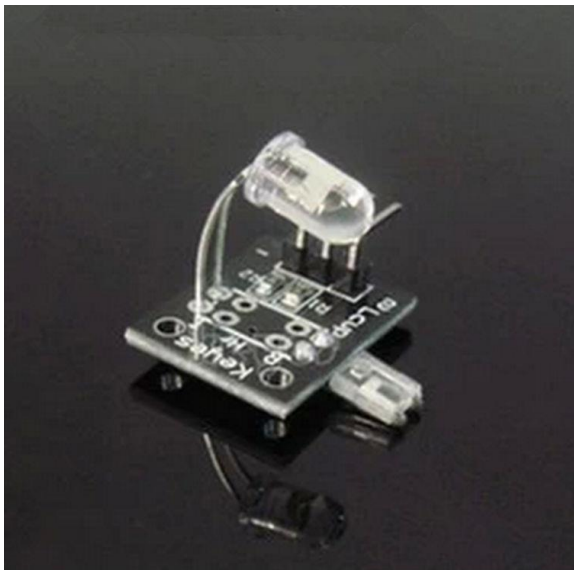
O sensor ky039 utiliza um LED infravermelho e um fototransistor para detectar o pulso cardíaco a partir de um dedo do paciente, onde um LED vermelho pisca a cada pulso emitido.

O LED é posicionado em um lado da ponta do dedo e o fototransistor no lado contrário para obter o fluxo de sangue

emitido. Quando o sangue passa pelo dedo e causa um pulso na pressão sanguínea, a resistência do fototransistor muda levemente e é medido e passado para o microcontrolador.

Enquanto o programa estiver sendo executado, os valores dessas variações são transformados em gráfico e mostrados na tela. Este sensor conta com:

Fig. 2. Sensor ky039:



Fonte: arduino.cc

Para o processamento e envio dos dados foi escolhido o Arduino no devido a sua praticidade e flexibilidade. Este microcontrolador permite, de maneira clara e eficiente, realizar uma infinidade de projetos utilizando apenas alguns componentes de baixo custo.

Outro fator de destaque é que o Arduino é um microcontrolador open-source, ou seja, em que se pode colaborar de maneira pouco burocrática com o desenvolvimento e aprimoramento dos softwares.

Sua flexibilidade é demonstrada na sua voltagem de operação que pode ser usada com 3.3V ou 5V. Além disso, podem-se usar vários componentes de maneira simultânea, pois possui 14 pinos digitais e 6 analógicos na versão UNO, a versão MEGA conta com 52 pinos digitais e 6 analógicos [1].

Arduino Uno:

Microcontrolador responsável em receber o sinal do sensor, processar esses valores e envia-los para o Arduino MEGA, abaixo temos suas principais especificações técnicas:

TABLE II. ESPECIFICÇÕES TÉCNICAS ARDUINO UNO

Especificações técnicas	
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P)
	of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz

Fonte: Arduino.cc

Arduino MEGA:

Responsável em receber os sinais enviados pelo Arduino UNO e enviar essa informação para um software qual irá realizar a plotagem desses valores (existem diversos softwares quais podem ser realizada essa plotagem, exemplos: MATLAB, LabView, Processing, etc...). Abaixo temos as principais especificações técnicas:

TABLE III. ESPECIFICÇÕES TÉCNICAS ARDUINO MEGA

Especificações técnicas	
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Fonte: Arduino.cc

D. Testes operacionais

Logo no início do projeto, foram realizados procedimentos experimentais apenas para verificar o funcionamento dos sensores. No caso do sensor de pulso, utilizamos um algoritmo que faz piscar um LED convencional (não incluído no produto final) conforme os dados fluem pelo sistema. Para o NRFL24L01 utilizamos alguns códigos da biblioteca NRF disponível na própria IDE do Arduino para comunicar os dois microcontroladores.

Teste 1: intensidade de luz do ky039

Observou-se, após alguns testes e leitura de especificações do sensor, que seu desempenho e os números expostos na tela Serial apresentavam variações consideráveis dependendo da quantidade de luz que o fototransistor era exposto.

Para minimizar esses efeitos foi tentado, inicialmente, isolar o sensor do contato com a luz colocando-o em um local fechado e com a mínima incidência de luz. Após algumas pesquisas, foi encontrada a equação:

$$S_t = \alpha \cdot Y_t + (1 - \alpha) \cdot S_{t-1}$$

Onde S_t é o valor filtrado pelo sensor, α é a constante que, nesses experimentos foi utilizada com os valores de 0,1, 0,2 e 0,3, Y_t representa o valor bruto lido pelo programa e S_{t-1} é o valor anterior da mesma equação, com o primeiro S_t padronizado em 0 [8].

Após sucessivos testes foi observado que a constante muda de acordo com o nível luminosidade do ambiente.

Teste 2: Comunicação do nRF24L01+

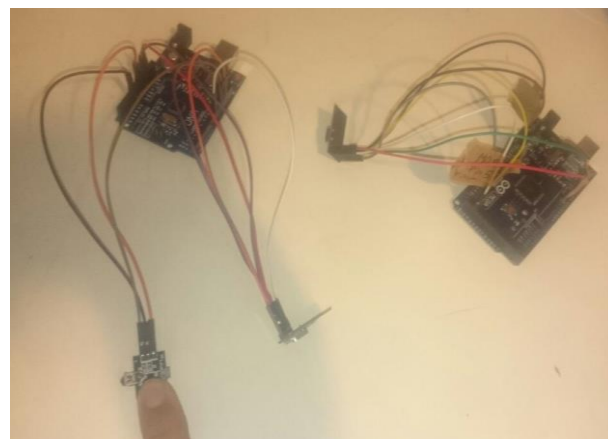
No início, foi utilizado, apenas para o fim de atestar o funcionamento dos sensores foi utilizado o exemplo GettingStarted da biblioteca NRF da IDE do Arduino em sua versão 1.6.

Após estes testes se mostrarem satisfatórios, foram realizados alguns experimentos para a verificação de envio e recebimento de informação pelos dois sensores do conjunto através dos exemplos GettingStarted_CallResponse e posteriormente do HandlingData também da biblioteca RF24.

E. Funcionamento

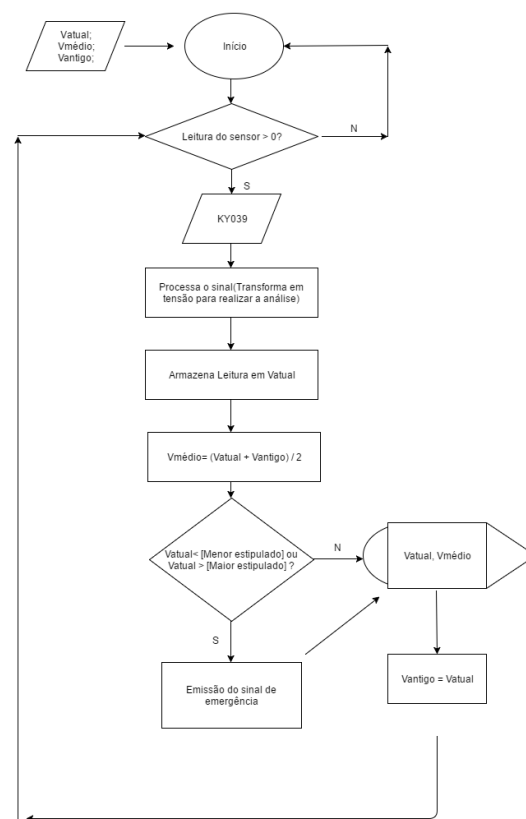
O princípio de funcionamento deste protótipo é relativamente simples: após o indivíduo posicionar o dedo no sensor os dados dos seus batimentos serão colhidos pelo sensor, enviados ao microcontrolador que, por sua vez irá processar este sinal, quantificar os batimentos cardíacos e enviar para a central (neste caso o segundo Arduino que irá transmitir para um PC) exibindo essas informações de maneira ordenada e de fácil entendimento.

O ciclo se repete enquanto o dedo estiver posicionado no sensor. Caso ocorra alguma alteração brusca ou anomalia a central é informada automaticamente e a informação fica registrada para futuras consultas.



Para um entendimento dinâmico deste processo é recomendado consultar o fluxograma deste processo abaixo:

Fig. 4. Fluxograma:



Fonte: draw.io

III. RESULTADOS OBTIDOS

Pode-se concluir que o projeto necessita de mais estudo e pesquisa para sua continuidade e aprimoramento. Foram encontrados muitos problemas relacionados ao ky039, tais como:

- A instabilidade nos parâmetros do envio da informação;
- O incorreto funcionamento de vários sketches encontrados;

E isso pode ser confirmado através de pesquisas online, através de vários fóruns de discussão. Outros usuários

deste mesmo sensor tiveram problemas parecidos e até o presente momento não chegaram a uma solução.

Crê-se, por tanto, que este sensor não atende as expectativas que lhe são atribuídas.

Portanto, em suma, pode-se dizer que devem ser verificados e pesquisados novos métodos para desenvolver este projeto de forma mais contundente.

IV. CONCLUSÕES E PERSPECTIVAS FUTURAS

Para a continuidade do projeto visamos a implementação de gadgets que interajam com o sistema inicial que foi planejado.

A. Implementação de aplicativo

Visa-se, para a rápida comunicação e aprimoramento da eficiência do produto, a inserção de um aplicativo para aparelhos móveis com o intuito de fazer a ponte entre os dados processados e os profissionais de saúde que fazem o gerenciamento manual dos pacientes.

Através de um aplicativo que emite um sinal sonoro ou vibratório no aparelho móvel, que seja equivalente ao sinal de emergência sinalizado pelo software de controle da central, visamos transformar o sistema, que atualmente é pensado somente para redes fechadas, um projeto multi plataforma.

B. Aprimoramento da captação de dados

Para o futuro do projeto, pensamos em expandir a quantidade e a diversidade de sinais vitais processados, como por exemplo, o monitoramento da pressão sanguínea do paciente (outro poderoso índice de anormalidade da saúde).

Além do processamento da pressão sanguínea, há também a preocupação com outros sinais vitais, como nível de oxigênio e temperatura do indivíduo assistido.

C. Banco de dados

Outra ferramenta importante para o projeto, seria um banco de dados de funcionamento conjunto ao sistema para o armazenamento e combinação de dados que auxiliariam no momento de esperar anormalidades e comportamentos diferentes dos dados captados.

Como exemplo, existem remédios que alteram os batimentos cardíacos do paciente a ponto de fazê-los ultrapassar os parâmetros do sistema. Caso existisse a informação de que o remédio seria inserido na corrente daquele indivíduo, o sistema poderia buscar e processar os dados dentro do banco de dados e, dessa forma, se ajustar as diversas situações presentes em um tratamento de saúde.

Outra implementação interessante seria as especificações do indivíduo que está sendo analisado, pois é sabido que os sinais vitais mudam conforme a faixa etária e outras especificidades do paciente. Tendo um banco de dados para tal facilitaria a adaptação do sistema a estas condições variáveis.

V. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BANZI, M. et al. Arduino Froum. **Arduino**, 2016. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 02 de Junho de 2016.
- [2] BARRETTO, E. O Globo. **Rio tem a maior taxa de mortalidade no SUS em três décadas**, 2016. Disponível em: <<http://oglobo.globo.com/brasil/rio-tem-maior-taxa-de-mortalidade-no-sus-em-tres-decadas-18770526>>. Acesso em: 22 de Junho de 2016.
- [3] BORGHUIS, D. Detection of the heartbeat module ky039. **Tkkrlab Enschede**, 2016. Disponível em: <https://tkkrlab.nl/wiki/Arduino_KY-039_Detect_the_heartbeat_module>. Acesso em: 05 de Junho de 2016.
- [4] DEPARTAMENTO DE MONITORAMENTO E AVALIAÇÃO DA GESTÃO DO SUS. Pannel de indicadores do SUS. **Portal saúde**, 2006. Disponível em: <<http://www.redeblh.fiocruz.br/media/indicadsus1.pdf>>. Acesso em: 23 de Junho de 2016.
- [5] LTD, J. Nursy Project.html. **Fluxograma**, 2016. Disponível em: <<https://www.draw.io/#DNursy%20Project.html>>. Acesso em: 07 de Junho de 2016.
- [6] MURPHY, J. About. **Pulsesensor**, 2016. Disponível em: <<http://pulsesensor.com/>>. Acesso em: 05 de Junho de 2016.
- [7] PRADO, S. Embarcados. **Forum**, 2007. Disponível em: <<http://www.embarcados.com.br/>>. Acesso em: 16 de Junho de 2016.
- [8] QUEVEDO, T. B.; SCALCO, M. T. Youtube. **Detector de Batimentos Cardíacos Com Ky039 e Arduino**, 2016. Disponível em: <https://www.youtube.com/watch?v=RgE3AXdq_zY>. Acesso em: 30 de Junho de 2016.
- [9] NORDIC SEMICONDUCTOR. nRF24L01+. **Preliminary Product Specification v1.0**, 2008. Disponível em: <https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus_Preliminary_Product_Specification_v1_0.pdf>. Acesso em: 22 de Junho de 2016.
- [10] EVERLIGHT. 5mm Infrared LED , T-1 3/4. **Technical DataSheet**, 2005. Disponível em: <https://cdn-shop.adafruit.com/datasheets/IR333_A_datasheet.pdf>. Acesso em: 07 de Julho de 2016.
- [11] FAIRCHILD SEMICONDUCTOR. Plastic Silicon Infrared Phototransistor. **Technical DataSheet**, 2007. Disponível em: <<https://www.fairchildsemi.com/datasheets/QS/QSC112.pdf>>. Acesso em: 07 de Julho de 2016.

Happy Pet

Porta automatizada para animais

Bruna Pinto, Douglas Liao, Gabriel Mafra, Gabriel Zimmer e Martha Bloemer

Universidade Federal de Santa Catarina
Blumenau, Brasil

Resumo— *Esse trabalho consiste na implementação de uma porta automatizada destinada a animais de estimação, prototipada com diferentes peças e componentes, sendo o principal deles o Arduino. A porta apresenta sensores de umidade, presença e buzzer que tem como funções reconhecimento de chuva, presença do animal e aviso sonoro para que o animal entre em casa assim que o sensor de umidade recebe a informação de que irá chover, respectivamente. Tem como objetivo facilitar a vida doméstica e trazer maior comodidade e segurança para os animais de estimação.*

Palavras-chave—automação; Arduino; sensor; animais; Porta;

I. INTRODUÇÃO

Esse trabalho tem como objetivo realizar um projeto sobre uma porta para animais de estimação que, ao receber informações climáticas por meio de componentes inseridos no arduino, destrava e o animal possa entrar em casa de forma segura, prática e evitando riscos à saúde do animal.

Foram feitas pesquisas prévias para a realização do projeto, e já existem protótipos de portas automatizadas, porém não em funcionamento com sensores que serão utilizados no projeto percebendo assim que a automação que visa animais domésticos é um campo ainda não muito explorado mas que tende a ser cada vez mais promissor.

O trabalho pretende facilitar a vida doméstica, fazendo com que o dono do animal fique tranquilo com o animal em casa enquanto ele não estiver presente. Desenvolveremos sobre como foi feita a montagem do projeto, quais peças sensores utilizamos, como foi desenvolvida a nossa programação e quais foram os nossos resultados, e também possíveis aprimoramentos futuros do projeto.

II. DESENVOLVIMENTO E IMPLEMENTAÇÃO DO PROJETO

Foi decidido desenvolver uma porta inteligente para condicionar animais domésticos a um local específico. Ela permite que em caso de chuva, o animal entre na residência, impedindo assim, acidentes. Utilizamos uma placa Arduino Uno e alguns sensores (Sensor de Umidade, Sensor de Toque, Sensor de presença, um servomotor, um protoboard. Fizemos

uso também de um Buzzer que emite um som permitindo assim que o animal saiba quando a porta estará destrancada.

A equipe HappyPet procurando unir a a vida corrida das famílias modernas com a qualidade de vida dos animais domésticos, teve como objetivo facilitar a locomoção dos animais de estimação para o ambiente interno em caso de problemas que possam influenciar no bem-estar do animal

Nosso projeto facilita a vida no ambiente familiar, pois havendo ou não presença de pessoas no ambiente, o animal de estimação tem a oportunidade de usufruir a parte externa do lar até o momento de interferências externas (como o fator climático, por exemplo) ou até a tarefa ser executada manualmente por alguém.

O objetivo foi implantar uma porta para cachorros contendo um sensor de umidade que detecta quando está chovendo. Então o cão recebe um sinal, a partir de um alto-falante, que deve entrar em casa em caso de mau tempo. A porta apresenta um sensor de movimento no sentido horizontal na parte inferior que detecta quando há algo na porta, evitando acidentes. O sistema também tem um pino de trava que após a entrada do cachorro, trava a porta evitando que o mesmo saia novamente.

A porta é feita com ferro e tem 20cm X 25cm de área, tem como principal função fazer com que o cachorro entre em casa após um aviso sonoro, permanecendo limpo e seguro, facilitando a vida do dono e do animal de estimação, e também conservando a saúde do pet.

Segundo dados da Pesquisa Nacional de Saúde (PNS) de 2013, feita pelo Instituto Brasileiro de Geografia e Estatística (IBGE), 44,3% dos domicílios do Brasil possuem pelo menos um cachorro, o equivalente a 28,9 milhões de unidades domiciliares. Em relação a presença de gatos, 17,7% dos domicílios possuem pelo menos um, o equivalente a 11,5 milhões de unidades domiciliares.

Animais domésticos em contato com a chuva podem adoecer, ficam sujos e podem passar a exalar um cheiro forte, trovoadas são perigosas para qualquer animal e mudanças repentinas de temperaturas podem tornar o ambiente externo de seu domicílio um risco a saúde. Como as previsões são atualizadas constantemente e por vezes erram muitos deixam simplesmente seus animais presos, pois saem por longos períodos e temem não estar em casa em um momento que é

necessário colocar o animal em área coberta ou dentro de seu domicílio.

Tem se a necessidade de controlar a entrada e saída de animais e acompanhamento da previsão do tempo para aquele local. Pensando nisso desenvolvemos uma porta inteligente que seja capaz de realizar essa função.

Utilizando o micro-controlador Arduino Uno desenvolvemos um protótipo que atenda as necessidades propostas anteriormente. Para isso, precisamos contar com algumas peças que foram listadas abaixo:

- Arduino

Para desenvolvermos a nossa porta inteligente será utilizado uma placa arduino Uno, a figura 1 apresenta um exemplo de placa Arduino Uno, ele será "cérebro" do nosso projeto, nele são colocadas as programações e partir dele os sensores realizam a tarefa desejada. O placa Arduino é um microcontrolador baseado no ATmega328. Ele possui 14 pinos de entrada/saída digital, 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação uma conexão ICSP e um botão de reset. Ele possui todos os componentes necessários para suportar o micro-controlador, simplesmente conecte a um computador pela porta USB ou alimentar com uma fonte ou com uma bateria e tudo pronto para começar.

A placa é programada através da comunicação serial, pois o microcontrolador vem já vem programado com o bootloader. Sendo assim não há a necessidade de um programador para fazer a gravação do binário na placa. A comunicação é feita através do protocolo STK500. A programação do microcontrolador também pode ser feita através do conector utilizando um programador ATMEL.

A placa Arduino possui 6 conectores:

IOREF - Fornece uma tensão de referência para que shields possam selecionar o tipo de interface apropriada

RESET - pino conectado a pino de RESET do microcontrolador. Pode ser utilizado para um reset externo da placa Arduino3,3 V.

GND - pinos de referência, terra

VIN - pino para alimentar a placa através de shield ou bateria externa.



Figura. 1. Exemplo de placa Arduino Uno

- Sensor de Umidade

O sensor de temperatura e umidade que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90% realizando com precisão a tarefa necessaria para controlar se há chuva no local.. O elemento sensor de temperatura é um termistor do tipo NTC e o sensor de Umidade é do tipo HR202, o circuito interno faz a leitura dos sensores e se comunica a um micro-controlador através de um sinal serial de uma via. Esse sensor terá a função de detectar se está chovendo ou não, para a partir daí poder destravar a porta. a figura 2 apresenta um exemplo de sensor de umidade.

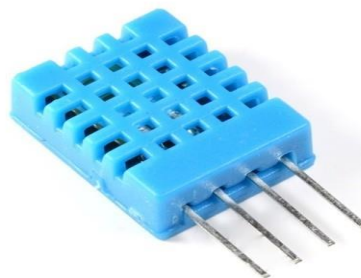


Figura. 2. Exemplo de Sensor de Umidade.

- Protoboard

O Protoboard é uma ferramenta para a montagem de circuitos eletrônicos, a figura 3 apresenta um exemplo de protoboard. Possui 400 pontos e em sua parte inferior há um adesivo que permite colá-lo em uma superfície isolante. São 100 pontos de distribuição e 300 pontos de conexão terminal. Possui coordenadas coloridas para facilitar a visualização na hora de colocar seu componente.

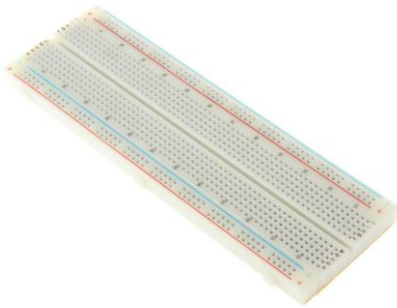


Figura. 3. Exemplo de Protoboard

- Servo

Os servo-motores possuem engrenagens e um eixo que pode ser controlado com muita precisão, a figura 4 apresenta um exemplo de servo. Servos standard permitem que o eixo seja posicionado em vários ângulos, geralmente entre 0 e 180 graus. A biblioteca Servo pode suportar até 12 motores. Esse motor será a “trava” do nosso projeto, quando o sensor de proximidade detectar que o animal passou pela porta o servo será acionado e irá girar 90° fechando assim a porta. Ao decorrer da evolução do projeto iremos devolver uma trava melhor que possa dar uma segurança maior ainda para o Happy pet.

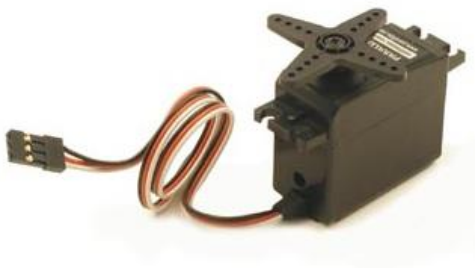


Figura. 4. Exemplo de Servomotor.

- Buzzer

Com o Buzzer podemos produzir todos os tipos de campainhas piezoelétricas e campainhas magnéticas. Iremos utilizar esse sensor para avisar quando o animal que ele deve entrar na casa, poderá levar um certo tempo até conseguimos treinar o animal mas com certeza essa é a melhor forma de atrair o animal para a residência.



Figura. 5. Exemplo de Buzzer.

- Sensor de presença

O Sensor de Movimento D- SUN consegue detectar o movimento de objetos que estejam em uma área de até 7 metros. Caso algo se movimentar nesta área o pino de alarme é ativo. É possível ajustar a duração do tempo de espera para estabilização, a estabilização pode variar entre 5-200 seg. É necessário uma fonte de 5v no GND e VCC. O pino DADOS indica o sinal de saída que poderá ser 'Alto' indicando movimento ou então 'Baixo' indicando nenhuma movimentação

Quando a saída é acionada pelo movimento detectado ela ficará em alto por um curto período de tempo, mesmo se não haja mais movimento, tendo assim um certo “delay”, mas que dificilmente atrapalhará o projeto. O tempo em alto pode ser setado variando o potenciômetro 'Time', sendo que o outro altera a sensibilidade.

Utilizaremos esse sensor para detectar quando o cachorro passou pela porta e então o servo será acionado.



Figura. 6. Exemplo de Sensor de presença.

A Tabela I. apresenta a relação de custo da implementação do projeto:

TABELA I.

Tabela de Custos		
Peça	Quantidade	Valor (em R\$)
Arduíno Uno	1	50,00
Servo	1	22,00
Buzzer	1	20,00

Tabela de Custos		
Peça	Quantidade	Valor (em R\$)
Jumper	30	10,00
Sensor de Umidade	1	9,00
Solda de estanho	-	6,00
Sensor de Movimento	1	12,00
Adesivo	-	70,00
Ferro	-	40,00

- Lógica da programação

A programação do Arduino desenvolvida pela equipe Happypet, tem como objetivo de verificar a umidade local, e também a pressão para detectar a chuva. Chamar a atenção do PET com o buzzer, com a intenção de chamar ele para dentro de casa, e quando ele entrar em casa, trancar a porta até que pare de chover.

A lógica da programação consiste em algumas etapas. Segue a seguir a lógica:

Incluir bibliotecas: Servo motor (para girar a trava da porta), o sensor de pressão barométrica (conhecido como wire). As bibliotecas `adafruit sensor` e `adafruit bmp 085` são complementos do sensor de pressão barométrica são os mesmos para o sensor `bmp180`. A biblioteca de temperatura e umidade é conhecida como `DHT`.

A seguir as definições de entradas para as portas do arduino (trabalhando diretamente na placa), tem como entrada a porta A2 (segunda porta analógica) e também está definido o tipo de sensor de umidade (`DHT11`).

Foram criadas nove funções, e cada uma definindo a entrada para cada sensor e motor.

O pino do Servo, está localizada na entrada nove.

O pino Buzzer, que é o motor que emite o som, e pode ser programado para emitir outras frequências, está localizado na entrada 10.

O pino do Led, que está na cor verde e tem uma função específica, está localizado na estrada 11.

O pino do sensor de presença, no qual não é necessário livreria e/ou biblioteca para a sua funcionalidade, está localizado na porta 6.

Definimos também duas funções para travar e destravar, fazendo com que o servo motor gire, em graus, de 0° a 90°. (0° considera-se porta destravada, e 90° considera-se porta travada).

A inicialização do `DHT` para colocar as portas, iniciar o servo motor, e as livrerias do `adafruit BMP 085` também são necessárias.

A programação também é necessária uma estrutura de dados, que serão relevantes para o controle.

Ponto flutuante ou virgula flutuante para definir a umidade local, uma função inteira para ler a pressão atmosférica, dois dados booleanos, que é um tipo de dado primitivo que possui dois valores, um que é falso, e outra que é verdadeira, que seria a condição se está chovendo ou não (se é falso, ou se é verdadeiro). E também executou ou não, isto é, se o cão passou ou não da porta.

Essas estruturas de dados fornecem dados atuais, em tempo real.

Em seguida serão descritos 9 funções que terão a função com o void. O void, que em inglês quer dizer vazio. Ele nos permite fazer funções que não é necessário retornar parâmetros, e nem retornar em nada.

Um void (função) específico para destravar a porta. Essa função que não precisa de retorno, descreve de como a porta será destravada. Ela começa com a porta travada (um função de um inteiro de noventa graus), gira constantemente o servo motor no sentido anti-horário de graus em graus até chegar em zero graus, e isso quer dizer que ela está destravada, tudo isso acontece com um retardo (delay) de trinta mili-segundos. Após ele destravar a porta, será escrita no monitor serial (que ele imprime dados na porta serial, em formato legível por seres humanos).

Um void (função) específico para travar a porta. Ela é iniciada com o parâmetro de zero graus, gira constantemente até que alcance noventa graus. Isso também ocorre com um retardo de 30 mili-segundos por grau. E quando terminar o giro do servo motor, irá aparecer no monitor serial que o servo TRAVOU a porta.

Outra função específica que irá ler a umidade com um retardo de apenas 250 milisegundos! Em outros casos, se o sensor for antigo, irá demorar 2 segundos para a leitura. A seguir será acrescentado para a leitura um ponto flutuante para ler a umidade.

Se tiver algum erro de leitura do sensor de umidade, será acrescentado no monitor serial um erro de leitura (isso em palavras), se não tiver algum erro, ele grava as condições de umidade e imprime os dados atuais de umidade em porcentagem.

Um void (função) em especial para os sensores de pressão barométrica (atmosférica), ele inicia buscando eventos e grava na memória, e se ele iniciar, ele imprime no monitor serial em hPa (hectoPascal, unidade de medida de pressão) da condição atual local.

Foram acrescentados informações não necessárias ao programa, como o de temperatura, que é captado pelos dois sensores, de umidade e de pressão, e é imprimido no monitor serial em graus Celsius.

Para mais precisão de altitude, foi colocado a pressão ao nível do mar. E assim, foi acrescentado mais um ponto flutuante para a pressão ao nível do mar. Após a este momento, será escrito de forma legível no monitor serial a altitude ao nível do mar em metros. Se ocorrer algum erro no sensor de pressão barométrica, será escrito no monitor serial que tem um erro no sensor. E haverá mais um delay de 1 segundo.

Outro void (função) será mantido para o tocar o Buzzer (motor de som), foi acrescentado mais uma função de tempo para adicionar com precisão a frequência. Enquanto tocar o buzzer, será ligado a luz alta, e depois com o a função tempo, irá diminuir gradativamente até que fique baixo.

Um dado booleano que é a condição de chuva, se irá chover verdadeiro ou falso, o programa retornará a ler as condições de clima tempo se o resultado for falso, isto é, se não estiver chovendo, se começar a chover, será escrito no monitor serial alertando que irá chover, se não for verdade (falso), ele indica que não irá chover no monitor serial.

A função void para acender a luz, com dado booleana será verdade se e somente se, ele acender o Led e deixar com a luz forte, e se for falso, ele ira deixar com a luz fraca.

Dado booleano é informado se o cachorro entrou, isso só é for verdade se e somente se o sensor de movimento for alto, até que ele não veja mais nenhum movimento, ele informará que o Cão entrou, se for falso, o cão não entrou ainda.

Void para imprimir os dados atuais (umidade, pressão, e a situação da porta, para verificar se o cão passou sim ou não), se isso for verdade, ela mantém travada, se isso for falso, ela mantém destravada.

Para todas essas condições, funções iniciarem e/ou funcionarem, haverá mais uma função void para inicializar o servo motor, o sensor de umidade (DHT11), sensor de pressão barométrica (BMP180), motor sonoro (BUZZER), o Led, e por último, o sensor de movimento, e será escrito no monitor serial os sensores que foram ativados e os que estão com defeito.

E por último, o void para operar de forma continua todos os sensores que foram instalados no arduino uno até que o usuário tire os cabos da tomada por certo motivo.

E a partir que loop foi iniciado, ele irá ler as condições de umidade e de pressão, e se tiver condição de chuva, o servo motor girará noventa graus para destravar a porta, o LED irá acender, o motor de som irá ser acionado (com o delay de 2 segundos), e enquanto o cão não passou, o buzzer irá continuar a tocar com o delay de 1 segundo.

Se o sensor de movimento detectar, e depois não detectar mais nada, isso quer dizer que o cão passou, e que o servo motor irá girar no sentido anti-horário para travar a porta, o buzzer irá parar de tocar e fim.

III. CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS

Não foi realizado tudo o que foi planejado inicialmente. A ideia seria conectar a placa Arduino a um módulo wi-fi, permitindo assim, o acesso à temperatura e a umidade quando houvesse previsão de chuva a porta destravaria de acordo com a previsão para que nosso objetivo final obtivesse maior sucesso, e ou monitoramento a distância, para saber das informações do tempo local.

No planejamento inicial, tínhamos como objetivo utilizar da voz do dono do animal de estimação como um “alerta” para que o cachorro passe pela porta quando houvesse mal tempo.

O cachorro teria também uma coleira e a porta teria um sensor, que ao detectar o cachorro com a coleira a uma determinada distância, permitiria que a porta destravasse e assim, o cachorro pudesse entrar.

E também mais sensores acoplados no interior do domicílio, para que se caso chover e o animal já esteja dentro de casa, não será necessário tocar o sensor do buzzer e avisar para o pet entrar.

Outro aprimoramento do projeto seria programar a porta para que ela detecta-se a presença de mais de um animal de estimação, ou seja, atualmente o projeto é unicamente para domicílios com apenas um pet.

Citamos também algumas dificuldades de aceitação do público. Ainda existe receio do público-alvo pelo uso de equipamentos automatizados, pois eventualmente podem apresentar problemas como falhas no sensor e o equipamento não funcionar corretamente, ou então prejudicar o animal de estimação lhe causando danos físicos no caso da porta não abrir, ou então pelo fato do custo benefício ser um pouco alto Ref.[1] e Ref.[2] pela inserção de todos os componentes no protótipo.

IV. RESULTADOS

Os resultados do nosso protótipo conseguiram chegar nas expectativas da equipe, a apresentação visual do trabalho foi acima do esperado. No começo do projeto, tivemos algumas dificuldades de programação e falhas nos sensores, tivemos dificuldade na soldagem e montagem no projeto entretanto após alguns testes, aproximadamente 9 das 10 tentativas foram acertos.

E também houve um imprevisto nos sensores de captação de chuva, pois devido a imprevistos climáticos, foi realizado uma simulação de chuva. Talvez a maior dificuldade foi desenvolver um padrão de programação para que o Happy pet realmente soubesse quando está chovendo, descobrir qual é o índice de umidade que indica que a chuva.

Tivemos também um resultado satisfatório com relação aos custos do projeto, havendo um custo total de R\$ 239,00. Considerando as expectativas acima do esperado acreditamos que mesmo desenvolvendo melhor o projeto, tendo assim um determinado encarecimento, ainda sim poderemos ter um produto de baixo custo e acessível sem perder a qualidade

Realizamos a aplicação do projeto com animais de estimação assim que o protótipo foi finalizado e houve boa adaptação dos mesmos a porta automatizada.

Por fim, os resultados foram satisfatórios, cumpriu os seus objetivos e está pronto para uso doméstico com a sua programação guardada na memória.

VI. REFERENCIAS

- [1] C. Lauro, *Engenharia de automação industrial*. Grupo Gen-LTC, 2000.
- [2] C. Benjamin. *A Revolução dos robos e o impacto socioeconomico da automação*. Busca Vida, 1989.

Gerenciamento Integrado de Parâmetros Agronômicos para Cultivo Indoor

André Filipe Obenaus – Bruno Hasse Leone – Christian Mailer
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina – Campus Blumenau
Blumenau, Brasil

Resumo — Alimentar os grandes centros urbanos no século XXI é um desafio crescente. Com a diminuição das áreas agrícolas aos arredores das cidades, o alimento precisa viajar distâncias cada vez maiores até chegar à mesa das pessoas. Isso, aliado ao estresse hídrico do plantio tradicional, acarreta problemas ambientais que tendem a crescer com o aumento da população. O objetivo deste artigo é apresentar o conceito de fazendas urbanas e demonstrar um protótipo de sistema de gerenciamento integrado de cultivo indoor que controla parâmetros agronômicos do plantio.

Palavras-chave—fazendas urbanas; controle; estufa; plantio urbano

I. INTRODUÇÃO

Quando o assunto é cidades inteligentes, agricultura não é a primeira coisa que vem em mente, porém, fazendas urbanas têm o potencial de ajudar a resolver vários problemas de grandes centros urbanos. Esse tipo de iniciativa já existe em vários grandes centros como Nova Iorque, Chicago, Berlim, Montreal, Toronto e Vancouver. Na capital alemã, Berlim, estima-se que 80.000 pessoas estão envolvidas com agricultura em áreas urbanas [1]. Esse tipo de atividade, porém, ainda é visto como *hobby*. Para realizar seu potencial completo é necessário o desenvolvimento de novas tecnologias capazes de interligar centenas ou até milhares de produtores com centros de distribuição e consumidores.

Investimentos em fazendas verticais já existem e a multinacional de tecnologia, Phillips, possui uma divisão especializada em iluminação para fazendas *indoor* [2]. No entanto, essa tecnologia precisa ir além. Como, por exemplo, um sistema que possa controlar um pavilhão inteiro de fazendas urbanas por um *smartphone*. Temperatura, umidade do solo, ventilação: tudo ao alcance do agricultor enquanto ele exerce outras atividades. Isso sem levar em conta outras vantagens como: a redução dos impactos ambientais - já que se evita a utilização de máquinas de grande porte movidas a combustíveis poluentes -, a utilização racional dos recursos hídricos - fazendas urbanas conseguem produzir uma mesma quantidade de vegetais usando apenas um décimo do necessário na agricultura convencional -, a diminuição do espaço necessário para o cultivo em grande escala - que poderá ser utilizado para outros cultivos ou até mesmo preservação ambiental - e a dispensa da necessidade de

transporte dos vegetais por grandes distâncias, reduzindo o tráfego urbano de transporte pesado.

Pensando nisso, desenvolveu-se uma estufa de pequeno porte que visa facilitar o cultivo de plantas *indoor*, automatizando ações que muitas vezes necessitam acompanhamento e contendo informações sobre o cultivo de plantas, a estufa é perfeita também para pessoas que não possuem conhecimentos de agricultura.

II. PROBLEMA

Com a crescente preocupação relacionada ao meio-ambiente e sustentabilidade, torna-se evidente a responsabilidade individual para contornar os problemas ambientais, contribuindo-se assim para a saúde humana.

O desperdício material e econômico para a comercialização em grande escala de produtos hortícolas (transporte, empacotamento, agrotóxicos, entre outros) é um dos fatores que podem ser facilmente evitados com o cultivo próprio dessas plantas. Além disso, o cultivo residencial por meio de estufas, evita os efeitos maléficos do uso de fertilizantes químicos e agrotóxicos.

Muitos indivíduos se veem incapazes de realizar o plantio de hortaliças devido ao espaço reduzido (apartamentos), ou ao tempo gasto para o cuidado de tais plantas.

III. PROPOSTA

Foi desenvolvida uma estufa de pequeno porte que visa facilitar o cultivo de plantas *indoor*, automatizando ações que necessitam acompanhamento e contendo informações sobre o desenvolvimento dos vegetais, cujo projeto está ilustrado na Fig. 1. A estufa torna automática a irrigação, a iluminação, o aquecimento e a ventilação das plantas, baseando-se em sensores de umidade do solo, de luz, e de temperatura presentes em seu interior. Por meio da internet, o usuário pode consultar as informações coletadas pelos sensores e escolher o perfil (pré-configurado) desejado para o cultivo (alface, hortelã, salsa, entre outros).

Baseando-se neste modelo, pode ser desenvolvido ainda um sistema que transforma um pavilhão inteiro em uma fazenda vertical, otimizando a produção de alimentos e fornecendo, ao alcance de um aplicativo, todas as informações envolvidas no plantio. Deste modo, haveria uma produção pensando-se no processo como um todo (cultivo,

empacotamento e transporte) que poderia ser administrada de qualquer localidade.

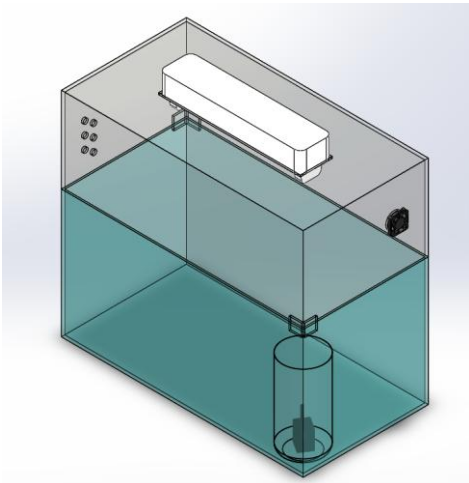


Figura 1- Projeto da estufa

IV. DESENVOLVIMENTO DO PROJETO

A. Especificações técnicas

A estufa automática, Fig. 2, possui dimensão de 30 centímetros por 60 centímetros, com uma altura de 50 centímetros.

Para armazenar a água necessária para irrigação das plantas, um recipiente cilíndrico de diâmetro igual a 8 centímetros e com altura de 20 centímetros foi utilizado.

Sua estrutura foi construída utilizando-se vidro na base e acrílico na cobertura, sendo a última removível para o manuseio das hortaliças. Já o recipiente de água é composto de plástico PET.

Na lateral superior, um cooler fornece a ventilação necessária para a planta e pequenos orifícios servem para auxiliar na oxigenação.

Uma pequena bomba de água é utilizada para bombear o líquido do recipiente de armazenamento para a estufa, por meio de uma mangueira que pode ser posicionada paralelamente ao solo.



Figura 2 - Estufa construída

B. Características do Projeto

A altura de terra utilizada na estufa foi de 10 centímetros a partir de sua base, o que irá fornecer um espaço adequado para o desenvolvimento das raízes das hortaliças.

Na parte superior, uma luz incandescente mantém o ambiente aquecido e duas luzes fluorescentes possibilitam o fornecimento de iluminação para as plantas.

Os sensores de luminosidade (LDR 5mm), Fig. 3, umidade do solo (HL-69), Fig.4, e temperatura (DHT11), Fig. 5, são usados, respectivamente, para controlar as condições de luz, água e calor necessárias para o ideal desenvolvimento do cultivo. Já o microprocessador que será utilizado para o controle de todos os componentes eletrônicos será o *Arduino Uno*, Fig. 6. Conectado ao *Arduino Uno* um módulo *ESP8266 (ESP-01)*, Fig. 7, será responsável por integrar os dados coletados com a nuvem.



Figura 3 - Sensor LDR [3]

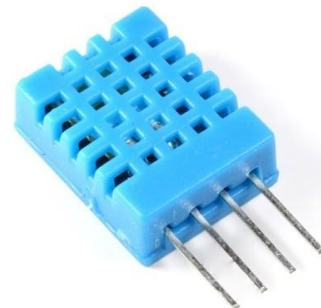


Figura 4 - Sensor DHT-11 [4]

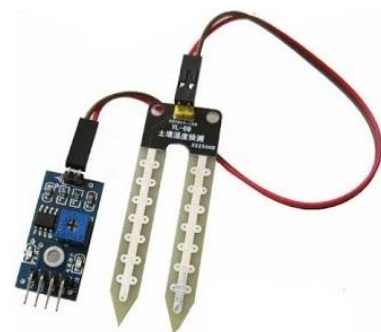


Figura 5 - Sensor HL-69 [5]



Figura 6 – Arduino UNO [6]

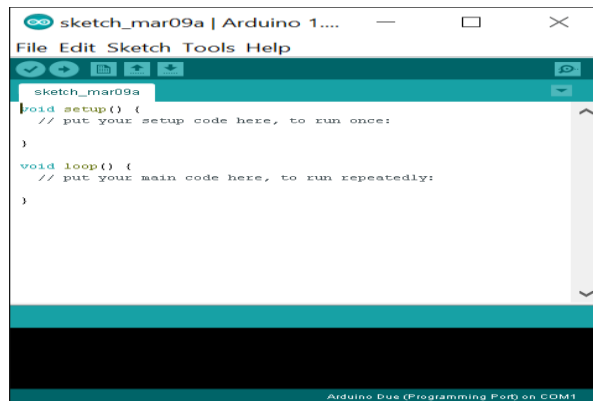


Figura 9 - Arduino IDE [9]



Figura 7 - ESP-01 [7]

Os componentes responsáveis pelo controle (*Arduino*), troca de informações (*ESP8266*), acionamento dos atuadores (relés) e outros necessários ao funcionamento do sistema (resistores, capacitores, etc.) se localizam na estrutura de plástico que suporta as lâmpadas fluorescentes, o que confere proteção aos eletrônicos. Ficarão dispostos como na Fig. 8.



Figura 8 - Disposição dos componentes na estrutura de plástico

A programação lógica para que tais componentes funcionem de forma sincronizada e eficiente será realizada por meio do próprio programa oferecido pelo microprocessador, que é chamado de *Arduino IDE* [8], Fig. 9.

C. Interface WEB

Uma plataforma virtual, Fig. 10, foi criada para melhor controle da estufa e dos atuadores. Para isso, conta com uma interface simples e opções pré-programadas para determinados cultivos.

Plataforma de Controle da Estufa Automatizada - v1.0

Status:

Umidade do Solo: 51%

Temperatura: 25°C

Luminosidade: 951 (Claro)

Luz: Desligada

Cooler: Desligado

Bomba: Desligada

Aquecedor: Desligado

Perfis:

1- Alfaca*
2- Aquecer
3- Resfriar
4- Regar
5- Batata
6- Cebolinha
7- Salsinha

* Perfil Atual

Gerenciar Perfis:

Perfil 1 ▾

Nome do Perfil: Alfaca

	Mínimo	Máximo
Umidade do Solo (%):	30	
Temperatura (°C):	14	28
Luz necessária (horas):	0	

Atual?

Figura 10 - Layout completo da plataforma online

O painel de status, Fig. 11, apresenta os status dos sensores e os estados dos atuadores, com valores que são atualizados com periodicidade de sessenta segundos.

Status:

Umidade do Solo: 51%

Temperatura: 25°C

Luminosidade: 951 (Claro)

Luz: Desligada

Cooler: Desligado

Bomba: Desligada

Aquecedor: Desligado

Figura 11 - Painel de Status

O painel de perfis, Fig. 12, apresenta os perfis selecionáveis e mostra o perfil no qual está funcionando a estufa no momento. Os perfis existentes são perfis pré-definidos com os valores selecionados após pesquisa com agricultores e análises em diversas condições para maximizar o crescimento do vegetal escolhido.

Perfis:

1-	Alface*
2-	Aquecer
3-	Resfriar
4-	Regar
5-	Batata
6-	Cebolinha
7-	Salsinha

* Perfil Atual

Figura 12 - Painel de perfis

O gerenciador de perfis, Fig. 13, é onde os perfis podem ser alterados e criados. Devem ser inseridos os valores na tabela ou utilizar os pré-programados. Para ativar o perfil, uma caixa de seleção *Atual* foi inserida.

Gerenciar Perfis:

Perfil 1 ▾ Atual?

Nome do Perfil:

	Mínimo	Máximo
Umidade do Solo (%):	<input type="text" value="30"/>	<input type="text"/>
Temperatura (°C):	<input type="text" value="14"/>	<input type="text" value="28"/>
Luz necessária (horas):	<input type="text" value="0"/>	<input type="text"/>

Figura 13 – Gerenciador de perfis

As informações enviadas à nuvem também podem ser visualizadas em forma de gráficos, Fig. 14, Fig. 15 e Fig. 16, o que possibilita verificar a condição da estufa em determinado horário.

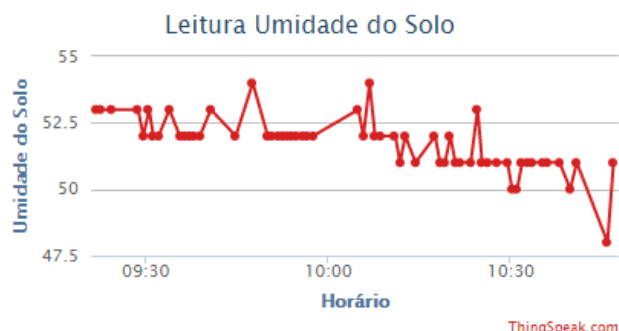


Figura 14 - Gráfico da leitura da umidade do solo

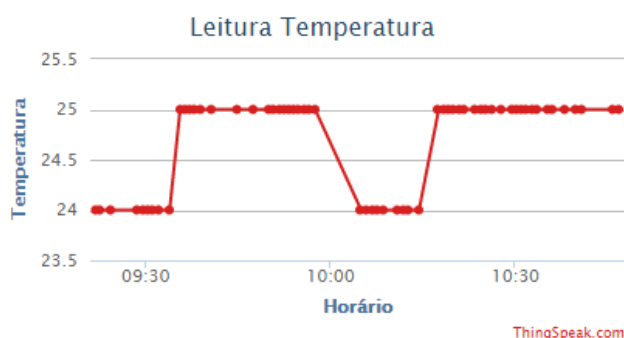


Figura 15 - Gráfico da leitura da temperatura



Figura 16 - Gráfico da leitura da luminosidade

D. Programação

A estufa age comparando as informações pré-configuradas virtualmente, pelo usuário, com as coletadas por meio dos sensores. Ao verificar que uma determinada condição não é satisfeita, o sistema atuará de modo a satisfazer os parâmetros ideais. Como por exemplo, se a temperatura mínima configurada é de 15°C e a temperatura dentro da estufa for de 14°C, o aquecedor será ligado para adequá-la. No caso da umidade do solo, se seu valor for abaixo do configurado, a bomba será ligada por 15 segundos e, então, haverá um tempo de 15 minutos que deve ser atingido até a próxima rega, deste modo

possibilitando a penetração da água no solo. O diagrama a seguir, Fig. 17, ilustra os passos seguidos.

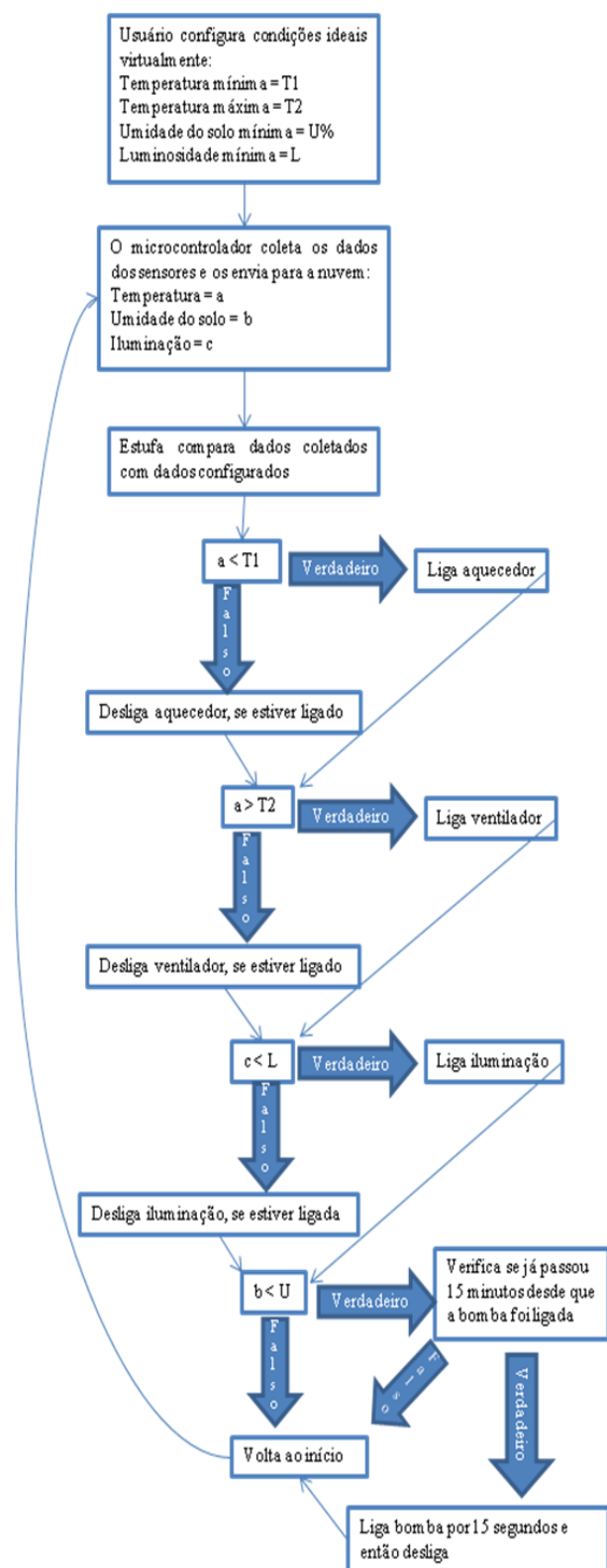


Figura 17 - Diagrama do programa seguido

V. CONCLUSÃO

Iniciativas similares à descrita no artigo já são realidade em países como Japão e Alemanha, onde espaços cultiváveis começam a se tornar escassos e sua adoção é incentivada por órgãos públicos e privados, que desenvolvem equipamentos de baixo custo de aquisição e manutenção. No Brasil, devido à abundância de terras e por fatores como não existência de fomento e financiamento para agroecologia, essa linha de pesquisa se mantém incipiente e restrita aos ambientes acadêmicos.

O protótipo construído prova a funcionalidade da parte eletrônica e avança o conceito de internet das coisas (IoT), o estendendo para uma área que, muitas vezes, não é relacionada a tecnologia virtual. Em curtos períodos de utilização, pode-se verificar o sucesso no cultivo de hortaliças de pequeno porte como hortelã, salsa e cebolinha, que se desenvolveram de forma semelhante aos cultivados manualmente e que possuíam cuidados constantes. Desafios futuros incluem o teste do protótipo por períodos prolongados, analisando a qualidade do cultivo, e operação por indivíduos leigos tanto no ramo de automação quanto no de agricultura, para que seja possível levantar dados sobre o potencial de utilização desses equipamentos na região e no país. Também, sugere-se a criação de um sistema de reconhecimento de imagem que possa auxiliar o usuário nos momentos essenciais como poda e colhimento, o que facilitaria ainda mais a operação, e o desenvolvimento de um protótipo para plantações verticais, aumentando a quantidade de vegetais cultivados por metro quadrado, oferecendo condições para a produção em grande escala.

Agradecimentos

Os autores agradecem os professores Marcelo Roberto Petry, Ebrahim Samer El Youssef, Alex Sandro Rotschildt Pinto, Adão Boava e demais professores da Engenharia de Controle e Automação e o departamento de biologia da FURB, pelos recursos e apoio prestados a este projeto.

Referências

- [1] <http://www.urbangreenbluegrids.com/agriculture>. Acessado em 07 de julho de 2016.
- [2] <http://www.lighting.philips.com/main/products/horticulture/city-farming.html>. Acessado em 07 de julho de 2016.
- [3] http://www.marinostore.com/973-thickbox_default/sensor-de-luminosidade-ldr-5mm.jpg. Acessado em 07 de julho de 2016.
- [4] <http://s3.amazonaws.com/img.iluria.com/product/6B8F7/2C5B05/450xN.jpg>. Acessado em 07 de julho de 2016.
- [5] http://www.alselectro.com/images/soil_moisture1_thumb_6shh6472.jpg. Acessado em 07 de julho de 2016.
- [6] https://www.arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg. Acessado em 07 de julho de 2016.
- [7] <http://s3.amazonaws.com/img.iluria.com/product/1F55AD/4A01E3/450xN.jpg>. Acessado em 07 de julho de 2016.

- [8] <https://www.arduino.cc/en/Guide/Environment>. Acessado em 07 de julho de 2016.
- [9] <https://blog.arduino.cc/wp-content/uploads/2016/03/before.png>. Acessado em 12 de julho de 2016.

Sistema Automatizado de Irrigação por Gotejamento Arduino

Cecili Duwe Fernandes
Jeane Almeida Santos
Letícia Terezinha Farias Schiochet
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina
Blumenau
cecili-fernandes@hotmail.com
jeane.almeidas@gmail.com
leticiafariass94@gmail.com

Resumo—Este trabalho apresenta a implementação de um sistema de controle de irrigação, cujo objetivo é otimizar o consumo de água na irrigação de jardins e assim, estabelecer maior comodidade ao usuário, que programará a frequência de irrigação das plantas. O sistema deve manter a umidade do solo dentro da faixa apropriada, evitando desperdício de água, além de permitir que o usuário acompanhe seu funcionamento através das informações relevantes que será mostrado no painel LCD, essa informação será a faixa de umidade do solo, em percentual. O objetivo foi alcançado através da construção de um sistema de irrigação por gotejamento, capaz de controlar o fluxo de água. Para a criação de tal sistema, foi necessário o estudo e aplicação de sensores de umidade do solo, módulo relé, bomba d'água e fluxo de água. O resultado deste trabalho é um protótipo, utilizando uma placa de Arduino para realizar a leitura dos sensores, acionamento da bomba d'água para ativar o sistema e controlar o fluxo quando necessário.

Palavras-chave—sistema de irrigação; Arduino; automatizado;

I. APRESENTAÇÃO

Considerando o perfil de pessoas que tem ou gostariam de ter plantas em casa, porém não dispõem de tempo suficiente para cuidar delas, este projeto tem por objetivo desenvolver um sistema eficaz no controle da água, visando o conforto desse usuário e o bom desenvolvimento da sua plantação doméstica.

É possível observar que quem cultiva alguma espécie de planta ou horta, por vezes comete o equívoco de despejar uma quantidade elevada de água, fazendo com que a terra encharque. Para evitar que isso ocorra, faz-se necessário a utilização de um sistema de irrigação adequado para o tipo de plantação bem como, vasos que auxiliam no escoamento da água.

Será implementado um sistema utilizando um Arduino Due que irá automatizar a irrigação de pequenas plantas ou até de jardins em residências. O funcionamento será de fácil entendimento podendo ser programado por qualquer pessoa que obtenha o irrigador. Será necessário apenas, digitar

por quanto tempo o irrigador precisará funcionar e então, o usuário terá suas plantas regadas de forma consciente e adequada.

A. Métodos de irrigação

A Irrigação é uma técnica utilizada na agricultura, foi desenvolvida durante o império persa arquemênida e tem por objetivo o fornecimento controlado de água às plantas, em quantidade suficiente e no momento adequado, de modo a garantir a produtividade e a sobrevivência da plantação. É uma maneira de agregar à precipitação natural, e em certos casos, enriquece o solo com a deposição de elementos fertilizantes.

Cada técnica tem um ou mais sistemas associados, a escolha do mais adequado depende de alguns fatores, tais como a topografia (declividade do terreno), o tipo de solo (taxa de infiltração), a cultura (sensibilidade da cultura ao molhamento) e o clima (frequência e quantidade de precipitações, temperatura e efeitos do vento). Além disso, a vazão e o volume total de água disponível durante o ciclo da cultura devem ser analisados. A eficiência de um sistema de irrigação refere-se à percentagem de água absorvida pela planta.

Os sistemas convencionais de irrigação são por gotejamento, aspersão convencional, micro aspersão, pivô central, canhão hidráulico, sulco e sub irrigação, apresentam vantagens se utilizados da maneira mais eficiente.

Neste projeto foi utilizado o sistema por gotejamento pois possui uma eficiência de 90% , é utilizado frequentemente em cultivo permanente e o mais indicado para fruticultura, embora seja usado por produtores de hortaliças e flores, em especial pela economia de água. Em comparação com demais sistemas de irrigação, este possui um alto custo de instalação. Pode ser instalado à superfície ou enterrado, dependendo da vegetação que ser irrigada. Nesse sistema, a água é levada sob pressão por tubos, até ser aplicada ao solo através de emissores diretamente sobre a zona da raiz da planta, em alta frequência e baixa intensidade.

Além de apresentar uma boa eficiência, o gotejamento se destaca na questão do manejo da irrigação, onde tem um menor gasto de água, economizando o recurso que em algumas regiões do país é escasso.

B. Vegetação e Fertilizantes

O constante uso de agrotóxicos na produção de alimentos no Brasil tem sido alvo de preocupações na saúde pública. Uma pesquisa feita pela Agência Nacional de Vigilância Sanitária encontrou elevados níveis de resíduos agrotóxicos em um terço das frutas, vegetais e hortaliças analisadas entre 2011 e 2012. “Quase um terço dos vegetais mais consumidos pelos brasileiros apresentam resíduos de agrotóxicos em níveis inaceitáveis, de acordo com a ANVISA”. [1]

Ingerir alimentos que contenham excesso de agrotóxicos pode causar câncer, problemas neurológicos e malformação fetal. Em 2010, foi usado 1 milhão de toneladas de agrotóxicos em lavouras do país ou seja, 5 kg por brasileiro. O que vem do campo pode não ter apenas nutrientes, mas também resíduos dos produtos usados para proteger as plantações.

A alternativa então é cultivar seus próprios orgânicos, mesmo que o espaço seja pequeno em casa é possível cultivar seus próprios alimentos. Paisagistas ensinam que é possível fazer hortas caseiras. Em vasos de cerâmica é possível cultivar tomates e morangos, por exemplo. É necessário observar as características do ambiente onde a planta ficará. Algumas não toleram vento, como é o caso dos Temperos.

É essencial dispor de horários para cuidar de uma planta, como o período de regá-las e deixá-las expostas ao sol. Plantar uma horta doméstica pode garantir plantas saudáveis, e contribuir fortemente com a sustentabilidade. Devido à compostagem, o solo fica rico em nutrientes e elimina a necessidade de fertilizantes e pesticidas, tornando-o ideal para uma pequena horta. O cultivador pode com isso reciclar seus próprios resíduos

II. IMPLEMENTAÇÃO

Visando estabelecer maior comodidade ao usuário, este projeto tem como objetivo promover economia de água, otimizar o tempo de quem deseja ter uma plantação caseira, proporcionando desta forma, um jardim em excelentes condições, sem que seja necessário a rega em horários desnecessários ou quando o solo estiver úmido.

Os principais itens utilizados são: Placa de Arduino Due, sensor de umidade para solo, bomba d'água, módulo relé, reservatório de água, tela de LCD e teclado.

São apresentadas as respectivas funções de cada item utilizado.

Ao digitar qualquer botão no teclado, o usuário liga o sistema. Através da tela LCD, será solicitado ao usuário que

digite a quantidade de dias em que o sistema deve-se manter em funcionamento.

O Arduino realiza uma leitura contínua do sensor de umidade. Em caso de solo seco, será acionada a bomba d'água que ficará ligada por um curto período de tempo. O gotejador inicia o processo de regar a planta e após detectar solo úmido, a bomba será desligada.

A. Microcontrolador

No início do projeto seria utilizado o Arduino Uno, porém era necessária uma quantidade maior de portas para conectar os jumpers (cabos), então a placa inicial foi substituída por uma maior, a Arduino Due.

A placa Arduino DUE possui um microcontrolador, com encapsulamento de 144 pinos LQFP e com core ARM Cortex-M3 como cérebro da placa, 54 pinos digitais de entrada ou saída, 12 entradas de conversores analógico-digital, com 12 bits de resolução.

Possui duas portas USBs, uma para carregar o código e a outra para conectar hardwares externos, como um mouse, teclado ou smartphone, 4 portas seriais.

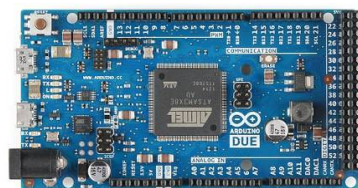


Fig. 1. Arduino Due

B. Sensor de Umidade

Este sensor é ideal para fazer a leitura da umidade do solo onde estiver instalado e informar ao microcontrolador. Quando o solo está seco a saída do sensor fica em estado alto / nível lógico 1, e quando o solo está úmido a saída do sensor fica em estado baixo / nível lógico 0. O sistema também é capaz de medir valores intermediárias.

Este sensor utiliza duas pontas de prova para passar a corrente pelo solo e sua leitura é baseada na resistência elétrica resultante. Quanto mais água no solo, mais baixa é a resistência do solo e mais fácil fica a condução entre as pontas de prova. Quando o solo está seco, a condutividade é baixa, logo a resistência é alta.

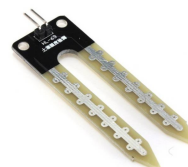


Fig. 2. Sensor de umidade

C. Bomba de água

A água é forçada a passar por uma câmara espiral onde aumenta a sua velocidade e pressão para que possa ser bombeada. Este tipo de bomba pode ser acionada por um motor elétrico, um moinho de vento ou qualquer coisa que possa gerar movimentos circulares.

Controlado pelo Arduino, libera quantidade específica de água, suficiente para umidificar o solo.



Fig. 3. Bomba de Água

D. Módulo Relé

Este módulo pode ser usado em diversos projetos afim de acionar cargas de até 250VAC a 7A ou 125VAC a 10A. Pode ser usado para controlar luzes, equipamentos ligados a rede elétrica, motores, etc.

Ele pode ser ligado diretamente a uma saída digital do Arduino, visto que possui saída padrão de 03 vias, lembrando que, como existe um relé com uma bobina de 5V no módulo, você deve ligar o pino VCC ao 5V regulado da placa Arduino.



Fig. 4. Módulo relé

E. Teclado

Este Teclado Matricial é um componente do Arduino utilizado para entrada de dados. Ele possui 16 teclas dispostas em 4 linhas e 4 colunas, e um conector de 8 pinos para ligação. Através desse teclado o cliente pode programar a quantidade de dias desejados para que o irrigador funcione automaticamente.



Fig. 5. Teclado

F. Display

Tela de LCD que apresenta as informações ao cliente. Esse display LCD tem 16 colunas e 2 linhas, com backlight (luz de fundo) azul e letras na cor branca. Para conexão, são 16 pinos, dos quais usamos 12 para uma conexão básica, já incluindo as conexões de alimentação, backlight e contraste. Para o ajuste do contraste, usamos um potenciômetro.



Fig. 6. LCD

G. Reservatório

Recipiente onde será armazenada a bomba de água e no máximo três litros de água para o abastecimento do sistema de irrigação.



Fig. 7. Reservatório

H. Gotejador

Localizado na ponta da mangueira, regula o fluxo da água que é liberado no solo para regar a plantação de acordo com os padrões pré estabelecidos.

Nesse projeto foi utilizado uma agulha de seringa como gotejador, a água sai em gotas devido a pressão da água exercida na mangueira e na agulha.



Fig. 8. Gotejador

I. Mangueira

Irá transportar a água da bomba para o gotejador. Trata-se de uma mangueira cristal, transparente normalmente utilizada em residências. E por ser muito maleável tem a possibilidade de ser colocada de diferentes formas.

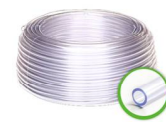


Fig. 9. Mangueira

III. LÓGICA

A. Lógica de funcionamento

O projeto de irrigação automatizado possui um sistema muito simples que pode ser manuseado por qualquer pessoa e pode ser utilizado em diferentes tipos de vasos.

Inicialmente o usuário deverá adicionar água até o limite do reservatório e colocar a haste com o gotejador dentro do vaso que deseja irrigar.

O LCD apresentará uma mensagem para o usuário de boas vindas e após isso irá pedir para que adicione a quantidade de dias que desejará que o sistema funcione. O usuário fará essa seleção pelo teclado.

Após ser feita a escolha o sistema irá verificar se deve ou não ligar o sensor de umidade, ou seja, se o número de dias for maior que um, o sensor será ligado, caso contrário, o sensor não irá ligar. Com o sensor ligado será verificado se a umidade do solo está baixa ou alta. Se a umidade estiver baixa então a bomba de água é acionada e irá levar a água até o destino desejado, ou seja, até a planta. Já se a umidade estiver alta a bomba permanecerá desligada até que o sensor detecte o contrário.

B. Fluxograma

O algoritmo a seguir apresenta a lógica de funcionamento do sistema de irrigação. Através destas instruções é possível esclarecer ao computador o que e como processar os dados relacionados ao problema computacional que partirá de uma situação inicial para outra desejada pelo seu usuário, denominada situação final.

Estas instruções programáveis possuem suas próprias lógicas de funcionamento e definem uma sequência de ações a serem executadas pelo computador.

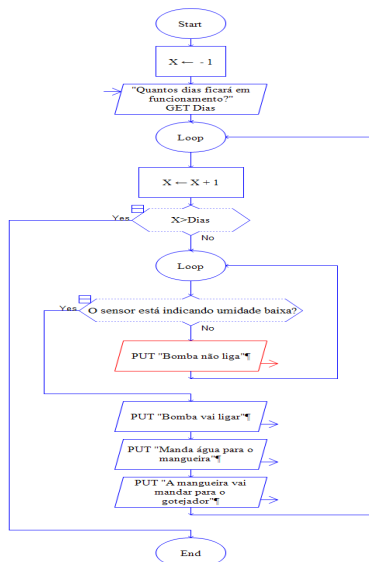


Fig. 10. Fluxograma

C. Programação

Para o desenvolvimento da programação do projeto, fez-se uso da linguagem C, no programa Arduino conforme exemplifica a figura abaixo:

```

sketch_jul07a $
Serial.println("2");
} else if((ti-2) == 2){
Serial.println("5");
} else if((ti-2) == 3){
Serial.println("8");
} else{
Serial.println("0");
}
while(digitalRead(8) == HIGH){
}

if (digitalRead(9) == HIGH){
if((ti-2) == 1){
Serial.println("3");
} else if((ti-2) == 2){
Serial.println("6");
} else if((ti-2) == 3){
Serial.println("9");
} else{
Serial.println("#");
}
while(digitalRead(9) == HIGH){};
}

if (digitalRead(10) == HIGH){
if((ti-2) == 1){
Serial.println("A");
}
}
  
```

Fig. 11. Programação do Arduino

É importante destacar que para cada componente que constitui o projeto faz-se necessário uma programação inicial a fim de teste, é o caso da bomba de água, teclado, display e sensor de umidade. Partindo desse princípio é possível unir as programações e com alguns ajustes chegar ao resultado final, que é a programação completa.

IV. TABELA DE CUSTOS

Originalmente pensava-se que a conservação da água era o principal benefício do sistema. A experiência demonstrou que a irrigação por gotejamento bem planejada e bem administrada oferece outras vantagens agrônômicas e econômicas igualmente importantes.

O custo total do sistema de irrigação é relativamente baixo. A partir da seguinte tabela de custos e pelas informações anteriores quanto a composição, temos os custos de desenvolvimento do mesmo.

TABELA I. TABELA DE CUSTOS

PEÇA	CUSTO (unidade)	QUANTIDADE	TOTAL (unidades monetárias)
Reservatório de água 3L	R\$ 13,00	01	R\$ 13,00 Reais
Bomba de água	R\$ 50,00	01	R\$ 50,00 Reais
Módulo Relé	R\$ 10,00	01	R\$ 10,00 Reais
Sensor de umidade	R\$ 9,90	01	R\$ 9,90 Reais
LCD com teclado	R\$ 34,90	01	R\$ 34,90 Reais
Mangueira 1 metro	R\$ 1,70	01	R\$ 1,70 Reais
Haste de madeira	R\$ 0,30	02	R\$ 0,60 Reais
Gotejador	R\$ 1,00	01	R\$ 1,00 Reais
TOTAL DE CUSTOS			R\$ 121,10 Reais

V. RESULTADOS

A ideia desde o princípio foi fazer um sistema de irrigação portátil que fosse de fácil utilização para qualquer pessoa e que facilitasse o dia a dia do usuário.

Foram muitas as ideias do formato e de como ele iria funcionar, pois existiam várias possibilidades. Entretanto foram escolhidos os materiais mais econômicos para que o produto final não fosse encarecido.

A placa inicial foi trocada por uma maior devido à necessidade de mais portas e a ideia de fazer um suporte com canos e mais gotejadores foi suspensa devido os problemas que iriam gerar.

Apesar de várias indecisões ao longo do projeto foi conseguido realizar com êxito.

Abaixo segue uma imagem de como ficou o projeto:



Fig. 12. Sistema de Irrigação finalizado

VI. CONCLUSÃO

Existem várias barreiras no desenvolvimento de um projeto que podem dificultar a realização de uma ideia, devido a esse motivo o projeto nem sempre irá terminar exatamente como foi imaginado no começo.

Entretanto, como já foi citado anteriormente, o projeto do Sistema de Irrigação foi bem sucedido, tendo como um diferencial um sistema automatizado, de fácil manuseio e que irá gerar uma grande comodidade ao usuário. Além de ser econômico manterá as plantas saudáveis e muito bonitas durante um bom tempo.

Referências

- [1] https://www.robocore.net/modules.php?name=GR_LojaVirtual&prod=258
- [2] <http://www.filipeflop.com/pd-aa99a-sensor-de-umidade-do-solo-higrometro.html>
- [3] <http://blog.filipeflop.com/display/controlando-um-lcd-16x2-com-arduino.html>
- [4] <http://blog.filipeflop.com/arduino/teclado-matricial-4x4-arduino.html>

PIANO ELETRÔNICO COM CI 555

Gabriel Alves Silvestre
Departamento de Engenharia
Universidade Federal de Santa Catarina
Blumenau, Brasil
g.alves@grad.ufsc.br

Ronaldo William de Oliveira Baggio
Departamento de Engenharia
Universidade Federal de Santa Catarina
Blumenau, Brasil
rwobaggio@hotmail.com

Resumo— O piano é um instrumento clássico com um som marcante e de características notórias, porém de grande complexidade na construção e alto custo. Com a evolução da eletrônica, tornou-se possível produzir pianos eletrônicos (ou digitais) que tem construção simples e baixo custo. Apesar de seu tamanho reduzido é um instrumento muito interessante e tem grande aplicação. Sua construção é feita com um circuito oscilador simples construído através do C.I 555. A ideia essencial do projeto é gerar uma saída com frequência que varia de tal modo que o som emitido pelo autofalante seja semelhante ao som das notas musicais. A frequência varia de acordo com o valor das resistências que são parte do circuito oscilador. Nesse artigo apresentaremos o projeto de um mini piano digital, que gera sons semelhantes aos obtidos em um piano clássico. Será descrita a montagem, bem como o desenvolvimento que relaciona a teoria e a prática de funcionamento desse circuito.

Palavras-chave— CI555, Áudio Mixer, Piano Eletrônico

I. INTRODUÇÃO

Raro encontrar uma pessoa que nunca tocou um piano, ou que nunca tenha ouvido o seu som. O piano é um instrumento musical fabuloso, teve origem no século 18 quando idealizado pelo seu inventor o italiano Bartolomeu Cristofori. Classicamente é definido como um instrumento de cordas, porém com a evolução da eletrônica novas versões como o piano digital vem ganhando espaço no mercado musical. Os pianos digitais são basicamente dispositivos eletrônicos que usam circuitos integrados e alto-falantes para reproduzir (simular) os sons do piano[1].

Com grande facilidade e pouco orçamento é possível construir replicas dos pianos digitais. Um dos circuitos que permitem fazer isto é o NE 555 ou circuito integrado 555, popularmente conhecido como temporizador 555. Este temporizador deve sua popularidade a simplicidade, empregada na sua construção, aliada ao grande número de aplicações, desde temporizadores a osciladores, em que é empregado[2].

Através da variação no valor da resistência a frequência será alterada, e conseqüentemente, o som transmitido para o alto-falante irá mudar. Ajustando os parâmetros do circuito para obter as frequências correspondentes as notas musicais, os sons ouvidos no alto-falante aproximam-se aos sons de aparelhos profissionais.

Nesse projeto será desenvolvido um piano eletrônico, utilizando temporizador 555, alguns resistores e um capacitor.

II. CI 555

A. Apresentação

O circuito integrado 555 é um dos circuitos de maior sucesso comercial da história da eletrônica. Foi desenvolvido no início dos anos 70 pelo engenheiro Hans R. Camenzind. O seu grande sucesso deve-se a simplicidade e baixo custo, pode ser utilizado com temporizador, oscilador e gerador de pulso.

B. Componentes

Basicamente é composto por diodos, transistores e amplificadores dispostos em um chip de silício com oito terminais. Cada um destes terminais tem uma função específica no funcionamento. A seguir, a Fig 1. traz a pinagem do 555, a a tabela I o nome e a função de cada pino:

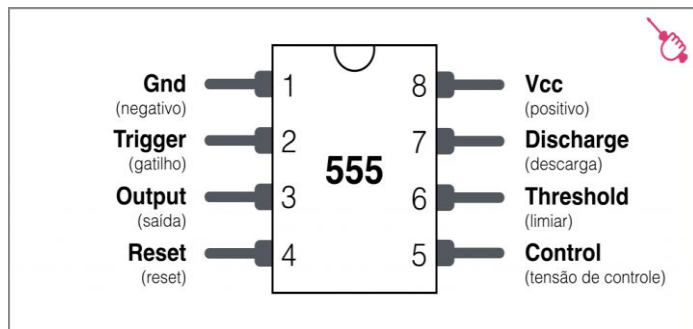


Fig. 1. Pinos do CI 555[2].

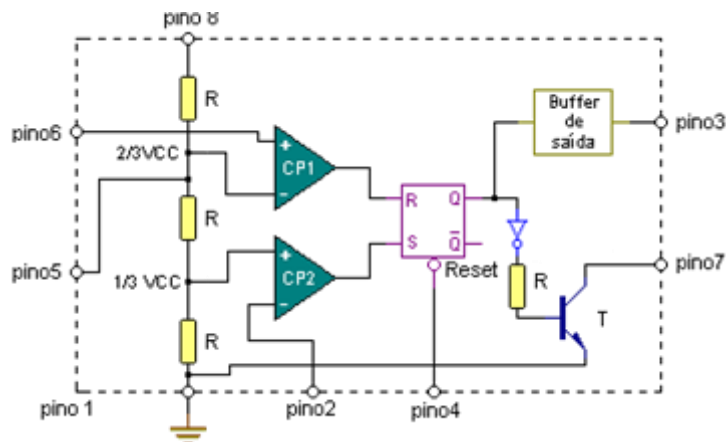


Fig. 2. Esquema interno do CI 555[2].

Tabela I – Descrição dos Pinos do CI 555. [1][2]

Número	Nome	Função
1	Terra	Conectado ao terra da fonte (alimentação negativa da fonte).
2	Disparo	Este pino ativa o biestável interno e a saída (OUTPUT) quando estiver com uma tensão abaixo de 1/3 da tensão VCC. Uma vez disparado, colocando-se o pino 4 (reset) aterrado ou o nível de tensão do pino 6 acima de 2/3 de VCC, ocorre o desarme do CI, comutando o pino 3 novamente ao nível de zero volt.
3	Saída	Com o CI em estado desarmado se encontra em zero volt, passando a VCC quando é disparado. O intervalo de tempo (tempo em que fica na tensão Vcc) é definido por alguns componentes externos como resistência e capacitância.
4	Reset	Uso normal em VCC. Quando ligado a terra provoca o desarme do CI, passando a saída de VCC para terra. Interrompe um ciclo de temporização quando conectado ao terra.
5	Controle	Quando polarizado externamente provoca uma variação no ponto de desarme, alterando o período de desligamento do circuito. Geralmente é usado para modular em frequência o circuito quando em funcionamento como oscilador.
6	Limiar	Desativa o biestável interno e a saída (OUTPUT) quando estiver com uma tensão acima de 2/3 da tensão VCC.
7	Descarga	É usado para descarregar o capacitor conectado a este terminal. O capacitor é um dos componentes externos ligado internamente ao coletor do transistor T, este entra em saturação quando o CI é resetado, é capaz de descarregar a tensão do capacitor que se encontra na rede de temporização do circuito.
8	Vcc	Pino de alimentação (alimentação positiva da fonte).

É composto basicamente por dois comparadores, um Flip-Flop SR (Set-Reset), um estágio de saída e uma rede resistiva com três resistores iguais de 5 kΩ (desta característica que surge o nome de 555)[3]. Dependendo do circuito montado e da tensão aplicada, o 555 varia a região de operação em dois modos de operação: monoestável e o astável.

C. Modo Monoestável

Neste modo o CI 555 funciona como um temporizador, ao receber um sinal adequado no pino 2 (disparo) é gerado um pulso de duração específica na saída. Após este tempo, a saída volta ao estado 0 até receber outro sinal. Em outras palavras, ao receber um sinal a saída do 555 fica ligada por um período de tempo determinado, após isto é desligada[3][4].

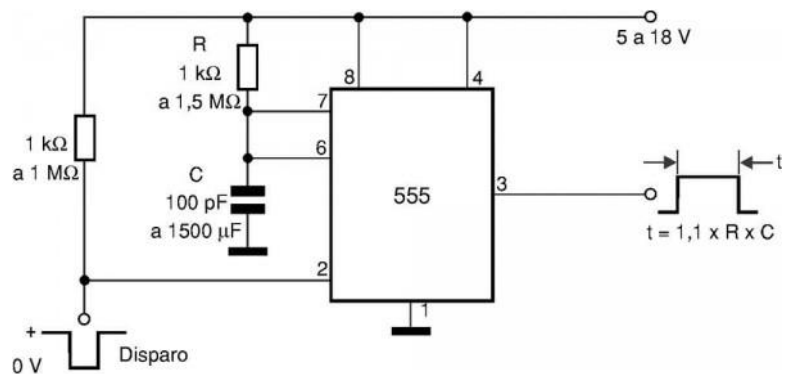


Fig. 3. Esquema do CI 555 no modo monoestável[2].

Nesta configuração, os pinos 6 e 7 que correspondem ao sensor de nível e terminal de descarga do capacitor são interligados e ligados a uma rede RC, ou seja, um capacitor e um resistor externo que vão determinar o tempo de acionamento do circuito[4]

Uma tensão menor que 1/3 VCC faz com que a saída do comparador CP2 (Fig.2) seja nível lógico 1, assim seguindo a tabela verdade do Flip-Flop SR a saída será nível alto, produzindo uma tensão de saída VCC. Esta tensão VCC faz com o transistor entre na região de corte. O capacitor C (Fig.3) é carregado pela fonte VCC através do resistor R (Fig.3).

A saída (pino 3) permanece ativa enquanto o capacitor carrega, ao chegar a uma tensão de 2/3 VCC o comparador CP1 (Fig.2) tem sua saída em nível lógico alto o que faz com que a saída do Flip-Flop RS tenha nível lógico baixo, desta forma o transistor passa a região de saturação, e por consequência descarrega o capacitor. Todo este processo é repetido sempre que há um novo disparo.

O tempo em que a saída terá um nível alto é dado por:

$$t = 1,1 \cdot R \cdot C^1 \quad (1)$$

Onde R é a resistência e C a capacitância associadas externamente ao circuito. Está é uma das grandes vantagens do 555, podemos determinar o tempo mudando características externas do circuitos.

¹ A dedução das fórmulas pode ser entrada em [3].

D. Modo Monoastável

Neste modo de operação o pino de disparo (pino 2) está conectado ao pino limiar (pino 6), e o capacitor é carregado e descarregado de acordo com a saída do Flip-Flop RS.

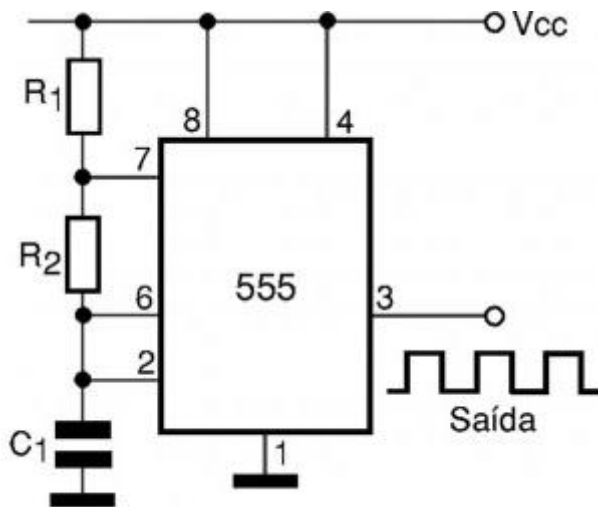


Fig.4 - Esquema do CI 555 no modo monoastável[2].

Considerando que inicialmente o capacitor está descarregado e a saída do Flip-Flop está em 0V. Nesta configuração a saída de CP1 está em 0 (pois a tensão em 6 é menor que $2/3 V_{CC}$) e a saída de CP2 está em 1 (pois a tensão em 2 é menor que $1/3$) desta forma (com S do FF em 1 e R do FF em 0) a saída do Flip-Flop terá nível 1. Com a saída do Flip-Flop em 1, o transistor entra em corte e o capacitor C2 começa a carregar-se.

O capacitor C carrega-se, tendendo ao valor de VCC, através dos resistores externos R1 e R2 até aproximar-se de $2/3$ da tensão de alimentação (VCC) quando acontece a mudança de estado do circuito. Quando a tensão chega em $2/3 V_{CC}$ no pino 6 ocorre a mudança no comparador CP1, que passa do nível 0 para nível 1, e isto faz com a entrada R do FF mude para nível lógico 1. Com R=1, o FF tem sua saída passada para nível lógico 0, como a saída está em nível baixo, o capacitor se descarrega até $1/3$ da tensão da alimentação, esta descarga ocorre através do resistor R2 e inicia-se um novo ciclo.

O tempo em que a saída permanece ativa (nível lógico alto) é sempre maior que o tempo em que a saída permanece inativa. Isto ocorre porque o circuito é carregado por R1+R2 e é descarregado apenas por R2, então o sinal em nível alto sempre terá duração maior que o nível baixo.

O tempo em que a saída permanece no nível alto, baixo e o tempo total podem ser calculados da seguinte forma:

$$T_{\text{alto}} = 0.7 * (R_1 + R_2)^2 \quad (2)$$

$$T_{\text{baixo}} = 0.7 * (R_2)^3 \quad (3)$$

$$T_{\text{total}} = T_{\text{alto}} + T_{\text{baixo}} \quad (4)$$

² A dedução das fórmulas pode ser entrada em [3].

³ A dedução das fórmulas pode ser entrada em [3].

Como o tempo total é conhecido, torna-se possível calcular a frequência já que:

$$f = 1/T \quad (5)$$

$$f = 1/T_{\text{alto}} + T_{\text{baixo}} \quad (6)$$

$$f = 1/0.7 * (R_1 + 2 * R_2) * C \quad (7)$$

III. CÁLCULO DOS PARÂMETROS DO CIRCUITO

Para obter as diferentes notas musicais é necessário alterar a frequência de saída do sinal (som). Para tanto, é necessário variar o capacitor, ou R1 ou R2, para facilitar, vamos manter constante o valor de R1 e do capacitor e variar o valor de R2.

O valor de R2 será alterado conforme a frequência de cada nota musical (a frequência correspondente a cada nota pode ser vista na tabela II) é possível encontrar o valor de R2 através da fórmula 7. Os valores escolhidos para R1 e C foram respectivamente $1k\Omega$ e $1\mu F$. Desta forma:

Isolando R2 em 7:

$$R_2 = 1/2 * [(1/(0.7 * C * f)) - R_1] \quad (8)$$

Com $R_1 = 1k\Omega$ e $C_1 = 1\mu F$:

$$R_2 = 1/2 * [(1/(7 * 10^{-7} * f)) - 1000] \quad (9)$$

Variando f conforme a tabela II obtêm-se os valores de R2 segundo a tabela III:

TABELA II – FREQUÊNCIAS DAS NOTAS MUSICAIS[6].

Nota	Frequência (Hz)
Dó	261.63
Ré	293.66
Mi	329.63
Fá	349.23
Sol	391.99
Lá	440.00
Si	493.88

TABELA III – VALORES DE R2 PARA CADA NOTA MUSICAL.

Nota	R2(Ω)	Resistências no circuito(Ω)
Dó	946	820+100
Ré	1123	820+100+220
Mi	1331	820+100+220+200
Fá	1546	820+100+220+200+220
Sol	1671	820+100+220+200+220+120
Lá	1932	820+100+220+200+220+120+270
Si	2236	820+100+220+200+220+120+270+300

Para alternar entre um resistência e outra, será utilizado um push button em série com cada um dos valores de R2. Além disso, os valores de R2 da tabela serão aproximados

para os valores dos resistores de uso comercial para facilitar a montagem.

IV. SIMULAÇÃO – LTSPICE IV

Antes de iniciar a montagem do circuito é interessante fazer a simulação via software para ter valores como base e compara-los com os valores experimentais encontrados. O software utilizado será o LTspice IV, a seguir na figura 5 é possível ver o circuito simulado:

Nota	$V_{OUT}(V)^4$	$I_{RB}(mA)^5$	Frequência (Hz)
Dó	5.15	2.01	261.78
Ré	5.15	1.81	293.25
Mi	5.15	1.65	327.87
Fá	5.15	1.51	347.22
Sol	5.15	1.44	390.63
Lá	5.15	1.31	436.68
Si	5.15	1.19	507.10

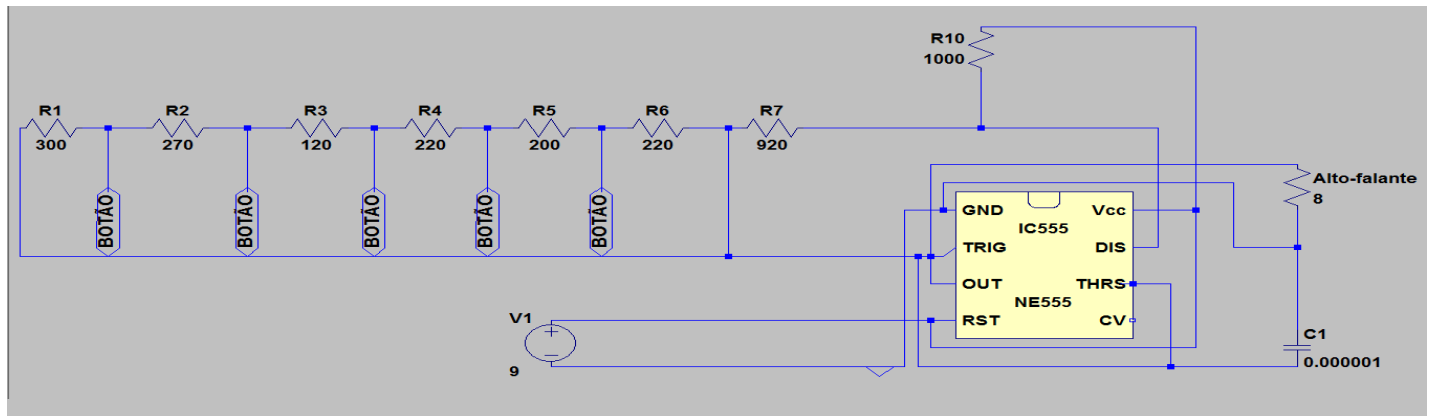


Fig.5 – Circuito Simulado no LTspiceIV.

A. Elementos do Circuito

Na figura 5 temos os seguintes componentes do circuito:

- Fonte de tensão - será utilizada na alimentação, pode ser uma bateria de 9V ou uma fonte de tensão DC regulável.
- Botão – representa cada push button instalado no circuito, de acordo com o botão escolhido será 'acionada' uma resistência diferente. A variação na escolha dos botões irá variar a frequência e conseqüentemente o som na saída (alto-falante). Cada botão corresponde a uma nota musical.
- Circuito Integrado 555 – será utilizado no modo monoastável, é ele quem controla a saída do sistema.
- Resistor R1 a R7 – de acordo com o botão acionado será selecionada uma soma das resistências, cada botão corresponde a soma de todas as resistências anteriores. Esta soma ocorre porque os resistores encontram-se em série. Estes resistores serão chamados de resistores de nota.
- Capacitor C1 - este capacitor será carregado e descarregado, conforme explicado anteriormente. Um dos fatores que afetam a constante de tempo do sistema é o valor de C1, neste caso o valor escolhido para a capacitância é $1\mu F$.
- Resistor R10 – Este resistor terá valor constante de $1k\Omega$, ele em conjunto com o capacitor C1 e com a soma de resistências selecionada determinará a constante de tempo do sistema.

B. Valores encontrados na simulação

TABELA IV – VALORES ENCONTRADOS NO LTSPICEIV

V. CONSTRUÇÃO DO CIRCUITO

A seguir estão listados os materiais necessários para construção do circuito:

- 1 Resistência 820Ω 1/4 Watt.
- 1 Resistência 100Ω 1/4 Watt.
- 2 Resistências 220Ω 1/4 Watt.
- 1 Resistência 200Ω 1/4 Watt.
- 1 Resistência 120Ω 1/4 Watt.
- 1 Resistência 270Ω 1/4 Watt.
- 1 Resistência 300Ω 1/4 Watt.
- 1 Capacitor Electrolítico $1\mu F$.
- 1 NE555.
- 1 Alto-falante de 8Ω 2W.
- 9 Push-buttons.
- Fonte de alimentação (bateria de 9V ou fonte de tensão DC).

A seguir seguem as instruções para montar o circuito:

1º passo:

Conectar o chip 555 na protoboard.

2º passo:

Conectar a resistência de 1000Ω , no pino 7 do chip 555 e na alimentação da bateria.

3º passo:

⁴ V_{OUT} – tensão no alto-falante.

⁵ I_{RB} – corrente nos resistores de nota selecionados.

Conectar as resistências restantes em série, na seguinte ordem: (820 Ω +100 Ω); 220 Ω ; 200 Ω ; 220 Ω ; 120 Ω ; 270 Ω ; 300 Ω .

4º passo:

Conectar os botões entre as resistências, para formar as notas musicais.

5º passo:

Conectar o capacitor nos pinos 1 e 2 do chip 555, sendo que o positivo deve estar no pino 2 e o negativo no pino 1.

6º passo:

Fazer a conexão através de um fio do pino 2 ao pino 6 do chip 555.

7º passo:

Conectar o pino 4 do chip 555 à alimentação da bateria.

8º passo:

Conectar o fio positivo do mixer no pino 3 do chip 555, e o fio negativo no terra do circuito.

Alguns componentes são polarizados, e se estiverem conectados de forma errada eles não funcionaram, deve-se ter cuidado para que eles não sejam danificados. Os componentes que devem ter uma atenção especial são: a conexão da bateria, e o temporizador 555, notando que o ponto de referência para a tensão deve ser comum a todos os elementos do circuito. A figura 5 ilustra a montagem final do circuito:

VII. TRATAMENTO DE DADOS E DISCUSSÃO

Os valores obtidos para a frequência no simulador LTspice IV foram próximos aos valores calculados teoricamente através da fórmula 7, porém as frequências medidas experimentalmente através do osciloscópio tiveram um desvio acima do esperado, ou seja, o som emitido pelo alto-falante não correspondia exatamente a nota musical.

Esta diferença encontrada entre a frequência teórica e a frequência real tem relação com a aproximação dos valores das resistências, já que os valores calculados das resistências não foram encontrados, e com as não idealidades como erros de medida e perdas no próprio CI555.

VIII. CONCLUSÃO

Com os avanços da eletrônica tornou-se possível substituir instrumentos caros e complexos por circuitos simples e de baixo custo.

O circuito integrado 555 é exemplo de um dispositivo simples e barato com grande gama de aplicações seja em osciladores, seja em temporizadores. Com o 555, um capacitor e algumas resistências pode-se produzir sons com qualidade próxima a instrumentos clássicos. A finalidade do CI555 no circuito é variar a frequência de saída conforme os valores das frequências das notas musicais. Tal variação é obtida ao alterar o valor da resistência associada ao circuito.

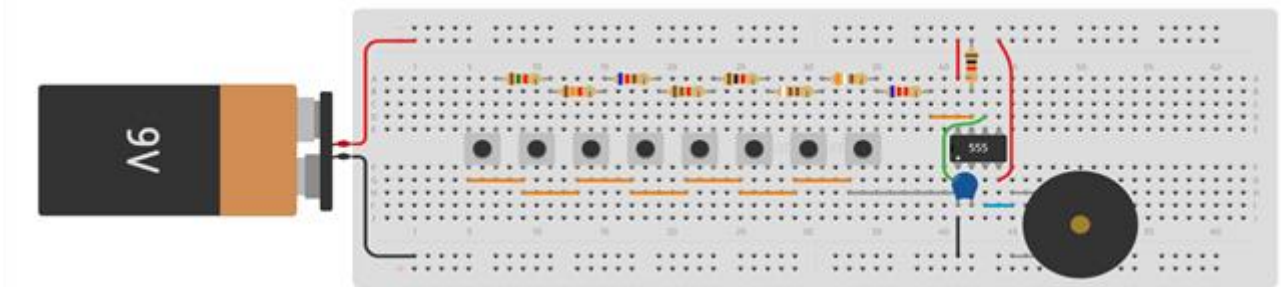


Fig.7 – Circuito final na protoboard[7].

VI. VALORES EXPERIMENTAIS

Após o embasamento teórico, a simulação e a construção do circuito, é possível ligar a fonte de alimentação e medir os valores reais. Tais valores estão presentes na tabela V.

TABELA IV – VALORES EXPERIMENTAIS

Nota	$I_{RB}(mA)$	$V_{OUT}(V)$
Dó	1.25	4.91
Ré	1.17	4.90
Mi	1.04	4.90
Fá	0.98	4.87
Sol	0.65	4.86
Lá	0.51	4.75
Si	0.44	4.74

Apesar do projeto ter apresentado características semelhantes ao de um piano clássico (em relação a frequência das notas) dificuldades como adaptação dos valores de resistência, alto-falante de baixa qualidade e problemas com aquecimento do 555 afetaram o resultado final, neste caso o som ouvido. Possíveis melhorias como implantação de resistências variáveis (potenciômetros) e alto-falante de maior qualidade trariam ganho considerável ao projeto.

O experimento no aspecto geral teve sucesso, já que os sons foram diferenciados no alto-falante quando a resistência era variada e foi possível assimilar a teoria, aprendida em sala de aula com a prática no laboratório.

REFERÊNCIAS

- [1] História de Tudo(2016, 4 Julho). História do Piano. Disponível em:<http://www.historiadetudo.com/piano>

[2]Fabrício Campos (2016, 4 de Julho). Universidade Federal de Juiz de Fora. Aplicações do CI555. Disponível em: http://www.ufjf.br/fabricio_campos/files/2011/08/pratica_2_aplicacoes_555.pdf

[3] Instituto Newton C. Braga LTDA (2016,4 Julho). O circuito Integrado 555. Disponível em: <http://www.newtonbraga.com.br/index.php/como-funciona/592-o-circuito-integrado-555-art011>

[4]Charles Borges de Lima (2016, 4 de Julho). Instituto Federal de Santa Catarina. Tutorial técnico: o temporizador 555. Disponível em: http://www.sr.ifes.edu.br/~secchin/Eletronicaanalogica/temp_555.pdf

[5] Saber Eletrônica (2016, 4 Julho). Conheça o circuito integrado 555. Disponível em: <http://www.sabereletronica.com.br/artigos/2446-conheca-o-circuito-integrado-555>

[6] Bruno Agora (2016, 4 Julho). Notas Musicais. Disponível em: <http://www.brunoagora.com/2012/03/notas-musicais-escala-cromatica-acordes.html>

[7]Auto Desk Circuits(2016, 4 Junho). Simple 555 Piano. Disponível em: <https://circuits.io/circuits/1098266-simple-555-piano>

Utilização de Circuitos Eletrônicos para Reproduzir Distorções de Som

Brunno Vanelli

Universidade Federal de Santa Catarina

Email: brunno.v@grad.ufsc.br

Guilherme Moser Manerichi

Universidade Federal de Santa Catarina

Email: guilherme.manerichi@grad.ufsc.br

Resumo—Por muito tempo os equipamentos de produção musical para guitarras e outros instrumentos como amplificadores e equipamentos de distorção foram implementados utilizando válvulas. No entanto, com o advento da microeletrônica, esses dispositivos foram lentamente sendo substituídos por dispositivos de estado sólido, que possuem mais fácil implementação e são mais baratos. Esse artigo tem foco na reprodução de circuitos capazes de aplicar distorções sonoras em sinais de áudio.

Keywords—Clipping, Overdrive, Distorção

I. INTRODUÇÃO

Uma das características mais marcantes do *rock 'n roll* foi a presença de distorção nas músicas. Vários artistas como Jimmy Hendrix, Chuck Berry, Link Wray, Slash, Angus Young utilizaram largamente de distorção em suas músicas. Existem dois tipos bastante utilizados de distorção: *Distortion* (ou distorção) e *Overdrive*.

O *Clipping*, como é popularmente chamado, consiste em distorcer o sinal de forma a limitar a altura máxima da onda sonora a partir do momento que ela passa de um limite máximo definido pelo circuito. Dessa forma, a onda de som senoidal fica achatada nas partes superiores e inferiores. Esse processo faz com que a música dos instrumentos tenha um som diferente da original. Exemplos de *Clipping* podem ser observados na Figura 2.

Overdrive consiste em utilizar *clipping* em intensidades diferentes para obter a distorção de som desejada. Existem *soft clipping* e *hard clipping*.

Soft Clipping consiste em arredondar o topo das ondas senoidais, de forma a minimizar a curva senoidal.

Hard Clipping consiste propriamente em cortar os picos e vales, fazendo a onda lembrar levemente uma onda quadrada. As distorções, mesmo que alterem minimamente a forma de onda, tem um efeito audível no som de saída.

Por outro lado, *distortion* consiste em alterar completamente o formato de uma onda, de forma a obter uma deformação muito maior do que o *overdrive*. Esse processo é bastante conhecido no meio musical por dar ao som um caráter muito "sujo" à música.

A implementação dos circuitos capazes de distorcer sinais de áudio pode ser feita utilizando AMPOPs. AMPOPs, ou amplificadores operacionais, são dispositivos construídos utilizando transistores que possuem muitos usos, entres eles como somador e amplificador. Um circuito com AMPOP na

configuração de amplificador inversor pode ser visto na Figura 1 [1].

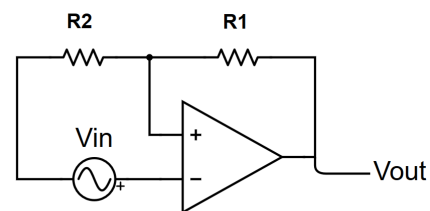


Figura 1. Circuito Amplificador AMPOP

Uma das características mais dos AMPOPs é a amplificação dos sinais de entrada. Variando-se razão das resistências $(R_1 + R_2)/R_2$ do circuito da Figura 1 pode-se alterar o valor do ganho do valor de saída V_{out} em relação à entrada V_{in} . O valor real do ganho varia de acordo com a alimentação do AMPOP. Se o ganho do sinal for maior do que a alimentação do AMPOP, o sinal será cortado.

No entanto, para alterar o formato da onda usa-se de uma propriedade dos diodos. Os diodos apresentam uma tensão de corte que varia conforme seu material. Quando o valor da tensão da onda ultrapassa o valor da camada de depleção, o diodo conduzirá. Com isso, pode-se utilizar diodos para cortar a onda em certos níveis de tensão.

Finalmente, para obter um bom resultado sonoro, deve-se utilizar filtros. Utiliza-se filtros passa alta e passa baixa para retirar as componentes que não estão na faixa audível que se quer amplificar, como o ruído.

II. PROJETO

Normalmente, os circuitos que aplicam distorção de som incluem vários pedais, que tem potenciômetros para controle de volume e quantidade de distorção aplicada. Os pedais são geralmente conectados entre o instrumento e a caixa de som amplificadora, todos em série. Os pedais podem ser ativados ou desativados com um toque no pedal. Os vários pedais em série podem aplicar combinações de distorção de forma a obter o som desejado.

O circuito desenvolvido deve oferecer em um mesmo pedal tanto *Hard Clipping* como *Soft Clipping* podendo obter dois sons distorcidos.

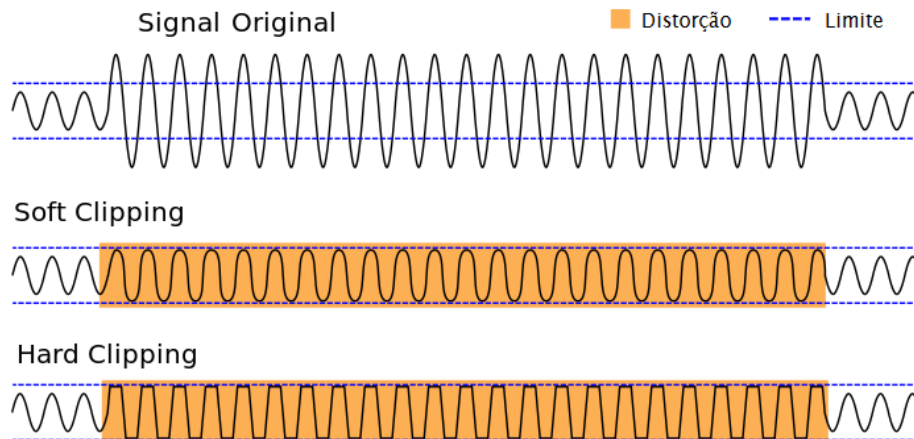


Figura 2. Forma de Onda após Clipping [2]

O projeto pode ser dividido em três partes principais: o amplificador, que tem a função de amplificar o sinal do instrumento musical que está sendo tocado, o circuito de alimentação, que vai garantir que o sinal de entrada seja compatível com o ganho dado, e o circuito de distorção em si, que vai realizar o *clipping* de acordo com a seleção do usuário. Por último, deve-se utilizar filtros em vários estágios do circuito para minimizar os ruídos.

A. Amplificador

Utilizar-se-á do circuito amplificador da Figura 1 com adição de um filtro passa alta na entrada (C_1) para eliminar baixas frequências que podem ser produzidas pelo instrumento. A entrada V_{in} é diretamente a saída do captador do instrumento musical. A resultado pode ser visto na Figura 3.

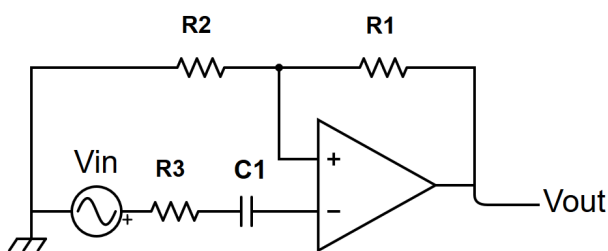


Figura 3. Circuito Amplificador AMPOP

Utilizou-se o TL071 [3], que é um amplificador operacional bem utilizado em aplicações para áudio, pois possui um *slew rate* bem menor que o famoso LM741.

B. Filtros

Existem três tipos de filtros essenciais:

- Passa alta: permite a passagem de apenas sinais acima de determinada frequência, normalmente alta.
- Passa baixa: permite a passagem de apenas sinais abaixo de frequência, normalmente baixa.

- Passa faixa: permite a passagem de frequências em uma determinada faixa.

Os filtros servirão para filtrar as faixas de frequências não audíveis do sinal como o ruído. Outras funções incluem diminuir o *ripple* da fonte e retirar o sinal DC da saída. É muito importante retirar eliminar qualquer sinal DC na saída pois isso pode causar danos físicos ao amplificador que será conectado na saída. Os filtros mais comuns são os filtros RC (Resistivo Capacitivo) passivos de primeira ordem por sua simplicidade e bons resultados. É passivo por possuir somente elementos eletrônicos passivos.

C. Alimentação

A alimentação do AMPOP normalmente é simétrica, sendo normalmente alimentado com $+V_{cc}$ e $-V_{cc}$. Para contornar essa restrição, utiliza-se um divisor de tensão e soma-se uma tensão de offset no sinal de entrada. Adiciona-se um capacitor (C_5) para minimizar os ruídos da fonte de alimentação no sistema, já que as fontes não possuem sinal DC perfeito. Por último, adiciona-se um filtro passa alta (C_4) para eliminar frequências muito altas e ruído. O circuito resultante pode ser visto na Figura 4. Por último, os filtros C_1 e C_6 atuam como passa faixa para limitar o sinal de entrada.

A alimentação do TL071 utilizada nas simulações pode ser proveniente de uma bateria ou uma fonte de tensão de 9V. Um divisor de tensão com resistores iguais fornecem uma tensão de aproximadamente 4.5 V, que será utilizado como offset para o sinal proveniente da guitarra. Desta forma, não haverá corte do sinal. Quando utiliza-se uma bateria não se tem problemas com variações no fornecimento da tensão além da diminuição da tensão durante a descarga, porém, como a maior parte das fontes presentes no mercado são bastante problemáticas por possuírem uma grande tensão de *ripple*, introduz-se um capacitor eletrolítico em paralelo (C_5) com o divisor de tensão, que garantirá um fornecimento mais estável.

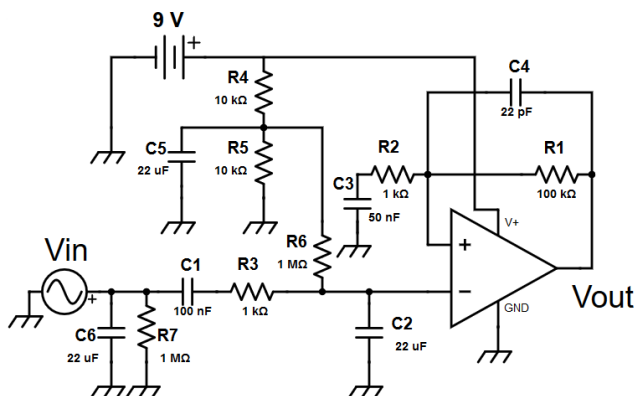


Figura 4. Circuito de Entrada

D. Clipping

Para realizar o *clipping* será utilizado um circuito com diodos como o da Figura 5. Se os diodos forem idênticos, o *clipping* será simétrico. Se os diodos tiverem pontos de ruptura diferentes, o *clipping* será assimétrico. Os dois tipos de *clipping* podem ser interessantes do ponto de vista sonoro.

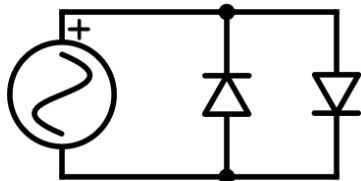


Figura 5. Circuito de Clipping

Diodos possuem variações em seu ponto de ruptura por conta dos diferentes materiais que o compõe, como exemplo Silício ou Germânio. Diodos de Silício em geral possuem uma tensão de ruptura de 0,6 V enquanto de germânio em volta de 0,2 V.

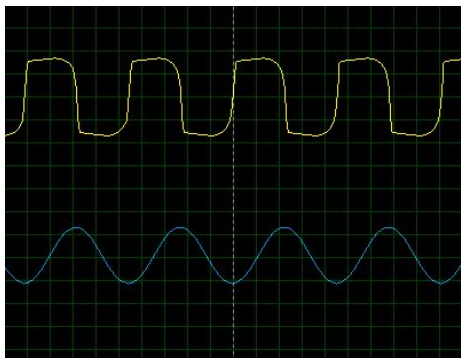


Figura 6. Resultado do Clipping Simplificado

O circuito da Figura 5 sozinho é capaz de modificar o sinal

(onda amarela) a partir do sinal de entrada (onda azul) da Figura 6.

III. CIRCUITO E SIMULAÇÃO

O circuito completo pode ser visto na Figura 7. Foram adicionado *switches* para selecionar o tipo de distorção utilizado (ou nenhuma distorção) e potenciômetros para variar tanto o ganho do circuito como a quantidade de distorção, além de um potenciômetro para variar a quantidade de filtragem do sinal.

A. Soft Clipping

Aplicando o sinal de entrada azul da Figura 6, e variando os valores dos potenciômetros de distorção e de filtro, pode-se obter algumas formas de onda interessantes a partir da senoidal inicial (Figuras 8, 9, 10 e 11). Cada uma das formas de onda tem um som característico, que normalmente é escolhido pelo músico tentando várias configurações diferentes e observando a que melhor se encaixa nas necessidades.

B. Hard Clipping

Aplicando o sinal de entrada (azul) da Figura 6 e variando os valores dos potenciômetros de distorção e de filtro, pode-se obter algumas formas de onda características do *hard clipping*, que distorce mais agressivamente a onda senoidal (Figuras 12, 13 e 14).

IV. RESULTADOS EXPERIMENTAIS

Os resultados experimentais obtidos no osciloscópio variando as configurações de filtro e distorção, de forma a tentar recriar as formas de ondas obtidas nas simulações podem ser vistos nas Figuras 15, 16, 17, 18, 19 e 20.

V. CONCLUSÃO

Nesse artigo foi abordado a construção de um circuito para distorção de áudio. Os resultados experimentais do circuito foram muito satisfatórios quando utilizados em um instrumento real, pois apresentaram um som razoável em distorção, mas ruído excessivo devido à falta de blindagem no circuito construído.

No entanto, o custo de produção do circuito integrado é muito inferior ao custo de um equipamento de áudio profissional. Vale lembrar que mesmo equipamentos musicais industriais não são livres de ruídos, e outros equipamentos ao pedal para amenizar esse feito, que está do escopo do artigo.

Uma das melhorias que podem ser feitas no circuito é a blindagem ou utilização de caixa metálica aterrada para melhorar a qualidade de som.

REFERÊNCIAS

- [1] R. C. Jaeger and T. N. Blalock, *Microelectronic circuit design*. McGraw-Hill New York, 1997, vol. 97.
- [2] Wikimedia Commons, "Clipping waveform," https://en.wikipedia.org/wiki/File:Clipping_waveform.svg, [Online].
- [3] T. Instruments, "Tl071, tl071a, tl071b, tl072, tl072a, tl072b, tl074, tl074a, tl074b low-noise jfet-input operational amplifiers (slos080d). august 1996," 2004.

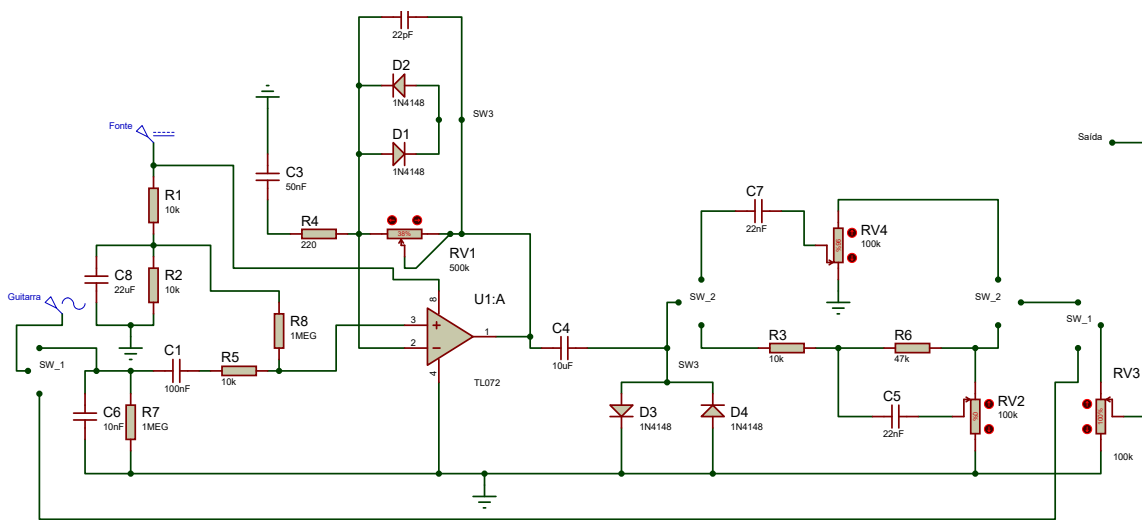


Figura 7. Circuito Completo



Figura 8. Simulação de *Soft Clipping* com parâmetros arbitrários 1



Figura 9. Simulação de *Soft Clipping* com parâmetros arbitrários 3



Figura 10. Simulação de *Soft Clipping* com parâmetros arbitrários 4

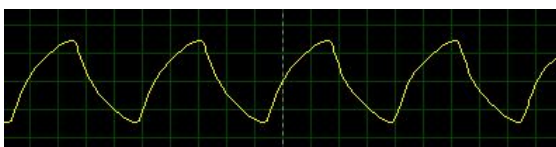


Figura 11. Simulação de *Soft Clipping* com parâmetros arbitrários 5

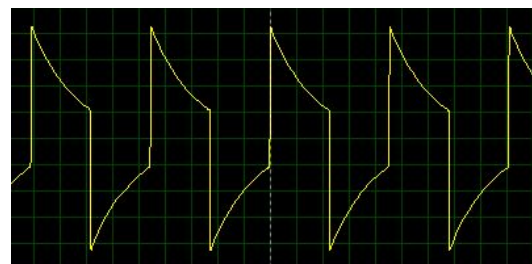


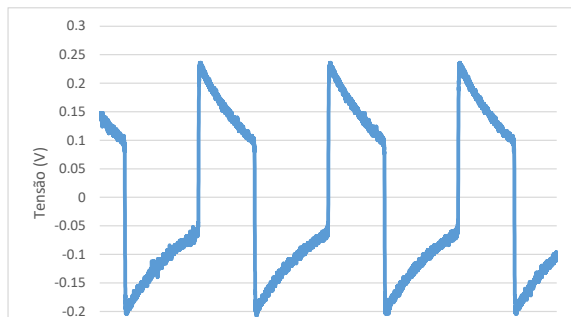
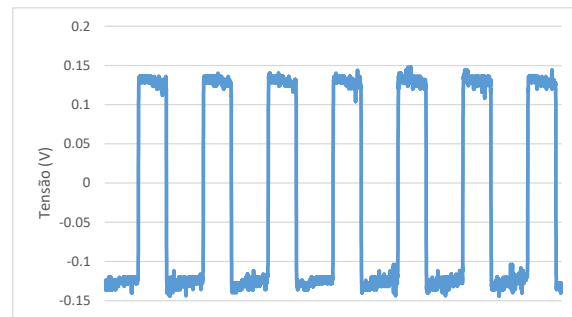
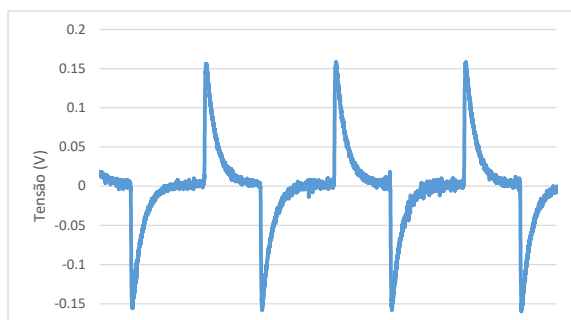
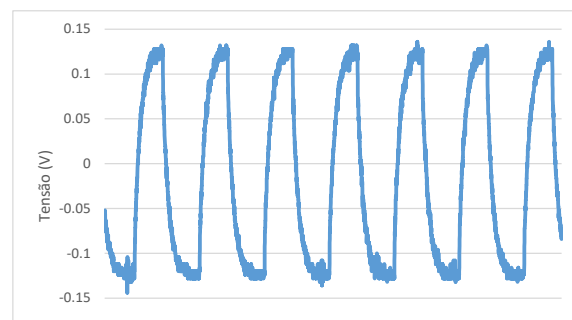
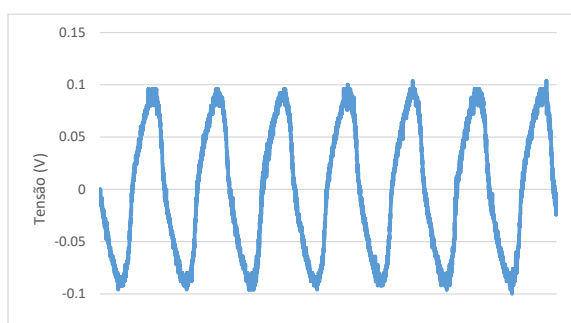
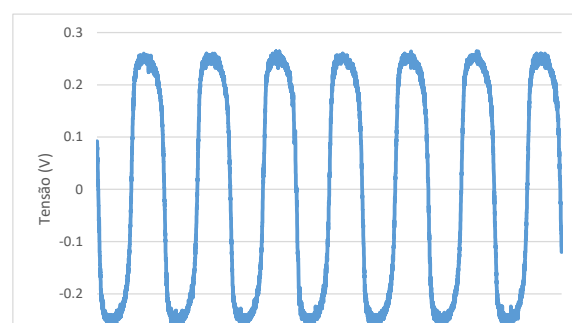
Figura 12. Simulação de *Hard Clipping* com parâmetros arbitrários 1



Figura 13. Simulação de *Hard Clipping* com parâmetros arbitrários 2



Figura 14. Simulação de *Hard Clipping* com parâmetros arbitrários 3

Figura 15. *Clipping* Experimental baseado na Figura 12Figura 18. *Clipping* Experimental baseado na Figura 14Figura 16. *Clipping* Experimental baseado na Figura 13Figura 19. *Clipping* Experimental baseado na Figura 8Figura 17. *Clipping* Experimental baseado na Figura 11Figura 20. *Clipping* Experimental baseado na Figura 9

Reguladores conversores de Tensão

Uma comparação prática

André Luiz de Sousa Marcondes Reuter
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina
Blumenau, Brasil
andre.reuter@grad.ufsc.br

Juliano Emir Nunes Masson
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina
Blumenau, Brasil
juliano.masson@grad.ufsc.br

Resumo—Este artigo apresenta uma breve explicação sobre conversores DC-DC Buck e reguladores lineares, bem como uma comparação entre um conversor Buck discretizado, um LM2676-ADJ e um conversor linear KA7808 com diferentes cargas.

Palavras-chave—regulador de tensão; LM2676-ADJ; KA7808; conversores Buck .

I. INTRODUÇÃO

A alimentação de microcontroladores em sistemas embarcados precisa ser estável devido à sensibilidade dos circuitos e para que a aplicação seja precisa. Com o uso de baterias, o nível de tensão varia com o tempo e é necessário a regulação da tensão além do monitoramento da carga da bateria. Muitos dos circuitos possuem alimentação que difere da tensão dos elementos atuadores que são controlados, por isso é necessário fazer a conversão e o controle da tensão de entrada, geralmente usa-se tensões menores do que a fonte (bateria) provê, por isso os conversores abaixadores de tensão são de grande utilidade nesses sistemas.

Um exemplo deste tipo sistema seriam os drones, que trabalham geralmente entre 12 e 15V, mas os controladores necessitam de níveis de tensão menores entre 7 a 9V com alta estabilidade e baixa corrente. O sinal de comando de motores também opera a um nível menor de 5V, e este sinal é provido pelo controlador que possui regulador interno, porém por questões de dissipação de energia é interessante que a tensão que alimenta o regulador seja menor.

Os reguladores conversores de tensão tem algumas classes diferentes, os mais populares são os tipo *switch* e os lineares. Reguladores tipo *switch* ainda são separados em algumas categorias, apresentaremos o regulador conversor *Buck* que abaixa a tensão [1]. Na Fig. 1. temos o esquemático de um conversor *Buck*.

O funcionamento se dá da seguinte forma: a fonte de tensão V_I é a entrada que pode ser uma bateria, por exemplo. Em seguida temos um transistor MOSFET que é o elemento comutador (*switch*), o MOSFET é fechado e aberto durante a operação por um sinal de onda quadrada PWM (*Pulse Width Modulation*). Em seguida o sinal é filtrado pelo filtro LC, fornecendo uma saída DC com o nível de tensão desejado na

carga representada por R. O diodo CR1 serve para impedir o acúmulo de carga no instante em que o *switch* está aberto, esse acúmulo ocorre devido a corrente inversa vinda do indutor durante a abertura do *switch*. De outra maneira teríamos uma grande tensão num instante de tempo muito pequeno no circuito e isso poderia destruir os componentes, com o diodo o circuito é protegido, além de fazer parte da filtragem [2].

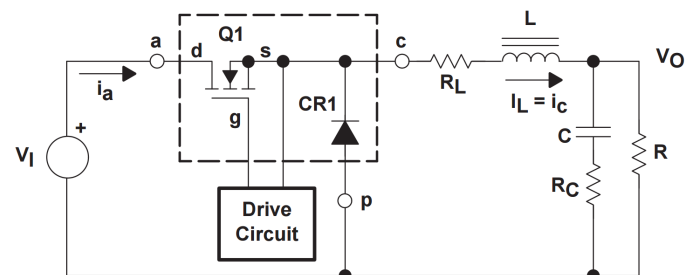


Fig. 1. Esquemático de um conversor do tipo Buck [2]

Na prática, um circuito de controle em malha fechada é necessário para que a saída se mantenha constante de acordo com a variação da entrada, isso aumenta consideravelmente a complexidade desses conversores, porém os torna mais precisos e eficientes. Os conversores lineares possuem construção bem mais simples, e se baseiam na limitação de potência fazendo o uso de um divisor de tensão [3]. Na Fig. 2. temos um exemplo de implementação de um conversor linear série.

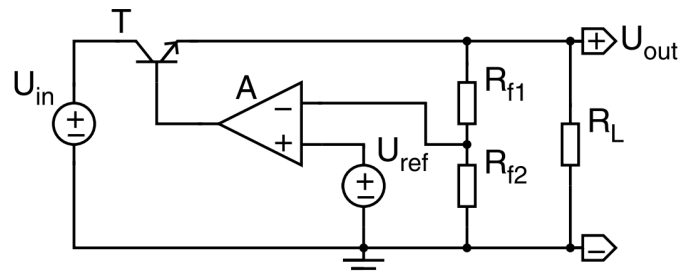


Fig. 2. Implementação simples de um conversor linear série [3]

A fonte U_{in} é a entrada do circuito e o amplificador operacional A é o elemento de controle. Essencialmente, a tensão U_{out} é mantida constante pela medição da saída e o

controle da tensão de referência. Na prática, como temos uma configuração do amplificador operacional de diferença, temos U_{out} dado pela fórmula em (1).

$$U_{out} = U_{ref} \frac{R_{f1} + R_{f2}}{R_{f2}} \quad (1)$$

A tensão de referência pode ser a mesma da entrada ou da saída. A simplicidade desses conversores os torna práticos para implementação e construção de circuitos integrados, porém como é apenas baseado na dissipação de energia, consome consideravelmente mais, aumentando o calor dissipado, fazendo o uso de dissipadores obrigatório, além de diminuir a eficiência [3].

Neste artigo será apresentada uma breve comparação entre conversores DC-DC *Buck* e conversores lineares para aplicação de corrente de saída até 1A com tensão de entrada entre 9 a 15V simulando a descarga de uma bateria, bem como a comparação entre o circuito integrado LM2676-ADJ e o seu equivalente discreto.

II. PROJETO E SIMULAÇÃO

O projeto consiste em duas partes, sendo a primeira a discretização do circuito integrado (CI) LM2676-ADJ, que é um conversor *Buck* ajustável. Para isso nos baseamos no seu princípio de funcionamento e no circuito da Fig. 1. O CI possui algumas características interessantes além da sua função principal, como proteção contra picos de corrente, baixo consumo de energia e alta eficiência [4]. No entanto, o custo desse chip é muito maior do que um conversor linear, principalmente pela complexidade da eletrônica de controle que existe para que a saída se mantenha estável de acordo com a variação da entrada. Na Fig. 3. temos o diagrama de bloco do circuito.

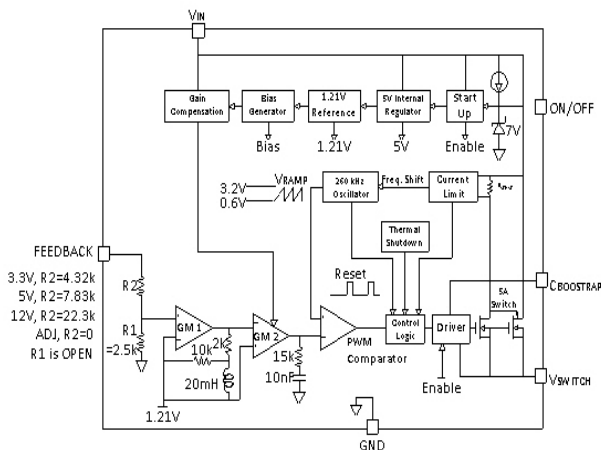


Fig. 3. Diagrama de bloco do circuito integrado LM2676-ADJ[4]

Como podemos ver na Fig. 3. a complexidade do circuito é elevada e a maioria das partes do circuito são omitidas, por isso para replicar a função desse CI utilizamos o circuito da Fig. 1. dimensionado para entradas entre 10 a 20V e uma saída de 8V, com corrente de saída de até 1A. Na Fig. 4. temos o digrama do circuito em LTSpice.

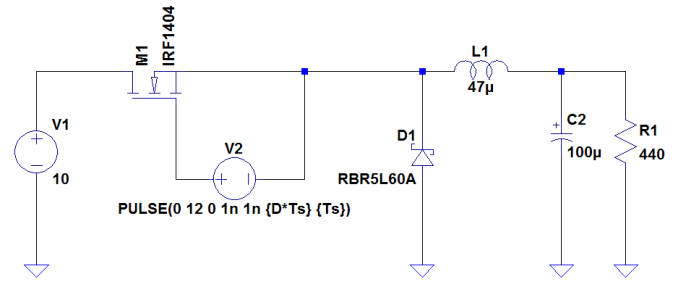


Fig. 4. Diagrama do circuito montado no LTSpice

Na Fig. 4. temos um diodo *Schottky* em contraste com a Fig. 1. que apresenta um diodo comum. Diodos *Schottky* apresentam uma queda de tensão muito menor do que os diodos convencionais, por isso são ideais para essa aplicação, pois aumentam a eficiência geral do circuito consumindo menos energia [5]. Usamos um circuito simples com o CI NE555 para geração do sinal PWM como pode ser visto na Fig. 5..

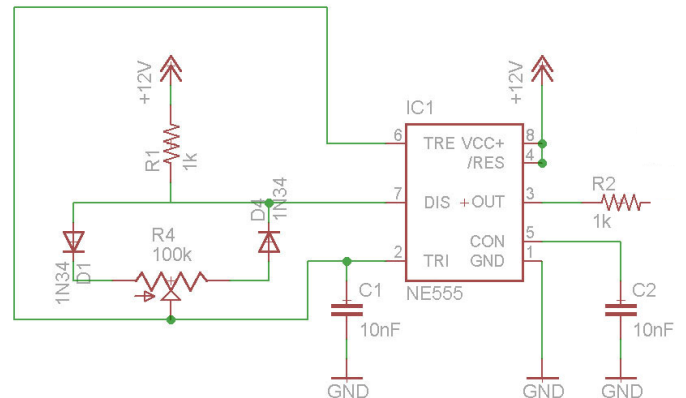


Fig. 5. Diagrama do circuito PWM [6]

No circuito a saída passa diretamente para o MOSFET, na Fig. 4. Com esse circuito a máxima frequência da saída é de 100Hz, bem abaixo do LM2676-ADJ que chega a 260kHz, como será mostrado a seguir esse fator altera a regulação.

Na Fig. 4. podemos ver que nos parâmetros da fonte que representa o circuito do PWM temos as variáveis D e T_s que seriam um fator de ampliação (D) e o período da onda, sendo o *duty cycle* igual a $D \cdot T_s$. O *duty cycle* é o tempo no qual a onda fica em estado alto, ou seja, para um *duty cycle* de 50% temos 50% do tempo 12V e os outros 50% 0V.

Com esses valores (mantendo a tensão do PWM fixa) alteramos a regulação. A seguir temos o resultado para a simulação para uma tensão de entrada de 15V, carga resistiva de 440Ω, com $D = 20m$ e $T_s = 10ms$, desta forma temos que o *duty cycle* é 0.2ms.

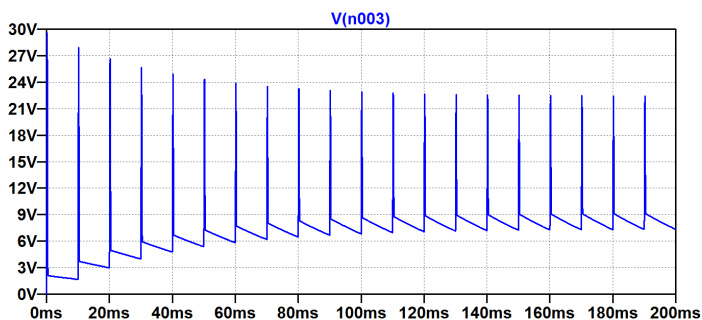


Fig. 6. Saída do circuito para frequência do PWM em 100Hz

O tempo de simulação foi pequeno pois em 200ms a resposta já está em regime permanente. Podemos ver que há vários picos de tensão e que o sinal também apresenta *ripple*, mas o sinal médio foi de 7.21V e o valor RMS foi de 7.71V, um resultado aceitável visto a baixa frequência. Podemos simular para uma frequência mais alta e verificar se há melhora no sinal.

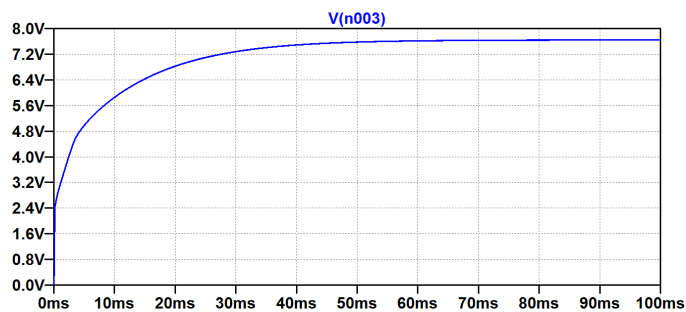


Fig. 7. Saída do circuito para frequência do PWM em 100kHz

Vemos que com uma frequência de 100kHz o sinal não possui nenhum pico, além de atingir o regime permanente na metade do tempo, sendo necessário para isto um *duty cycle* maior. Com isso podemos ver que a frequência com que o MOSFET é aberto e fechado influencia totalmente a saída do circuito. No entanto, nosso intuito é ver o comportamento do sistema de acordo com a queda na tensão de entrada, simulando uma bateria descarregando em um drone. A seguir temos a mesma simulação da Fig. 6. porém com a tensão de entrada em 10V.

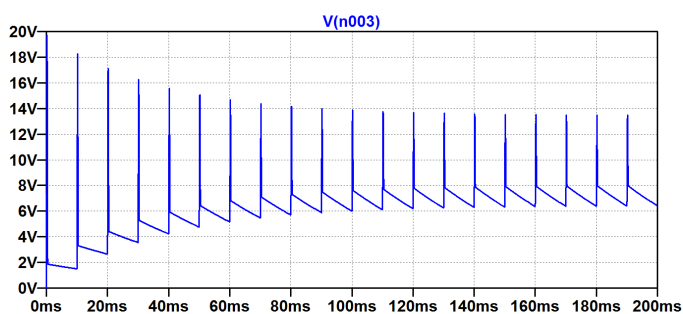


Fig. 8. Simulação para entrada de 10V

Agora, para o mesmo *duty cycle*, a saída é menor com valor médio de 6.2V e 6.5V RMS. Isso significa que algo deve ser alterado para compensar a queda, a única variável que podemos alterar é o *duty cycle*, variando o fator D que agora é de 20m. Alterando D para 50m já obtemos o valor médio de 7.7V com

7.75V RMS. Isso mostra que apenas uma pequena variação no *duty cycle* pode compensar a diferença na tensão de entrada, o mesmo vale para uma carga variável.

Esse parâmetro é ajustado automaticamente no CI LM2676-ADJ pela eletrônica de controle em malha fechada, mas no circuito discreto isso é feito ajustando o potenciômetro do circuito do PWM como é visto na Fig. 5. Se aplicássemos esse circuito em um drone, por exemplo, o microcontrolador que está sendo alimentado poderia fornecer o sinal PWM e ajustar o *duty cycle* de acordo com a tensão que está recebendo.

III. RESULTADO EXPERIMENTAIS

Para montarmos o circuito utilizamos os seguintes materiais:

Conversor *Buck*:

- 1 Transistor MOSFET IRF1404
- 1 Diodo *Schottky* SB5100
- 1 Indutor 47 μ H/1A
- 1 Capacitor eletrolítico 100 μ F/25V
- 1 Resistor carbono 440 Ω 1W
- 1 placa de ensaio

PWM:

- 1 CI NE555
- 1 Potenciômetro 10k Ω
- 2 Capacitores poliéster 100nF
- 2 Diodos 1N34
- 2 Resistores 1k Ω
- 1 placa de ensaio

Através da Fig. 9. podemos conferir ambos os circuitos montados.

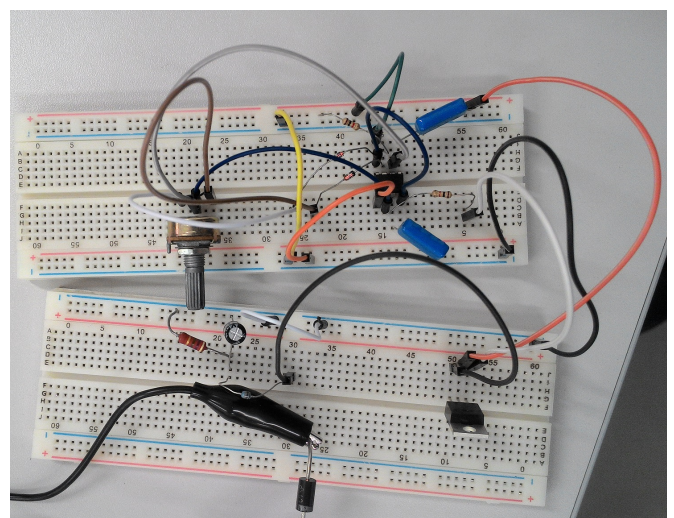


Fig. 9. Conversor Buck sendo alimentado com a mesma tensão de entrada do PWM

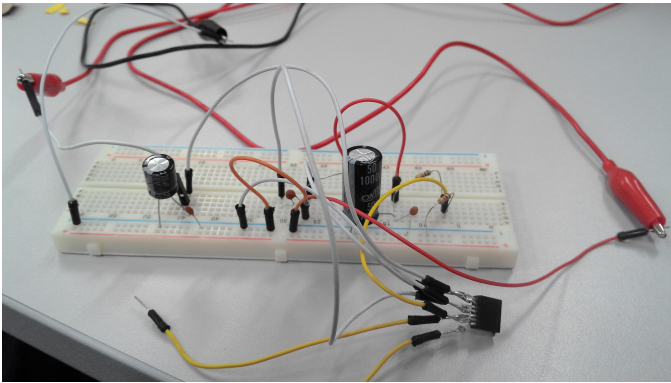


Fig. 12. Montagem do circuito com CI LM2676-ADJ

Em seguida testamos o circuito com um regulador conversor linear KA7808. Para esse circuito utilizamos apenas dois capacitores eletrolíticos de $10\mu\text{F}$ conectados ao *input* e ao *output*. As cargas foram as mesmas utilizadas anteriormente. Os resultados foram como esperado com a tensão de saída ficando entre 7.8 e 7.9V de forma mais estável que o LM2676-ADJ e muito parecido com o circuito *Buck* discretizado. O circuito montado é apresentado na Fig. 13.

IV. ANÁLISE DOS DADOS

Após a comparação entre os circuitos podemos perceber que os que forneceram melhores resultados foram o conversor *Buck* discretizado e o conversor linear. A eficiência do conversor linear foi baixa com a carga resistiva ficando apenas em 44%, porém a estabilidade da tensão foi alta e a regulação e conversão foi efetiva. O conversor *Buck* discreto também forneceu ótimos resultados apesar da baixa frequência que o chip gerador do sinal PWM consegue atingir, formando um sinal com *ripple*. Esse circuito também se saiu bem com uma carga mais alta em um teste com uma pastilha *peltier* TEC1-12705 que chegou a drenar 900mA em 5V, onde não houve aumento de temperatura significativo em nenhum dos componentes do circuito. A corrente teve que ser limitada por conta do indutor que trabalha apenas até 1A.

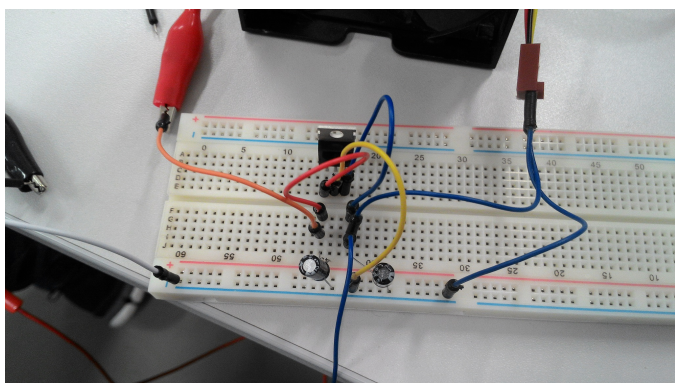


Fig. 13. Circuito com regulador conversor linear

Em contraste o CI LM2676-ADJ não apresentou bom rendimento e se mostrou um componente de difícil manuseio. Claro que se as condições impostas pelo fabricante forem respeitadas estritamente, esse conversor pode chegar a 94% de eficiência. Porém não foi o caso, devido as grandes oscilações na saída, um valor de eficiência preciso não pode ser calculado.

Em todo o caso, para aplicações simples onde há apenas necessidade de abaixamento da tensão para um valor de entrada que não seja mais que 3 vezes maior que a saída, com correntes de saída entre 100mA a 500mA o conversor linear é a melhor opção, além de não haver fluxo indutivo no circuito fazendo com que haja muito menos interferência eletromagnética. Para o caso de correntes maiores o conversor *Buck* é recomendado pelo menor consumo de energia, além de ser capaz de mudar de forma dinâmica a regulação de acordo com a carga ou a variação da entrada, além de ser uma ótima opção para uma fonte de tensão regulável, ou se fixarmos o *duty cycle*, uma fonte de corrente variável.

Alguns fatores podem contribuir para melhora dos resultados para o projeto do conversor *Buck*, como diminuir a distância entre os componentes, aumentar a capacidade de corrente do indutor, gerar um sinal PWM de maior frequência e adicionar resistências antes do indutor e do capacitor.

Para efeito comparativo o circuito conversor *Buck* foi testado com o sinal PWM em 100kHz com amplitude de 10V fornecido por um gerador de funções. Isso nos forneceu uma saída muito mais estável do que em 100Hz, comprovando os resultados das simulações. A tensão foi regulada para 6.8V neste caso.

V. CONCLUSÃO

Com esse projeto conseguimos ter a noção de propriedades dos reguladores conversores de tensão e determinar o que se encaixa melhor de acordo com as necessidades impostas. Apesar de existirem circuitos integrados prontos que fazem a tarefa, em alguns casos o projeto próprio pode oferecer melhor desempenho por menos custo.

Os testes realizados foram satisfatórios e suficientes para esse tipo de circuitos. No entanto, algumas dificuldades foram encontradas, principalmente no quesito metodologia de testes, visto que para que os dados signifiquem algo, é preciso que o teste seja justo com as três modalidades de circuito. O que poderia ser melhorado nos testes realizados seria a padronização da carga, com corrente e tensão de alimentação fixa, isso faria com que a determinação da eficiência dos circuitos fosse muito mais precisa.

Se um dos circuitos fosse escolhido para implementação prática em um drone, o conversor *Buck* discretizado seria esse circuito, pois com alguns componentes de baixo custo e usando o sinal PWM fornecido pelo controlador, temos alta eficiência e precisão. No geral, o projeto proporcionou uma visão mais abrangente sobre esse tipo de circuito e também sobre as dificuldades de implementação de circuitos de controle.

REFERÊNCIAS

- [1] Keeping, S., Understanding the Advantages and Disadvantages of Linear Regulators. Disponível em: <http://www.digikey.com/en/articles/techzone/2012/may/understanding-the-advantages-and-disadvantages-of-linear-regulators>. Acesso em: 04 de junho 2016
- [2] Texas Instruments, "Understanding Buck Power Stages In Switchmode Power Supplies", TI, Dallas, TX, Relatório de aplicação SLVA057, Março 1999.

- [3] Wens, M. e Steyaert, M., "Basic DC-DC Converter Theory", em Design and Implementation of Fully-Integrated Inductive DC-DC Converters in Standard CMOS, Holanda, Springer, 2011, cap. 2, p. 27-63.
- [4] Texas Instruments, LM2676 SIMPLE SWITCHER High Efficiency 3-A Step-Down Voltage Regulator, Disponível em: <http://www.ti.com/lit/ds/symlink/lm2676.pdf>
- [5] Future Electronics, What is a Schottky Diode?, Disponível em: <http://www.futureelectronics.com/en/diodes/schottky-diodes.aspx>. Acesso em: 04 de junho 2016
- [6] Afrotechmods, What is PWM? Pulse Width Modulation Tutorial in HD, Disponível em: <http://afrotechmods.com/tutorials/2011/11/28/pulse-width-modulation-pwm-tutorial/>. Acesso em: 04 de junho 2016

- A alimentação foi feita com uma fonte de 12V, mas o TDA aquece até 24V
- o sinal de entrada um conector p2 3.5mm

Calculamos o ganho teórico do circuito usando a equação (1) anteriormente citada.

$$G = 1 + \frac{R_f}{R_g} = 1 + \frac{22000}{680} \approx 33,35 \quad (2)$$

Obtendo como resultado que o ganho teórico do circuito é $G \approx 33,35$.

A. Simulação

Para realizar a simulação utilizamos o software LTSpice. Construímos o circuito da Fig. 3 na ferramenta de simulação, usando um amplificador operacional TDA 2030 que apresenta as mesmas características do TDA 2050. Alimentamos o amplificador com 12V e aplicamos uma entrada senoidal com amplitude de 50mV e frequência de 440Hz Fig. 4.

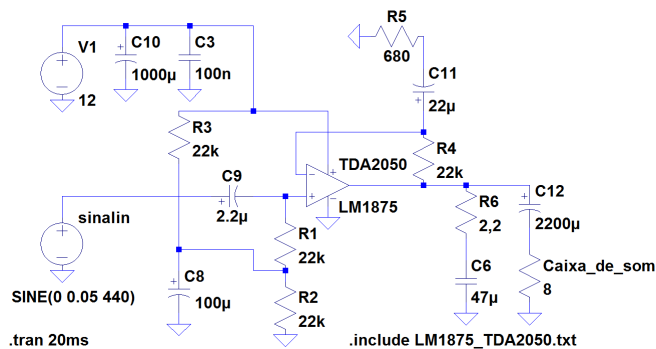


Figura 3. Sinal aplicado na entrada da simulação do circuito.

Na figura 5 podemos ver a entrada, em verde, e a saída, em azul claro, mostrando que houve um ganho significativo de amplitude sem distorcer ou defasar o sinal. Pela simulação obtivemos um ganho de amplitude do sinal de 31,7 vezes.

B. Protótipo e Circuito Impresso

Por fim fizemos a montagem do circuito impresso do amplificador. Para obter um resultado com melhor acabamento decidimos projetar a placa de circuito impresso usando o software EAGLE. Usando as bibliotecas de componentes do Eagle foi possível encontrar os componentes, semelhantes aos usados na protobord, formando uma placa com o tamanho ideal para os componentes encontrados nas lojas locais. Além do uso das mesmas peças do circuito em protobord foi acrescentados um conector p2/3.5mm, comum em aplicações de áudio stereo para entrada e na saída foi usado um terminal padrão para áudio estéreo, assim como um Jack P4 para entrada de energia, também foi acrescentado um potenciômetro de dois canais para uma regulagem de volume. Na figura 5 podemos ver a entrada, em verde, e a saída, em azul claro, mostrando que houve um ganho significativo de amplitude sem distorcer ou defasar o sinal. Pela simulação obtivemos um ganho de amplitude do sinal de 31,7 vezes.

Essa modelagem gerou a seguinte trilha de circuitos Fig. 6.

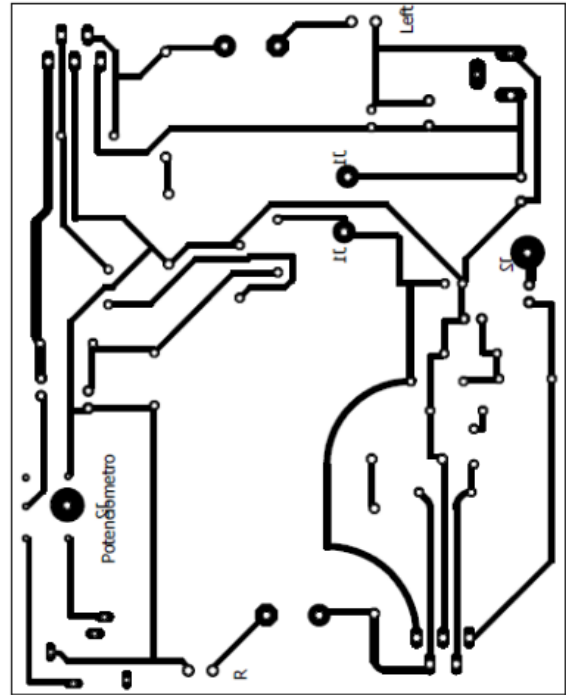


Figura 6. Projeto do circuito impresso.

III. RESULTADOS EXPERIMENTAIS

Com o circuito montados em *protoboard* fizemos uma bateria de testes, aplicando diferentes sinais de entrada, de diferentes formatos de onda até algumas músicas, para observar se o circuito apresentava ganho de amplitude satisfatoriamente próximo do valor obtido pelas simulações e se ocorriam distorções no sinal de saída.

Para uma melhor representação dos resultados, o circuito foi ligado a um osciloscópio, onde é possível ver os resultados apresentados nas Fig. 7 e Fig. 8 em que temos os sinais de entrada representados em azul e as saídas em amarelo.

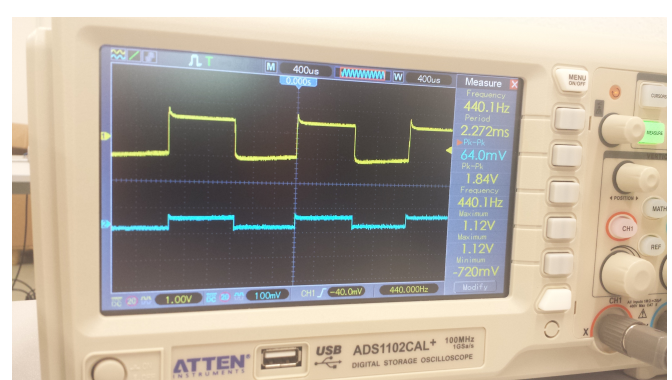


Figura 7. Sinal de entrada e saída do teste aplicando onda quadrada.

Aplicando um sinal de forma senoidal com amplitude de 52mV na entrada obtivemos um sinal senoidal de amplitude 1,6V e mesma frequência na saída. Quando aplicamos uma

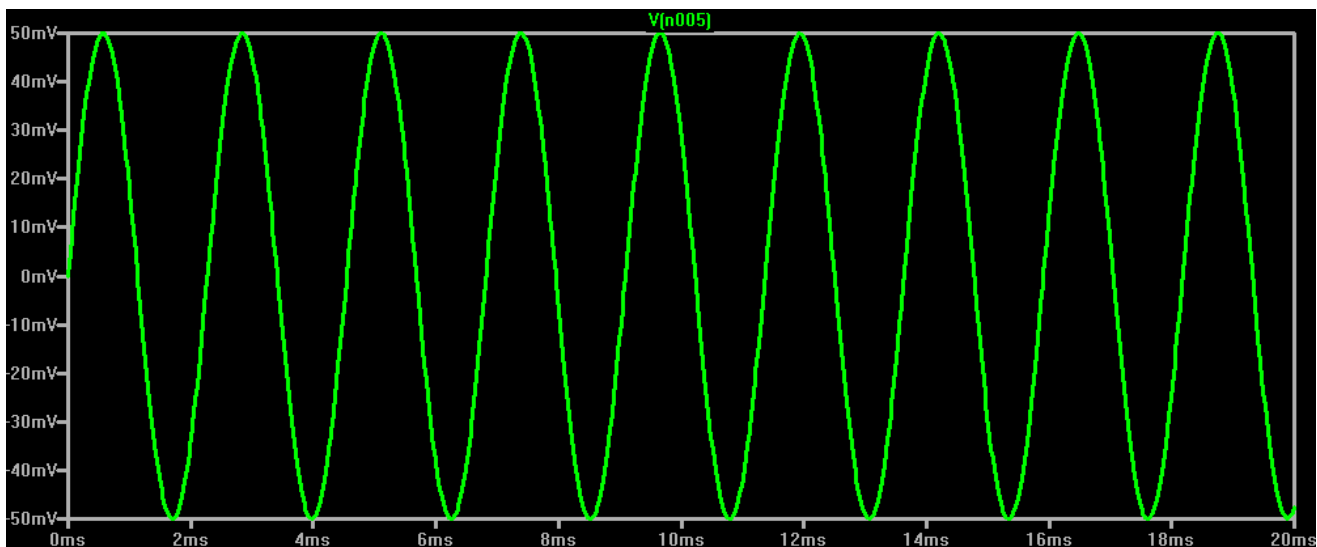


Figura 4. Sinal aplicado na entrada do circuito simulado.

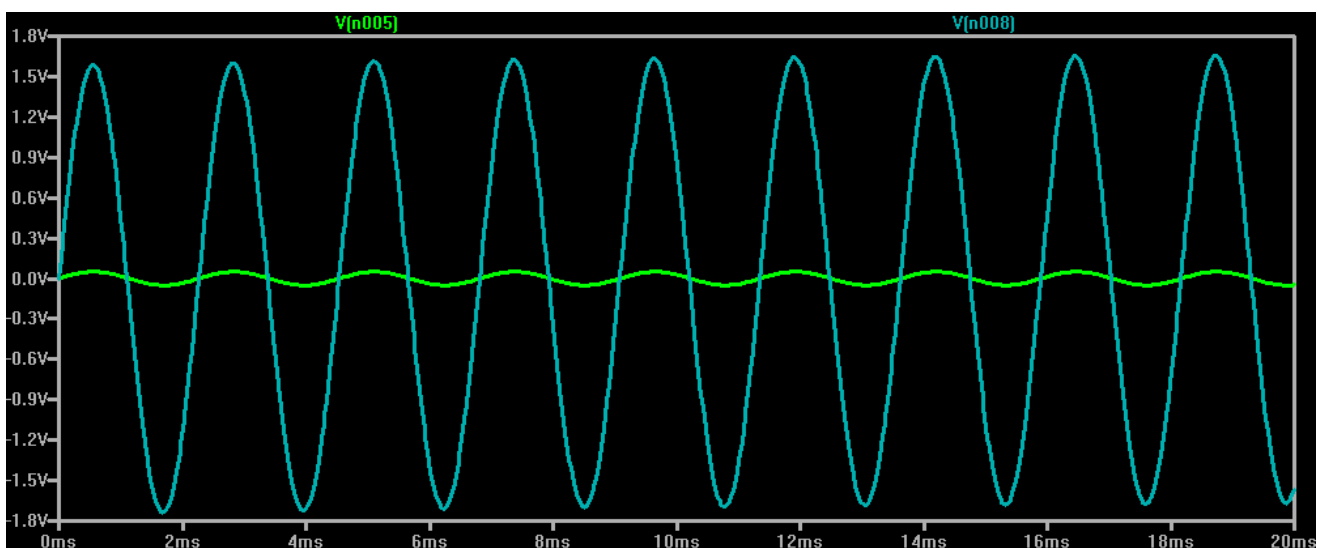


Figura 5. Sinal de saída aplicado na carga (azul claro) comparado com o sinal de entrada (verde).

entrada em forma de onda quadrada com 64mV o sinal de saída apresentou amplitude de 1,84V. O circuito apresentou comportamento semelhante para todas as entradas que testamos.

Ao usar o circuito em protobord por alguns minutos observou-se que os amplificadores esquentavam muito, então foi necessário o uso de dissipadores de calor.

IV. ANÁLISE DOS DADOS

Pudemos observar que os sinais não sofriam distorções perceptíveis desde que os sinais de entrada permanecessem dentro de uma faixa específica de amplitude, entre 50 a 150 mV. O ganho de amplitude real ficou suficientemente próximo do valor teórico e do valor obtido pela simulação. Sendo o valor do ganho teórico 33,35 vezes, o valor do ganho

obtido pela simulação aproximadamente 33 vezes e o valor dos ganhos obtidos pelos experimentos com o protótipo entre 28 a 30 vezes maior que a amplitude da entrada. O ganho real apresenta uma pequena redução em relação ao teórico, o que já era esperado devido as características não ideais do circuito físico.

V. CONCLUSÃO

Ao fim do projeto obtivemos resultados muito satisfatórios, conseguindo implementar um circuito amplificador de áudio stereo com distorção mínima. Tivemos alguns problemas durante a fase de prototipagem devido a estrutura física do TDA 2050, especificamente os terminais do amplificador, que criaram problemas de mau contato com a protoboard. O circuito final apresentou comportamento muito próximo da

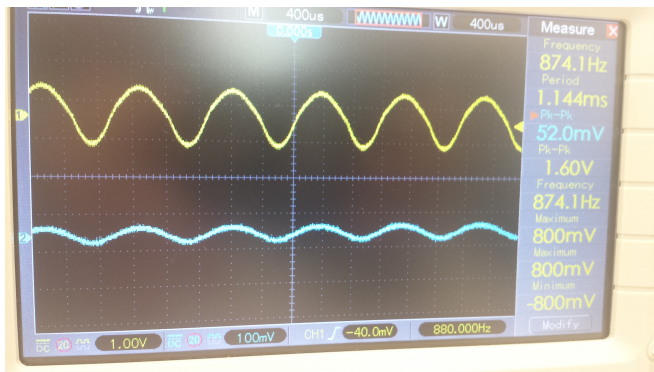


Figura 8. Sinal de entrada e saída do teste aplicando senóide.

simulação, tendo apenas uma pequena diminuição no ganho de amplitude do sinal.

REFERÊNCIAS

- [1] A. J. Pertence, *Amplificadores operacionais e filtros ativos: teoria, projetos, aplicações e laboratório*. São Paulo, Brasil: McGraw, 1988.

Stun Gun

Arma de Eletrochoque

Caique Sanches

Diogo Luiz Demarchi

Enrique Adami Geske

Engenharia de Controle e Automação
Universidade Federal de Santa Catarina - UFSC
Blumenau, Santa Catarina - Brasil

Resumo—Este trabalho consiste na construção de uma *Stun Gun*, um aparato que utiliza eletrochoques para imobilização momentânea de alguma pessoa. Comumente utilizada para defesa pessoal em situações de risco, seu uso gera controvérsias até os dias de hoje. Comparado com as tensões alcançadas por armas que são comercializadas, o circuito criado tem uma saída muito inferior. Utilizamos como base uma série de projetos achados na internet que possibilitaram o desenvolvimento de nosso dispositivo.

Palavras-Chave—*stun gun*; *taser*; *eletrochoque*; *circuito*; *eletrônica*.

I. INTRODUÇÃO

Uma arma de eletrochoque (*stun gun*) consiste em um dispositivo que utiliza de descargas elétricas de alta tensão para afetar a vítima. Têm o efeito de imobilizar uma pessoa, por curto período de tempo. É largamente utilizada pela polícia em diversos locais do mundo quando não se faz necessário o uso de uma arma de fogo, visto que essa é uma arma de baixa letalidade.

O objetivo do trabalho foi construir um dispositivo semelhante à uma *stun gun*. O princípio básico do circuito é transformar a corrente contínua da bateria em alternada e elevar a tensão significativamente, com uma corrente baixíssima, criando assim um arco elétrico e uma descarga elétrica segura, por causa da baixa corrente.

Foram desenvolvidos desde a simulação até a montagem e realizados vários testes, os quais estão descritos no decorrer do artigo, juntamente com os resultados obtidos.

II. STUN GUN

Stun Gun é o nome comumente utilizado para designar uma arma não letal que usa descargas elétricas de alta tensão para causar perda do controle voluntário dos músculos momentaneamente ou causar dor machucar significativamente uma pessoa. Diferentemente de um taser, a *Stun Gun* é uma arma de contato, ou seja, os eletrodos estão acoplados ao dispositivo, no caso do taser, os mesmos são projéteis disparados pelo aparato. [1]

Stun Guns são utilizadas comumente por civis para defesa pessoal em caso de algum ataque, roubo, agressão, enquanto tasers são largamente utilizados por policiais e militares,

principalmente nos EUA e Canadá, porém, o controle dos mesmos é muito rígido, sendo que para um portar um taser, ele tem de passar por uma prova onde ele é atingido pelo dito cujo.

III. PROJETO E SIMULAÇÃO

Para obtenção do objetivo do projeto e a elaboração do circuito foram utilizados os seguintes materiais:

- Resistores
 - 1M Ω
 - 47k Ω
 - 1,2k Ω
 - 1k Ω
 - 560 Ω
- Diodos
 - 1N4007
 - 1N4148
- Capacitores
 - 0,47 μ F
 - 0,01 μ F
- Transistor
 - TIP31
- Temporizador
 - LM555
- Transformador
 - 220V/12V
- Bateria
 - 9V

Com isso foi elaborado o seguinte circuito:

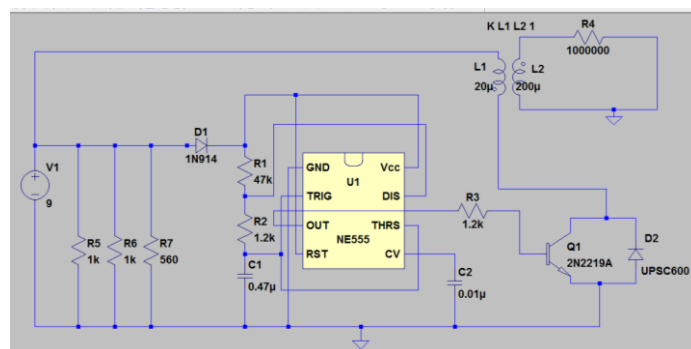


Figura 1. Circuito da *stun gun*.

Os resistores de $1K\Omega$ e 560Ω , do lado esquerdo do circuito, foram posicionados para criar um dreno de corrente. Assim a corrente que circula do circuito não será alta como para queimar os componentes e a corrente na saída do transformador será menor ainda.

O temporizador LM555 foi utilizado para gerar pulsos com o sinal recebido da fonte. Esses pulsos resultantes passam por um transistor, que é utilizado para dissipar potência, depois amplifica o sinal, junto com o diodo, para depois envia-lo para o transformador. O transformador funciona apenas com variações na corrente, como uma corrente alternada, por isso foi necessária à geração de pulsos e não a utilização direta da fonte. Ele converte a tensão mantendo a mesma potência em ambos os terminais, fazendo com que a tensão aumente e diminua a corrente ou vice-versa.

A tensão gerada nos terminais do transformador deve ser suficientemente alta para romper a constante dielétrica do ar, ou seja, fazer com que o ar passe a conduzir eletricidade e seja possível observar o arco voltaico.

Foi criada uma simulação no LTSpice para verificar as propriedades do circuito. Para isso foi colocada uma resistência alta na saída do transformador para simular um circuito aberto ou a resistência de um corpo.

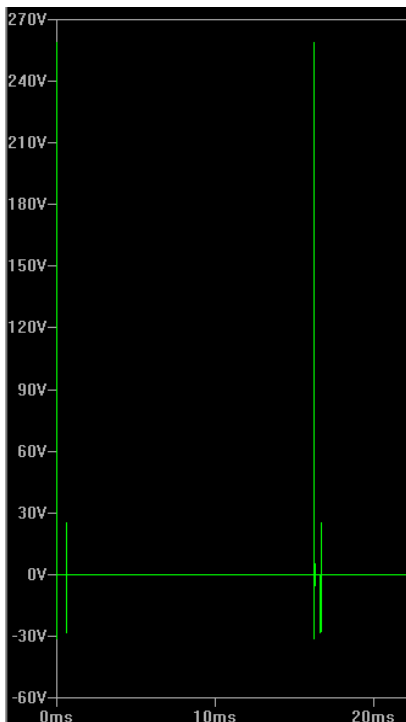


Figura 2. Tensão gerada na saída do transformador.

Analisando o gráfico da tensão obtido podemos observar que foram criados picos de tensão que atingem aproximadamente os 260V. E na figura 3 pode ser visto que a corrente no resistor é muito baixa, não havendo perigo ao utilizar a arma.

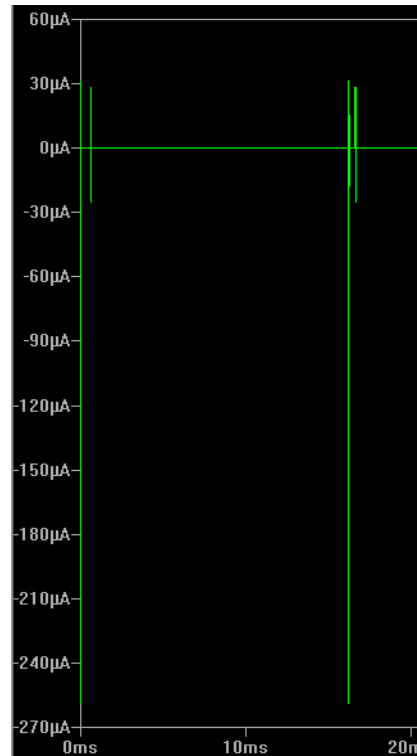


Figura 3. Corrente no resistor da saída.

Os componentes, como diodos e transistores, diferem entre a simulação e a realização experimental, porém atendem as mesmas especificações para que não haja uma grande discrepância entre os resultados.

IV. RESULTADOS EXPERIMENTAIS

A partir do modelo da figura 1, foi montado um circuito utilizando uma *protoboard* e os devidos componentes. Nela foram realizados testes para verificar as conexões e os parâmetros do circuito.

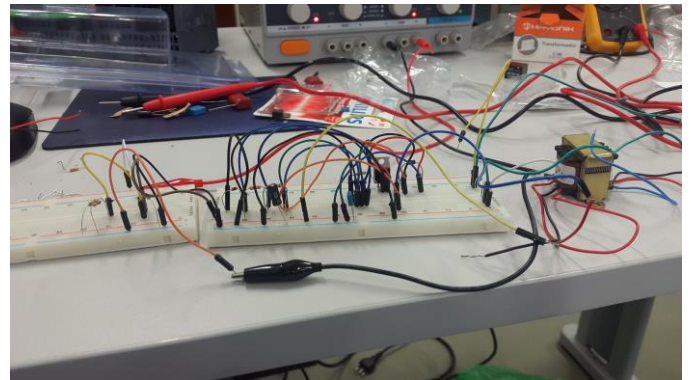


Figura 4. Circuito montado na protoboard.

Após a verificação, com uma placa de circuito padrão 10×10 , foram soldados os componentes em seus devidos lugares para obter um circuito fixo. Nela, foi colocado conectores para as entradas da fonte (bateria) e para as saídas do transformador, onde, após ligado, poderá ser visto o arco voltaico. A placa já finalizada pode ser observada na figura 5.

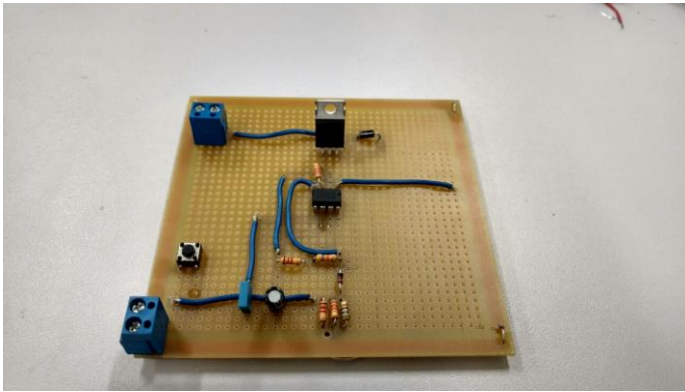


Figura 5. Circuito final.

Com a bateria de 9V na entrada, foi obtido um arco facilmente visível. Para verificar a tensão de saída foi utilizado o multímetro, que registrou aproximadamente 230V com variações devido a que o sinal é formado por pulsos.

Ouve uma diferença em torno de 30V entre os valores obtidos na simulação e os do experimento elaborado na placa. Esta discrepância é devido à dificuldade de recriar um sistema exato como o de uma simulação. Os componentes utilizados na simulação são diferentes aos do experimento, porém possuem as mesmas características, o que pode alterar levemente os resultados. Também, é bastante complicada a reprodução de um transformador real na simulação devido a que utiliza apenas valores de indutâncias e não valores para a transformação de tensão como temos nos exemplares reais.

V. CONCLUSÃO

A partir da realização do experimento, é possível ver que não é muito complicado fazer uma arma de imobilização que é utilizada, muitas vezes, pelos órgãos públicos de defesa. A princípio, é difícil imaginar que uma mera bateria de 9V possa fazer um efeito tão grande no corpo de uma pessoa, pois é possível segurar os dois polos da mesma com as mãos sem sentir absolutamente nada. Porém, depois de algumas etapas em um circuito básico, transforma-se em uma poderosa arma que já ajudou a sociedade em várias ocasiões.

Houveram algumas dificuldades para a realização do experimento, principalmente por causa do transformador. Necessita-se, idealmente, de um transformador de áudio, item não muito fácil de ser encontrado em lojas de eletrônica. Por conta disso, foi usado um transformador normal, que não deixa de seguir o mesmo princípio de funcionamento de um transformador de áudio, porém com algumas diferenças. Estes transformadores normais suportam uma corrente baixíssima e é extremamente fácil de queimá-los. Outro fato importante é que eles têm uma impedância relativamente baixa, o que prejudica a saída de alta tensão.

Também foram enfrentadas dificuldades com o resto do circuito, a exemplo do excesso de corrente que chegava ao mesmo, causando, assim, certo perigo à vítima. Isto foi facilmente resolvido criando um dreno de corrente e, deste modo, a arma ficou completamente segura e pronta para o uso sem o perigo de danos permanentes.

AGRADECIMENTOS

Deixamos expressos nossos sinceros agradecimentos às seguintes instituições e pessoas, sem as quais o presente trabalho teria sido impossível ou ao menos mais trabalhoso:

À UFSC pela receptividade, acolhida e apoio técnico e administrativo;

À Prof. Janaína, pelo incentivo ao desenvolvimento deste trabalho;

Aos colegas, que sempre me acolheram com afeto e amizade;

Ao Dudu, pelas valiosas discussões e sugestões no decorrer do trabalho;

Ao querido amigo Ronaldo Baggio, pai do Lorenzo, pelo estímulo, amizade, carinho, críticas, sugestões e paciência nestes 2 anos.

REFERÊNCIAS

- [1] Silja J., “*Stunning Revelations*”(2006), In These Times, (Acessado em: 4 de julho de 2016)

Implementação física de um neurônio artificial eletrônico

Bianca Menon da Silva

Universidade Federal de Santa Catarina
Blumenau-SC, Brasil
bianca.menon@grad.ufsc.br

Christian Correa de Borba

Universidade Federal de Santa Catarina
Blumenau-SC, Brasil
christian.borba@grad.ufsc.br

Janaina Gonçalves Guimarães

Universidade Federal de Santa Catarina
Blumenau-SC, Brasil
janaina.guimaraes@ufsc.br

Resumo— Neste trabalho serão propostas duas implementações físicas eletrônicas de um neurônio artificial. A primeira será baseada em circuito integrado (CI), e a segunda em *Field Programmable Gate Array* (FPGA). O objetivo desse trabalho será a comparação da eficiência de cada implementação, através de uma aplicação que distingue bípede e quadrúpede.

Palavras-chaves- nerônio artificial; circuito integrado; FPGA.

I. INTRODUÇÃO

Um neurônio artificial é similar a um neurônio biológico, sendo os terminais de entradas comparados com dendritos, a função de ativação com o núcleo, os pesos com as sinapses nervosas e sua saída com o terminal axonal (*fig. 1*).

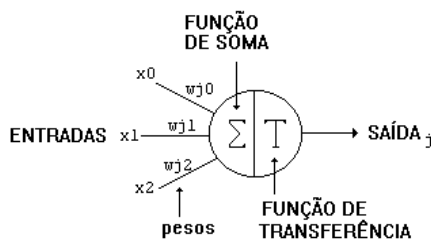


Figura 1: Modelo do Neurônio

Sistemas computacionais embasados em redes neurais possuem características exclusivas de um sistema biológico. Eles são capazes de receber várias entradas simultaneamente e distribuí-las de uma forma organizada. Assim, as informações contidas na rede, são compartilhadas por todas as suas unidades de processamento, o que evita a perda de informação caso uma parte da rede seja destruída.

Com o avanço dos estudos e do desenvolvimento da área, houve a necessidade da implementação em *hardware*, buscando o aumento da capacidade de processamento, desenvolvimentos da arquitetura neural e a redução de custos e tamanhos de equipamentos eletrônicos. Assim, nesse trabalho a rede será implementada no circuito integrado e no FPGA para obter a comparação de desempenho.

O presente trabalho foi realizado com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq – Brasil.

II. METODOLOGIA

Foram escolhidas duas tecnologias distintas para implementação e comparação da eficiência de *hardware* para um neurônio artificial, o FPGA e o circuito eletrônico. O modelo do neurônio usado foi implementado através da rede LIRA [1]. Esse neurônio possui seis entradas e duas saídas, sendo duas entradas de dados positivas, duas entradas de dados negativas, e duas entradas funcionam como *threshold*, responsáveis por setar o limiar de ativação do neurônio. A saída principal funciona para quando o neurônio estiver ativo, enquanto a saída secundária funciona como um *carry* para indicar que houve um *overflow* (*fig. 2*).

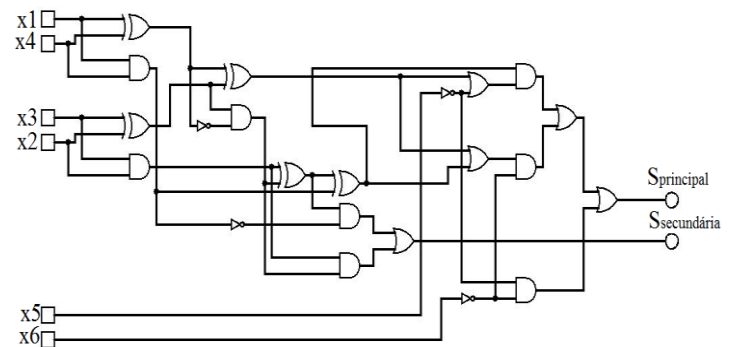


Figura 2: Neurônio da rede LIRA

Inicialmente a simulação para o entendimento do funcionamento do neurônio foi realizada no *software* Logsim. Com a implementação da aplicação escolhida, que diferencia bípedes e quadrúpedes[2], houve a necessidade da mudança de softwares para a realização da implementação física e possibilidade de testes na *protoboard* do circuito e no FPGA. O *software* utilizado para a montagem e extração de dados para o circuito foi o LTSpice (*fig.3*) e para o FPGA foi o Quartus(*fig. 4*).

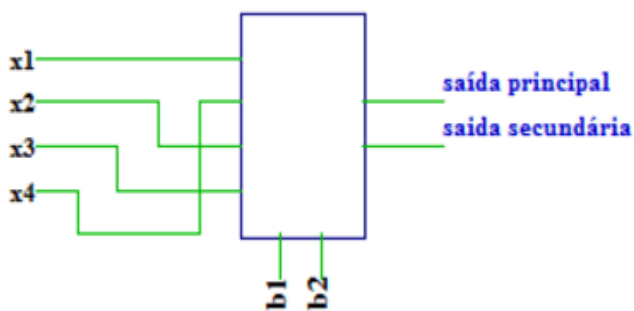


Figura 3: Rede neural LIRA para classificação de bípedes e quadrúpedes

Para a aplicação, foi preciso fazer ajustes dos vetores de treinamentos, pois o neurônio considerado para a aplicação variava entre -1 e 1 (tabela I), enquanto o neurônio baseado na LIRA aceita apenas entradas 0 e 1 (tabela II). Desse modo, os valores dos vetores de treinamento para o homem e para o gato, foram modificados para ajuste dos pesos.

TABELA I. Vetores de treinamento da rede original

	Vetor de treinamento				Resultado
Cão	1	-1	1	1	1
Gato	1	1	1	1	1
Cavalo	1	1	-1	1	1
Homem	-1	-1	-1	1	-1
Galinha	-1	1	-1	1	-1
Avestruz	1	-1	1	-1	-1

TABELA II. Vetores de treinamento ajustados

	Vetor de treinamento				Resultado
Cão	1	0	1	1	1
Gato	1	1	1	0	1
Cavalo	1	1	0	1	1
Homem	0	0	0	0	0
Galinha	0	1	0	1	0
Avestruz	1	0	1	0	0

Com os padrões de entrada ajustados, foi implementado o neurônio nas plataformas físicas, *protoboard* e FPGA. Foi testadas todas as combinações de entradas, e constatou-se que não houve erro, podendo assim serem comparados.

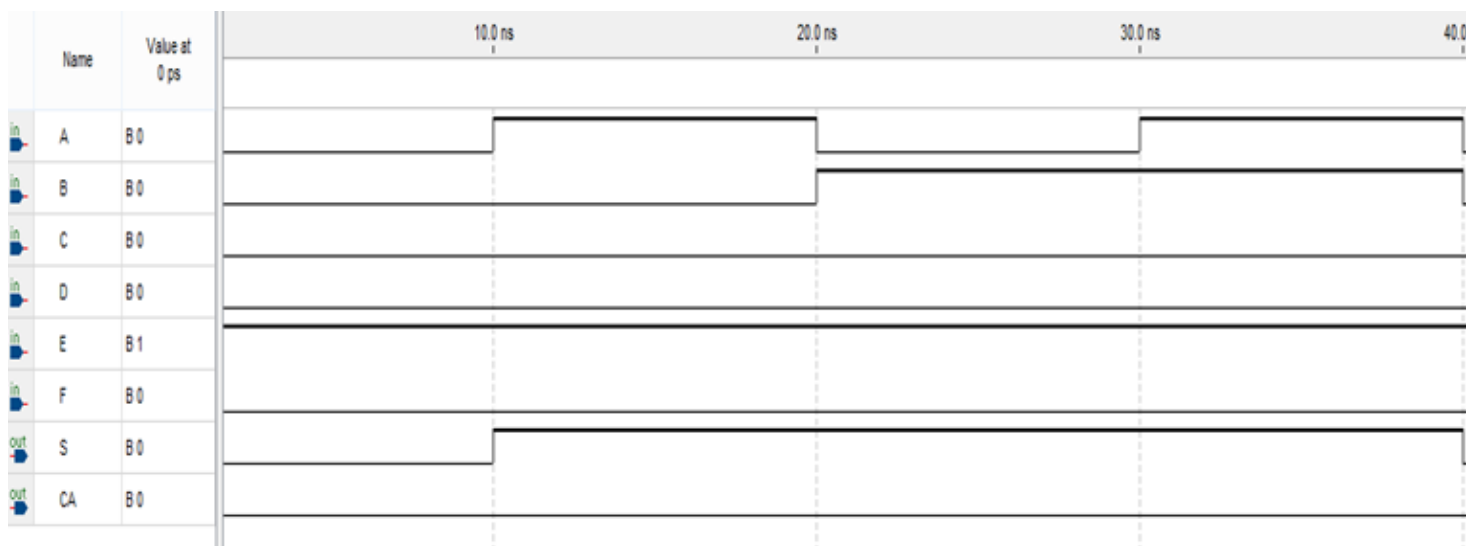


Figura 4: Simulação no Quartus

III. RESULTADOS

Após testes para todas as entradas fornecidas pela tabela II, pode-se obter a tomada de dados da potência e de atraso do tempo de resposta (tabela III). As imagens a seguir mostram as simulações para duas combinações de entradas para o

circuito e para o FPGA, respectivamente. A *fig. 5* e a *fig. 6* identificam o vetor [1,1,1,0] (tabela II) como quadrúpede, mantendo as saídas primárias em 1 e os *leds* acesos. Enquanto a *fig. 7* e a *fig. 8*, identificam o vetor [0,0,0,0] (tabela II) como bípede, mantendo sua saída primária em 0 e os *leds* apagados.

TABELA III. Comparação de desempenho entre o CI e o FPGA

	Atraso	Potência
Circuito	0,1 μ s	4 mW
FPGA	7 ps	20 mW

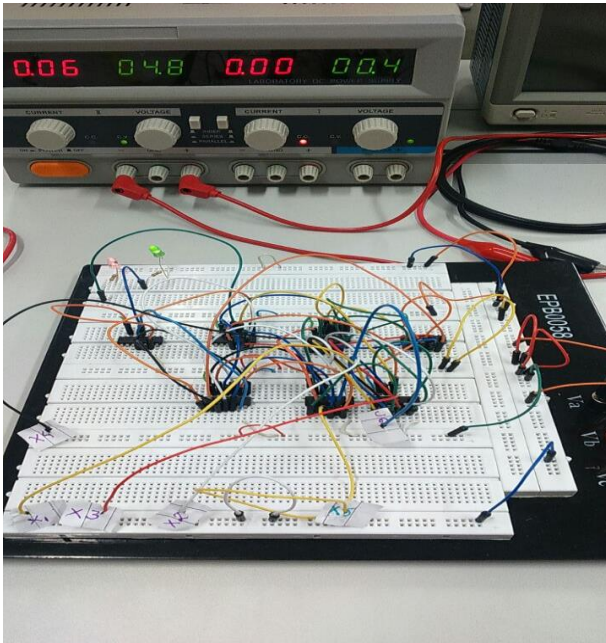


Figura 5: Saída para a identificação do vetor “gato” como quadrúpede em CI’s

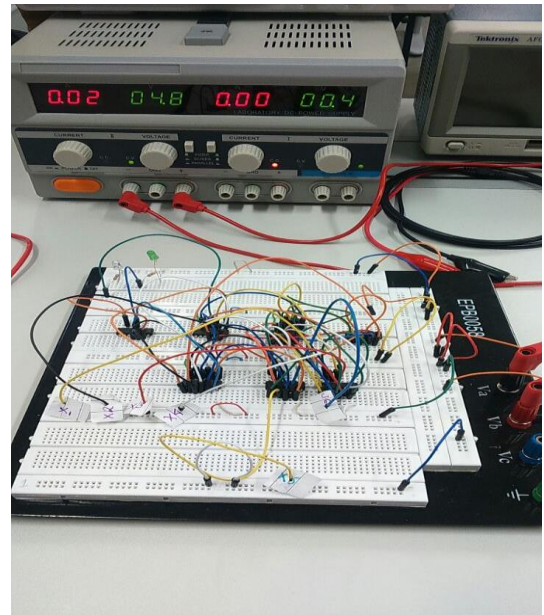


Figura 7: Simulação de saída para o padrão de entrada referente ao homem em CI’s.



Figura 8: Simulação do vetor de entrada referente ao homem em FPGA



Figura 6: Simulação de saída referente a entrada padrão para o gato em FPGA

IV. CONCLUSÃO

Através da análise dos resultados, obteve-se um melhor tempo de resposta para o FPGA. Todavia, o circuito dissipa uma potência menor. Dessa forma, embora o *delay* do circuito seja maior ao comparar-se com o FPGA, para um circuito pequeno, o tempo de resposta é considerado bom. Já em circuitos maiores, com o aumento de tamanho físico a potência dissipada pelo circuito e o atraso de resposta também aumentam, enquanto a do FPGA pode ser considerada

constante. Os dados obtidos no circuito foram esperados, no *datasheet* do CI o tempo de atraso é de 5 a 15 ns, para cada porta [4]. O *datasheet* do FPGA diz um atraso de 1 ns[5], sendo mais alta do que obtido, a análise do tempo de atraso neste trabalho foi preliminar, existem varias técnicas para caracterizar atrasos nos FPGA's, na continuação desse trabalho uma técnica será escolhida, mas cabe ressaltar que o valor medido esta compatível com os valores de leitura, que falam em 1 ns para caminhos de propagação do sinal mais complexos que o do neurônio.

V. REFERÊNCIAS

- [1] Vega, A.; Perez, J.L.; Development and Implementation of the LIRA Neural Classifier; WCECS 2010, October 20-22, 2010, San Francisco, USA.
- [2] Disponível em: <https://linguagensdeprogramacao.wordpress.com/2011/09/03/perceptron-usando-python/>. Acessado em Julho de 2016.
- [3] Tafner, M.A; Xerez, M; Rodrigues Filho, I.W. Redes neurais artificiais: introdução e principio de neurocomputação. Blumenau: Editora Furb,1995.
- [4] Disponível em: <https://www.fairchildsemi.com/datasheets/MM/MM74HC00.pdf> . Acessado em Julho de 2016.
- [5] Disponível em: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-iv/cyiv-53001.pdf. Acessado em Julho de 2016.

Problema de um Cruzamento Controlado: Modelagem e Análise por Redes de Petri

Brunno Vanelli

Universidade Federal de Santa Catarina
Email: brunno.v@grad.ufsc.br

Rômulo Pacher Júnior

Universidade Federal de Santa Catarina
Email: romulo.pacher.j@grad.ufsc.br

André Reuter

Universidade Federal de Santa Catarina
Email: a.reuter@grad.ufsc.br

Resumo—Um dos principais requisitos de qualquer programa ou processo industrial é a segurança do processo. Para garantir que o processo funcionará corretamente sem problemas ou interrupções utilizam-se simulações. Uma das ferramentas que podem ser usadas para modelagem de sistemas a eventos discretos são as redes de Petri.

O objetivo deste artigo é resolver o problema de um cruzamento controlado utilizando como ferramenta redes de Petri, implementar a lógica do controlador e simular os resultados para se certificar das propriedades requisitadas.

Keywords—Redes de Petri, TINA, Cruzamento Controlado

I. INTRODUÇÃO

O objetivo deste trabalho é utilizar redes de Petri para simular e controlar um cruzamento, sendo que o controlador tem visão geral do sistema e pode controlar as entradas e saídas dos veículos em cada seção do cruzamento.

A rede de Petri é um modelo matemático com representação gráfica utilizado em simulações de sistemas, entre os quais se destacam sistemas de manufatura, de comunicação, de transporte, de informação, logísticos e, de forma geral, todos os sistemas a eventos discretos [1].

Existem três elementos básicos em redes de Petri: lugares, transições e arcos. Um exemplo de Rede de Petri pode ser visto na Figura 1. Os elementos p_1 , p_2 e p_3 representam os lugares. Os lugares podem ter fichas, que representam os recursos de dado lugar. As transições, denotadas por a , b , c e d , são funções que alteram as fichas no sistema. Os arcos determinam quantas fichas são necessárias para sensibilizar cada transição.

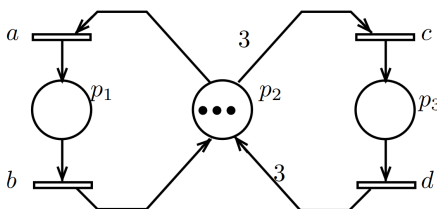


Figura 1. Rede de Petri [1].

Por exemplo, na Figura 1, para sensibilizar a transição a , uma ficha é retirada de p_2 e colocada em p_1 . Para a transição c ser sensibilizada, são necessárias três fichas em p_2 , pois o peso do arco é três. Se c for ativada, as três fichas em p_2 são

consumidas e uma ficha é depositada em p_3 , pois o peso do arco após c é um.

Para realizar a modelagem e simulação da Rede de Petri utilizou-se o software TINA [2] (*Time petri Net Analyzer*) disponibilizado pelo *Laboratoire d'analyse et d'architecture des systèmes*.

O artigo é estruturado em Descrição do problema, onde será descrito brevemente como funciona o cruzamento controlado e suas especificações, bem como as especificações do controlador, em Modelagem por Redes de Petri, descrevendo os passos realizados para modelagem do problema e a Análise da rede completa, que visa assegurar as propriedades desejadas da rede, como ausência de bloqueios.

II. DESCRIÇÃO DO PROBLEMA

O cruzamento é representado na Figura 2. Os carros, que vem dos cruzamentos V_i , ($i = 1, \dots, 4$) podem seguir reto, virar à direita ou à esquerda. Ao entrar no cruzamento, eles ocupam as posições correspondentes ao seu caminho, como indicado na Figura 2. As posições são denotadas por S_i , ($i = 1, \dots, 4$).

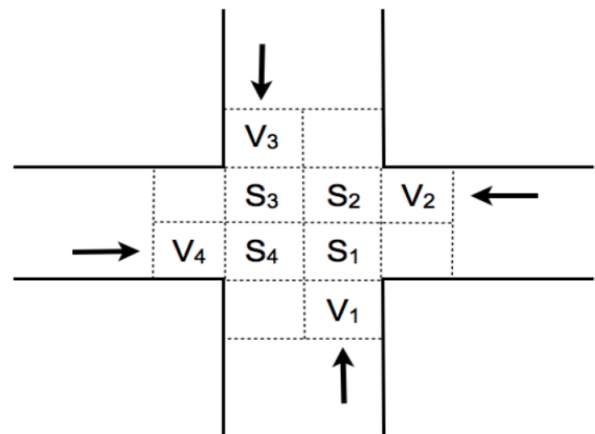


Figura 2. Cruzamento.

O sistema de controle desenvolvido para o controlador em redes de Petri deve garantir que nenhum “acidente” ocorra (dois veículos nunca ocupem a mesma posição) e que os veículos não possam ficar “presos” em nenhuma parte do processo. Um exemplo que resultaria em um congestionamento

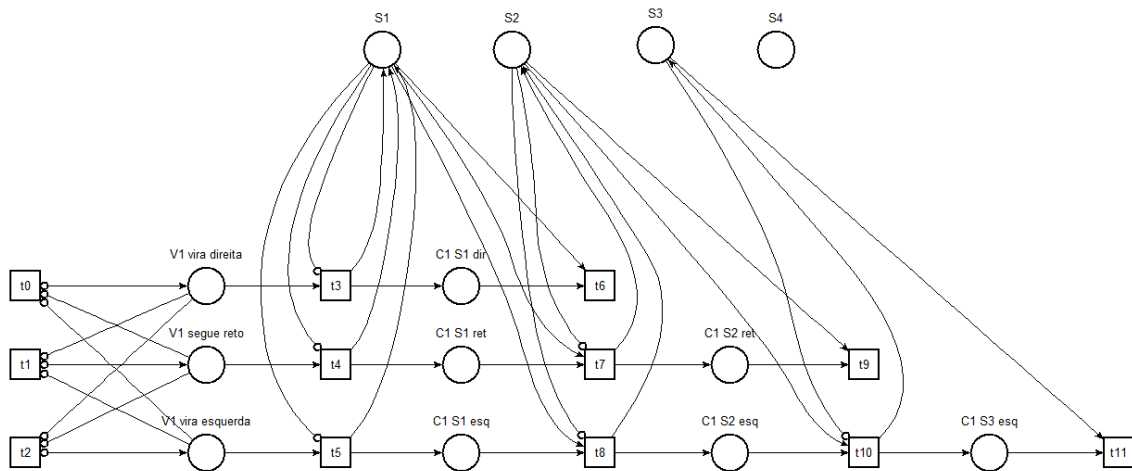


Figura 3. Modelagem da Via V1.

da rede é todos os carros seguirem reto na via e entrarem ao mesmo tempo nas seções respectivas. Como nesse caso as quatro posições no cruzamento estariam ocupado, nenhum veículo poderia se mover.

As especificações de funcionamento para o controlador são:

- Dois veículos não podem estar no mesmo tempo numa das 8 seções pré-definidas do cruzamento.
- Todo veículo no cruzamento (seções S_i) tem preferência sobre um veículo na via (seções V_i)

O modelo simplificado da posição dos carros ao entrar no cruzamento pode ser visto na Tabela I.

Tabela I
COMPORTAMENTO

Via de Origem	Ação	Sequência
V1	Vira a direita	S1
V1	Vira a esquerda	S1 > S2 > S3
V1	Segue reto	S1 > S2
V2	Vira a direita	S2
V2	Vira a esquerda	S2 > S3 > S4
V2	Segue reto	S2 > S3
V3	Vira a direita	S3
V3	Vira a esquerda	S3 > S4 > S1
V3	Segue reto	S3 > S4
V4	Vira a direita	S4
V4	Vira a esquerda	S4 > S1 > S2
V4	Segue reto	S4 > S1

III. MODELAGEM POR REDES DE PETRI

A modelagem do sistema completo pode ser obtido modelando separadamente os elementos e especificações do controlador. Alternativamente, pelo fato do problema ser simétrico, apenas uma das vias precisa ser modelada, e as outras respeitarão modelos idênticos.

A. Modelagem das Vias

A via pode ser modelada de forma bastante simples, de acordo com a Figura 4. A via possui dois estados: ocupada, quando não pode receber mais carros, ou disponível, quando o carro acabou de entrar no cruzamento.

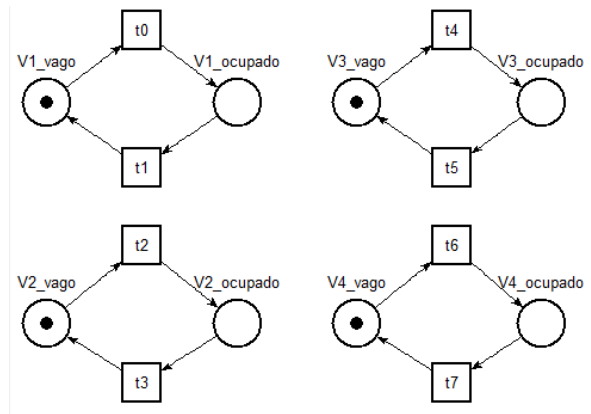


Figura 4. Modelagem das Vias.

B. Modelagem dos Caminhos

Ao entrar na via, o comportamento de cada carro está limitado ao que está ao seu redor (o rumo que os outros carros tomaram). Se for desconsiderada por enquanto essas limitações, pode-se simplificar o comportamento dos carros de acordo com a Tabela I.

No entanto, o sistema deve travar qualquer carro de avançar para uma posição ocupada. Isso pode ser feito criando um recurso compartilhado entre as quatro vias (S1, S2, S3 e S4). Para avançar, o controlador precisa checar se a S_i que o carro passará a ocupar está ocupada. O esquemático desse sistema pode ser observado na Figura 3.

O funcionamento é bastante simples. Ao entrar na via, o controlador aciona a transição que o carro escolher. Isso trava

qualquer outro carro de entrar na via, já que as transições t_0 , t_1 e t_2 observam os lugares de V_1 (vira direita, vira esquerda e segue reto). Daí em diante, ele já tem a sequência de ações que podem ser tomadas, e as transições são acionadas de acordo com a disponibilidade do recurso seguinte e com a vontade do controlador. A checagem do recurso é feita por observadores na Figura 3 para facilitar na leitura da Rede de Petri.

Ao final da sequência, a ficha é tirada do sistema (em t_6 , t_9 e t_{11}) para indicar que o carro saiu do cruzamento.

As outras vias tem modelagem idêntica, com exceção da ordem dos lugares que elas percorrem, como indicado na Tabela I.

A rede de Petri final será a união dos 4 modelos independentes. Os lugares que serão unidos, por serem compartilhados, são os locais S_1 , S_2 , S_3 e S_4 .

C. Identificando Problemas

Claramente a rede gerada ainda possui problemas. Um bastante fácil de identificar é quando todos os carros tentam seguir reto ao mesmo tempo. Desse modo, dadas as condições de todos partirem ao mesmo tempo, o sistema pode gerar um *deadlock*, pois nenhum dos 4 carros no cruzamento poderia avançar por estar sendo barrado pelo próximo. O mesmo ocorreria se todos tentassem virar à esquerda, mas não ocorre se algum deles for tentar virar a direita.

Uma maneira de corrigir esse problema é identificar carros que tem a possibilidade de cruzar pelo espaço que o carro ocuparia ao entrar no cruzamento. Isso ainda garante a prioridade dos carros no cruzamento pedido pelo controlador.

Portanto, para o carro em V_1 entrar em S_1 , a posição S_4 deve estar livre (dando prioridade a um carro vindo de V_4 que segue reto e a um carro vindo de V_3 que virou à esquerda). Para o carro em V_2 se mover para S_2 , a posição S_1 deve estar livre (dando prioridade a um carro vindo de V_1 que segue reto e a um carro vindo de V_4 que virou à esquerda). Para o carro em V_3 se mover para S_3 , a posição S_2 deve estar livre (dando prioridade a um carro vindo de V_2 que segue reto e a um carro vindo de V_1 que virou à esquerda). Finalmente, para um carro em V_4 se mover para S_4 , a posição S_3 deve estar livre (dando prioridade a um carro vindo de V_3 que segue reto e a um carro vindo de V_2 que virou à esquerda). A implementação por redes de Petri pode ser visto na Figura 5.

Note que esse método de controle não é ideal. Por exemplo: se um carro de V_4 virar à direita, mesmo assim bloqueará a entrada de um carro vindo de V_1 , mesmo que não interfiram nos caminhos uns dos outros.

D. Controle "ótimo"

Para realizar o controle ótimo, é necessário que as exceções indicadas em III-C sejam tratadas. Em outras palavras, deve-se definir um modelo de prioridades mais específico que trate melhor do acesso dos carros ao cruzamento.

Ao invés de a transição de V_i contabilizar apenas um lugar, independente do sentido do carro (ou seja, independentemente de ele interferir com o lugar que o carro saindo de V_i ocupará em seguida), criar quatro lugares que servirão como *buffers*,

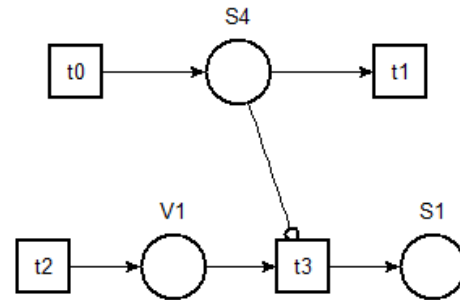


Figura 5. Modelo Prioridade Simples.

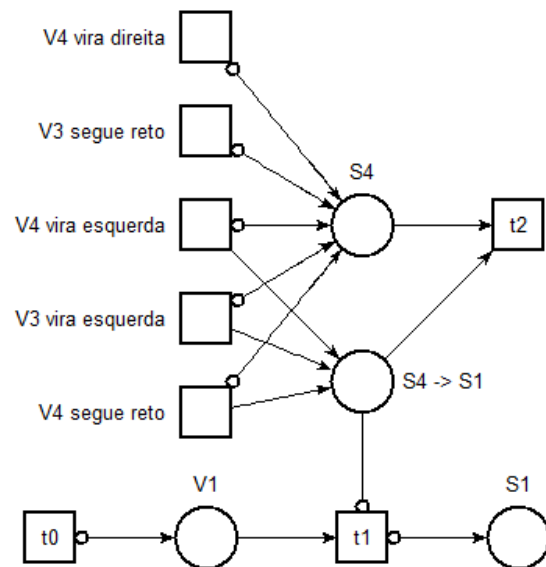


Figura 6. Modelo Prioridade Específico.

indicando que há um carro que pretende ocupar a frente da sua via em breve.

Em outras palavras, se há um carro querendo entrar por V_1 , o controlador checaria se há um carro em S_4 que pretende seguir para S_1 . Como as únicas opções de isso acontecer são um carro vir de V_4 e seguir reto, vir de V_4 e virar à esquerda ou vir de V_3 e virar a esquerda, essas sequências colocariam uma ficha no buffer $S_4 \rightarrow S_1$. Assim, o carro em V_1 seria impedido de avançar, dando preferência ao carro em S_4 para terminar seu trajeto. Ao passar para S_1 , o controlador automaticamente remove a ficha do buffer, possibilitando a continuação do sistema. Os outros buffers ($S_1 \rightarrow S_2$, $S_2 \rightarrow S_3$ e $S_3 \rightarrow S_4$) funcionam de forma simétrica.

A implementação desse modelo por redes de Petri pode ser visto na Figura 6.

E. Modelo Completo

O modelo completo é dado pela união dos Modelos dos Caminhos descrito em III-B com o Controle Ótimo descrito

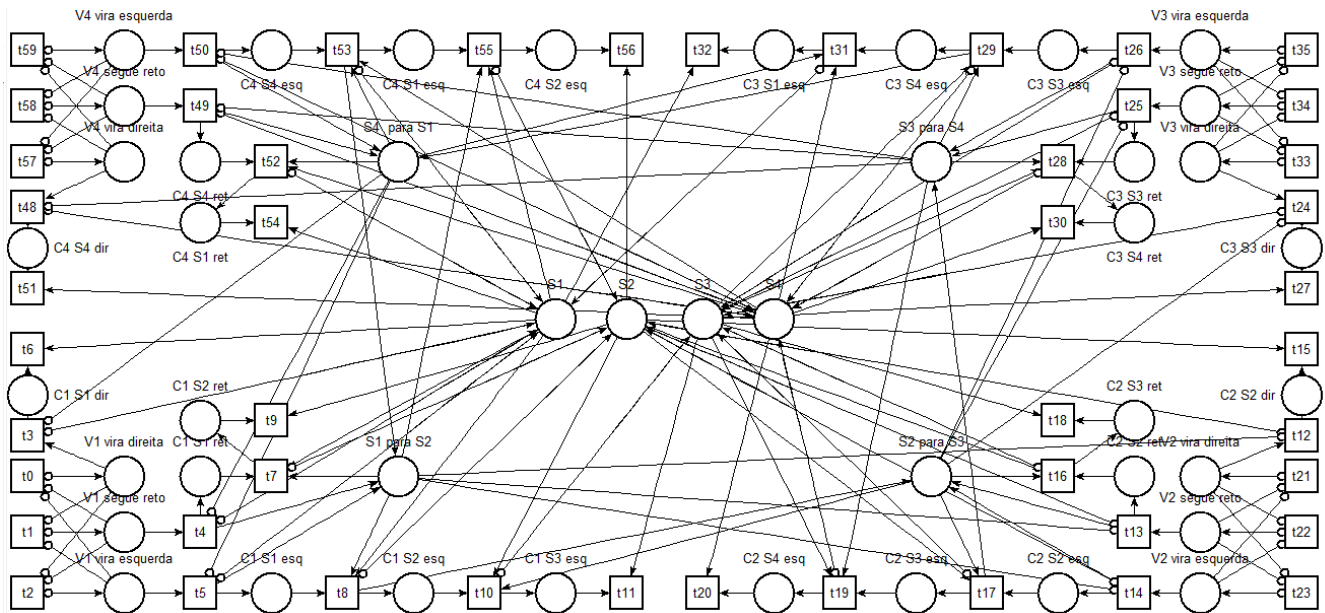


Figura 7. Modelo Completo.

em III-D, e pode ser visto na Figura 7. Note que como todos os recursos necessários para o bom funcionamento de cada via V_i individual (S_i , $i = 1, \dots, 4$ e buffers) são compartilhados, é automaticamente garantido um bom funcionamento do sistema se as partes individuais funcionarem corretamente.

IV. ANÁLISE DA REDE DE PETRI

A. Impossibilidade de Bloqueio

Deve-se analisar a condição de que não existe possibilidade de bloqueio no sistema. Como descrito em III-C, esse problema foi contornado utilizando as escolhas dos carros para evitar bloqueios. No entanto, não há como analisar a rede computacionalmente com a ferramenta TINA pela grande quantidade de estados do gráfico de marcações, e a análise é apenas teórica.

B. Cruzamento Vivo

Essa condição presume que todo veículo que entra no cruzamento sairá "um dia". No entanto, nenhuma condição de controle garante isso. Um carro poderia entrar no cruzamento e o controlador nunca acionar a condição para ele avançar.

No entanto, a rede garante que existe sempre um sequência de acionamentos que retire todos os carros do cruzamento devido ao fato de assegurar que nem todos os carros no cruzamento interfiram nas rotas uns dos outros, ou se certificando de que sempre haverá um caminho livre para um carro que precisa ocupar o lugar de um dos carros que querem entrar no cruzamento.

C. Um veículo por posição

Como sugeriu Newton, dois corpos não devem ocupar o mesmo lugar no espaço, o que significa que dois veículos

nunca podem estar na mesma seção. Isso é facilmente contornado com a condição imposta a cada veículo que ele só poderá avançar se a posição para a qual ele pretende avançar está livre. Com essa condição, é fácil ver que dois veículos nunca ocuparão o mesmo lugar.

D. Prioridade do Veículo no Cruzamento

Essa condição garante que sempre que um veículo está no cruzamento é prioritário sobre um veículo que está na via. Isso é garantido pela especificação feita em III-D, que garante que se um veículo pretende entrar no cruzamento, o controlador só irá permitir se a posição que ele pretende ocupar não interfere com a rota dos veículos já no cruzamento.

V. CONCLUSÃO

Esse artigo apresenta uma visão geral da implementação de um controlador para um cruzamento controlado utilizando Redes de Petri. Obviamente que algumas condições são limitadas para que o problema não fosse extremamente complexo. Não foi possível obter computacionalmente a confirmação da ausência de bloqueio devido a grande quantidade de estados gerados, no entanto a própria forma com que a rede foi construída elimina essa possibilidade. No geral, todas as condições foram atendidas e o resultado obtido foi satisfatório.

Na prática, um cruzamento real possui algumas variáveis a mais e necessita além do controle básico, a otimização do tráfego, por exemplo, que torna essa área uma das mais complexas no âmbito do controle de sistemas urbanos. Ainda é preciso pensar na segurança e na viabilidade do controle.

REFERÊNCIAS

- [1] R. Cardoso, Janette e Valette, *Redes de Petri*. Editora da UFSC, 1997.
- [2] Tina - time petri net analyzer. [Online]. Available: <http://projects.laas.fr/tina>

Sistema Controlado de Trens

Guilherme Moser Manerichi

Universidade Federal de Santa Catarina
guilherme.manerichi@grad.ufsc.br

Juliano Emir Nunes Masson

Universidade Federal de Santa Catarina
juliano.masson@grad.ufsc.br

Stephan August R. D. dos Santos

Universidade Federal de Santa Catarina
stephan.august@grad.ufsc.br

Resumo — A utilização de Redes de Petri para modelagem de sistemas é comum na engenharia por se tratar de um método intuitivo. No caso a modelagem de um sistema de trens foi inteiramente feito através da Rede de Petri. Neste problema três trens devem circular por um mesmo trajeto sem que existam colisões, dividindo estações para a carga e descarga de mercadorias. Foram implementadas especificações para evitar colisões e foi realizada a sincronização de cargas e descargas. Através de testes e análises na rede, foi apontado que não há possibilidade de colisões entre os trens, e a carga e a descarga dos trens se mantém sincronizada em todos os instantes.

Palavras-Chave — Redes de Petri; Modelagem; Sincronização de linhas ferroviárias

I. INTRODUÇÃO

Redes de Petri é uma técnica de especificação formal bem estabelecida e largamente difundida e adequada para a modelagem de sistemas que tenham atividades paralelas, concorrentes, assíncronas e não-determinísticas. É uma técnica muito útil para a modelagem e análise de sistemas, ajudando na avaliação de desempenho até o diagnóstico de falhas, ela também possibilita uma representação matemática do sistema [1].

Existem três elementos básicos na rede de Petri [2]:

- Lugar: pode ser interpretado como uma condição, um estado parcial uma espera, um procedimento, um conjunto de recursos, etc. É um elemento passivo na rede.
- Transição: é associado a um evento que ocorre no sistema, como iniciar alguma operação. É um elemento ativo na rede.
- Ficha: é um indicador significando que a condição associada ao lugar é verificada. Pode representar um trem parado em uma estação, por exemplo.

Utilizando desta ferramenta propomos a modelagem de um problema de controle de trens.

Neste problema três trens devem transportar mercadorias entre as estações localizadas em dois trilhos, um de ida e outro de volta. Os trens têm de dividir os mesmos trilhos, sendo que de forma alguma possam ocupar a mesma seção de um mesmo trilho, pois haveria um acidente. Além disso os trens devem seguir sete especificações de forma que não ocorra nenhum bloqueio.

Para que seja melhor o entendimento, a Fig. 1. representa a linha ferroviária:

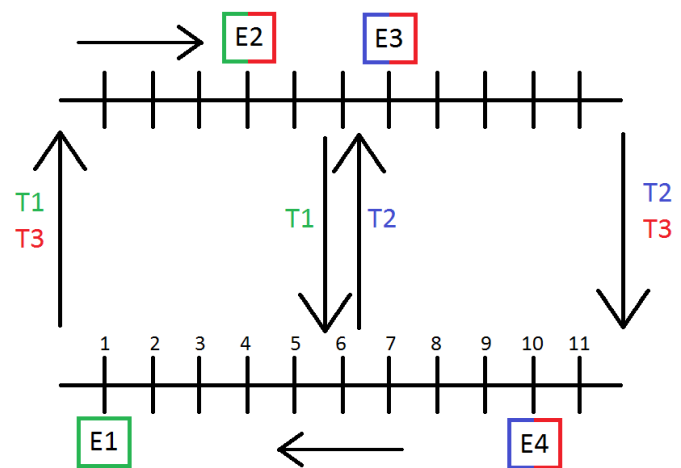


Fig. 1.

Representação do problema De 1 até 11 são as seções dos trilhos. T1, T2 e T3 são os trens e os quadrados E1, E2, E3 e E4 as estações de carga e descarga. As estações posicionadas no trilho superior são de carga e as posicionadas no trilho inferior são de descarga. Observando a Fig. 1. podemos perceber que o trem 1 carrega a mercadoria na estação E2 (Seção 4), e descarrega na estação E1 (Seção 1), o trem 2 carrega mercadoria na estação E3 (Seção 7), e descarrega na estação E4 (Seção 10), por fim, o trem 3 carrega mercadoria tanto na estação E2, quanto na estação E3, e descarrega na estação E4.

Para solucionar este problema em sua totalidade, devemos cumprir certas propriedades:

- P1: o bloqueio do sistema é proibido.
- P2: dois trens não podem estar simultaneamente na mesma seção.
- P3: cada trem poderá sempre percorrer o trajeto para o qual foi programado.
- P4: sempre que houver carga ou descarga, o trem estará parado na estação prevista para isto.
- P5: todo trem carregado será descarregado.

- P6: sempre que T1 estiver descarregando T2 estará carregando.
- P7: cada trem carrega e descarrega mercadorias, infinitas muitas vezes.

O trabalho será organizado da seguinte forma, a seção II mostrará o modelo proposto para a resolução do problema, a seção III mostrará as análises realizadas sobre a rede, a seção IV mostrará as simplificações do modelo completo para facilitar a compreensão e análise pelo *software* TINA[3] e por último na seção V uma conclusão.

II. MODELO PROPOSTO

O modelo desenvolvido partiu da modelagem da coordenação entres os três trens para que não ocorra nenhuma colisão. Um sistema de fichas foi desenvolvido para que o trem possa se deslocar para a próxima seção, conforme Fig. 2. Desta forma ele deve esperar que a ficha daquela seção esteja disponível. Essa ficha é devolvida no momento que o trem avança para a próxima estação.

Através da descrição do problema na seção I, podemos perceber que o modelo pode ser simplificado pelo fato de que o trem 1 e o trem 2 só terão de sincronizar na seção 6 no restante elas não terão problemas de colisão, sendo assim a seção 6 será a única com risco de três trens colidirem, todas as outras seções terão apenas dois trens com risco de colisão.

Na seção 6 o modelo exige um certo grau de atenção, pois é nele que os três trens terão a maior chance de conflito, visto que é nela que o trem 1 e o trem 2 mudam o sentido. Desta forma foi utilizado um estado com uma única ficha de recurso para o controle. O trem que necessitar utilizar a seção 6 pegará a ficha de S6 e a utilizará, após a utilização o trem devolve a ficha para S6, esse mecanismo pode ser observado na Fig. 3.

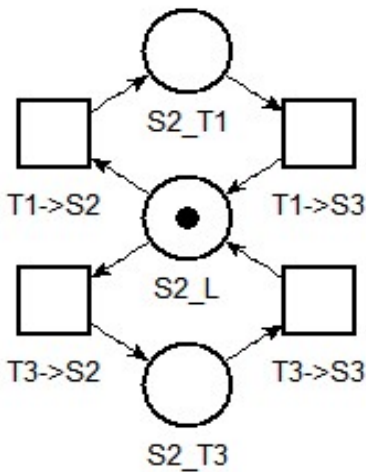


Fig. 2. Exemplo do sistema de fichas para as estações

No restante das seções o sistema é contínuo, sempre seguindo o sistema de ficha. A partir deste modelo da locomoção, modelamos as paradas dos trens nas estações de

carga e descarga para atender as especificações do sistema que estão na seção I.

Para o carregamento e descarregamento não chegamos a criar lugares específicos para as estações, já que o trem é obrigado a carregar ou descarregar todas as vezes que passa pela seção que está a estação. O único trem que foi feito o controle direto do carregamento e descarregamento foi o trem 3, já que o carregamento deve alternar entre as estações 2 e 3.

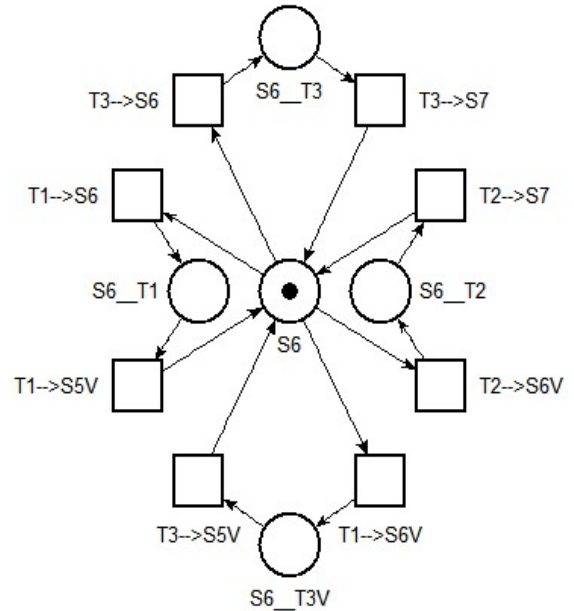


Fig. 3. Sistema de fichas para a seção 6

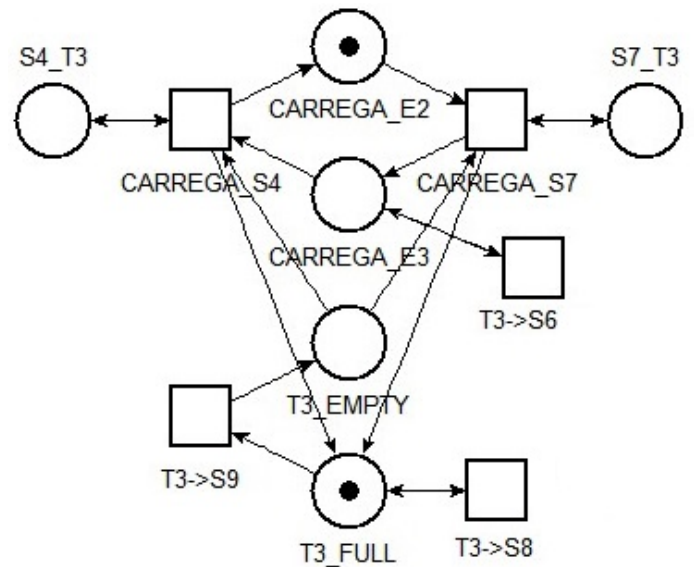


Fig. 4. Controle de carga e descarga do trem 3

O carregamento e descarregamento do trem 3 foi modelado da seguinte forma: temos dois lugares que, quando possuem ficha, indicam qual será a próxima estação que deve carregar o trem 3, estes lugares são: CARREGA_E2 e CARREGA_E3. Caso o trem 3 passe pela estação 2 (Seção 4) e tenha alguma

ficha em *CARREGA_E3*, ele passará direto, já que o carregamento deve ser feito na estação 3.

No caso em que ele carrega na estação 2, temos que criar um lugar de controle: *T3_FULL*, que indica para a estação 3 que o trem 3 já está cheio e portanto não deve ser carregado. Temos também o lugar de controle *T3_EMPTY* que indica quando o trem está vazio, variável necessária para fazer com que a transição *CARREGA_S7* não seja erroneamente disparada. Para o descarregamento estamos retirando a ficha de *T3_FULL* e adicionando uma ficha em *T3_EMPTY*, estas modificações ocorrem assim que ele sai da estação 4 (seção 10). Podemos observar este controle na Fig. 4.

III. ESTUDO DA REDE E ESPECIFICAÇÕES ATENDIDAS

Com a linguagem LTL[4] através do simulador TINA[3], foi verificado a vivacidade da rede, com o comando “< dead”. Este comando significa eventualmente morto e verifica se há alguma situação onde a rede apresentará a não vivacidade, porém o comando retornou falso, significando a não existência de *deadlocks*.

P2 é atendido por conta do sistema de fichas para que um trem ocupe a próxima seção, evitando que dois trens ocupem a mesma seção já que o trem pega a ficha e só devolve quando passar para a seção posterior a esta. P3 também é atendido pois ele poderá circular dentro das seções delimitadas pelo sistema.

Para atender as especificações P4 e P5, o sistema de carga e descarga foi criado para ser acionado, via as transições de entrada nas seções com estações, com exceção das estações E2 e E3 para o trem 3, desse modo os trens sempre são carregados ou descarregados em suas respectivas estações.

E para resolver o problema do carregamento e descarregamento síncrono do trem 1 e 2, adicionamos um lugar para cada trem. Estes lugares indicam quando os trens estão na posição de carregamento, no caso do trem 2 (*T2_S7*), e descarregamento, no caso do trem 1 (*T1_S1*). A saída dos trens só é permitida quando ambos os trens estão na posição correta.

Após a implementação deste controle, percebemos que algum tempo depois da inicialização da rede o trem 3 poderia ficar na frente do trem 1, causando assim um bloqueio na rede. Pois caso o trem 3 passe na frente do trem 1, este ficaria atrás do trem 2, que está na seção 7 esperando o trem 1 chegar na seção 1, mas o trem 1 não pode chegar até a seção 1 pois está atrás do trem 3, que está bloqueado pelo trem 2.

Para resolver este problema, criamos um tipo de sinaleira que só permite a passagem do trem 2 para a seção 7 se a variável de controle, que prenderia o trem 2 na seção 7 estiver com uma ficha, caso contrário ele tem que esperar o trem 1 gerar esta ficha, garantindo assim que o trem 3 nunca fique preso na seção 6 atrás do trem 2. Sendo assim a especificação P6 é atendida.

Com o modelo implementado, os trens carregam e descarregam toda vez que chegam às respectivas estações referentes a cada um e não apresenta bloqueios, sendo assim, a especificação P7 também é atendida.

IV. SIMPLIFICAÇÕES

Em uma modelagem inicial, que pode ser observada na Fig. 6., haviam muitas transições e estados, o que impedia o simulador TINA[3] de fazer a maior parte das análises, além do que o grafo de acessibilidade tinha 5622 estados e 14686 transições. Para contornarmos isso foram necessárias simplificações sem perda do foco da modelagem.

Para isso foram unidas algumas seções dos trilhos, restando apenas as seções: 4, 6 e 7 no sentido crescente, e 10, 7, 6, 2 e 1 no sentido decrescente, a seção 6 não foi alterada. As transições de acionamento de carga e descarga continuaram a ser as referentes a cada estação, visto que as seções com estação foram mantidas como pode ser visto na Fig. 5.

Essa simplificação diminuiu consideravelmente o número de estados do modelo, de 71 estados para 29 estados, assim como no número de transições que de 46 foi reduzido para 18. Obtivemos uma redução considerável de estados e transições, porém não o suficiente para tornar o grafo de acessibilidade analisável, agora com 288 estados e 549 transições.

V. CONCLUSÕES

Podemos perceber que a modelagem conseguiu suprir todas as especificações impostas pelo problema, mesmo existindo alguns problemas com a análise do modelo completo, por conta de seu elevado número de transições e lugares. O modelo completo pode ser observado na Fig. 6. Para resolver estes problemas encontrados na análise do modelo completo foi criado o modelo simplificado.

A partir da Fig. 5. podemos ver que na versão simplificada houve a perda de controle seção a seção, mas esse controle, segundo o enunciado do problema, não tem grande importância no resultado final visto que só é necessário controlar a sincronia dos trens, o sistema de carga e descarga e garantir que não haja colisões nem bloqueio. Sendo assim o modelo simplificado já se fez suficiente para completarmos todas as especificações necessárias para o sucesso da modelagem, com consideravelmente menos lugares e transições.

REFERÊNCIAS

- [1] Maciel, Paulo Romero Martins. Introdução as Redes de Petri e Aplicações. Editora UNICAMP, 1996.
- [2] R. Cardoso, Janette e Valette, Redes de Petri. Editora da UFSC, 1997.
- [3] B. Berthomieu, P.-O. Ribet, F. Vernadat, The tool TINA -- Construction of Abstract State Spaces for Petri Nets and Time Petri Nets, International Journal of Production Research, Vol. 42, No 14, July
- [4] E. M. Clarke, Orna Grumberg, Doron Peled. Model Checking. MIT Press, 1999.

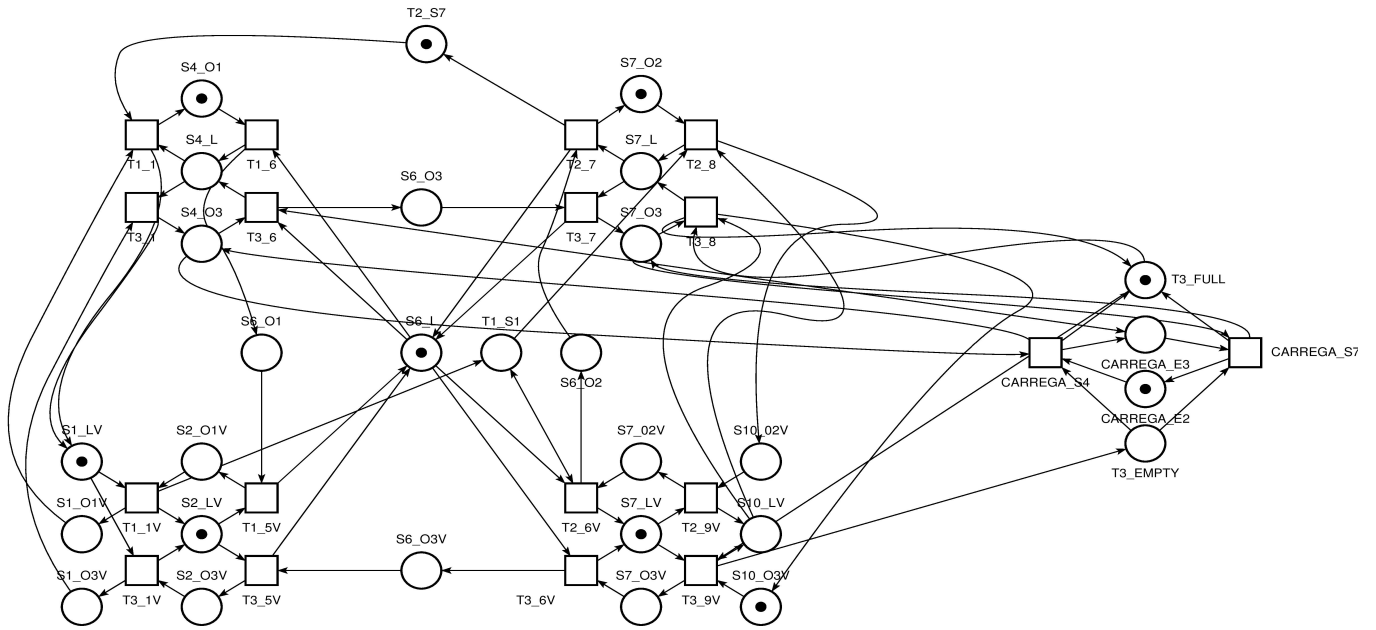


Fig. 5. Modelo Simplificado

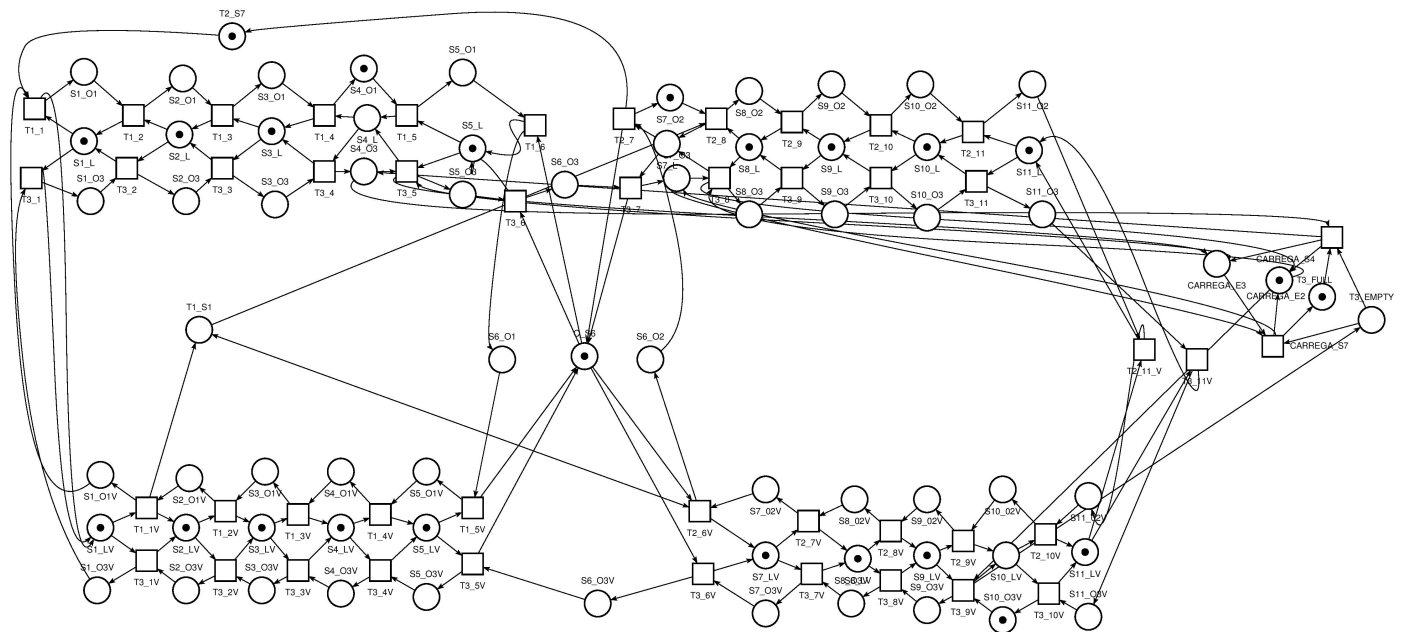


Fig. 6. Modelo Completo

Modelagem do Problema de Sincronização “Passeio dos Filósofos” Através da Rede de Petri

Caique Sanches
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina –
Campus Blumenau
Blumenau, Brasil
caiue.sanches@grad.ufsc.br

Ciro Outa Umezawa
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina –
Campus Blumenau
Blumenau, Brasil
ciro_umezawa@hotmail.com

Pedro Arthur S. P. Cogliatti
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina –
Campus Blumenau
Blumenau, Brasil
pedrosapeixoto@gmail.com

Resumo — Este artigo tem por fim demonstrar a resolução de problemas de sincronização utilizando a lógica das Redes de Petri através de softwares de modelagem TINA. As Redes de Petri são utilizadas para analisar sistemas a eventos discretos, muito utilizados na engenharia de controle e automação. Este artigo foca somente em um problema específico chamado de “Passeio dos Filósofos”.

Palavras-chave — Redes de Petri; modelagem; problema de sincronização; Tina, Passeio dos Filósofos.)

I. INTRODUÇÃO

O uso de softwares de modelagens, independente da linguagem e lógica, são fundamentais para que o aluno em seu meio acadêmico possa simular situações reais e buscar as melhores soluções. Para o aluno de engenharia de Controle e Automação, simular projetos industriais faz parte fundamental da formação deste profissional, visto que este, provavelmente, vai atuar diretamente em ambiente fabril. Vale ressaltar que nem sempre é viável reproduzir uma simulação feita em um laboratório para um ambiente de fato, físico. Uma das ferramentas muito utilizada para simulação são as Redes de Petri.

Este artigo vai abordar sobre Redes de Petri na seção II. A descrição do problema “Passeio dos Filósofos” na seção III. Em seguida será apresentado a resolução do problema na seção IV e os resultados obtidos na seção V. Finalizamos com a seção VI, com uma breve amostragem dos dados encontrados do problema.

II. REDES DE PETRI

Uma modelagem é a relação de equações matemáticas podendo ser representadas de forma gráfica. Sendo dirigidas por dogmas bem instruídos por leis de princípios físicos ou criadas por que as equações se encaixem na abstração.

Os modelos podem variar de analíticos ou discretos. O de tipos analíticos utiliza equações matemáticas para prever determinadas situações, em outras palavras estudando o sistema matemático aplicado. O modelo discreto também utiliza de um sistema de equações matemáticas, porém podem ser modeladas para a simulação possibilitando o estudo de seu comportamento perante o sistema aplicado.

Existindo inúmeras linguagens de programação para tal fim, na automação industrial é utilizado as Redes de Petri pois tem características específicas, como:

- análise de processos;
- controle de processos;
- automação em campo geral;
- banco de dados;
- circuitos integrados;
- sistemas distribuídos;
- sistema de produção;

A maneira mais básica de representar a rede de Petri é por dois componentes, um que passe a ser o ativo denominado por transição (graficamente representando por uma barra ou quadrado) que representa as ações, e o passivo denominado lugar (graficamente representando por um círculo) que representa as variáveis de estado do sistema.

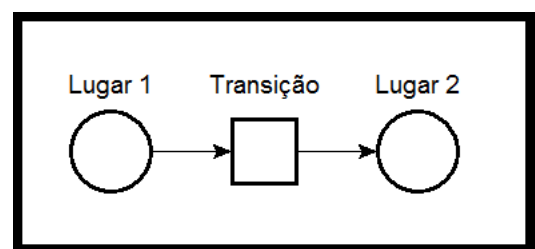


Figura 1. Simulação em Rede de Petri.

Algumas propriedades das Redes de Petri são:

- **Reiniciabilidade:** se existir uma marcação base, acessível de qualquer outra marcação do conjunto de alcançabilidade da rede.
- **Vivacidade:** é dita viva se for possível executar todas as suas ações a partir de qualquer uma das marcações alcançáveis da rede. Esta propriedade é muito importante na análise de sistemas digitais, uma vez que, quando verificada, indica a inexistência de bloqueios no sistema modelado.

- Limitação: Petri é limitada se em cada lugar da rede, o número total de marcas nunca exceder a um inteiro k . Neste caso a rede é dita k -limitada.

III. DESCRIÇÃO DO PROBLEMA

O problema do Passeio dos Filósofos é mais um problema clássico de sincronização multi-processo. Além dele, temos também o mais conhecido “jantar dos filósofos”.

A diferença entre o jantar dos filósofos é que cada processo compete repetidamente por uma única seção crítica. Já no caso do passeio dos filósofos, compete por vários recursos em momentos diferentes, sendo que estes são solicitados seguindo uma ordem específica.

Esses nomes são utilizados como analogia para melhor abstração do problema, porém a solução em si pode ser aplicada para qualquer situação que envolva este tipo de problemas entre processos e recursos.

IV. RESOLUÇÃO DO PROBLEMA

O problema pode ser visto como uma rótula dividida em recursos, na qual processos entram de várias direções e saem também em várias direções. Neste problema, utilizaremos 8 recursos e processos vindo e indo para 4 direções.

Esses processos e recursos devem satisfazer uma série de propriedades. Os processos devem acessar uma sequência ordenada de recursos, podendo iniciar em recursos diferentes. Além disso, deve-se utilizar a propriedade de exclusão mútua, isto é, cada recurso é acessado por um único processo ao mesmo tempo. A propriedade de justiça também terá que ser utilizado. Então cada recurso solicitado será sempre atribuído em algum momento futuro.

Além da utilização dessas propriedades, o problema deve contemplar e respeitar algumas condições como:

- Um processo pode solicitar um recurso somente se possuir o recurso anterior ou se não possuir qualquer recurso.
- Liberar um recurso que ele possui e nenhum processo voltará a solicitá-lo imediatamente.
- Se cada processo obtém cada recurso que solicita, ele liberará sempre qualquer recurso que lhe pertence.

Algumas situações que devem ser evitadas na modelagem do problema são:

- Gridlock.
- Cada recurso possa pertencer a um processo.
- Cada processo possa obter o novo recurso.
- Nenhum processo libera seu recurso atual.

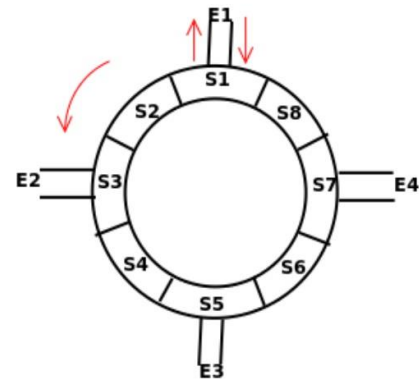


Figura 2. Analogia com rótula.

V. MODELAGEM

Na modelagem são definidos 8 recursos que podem ser vistos como lugares pelos quais os filósofos (processos) realizaram o seu passeio. Foram enumerados de 1 a 8, dos quais apenas nos lugares 1, 3, 5 e 7 é possível a entrada e saída dos filósofos.

Pela definição do problema, os filósofos devem seguir uma sequência, ou seja, o mesmo caminho, por isso foi definido que sempre seguirão na direção anti-horária.

Cada recurso possui um estado adicional para representar se encontra livre ou ocupado. Se ele está livre e o anterior ocupado, então o filósofo poderá avançar para esse novo estado. Da mesma maneira que se um filósofo deseja ingressar no PASSEIO, o estado no qual entrará, deve estar desocupado.

Devido às especificações de que pode haver apenas um filósofo por lugar, o número máximo de processos simultâneos durante o problema é de 7 pois se houverem 8 ao mesmo tempo, os filósofos não poderão continuar no passeio e um dos que se encontram na saída deverão sair. O exemplo pode ser observado na figura 5, mostrada em resultados.

VI. RESULTADOS

A modelagem, feita do software TINA, obtida a partir do problema pode ser observado na figura 3.

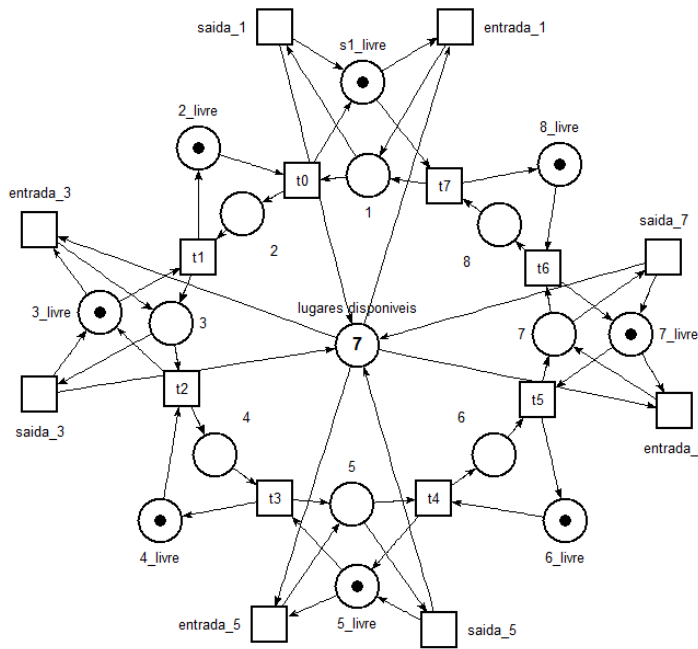


Figura 3. Modelagem.

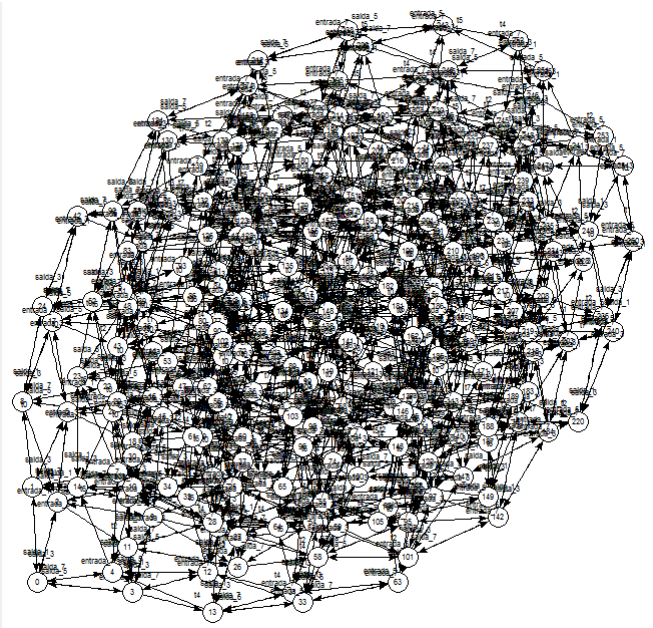


Figura 4. Grafo de marcações.

Com esta modelagem foi feita uma análise de suas propriedades.

O número total de lugares e transições foi de 17 e 16 respectivamente. Os lugares que formam um círculo interno representam quando as respectivas posições do passeio estão ocupadas e os do círculo externo, quando estão livres. O lugar auxiliar no centro da modelagem é utilizado para limitar o número de filósofos que ingressam no passeio pois, se houver 8 filósofos dentro, eles não conseguiriam avançar e não seguiria as especificações do problema.

Devido a esse lugar auxiliar a rede é 7-limitada pois é o número máximo de fichas que um lugar dentro da rede pode ter. Se não houvesse esse lugar, a rede seria binária já que apenas representaria se os lugares estão ocupados ou não.

Com a análise no programa TINA foi verificada que a rede não possui deadlocks, ou seja, não é bloqueada em nenhum momento. Também que a rede possui 255 estados diferentes.

A rede é dita reiniciável já que a partir de qualquer marcação é possível voltar à marcação inicial que seria quando nenhum filósofo está realizando o passeio, ou seja que independente de quantos estejam dentro (seguindo a limitação) e das suas posições, todos podem sair em algum momento.

Também pode ser descrita como uma rede viva, devido a que a partir de qualquer marcação em que a rede se encontra é possível realizar uma sequência de transições que, no final, sejam executadas todas suas ações possíveis. Esta propriedade ajuda a verificar que a rede não tem bloqueios. Se houvesse um bloqueio no meio da rede, não seria possível acessar as seguintes transições.

Quando trata-se de uma rede limitada, é possível realizar um grafo de marcações utilizando o programa TINA. Foi obtido o grafo mostrado na figura 4.

O grafo de marcações obtido possui as 255 marcações possíveis da rede e transições para alcançar os respectivos estados.

Como a rede possui muitos estados, não foi possível obter um grafo de marcações que pudesse ser observado com clareza.

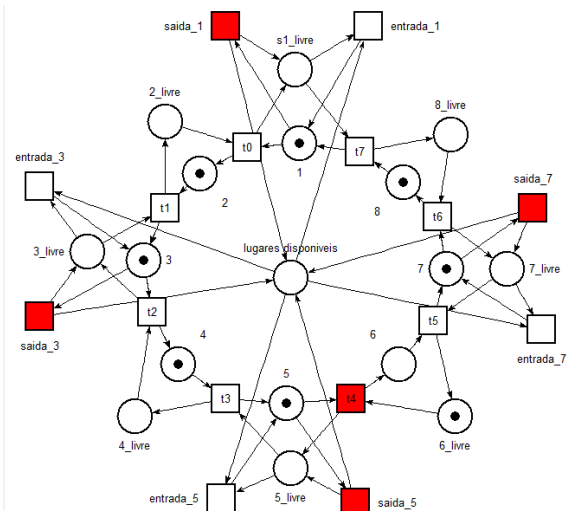


Figura 5. Caso especial da modelagem.

A figura 5 mostra uma marcação especial em que todas as 7 vagas limitadas são utilizadas. Assim pode ser observado que ainda há uma transição disponível (mostrada em vermelho) para que um filósofo siga percorrendo seu caminho e com isso irá liberar espaço para o seguinte e sucessivamente.

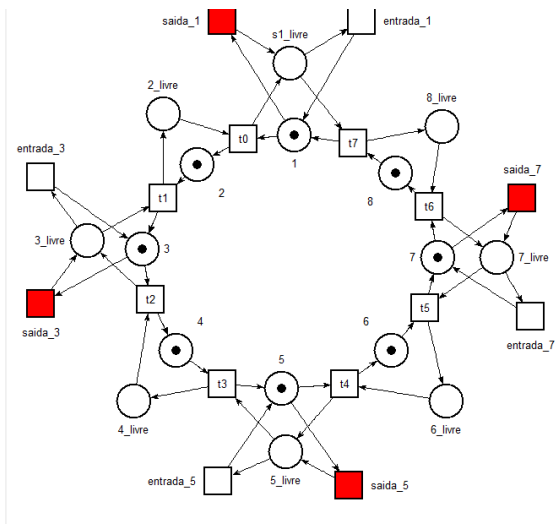


Figura 6. Modelagem sem limitação de vagas.

Na figura 6 é mostrado uma modelagem incorreta que seria sem a limitação das vagas. É possível observado que com 8 filósofos dentro do passeio, nenhum deles poderia mais seguir seu caminho e isto cria um bloqueio no problema.

VII. CONCLUSÃO

Apesar do desenvolvimento final parecer um pouco curto e simples, a modelagem e os raciocínios para se chegar ao modelo final passou por uma série de dificuldades.

Estas dificuldades se baseiam em que o problema “Passeio dos Filósofos” não é um problema de sincronismo comum, no sentido de se achar facilmente, no que resultou uma falta de referências bibliográficas para se basear. Em função disso, o raciocínio lógico foi todo montado somente em cima da lógica das redes de Petri e suas referências bibliográficas. Além da complexidade que o problema trazia, várias propriedades e

condições foram exigidas pelo problema e tiveram que ser obedecidas.

O artigo contribuiu para o crescimento dos autores no conteúdo abordado por este projeto, podendo demonstrar que sistema muito complexos só podem ser analisados e estudados se houver uma modelagem apropriada. Além do raciocínio lógico desenvolvido, o conhecimento de Redes de Petri aplicado a um problema que remete o mundo real contribuiu para o crescimento acadêmico e, consequentemente, profissional dos autores.

Para trabalhos futuros, a sugestão seria modelar este mesmo problema em linguagens e métodos diferentes para comparativos de desempenho e precisão de dados, para em fim, criar um modelo físico para ensaios.

VIII. REFERÊNCIAS

- [1] S.Baehni, R. Baldoni, R. Guerraoui, B. Pochon, “The Driving Philosophers”, Distributed Programming Laboratory, EPFL, Switzerland 2 Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza,” Italy.
<http://www.dcce.ibilce.unesp.br/~norian/cursos/tds/ApostilaRdP-PA.pdf>
Último acesso em 01/07/2016
- [2] N. Marranghello, “Redes de Petri: Propriedades e Análise”, DCCE/IBILCE/UNESP, Agosto 2005
http://www.dca.ufrn.br/~affonso/FTP/DCA409/redes_de_petri.pdf
Último acesso em 01/07/2016
- [3] C. R. L. Francês, “Introdução às Redes de Petri”, Laboratório de Computação Aplicada – LACA Universidade Federal do Pará - UFPA Agosto de 2003,
http://www.dca.ufrn.br/~affonso/FTP/DCA409/redes_de_petri.pdf
Último acesso em 01/07/2016

Misturador Automático

André Luiz Granemann, Christian de Borba e Martin H. Anschau

*Engenharia de Controle e Automação - UFSC Blumenau
Blumenau, Brasil*

Resumo – Neste trabalho é apresentado um processo controlado por um CLP (Controlador Lógico Programável). A aplicação escolhida foi um misturador, que faz a mistura entre dois produtos no interior de um tanque onde ela é aquecida. Para isto foram usados diversos componentes de forma a garantir maior eficiência e segurança do processo. Além disso, o processo pode ser monitorado via interface gráfica.

Palavras-chave — CLP; Automação; misturador; líquidos; sensores.

I. INTRODUÇÃO

Para uma empresa onde existe a necessidade de se realizar a mistura entre dois produtos líquidos de forma automatizada, segura e eficiente, tem-se a possibilidade de implementar o misturador automático. Neste misturador ocorre a entrada de dois líquidos em um reservatório (tanque) onde os mesmos são misturados, aquecidos e, por fim, escoam para uma saída de forma controlada, assim podendo ser movidos para outro processo ou finalidade, dependendo das necessidades da empresa. Para o controle deste processo, utiliza-se um CLP.

O processo consiste na mistura entre dois produtos líquidos em um reservatório. Os produtos entram no reservatório por intermédio de válvulas. Há uma válvula de entrada para cada produto e a entrada dos líquidos ocorre separadamente (um líquido após o outro). Durante a entrada dos líquidos, há a medição de nível volumétrico através de dois sensores de nível, um analógico e outro digital, caso aconteça alguma falha em um deles. O sensor de nível analógico informa a porcentagem do volume. Quando o tanque se encontra cheio, o motor do misturador é acionado movimentando as pás para a realização da mistura. Enquanto a mistura ocorre, há o aquecimento da solução por uma resistência com temperatura controlada. O processo, então, ocorre durante um tempo de 30 minutos. Após isso, ocorre a saída da mistura através de uma válvula proporcional, que controla a saída da mesma, segundo as necessidades da empresa. O mesmo pode ser interrompido através de um botão desliga a qualquer momento ou se houve algum problema durante a realização do processo.

Neste artigo: a seção II se refere aos componentes do processo; a seção III, sobre a solução física; a seção IV, sobre o mapeamento de entradas e saídas do processo; a seção V,

sobre a máquina de estados; a seção VI, sobre a solução lógica; a seção VII, sobre a IHM; a seção VIII, sobre a *Web*

Server; a seção IX, sobre a simulação; a seção X, sobre as conclusões finais e a seção XI sobre as referências.

II. COMPONENTES DO PROCESSO

A. CLP:

[1] A definição de um CLP (Controlador Lógico Programável) segundo a NEMA (National Electrical Manufacturers Association) é um aparelho eletrônico digital que utiliza uma memória programável para o armazenamento interno de instruções para implementações específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, para controlar, através de módulos de entradas e saídas, vários tipos de máquinas ou processos. Por essas e outras aplicações pode se dizer que o CLP é um tipo de computador usado nas indústrias para fazer o controle de diversos processos e aperfeiçoar a produção dos mesmos.

Os CLP's foram desenvolvidos para substituir os relés de controle, pois um painel de relés de controle ocupava muito espaço para processos pequenos e não possuíam muita confiabilidade por causa do excesso de fios necessário e desgaste mecânico no decorrer do tempo.

Pode-se dizer, então, que os CLP's são dispositivos mais confiáveis, bastando apenas a instalação das entradas e saídas e alimentá-lo. É possível programar processos no CLP em pelo menos cinco linguagens de programação. Porém, isso depende do fabricante do CLP. Dentre as cinco linguagens, a mais usada se chama "Ladder" que é similar ao sistema antigo de relés, permitindo uma maior facilidade na programação para quem era acostumado com esse sistema.

O CLP é ilustrado a seguir, conforme a figura 1.



Figura 1: CLP Siemens S7-1200 [2]

B. Sensor de temperatura :

O sensor de temperatura é utilizado para medir a temperatura dos produtos de acordo com o aquecimento da resistência. O sensor utilizado no processo é um termopar do tipo J, feito de ferro, que mede temperaturas de -210°C até 760°C [3]. Este sensor é analógico e informa os valores da temperatura atual do processo para o CLP que são mostrados na IHM. O termopar é ilustrado conforme a figura 2.



Figura 2: Termopar tipo J [4]

C. Sensor de nível ultrassônico:

O sensor de nível é utilizado para medir o nível do volume dos produtos utilizados no processo de mistura. O sensor de nível analógico utilizado no processo é um sensor ultrassônico. Este sensor varia seus sinais de saída de acordo com a diferença de tempo entre a emissão e recepção de uma onda eletromagnética quando a mesma encontra uma barreira física (neste caso, o nível a ser medido). Os valores de nível são lidos pelo CLP e também são mostrados instantaneamente na IHM. O sensor de nível é ilustrado conforme figura 3.



Figura 3: Sensor de nível ultrassônico [5]

D. Sensor de nível digital:

O sensor de nível digital utilizado no processo é um sensor de interruptor de boia. Quando o nível atinge este sensor, o mesmo acusa valor de saída digital alto. Este sensor é ilustrado conforme figura 4.



Figura 4: Boia [6]

E. Aquecedor:

O aquecedor neste protótipo é uma resistência que tem sua temperatura controlada através do CLP. Esta resistência elétrica é do tipo tubular para imersão, especialmente desenvolvida para o aquecimento de líquidos em tanques e normalmente é feita de inox, chumbo ou titânio, com isolamento de cerâmica e óxido de magnésio [7]. A temperatura de trabalho suportada é de aproximadamente 600°C . A resistência é ilustrada conforme a figura 5.



Figura 5: Aquecedor [8]

F. Termostato:

No processo, há a presença de um termostato, por precaução. Quando a temperatura da mistura passa do valor selecionado no mesmo, há o envio de um sinal de emergência para o CLP.

G. Misturador:

O misturador consiste em um conjunto de pás que sofre rotação quando seu motor é acionado, fazendo com que os líquidos se agitem e se misturem um ao outro. O misturador é ilustrado conforme figura 6.



Figura 6: Misturador [9]

III. SOLUÇÃO FÍSICA

Para a solução física, utilizaram-se os componentes citados acima. Há a presença de dois sensores de nível (um analógico e outro digital), para caso houver falha no sensor analógico.

Para o controle de temperatura da mistura há um termopar e, caso o mesmo sofra algum dano, há um termostato que enviará um sinal para o CLP, avisando que houve um erro no termopar quando a temperatura passar do valor *setado* no termostato. Há uma resistência regulável, que pode aquecer a mistura a temperaturas de até 600°C. Para fins de maior confiabilidade no processo, usa-se também a IHM, onde se pode monitorar o comportamento das variáveis relativas ao processo. Diante disso, viabiliza-se a garantia da qualidade da mistura no processo.

IV. MAPEAMENTO DE ENTRADAS E SAÍDAS DO PROCESSO

Para a realização e controle do processo, são necessários os devidos sensores e atuadores, mapeados a seguir, cada qual com sua função descrita.

No CLP:

A. Entradas:

Tabela 1: Portas de entrada

Entrada	Tipo	Descrição
Botão liga	Digital	Botão que inicia o processo.
Botão desliga	Digital	Botão que interrompe o processo.
Sensor de temperatura	Analógico	Sensor para medir a temperatura do produto.
Sensor de nível	Analógico	Sensor para medir o nível do volume ocupado pelo produto.
Sensor de nível	Digital	Sensor que mede os níveis de volume quando cheio, médio ou vazio.
Termostato	Digital	Dispositivo que varia a temperatura do processo.

B. Saídas:

Tabela 2: Portas de saída

Saída	Tipo	Descrição
Motor	Digital	Motor do misturador.
Resistência	Digital	Resistência que aquece a mistura.
Válvula 1	Digital	Válvula para a vazão do primeiro produto.
Válvula 2	Digital	Válvula para a vazão do segundo produto.
Válvula proporcional	Analógica	Válvula de saída da mistura.
Alarme	Digital	Alarme de emergência por falha durante o processo.

No IHM:

Tabela 3: Entradas do IHM

Botão	Descrição
Start	Inicia o processo
Stop	Interrompe o processo
Emergência	Suspende o processo

V. MÁQUINA DE ESTADOS

A máquina de estados é uma ilustração geral das etapas do processo. Com ela é possível analisar de forma mais visual as

mudanças de estados através de transições. Resumidamente, o que ocorre é:

1) No estado 0, o valor de todas as entradas é zero, ou seja, as mesmas se encontram, desligadas. Para haver a mudança do estado 0 para o estado 1, ou Botão Liga deve ser acionado ou o Botão Liga da IHM deve ser pressionado;

2) No estado 1, a válvula 1 é acionada. Para haver mudança do estado 1 para o estado 2, o Sensor de Nível Analógico deve acusar o mesmo valor que valor *setado* no CLP;

3) No estado 2, a válvula 2 é acionada. Para haver mudança para o estado 3, o Sensor de Nível Analógico deve acusar nível alto;

4) No estado 3, o motor do misturador e a resistência são acionados. A mudança para o estado 4 ocorre quando o sensor de temperatura acusa o valor da temperatura alta;

5) No estado 4, o motor continua acionado e a resistência é desligada. Ocorre mudança para o estado 5 quando o tempo de operação for maior que 30 minutos. Há regressão do estado 4 para o estado 3 se o sensor temperatura acusar o valor de temperatura baixa;

6) No estado 5, a válvula proporcional de saída continua acionada, há mudança para o estado 0 (estado inicial) quando o sensor de nível analógico acusar nível baixo;

7) Dos estados 1 a 5, pode-se haver mudança para o estado 6, caso o termostato for acionado e o sensor de nível digital acusar o valor de nível alto e os botões de emergência da IHM também.

A máquina de estados é ilustrada abaixo, conforme a figura 6, onde:

- E0, E1, E2, E3, E4, E5, E6, representam os estados de 1 a 6;
- SNA é o sensor de nível analógico;
- SND é o sensor de nível digital;
- ST é o sensor de temperatura;
- M é o motor do misturados;
- R é a resistência;
- V1 é a válvula do líquido 1;
- V2 é a válvula do líquido 2;
- VP é a válvula de saída proporcional;
- A é o alarme.

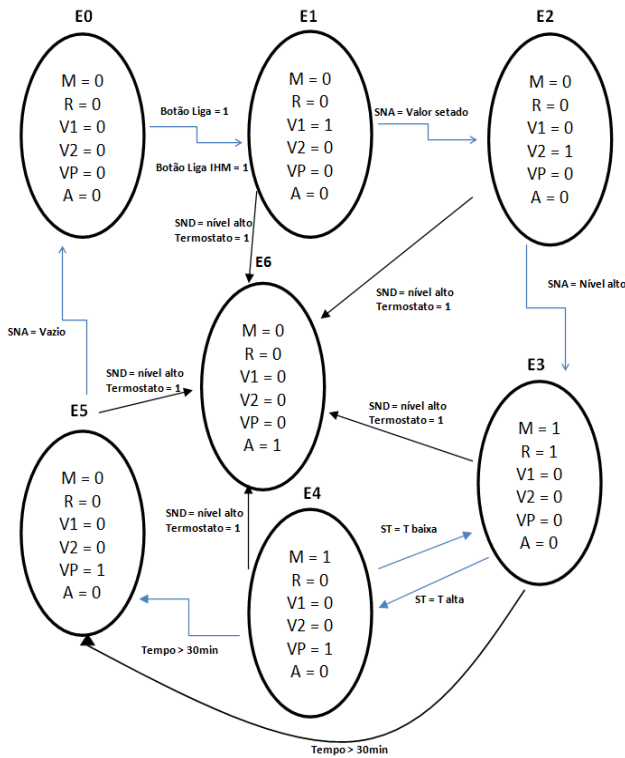


Figura 7: Máquina de estados.

VI. SOLUÇÃO LÓGICA

A solução lógica do problema consiste em uma sequência de códigos em que se procura simplificar o processo, tornando-o mais eficiente o possível. Cada Network (conexão) representa uma etapa do processo e, ao todo, o código se apresenta em 21 Networks. O programa é escrito em Ladder e Blocos de Função (ou FDB) e é ilustrado em seguida conforme as figuras 8, 9, 10, 11, 12 e 13.

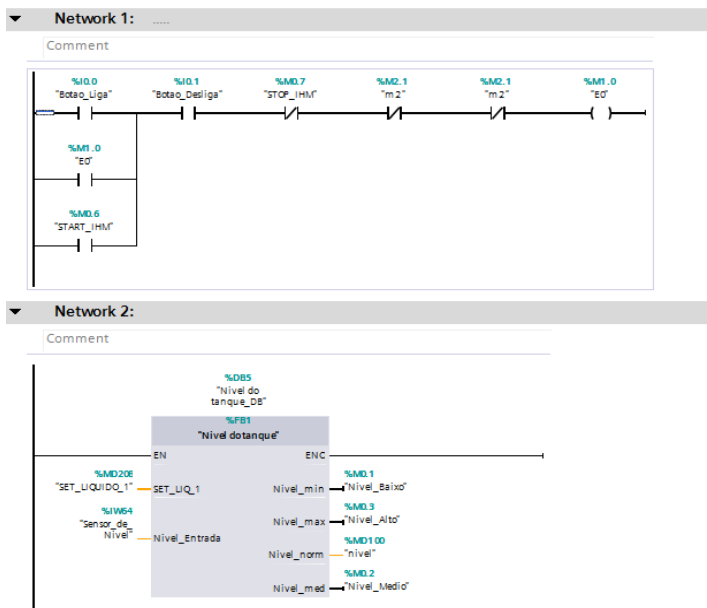


Figura 8: Networks 1 e 2.

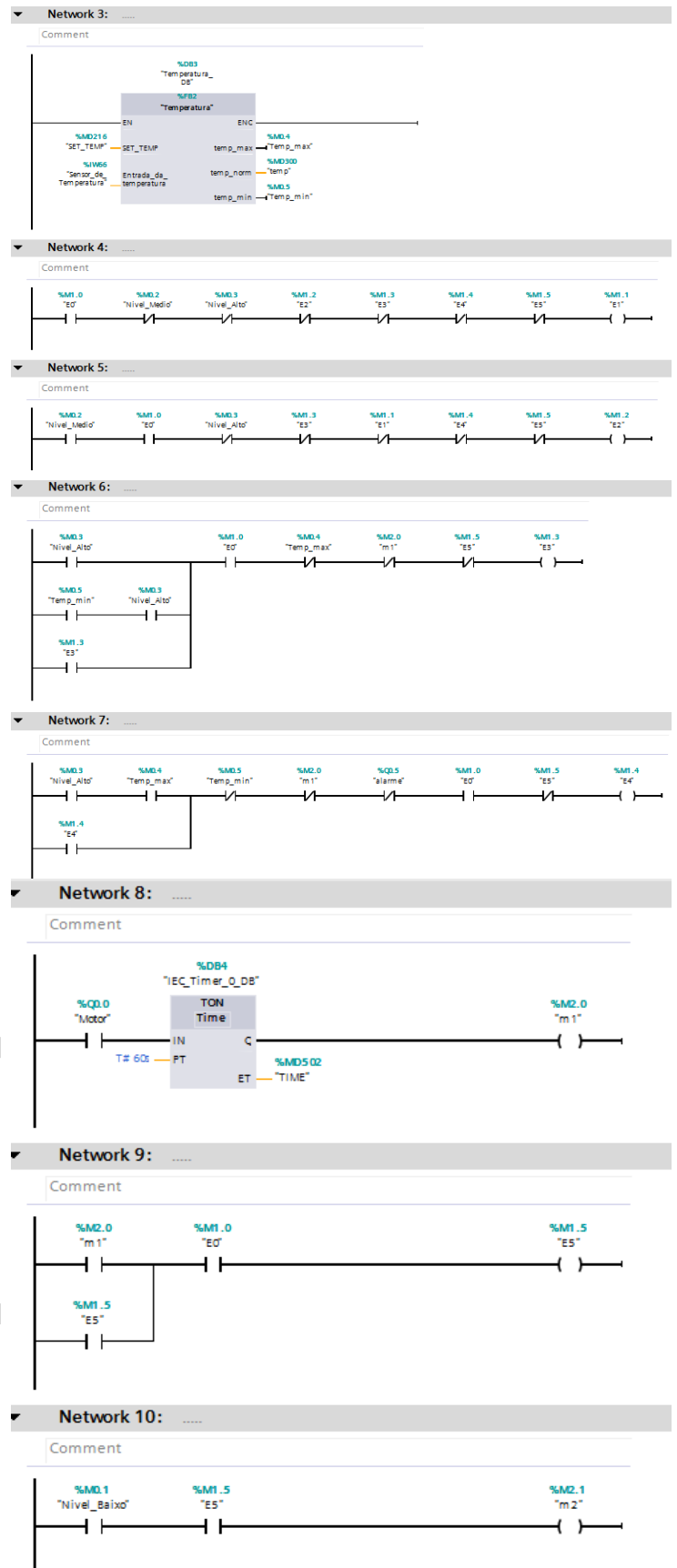


Figura 9: Networks 3 a 10.

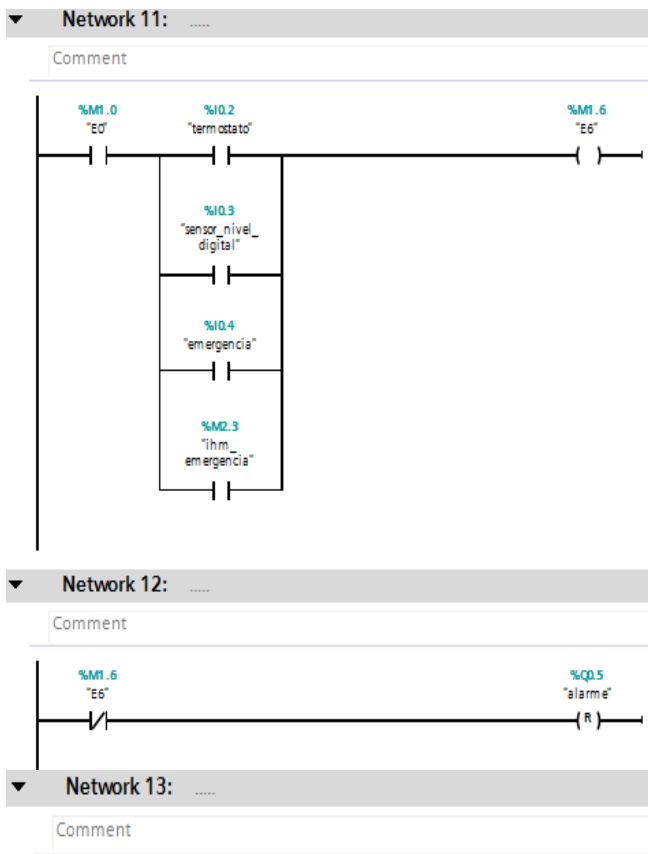


Figura 10: Networks 11 a 13.

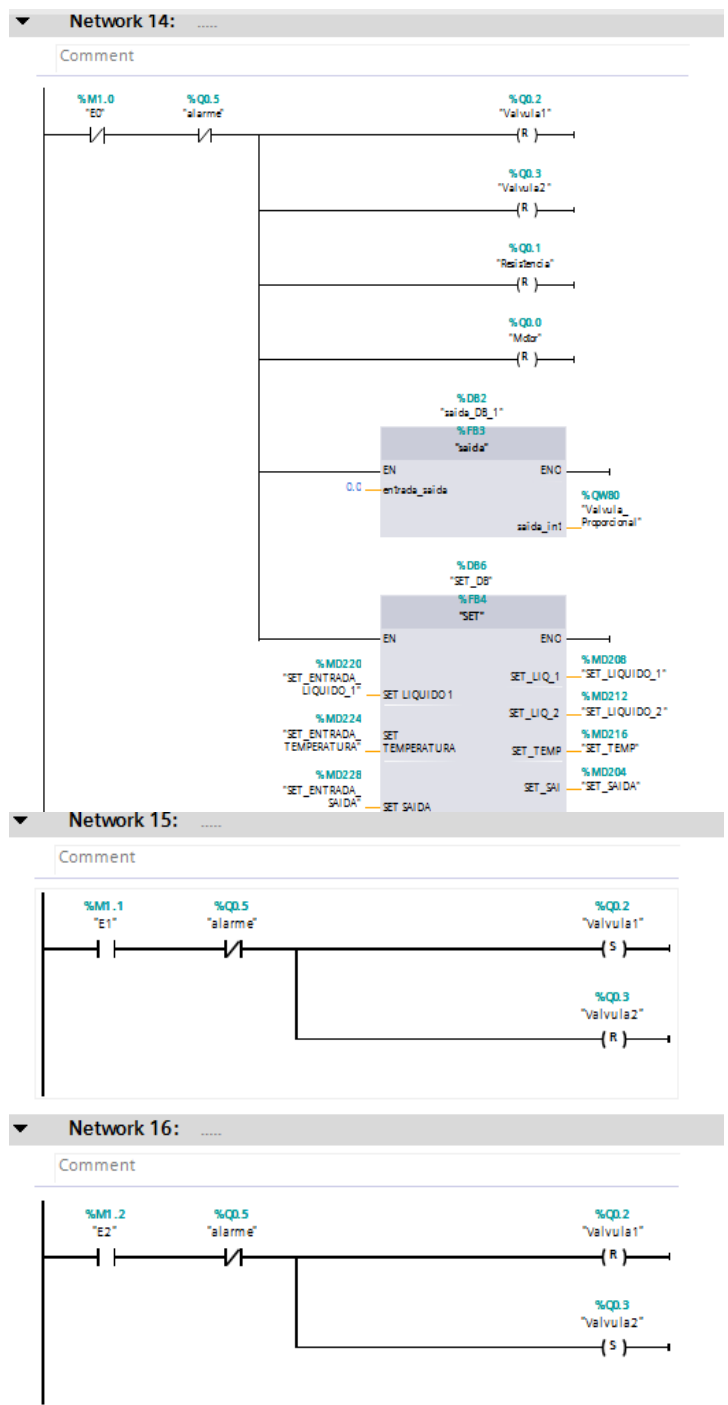


Figura 11: Networks 13 a 16.

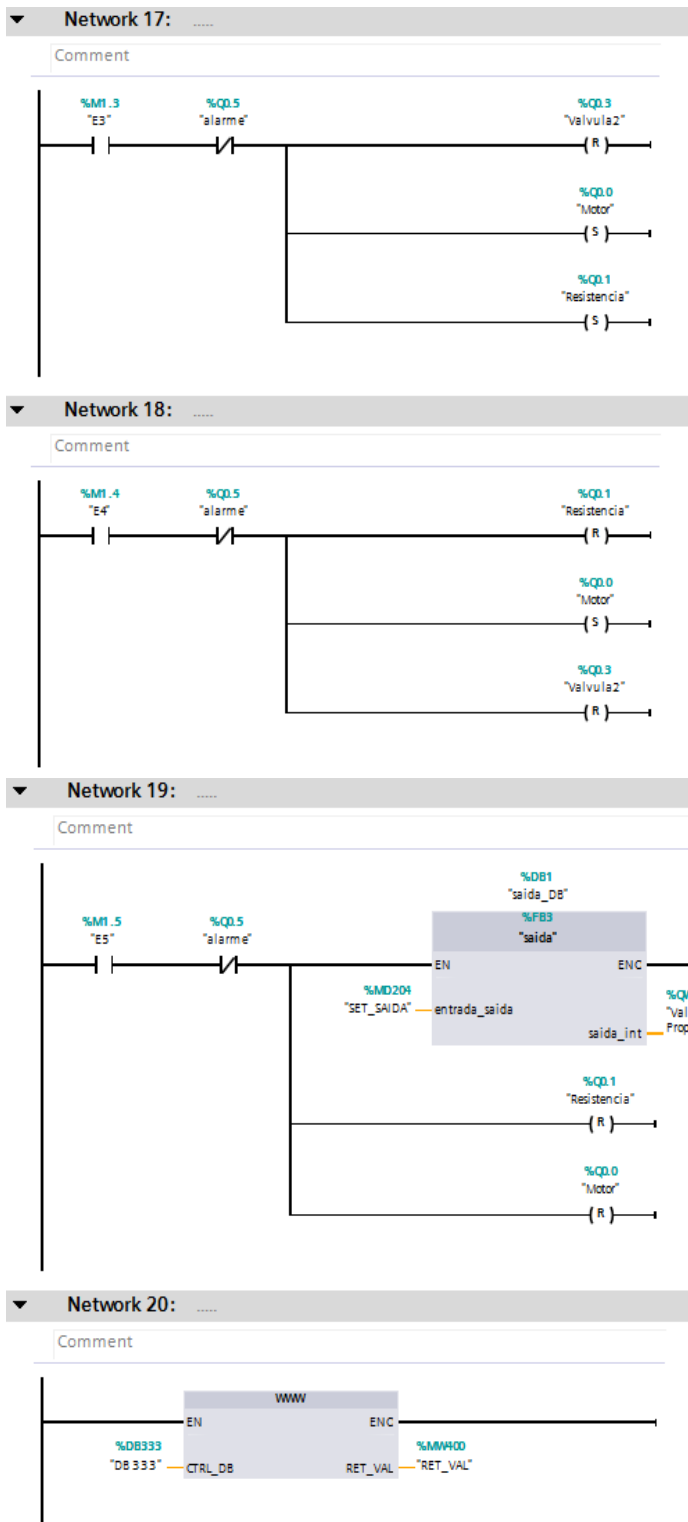


Figura 12: Networks 17 a 20.

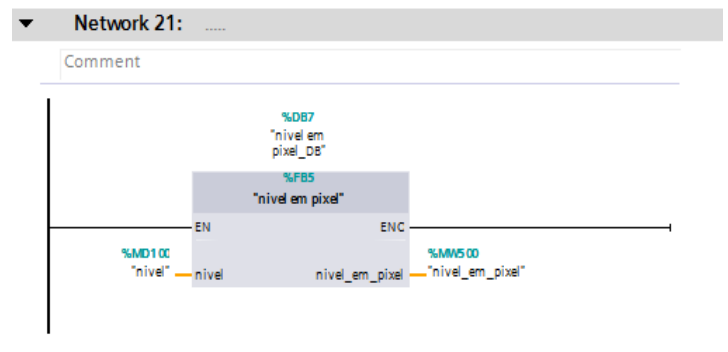


Figura 13: Network 21.

As imagens acima representam o que se chama de diagrama de transições e ações do processo. Um diagrama de transições para a linguagem “Ladder” consiste em criar uma variável para cada estado. Por exemplo, para o estado 1, tem-se a variável E1. O diagrama de ações refere-se às ações consequentes de cada estado. No processo, por exemplo, o E2 seta a válvula 2, caso o alarme não estiver acionado.

VII. IHM

IHM (Interface Homem Máquina) é um hardware industrial composto por uma tela de cristal liquido e um conjunto de teclas para navegação ou inserção de dados, é um sistema utilizado em chão de fábrica, com capacidade de gerenciamento de um numero limitado de variáveis.

A IHM utilizada no processo foi programada contendo três telas. Uma tela de início, onde é mostrado se as saídas estão ativadas ou não e onde é possível acionar e desligar o sistema, um botão de emergência. Na tela inicial, estão presente também dois botões que, quando pressionados, alternam a tela inicial para as telas 2 e 3, respectivamente. A segunda tela é a tela de leitura, é possível ver o tempo que o motor está ligado, ver o nível do tanque e a temperatura do liquido. A última tela é de configuração, onde é possível fazer a configuração da ocupação do líquido em porcentagem. Nela também é possível escolher o percentual de abertura da válvula de saída. As telas da IHM são ilustradas conforme as figuras a seguir.

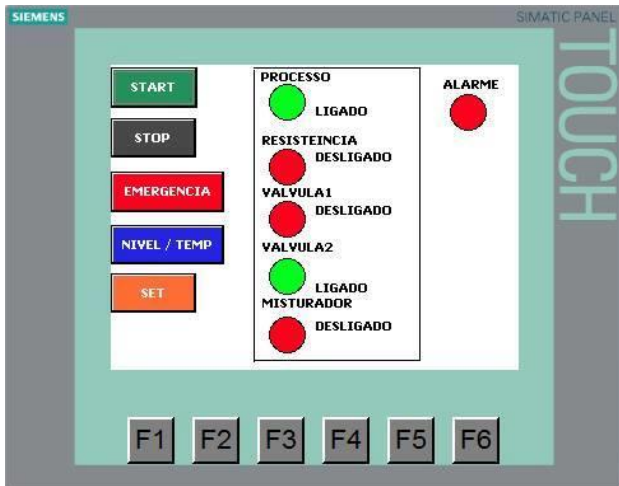


Figura 14: Tela 1 da IHM.

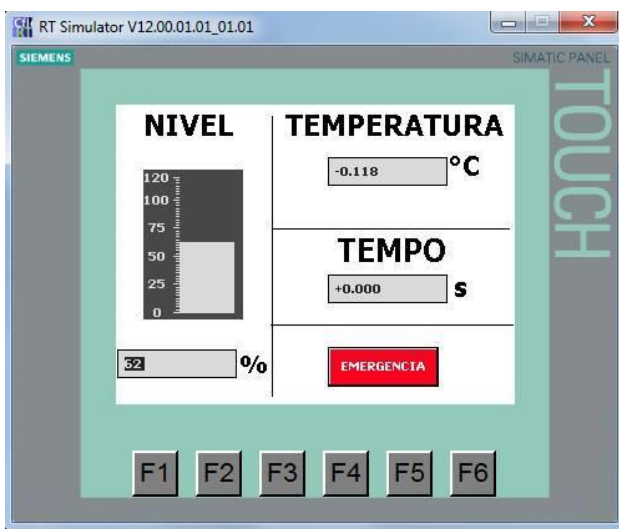


Figura 15: Tela 2 da IHM.



Figura 16: Tela 3 da IHM.

VIII. WEB SERVER

A Web Server criada para o processo foi feita em *html*, onde é possível visualizar o processo em tempo real através do IP da rede onde o CLP está conectado. Na *Web Server* é possível verificar o nível atual do tanque, a temperatura, os estados das válvulas, misturador, aquecedor e do alarme. Conforme a figuras abaixo é possível observar a *Web Server* com as leituras e a ilustração do nível do tanque.

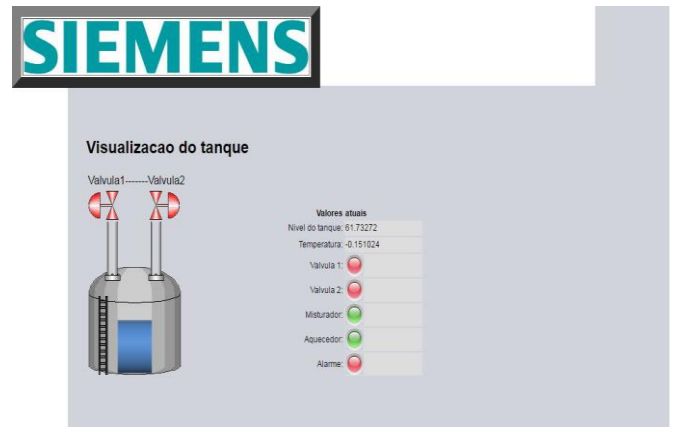


Figura 17: Visualização do tanque através da Web Server.

IX. SIMULAÇÃO

Para a simulação dos sensores de temperatura e de nível do processo utilizaram-se fontes variadas. A tensão de saída da fonte foi de 0 a 5V, simulando a tensão de um sensor analógico. Para a simulação dos sensores digitais como o de nível e temperatura foram usados botões e as saídas são sinalizadas através dos leds do CLP. Para verificar a abertura da válvula proporcional foi usado um multímetro, que mede a tensão de saída. Através da IHM, simulada em um computador, também é possível monitorar a simulação. A simulação foi feita seguindo o diagrama de estados e testando possíveis erros. Os dispositivos utilizados para a mesma são ilustrados conforme a figura 18.

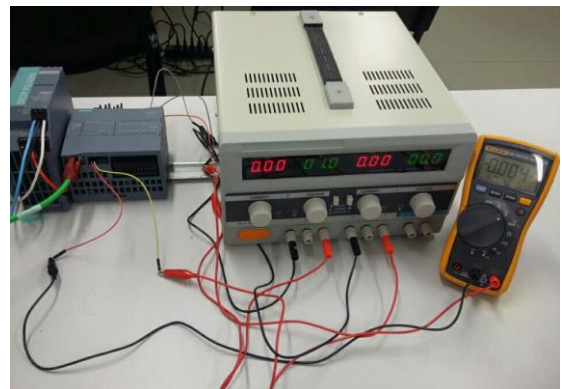


Figura 18: Dispositivos utilizados para a simulação do processo.

X. CONSIDERAÇÕES FINAIS

Diante disso tudo é possível dizer que as exigências de uma empresa que necessita de um processo de mistura entre

dois líquidos podem ser atendidas. Visto que, com os componentes utilizados e suas implementações físicas, além da implementação lógica feita por intermédio do CLP, do monitoramento das variáveis através da IHM e da *Web Server*, se faz possível a realização de um processo de mistura seguro e eficiente.

XI. REFERÊNCIAS

- [1] Disponível em:
http://www.digel.com.br/novosite/index.php?option=com_content&view=article&id=116:clp;
- [2] Disponível em:
[http://plccompare.com/siemens-s7-1200/;](http://plccompare.com/siemens-s7-1200/)
- [3] Disponível em:
https://pt.wikipedia.org/wiki/Termopar#Tipo_J_.28Ferro_.2F_Constantan.29;
- [4] Disponível em:
<http://www.exacta.ind.br/?p=conteudo&id=203;>
- [5] Disponível em:
[http://www.digitrol.com.br/transmissores-de-nivel/transmissor-ultra-sonico-de-nivel-probe-lu-siemens.php;](http://www.digitrol.com.br/transmissores-de-nivel/transmissor-ultra-sonico-de-nivel-probe-lu-siemens.php)
- [6] Disponível em:
<http://eletronicos.mercadolivre.com.br/mini-nivel-digital;>
- [7] Disponível em:
<http://www.pmrresistencias.com.br/resistencia-tubular-imersao.php>
- [8] Disponível em:
[http://www.resistenciassp.ind.br/resistencias-eletricas/tubulares/imersao.php;](http://www.resistenciassp.ind.br/resistencias-eletricas/tubulares/imersao.php)
- [9] Disponível em:
[http://portuguese.agitatormixer.com/sale-986984-high-strength-belt-drive-cp5-18-5-230-side-entry-chemicals-liquid-mixer-agitator.html;](http://portuguese.agitatormixer.com/sale-986984-high-strength-belt-drive-cp5-18-5-230-side-entry-chemicals-liquid-mixer-agitator.html)

Misturador de tintas

Processo controlado por CLPs

Artur Henrique Lauth, Arthur Phillip B. Siegel, Nikolas Sbaraini Hammes
 Universidade Federal de Santa Catarina
 UFSC
 Blumenau, Brasil

Resumo: Esse trabalho tem como objetivo o aprendizado da programação de CLP por meio da simulação de um sistema misturador de tintas. Utilizaremos a linguagem Ladder e Texto Estruturado para essa programação e a comunicação de dois CLP's independentes.

Palavras-chave: CLP, programação, diagrama de estados, sensores.

I. INTRODUÇÃO

Este artigo descreverá um processo industrial em que ocorre a mistura de duas ou mais tintas de cores diferentes em um sistema majoritariamente automatizado onde também é feito o engarrafamento do produto para sua distribuição ao consumidor.

A solução encontrada foi usar a comunicação entre dois CLPs (Controlador Lógico Programável). O primeiro irá controlar o processo de mistura e o segundo o de engarrafamento. O sistema terá um controle de variáveis booleanas (simuladas com chaves, sensores, motores) e será monitorado através de um aplicativo para smartphone.

As metas da automatização desse processo é otimizar tempo, custo e produção de uma indústria que necessite de uma configuração de produção similar. Outro objetivo é garantir que o processo não funcione de maneira indevida, adicionando vários mecanismos de emergência e de controle.

II. DESCRIÇÃO DO PROCESSO

A. Processo de mistura das tintas:

A resolução do problema foi baseada em um misturador de duas tintas, porém pode-se facilmente adicionar um número maior de tintas.

O processo começa com o acionamento da botoeira START, energizando o sistema. As bombas B1 e B2 são então acionadas fazendo com que as tintas A1 e A2 comecem a preencher o tanque do misturador. O preenchimento do tanque D é controlado pela chave de nível SA que indica quando o tanque está cheio.

As bombas são desligadas quando o fluido atinge a chave de nível SA e então o misturador MIS é energizado. Após o tempo de mistura escolhido pelo operador ser completado a válvula P é aberta e o produto resultante escorre pelo cano até o processo de engarrafamento.

A chave de nível SB determina quando o nível do tanque D atinge o nível mínimo, então válvula P é fechada e as bombas B1 e B2 são acionadas novamente. O processo entra então em ciclo, ondeo misturador MIS não é mais desligado a fim de evitar esforços de partida e interrupção.

O sistema pode ser interrompido completamente em qualquer estágio do processo apertando o botão de emergência BE. Os sensores de temperatura S1, S2 e S3 nas bombas e no misturador MIS também param o sistema caso identifiquem uma temperatura muito elevada. O esquema do processo pode ser visualizado na figura 1.

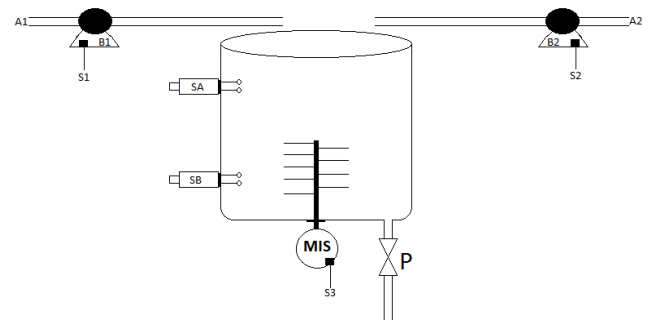


Fig. 1. Ilustração processo de mistura.

B. Processo de engarrafamento:

O processo de engarrafamento é energizado quando a válvula P do processo de mistura é aberta. O motor M da esteira é então acionado até que uma garrafa chegue no local definido pelo sensor de presença SL1. O injetor ENC é ativado para encher uma garrafa e o sensor SU indica que esta foi preenchida.

O injetor é então desativado até que outra garrafa seja posicionada. As garrafas cheias são posicionadas debaixo do lacrador TAM, sua posição é definida pelo sensor de presença SL2. As garrafas são espaçadas de uma forma que uma garrafa vazia esteja debaixo do injetor ao mesmo tempo que uma garrafa cheia esta sendo tampada. O processo se repete até que a mistura acabe (válvula P fecha).

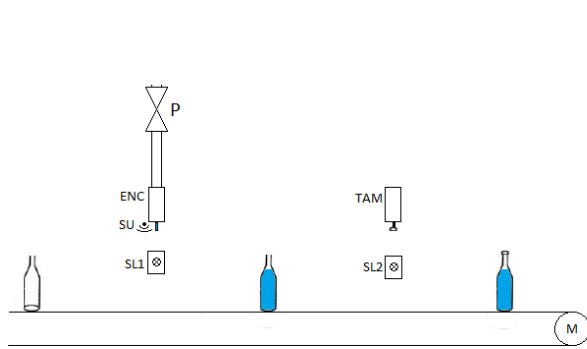


Fig. 2. Ilustração processo de engarrafamento.

III. COMPONENTES DO PROCESSO

A. Controlador lógico programável :

O CLP é um equipamento composto de componentes eletrônicos, memória, que será responsável pelo controle do processo através de dispositivos de entrada e saída. O processo em questão usará somente E/S do tipo digital. O CLP usado será Siemens S7-1200 e contém 8 entradas e 8 saídas.



Fig. 3. CLP Siemens S7-1200

B. Chave de nível :

Chave de Nível monitora a altura da tinta no tanque D. A chave fecha o circuito quando o nível de líquido foi atingido enviando um sinal para o CLP. No processo são utilizados duas chaves de nível, um para determinar o nível mínimo do reservatório e o outro para determinar o nível máximo.



Fig. 4. Sensor de nível

C. Bombas:

As bombas são utilizadas para enviar as tintas para o reservatório e são controladas pelo CLP. Ambas trabalham com a mesma vazão.



Fig. 5. Bomba

D. Sensores de temperatura:

O processo possui três sensores de temperatura (sensores analógicos), um em cada bomba e um no misturador. O CLP monitora os dados dos sensores, que se indicarem uma temperatura de risco (superaquecimento dos motores), interrompe o sistema.



Fig. 6. Sensores de temperatura

E. Misturador:

O misturador é responsável pela mistura das tintas, e é controlada pelo CLP.



Fig. 7. Misturador

F. Sensor ultrassônico:

O sensor ultrassônico (sensor analógico) mede a altura de tinta nas garrafas. O CLP usa essa informação para controlar a esteira do processo de engarrafamento.

Observação: Na simulação do processo o sensor foi considerado uma entrada digital (1 para < 10 cm , 0 para > 10 cm)



Fig. 8. Sensor ultrassônico

G. Sensor luminoso:

São sensores digitais cujos elementos de emissão e recepção estão justapostos no mesmo conjunto óptico. Os raios emitidos pelo transmissor refletem na superfície do objeto detectado e retornam ao elemento receptor. Determinam quando as garrafas estão posicionadas para encher e lacrar.



Fig. 9. Sensor luminoso

IV. DIAGRAMA DE ESTADOS

Para automatizar o processo de mistura e engarrafamento das tintas através de CLPs, foi usada a metodologia de programação chamada Diagrama de Estados. Abaixo vamos descrever os Estados, Saídas e Transições.

A. Mapeamento Entradas e Saídas:

Entradas processo de mistura		
Descrição	Variável	Endereço
Botão de início, energiza o sistema	START	I0.0
Botão de parada, desenergiza o sistema	STOP	I0.1
Sensor de nível alto	AS	I0.2
Sensor de nível baixo	SB	I0.3

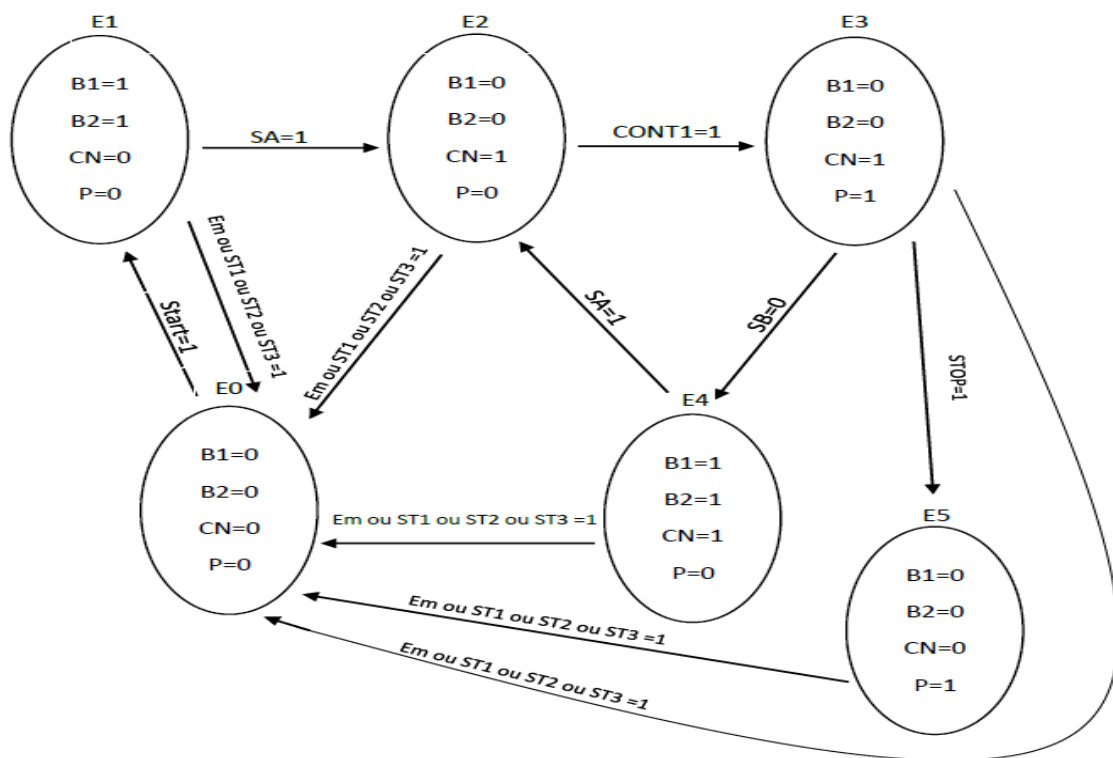
Entradas processo de mistura		
Descrição	Variável	Endereço
Sensor de temperatura bomba 1	ST1	I0.4
Sensor de temperatura bomba 2	ST2	I0.5
Sensor de temperatura do motor da misturadora	ST3	I0.6
Botão de emergência	EM	I0.7
Botão de parada, desenergiza o sistema	STOP	I0.1

Saídas processo de mistura		
Descrição	Variável	Endereço
Bomba 1	B1	Q0.0
Bomba 2	B2	Q0.1
Motor da misturadora	MIS	Q0.2
Válvula (permite a passagem do processo de mistura para o engarrafamento)	P	Q0.3

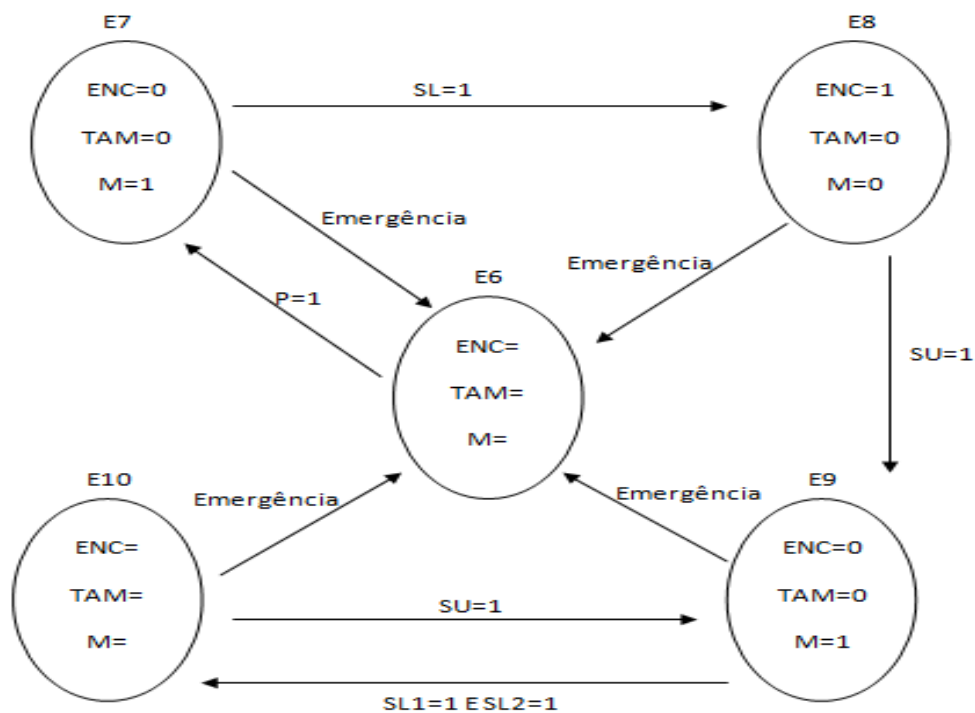
Entradas processo de engarrafamento		
Descrição	Variável	Endereço
Sensor luminoso 1	SL1	I1.0
Sensor luminoso 2	SL2	I1.1
Sensor ultrassônico	SU	I1.2

Saídas processo de engarrafamento		
Descrição	Variável	Endereço
Injetor de tintas	ENC	
Lacrador	TAM	I1.1
Motor que movimentava a esteira	M	I1.2

B. Máquina de estados processo de mistura:



C. Máquina de estados do processo de engarrafamento:



Emergência= Em ou ST1 ou ST2 = 1

D. Diagrama de transição do processo de mistura:

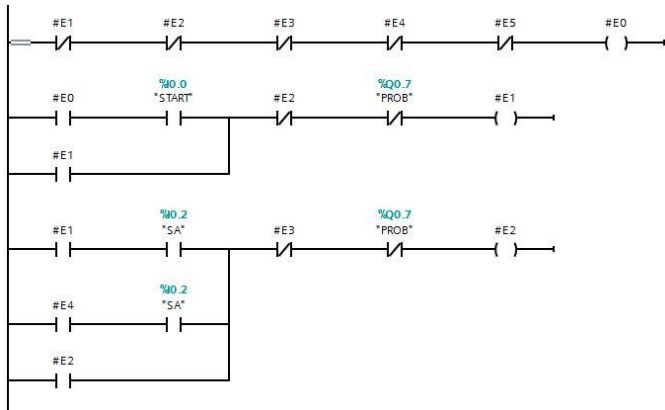


Fig. 10. Transições Etapas 0, 1 e 2.

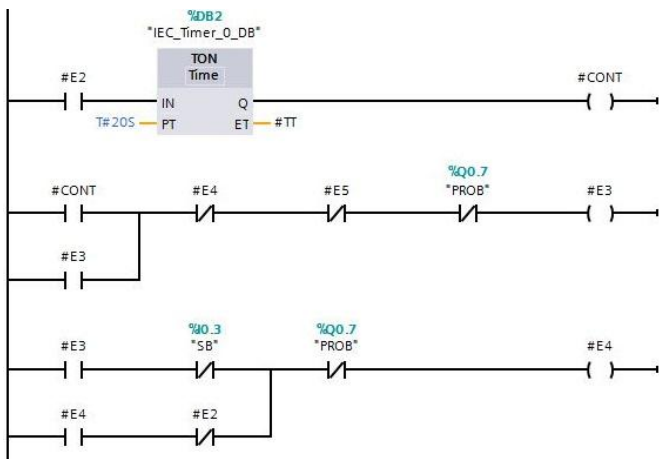


Fig. 11. Transições Etapas 2, 3 e 4.

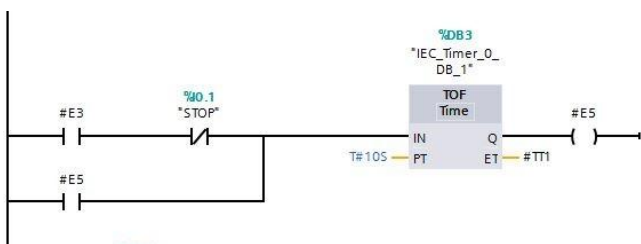


Fig. 12. Transição Etapa 5.

E. Diagrama de transição do processo de engarrafamento:

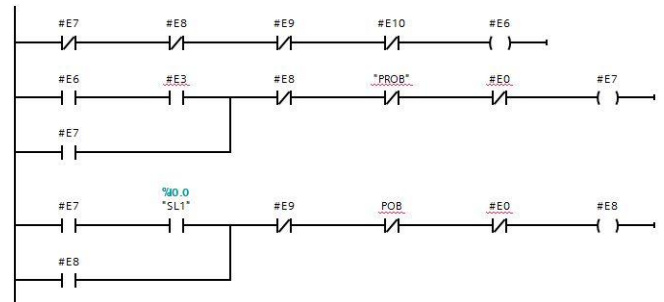


Fig. 13. Transições Etapas 6, 7 e 8.

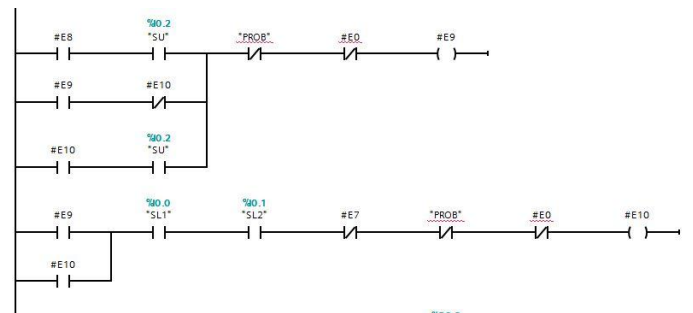


Fig. 14. Transições Etapas 8, 9 e 10

F. Diagrama de ações do processo de mistura:

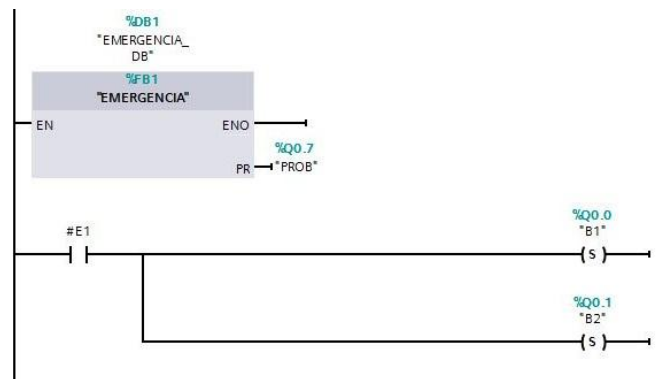


Fig. 15. Ações Etapa 1.

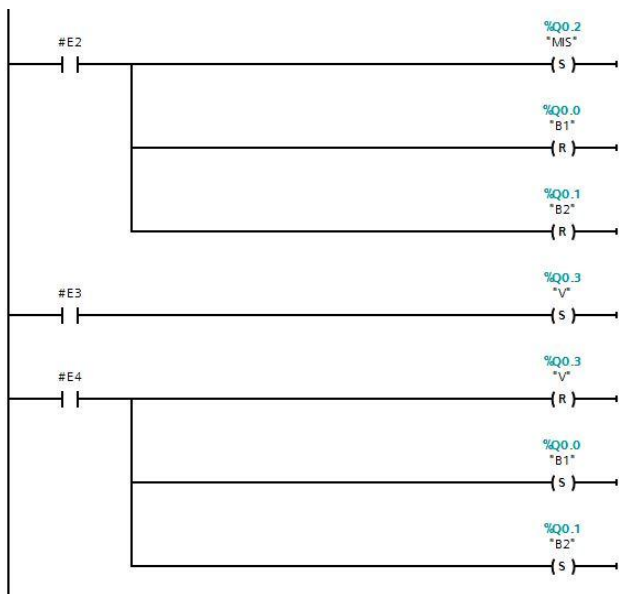


Fig. 16. Ações Etapas 2, 3 e 4.

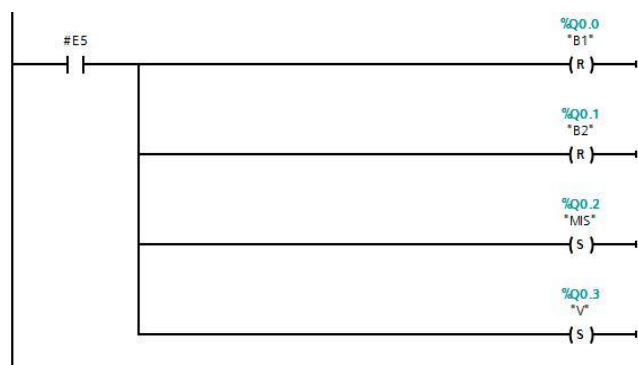


Fig. 17. Ações Etapa 5.

```

1 IF "EM" OR "ST1" OR "ST2" OR "ST3" THEN
2     // Statement section IF
3     #PR:=TRUE;
4 END_IF;
5

```

Fig. 18. Bloco de emergência em Structured text.

G. Diagrama de ações do processo de engarramento:

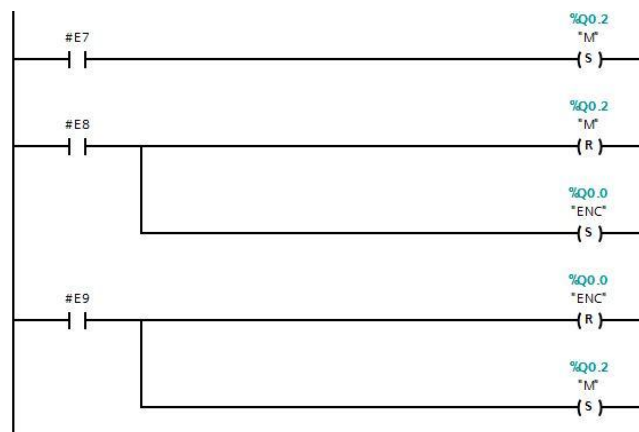


Fig. 19. Ações Etapas 7, 8 e 9.

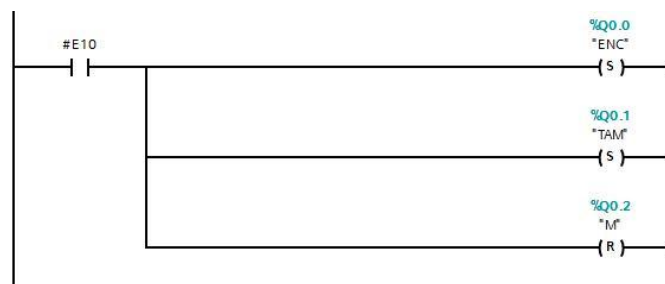


Fig. 20. Ações Etapa 10.

V. CONTROLE DE ENTRADAS COM APP

Utilizamos o aplicativo SIMATIC S7 para fazer a visualização das variáveis do nosso processo e para poder alterar seu estado lógico.

VI. CONCLUSÃO

Pôde-se observar que o uso de vários CLPs na indústria ajuda no controle de processos interligados, acarretando em um ganho maior na questão financeira e produtiva, pois a comunicação entre estes é mais precisa e sem muito ruídos fazendo com que todo o processo tenha menos erros e caso aconteça o erro em uma das partes as outras não serão prejudicadas. Observamos também que as variáveis de um CLP podem ser alteradas e visualizadas por meio do aplicativo SIMATIC S7. Portanto podemos aprender muito sobre a programação em linguagens diferentes, os modos de usar um ou mais CLPs em uma indústria e o controle das variáveis em tempo real a distancia.

VII. REFERÊNCIAS

[1] PRUDENTE, Francesco. PLC S7-1200 Teoria e Aplicações: Curso Introdutório. Rio de Janeiro: Ltc, 2014
 [2] Prudente, Francesco. PLC S7-1200 Teoria e Aplicações Curso Introdutório. LTC, 1, 2014.
 [3] Petruzella, Frank D. Controladores Lógicos Programáveis. Bookman Companhia Ed, 4,2013.

Fabricação de Refrigerantes

Processo controlado por CLPs

Eduardo Seiti Sonohara, João Gabriel Godoy, Leonardo Madeira Almeida
Universidade Federal de Santa Catarina
UFSC
Blumenau, Brasil

Resumo: Este trabalho consiste em uma esteira automatizada para a produção de refrigerante, sendo controlada por dois controladores lógico programáveis.

Palavras-chave: Informática industrial, CLP, programação.

I. INTRODUÇÃO

A partir da década de 60 e com o avanço da informática, foram desenvolvidos os controladores lógicos programáveis (CLP's) com o objetivo de substituir os painéis de relés, que eram utilizados para controlar máquinas e equipamentos industriais, principalmente na área automobilística. Por serem eletromecânicos os relés apresentavam diversos problemas como o desgaste nos contatos, devido à repetitividade do movimento, também a dificuldade e inflexibilidade do sistema. Diante deste problema a General Motors (GM) desenvolveu o primeiro projeto do controlador lógico programável para automatizar de forma mais ágil os métodos em sua linha de montagem. No qual exigia um equipamento mais robusto, maior facilidade na programação, alta confiabilidade, menores custo, dimensão e baixa manutenção.

Este artigo tem como objetivo apresentar um protótipo de uma esteira automatizada para a produção de refrigerante e mostrar cada passo do processo. Foram usados dois CLP's para a comunicação do processo da esteira e do misturador, podendo ser monitorado através de aplicativo para smartphone.

II. DESCRIÇÃO DO PROCESSO

A. Processo da esteira

A ideia do sistema é facilitar o modo de produzir refrigerantes, sendo ele mais eficaz, rápido e moderno. Inicialmente ao pressionar o botão (L0) a esteira é energizada, após o sensor fotoelétrico (S1) for acionado, para-se a esteira e inicia-se o processo de esterilização da garrafa, entretanto se o sensor de temperatura (S2) for ativado o sistema para completamente. Em seguida a esteira volta a se movimentar, logo depois começa o processo de rotulação (R1 e R2), caso o sensor fotoelétrico (S3) for acionado.

Logo após, a válvula de enchimento (M6) é aberta se o sensor fotoelétrico (S4) for ligado. Por último, inicia-se o processo de rosquear (T1) caso o sensor fotoelétrico (S5) estiver ligado. Além disso, cada um dos processos pode ser desligado ao pressionar o botão de emergência (D1) e também

a esteira é desenergizada a cada acionamento dos sensores fotoelétricos e energizada no final de cada processo. O esquema do processo pode ser visualizado na figura 1.

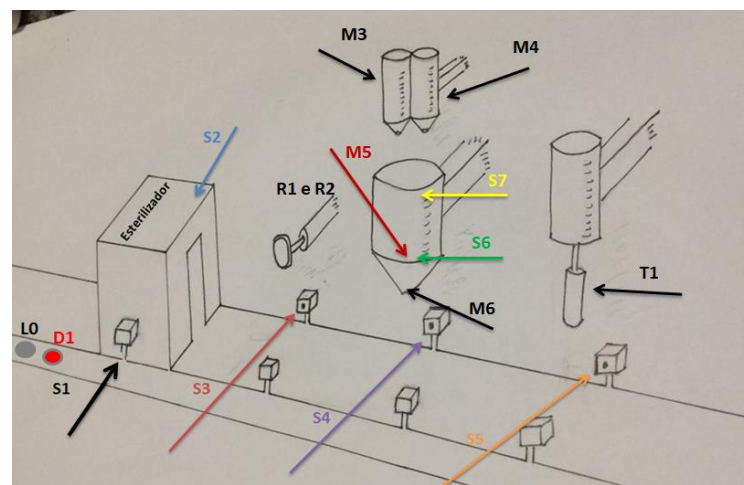


Fig. 1. Ilustração da esteira automatizada.

B. Processo do misturador

O processo do misturador funciona de acordo com o nível da solução dentro do tanque, se o sensor de nível baixo (S6) for acionado, as válvulas de xarope (M3) e água com gás (M4) são abertas até atingir um certo nível. Caso o sensor de nível alto (S7) for acionado o motor do misturador é ligado (M5) e as válvulas M3 e M4 são fechadas. O esquema do processo pode ser visualizado na figura 1.

III. COMPONENTES DO PROCESSO

A. Controlador lógico programável

O CLP é um controlador lógico programável, sendo ele normalmente usado na indústria e é composto por componentes eletrônicos, memória, no qual é responsável pelo controle do processo.



Fig. 2. CLP Siemens S7-1200

B. Sensor fotoelétrico

São sensores cujo funcionamento baseia-se na emissão de um feixe de luz, o qual é recebido por um elemento fotossensível. Ao interromper o feixe de luz, nota-se que a garrafa está posicionada no local desejado e assim inicia-se o processo.



Fig. 3. Sensor fotoelétrico.

C. Esterilizador

O esterilizador funciona como um processo de autoclavagem, no qual mantém a garrafa de vidro em contato com o vapor da água em altas temperaturas, por um determinado período suficiente para matar as impurezas.

D. Misturador de líquidos

O misturador tem a função de deixar a solução de xarope e água com gás em um estado homogêneo, através da rotação da pá interna.



Fig. 4. Misturador de líquidos.

E. Rosqueador

Este equipamento é utilizado para rosquear a tampa na garrafa que contem o refrigerante.



Fig.5. Rosqueador.

F. Rotuladora

Este processo é quando a garrafa recebe a identificação e a colagem do rótulo.



Fig. 6. Rotulador de garrafas.

G. Sensor de temperatura

Este sensor é responsável pelo monitoramento da temperatura durante a limpeza das garrafas, o qual se acionado, determina o desligamento do processo.



Fig. 7. Sensor de temperatura.

H. Sensor de nível

São sensores que detectam o nível de líquidos na altura em que foram instalados, convertem informações físicas em sinais elétricos.



Fig. 8. Sensor de nível.

I. SIMATIC S7

Este aplicativo para smartphone conecta com o controlador lógico programável através de via wi-fi, possibilitando o acesso global às múltiplas estações, para o controle e diagnóstico do sistema, também monitorar e modificar as variáveis e tags.

IV. DIAGRAMA DE ESTADOS

Abaixo contem a metodologia empregada e a solução lógica do controle do processo, o diagrama de estados (descreve os estados, saídas e transições).

A. Mapeamento Entradas e Saídas

TABELAI. ENTRADAS PROCESSO DA ESTEIRA

Entrada do processo da esteira		
Variável	Endereço	Descrição
D1	I0.6	Botão de emergência, desenergiza o processo
L0	I0.0	Botão de ligar, energiza o processo
S1	I0.1	Sensor de posição para a esterilização
S2	I0.5	Sensor de temperatura
S3	I0.2	Sensor de posição para a rotulação
S4	I0.3	Sensor de posição para encher a garrafa
S5	I0.4	Sensor de posição para tampar a garrafa

TABELAII. SAÍDAS PROCESSO DA ESTEIRA.

Saídas processo da esteira		
Variável	Endereço	Descrição
M1	Q0.0	Motor da esteira
M2	Q0.1	Esterilizador
M6	Q0.2	Válvula para encher a garrafa

R1	Q0.3	Rotulador 1
R2	Q0.4	Rotulador 2
T1	Q0.5	Rosqueadora

TABELAIII. ENTRADAS PROCESSO DO MISTURADOR.

Entradas processo do misturador		
Variável	Endereço	Descrição
S6	I0.2	Sensor de nível baixo
S7	I0.1	Sensor de nível alto
L0	I0.0	Botão de emergência, energiza o processo
D1	I0.5	Botão de emergência, desenergiza o processo

TABELAIV. SAÍDAS PROCESSO DO MISTURADOR.

Saídas processo do misturador		
Variável	Endereço	Descrição
M3	Q0.0	Válvula de melação
M4	Q0.1	Válvula de água com gás
M5	Q0.2	Motor do misturador de líquidos

B. Máquina de estados

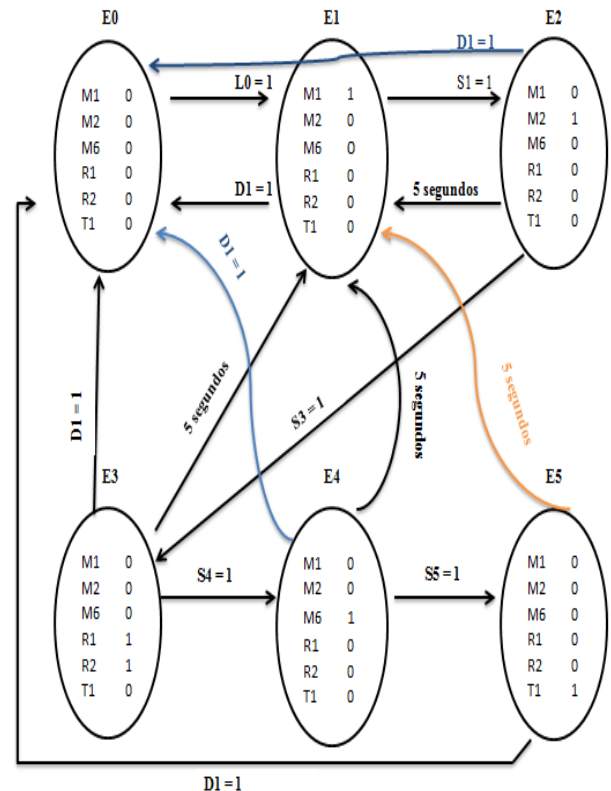


Fig. 9. Máquina de estados da esteira.

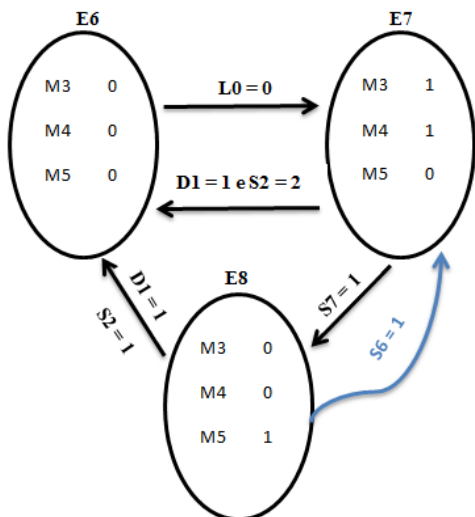


Fig. 10. Máquina de estados do misturador.

C. Diagrama de transição do processo da esteira

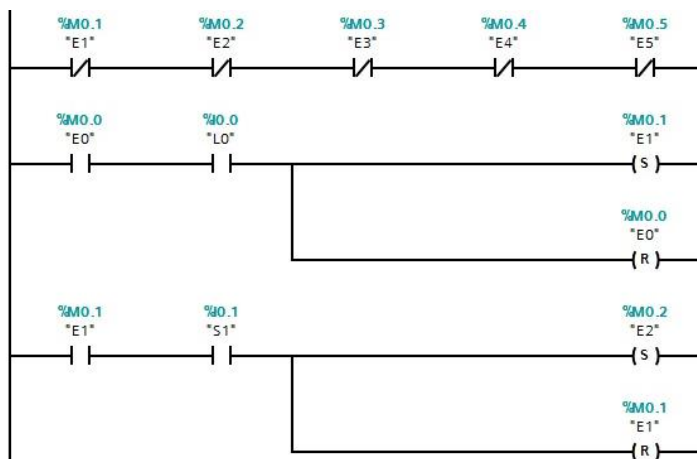


Fig. 11. Transições Etapas 0 e 1.

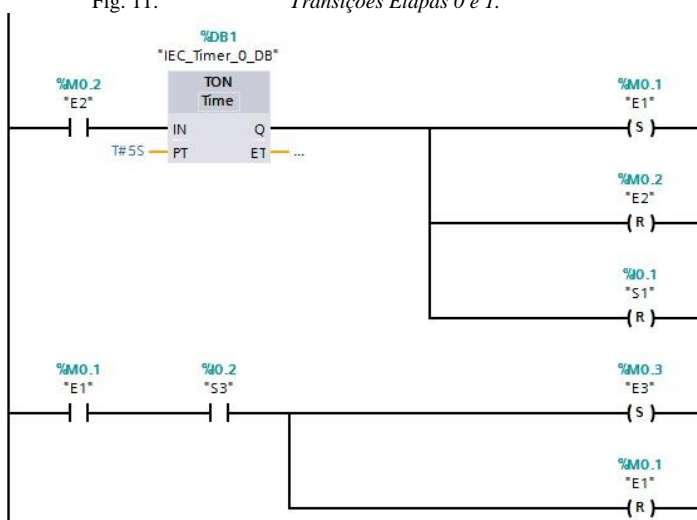


Fig. 12. Transição Etapa 2.

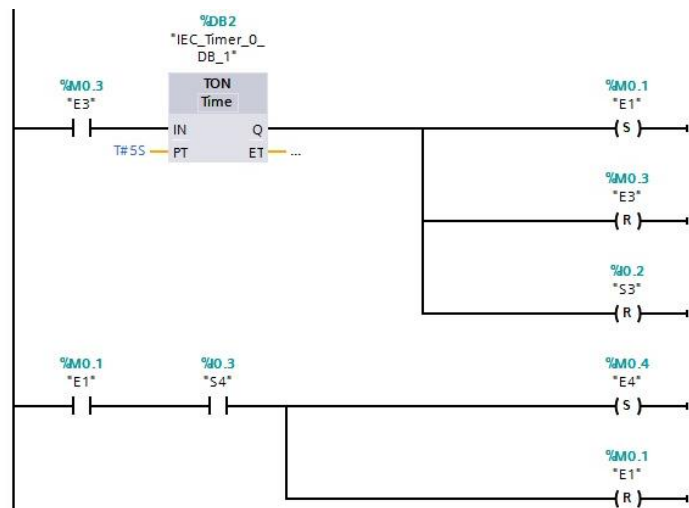


Fig. 13. Transição Etapa 3.

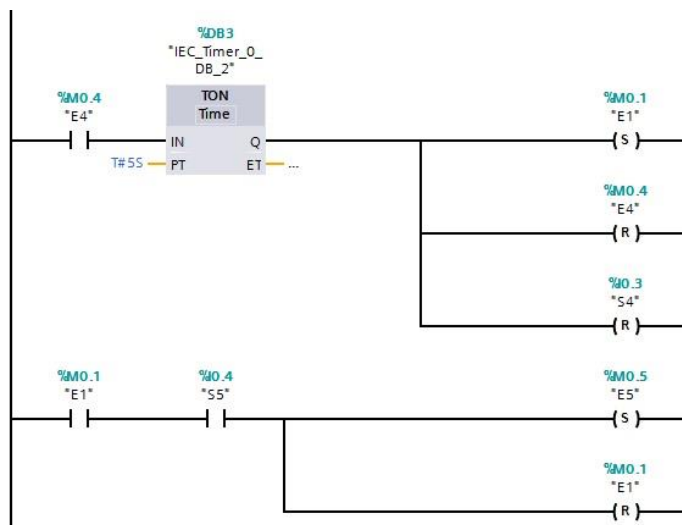


Fig. 14. Transição Etapa 4.

E. Diagrama de ações do processo da esteira

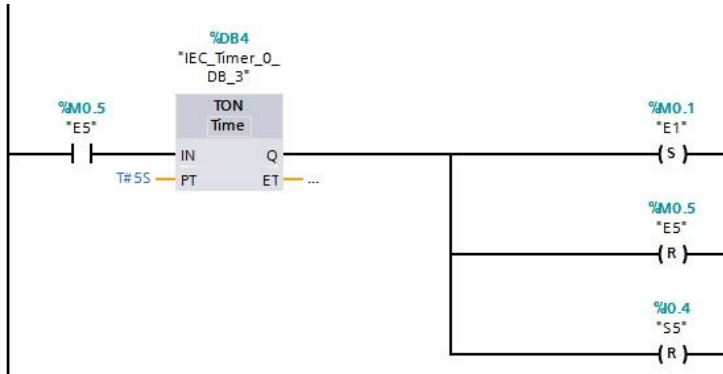


Fig. 15. Transição Etapa 5.

D. Diagrama de transição do processo do misturador

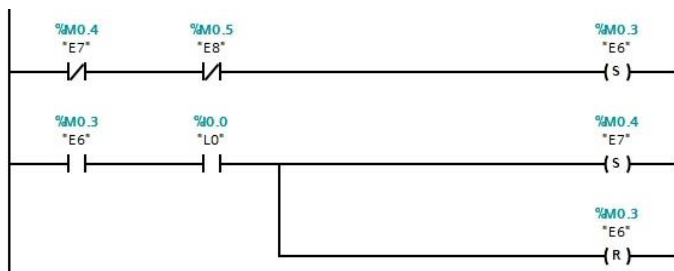


Fig. 16. Transição Etapa 6.

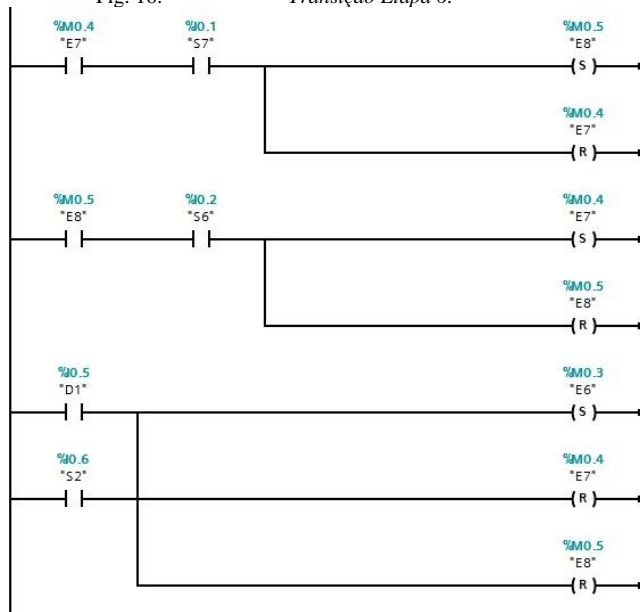


Fig. 17. Transições Etapas 7 e 8.

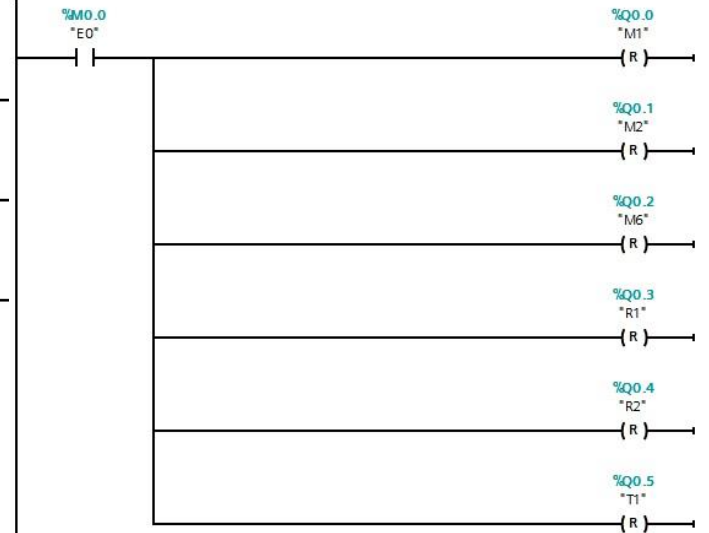


Fig. 18. Ação Etapa 0.

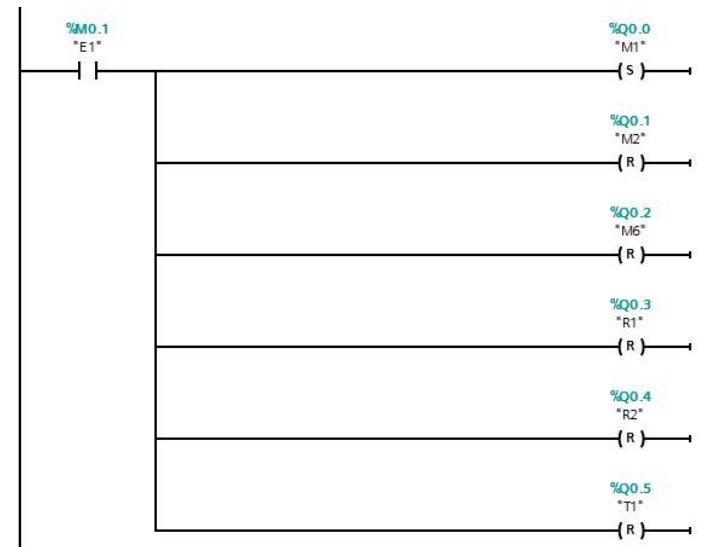


Fig. 19. Ação Etapa 1.

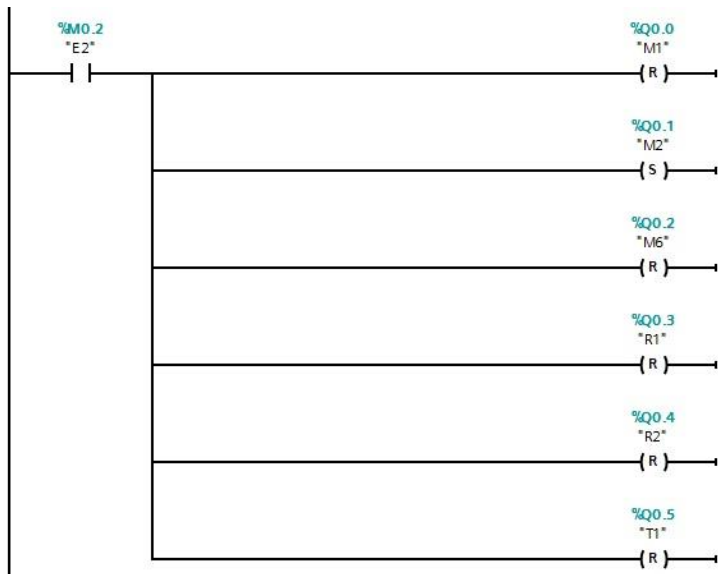


Fig. 20. Ação Etapa 2.

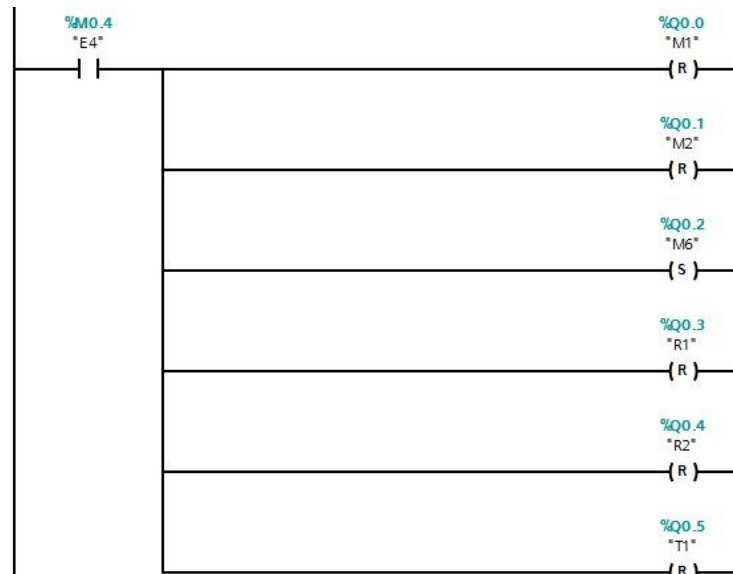


Fig. 22. Ação Etapa 4.

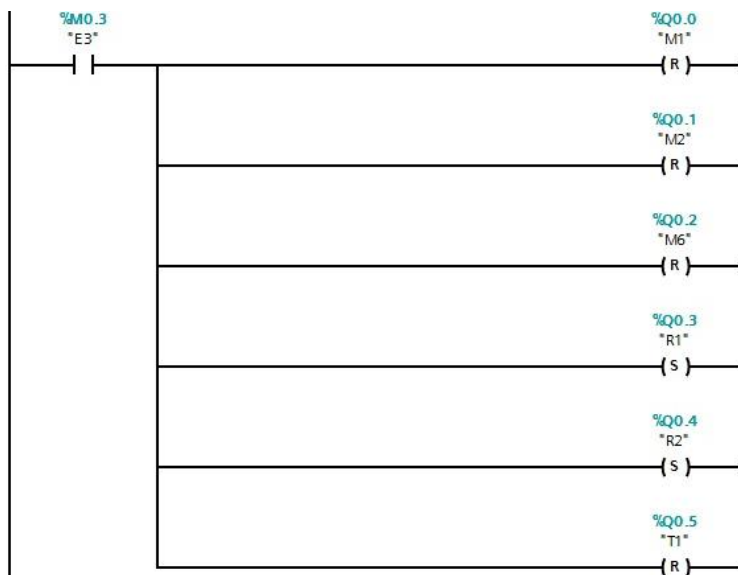


Fig. 21. Ação Etapa 3.

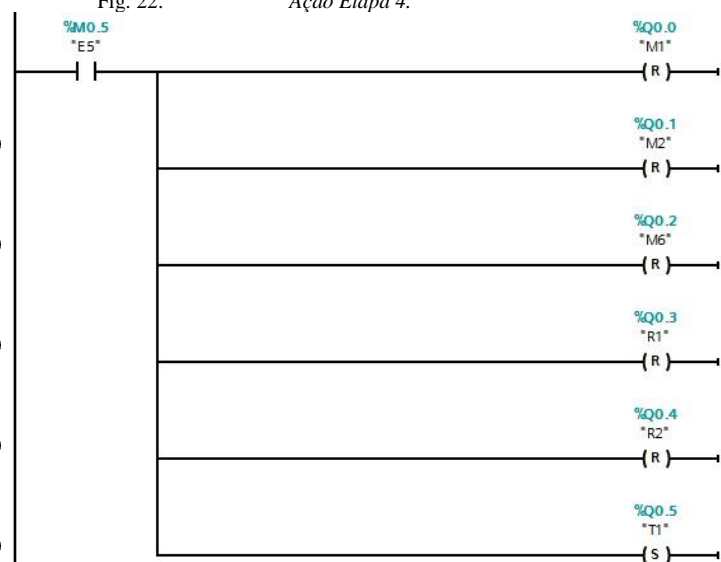


Fig. 23. Ação Etapa 5.

F. Diagrama de ações do processo do misturador

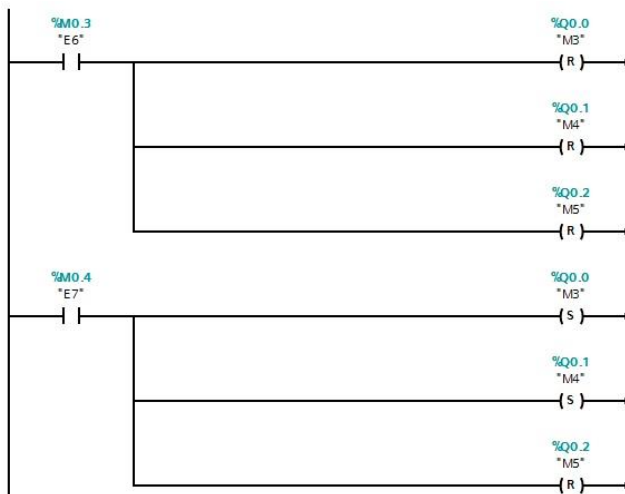


Fig. 24. Ações Etapas 6 e 7.

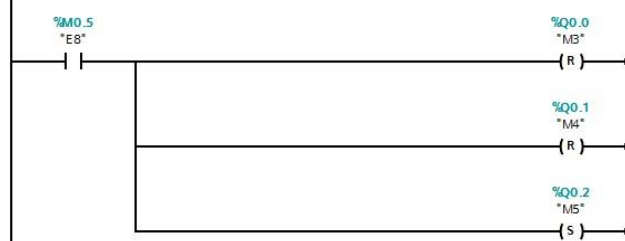


Fig. 25. Ação Etapa 8.

V. ERROS

Os possíveis erros encontrados no sistema ocorrem quando são acionados dois ou mais sensores ao mesmo tempo, visto que o processo é apenas para uma garrafa e sequencial, ou seja, cada etapa do processo ocorre somente se a etapa anterior por concluída.

Foram encontrados problemas na simulação dos sensores de proximidade em chaves no CLP, pois quando é reativada os sensores deveriam resetar e isso não acontece com as chaves.

VI. CONCLUSÃO

Por fim conclui-se que uso do controlador lógico programável no processo de fabricação de refrigerantes melhora a produção em diversos aspectos, sendo eles, comerciais, financeiros e de produção. Além disso, o CLP oferece maior confiabilidade, segurança para trabalhar dentro da empresa e com maior resistência aos fatores que é exposto dentro de uma fábrica.

Portanto com o estudo da comunicação entre os CLP's e o uso do aplicativo SIMATIC S7, possibilitou a maior aprendizagem nas linguagens de programação e as interligações de CLP's dentro de uma indústria.

VII. REFERÊNCIAS

- [1] Controlador lógico programável . Disponível em: <https://pt.wikipedia.org/wiki/Controlador_l%C3%B3gico_program%C3%A1vel>. Acesso em: 05/06/2016.
- [2] Pertruzella, F. D. Controladores lógicos programáveis. 4. ed. Porto Alegre: AMGH, 2014.
- [3] C. B. Frefonez Graziela, Fabricação de refrigerante, "Descreve o processo produtivo," Rio de Janeiro, Outubro 2012.
- [4] SIMATIC S7. Disponível em: <<http://w3.siemens.com/topics/global/en/industry/future-of-manufacturing/industry-apps/s7/Pages/simatic-s7.aspx>>. Acesso em: 06/06/2016.

Controle de Processo Industrial Genérico com interface entre CLP e Microcontrolador

André Filipe Obenaus – Júlio Joel da Costa Neto – Pedro Arthur S. P. Cogliatti

Engenharia de Controle e Automação
Universidade Federal de Santa Catarina – Campus Blumenau
Blumenau, Brasil

Resumo — Este artigo tem por fim demonstrar de maneira didática como utilizar um controlador lógico programável (CLP) em conjunto com um microcontrolador de licença livre. A montagem será utilizada para controlar um processo industrial genérico em escala reduzida, e a necessidade de utilizar o microcontrolador é para possibilitar o uso de sensores e atuadores mais baratos e amplamente disponíveis no ambiente acadêmico e o no mercado, gerando assim maquetes funcionais de conteúdos teóricos ensinados em sala de aula.

Palavras-chave — aprender fazendo; controle; CLP; ensino prático; modelagem; sistema de nível

I. INTRODUÇÃO

O ensino prático nas engenharias, além de auxiliar o aluno na compreensão dos conceitos das disciplinas, também visa simular situações reais da indústria para preparar o aluno ao mercado de trabalho, porém, existe uma lacuna entre o ambiente acadêmico e fabril devido ao fato de nem sempre ser viável reproduzir os processos estudados em um laboratório. Uma maneira de contornar esse problema é utilizando simulações em software, que modelam com grande confiabilidade casos clássicos de estudos técnicos. Mas, apesar dos grandes avanços em softwares de modelagem e simulação, eles ainda não substituem totalmente o “aprender fazendo”, de modo que a construção de modelos reais e funcionais continua sendo o método que melhor preenche a lacuna mencionada anteriormente.

Este artigo visa proporcionar uma base teórica e prática para a construção de um sistema de nível de tanque, processo clássico da indústria, descrito por [1]. Na Seção II estão apresentados o desenvolvimento teórico, equações, imagens e definições que serviram de base para o trabalho. Na Seção III, é listado o material utilizado, e suas características. A Seção IV contém as informações referentes ao desenvolvimento prático e simulação do processo. O artigo é concluído na Seção V.

II. DESENVOLVIMENTO TEÓRICO

O processo desenvolvido pode ser dividido em duas partes. Uma envolve o problema de controle de nível de líquido em um tanque, onde é necessário descrever matematicamente o comportamento do tanque e do controle que aciona uma bomba responsável pela vazão de entrada no sistema. A outra é implementada no CLP, que dá o comando de início para o controle, além de ativar outras etapas do processo descrito a seguir.

A. Modelo Matemático da Planta de Controle

Em análise de sistemas que envolvem fluxo de fluídos, é necessário dividir os regimes de fluxo em regime laminar e regime turbulento, de acordo com o valor do número de Reynolds (Re) [1].

Para $Re > 4000$, o sistema é turbulento. Para valores de $Re \leq 2300$, o regime é considerado laminar. A região intermediária é considerada de transição, onde ambos os regimes são possíveis, dependendo, entre outros fatores, de rugosidade do tubo e uniformidade do fluxo.

A importância dessa distinção é que fluxos laminares podem ser representados por equações diferenciais lineares, ao passo que fluxos turbulentos são representados por equações diferenciais não lineares.

Processos industriais, de maneira geral, têm escoamentos turbulentos, porém, se a região de operação for limitada, as equações que descrevem esses processos podem ser linearizadas. Utilizam-se conceitos de resistência e capacitância para sistemas de nível de líquido que permite descrever suas características dinâmicas de modo simples [1]. A Fig. 1 descreve o diagrama de funcionamento do sistema:

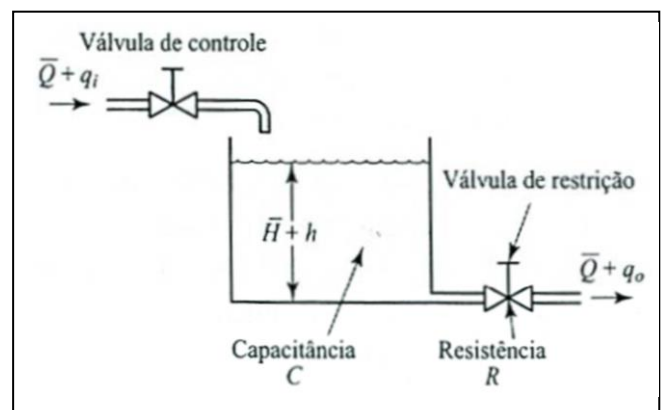


Fig. 1. – Diagrama de sistema de nível de primeira ordem [1].

Onde, para em regime permanente (RP):

\bar{Q} = vazão em volume em RP

q_i = desvio da vazão de entrada em relação ao RP

q_o = desvio da vazão de saída em relação ao RP

\bar{H} = altura do nível em RP

h = desvio de nível a partir do seu valor em RP

Considerando um fluxo ao longo de uma tubulação curta ou através de uma restrição, a resistência R é definida como a variação na diferença de nível necessária para causar a variação unitária na vazão, isto é:

$$R = \frac{\text{variação na diferença de nível (m)}}{\text{variação na vazão em volume } \left(\frac{\text{m}^3}{\text{s}}\right)} \quad (1)$$

Para um fluxo turbulento no regime permanente, a resistência R_t é aproximadamente:

$$R_t = \frac{2\bar{H}}{\bar{Q}} \quad (2)$$

Esta relação é verdadeira para a condição de operação nas proximidades de um ponto P. Se definirmos uma pequena variação a altura de RP como h e a pequena variação correspondente da vazão como q , a validade da equação de R_t pode ser vista na Fig. 2 abaixo.

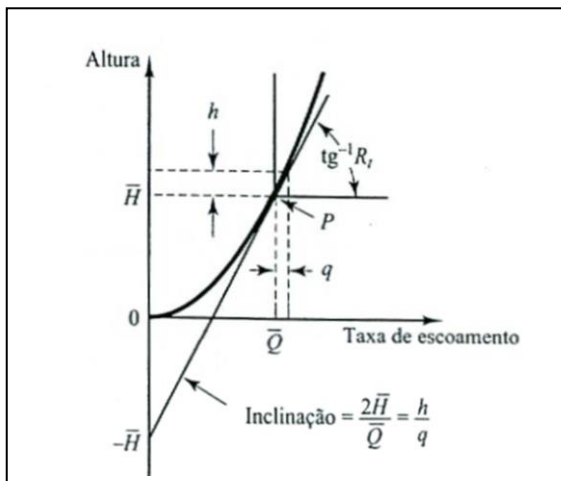


Fig. 2. – Curva real de operação [1].

Como se pode observar, em um ponto P de operação, se as variações de altura e vazão forem suficientemente pequenas, a inclinação da curva real no ponto é igual à inclinação da reta tangente ao ponto.

A capacitância C de um reservatório é definida como a variação na quantidade de líquido armazenada necessária para causar a mudança unitária no potencial (altura), então,

$$C = \frac{\text{variação no volume armazenado (m}^3\text{)}}{\text{variação na altura (m)}} \quad (3)$$

A capacitância do reservatório é igual à sua seção transversal. Se esta for constante, a capacitância será constante para qualquer altura do nível. Para o caso de um tanque cilíndrico, a área A da seção transversal é,

$$A = C = \frac{\pi D^2}{4} \quad (4)$$

Onde D é o diâmetro do tanque.

Agora que definimos as variáveis do sistema, e restringimos sua condição de operação para um regime linearizado, podemos descrever seu comportamento com a equação a seguir:

$$C dh = (q_i - q_o) dt \quad (5)$$

Da equação (1):

$$q_o = \frac{h}{R} \quad (6)$$

Para um sistema linear, R é constante e reescrevemos (5) como:

$$RC \frac{dh}{dt} + h = Rq_i \quad (7)$$

Aplicando a transformada de Laplace em ambos os membros da equação e considerando condições iniciais nulas, obtemos:

$$(RCs + 1)H(s) = RQ_i(s) \quad (8)$$

Para q_i como entrada e h como saída, a função de transferência do sistema é:

$$\frac{H(s)}{Q_i(s)} = \frac{R}{RCs + 1} \quad (9)$$

B. O Controle Proporcional-Integral

O método escolhido para efetuar o controle do sistema é o proporcional-integral (PI), formado por duas componentes. A componente proporcional depende apenas da diferença entre o ponto de ajuste e a variável de processo. Esta diferença é referida como o termo de erro. O ganho proporcional determina a taxa de resposta de saída para o sinal de erro. Em geral, aumentando o ganho proporcional irá aumentar a velocidade da resposta do sistema de controle. No entanto, se o ganho proporcional é muito grande, a variável de processo começará a oscilar. Se o ganho é aumentado ainda mais, as oscilações ficarão maiores e o sistema ficará instável e poderá oscilar até mesmo fora de controle. A componente integral soma o termo de erro ao longo do tempo. O resultado é que mesmo um pequeno erro fará com que a componente integral aumente lentamente. A resposta integral irá aumentando ao longo do tempo a menos que o erro seja zero, portanto, o efeito é o de conduzir o erro de estado estacionário para zero. O regime permanente de erro é a diferença final entre as variáveis do processo e da referência [8].

A equação de um controle PI no domínio da frequência pode ser escrita como:

$$G(s) = K_p + \frac{K_i}{s} \quad (10)$$

Onde K_p é o ganho proporcional e K_i o ganho do integrador.

C. Mapa de Entradas e Saídas e Diagrama de Estados

O processo tem seu início definido por comandos dados ao CLP, que inicia a rotina de controle.

Antes de implementar o código, porém, é importante mapear as entradas e saídas do sistema, bem como construir seu diagrama de estados, de modo a ter o processo bem documentado. O mapeamento de entradas e saídas pode ser visto na Tabela 1, sendo que a letra em frente ao rótulo indica se a porta usada é digital ou analógica.

TABELA 1 – Mapa de entradas e saídas

Entradas			Saídas		
D	B1	Botão de partida	D	M1	Bomba
D	BE	Botão de emergência	D	HE	Aquecedor
A	SD	Sensor Ultrassônico	D	M2	Aerador
D	BD	Botão de desligar	D	LUV	Luz UV
			A	SPoint	Valor de referência
			D	AL	Alarme

O diagrama de estados finito é encontrado na Fig. 3.

III. MATERIAIS

A. O Controlador Lógico Programável

O Controlador Lógico Programável (CLP) é um equipamento composto de componentes eletrônicos, memória programável ou não programável que contém dados e programas com a finalidade de ler e executar instruções, interagindo com um sistema que deve ser controlado, através de dispositivos de input e output do tipo digital ou analógico (norma IEC 61131-2).

O CLP é um computador que realiza funções de controle em vários níveis de complexidade. O aspecto interessante do CLP é que ele pode ser programado e utilizado por pessoas sem um grande conhecimento no uso do computador. É projetado para trabalhar em ambiente industrial com variação de temperatura, umidade, vibrações, distúrbios elétricos e outras variantes que existem no meio industrial [5].

O CLP utilizado é o SIMATIC S7-1200 da marca Siemens, disponível para utilização dos acadêmicos no LABCID – Laboratório de Circuitos e Sistemas Digitais.

B. O Microcontrolador e Placa de Prototipagem

O microcontrolador é um circuito integrado encapsulado e destinado ao processamento de sinais digitais. Em geral, possuem todos os periféricos necessários em um único chip. Seu tamanho é reduzido, mesmo contendo vários periféricos, como memória, barramento, *timers*, portas de comunicação, conversores analógico-digital, etc. Possuem desempenho menor que os microprocessadores, mas são ideais em aplicações que necessitam de menores dimensões, tempo e custos. É amplamente utilizado em automação industrial, residencial e predial, eletrodomésticos, brinquedos, eletrônicos, e em qualquer situação onde seja necessário o controle de um dispositivo de sinais eletrônico [10].

O microcontrolador utilizado é o ATmega328, instalado na placa de prototipagem Arduino UNO.

Arduino é uma plataforma de prototipagem de código aberto baseado em hardware e software *easy-to-use*. Placas Arduino são capazes de ler entradas e transformá-las em uma ou mais saídas.

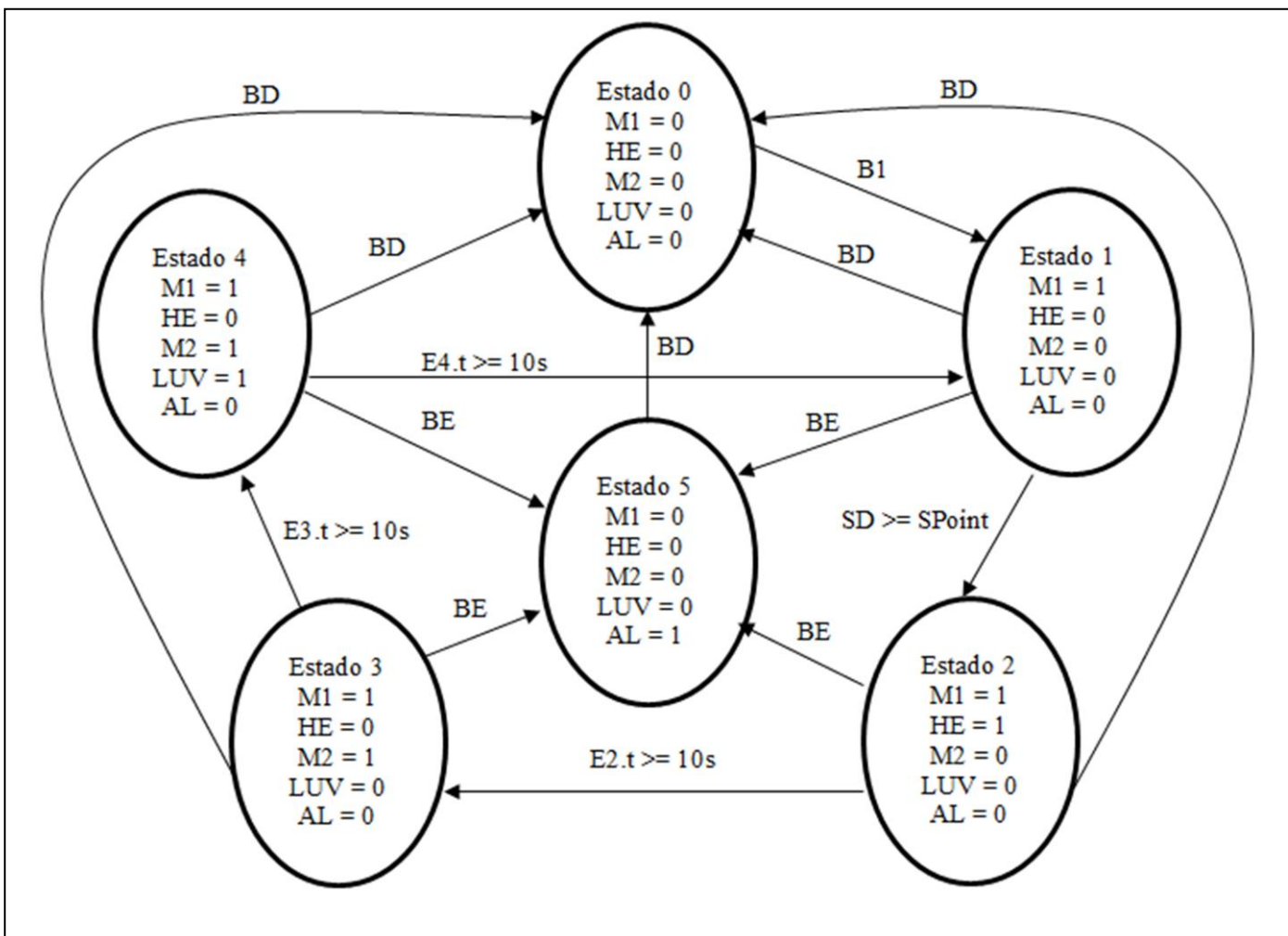


Fig. 3 – Diagrama de máquina de estados finitos.

Para fazer isso se usa a linguagem de programação Arduino (baseada em conexões por fios), e o software Arduino (IDE), com base em processamento.

As placas Arduino são completamente *open-source*, capacitando os usuários para construí-las de forma independente e adaptá-las às suas necessidades específicas. O software também é *open-source* [9].

C. Motobomba e Sensor de Nível

A bomba utilizada para encher o tanque é uma bomba de gasolina para automóveis, de 12 volts CC.

A rotação do motor é controlada com uma ponte H, baseado no chip L298N, construído para controlar cargas indutivas como relés, solenoides, motores DC e motores de passo.

O sensor usado para medir o nível é o sensor ultrassônico HC-SR04, comumente utilizado com a plataforma Arduino.

Quando um pulso de disparo é enviado para a entrada *trigger*, o sensor emite uma rajada de oito sinais ultrassônicos de 40 kHz, e o tempo de eco desses sinais é proporcional à distância de medição. Parâmetros do sensor são encontrados na Tabela II.

TABELA II – Parâmetros do sensor

Tensão de operação	5 V CC
Corrente de operação	15 mA
Frequência de operação	40 Hz
Alcance máximo	4 m
Alcance mínimo	2 cm
Ângulo de medição	15 graus
Sinal de disparo de entrada	10 μ s pulso TTL
Sinal de eco de saída	Sinal TTL de entrada e o alcance proporcional
Dimensões	45x20x15 mm

IV. DESENVOLVIMENTO PRÁTICO

Uma foto do sistema construído pode ser vista na Fig. 4.

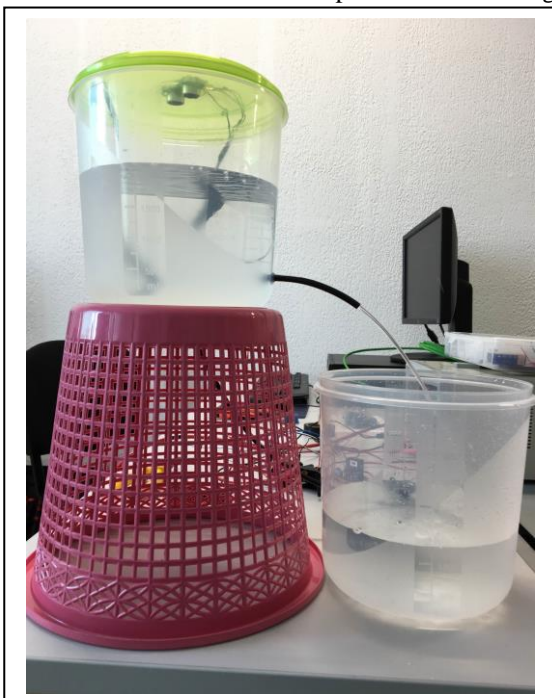


Fig. 4 – Tanque de controle e reservatório.

A. Parâmetros C e R_t

Utilizando um pote plástico cilíndrico de quatro litros como o tanque e outro como reservatório, podemos definir os parâmetros da planta, conforme visto na seção II.

O parâmetro C foi definido como sendo a área da seção do tanque. Tendo o diâmetro de 18 cm então:

$$C = 25,45 * 10^{-3} m^2 \quad (11)$$

A vazão de saída e regime permanente pode ser definida como:

$$\bar{Q} = C_d \cdot A_o \cdot \sqrt{2 \cdot g \cdot \bar{H}} \quad (12)$$

Onde A_o é a área do tubo de vazão e C_d é o coeficiente de descarga, uma constante adimensional obtida experimentalmente.

Portanto, R pode ser calculado como:

$$R_t = \frac{2\bar{H}}{C_d \cdot A_o \cdot \sqrt{2 \cdot g \cdot \bar{H}}} \quad (13)$$

A saída do tanque é feita por um tubo de 10 cm de comprimento por 4 mm de diâmetro, sendo classificado como um tubo curto.

Considera-se um tubo curto aquele em que seu comprimento (L) supera três vezes o seu diâmetro e fica abaixo de quinhentas vezes o seu diâmetro.

Da referência [2], obtemos valores práticos de C_d que para $L/D = 25$, $C_d = 0,715$.

Com $\bar{H} = 10$ cm;

$$R_t = 15891,5 \frac{s}{m^2} \quad (14)$$

A função de transferência fica:

$$\frac{H(s)}{Q_i(s)} = \frac{15891,5}{404,44s + 1} \quad (15)$$

O diagrama de blocos para o sistema de controle é apresentado na Fig. 5.

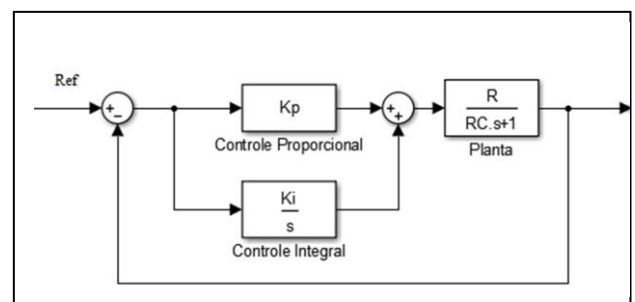


Fig. 5 – Diagrama de blocos do controle e planta.

O atuador tem uma resposta muito mais rápida do que a planta para que seja possível efetuar um controle, ou seja, seus polos estão muito afastados da origem, reduzindo o sua influencia na resposta total do sistema. Por este motivo, vamos desconsiderar a parcela de ganho do atuador, conforme o esquema da Fig. 5.

B. Controle com Arduino

a) O microcontrolador foi programado utilizando IDE próprio da plataforma, que utiliza linguagem C, e o código de controle utilizado foi o de uma biblioteca PID encontrada em [3], onde zeramos o termo relacionado ao controle diferencial.

A determinação dos termos K_i e K_p foi feita utilizando a ferramenta computacional Matlab. O requisitos para o ajuste foi de menor tempo de subida sem que haja *overshoot*. Devido à natureza de medição do sensor, convertemos os valores de C e R_t para cm^2 e s/cm^2 , e usando a opção de ajuste do software, obtivemos:

$$K_i = 0,5 \quad (16)$$

$$K_p = 45 \quad (17)$$

A simulação feita no computador da resposta no tempo do sistema é apresentada na Fig. 6.

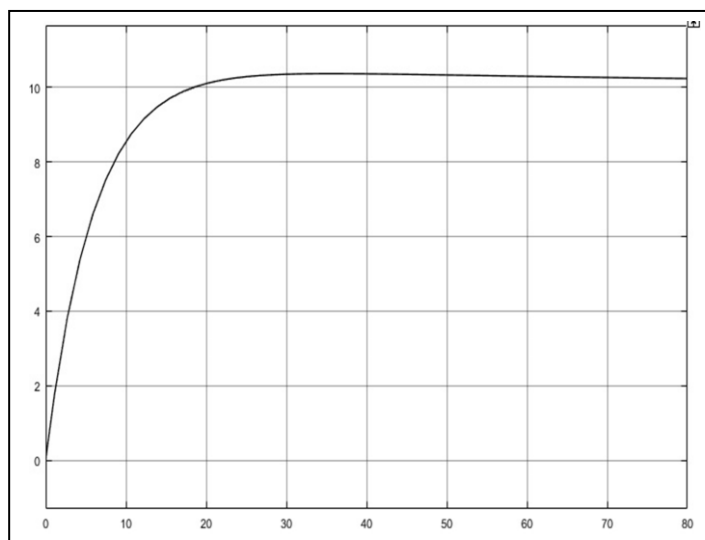


Fig. 6 – Gráfico da resposta do sistema.

O modelo físico atinge a altura de referência no tempo previsto pela simulação, em torno de vinte segundos, e leva um tempo mais longo para estabilizar, o que também era esperado.

C. Controle no CLP

Três linguagens foram utilizadas na programação do CLP, a linguagem Ladder, texto estruturado e FBD (diagramas de blocos). Todas estão definidas na norma IEC 61131-3.

A linguagem Ladder foi criada para melhorar a aceitação do CLP no mercado, pois emula a lógica tradicional de controle que era representada por diagrama de relés. É uma linguagem gráfica de alto nível que representa um circuito elétrico [4]. Foi a linguagem utilizada para programar as principais partes do diagrama de estados da Fig. 3.

O FBD é uma linguagem gráfica onde os elementos aparecem interligados como em circuitos elétricos. Usa

blocos padrão semelhantes aos do Ladder e permite aos usuários construir seus próprios blocos de função de acordo com a exigência da aplicação, o que se chama de encapsulamento.

O bloco pode ser programado em qualquer outra linguagem de programação [4]. Esta linguagem foi utilizada para converter valores em centímetros para valores de tensão, para comunicar o valor de referência para o Arduino e também fazer a leitura dos valores medidos e converter em unidades convenientes de se apresentar ao usuário.

O texto estruturado, ou em inglês *structured text* (ST), é uma linguagem de texto de alto nível, similar a BASIC e PASCAL. Permite a implementação de procedimentos complexos, difíceis de representar em linguagens gráficas. Possibilita o uso de laços de repetição (*while-do*) e de condições (*if-then-else*) [4]. Essa linguagem foi utilizada para fazer uma média das medições enviadas pelo sensor, para reduzir o efeito de erros de precisão e ruído.

Além da programação, o CLP permite criar uma interface homem-máquina (IHM), que pode ser implementado em uma tela do tipo *touchscreen*, onde é possível acompanhar e controlar parâmetros que o programador julgue necessário. A interface criada pode ser vista na Fig. 7.

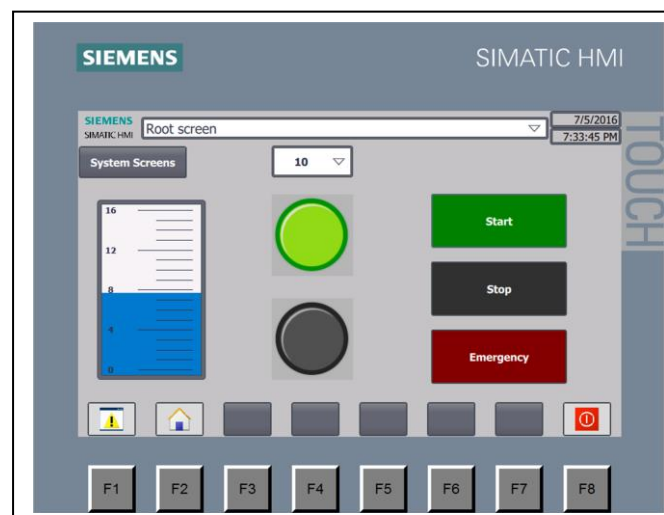


Fig. 7 – IHM com indicador de nível e LED de operação ativado.

Além da IHM, também foi desenvolvida a aplicação *webserver*, que permite controlar o processo através de um computador com acesso a mesma rede do CLP. Essa funcionalidade cria uma página em HTML, na qual o programador pode liberar ou restringir o acesso de usuários mediante o uso de senha, assim como liberar ou restringir o acesso e controle de parâmetros e informações conforme a necessidade.

As páginas da aplicação podem ser vistas nas Fig. 8 e Fig. 9.

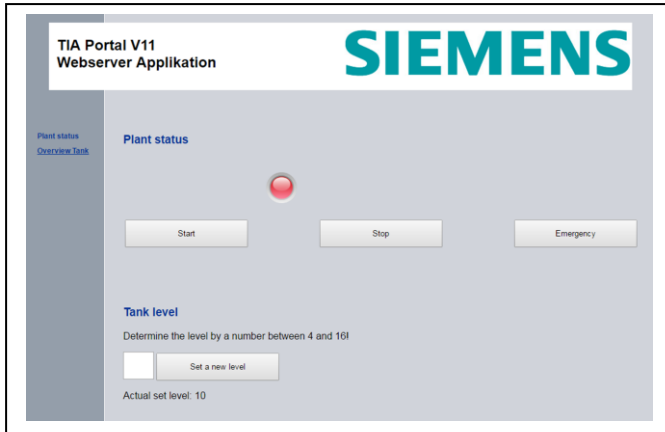


Fig. 8 – Página de comando do processo.

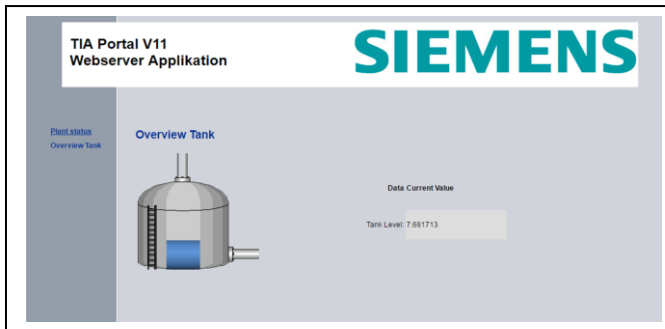


Fig. 9 – Página de visão geral do processo.

V. CONCLUSÃO

Durante a montagem do projeto, muitas dúvidas de natureza prática surgiram, e foram posteriormente solucionadas através de pesquisa e ajuda do corpo docente. O controle PI foi implementado com sucesso, apesar de pequenas variações causadas pela precisão do sensor. O controle do CLP operou conforme projeto, sem ressalvas. O artigo avançou o entendimento dos autores nos conteúdos aqui abordados, trazendo sentido prático para conteúdos antes muito abstratos. A sugestão para trabalhos futuros é usar o controle PI industrial do CLP, que apresenta melhor desempenho, utilizando o Arduino apenas como interface para os sensores.

AGRADECIMENTOS

Os autores agradecem o corpo docente da Engenharia de Controle e Automação – UFSC – Campus Blumenau, pelos recursos e apoio prestados ao desenvolvimento deste projeto.

REFERÊNCIAS

- [1] K. Ogata, Engenharia de Controle Moderno, 5ª ed. São Paulo: Pearson Prentice Hall, 2010, pp 92-95.
- [2] J. M. Azevedo Netto, Manual de Hidráulica, 8ª ed. São Paulo: Edgar Blüchler, 1998, pp.84-85.
- [3] (<http://playground.arduino.cc/Code/PIDLibrary>) PID Library, acessado em 06/07/2016.

- [4] M.R. Petry – Notas de aula - Informática Industrial, 2016.
- [5] F. Prudente, PLC S7-1200 – Teoria e Aplicações, 1ª ed. Rio de Janeiro: LTC, 2014.
- [6] B.P. Lathi, Sinais e Sistemas Lineares, 2ª ed. Porto Alegre: Bookman, 2008.
- [7] G.F. Franklin; J.D. Powell; A. Emami-Naeini. Sistemas de Controle para Engenharia. 6. ed. Porto Alegre: Bookman, 2013.
- [8] (<http://www.ni.com/white-paper/3782/pt/>) National Instruments - Explicando a Teoria PID, acessado em 12/07/2016.
- [9] (<https://www.arduino.cc/en/Guide/Introduction>) What is Arduino, acessado em 12/07/2016.
- [10] A.S. de Oliveira, F. S. de Andrade, Sistemas Embarcados: hardware e firmware na prática, 2ª ed. São Paulo: Érica, 2010, pp 34.

Modelo de Sistema de Controle para Motor DC

Concepção e Simulação

Nikolas Sbaraini Hammes
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina
Blumenau, Brasil

Rodrigo Lucas Weber
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina
Blumenau, Brasil

Resumo—Esse trabalho tem como objetivo o aprendizado por meio da simulação de um sistema controlador PI de motor DC. Onde iremos demonstrar o par entrada-saída utilizando simulações em software Matlab e planta física.

Palavras-chave—controle, motor, realimentado.

I. INTRODUÇÃO

Esse trabalho, que foi nos passado na disciplina de Sinais e Sistemas Lineares, pretende colocar em prática os conhecimentos adquiridos em sala. Iremos demonstrar o par entrada-saída de um sistema de primeira ordem, e para isso escolhemos representar na forma de equações e fisicamente o motor DC. Utilizaremos para essa simulação o software matlab, o controlador arduino e a planta do motor DC. Veremos qual a resposta para algumas entradas diferentes, de que forma as constantes do motor interferem nas especificações de controle e como o sistema reage às perturbações.

O trabalho físico foi montado a partir de uma placa arduino, conectada diretamente como fonte de alimentação de um dos motores DC. O eixo do rotor do motor alimentado foi então acoplado ao eixo do rotor do segundo motor DC através de um pequeno tubo termoretrátil. À medida em que motor alimentado rotacionava, o segundo motor rotacionava à mesma velocidade. Então, colocamos os conectores do segundo motor DC como entrada no arduino, para então podermos analisar a saída do sistema através da energia gerada pela velocidade de rotação do motor.

II. COMPONENTES DO TRABALHO

A. Arduino Uno

Utilizamos a placa arduino para o fim de alimentação do primeiro motor DC, ou seja, como fonte do sinal de entrada para o sistema.

Utilizamos o arduino para também receber o sinal de saída do sistema, ou seja, o sinal gerado pela rotação do segundo motor.

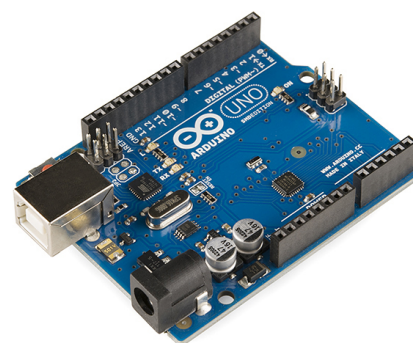


Fig. 1. Exemplo do modelo de arduino utilizado.

B. Motores DC

O primeiro motor DC foi usado para receber o sinal de alimentação do sistema. O segundo motor teve o eixo de rotação do seu rotor ligado ao eixo de rotação do rotor do primeiro motor, a fim de que rotacionassem à mesma velocidade.

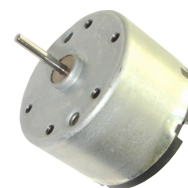


Fig. 2. Motor DC retirado de um leitor de DVD.

C. Tubo Termoretrátil

Utilizamos o tubo termoretrátil para fazer a conexão entre os eixos dos rotores dos dois motores DC.



Fig. 3. Tubo termoplástico.

III. REPRESENTAÇÃO DO MOTOR DC

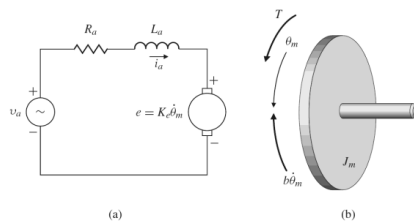


Fig. 4. Representação gráfica de um motor DC.

A planta do motor DC pode ser representada pela seguinte equação diferencial, como visto em [1]:

$$J_m \ddot{\theta}_m + \left(b + \frac{K_t K_e}{R_a} \right) \dot{\theta}_m = \frac{K_t}{R_a} v_a \quad (1)$$

Onde o par entrada-saída é dado pela posição angular $\theta_m(s)$ como saída e pela tensão de entrada $v_a(s)$ como entrada, com a seguinte função de transferência:

$$\frac{\Theta_m(s)}{V_a(s)} = \frac{\frac{K_t}{R_a}}{J_m s^2 + \left(b + \frac{K_t K_e}{R_a} \right) s} \quad (2)$$

Definindo K e τ , respectivamente, como o ganho estático e a constante de tempo, temos as seguintes relações:

$$K = \frac{K_t}{b R_a + K_t K_e} \quad (3)$$

$$\tau = \frac{R_a J_m}{b R_a + K_t K_e} \quad (4)$$

Simplificando a função de transferência, temos então que ela é igual a:

$$= \frac{K}{s(\tau s + 1)} \quad (5)$$

Por fim, a partir do modelo acima, construímos a função de transferência entre a velocidade $s.\theta_m(s)$ e a tensão de entrada $v_a(s)$, em que $s.\theta_m(s)$ é a saída e $v_a(s)$ é a entrada:

$$\frac{\Omega(s)}{V_a(s)} = s \frac{\Theta_m(s)}{V_a(s)} = \frac{K}{\tau s + 1} \quad (6)$$

IV. SIMULAÇÃO DO SISTEMA EM MALHA ABERTA

Inicialmente, utilizamos para a simulação a resistência interna do motor (R_a), a inércia do motor (J_m), a constante de torque (K_t), a constante de força eletromotriz (K_e), e o coeficiente de atrito viscoso (b), como exemplificado em [1]:

$$\begin{aligned} R_a &= 10 \Omega & J_m &= 0,01 \text{ Kg} \cdot \text{m}^2 \\ K_t &= 0,02 \text{ N} \cdot \text{m/A} & K_e &= 0,02 \text{ V} \cdot \text{s} \\ b &= 0,001 \text{ N} \cdot \text{m} \cdot \text{s} \end{aligned} \quad (7)$$

Com tais valores, podemos calcular o K e o τ :

$$K = \frac{0,02}{0,001 + 0,02 \cdot 0,02} = 1,923077 (\text{V} \cdot \text{s})^{-1} \quad (8)$$

$$\tau = \frac{10 \cdot 0,01}{0,001 \cdot 10 + 0,02 \cdot 0,02} = 9,615385 \text{ s} \quad (9)$$

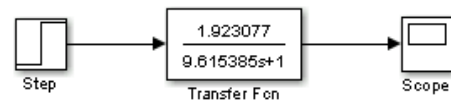


Fig. 5. Diagrama de blocos da simulação em malha aberta.

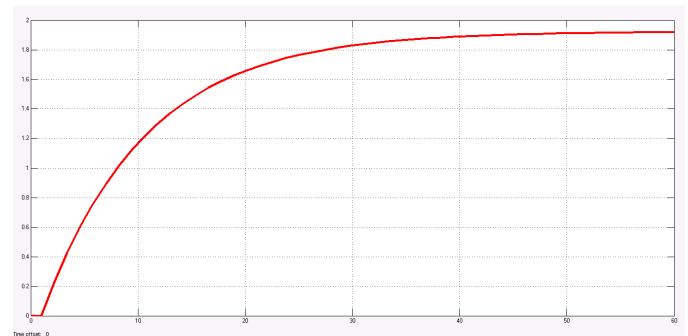


Fig. 6. Gráfico da simulação em malha aberta.

V. SIMULAÇÃO DO SISTEMA EM MALHA FECHADA

Como o sistema exemplificado pelo livro [1] leva um tempo muito longo para atingir o valor final, assim o colocamos em uma malha de controle PI a fim de diminuir esse tempo de subida:

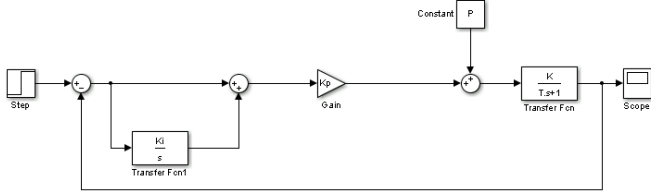


Fig. 7. Diagrama de blocos da simulação em malha fechada.

E equacionando o sistema em malha fechada, obtemos:

$$\frac{K_p K S + K_p K_i K / \tau}{S^2 + (\frac{1 + K_p K}{\tau}) S + \frac{K_i K_p K}{\tau}} \quad (10)$$

Especificamos, então, nossos parâmetros de controle, com o tempo de subida menor ou igual a 7 segundos, e a máxima ultrapassagem como sendo menor ou igual a 8%:

$$t_r \leq 7s \Rightarrow \omega_n \leq \frac{1,8}{t_r} \Rightarrow \omega \leq 0,257143 \quad (11)$$

$$M_p \leq 0,08 \Rightarrow \xi \geq \frac{-\ln(M_p)}{\sqrt{\pi^2 + (\ln(M_p))^2}} \Rightarrow \xi \geq 0,626577 \quad (12)$$

A partir dos valores acima, pudemos calcular os ganhos do controlador PI (Ki e Kp), utilizando da relação entre a equação do sistema em malha fechada e a equação abaixo [2]:

$$\frac{\omega_n^2}{S^2 + 2\xi\omega_n S + \omega_n^2} \quad (13)$$

$$\frac{K_i K_p K}{\tau} = \omega_n^2 \Rightarrow K_i = 0,302981 \quad (14)$$

$$\frac{1 + K_p K}{\tau} = 2\xi\omega_n \Rightarrow K_p = 1,091198 \quad (15)$$

Colocamos, então, os valores de Ki e Kp no diagrama de blocos da simulação, como visto abaixo:

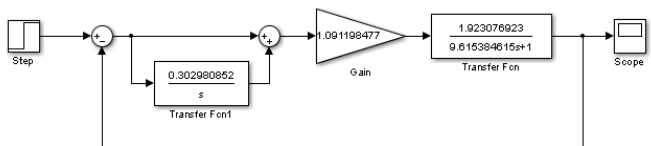


Fig. 8. Diagrama de blocos com os ganhos Ki e Kp calculados.

Simulando o sistema acima, obtemos:

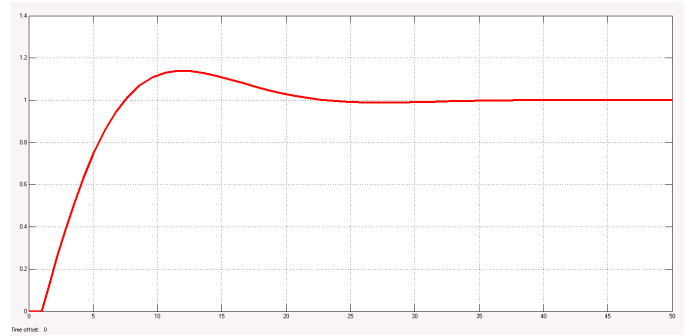


Fig. 9. Gráfico da simulação em malha fechada.

Analisando o gráfico da Fig.9, pode-se observar que o tempo de subida e a máxima ultrapassagem não ficaram de acordo com nossas especificações, e isso se dá porque há um zero no sistema que influencia esses dois parâmetros. Assim, tivemos que assumir valores de Ki e Kp de modo a atingir as especificações desejadas:

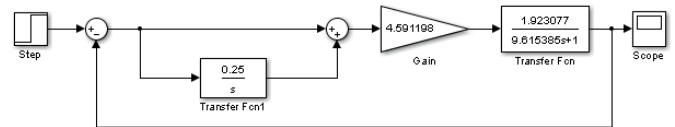


Fig. 10. Diagrama de blocos com os ganhos Ki e Kp ajustados.

Simulando o sistema com os valores de Ki e Kp ajustados, obtemos então:

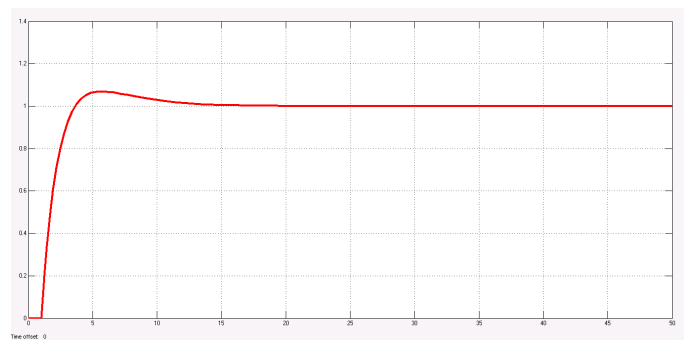


Fig. 11. Simulação com os valores de Ki e Kp ajustados.

E com o modelo de controle PI ajustado, obtemos então as especificações desejadas, tanto em relação ao tempo de subida, quanto em relação à máxima ultrapassagem.

VI. SISTEMA REAL EM MALHA ABERTA

A partir do gráfico de malha aberta (Fig. 14) percebemos que ao ser definido uma entrada de $125 \cdot 10^{-2}$ volts a saída foi, em média, igual a $63 \cdot 10^{-2}$ volts mostrando que o ganho estático do motor é aproximadamente 0,5 ($K=0,5$). E utilizando o gráfico e o tempo da tabela de valores vemos que se leva 0,45 segundos para o valor sair de 0 e alcançar seu valor médio. Para calcular a constante de tempo tau utilizaremos a resposta ao degrau unitário que em qualquer sistema linear invariante no tempo é dada por:

$$s(t) = \int_{-\infty}^t h(t') dt' \tag{16}$$

Nosso $h(t')$ é igual à transformada inversa de Laplace da nossa função de transferência:

$$h(t') = L^{-1}\left(\frac{K}{\tau s + 1}\right) = \frac{K}{\tau} e^{-\frac{t'}{\tau}} \tag{17}$$

Logo nossa resposta será:

$$s(t) = (1 - e^{-\frac{t}{\tau}}) u(t) \tag{18}$$

Dessa forma quando nosso $s(t)$ variar de 10% a 90% do valor final nosso t irá variar em aproximadamente $2,2\tau$. Logo podemos achar o tau fazendo a variação do tempo para $s(t)$ variar de 10% a 90% e dividimos esse valor por 2,2.

$$t(90\%) - t(10\%) = 2,2 \tau \Rightarrow \tau = \frac{0,435 - 0,020}{2,2} = 0,2 \tag{19}$$

E podemos observar que o valor aproximado de tau está bom pela simulação no matlab, onde entramos com um degrau de valor final 12:

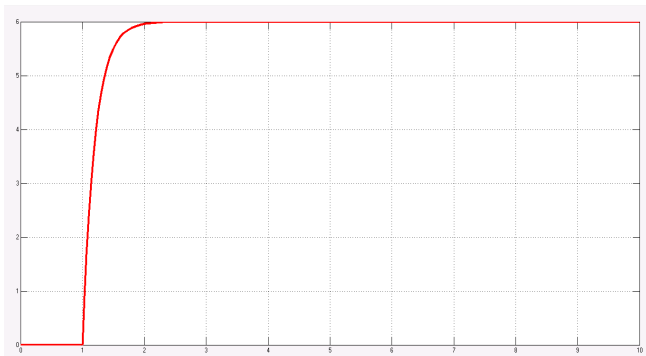


Fig. 12. Gráfico de simulação.



Fig. 13. Diagrama de blocos.

E esse gráfico é aproximadamente parecido com o nosso gráfico obtido a partir do arduino, onde entramos com um degrau de valor final 125 e estabilizou em média no valor 63:

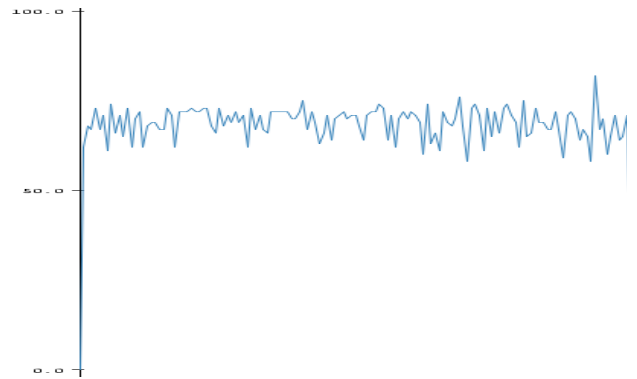


Fig. 14. Gráfico dos valores lidos pelo arduino.

Assim podemos simular uma malha de controle fechado no software matlab, onde definimos $K_p=1$ e $K_i=0,5$, e usamos esses mesmos valores no programa de controle PI do arduino:

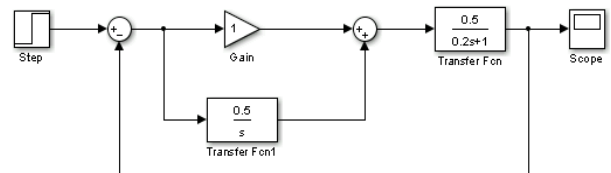


Fig. 15. Diagrama de blocos.

Obtendo os gráficos em matlab e arduino:

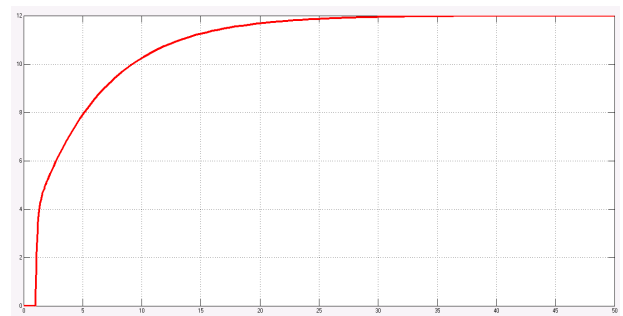


Fig. 16. Gráfico de simulação.

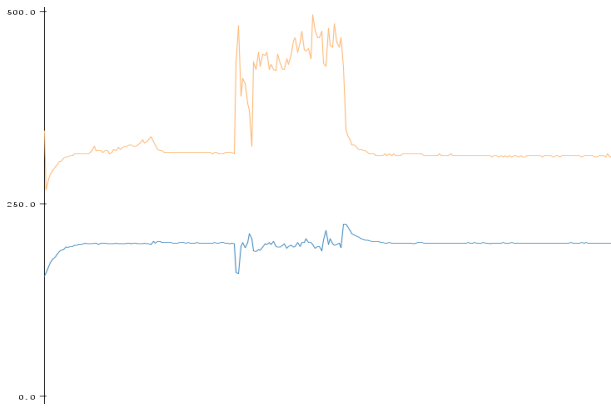


Fig. 17. Gráfico dos valores lidos pelo arduino.

Assim observamos que os gráficos estão parecidos, a diferença se dá pois no sistema real a uma perda e um erro fazendo com que o valor de entrada seja diferente da saída. Já na simulação o valor da entrada é o mesmo da saída, pois não

há essas perdas. Pode-se observar também que a uma variação no gráfico da saída do arduino (gráfico azul), isso se deu por conta da perturbação que impusemos ao sistema, nesse momento que há a perturbação pode-se observar que o gráfico da entrada (gráfico laranja) aumenta a entrada para voltar para o mesmo valor de controle.

REFERÊNCIAS

- [1] FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A.. Sistemas de controle para engenharia. 6. ed. Porto Alegre: Bookman, 2013. 702 p. Tradução de: Fernando de Oliveira Souza. Revisão Técnica: Antonio Pertence Júnior.
- [2] LATHI, B. P.. Sinais e sistemas lineares. 2. ed. Porto Alegre: Bookman, 2007. 856 p. Tradução de: Gustavo Guimarães Parma.
- [3] ALMEIDA, L. B.. "Resposta no tempo de sistemas de primeira e de segunda ordem só com pólos," 2011. [Online]. Disponível: <http://users.isr.ist.utl.pt/~aguilar/respostas%20no%20tempo.pdf>. Acessado: Jul. 3, 2016.

Uma breve descrição da rede PROFINET

Juliano Emir Nunes Masson
 Engenharia de Controle e Automação
 Universidade Federal de Santa Catarina
 Blumenau, Brasil
 juliano.masson@grad.ufsc.br

Resumo—Este trabalho apresenta uma breve explicação sobre o protocolo PROFINET e suas características, mostrando os tipos de PROFINET e suas especificidades. Explica os modos de comunicação presentes no PROFINET, assim como as características de cada um, e apresenta um breve resumo sobre a camada física, os conectores suportados e a segurança da rede.

Palavras-chave— *PROFINET; Redes Industriais; Ethernet Industrial; Automação*

I. INTRODUÇÃO

Existe a muito o desejo da interconexão das redes em uma empresa, fazendo com que a gerência possa ter conhecimento em “tempo real” do que acontece em todas as camadas, desde a quantidade de vendas na filial mais distante, até a quantidade de matéria-prima no estoque da fábrica localizada em outro país. A primeira parte do problema já está solucionada, pois as redes dos níveis mais altos da empresa já são padronizadas. Levando-se em consideração que uma filial normalmente possui computadores com padrão Ethernet(802.3), sendo trivial a conexão deste com a empresa. A segunda parte do problema é mais complicada, pois estamos trabalhando em uma área onde a padronização está caminhando em passos pequenos, já que cada empresa de componentes de automação industrial tenta fazer com que o seu protocolo de comunicação se torne o padrão.

Para resolver este problema, foram criadas normas de padrões abertos para o desenvolvimento de redes industriais, desta forma todos podem padronizar seus componentes, facilitando a interconexão. A rede PROFINET é um destes padrões, está focada em tornar viável esta realidade, através da automação industrial utilizando o padrão Ethernet(802.3) e o protocolo TCP/IP, facilitando a conexão com os sistemas de Tecnologia da informação existentes [1].

O padrão de rede Ethernet industrial PROFINET é mantido pela associação de empresas de automação industrial, denominada PROFIBUS & PROFINET International (PI), e foi padronizado e incluído nas normas: “*Industrial communication networks*” IEC 61158 e IEC 61784 [7].

A utilização das redes Ethernet industriais para automação traz muitas vantagens além da padronização. Pois se compararmos o modelo tradicional de automação, com os componentes tendo uma conexão ponto a ponto diretamente com o CLP, com a Ethernet industrial temos uma diminuição drástica da quantidade dos cabos, já que agora só precisaremos

de um único cabo interligando todos os componentes(barramento industrial) [1].

Pelo fato da rede PROFINET utilizar o padrão Ethernet(802.3), ela pode utilizar as topologias: linha, anel, estrela e árvore. Podendo utilizar cabo ou fibra óptica. Além de permitir conexão com WLAN e Bluetooth [4].

II. A REDE PROFINET

A rede PROFINET pode ser dividida em dois tipos: PROFINET CBA e PROFINET I/O [4]. Podemos ver o relacionamento deles através da Fig. 1.

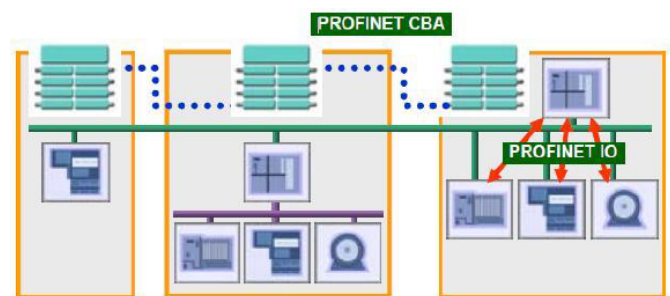


Fig. 1. Relacionamento entre PROFINET CBA e PROFINET I/O [4]

Originalmente somente o tipo PROFINET CBA tinha sido definido, ele era indicado para conexão entre dispositivos de automação inteligentes ou computadores[9], ou seja, aplicações que não exigem tempo real, como, por exemplo, na conversão de PROFINET para a rede PROFIBUS DP [13].

Já o PROFINET I/O, analogamente ao PROFIBUS DP, é baseado em um modelo de dispositivos modulares e em uma cíclica troca de mensagens entre o dispositivo e o sistema supervisor[9]. Ele é utilizado para aplicações onde o tempo é uma variável crítica [13].

A rede PROFINET segue o modelo Provedor/Consumidor para a troca de dados, são definidos três tipos de dispositivos básicos [3][8]:

- I/O-Controller: O I/O-Controller se refere a estação central de inteligência, comumente um CLP. É responsável pela configuração e parametrização de seus dispositivos e controla a transferência de dados do processo. Pode ser comparado a um mestre classe 1 no PROFIBUS.

- **I/O-Device:** Representa o dispositivo de campo remoto que mantém comunicação com um ou mais I/O-Controllers. Ele tanto transmite quanto recebe dados ciclicamente do I/O-Controller, e também provém diagnósticos e alarmes ao I/O-Controller. Pode ser comparado a função de um escravo no PROFIBUS.
- **I/O-Supervisor:** Pode representar um dispositivo de programação, um computador ou alguma interface homem-máquina, é um dispositivo que comissiona e tem funções de diagnóstico na rede. Pode ser comparado a um mestre classe 2 no PROFIBUS.

A. Estabelecimento de Comunicação

Para um I/O-Controller e um I/O-Device estabelecerem comunicação é necessária a criação de um caminho de comunicação. Estes caminhos são definidos pelo I/O-Controller durante a inicialização do sistema, com base nos dados recebidos do sistema de engenharia. Esta etapa especifica explicitamente a troca de dados [3].

Toda a troca de dados é baseada em um AR(*Application Relation*). Dentro de um AR, CRs(*Communication Relations*) definem a explicitamente os dados. Desta forma todos os dados necessários para a modelagem de um I/O-Devices, incluindo os parâmetros de comunicação são inseridos nos I/O-Devices. Um I/O-Device pode ter várias ARs estabelecidas com diferentes I/O-Controllers [4]. Podemos observar a estrutura de um AR na Fig. 2.

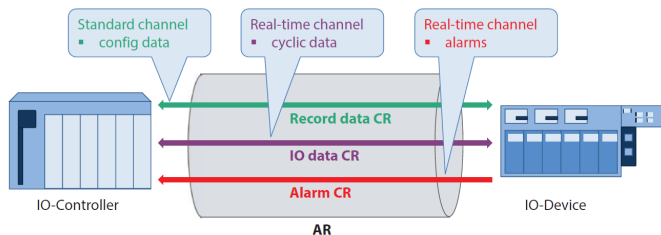


Fig. 2. Comunicação entre um I/O-Device e um I/O-Controller [3]

Os canais para troca cíclica de dados(*I/O data CR*), troca acíclica de dados(*Record data CR*) e alarmes(*Alarm CR*) são definidos simultaneamente. Deve ser especificado previamente na configuração do sistema, caso se deseje o acesso simultâneo de dados em um I/O-Device [3].

B. Troca cíclica de dados

Os dados cíclicos são transmitidos via o canal *I/O data CR*, os dados que trafegam neste canal são determinados como dados em tempo real entre o consumidor e o provedor. Os tempos de ciclo podem ser configurados individualmente nos elementos da rede, fazendo-a flexível para suprir os requisitos das aplicações. Os tempos de ciclo podem variar de 250µs até 512ms [4].

A conexão é monitorada através de um tempo de monitoramento, que é definido através de um múltiplo do tempo de ciclo. Durante a transmissão, os dados são seguidos pelo estado do provedor. Este estado serve para que o consumidor avalie a validade dos dados enviados, o consumidor também envia seu estado para o provedor, desta

forma a própria troca cíclica de dados informa o estado dos elementos da rede, evitando tráfego desnecessário na rede para estas verificações [3].

O contador de ciclo do provedor também é enviado ao consumidor, fazendo a taxa de atualização algo fácil de ser determinado. A falha da comunicação é monitorada pelo próprio consumidor, e caso os dados não cheguem dentro do intervalo de tempo de monitoramento, o consumidor envia uma mensagem de erro diretamente para a aplicação [3]. Podemos observar um exemplo do canal de troca cíclica na Fig. 3.

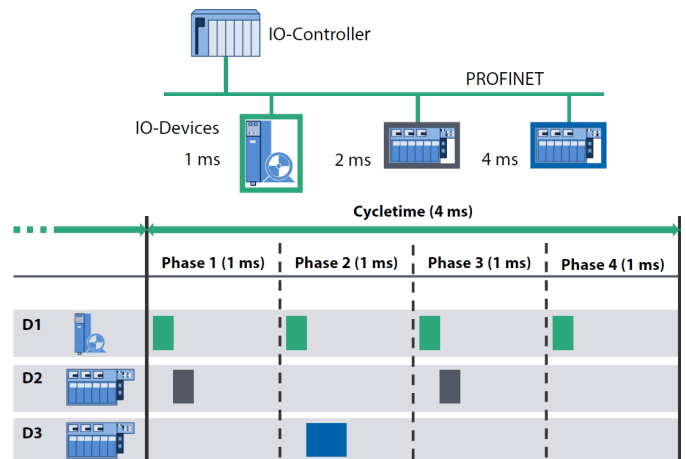


Fig. 3. Comunicação em tempo real com monitoração de tempo de ciclo [3]

C. Modelo ISO/OSI

Podemos observar na Fig. 4. como o PROFIBUS define suas camadas:

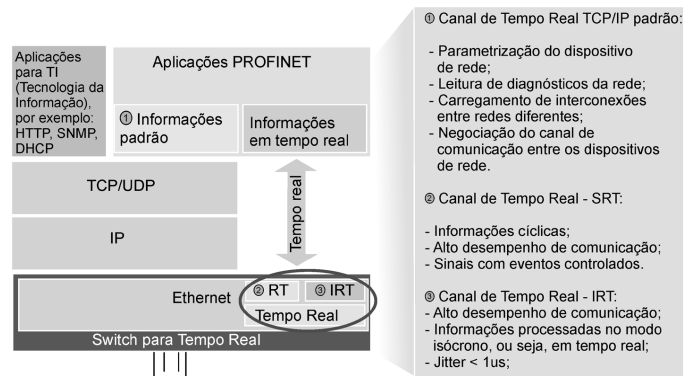


Fig. 4. Camadas PROFIBUS[6]

Temos três modos distintos de comunicação no PROFIBUS, *Non-real time*(NRT): comunicação em tempo não real(suíte de protocolos TCP/IP padrão), *Real time*(RT): comunicação em tempo real e *Isynchronous Real Time*(IRT): comunicação em tempo real isócrono.

Podemos perceber que a comunicação NRT apenas implementa as camadas 1, 2, 3, 4 e 7 do modelo ISO/OSI, descartando as camadas 5 e 6 para diminuir o processamento, e conseqüentemente o tempo dos pacotes. Utiliza Ethernet na camada um e dois, o IP na camada três, o TCP ou UDP na camada quatro e na camada sete os protocolos para aplicações TI e aplicações PROFIBUS [1]. Toda comunicação para o

gerenciamento, parametrização e configuração é considerada NRT e é definida como um protocolo de aplicação utilizando chamadas de procedimento remoto(RPC) no protocolo UDP/IP [9]. Este tipo de comunicação também é utilizado na comunicação com os *Proxis*(ou *Gateways*), como podemos observar na Fig. 5.[13] A comunicação NRT utiliza basicamente o PROFINET CBA [6].

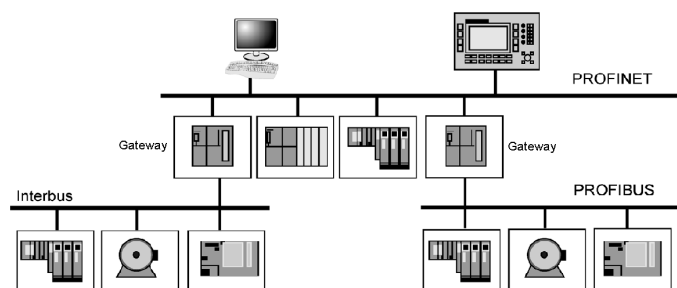


Fig. 5. Gateways PROFINET/PROFIBUS DP e PROFINET/Interbus [6]

Para obter um tempo cíclico na ordem de alguns milissegundos com um *jitter* moderado, a comunicação RT descarta a camada quatro(TCP/UDP) e três(IP), e interliga diretamente a camada de aplicação a camada Ethernet. Ambas as cíclicas trocas de dados e as mensagens de alarmes usam o mecanismo de comunicação RT [9]. A comunicação RT pode utilizar tanto o PROFINET CBA, quanto o PROFINET I/O [6].

Para cenários onde o tempo é de extrema importância, como por exemplo, o controle de aplicações móveis, onde o tempo de resposta fim a fim tem que ser menor que 1ms, temos a comunicação IRT. A comunicação IRT implementa o protocolo PTCP(*Precision Transparent Clock Protocol*) de acordo com o IEEE 1588 para sincronizar os *clocks* em todas as estações e *switches*, mas para isto o IRT requer equipamentos específicos [9]. É configurado um circuito virtual, de forma que todos os elementos da rede saibam que porta física devem acessar. Para garantir que não ocorram atrasos, os quadros IRT possuem uma reserva da rede no início de cada ciclo de comunicação [7].

Podemos analisar as diferenças entre o *delay* e o *jitter* entre os tipos de comunicação através da Tabela I.

TABELA I. MODOS DE COMUNICAÇÃO PROFINET[12]

Modo de comunicação	Delay	Jitter(%)
Non-real time(NRT)	$\geq 100\text{ms}$	≥ 100
Real time(RT)	$\geq 5\text{ms}$	≥ 15
Isynchronous Real Time(IRT)	$\geq 250\mu\text{s}$	$\leq 0.4 (1\mu\text{s})$

D. Quadro PROFINET

O quadro para a rede PROFINET I/O segue o mesmo padrão que o quadro Ethernet, a grande diferença é no campo *Frame ID*, no qual o valor define se a comunicação é NRT, RT ou IRT. Por exemplo, os valores de 0000h a 00FFh definem um quadro IRT, de 8000h a FBFFh, definem um quadro RT e de FC00h a FE02h definem um quadro NRT [6].

Podemos visualizar o formato do quadro na Fig. 6. O campo inicial *InterFrameGap* que contém 12 bytes é utilizado para criar um pequeno tempo de recuperação entre os quadros, para que os dispositivos possam se preparar para o próximo quadro [10].

O campo preâmbulo contém 7 bytes auxiliares que identificam o início da transmissão, auxiliam na sincronização do dispositivo com o *clock* da rede, o padrão é 10101010. É seguido pelo campo *Start of frame delimiter*(SFD) que possui 1 byte, ele delimita o início do quadro, e tem o padrão 10101011. Após isto temos o campo do endereço MAC de destino e o campo do endereço MAC do transmissor, o endereço MAC é o endereço físico do componente, ambos com 6 bytes. Sendo os três primeiros bytes reservados para a identificação do fabricante, e os três últimos reservados para a identificação do produto na rede [10].

Agora temos o campo *Ethertype*, ele é usado para definir qual é o protocolo utilizado na camada superior, possui dois bytes, o valor que representa o protocolo PROFINET é 0x8892, este valor é utilizado para as comunicações RT e IRT, já a comunicação NRT utiliza o valor 0x0800, o protocolo IP [10]. Temos também o campo VLAN, que indica qual é a rede virtual local em que o dispositivo está conectado, possui o comprimento de 2 bytes [11]. Logo em seguida temos o campo *Frame ID*, como já explicado possui valores definidos para cada tipo de comunicação NRT, RT e IRT, possui um tamanho de 2 bytes [6]. Em seguida temos o campo *Data*, é ele que contém os dados do pacote, ou seja, a informação que queremos enviar ou receber, podendo conter de 40 a 1440 bytes [1]. No campo *Cycle Counter* cada bit representa um incremento de tempo de 31.25 μs , serve para o receptor checar os quadros recebidos [15].

O campo *Data Status* e *Transfer Status*, servem apenas para indicar o estado da comunicação, ambos tem 1 byte [15]. Por fim temos o campo FCS(*Frame check Sequence*) este campo possui o valor do polinômio resultante CRC-32(*Cyclic Redundancy Check* de 32 bits), que verifica a quantidade de erros na comunicação, e pode detectar até 32 bits errados em sequência [2].

E. Segurança

Como o PROFINET interliga os níveis mais altos da empresa na mesma rede que os sistemas de automação, a segurança é algo que deve ser analisado. O PROFINET é baseado em um sistema de segurança em zonas, recomenda-se uma otimização de segurança dependendo da necessidade do processo. Isto faz com que os dispositivos de automação não sejam sobrecarregados com especificações de segurança, e também permite que os conceitos de segurança sejam adaptados aos requisitos do projeto de automação [3]. Podemos analisar um exemplo de rede na Fig. 7.

Apesar de não sobrecarregar os dispositivos de automação, esta abordagem em “zonas de confiança” faz com que qualquer conexão mal intencionada na rede seja extremamente perigosa, podendo levar a grandes prejuízos. Já que não existe nenhum tipo de restrição entre os nós após a conexão ser estabelecida [8].

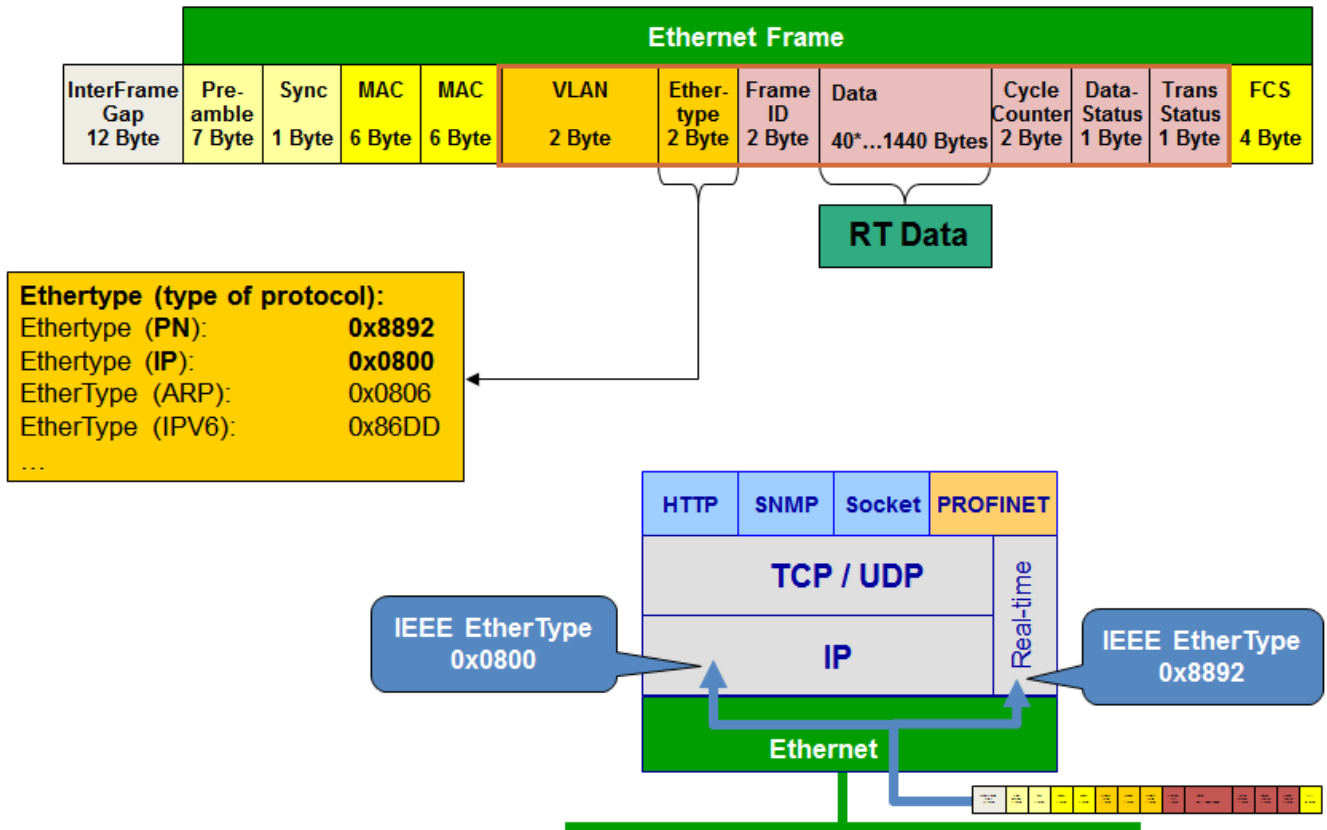


Fig. 6. Quadro PROFINET[14]

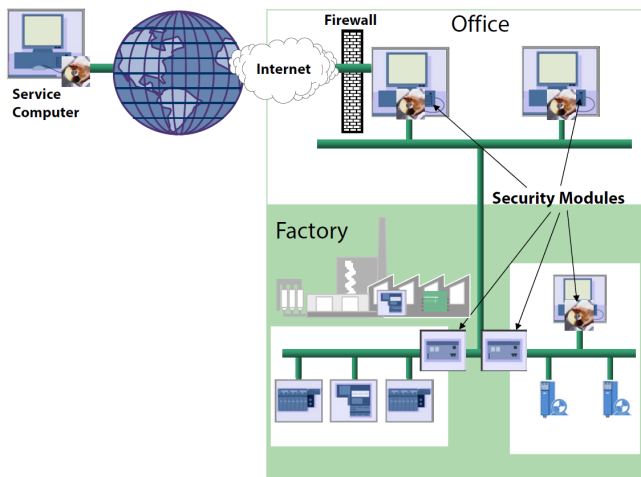


Fig. 7. Exemplo de rede utilizando acesso seguro [3]

III. MEIO FÍSICO

Antes de definirmos o meio físico, devemos entender as classes de conformidade existentes no PROFINET, elas servem para garantir que os dispositivos certificados com elas terão as propriedades mínimas de funcionamento e interoperabilidade [5]. Podemos observar as classes através da Fig. 8.

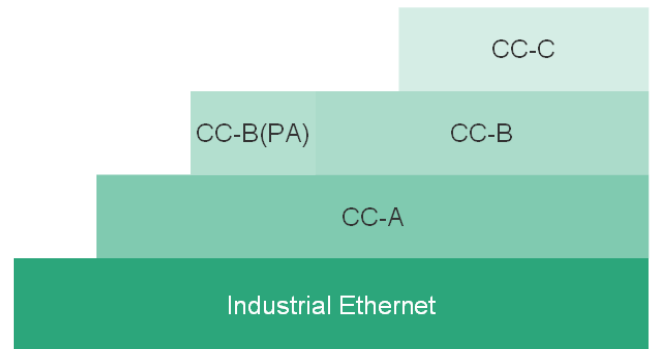


Fig. 8. Classes de conformidade do PROFINET[5]

A classe CC-A provê as funcionalidades básicas para o PROFINET I/O com comunicação RT, todos os serviços de TI podem ser utilizados nesta classe, é utilizada tipicamente em automação predial. A classe CC-B inclui os diagnósticos via mecanismos TI assim como o levantamento da topologia de rede. A função de redundância do sistema está presente na versão estendida da classe CC-B(PA). A classe CC-C descreve as funções básicas para dispositivos reserva de banda e sincronização suportada por *hardware*(comunicação IRT), ou seja, é a base para as aplicações isócronas [5].

De acordo com a classe de conformidade temos uma definição de conectores, cabos e dispositivos de infraestrutura, que podem ser observados na Tabela II. :

TABELA II. DEFINIÇÕES DAS CLASSES DE CONFORMIDADE[4][5]

Cabeamento da rede e componentes de infraestrutura	Solução	Classe de conformidade
Componentes de redes passivos (conectores, cabos)	RJ45, M12, SC-RJ	A, B, C
Sistemas de transmissão com fio de cobre e fibra óptica	TX, FX, LX	A, B, C
Conexões sem fio	WLAN, Bluetooth	A
Switches TI compatíveis	Com tag VLAN de acordo com IEEE 802.x e LLDP 802.1ab	A
Switch com função de dispositivo	PROFINET com RT	B
Switch com função de dispositivo e banda reservada	PROFINET com IRT	C

Os dispositivos de campo PROFINET são sempre conectados a rede através de *switches*. Tipicamente os dispositivos de campo possuem um *switch* integrado para suprir esta necessidade [3].

Switches adequados para a rede PROFINET devem suportar “*autonegotiation*” (negociação dos parâmetros de transmissão) e “*autocrossover*” (cruzamento autônomo de linhas de envio e recepção) [3].

As topologias suportadas pela rede PROFINET são[3]:

- Topologia em linha: Utiliza os *switches* integrados para conectar os dispositivos, temos um exemplo na Fig. 9.
- Topologia em estrela: Requer um *switch* central para conectar os dispositivos.
- Topologia em anel: Onde uma linha é fechada para formar um anel, criando assim redundância nos cabos.
- Topologia em árvore: Uma combinação das topologias citadas acima.

O máximo comprimento de um cabo de cobre, para a transmissão elétrica dos dados entre dois nós (dispositivo de campo ou *switch*) é de 100 metros. Os cabos devem ter bitola AWG 22. Já que os *switches* (externos ou integrados) servem como repetidores, não há comprimento máximo da rede [4].

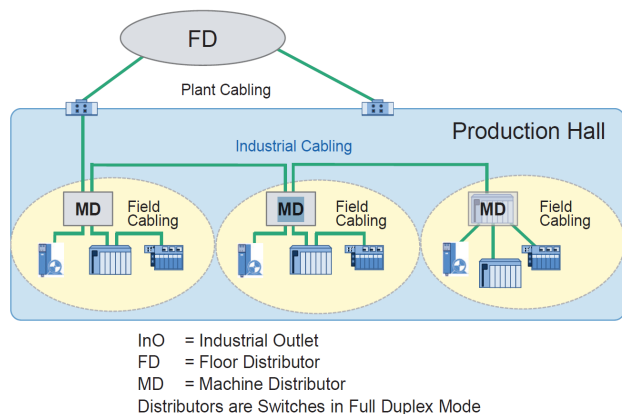


Fig. 9. Redes Ethernet usualmente utilizam topologia linha [3]

Os cabos PROFINET seguem os tipos de cabos utilizados na indústria[4]:

- PROFINET tipo A: Cabo para instalação fixa padrão, não há movimentos após a instalação.
- PROFINET tipo B: Cabo flexível padrão, movimento ou vibração ocasional.
- PROFINET tipo C: Cabo para aplicações especiais, como alta flexibilidade ou movimento constante.

Além da transmissão via cabo de cobre, podemos utilizar a fibra óptica, visto que ela apresenta várias vantagens em relação ao cabo de cobre, principalmente devido a sua isolamento elétrica, fazendo-a adequada quando a conexão equipotencial é difícil de se realizar [3].

Para pequenas distâncias, é suportado o uso de fibras ópticas de polímetro de 1 mm (*polymer optic fibers* – POF), cuja manipulação se adapta muito bem aos requisitos industriais [3].

A fibra óptica também é útil quando temos requisitos extremos de EMC (*Electromagnetic compatibility*), por exemplo uma linha de alta-tensão perto do cabo de transmissão de dados. Além de poder transmitir dados a distâncias na casa dos quilômetros, sem repetidores [3].

A. Conectores

O PROFINET dividiu em as condições ambientais em duas classes, uma classe para ambientes protegidos internos, como dentro de um painel de controle e uma classe externa a painéis de controle, para aplicações localizadas diretamente no campo[3]. Podemos observar os conectores na Fig. 10.

A seleção do tipo de conector depende da ênfase do projeto. Se a ênfase for em uma rede universal, com conectividade com os equipamentos de escritório, deve-se utilizar a transmissão elétrica de dados através do RJ 45, que é recomendado para condições de ambientes internos [3].

Para ambientes externos, um conector *push-pull* foi desenvolvido, ele também utiliza internamente o conector RJ 45 para transmissão elétrica de dados. O conector M12 também é especificado para PROFINET [3].

Para transmissão óptica de dados com fibras ópticas de polímero o conector SC-RJ é especificado, ele foi baseado no conector SC. Ele é utilizado tanto em ambientes internos como em ambientes externos, mas para o ambiente externo foi desenvolvido um conector *push-pull* com maior grau de proteção [3].

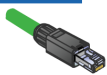



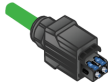
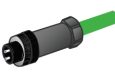
	Copper	Fiber Optic
IP 20 Inside	RJ 45 	SC-RJ 
IP 65/67 Outside	RJ 45  M12 	SC-RJ  M12 

Fig. 10. Conectores para a rede PROFINET [3]

IV. CONCLUSÃO

Podemos perceber que o fato do PROFINET ser compatível com outras redes já usadas no ambiente industrial, através de *gateways*, e ter classes que permitam o uso de protocolos comuns para redes de escritório, faz com que ele seja muito interessante para a ligação entre o escritório e o “chão de fábrica”.

A divisão de classes também é muito interessante, possibilita uma grande variedade de pacotes trafegar em um mesmo meio, mesmo cada um possuindo algum tipo de requisito de conexão diferente, fazendo com que a rede possa trabalhar em vários níveis da empresa, sem perder desempenho.

Sendo assim a rede PROFINET é sem dúvida uma ótima opção para a automação, principalmente pela sua flexibilidade, mas não deve-se esquecer sobre as características de segurança mencionadas aqui, as quais podem ser um problema dependendo do projeto.

REFERÊNCIAS

- [1] Lugli, A. B. e Santos, M. M. D.. Redes Industriais para Automação Industrial: AS-I, PROFIBUS e PROFINET. 1 ed. São Paulo: Érica, 2010, p. 126-135.
- [2] Tanenbaum, A. S. Redes de Computadores. 5 ed. São Paulo: Perarson Pretice Hall, 2011, p. 132-134.
- [3] PROFIBUS & PROFINET International, PROFINET System Description Technology and Application, Versão outubro 2014, Karlsruhe, Alemanha, PROFIBUS Nutzerorganisation e. V., 2014, p. 2-18.
- [4] PROFINET – Descrição do Sistema Tecnologia e Aplicação(Versão 2013). Disponível em: <http://www.profibus.com/nc/download/technical-descriptions-books/>
- [5] PROFIBUS & PROFINET International, PROFINET IO Conformance Classes, Versão 1.1, Karlsruhe, Alemanha, PROFIBUS Nutzerorganisation e. V., 2011, p. 4-13.
- [6] Lugli, A. B.. Uma Arquitetura Distribuída Flexível para Aplicações Industriais Baseada no Padrão Ethernet. Itajubá: UNIFEI, 2013. Tese (Doutorado em Ciências em Engenharia Elétrica.) - Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Itajubá, Itajubá, 2013.
- [7] Dias, A. L., Sestito, G. S., Turcato, A. C., Oliveira e Souza, P. H. T. e Brandão, D.. “Um estudo sobre a tecnologia PROFINET” em I Congresso de Automação e Inovação Tecnológica Sucrenergética, Sertãozinho, SP, 2013.
- [8] Turcato, A. C., Flauzino R. A., Sestito, G. S., Dias, A. L. e Brandão, D.. “Ataque *Denial of Service* em redes PROFINET: Estudo de caso” em XII Simpósio Brasileiro de Automação Inteligente, Natal, RN, 2015, p. 217-222.
- [9] Kleines, H., Detert, S., Drochner, M. e Suxdorf, F.. “Performance Aspects of PROFINET IO” *IEEE Trans.* Vol. 55, no. 1, Fevereiro 2008, p. 290-294.
- [10] IEEE Standard for Ethernet, IEEE Standard 802.3, 2012.
- [11] IEEE Standard for Local and metropolitan area networks, IEEE Standard 802.1Q, 2005.
- [12] Ferrari P., Flammini A., Vitturi S., Performance analysis of PROFINET networks, *Computer Standards & Interfaces*, v.28 n.4, p.369-385, April, 2006, p. 369-385.
- [13] Lugli, A. B.. Uma visão do protocolo Profinet e suas Aplicações. Disponível em: <http://www.profibus.org.br/news/marco2009/news.php?dentro=4>.
- [14] Henning, C.. (2013, November 12). PROFINET for Network Geeks (and Those Who Want to Be). Disponível em: <http://www.geautomation.com/blog/profinet-network-geeks-and-those-who-want-be>
- [15] Felser, M., PROFINET IO Measurements of telegram – sequences with Etherreal(Versão 2012). Disponível em: http://www.felser.ch/download/6_PN_Measurements_E02.pdf

Protocolo de Comunicação Sem Fio Zigbee

Brunno Vanelli

Universidade Federal de Santa Catarina

Email: brunno.v@grad.ufsc.br

Resumo—As redes sem fio proporcionam muitas vantagens para uso doméstico, comercial e industrial. Devido à necessidade de comunicação adequada sem perda de mobilidade dos dispositivos, surgiram alternativas que contornam a imensa quantidade de ruídos presentes no meio e proporcionam um meio de comunicação seguro e eficiente. Esse artigo tem o foco de discutir a aplicação do protocolo Zigbee, um protocolo de comunicação de baixo custo energético, utilizado em aplicações industriais e em dispositivos IoT como smart-homes e healthcare.

Keywords—Zigbee, IEEE 802.15.4, RSSF

I. INTRODUÇÃO

As redes sem fio se popularizaram muito nos últimos anos devido à rapidez de implantação e grande aumento de conectividade. Dezenas de metros de cabos podem ser substituídos por apenas um roteador com conectividade sem fio, o que ainda adiciona mobilidade aos dispositivos que estão em campo. No entanto, ao utilizar o ar como meio de transmissão sem fio, alguns problemas precisam ser contornados como a grande quantidade de erro nos bits de informação que trafegam no meio, que é muito superior aos meios com fio, em especial o de fibra óptica [1].

Em um meio industrial e aplicações críticas, diferente de meios domésticos, os critérios de avaliação de redes são diferentes. Em geral, não é necessário grande largura de banda ou grande quantidade de dados trafegados, que acarreta em arquitetura de mensagens menores e mais enxutas. O *delay* ou atraso na entrega das mensagens passa a se tornar um fator importante, bem como o determinismo da rede, já que alguns tipos de aplicações exigem respostas rápidas para o controle de processos.

Uma dos protocolos disponíveis para implementação de redes que atendem a esses requisitos é o Zigbee. Criado pela Zigbee Alliance [2], baseado na norma IEEE 802.15.4, o protocolo Zigbee visa ser uma tecnologia de baixo custo e baixo consumo energético para comunicação sem fio entre dispositivos, de forma a fornecer máxima vida útil para baterias em aplicações de monitoramento [3]. O protocolo opera nas frequências 868 MHz, 915 MHz e 2,4 GHz, com quantidade máxima de dados de 250 kbps.

O que torna os dispositivos baseados em Zigbee tão interessantes é o fato de ele poder passar grande parte do seu tempo no modo de dormência (*sleep mode*), modo de operação de baixo consumo energético. Isso pode fazer com que as baterias dos dispositivos possam durar anos antes de precisarem ser substituídas, que pode ser um enorme atrativo para monitoramento de áreas remotas onde recarregar os substituir as baterias não são opção viável.

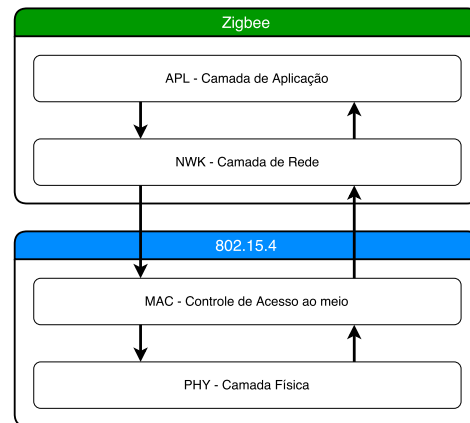


Figura 1. Camadas modelo OSI

As aplicações envolvendo Zigbee são variadas, podendo ser na área médica, automação residencial e industrial, monitoramento, segurança, etc [4]–[7].

A comparação das camadas em relação ao modelo de referência OSI pode ser visto na Figura 1. Em si, protocolo Zigbee define as camadas de Redes e Aplicação (NTW e APL), e utiliza as camadas física (PHY) e subcamada de Controle de Acesso ao Meio (MAC) descritas no padrão IEEE 802.15.4 [8] [9].

II. IEEE 802.15.4

O padrão IEEE 802.15.4 define as camadas física e de controle de acesso ao meio (MAC) para redes sem fio de área pessoal de baixa taxa (LR-WPANs). O objetivo desse padrão é definir as camadas de baixo nível para outros protocolos como o Zigbee e o WirelessHART, que então definirão as camadas acima.

A. Camada Física

De acordo com a norma IEEE, a camada física define basicamente como transmitir e receber dados dos dispositivos no meio físico, qual frequência utilizar e que tipo de modulação. Outras funções da camada física (PHY) são a ativação e desativação de dispositivos, detectar energia dentro do canal de comunicação, indicador de qualidade de conexão (LQI) para pacotes recebidos, garantir livre acesso ao meio e seleção da frequência do canal [8].

1) *Detecção de Energia (Energy Detection - ED)*: Quando um dispositivo quer enviar uma mensagem, se faz necessário

observar se nenhum outro dispositivo planeja enviar sua própria mensagem, a fim de evitar colisões. Para isso, pode-se utilizar essa função da camada física para detectar o nível de energia na banda de interesse. O ED é apenas capaz de revelar se algum sinal ocupa a banda de interesse, não se esse sinal é padrão 802.15.4, ruído, ou sinal de outro tipo de dispositivo.

2) *Detecção de Portadora (Carrier Sense - CS)*: De modo similar ao ED, a Detecção de Portadora consiste em detectar o meio, mas demodular o sinal encontrado e verificar se ele é compatível com o que está definido na camada física do padrão 802.15.4.

3) *Indicador de Qualidade da Conexão (Link Quality Indicator - LQI)*: É uma indicação da qualidade dos pacotes que chegam em um dispositivo. Normalmente, um bom indicador de qualidade da conexão é o RSSI (Received Signal strength indicator, ou Indicador da intensidade do sinal recebido). O RSSI também pode ser utilizado como indicador da disposição física dos nodos, em relação à posição e distância [10].

4) *Livre acesso ao meio (Clear Channel Assessment - CCA)*: Utilizado pela camada MAC para checar se o meio está livre ou ocupado para transmissão (CSMA-CA). Pode utilizar ED, CS, ou uma combinação dos dois para determinar a disponibilidade do meio.

As frequências escolhidas vão depender do tipo de modulação utilizado e como são distribuídos os canais, além do meio em que será utilizado. O Zigbee disponibiliza a modulação BPSK nas frequências 868 MHz e 915 MHz, e OQPSK em 2.4 GHz. Embora a frequência 2.4 GHz seja mundialmente aceita e utilizada por muito fabricantes, pois apresenta a maior taxa de dados e número de canais (250 kbps e 16 canais), pode-se argumentar que as frequências 868/915 MHz (30/40 kbps e 10 canais), por serem menores, conseguem penetrar melhor paredes e objetos, além de não terem de concorrer com outros dispositivos que utilizem a mesma frequência como o IEEE 802.11.

Todas as faixas de frequência empregam como modo de propagação sequência direta de espalhamento do espectro (DSSS). O DSSS consiste em espalhar um sinal em uma largura de banda muito maior do que ele precisa. O espalhamento é feito multiplicando o sinal original por um um sinal com um código binário, referido como o Código de Propagação, gerando o sinal com grande largura de banda a ser enviado. O sinal original pode ser recriado no receptor multiplicando o sinal recebido pelo menos código usado na geração do sinal [11]. Embora esse processo utilize maior largura de banda na transmissão, ele garante mais imunidade à interferências, intencionais ou não.

O frame de dados enviado pela camada física, chamado PPDU (PHY protocol data unit), pode ser visto na Tabela I

O primeiro conjunto de dados é o SHR, ou cabeçalho de sincronização, que é constituído pelo preâmbulo e pelo SFD, que é o delimitador de frame, um campo de 8 bits para indicar onde cada frame começa.

O segundo conjunto de dados é o PHR, ou cabeçalho físico, que serve para delimitar o tamanho do frame (7 bits), com um bit reservado, totalizando 8 bits.

Tabela I
PPDU

Preâmbulo	SFD (8 bits)	Tamanho do Frame (7 bits)	Reservado (1 bit)	PSDU
SHR		PHR		Payload

Por último, o payload (PSDU) contém todos os dados a serem transmitidos da camada acima. O PSDU pode ter de 0 a 1016 bits (127 octetos).

B. Subcamada MAC

A principal função da subcamada de controle de acesso ao meio (MAC) é organizar as PANs e fornecer conexões confiáveis, sem risco de colisão, além de servir como interface entre a camada física e a camada superior. A subcamada MAC também gerencia o GTS (*Guaranteed Time Slots*), ou tempo de slot garantido, que se ativado faz com que dispositivos que necessitam transmitir dados sempre o façam dentro dos limites estabelecidos [8].

O método de acesso ao canal normalmente utilizado é o CSMA-CA, ou acesso múltiplo com sensoramento da portadora evitando colisão, onde o meio é ouvido para se certificar de que nenhum outro dispositivo está enviando mensagens, o que acarretaria em uma colisão. Se dispositivo quer enviar uma mensagem, ele primeiramente faz a leitura do canal de comunicação (CCA, fornecido pela camada física). Se o meio está ocupado, o dispositivo espera um tempo aleatório após ele se tornar livre, evitando colisão com outro dispositivo que possa estar transmitindo. Embora o algoritmo não seja perfeito, ele é essencial para a eficiência da rede, pois uma mensagem que precisa ser reenviada devido a colisões significa um grande gasto energético desnecessário.

Em um método de acesso GTS (*guaranteed time slot*) o dispositivo coordenador da rede deve garantir que todos os dispositivos estejam sincronizados. Isso pode ser obtido através do envio de frames chamados Beacons, que sincronizam o relógio de todos os dispositivos na rede. Se todos os nodos estão sincronizados, o coordenador pode dar a cada dispositivo um tempo específico garantido para transmitir mensagens. Esse modo de acesso evita colisões e torna a rede determinística. No entanto, isso significa que os dispositivos devem sair do modo sleep regularmente para se sincronizarem com o beacon, o que pode gerar um consumo maior de energia.

O padrão 802.15.4 define quatro tipos de frame MAC:

- **Frame Beacon**: serve para sincronizar os dispositivos da rede com o relógio do coordenador no método de acesso GTS. Não só isso, mas também serve para avisar determinados dispositivos da rede que há dados a serem transmitidos. Assim, o dispositivo pode contactar o coordenador e requisitar os dados. Isso é chamado transmissão indireta.
- **Frame de Dados**: serve para transmitir os dados fornecidos pela camada de Rede.
- **Frame de confirmação (ACK)**: é o frame mais simples, pois não carrega nenhum payload. Pode ser enviado por

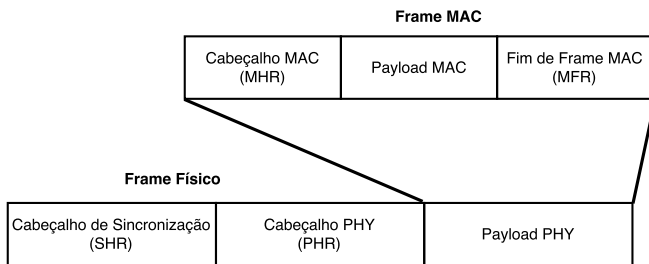


Figura 2. Empilhamento de camadas

dispositivos para confirmar o recebimento de pacotes com sucesso.

- Frame de Comando MAC: comandos MAC como requisições de associação e desassociação na rede são transmitidos através de frams de comando MAC.

O header do frame MAC (MHR) precisa conter informações de origem e destino do PAN ID (*personal area network identifier*) e do endereço, além de bytes de controle.

Em seguida, o próximo campo é o payload, que pode ter tamanho variável e representa os dados a serem transmitidos da camada acima.

Por último, o fim do frame (MFR) contém o CRC-16 de todo o quadro (MHR e payload) utilizando o polinômio $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$.

Em uma situação de envio, os dados da camada acima são inseridos no payload do frame MAC, que por sua vez é inserido no payload do frame PHY.

III. ZIGBEE

O protocolo Zigbee é definido em cima das definições de camada física e subcamada MAC do padrão IEEE 802.15.4. As camadas específicas do Zigbee são a de Rede, que tem a função de interfacear as camadas de aplicação e a subcamada MAC, e a camada de Aplicação, que hospeda os Objetos de Aplicação [12].

A. Camada de Rede

A função da camada de rede do protocolo Zigbee é a formação da rede (hierarquia, topologia, endereçamento, descoberta de nodos) e roteamento de mensagens [12].

O endereçamento pode ser obtido com endereços curtos de 16 bits (permite 65535 nodos em um cluster, sendo o endereço `0xffff` indicado como broadcast) dentro de uma rede ou endereço longo de 64 bits (padrão IEEE, que deve ser único a cada dispositivo), que endereça o nodo globalmente, mesmo em outros PANs. Analogamente, o PAN ID é também um endereço de 16 bits, mas pode ser estendido para 64 bits. A utilização de endereços menores de 16 bits diminui o overhead das mensagens e a utilização de memória dos dispositivos para armazenar endereços e tabelas de roteamento.

As mensagens podem ser enviadas em broadcast, quando é enviada para todos os nodos em uma rede, multicast, quando é enviada para apenas alguns endereços específicos na rede, ou

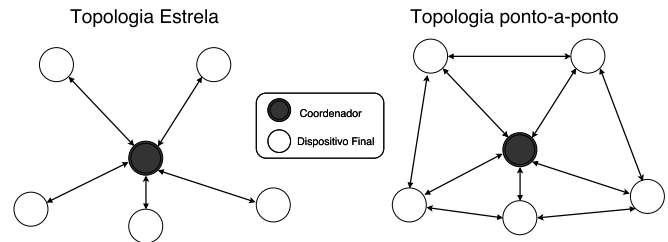


Figura 3. Topologias de Rede IEEE 802.15.4

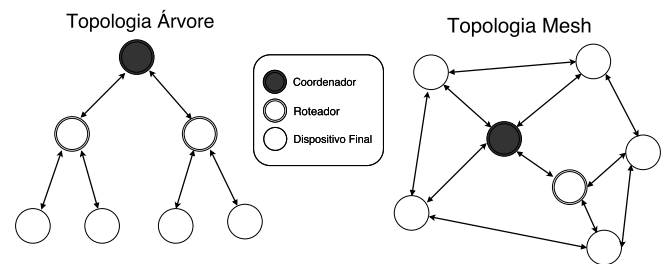


Figura 4. Topologias de Rede Zigbee

unicast, quando é enviada apenas para um único dispositivo específico.

As topologias definidas pela norma IEEE 802.15.4 são ponto-a-ponto (peer-to-peer), onde os dispositivos conversam um com o outro diretamente, e estrela, onde um dispositivo atua como coordenador da rede, conversando com os outros dispositivos definidos como dispositivos finais (Figura 3). A função do coordenador é iniciar, rotear ou terminar a comunicação na rede. Há ainda os roteadores, que tem as mesmas funções do coordenador mas não podem iniciar a rede. Os dispositivos finais (End Devices ou ED) são geralmente os dispositivos que vão à campo e tem as restrições de memória, processamento e energia. A função dos dispositivos na rede pode ser visto na Tabela II.

A camada de Redes do protocolo Zigbee adiciona a topologia Tree, ou árvore, onde a rede é organizada de forma hierárquica e o coordenador é a "raiz" da árvore (Figura 4).

Alternativamente, o protocolo Zigbee adiciona a topologia mesh, onde não há hierarquia e as rotas são criadas e modificadas dinamicamente. Essa configuração pode estender o alcance da rede, mas aumenta a latência, pois os pacotes deverão ser roteados até o destino por vários nodos caso o destino esteja muito distante.

A formação da rede Zigbee mesh se dá de forma automática, tão logo os dispositivos se tornam ativos. Assim que um dispositivo se torna coordenador, outros podem se juntar à rede enviando requisições de associação. Isso é o que se define *self-forming networks*. Normalmente, há mais de uma forma de rotear as mensagens dentro da rede, sendo as tabelas de roteamento criadas automaticamente pela camada de rede. Se por acaso um nodo sair da rede, for desligado ou tiver sua comunicação interrompida por um obstáculo físico, a rede seleciona uma rota alternativa. Isso é o que se chama

Tabela II
FUNÇÃO DOS DISPOSITIVOS NA REDE

Função	Coordenador	Roteador	End Device
Estabelecer Rede Zigbee	✓		
Permitir dispositivos entrar na rede	✓	✓	
Distribuir endereço 16-bit	✓	✓	
Rotear pacotes na Rede	✓	✓	
Enviar ou receber pacotes	✓	✓	✓
Entrar ou sair da rede	✓	✓	✓
Entrar em modo Sleep			✓

self-healing. Essas duas propriedades garantem escalabilidade, outra propriedade importante das redes Zigbee.

B. Camada de Aplicação

A camada de Aplicação é a camada de mais alto nível do protocolo Zigbee. Ela define os objetos Zigbee e como eles devem interagir com a camada de Rede [12]. A camada de aplicação pode ser dividida em três subcamadas:

1) *Objetos de Aplicação (APO)*: Fornece um conjunto de no máximo 240 serviços que serão utilizados pelo ZDO e pelo APS (outras subcamadas da camada de aplicação). Esses serviços são os “objetos” do dispositivo, como enviar a receber dados ou ler sensores conectados, e geralmente são dados pelo fabricante. Os objetos de aplicação podem estar também relacionados aos objetos físicos como sensores e atuadores. O conjunto de serviços fornecidos pelo dispositivo se define perfil de aplicação. Para um bom funcionamento da rede os dispositivos devem concordar com o perfil de aplicação. Se dois dispositivos forem de fabricantes distintos, mas tiverem o mesmo perfil de aplicação, então terão as mesmas funcionalidade.

2) *Objeto de Dispositivo Zigbee (ZDO)*: Sua principal função é inicializar as outras duas camadas (APS e NWK), a descoberta de nodos e serviços, definir o modo de operação do dispositivo (coordenador, roteador, dispositivo final) e segurança.

3) *Subcamada de Suporte à Aplicação (APS)*: Tem a função de interfacear as duas subcamadas acima (APO e ZDO) com a cada de redes, garantir o endereçamento e o transporte de dados confiável.

IV. CARACTERÍSTICAS

A. Segurança

As mensagens trocadas pelos dispositivos apresentam um grande grau de segurança por utilizar propagação DSSS, mas ainda são suscetíveis à serem interceptadas e/ou modificadas. Como é definido na norma 802.15.4, deve-se garantir confidencialidade e autenticidade, além de proteção contra ataques do tipo Replay [8].

Um dispositivo intruso pode conseguir informações relevantes apenas escutando o meio. Uma das técnicas para garantir confidencialidade dos dados é utilizando criptografia. É utilizado uma chave simétrica AES de 128 bits para criptografar

os dados e é partilhada entre todos os dispositivos da rede. A chave pode ser obtida localmente, vir pré-instalada pelo fabricante ou ser enviada de um dispositivo ao outro durante o funcionamento. Assim, quando uma mensagem é enviada, apenas os dispositivos com a chave podem descriptografar e ler o conteúdo [14].

No entanto, outro problema é que mesmo que o intruso não saiba a chave criptográfica, ele pode tentar modificar e reenviar uma mensagem antiga. Para garantir a autenticidade dos dados, é enviado um código de integridade (MIC) em cada frame, que não pode ser alterado sem ser notado [15]. O processo de garantir a autenticidade dos dispositivos que entram na rede é realizado pelo Centro de Confiança (*Trust Center*). O novo dispositivo deve receber a chave da rede (ou uma chave de zeros caso já a tenha, como parte do procedimento) e configurar alguns parâmetros para ser considerado autenticado, ou então será removido da rede.

Mas mesmo que o intruso não saiba a chave nem modifique os dados, ele ainda pode enviar mensagens antigas na rede, que podem ter impactos negativos (leituras incorretas de sensores, ativação de atuadores). Esse tipo de ataque se chama *Replay attack* ou *playback attack*. Para evitar isso, o Zigbee usa *freshness counters*, contadores que indicam o quão frescas as informações estão.

Uma das maiores restrições em implantar medidas mais sofisticadas são os recursos computacionais disponíveis, tais como processamento, memória e vida útil das baterias, o que pode tornar o dispositivo vulnerável [16] [17].

B. Interoperabilidade

Como o Zigbee tem uma grande gama de aplicações, muitos fabricantes devem fornecer soluções Zigbee que sejam capazes de se comunicar umas com as outras. Essa é uma das grandes vantagens do Zigbee: os dispositivos são interoperáveis mesmo se as mensagens são criptografadas [18].

C. Baixo Consumo

O Zigbee foi projetado para seus dispositivos operarem com restrições de energia. É esperado que dispositivo em campo durem meses ou até anos sem precisarem substituir as baterias. Existem três estágios de operação que precisam ser levados em conta ao estimar a vida útil de um dispositivo: o quando de corrente ele consome ao enviar ou receber dados, ficar em modo de espera (ou aguardar por mensagem) e o modo sleep (modo de dormência).

Geralmente, o modo com maior gasto energético é o modo onde ele envia ou recebe dados. Utilizando como exemplo o XBee, produzido pela empresa Digi Internacional, o gasto nesse modo é de 45 mA ao enviar mensagens e 50 mA ao receber mensagens (considerando alimentação 3,3 V). Esse gasto é extremamente pequeno, mas ainda é elevado para aplicações que precisam ficar ativas por longos períodos de tempo.

Em modo de espera, o dispositivo ainda possui suas funcionalidades, portanto continua tendo praticamente o mesmo consumo de 50 mA.

Tabela III
COMPARATIVO ENTRE PROTOCOLOS [13]

	Bluetooth	UWB	Zigbee	Wi-fi
Especificação IEEE	802.15.1	802.15.3a	802.15.4	802.11a/b/g
Frequência	2,4 GHz	3,1 - 10,6 GHz	868/915 MHz e 2,4 GHz	2,4 GHz e 5 GHz
Máxima transmissão de dados	1 Mbps	110 Mbps	250 Kbps	54 Mbps
Alcance	10 m	10 m	10 - 100 m	100 m
Potência de Transmissão	0 a 10 dBm	-41.3 dBm/MHz	-25 a 0 dBm	15 a 20 dBm
Número de Canais	79	1 - 15	1/10 - 16 (2,4 GHz)	14 (2,4 GHz)
Largura de Banda do Canal	1 MHz	500 MHz - 7,5 GHz	0,3/0,6 MHz - 2 MHz	22 MHz
Modulação	GFSK	BPSK, QPSK	BPSK, OQPSK	BPSK, QPSK, COFDM, CCK, M-QAM
Propagação	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Número Máximo de Nós	8	8	Mais de 65000	2007
Criptografia	E0	AES	AES	RC4, AES
Verificação	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC

No modo sleep, porém, todas as atividades não essenciais do módulo são desligadas, apenas o clock e primitivas básicas continuam sendo executadas. Nesse modo, o dispositivo XBee necessita de menos de $50 \mu A$ para funcionar. No caso do XBee, o modo sleep pode ser tanto cíclico, quando o nodo realiza ciclos de acordar, enviar dados e voltar ao modo sleep por um período determinado, ou por pinos, quando o XBee é acionado através de pinos do dispositivo.

V. COMPARATIVO

Outros protocolos podem ser utilizados com a mesma finalidade de comunicação sem fio, como o Bluetooth, UWB (Ultra-wideband) e Wi-fi. A comparação dos protocolos pode ser visto na Tabela III. Todos os protocolos apresentam vantagens e desvantagens, seja em transmissão de dados, uso energético, facilidade de implementação, etc.

Se considerada o fluxo máximo de dados, o UWB e o Wi-fi se destacam pelo volume de dados que podem transportar. No entanto, como pode ser visto na Figura 5, eles precisam de grande quantidade de energia para isso. Se o volume de dados for pequeno, pode ser mais vantajoso utilizar dispositivos de mais baixo custo e menor consumo como Zigbee.

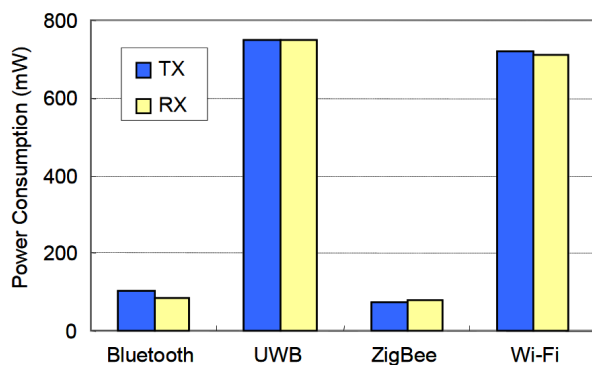


Figura 5. Consumo de energia por protocolo [13]

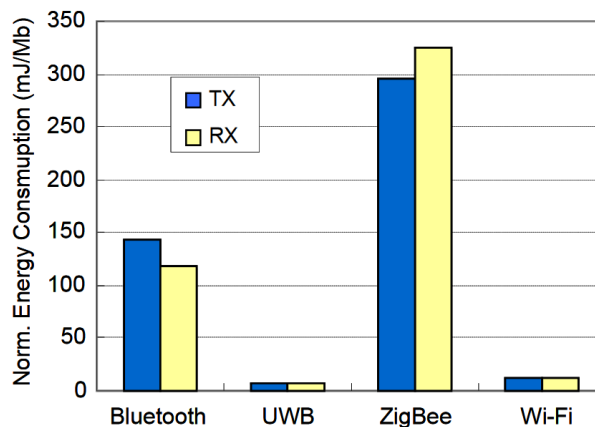


Figura 6. Consumo de energia normalizado [13]

No entanto, se o volume de dados for alto e não houver limitações energéticas, outros protocolos conseguem oferecer custo-benefício energético melhor do que o Zigbee, por conseguirem enviar mais dados de uma só vez. A Figura 6 fornece a relação entre gasto de energia por Mb de dados trafegados.

VI. CONCLUSÃO

Esse artigo apresenta uma visão geral da implementação do protocolo Zigbee e uma análise de alguns dos seus pontos positivos e negativos em relação aos problemas que ele tenta resolver.

Um dos fatores que podem impulsionar a utilização do protocolo Zigbee é a crescente quantidade de dispositivos que necessitam de comunicação sem fio e a Internet das Coisas (IoT). Uma das adaptações que se faz necessária nesses casos ao protocolo Zigbee é o interfaceamento entre Zigbee e protocolo IP, pois isso garantiria aos dispositivos grande aumento na conectividade.

REFERÊNCIAS

- [1] A. S. Tanenbaum *et al.*, “Computer networks, 4-th edition,” *ed: Prentice Hall*, 2003.
- [2] Zigbee Alliance. [Online]. Available: <http://www.zigbee.org>
- [3] S. Farahani, *ZigBee wireless networks and transceivers*. Newnes, 2011.
- [4] M. P. Silva, A. L. Gon, M. Dantas, B. Vanelli, G. Manerichi, S. A. dos Santos, M. Ferrandim, A. Pinto *et al.*, “Implementation of iot for monitoring ambient air in ubiquitous aal environments,” in *2015 Brazilian Symposium on Computing Systems Engineering (SBESC)*. IEEE, 2015, pp. 158–161.
- [5] K. Gill, S.-H. Yang, F. Yao, and X. Lu, “A zigbee-based home automation system,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 422–430, 2009.
- [6] P. Frehill, D. Chambers, and C. Rotariu, “Using zigbee to integrate medical devices,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 6717–6720.
- [7] J. Hou, C. Wu, Z. Yuan, J. Tan, Q. Wang, and Y. Zhou, “Research of intelligent home security surveillance system based on zigbee,” in *Intelligent Information Technology Application Workshops, 2008. IITAW’08. International Symposium on*. IEEE, 2008, pp. 554–557.
- [8] “IEEE standard for local and metropolitan area networks - part 15.4: Low-rate wireless personal area networks (LR-WPANs),” *IEEE Std. 802.15.4-2011*, 2011.
- [9] “IEEE standard for local and metropolitan area networks - part 15.4: Low-rate wireless personal area networks (LR-WPANs) - Amendment 1: MAC sublayer,” *IEEE Std. 802.15.4-2012*, 2012.
- [10] M. Sugano, T. Kawazoe, Y. Ohta, and M. Murata, “Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard,” *Target*, vol. 538, p. 050, 2006.
- [11] G. Vannucci, “Direct sequence spread spectrum (dsss) communications system with frequency modulation utilized to achieve spectral spreading,” Sep. 22 1992, uS Patent 5,150,377.
- [12] ZigBee Alliance, “Zigbee specification,” 2008, Document 053474r17.
- [13] J.-S. Lee, Y.-W. Su, and C.-C. Shen, “A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi,” in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*. IEEE, 2007, pp. 46–51.
- [14] H. Li, Z. Jia, and X. Xue, “Application and analysis of zigbee security services specification,” in *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, vol. 2. IEEE, 2010, pp. 494–497.
- [15] Zigbee Alliance, “Zigbee security specification overview,” 2006.
- [16] T. Zillner and S. Strobl, “Zigbee exploited-the good, the bad and the ugly,” *Black Hat USA*, vol. 2015, 2015.
- [17] N. Vidgren, K. Haataja, J. L. Patino-Andres, J. J. Ramirez-Sanchis, and P. Toivanen, “Security threats in zigbee-enabled systems: vulnerability evaluation, practical experiments, countermeasures, and lessons learned,” in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, 2013, pp. 5132–5138.
- [18] W. Kluge, F. Poegel, H. Roller, M. Lange, T. Ferchland, L. Dathe, and D. Eggert, “A fully integrated 2.4-ghz ieee 802.15. 4-compliant transceiver for zigbee™ applications,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2767–2775, 2006.

Bluetooth

Jorge Lucas de Sant'Ana

Universidade Federal de Santa Catarina – Campus Blumenau

Departamento de Engenharia

Rua João Pessoa, Nr. 2750, Velha

Blumenau/SC, Brasil, 89036-256

Resumo – A utilização de dispositivos de comunicação sem fio (wireless) vem crescendo na área industrial e residencial. Essas tecnologias têm por objetivo eliminar os cabos, tornando assim a comunicação mais simples, e barata. O bluetooth é um exemplo de tecnologia wireless que adquiriu destaque nos últimos anos. Ele utiliza o princípio de comunicação por ondas de rádio, e é uma solução barata, de baixo consumo de energia e confiável. Este trabalho está focado em abordar as principais características dessa comunicação, que é voltada para curtas distâncias, corresponde ao padrão IEEE 802.15.1 para redes pessoais, e está presente em diversos equipamentos.

Palavras chave – bluetooth, IEEE 802.15.1, WPAN, comunicação wireless, redes sem fio.

I. INTRODUÇÃO

A comunicação sem fio (wireless) surgiu com o objetivo de eliminar cabos nas conexões entre dispositivos eletrônicos. Ela vem crescendo cada vez mais em número de aplicações, dessa forma cresce também a necessidade de tecnologias e padrões mais rápidos e eficazes.

Para obter um transmissor sem fio que fosse seguro, robusto, de baixo custo, baixo consumo de energia, e com pequeno alcance para conectar seus periféricos, a Ericsson em 1940 criou o Bluetooth. O desenvolvimento dessa tecnologia ganhou força em 1998 com as empresas Nokia, IBM, Toshiba, e Intel, que junto com a Ericsson formaram o consorcio chamado: Bluetooth SIG (Special Interest Group), para expandir o conceito Bluetooth e estabelecer um novo padrão industrial. [1][8]

O Bluetooth pode ser descrito como a maneira na qual os mais diversos dispositivos móveis ou fixos se comunicam através de uma conexão sem fio de curto alcance (10m normalmente ou até 100m com amplificadores) com um baixo custo e alta operabilidade. [1] [3]

Por solicitação do Sig, o IEEE 802 adotou as especificações do Bluetooth como o padrão IEEE 802.15.1. A partir daí passou a ser empregado principalmente em redes pessoais sem fio conhecidas como WPAN (Wireless Personal Area Network). [1]

O crescimento da utilização dos dispositivos móveis pessoais tornou famosa essa tecnologia, presente em notebooks, celulares, câmeras e outros equipamentos.

Neste trabalho serão discutidas as principais características do bluetooth, que envolvem seu método de funcionamento, arquitetura do protocolo, segurança, vantagens e aplicações. Serão citados também alguns outros dispositivos wireless, quando for necessário realizar um estudo comparativo com o bluetooth, para detalhar melhor sua funcionalidade. Por fim, será abordado sua capacidade de se manter no mercado como

tecnologia eficaz, e qual sua perspectiva de crescimento futuro.

II. CARACTERÍSTICAS GERAIS

O bluetooth realiza comunicação de curta distância via rádio. Dessa forma cada “equipamento Bluetooth” carrega dentro dele um chip, que nada mais é que um micro rádio que envia e recebe os dados.

A faixa de frequência em que ele opera varia de 2,4GHz a 2,48GHz conhecida como ISM (Industrial, Scientific & Medical), essa faixa de frequência é aberta para uso geral em vários países (sem necessidade de licenciamento individual). Como essa faixa de frequência é compartilhada com outras aplicações a exemplo do wifi, o protocolo evita interferências dividindo a faixa em 79 canais diferentes, espaçados em 1MHz. Ele muda de canal até 1600 vezes por segundo de forma pseudoaleatória, esse método é chamado de FHSS (Frequency Hopping Spread Spectrum). [7] [3]

Para haver comunicação entre os dispositivos bluetooth, eles devem atender a três requisitos: estarem na mesma área, conhecer o outro dispositivo e ter uma rota de comunicação preestabelecida.

Quanto a área de cobertura, os dispositivos bluetooth foram divididos em 3 classes:

- Classe 01: alcance de até 100 metros, com potência máxima de saída de 100mW (20 dBm), e mínima de 1mW (0dBm).
- Classe 02: alcance de até 10 metros, com potência máxima de saída de 2.5mW (4 dBm) e mínima de 0.25mW (-6dBm).
- Classe 03: alcance de até 1 metro, com potência máxima de 1mW (0 dBm). [7]

III. TOPOLOGIA E FUNCIONAMENTO

A forma como os dispositivos bluetooth se comunicam, formando assim uma rede é chamada de Piconet. A Piconet é uma comunicação mestre-escravo constituída por um dispositivo chamado mestre, e 7 dispositivos chamados escravos. [4]

A Piconet é uma unidade básica de uma rede Bluetooth, trata-se de um conjunto de dispositivo ligados de forma "ad-hoc", onde a Frequency Hopping é quem define qual o dispositivo ira se comunicar em cada slot de tempo. O escopo “ad hoc” é desprovido de infraestrutura ou organização central, composta por dispositivos móveis sem fio, que dada a sua mobilidade e liberdade, podem entrar ou sair da rede em

modo aleatório, ou seja não há a interrupção da transmissão, se um dos dispositivos for desconectados. [6]

Nesse tipo de comunicação mestre-escravo, o mestre tem função de solicitar os dados aos escravos e processar esses dados, enquanto os escravos somente enviam os dados quando solicitados pelo mestre. Quando os escravos estão inativos (esperando comunicação) o seu número pode crescer para até 255 dispositivos. [4]

O mestre como dito anteriormente é o dispositivo que realiza a comunicação, essa função pode ser atribuída a qualquer um dos escravos de forma dinâmica, porém deve se restringir a apenas 1 dispositivo com essa função na rede. Os escravos não podem realizar comunicação entre si. [4]

Outra topologia de rede que pode existir no bluetooth é a Scatternet. Ela corresponde a união de várias redes Piconet independentes, que podem se sobrepor ou existir na mesma área. Isso ocorre quando um dispositivo pertence de forma simultânea a outra rede Piconet, e caso ele for mestre não poderá exercer essa função em ambas as redes. A figura 1 ilustra essas topologias da rede bluetooth. [4]

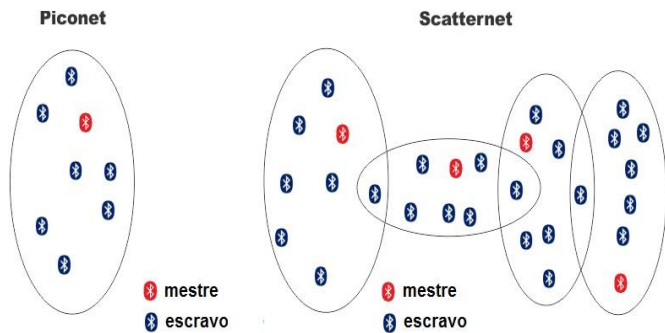


Figura 1 - Topologia do bluetooth [4]

Os dispositivos em uma Piconet compartilham este canal físico de comunicação. O compartilhamento ocorre usando o método TDMA (Time Division Multiple Access), que define slots específicos para cada um dos dispositivos conectados através do canal físico, alocados em tempos distintos, sobre esse canal. Todos os dispositivos participantes de uma Piconet utilizam a mesma frequência, porém, ao longo do tempo e de forma organizada, apenas um deles usa o canal para transmitir ou receber informações, de acordo com o slot (tempo) que lhe foi atribuído. [8]

A partir da versão bluetooth 1.2, definiu um canal Piconet adaptado que possui 2 diferenças básicas:

- As frequências nas quais um escravo transmite são as mesmas que o mestre acabou de transmitir. Ou seja, não há um salto de frequência entre um pacote do mestre e o pacote do escravo que vem logo a seguir.
- É possível excluir algumas frequências entre as 79 disponíveis para a sequência de salto de frequências, que são marcadas como fora de uso. Evita-se desta forma a utilização de frequências com alto grau de interferência. [6]

Quando ocorre um salto de frequência os seus transmissores e receptores são sintonizados ao mesmo tempo na nova frequência. A comunicação por rádio do bluetooth usa a técnica TDD (Time Division Duplex), onde a

transmissão ou recepção dos dados é feita em uma única direção no tempo, com alternância entre as duas direções, usando a mesma frequência. A vantagem desse método é a possibilidade de alocar dinamicamente largura de banda. A Figura 2 ilustra a técnica TDD, onde a mesma frequência é utilizada para transmitir e receber informações. [8]

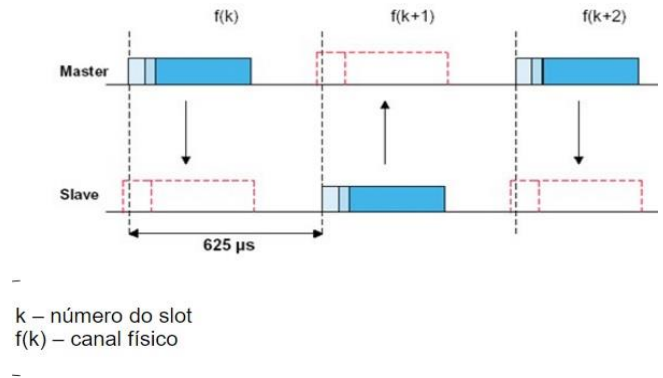


Figura 2- Método TDD [8]

Apesar de diferentes piconets na mesma área terem diferentes mestres, e, portanto, saltos de frequência diferentes, ocasionalmente duas piconets usarão o mesmo canal físico no mesmo intervalo de tempo. Para resolver esse problema é utilizado o CDMA (Code Division Multiple Access). O CDMA é uma tecnologia que utiliza espalhamento espectral (Spread Spectrum) como meio de acesso para permitir que vários usuários compartilhem uma mesma banda de frequências. [12][8]

IV. ARQUITETURA DE PROTOCOLOS

Os protocolos da rede bluetooth, assim como outros protocolos de redes, são as regras predefinidas para iniciar um certo tipo de comunicação entre dispositivos. Esse protocolo é dividido em camadas, cada uma tendo uma função diferente. É no protocolo que é definido o formato dos dados e como eles serão enviados e processados em cada camada. A figura 3 ilustra as camadas de protocolos da rede bluetooth. [2][6]

As 4 camadas de protocolos mais relevantes no estudo da rede bluetooth, indo do menor nível para o de maior são: Rádio frequência, Banda básica, gestor de enlace e o adaptador/controlador de enlace lógico:

- Rádio frequência (Bluetooth Radio): nessa camada estão definidos os aspectos físicos da transmissão de sinal como: modulação, potência e nível de sensibilidade do receptor.
- Banda básica (Baseband): essa é a camada responsável pela transmissão dos bits, ela especifica: a forma do salto de frequência (FHSS), os endereços dos dispositivos, e os tipos de pacotes e conexão. Ainda tem função de gerenciar: consumo de energia, sincronia, estado do dispositivo na Piconet e o controle de autenticação/criptografia.
- Gestor de enlace (LMP): essa camada o Link Manager faz o controle geral de todo enlace (ligação entre emissor e receptor) lógico, assim como a parametrização do enlace físico entre dispositivos.

- HCI: representa a interface de controle do host, provendo as camadas superiores uma interface padrão de acesso ao controlador e ao gerenciador de enlace.
- Adaptação e controle de enlace lógico (L2CAP): essa camada é responsável por fornecer os serviços de dados para as camadas superiores, organizando o envio de pacotes e ajustando os parâmetros de tráfego. [5][8]

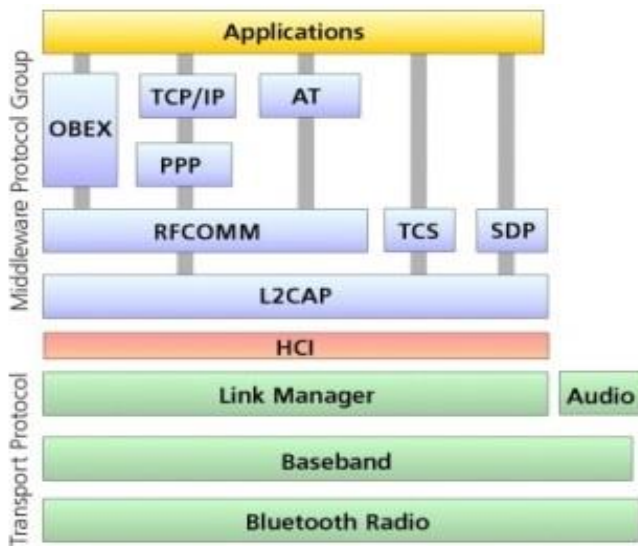


Figura 3 - Pilhas de protocolos[10]

Existem ainda os protocolos adaptados, que podem ser definidos por outras organizações. Assim é possível criar novos protocolos quando necessário. Eles são:

- RFCOMM: protocolo que cria uma porta serial convencional, permitindo que dispositivos já existentes possam ser facilmente incorporados ao serviço.
- SDP: esse protocolo permite que sejam descobertos quais serviços são disponíveis nos dispositivos bluetooth.
- Protocolo ponto-a-ponto (PPP): na tecnologia bluetooth é utilizado para efetuar ligações do tipo ponto a ponto.
- OBEX: é utilizado para que um mesmo dispositivo seja capaz de utilizar bluetooth e infravermelho.
- TCP/UDP/IP: são os protocolos responsáveis por efetuar a comunicação entre dispositivos bluetooth ligados via internet. O TCP realiza uma comunicação síncrona, e o UDP assíncrona, esses conceitos serão explicados mais adiante. [6][5]

V. MODOS DE FUNCIONAMENTO

A operação de uma piconet pode ser entendida de acordo com os estados de operação durante a comunicação. Existem dois principais estados o Standby que é o estado de baixo consumo onde as principais funções estão inativas, e o estado Conectado quando ele é um mestre ou um escravo ativo na Piconet. [5]

A arquitetura do bluetooth foi projetado para ter um baixo consumo de energia. Sendo assim ele possui modos de

operação voltados a reduzir o consumo como: sniff, hold e park. Nos dois primeiros modos o dispositivo continua sendo um membro ativo da Piconet, enquanto que no modo park se mantém sincronizado apenas com o mestre. Nos três modos elas diminuem sua participação na Piconet, diminuindo também o consumo de energia. Existe ainda o modo idle (não conectado a piconet), um dispositivo nesse modo fica ligado apenas 10ms a cada ciclo de tempo, que pode variar de 1,28 a 1,34 segundos. [5]

Para que o dispositivo bluetooth descubra o endereço de unidades próximas a ele, sem a necessidade de intervenção do usuário, existem três procedimentos: inquiry, page e scan. O scan garante que os dispositivos que estiverem ociosos possam entrar em modo de economia de energia. Assim eles acordam para verificar se há outro dispositivo tentando se conectar por uma portadora, esse intervalo de varredura é de aproximadamente 10ms. O page é utilizado pelo dispositivo que quer estabelecer uma nova conexão. E o inquiry é utilizado pelo dispositivo que deseja descobrir outros dispositivos na sua área de alcance, para reduzir colisão quando há mais de uma resposta existe uma temporização aleatória. [3]

No que se refere a enlace, o bluetooth utiliza basicamente dois padrões: SCO (conexão síncrona) e ACL (conexão assíncrona). No SCO é estabelecido um link sincronizado entre o dispositivo emissor e o dispositivo receptor, separando um canal para comunicação. Assim ele é utilizado principalmente em aplicações de envio contínuo de dados, como transmissão de voz, e opera com uma taxa de transmissão de 432kbps. Por funcionar dessa maneira ele não permite a retransmissão de pacotes perdidos. Quando ocorre perda em uma transmissão de áudio, por exemplo, o receptor acaba reproduzindo som com ruído. O padrão ACL, estabelece um link entre o dispositivo que inicia e gerencia a comunicação e os demais que estão em sua rede. Esse link é assíncrono, pois utiliza canais previamente livres. Ele permite o envio de pacotes de dados perdidos, garantindo a integridade das informações enviadas. Sendo assim é mais utilizado para aplicações que envolvam transferência de arquivos. Ele opera com uma taxa de transmissão de 721kbps. [7]

VI. PACOTE DE DADOS

A figura 4 mostra a estrutura geral de um pacote bluetooth.



Figura 4 – Formato geral de um pacote bluetooth [13]

O Access code (72 bits) constitui a primeira parte do pacote, permitindo a sincronização das unidades da piconet, a compensação do componente DC, e a identificação da piconet ou do dispositivo endereçado. [13]

O Header (54 bits) corresponde ao cabeçalho, que é utilizado para identificar o tipo de pacote e a informação de controle do protocolo de transmissão. [13]

O conteúdo do campo de payload (0 a 2745 bits) depende do tipo de pacote enviado; de forma geral, quando o pacote em questão não é de controle, este campo pode ser composto por um campo de voz, um campo de dados, ou ambos. Um

campo de dados é geralmente subdividido em 3 campos: um de cabeçalho, um com o corpo da mensagem, e um terceiro com o código CRC para a verificação de erros de transmissão. [13]

A estrutura do código de acesso (Acces Code) pode ser visualizada na Figura 5.

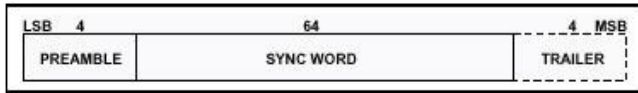


Figura 5 – Código de Acesso [13]

- O preâmbulo é composto por uma sequência de 0's e 1's utilizada para compensação DC.
- O campo de sincronismo (sync word) é composto por um código derivado da parte baixa do endereço do dispositivo, denominado LAP (Lower Adress Part).
- O campo trailer é composto por 0's e 1's alternados, isso permite que quando um cabeçalho (header) é enviado se tenha uma melhor compensação DC. [13]

A estrutura do cabeçalho (Header) pode ser visualizada na Figura 6.

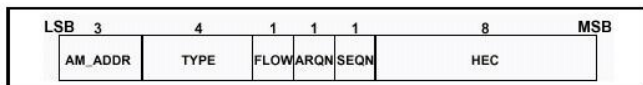


Figura 6 – Cabeçalho [13]

- AM_ADDR: Endereço do membro ativo. Constitui-se de um rótulo de 3 bits único para cada escravo ativo na piconet. O rótulo 0 é utilizado para mensagens de broadcast.
- TYPE: é o tipo do pacote, ele permite saber o número de slots de tempo que o pacote em questão irá ocupar.
- FLOW: é utilizado para controle fluxo no caso de pacotes ACL. 0 interrompe temporariamente, e 1 libera o fluxo de transmissão.
- ARQN: 1 indica sucesso e 0 indica falha no envio de um pacote ACL de informação com CRC.
- SEQN: determina a sequência de um pacote de informação com CRC, de forma que o receptor possa identificar retransmissões de pacote.
- HEC (Head Erro check): permite a verificação da integridade do cabeçalho enviado. [13]

Como foi dito anteriormente o campo payload de um pacote de dados bluetooth dependo do tipo dos dados, e os tipos de pacotes que podem ser enviados são: os específicos do link SCO, os específicos do link ACL e os comuns a ambos. [13]

Exemplos de pacotes comuns são o ID (identidade) que é utilizado em rotinas de interrogação e o POLL (pacote poling) que verifica se existe algo a ser transmitido. [13]

Um exemplo de pacote SCO é o DV que permite o envio conjunto de voz e dados. [13]

Já os pacotes de ACL são utilizados para envio de dados sem restrição de tempo. [13]

VII. VERSÕES

A tecnologia bluetooth vem evoluindo ao longo dos anos. Para cada evolução significativa do sistema, surgem novos modelos, os principais que podem ser citados:

- Bluetooth 1.0: foi a primeira versão lançada, o que acabava dificultando a interoperabilidade entre dispositivos via bluetooth. Com velocidade padrão de 721 Kbps.
- Bluetooth 1.1: o sistema passou a ter suporte para conexões não criptografadas.
- Bluetooth 1.2: nessa versão houve um avanço significativo na velocidade, chegando até 1Mbps. Teve melhoras no controle de interferências, e passou a retransmitir pacotes corrompidos, melhorando o processamento de voz.
- Bluetooth 2.0 + EDR: anunciado em 2004, essa versão teve como principal melhoria o aumento de velocidade para 3Mbps. Outras melhorias foram na diminuição de consumo da bateria, na correção das falhas da versão 1.2 e melhor comunicação. O padrão EDR (Enhance Data Rate) não era obrigatório, mas foi o responsável por triplicar a velocidade.
- Bluetooth 2.1 + EDR: nessa versão ouve melhoria nos protocolos de segurança, informação de conexão e consumo de energia. Ouve ainda um acréscimo das informações nos sinais Inquiry e melhorias na segurança.
- Bluetooth 3.0 + HS: lançado em 2009, utiliza o protocolo 802.11 a mesma utilizada pelo wifi. Possui taxas altas de velocidade na transferência dos dados, na faixa de 24Mbps. Outra vantagem é o consumo inteligente de energia.
- Bluetooth 4.0: seu diferencial é no consumo de energia, este padrão exige muito menos eletricidade quando o dispositivo está ocioso. [8]

VIII. SEGURANÇA

Como dito anteriormente o bluetooth opera em uma faixa de frequência compartilhada. E utiliza FHSS que realiza divisão em até 79 canais e até mudando entre eles até 1600 vezes por segundo, porém ainda podem ocorrer interferências. [13]

Para garantir a segurança dos dados trafegados, o protocolo possui níveis de segurança: autenticação, modo inseguro, Serviço Level Security (SLS) e serviço Link Level Security (LLS). [13]

No nível de Autenticação não é permitido o envio de informações de origens duvidosas, e acessos não desejados a funções e dados importantes do dispositivo. No modo Inseguro não existem aplicações críticas, ou seja, só trafegam dados sem grande importância. [13]

O SLS permite um acesso dinâmico para acionar aplicações com diferentes níveis de segurança simultaneamente. [13]

O nível de segurança LLS é comum a todas as aplicações que são iniciadas neste nível, mantem um nível de segurança comum. Quando é exigida, a comunicação do protocolo faz uso de uma criptografia de 128 bits solicitada pelo dispositivo, sendo que a informação só é liberada após a confirmação do PIN (Número de identificação Pessoal) disponibilizado assim, a troca de informações. [13]

IX. VANTAGENS E DESVANTAGENS

A tabela 1 faz um comparativo entre a rede bluetooth e outros protocolos de comunicação sem fio.

Tabela 1 - Comparação dos padrões IEEE.802 [4]

Norma IEEE (Nome de Mercado)	802.15.1 (Bluetooth)	802.11b (Wi-Fi)	802.15.4 (ZigBee)	802.15.4 (Wireless HART)	802.15.1 (WISA)
Aplicação Principal	Eliminar a fixação atual,	Ethernet industrial	Controle e Monitoração	Medições de Processo e Controle	Interface Sensores e Atuadores
Frequência de operação	2,4 GHz	2,4 GHz	2,4 GHz	2,4 GHz	2,4 GHz
Taxa de comunicação (Kbps)	1000-3000	11000	20-250	11000	1000
Distâncias alcançadas com visada direta (m)	30(Classe 2) 100+(Classe 1)	100+ (Antenas direcionais)	30-70, 100+ (Com amplificador externo)	100, até 2000 com repetidor	5 m (ambiente industrial; tipicamente 10 m)
Número de Dispositivos	7	32	2*64	250	120
Autonomia da bateria (dias)	1 a 7	0,5 a 5	100 - 1000+	3,5 anos: taxa de trans de 30 min.	3 a 4 anos
Consumo na Transmissão	45mA (Classe 2) <150mA (Classe 1)	300mA	30mA	150-300mA	100mA
Conveniência para Controle e Supervisão, aplicações industriais	Baixa (Boa média, mas conexão inicial lenta)	Baixa (Taxa alta, mas conexão inicial lenta)	Baixa (Bom compromisso entre taxa e custo de conexão)	Alto (Bom compromisso entre taxa e custo de conexão)	Alto (Boa taxa e custo de conexão)
Tecnologia de espalhamento espectral	FHSS	DSSS	DSSS	FHSS/DSSS	FHSS/DSSS
Vantagens relativas	Custo, Flexibilidade	Velocidade, Flexibilidade	Potência, Custo	Flexibilidade, Potência e Velocidade	Velocidade, Custo

O padrão que possui aplicações semelhantes à do bluetooth é o Zigbee. Ambos os protocolos são classificados como WPAN que são as redes pessoais sem fio, cujo alcance é de 10m. Enquanto o bluetooth permite 8 dispositivos se comunicando nessa área, o zigbee possui capacidade de 65000. [2] [9]

A vantagem do bluetooth em comparação com o zigbee está na comunicação, que possui uma alta velocidade de resposta na faixa de 2,5ms, enquanto o zigbee leva em torno de 15ms. Um fator que pode pesar um pouco na escolha entre esses dois dispositivos é o consumo da bateria, que como o bluetooth altera constantemente o modo mestre escravo tem um consumo um pouco maior de energia. [9]

O protocolo de rede bluetooth tem na sua versão 1.0 uma taxa de transferência de dados (bitrate) de 1024kbps e a versão mais recente 4.0 uma velocidade de 24576kbps. Enquanto o zigbee tem um bitrate de 20 a 250kbps. Esse é um fator que também influencia no consumo de energia, é o bluetooth tem uma ligeira vantagem. [9]

Esses dados evidenciam que o bluetooth não é uma solução tão adequada na automação industrial, pois tem um consumo superior de energia e é mais sensível a interferências eletromagnéticas quando comparado ao zigbee. Ainda assim seu consumo pode ser considerado baixo, e continua relevante em outras aplicações que não sejam tão hostis. [2]

Outra tecnologia de conexão sem fio muito comparada ao bluetooth é a IrDa ou conexão via porta infravermelhas. Cada

tipo de tecnologia nessa comparação tem suas vantagens, desvantagens e recomendações variadas de usos. O infravermelho, por ser mais seguro, inofensivo à saúde e não interferir em outros sinais, pode ser recomendado a lugares que necessitem maior sigilo de informações ou cuidados com equipamento, como hospitais e escritórios de empresas, mas ainda extremamente limitado em termos de alcance e praticidade. Já o bluetooth é acessível, e por ser uma tecnologia direcionada para compartilhamento e transferência de dados, é rápido e prático, porém como foi visto ainda tem alguns problemas com interferência. E grande parte dos celulares já possuem Bluetooth implantado. [11]

Bluetooth tem uma desvantagem quando se trata de apresentar velocidade com alta taxa de comunicação. Em comparação com a taxa de transferência de até 4.0Mbps de infravermelho, Bluetooth só pode chegar a até 1.0Mbps, o que significa que transfere arquivos lentamente. Ao transferir ou compartilhar arquivos maiores, a uma distância mais perto, outras tecnologias sem fio (como infravermelho) são mais eficientes. [9]

A conexão Bluetooth pode ser firmemente estabelecida, sem a interferência de dispositivos não reconhecidos digitando um número de identificação para a conexão. O usuário do dispositivo mestre (aquele que começa a conexão) pode optar por introduzir um PIN que cria o loop uso seguro. Outra vantagem em segurança é que a menos que um dispositivo já está emparelhado com o dispositivo, existe a opção de aceitar ou rejeitar a transferência de conexão e de arquivo. Isso evita que os arquivos desnecessários ou infectados de usuários desconhecidos sejam transferidos para o dispositivo. [9][2]

X. APLICAÇÕES

O bluetooth é uma das redes wireless mais flexíveis, podendo ser utilizado em vários equipamentos. Tendo a disposição um aparelho celular, notebook, câmera fotográfica, impressora ou outros dispositivos eletrônicos é possível se ter uma aquisição de arquivos, enviar mensagens, fazer ligações, etc. sem que esse esteja conectado por meio de fios.

Uma aplicação que ganhou força nos últimos anos foram as conexões de periféricos via bluetooth, como: mouses, fones de ouvido, teclado e outros aparelhos onde essa comunicação de curta distância pode eliminar a utilização de fios. Outra aplicação muito útil é a transmissão simultânea de dados adquiridos, como por exemplo uma câmera fotográfica que transmite as fotos assim que tiradas para um notebook próximo, ou uma caneta que o que é escrito é simultaneamente transmitido de forma digitalizada ao notebook. [5]

Outra aplicação que poderá se tornar popular são as redes sem fio. Apesar da velocidade de transmissão do Bluetooth ser mais baixa que permitida pelas redes Ethernet (10 ou 100 Mbps), caso os desktops e notebooks passassem a vir com os transmissores Bluetooth embutidos, muita gente poderá preferir utilizar o Bluetooth pela praticidade de conexão, já que 1 Mbps já são suficientes para compartilhar uma conexão com a Internet via ADSL ou cabo, e transmitir pequenos arquivos. Com essa largura de banda aplicações de vídeo digital ainda não estão no escopo do Bluetooth. [1]

Existem aplicações das mais variadas que vem sendo testadas com essa tecnologia, como a carteira eletrônica, e na área de automação residencial para que os dispositivos se comuniquem no controle do ambiente. Alguns exemplos de tecnologias atuais onde o bluetooth é empregado podem ser vistos na figura 7.

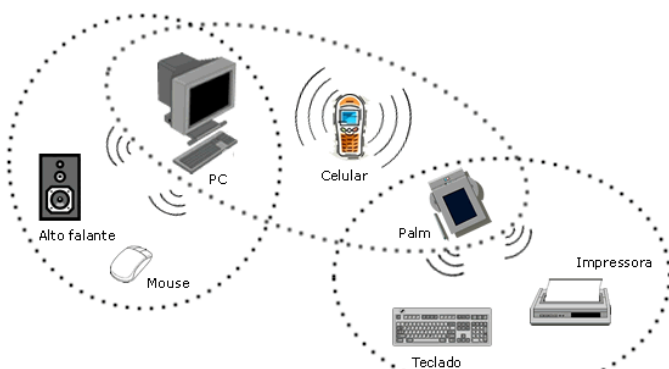


Figura 7 - Aplicações do bluetooth [6]

Existem ainda aplicações novas surgindo, como equipamentos médicos como dosadores de medicamentos que além de informar as necessidades diárias, enviam informações aos médicos das condições saúde dos pacientes. Existem ainda aplicações nos esportes, sendo utilizado para envio de imagens de prova esportivas, a câmera capta as imagens que são enviadas via Bluetooth para um celular em tempo real, e o envio de informações sobre desempenho de atletas através de uma palmilha, que foi um protótipo lançado em 2010 que permite o acompanhamento e a transferência dessas informações via Bluetooth para um dispositivo móvel. [11]

A indústria automobilística é outro setor que também pode ser beneficiado. Os cabos que levam algumas informações – como pressão e temperatura dos pneus, por exemplo – para o painel do carro, no futuro poderão ser substituídos pelos chips Bluetooth. [11]

XI. FUTURO DO BLUETOOTH

Sempre existiu uma certa dúvida se o bluetooth conseguiria se manter no mercado, afinal surgiram novas redes como o wifi que ficaram muito populares. Porém tem ficado muito claro nos últimos anos que cada vez mais dispositivos chegaram com esse recurso.

Quando se necessita de uma comunicação de curto alcance, o bluetooth possui um custo reduzido para uma alta aplicação. Portanto se obtiver sucesso, deverá ser para dispositivos pequenos, que não requeiram alta taxa de transmissão de dados. Ele não deverá ser muito aplicado em áreas que necessite de uma taxa de transferência de dados maior, tendo em vista que já existem tecnologias que atendem melhor a esta necessidade.

Em 2008 o grupo Bluetooth SIG comemorou 10 anos da chegada do bluetooth no mercado. Como foi citado no começo do trabalho o grupo era formado por 5 empresas, e hoje esse número passa de 10 mil. Isso evidencia ainda mais o crescimento dessa tecnologia. [8]

Com as facilidades proporcionadas e por ser relativamente barato, o Bluetooth 4.0 tem sido apontado como o motor para expandir a internet das coisas. [8]

XII. CONCLUSÃO

O bluetooth é uma tecnologia criada há 18 anos, e apesar de recente, está muito presente nos mais variados equipamentos como celulares, impressoras, computadores, headsets, etc. Ele foi desenvolvido pela empresa Ericsson, com objetivo de realizar comunicação sem fio (wireless) entre equipamentos que estejam a uma curta distância (cerca de 10m). [1]

Ele representou uma inovação na área de comunicação, com uma comunicação eficiente, relativamente segura e de baixo custo. Tem evoluído ao longo dos anos, com novos modelos que atendem as tendências do mercado. Os principais fatores que se procura melhorar são a velocidade de transmissão e reduzir o consumo de energia. Encontra-se atualmente na versão bluetooth 4.0. [8]

O bluetooth foi definido como padrão IEE 802.15.1, e é voltado para redes pessoais. Ele já apresenta diversas aplicações nas áreas de medicina, esportes, militar e automobilística. Tendo em mente o crescimento exponencial dessa tecnologia, junto com a utilização dele em smartphones e outros aparelhos móveis, se tem uma boa perspectiva do crescimento e potencial dessa tecnologia que já é mundialmente conhecida. [4]

REFERÊNCIAS

- [1] Kobayashi, Carlos Yassunori (2004) A tecnologia bluetooth e aplicações. USP. Disponível em: http://grenoble.ime.usp.br/movel/monografia_bluetooth.pdf
- [2] Costa, Rafael Antônio Asatiany & Mendes, Luis Augusto Mattos (2016, 01 julho) Evolução das redes sem fio: um estudo comparativo entre bluetooth e zigbee. UNIPAC/FACEC. Disponível em: <http://www.unipac.br/site/bb/tcc/tcca010b188f93af4c28ca9af23b9e3c476.pdf>
- [3] Loureiro, Antônio AF (2016, 01 julho) Base para bluetooth. Disponível em: http://s3.amazonaws.com/academia.edu.documents/34741145/Base_para_Bluetooth_1.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1467514332&Signature=XXvotcKN%2FqLeVYXHFDxwJdLN3jY%3D&response-content-disposition=inline%3B%20filename%3DBLUE%20TOOTH%20TECHNOLOGY%20IEEE%20802.15.pdf
- [4] Viana, Hugo Magalhães & Baratella, Alexandre Lugli (2016, 01 julho) Redes sem fio para automação industrial: estudo e aplicação. INATEL.
- [5] Loureiro, Antônio AF & Nogueira (2003) Redes de sensores sem fio. Disponível em: <http://homepages.dcc.ufmg.br/~loureiro/cm/docs/sbr03.pdf>
- [6] Teleco inteligência em telecomunicações (2016, 01 julho) Bluetooth o que é? Disponível em: <http://www.teleco.com.br/pdfs/tutorialbluezig.pdf>
- [7] Sacks, Anelise Godolphim (2016, 01 julho) Sistema de gerenciamento de redes e processos através de computadores portáteis via bluetooth. Disponível em: <http://www.gta.ufrj.br/ftp/gta/TechReports/Anelise03/Anelise03.pdf>
- [8] Alecrim, Emerson (2016, 01 julho) Tecnologia bluetooth o que é e como funciona? Disponível em: <http://www.infowester.com/bluetooth.php>

- [9] Junior, André Lisboa da Conceição (2016, 01 julho) Redes sem fio: estudo comparativo dos protocolos bluetooth e zigbee aplicados em ambientes industriais. Disponível em: <http://www.teleco.com.br/pdfs/tutorialbluezig.pdf>
- [10] Escola politécnica UFRJ (2016, 01 julho) Redes de sensores sem fio. Disponível em: http://www.gta.ufrj.br/grad/10_1/rssf/protocolos.html
- [11] Junior, Andre Lisboa da Conceição (2016, 01 julho) Protocolo bluetooth aplicado em interconexão de dispositivos. Disponível em: <http://www.teleco.com.br/pdfs/tutorialredespaid.pdf>
- [12] Teleco inteligência em telecomunicações (2016, 01 julho) CDMA arquitetura. Disponível em: www.teleco.com.br/tutoriais/tutorialcdma/pagina_1.asp
- [13] Stein, Paulo (2016, 01 julho) Bluetooth. Disponível em: http://www.gta.ufrj.br/seminarios/semin2003_1/stein/pacotes.html

Redes de Comunicação Automotiva: Apresentação e perspectivas

Gabriel Alves Silvestre

Universidade Federal de Santa Catarina – Campus Blumenau

Departamento de Engenharia

Rua João Pessoa, Nr. 2750, Velha

Blumenau/SC, Brasil, 89036-256

Resumo – A constante evolução do sistema automotivo esta ligada diretamente ao crescimento da eletrônica e dos protocolos que permitem interligar os diversos processos como controle de temperatura, alarme e até mesmo frenagem. Cada um destes processos necessita de um protocolo específico que atenda de maneira adequada os requisitos. Para facilitar a padronização de tais protocolos, foi proposta uma divisão dos tipos de redes automotivas em classes. Este artigo apresenta os tipos de redes, e faz um breve resumo dos principais protocolos abordando o funcionamento e suas vantagens.

Palavras-Chave: Sistemas Intraveiculares, Protocolo de comunicação.

I. INTRODUÇÃO

Não é novidade que o mercado automotivo busca cada vez mais na tecnologia uma forma de atrair clientes e reduzir os custos de produção. Para satisfazer ambos, reduzir custos e atrair clientes, a eletrônica tem papel de destaque já que através dela a produção é barateada e ao mesmo tempo há uma melhora significativa no desempenho e eficiência do veículo automotor. Além de atender a estas duas necessidades, os dispositivos eletrônicos também vem se destacando pois podem substituir seus semelhantes mecânicos com segurança e confiabilidade semelhantes, e conforto e comodidade muito superiores.

Constatadas estas vantagens, a aplicação da eletrônica apenas aumenta, e este aumento é traduzido no aumento do número de elementos eletrônicos presentes nos veículos. Acompanhando este crescimento, vem a necessidade de integrar todos estes elementos (sensores, atuadores) a uma (ou várias) centrais de controle. Tais centrais são responsáveis por comandar todo o funcionamento, desde o acendimento do farol ao sistema de freios. Para realizar esta integração, entre sensores, atuadores e centrais de comando é fundamental definir protocolos e modelos de comunicação, já que é com base nestas informações que serão tomadas todas as decisões.

Criando barramentos¹ e com projetos de controladores específicos, estes modelos e protocolos (por

vezes chamados de padrões) são de suma importância, pois facilitam a comunicação reduzindo o número de fios, a quantidade de processamento e conseqüentemente o custo final. Mais do que isso, possibilitam uma gama de novas estruturas, por exemplo, reduzindo o número de fios, há espaço para mais sensores, para um novo design, e o somado a tudo isso vem à facilitação do projeto.

Neste trabalho serão abordados alguns destes modelos, trazendo suas vantagens, desvantagens e área de abrangência. Além dos citar os principais padrões existentes, detalhando seu funcionamento, também serão analisados alguns módulos eletrônicos, pois a construção destes também sofreu alterações devido à necessidade de comunicação com sensores, atuadores e até mesmo com outros módulos.

Previamente serão apresentados os circuitos eletrônicos, descrevendo suas peculiaridades, as quais são responsáveis pela captura, transmissão, e envio de dados e informações. A seguir, são citados os modelos e seus modos de funcionamento. Ao final, a conclusão trata das perspectivas para o futuro, mostrando alguns dos desdobramentos possíveis, e esclarece quais pontos devem ser otimizados e quais pontos têm funcionado de maneira correta.

II. ESTRUTURA ELETRÔNICA

O sistema eletrônico automotor passou por diversas mudanças ao longo do tempo, mudanças estas que trouxeram grandes facilidades tanto por aumentar a eficiência energética como por proporcionar maior conforto e comodidade.

Dentre os componentes do sistema eletrônico podemos citar sensores de estacionamento, sensor de umidade (para ligar automaticamente os limpadores), sistema de injeção eletrônica e frenagem automática. Para interligar todos estes sistemas e sensores, e possibilitar uma ação correta, as centrais eletrônicas (também conhecidas por módulos eletrônicos) tem papel fundamental desde o controle do tráfego de informações, até o comando de um atuador.

A Unidade de Controle Eletrônico (ECU - do inglês, Electronic Control Unit) é o módulo responsável por organizar e controlar todas as ações, ou seja, é o centro por onde todas as informações vindas de sensores devem passar. Com tais informações, é feito o processamento e com base no software e na lógica implementada são tomadas as decisões finais que devem repassadas aos atuadores, por exemplo.[1][2]

Em relação à arquitetura de ECU's temos dois tipos: arquitetura centralizada e arquitetura distribuída.

¹ Barramento é o termo técnico utilizado para fazer referência aos condutores elétricos das linhas de comunicação e a forma como eles são montados.

Na arquitetura centralizada, apenas uma unidade de controle recebe os dados dos sensores e interruptores, processa e os envia aos atuadores. Dentre as vantagens dessa configuração está a simplicidade, já que, é necessário projetar e implementar apenas uma unidade lógica, como fatores negativos destacam-se o grande número de cabos (pois todos os dispositivos precisam de um conexão com a ECU)[4].

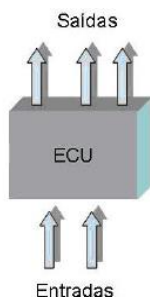


Fig. 1 – Arquitetura Centralizada[4].

Já na arquitetura distribuída, várias unidades de controle estão presentes no sistema. Tem como vantagens a simplificação, devido ao número menor de fios(já que as ECU's podem ser instaladas diretamente no sistema controlado), controle mais robusto (também devido ao úmero menor de fios) e facilidade no aumento do número de nós (dispositivos) conectados. Para interligar todas as unidades de controle utiliza-se um barramento de comunicação, que tem por função levar as informações de um sensor a uma ECU, de uma ECU a outra, e de uma ECU a um atuador.

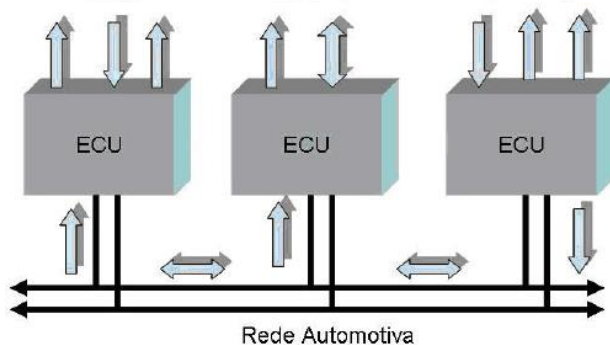


Fig. 2 – Arquitetura Distribuída[4].

Justamente a criação deste barramento, esta ligada a uma das desvantagens, que é a necessidade de um protocolo de comunicação. Este protocolo está diretamente ligado ao tempo de resposta do sistema e aos possíveis erros de transmissão. Neste cenário, surgem os diversos protocolos de comunicação, na tentativa de criar modelos e padronizar os sistemas.

III. PROTOCOLOS

Protocolo é um conjunto de regras pré-definidas para o gerenciamento da comunicação e, em particular, para conceder acesso ao barramento por parte dos nós. Protocolos de comunicação automotivos são meios de transmissão e recepção de dados utilizados para intercomunicarem módulos eletrônicos e/ou sensores e atuadores inteligentes equipados com microcontroladores e transceivers, por exemplo[1].

Não existe um protocolo certo, e que seja de uso universal, já que existem diversos fabricantes de veículos, e, cada um destes tem diversos tipos de veículos, desde uma motocicleta até um caminhão e cada um deles tem problemas de comunicação diferentes. Para distinguir estes problemas, existe uma classificação das redes de comunicação automotivas. Esta classificação pode levar em consideração diversos fatores, tais como, funcionalidade, capacidade de transmissão, satisfação dos requisitos operacionais, facilidade de integração e manutenção[2].

IV. CLASSIFICAÇÃO DAS REDES AUTOMOTIVAS

O SAE (Society of Automotive Engineers) apresenta uma classificação das tecnologias de redes automotivas em função dos requisitos das aplicações. São descritas da seguinte forma: Classe A, Classe B e Classe C[5]:

A. Classe A

São redes de comunicação com baixa largura de banda utilizadas em função de conforto e diagnóstico como vidro elétrico, retrovisor, controle de bancos, lâmpadas, etc. Alguns exemplos de rede automotivas do tipo Classe A são: UART, I2C, BEAN, A-BUS, LIN, TTP/A entre outras. Em uma arquitetura distribuída com rede Classe A pode se encontrar ECUs para controle de teto solar, dashboard e controle de bancos do motorista e passageiro[5]. A velocidade de transmissão varia até 10kbps (kbps = 1000 bits por segundo)[1].

B. Classe B

São redes utilizadas para aplicações que são importantes, mas não essenciais para a operação do automóvel, como display de informação de velocidade, controle de nível de combustível, powertrain (comunicação de tarefas), etc. Alguns exemplos de rede automotiva do tipo Classe B são: VAN, J1850, J9139 e CAN. Na arquitetura distribuída com rede Classe B nota-se a presença de ECUs para controle do motor, transmissão, amortecedor entre outras funções[5]. Taxa de transmissão de até 125kbps [1].

C. Classe C

São redes utilizadas em aplicações de segurança crítica com requisitos de tempo real e tolerantes a falhas em que esteja diretamente ligada à dinâmica do automóvel. Aplicações baseadas na tecnologia *x-by-wire* requerem que as redes ofereçam transmissão de dados com baixo atraso de transmissão, alta frequência, tolerância a faltas e outros mecanismos essenciais para segurança crítica. Em veículos com uma arquitetura distribuída com rede Classe C, existem ECUs para realizar funções críticas como *steer-by-wire* e *brake-by-wire*[5]. As velocidades são superiores a 125kbps[1].

Tendo definidas as classes de redes automotivas, é possível agora estudar os vários tipos de redes empregados em cada uma destas classes.

IV. REDES AUTOMOTIVAS CLASSE A

Uma rede classe A, requer baixa velocidade e pouca transmissão de dado. É responsável por ações não críticas, como conforto (entretenimento e áudio) e diagnóstico.

Conhecendo os principais requisitos de uma rede classe A, veremos a seguir alguns dos tipos de rede desta classe.

A. UART

A UART (do inglês, Universal Asynchronous Receiver/ Transmitter)[6] é utilizada para comunicação a maiores distâncias, consiste em um dispositivo com um software(programa) que regula a comunicação entre dispositivos da rede. Tem velocidade limitada, pois a sincronização é feita por software. A figura 3 ilustra a sequência de bits da UART:

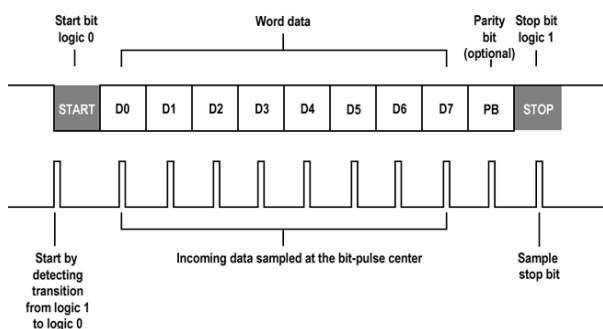


Fig. 3 – Sequência de bits da rede UART[7].

Para iniciar a transmissão existe um bit de início (start bit), após este bit são enviados os bit que contém a informação a ser transmitida, e por fim, o último bit sinaliza o fim da transmissão(o stop bit). Também é comum adicionar um bit de paridade(parity bit), que verifica a paridade de bytes de entrada e enviado este dado, então ao chegar ao destino são comparadas as paridades.

B. SPI

SPI(do inglês, Serial Peripheral Interface) é um protocolo serial de comunicação síncrona desenvolvido pelo Motorola para simplificar a interface entre microcontroladores e seus periféricos, podendo também ser usada para comunicação entre microcontroladores.

Funciona no esquema mestre-escravo, em que o haverá apenas um mestre (master), e todo o resto conectado a rede será escravo (slave). Os pinos básicos de comunicação entre dispositivos SPI e o esquema padrão de ligação podem ser vistos na tabela I e na figura 4:

TABELA I – PINOS DO SPI[8].

Pino	Nome Padrão	Significado	Nomes Alternativos
Do Master para o Slave	MOSI	Master Output Slave Input	SDO, DO, SO
Do Slave para o Master	MISO	Master Input Slave Output	SDI, DI, SI

Clock	SCLK	Serial Clock	SCK, CLK
Seleção de Slave	SS	Slave Select	CS, nSS, nCS

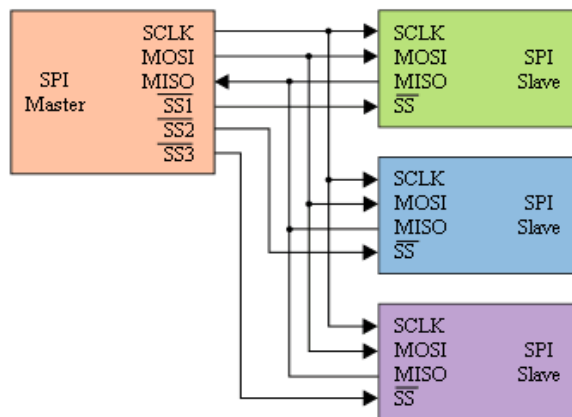


Fig. 4 – Esquema da rede SPI, como descrito antes, só pode haver um mestre. Também nesta figura é possível ver os pinos de comunicação[9].

Além disso, é possível definir o bit mais significativo e alterar o clock da comunicação para borda de subida ou borda de descida, por exemplo.

O envio e o recebimento de informação podem ser feitas simultaneamente, e não há a necessidade de esperar o fim de um para o início do outro[8]. Essa comunicação é chamada de full-duplex. O SPI não apenas permite que os dados sejam transmitidos e recebidos simultaneamente, como é uma exigência de hardware. Cada bit de dado enviado do mestre para o escravo transfere também um bit de dado do escravo para o mestre.

Entre as vantagens destacam-se a facilidade de implementação, custos energéticos menores que redes semelhantes e comunicação full duplex. Como desvantagens tem-se a limitação de apenas um mestre na rede, falta de padronização(existem diversas variantes) e um algoritmo de detecção de erro ineficaz.

C. LIN

O LIN (Local Interconnect Network) surgiu como uma solução de baixo custo cujo protocolo foi especificado pelo LIN Consortium, uma iniciativa formada por cinco indústrias automobilísticas Europeias (Audi AG, BMW AG, Daimler Chrysler AG, Volkswagen AG e Volvo Cars Corporation AB), o grupo Motorola e o grupo Vulcano Communication Technology (VCT). É um padrão de comunicação veicular e sua especificação do é mantida pelo LIN-consortium.

Nasceu da necessidade de baratear custos, ou seja, é uma solução de baixo custo com a finalidade de dar suporte a redes ‘maiores’, em outras palavras, atua como uma sub-rede. A velocidade máxima de transmissão de dados chega a 20kbps, sendo que permite mensagens multicast e broadcast. Entre suas vantagens estão a necessidade um único fio, baixo custo, e padronização.

É baseado na estrutura mestre-escravo em que um mestre pode ter até 15 escravos a uma distância de até 40 metros. O nó escravo executa apenas a tarefa escrava (slave task), mas o nó mestre executa a tarefa escrava e a tarefa mestre (master task). A tarefa do mestre é gerar a tabela de escalonamento das mensagens produzir o cabeçalho do frame. A tarefa do escravo é responder ao cabeçalho do frame e repassar as mensagens recebidas para a camada de aplicação[10].

É um dos protocolos mais indicados para ações não críticas, tanto pela simplicidade quanto pelo baixo custo.

O Cabeçalho do LIN é composto pelos campos Break, Sync e Identificador Protegido. O subframe de Resposta é composto de 1 a 8 bytes de dados seguido pelo campo Checksum. O campo Checksum é gerado pelo resto da divisão, da soma de todos os valores, por 256[10].

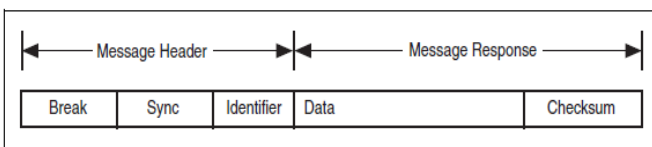


Fig. 5 – Cabeçalho do LIN[7].

V. REDES AUTOMOTIVAS CLASSE B

Uma rede classe tem velocidades de até 125 kbps, é empregada em tarefas mais importantes que a classe A, porém não críticas como a classe C. Está presente em funções como: controle do nível de combustível e monitoramento do motor e transmissão.

A. CAN

O CAN (Controller Area Network) é sem dúvidas uma das redes automotivas mais reconhecidas e de maior sucesso do mercado. É um protocolo de comunicação serial baseado no sistema multi-mestre que utiliza módulos para realizar a comunicação. Foi desenvolvido pela empresa BOSCH em meados dos anos 80, nasceu da necessidade de um protocolo padrão (para atender periféricos de diversas empresas) com baixo custo e alta confiabilidade.

Para atender a estes requisitos a rede CAN utiliza o CSMA/CD², que verifica (escuta) o barramento antes de enviar os dados, e caso alguma outra estação esteja transmitindo espera até que o canal esteja livre. Além disso, bits de prioridade definem quem tem preferência na transmissão, assim tarefas mais importantes (críticas) sempre terão prioridade em relação a ações simples e não críticas. Também conta com um robusto controle de detecção e correção de erros. Outro conceito bastante interessante é o NRZ (Non Return to Zero), onde cada bit (0 ou 1) é transmitido por um valor de tensão específico e constante[12].

Cada pacote de dados ou datagrama é transmitido em tempos que variam entre 0.25 e 20 milissegundos,

² CSMA/CD significa, em inglês, carrier-sense multiple access with collision detection (acesso múltiplo com detecção de portadora e detecção de colisão) e descreve como o protocolo de Ethernet regula a comunicação entre os nós de uma rede.

dependendo da unidade e da função, porém a velocidade varia com a distância, sendo assim, quanto maior a distância menor será a velocidade de transmissão. Em geral, as distâncias costumam ser de alguns metros[12].

A organização dos bits de uma rede CAN é mostrada a seguir:

- Começo - esse campo utiliza um bit para marcar o começo da transmissão ou recepção de dados.
- Estado - esse campo utiliza 11 bits para a definição da prioridade do protocolo.
- Controle - são 6 bits que definem a quantidade de informações que vai ser recebida ou enviada.
- Dados - são as informações transmitidas ou recebidas propriamente ditas.
- Verificação - este campo é composto por 16 bits que são utilizados para detectar falhas na transmissão dos dados.
- Conformação - são dois bits utilizados para indicar que os dados foram recebidos corretamente.
- Fim - esse campo tem 7 bits que são utilizados para indicar o fim da transmissão do "pacote" de dados[3].

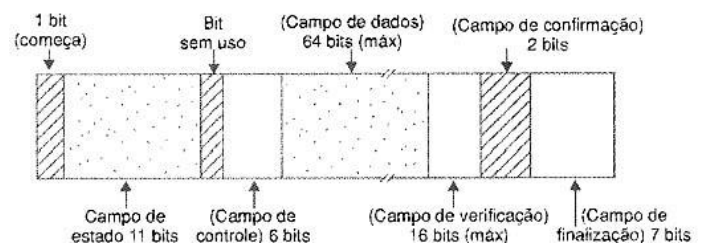


Fig. 6 – Esquema do Datagrama do CAN[3].

As vantagens do Controller Area Network são: facilidade e na instalação, flexibilidade para possíveis alterações, interoperabilidade com sensores/atuadores de diversos fabricantes, redução do número de fios devido ao uso do barramento (e conseqüente redução dos custos) e confiabilidade em razão do uso de dois barramentos (redundância).

B. SAE J1939

A SAE(Society of Automotive Engineers) criou a norma SAE J1939, um protocolo indicado para a comunicação e diagnóstico entre os componentes do veículo(motor, transmissão). É baseada na CAN 2.0, define como a informação é passada da rede para as unidades de controle eletrônico (ECU's), sendo amplamente usada no setor automotivo (principalmente em ônibus e caminhões).

Atende distâncias de até 40 metros, para tanto faz uso do cabo de par trançado blindado³ com velocidade de até 250kbps. Em relação ao CAN, é mais confiável, utiliza níveis de prioridade para definir quais aplicações são mais importantes(críticas).

³ O cabo coaxial blindado ou par trançado blindado é constituído por um núcleo de cobre sólido cercado por um isolante, uma blindagem de malha metálica e uma cobertura externa[13].

O J1939 trabalha com cinco das sete camadas do modelo de referência OSI⁴: física, enlace, rede, transporte e aplicação.

VI. REDES AUTOMOTIVAS CLASSE C

As redes classe C são utilizadas em aplicações críticas, com requisitos de tempo real, alta frequência, tolerância a falhas e outros mecanismos essenciais para segurança crítica. Em veículos com uma arquitetura distribuída com rede Classe C, existem ECUs para realizar funções críticas como, por exemplo, sistemas de frenagem, nestes sistemas as velocidades são superiores a 125kbps.

A. FlexRay

Com o objetivo de facilitar a comunicação dos módulos eletrônicos de uma maneira simples a BMW criou o Byteflight.

O Byteflight é orientado a conexão, ou seja, é estabelecida uma conexão, os dados são transmitidos e então é corta a conexão. Utiliza a FTDMA como controle de acesso ao barramento, onde um nó mestre é responsável pelo clock, além disso, as mensagens são enviadas de acordo com uma ordem de preferência.

Tem taxas de transmissão de até 10 Mbps, e sua aplicação mais conhecida é no sistema de airbag.

B. TTP/C

É um protocolo de comunicação projetado especificamente para aplicações automotivas relacionadas com a segurança. Neste protocolo quando uma ECU falha, é possível que outras ECU's assumam o papel da ECU que falhou. Aplicações como airbag, abs e injeção eletrônica podem utilizar este protocolo.

O cabo padrão utilizado é o par trançado, onde as taxas de transmissão variam até 25 Mbps. A técnica de correção de erro é a CRC, em que é gerado um polinômio, o valor deste polinômio é enviado ao receptor e ao fim é recalculado o valor.

B. FlexRay

Analisando a necessidade de evolução das redes de comunicação automotivas a BMW uniu-se a Daimler-Chrysler e criou o consórcio FlexRay. O principal objetivo deste novo protocolo é atender aplicações X-by-wire⁵, ou seja, aplicações extremamente críticas para o funcionamento do veículo como freio, direção, suspensão e motor[1][16].

O protocolo FlexRay é uma combinação do protocolo Byteflight da BMW e do protocolo TTP da Universidade Técnica de Vienna. O Byteflight foi originalmente projetado

para sistemas de segurança passivos em aplicações como acionamento de *airbag*, onde curtos tempos de resposta são necessários.

O FlexRay utiliza dois canais de comunicação para reduzir o risco de falhas, cada um destes canais pode opera em velocidades de até 10Mbps. A transmissão de dados ocorre por meio da diferença de tensão aplica no cabo, para reduzir a interferência (ruído) de uma linha na outra os cabos são trançados.

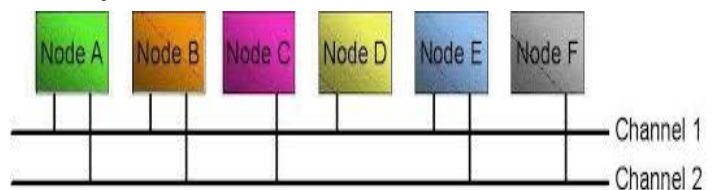


Fig. 7 – Barramento Duplo de uma rede FlexRay[17].

As topologias possíveis são: barramento, estrela e híbrido. Na topologia de barramento, cada uma das ECUs pode ser ligada diretamente a um ou a dois canais. Numa topologia em forma de estrela, os pontos centrais de ligação são denominados acopladores de estrela, e caso seja usada esta topologia é possível construir redes em cascata ligando assim dois acopladores de estrela diretamente. Por último, a topologia híbrida, resulta de uma combinação de uma topologia de barramento e uma topologia estrela[1].

Utiliza TDMA(Time Division Multiple Access) que divide o meio (canal) em intervalos de tempo, sendo assim, durante este intervalo o nó poderá usar toda a banda disponível. Para melhorar o desempenho, o FlexRay também utiliza um sistema de preferência para dividir o acesso ao meio, o que evita que a banda seja desperdiçada. Esta preferência está ligada a importância da informação a ser transmitida, ou seja, no projeto é especificado qual módulo (nós) trabalha com o setor mais crítico[2].

Basicamente, a estrutura do FlexRay é formada de um controlador principal (host), um controlador de comunicação (communication controller) e um controlador de barramento (bus guardian). O host controla os componentes dos nós e processa os dados, o controlador de comunicação cuida das informações recebidas e o controlador de barramento controla o acesso ao barramento.

As vantagens de utilizar uma rede FlexRay são: segurança, tolerância a erro, alta taxa de transmissão de dados e arquitetura simplificada. Além disso, através do FlexRay é possível substituir alguns componentes mecânicos e hidráulicos por componentes eletrônicos, o que facilita o projeto e reduz os custos e a manutenção. Por tais motivos este tipo de rede é considerado o futuro tratando-se de redes de comunicação automotivas.

VII. SISTEMAS X-BY-WIRE

Em sinopse, um sistema x-by-wire é aquele no qual os componentes mecânico/hidráulicos são substituídos por componentes eletrônicos conectados a um módulo que processa os comandos.

Historicamente o desenvolvimento destes sistemas teve início na aeronáutica com o fly-by-wire, nos aviões os sistemas mecânicos e hidráulicos representavam muito mais

⁴ A ISO lançou o modelo de referência OSI em 1984. Trata-se de uma descrição ou modelo de referência e não um novo padrão estabelecendo o modo como a informação deve ser transmitida entre os pontos de uma rede. Esse modelo propõe uma estrutura em camadas para o desenvolvimento de soluções que funcionem sob qualquer plataforma, independentemente do hardware ou software utilizado[14].

⁵ O X-by-wire consiste em substituir um comando essencialmente mecânico por um eletromecânico, que é repassado para um computador que executa a tarefa solicitada[15].

peso, o que diminuía a autonomia do voo e aumentava a complexidade do projeto devido ao grande número de fios e cabos.

O sistema de vôo fly-by-wire foi utilizado pela primeira vez no caça F-16 Fighting Falcon, que entrou oficialmente em operação nos EUA em 1979. Em jatos comerciais, o sistema foi implantado pela primeira vez no Air-bus A320, que entrou em serviço no ano de 1988 e é construído por um consórcio que engloba empresas da França, Inglaterra, Espanha e Alemanha. Nesses aviões, o fly-by-wire exclui da cabine de controle o manche convencional, que é operado pelo piloto de forma semelhante a um volante de carro, com as duas mãos, e o substitui por uma alavanca parecida com a que controla os caças - que lembra mais um joystick de video-game e está situada ao lado do piloto[15].

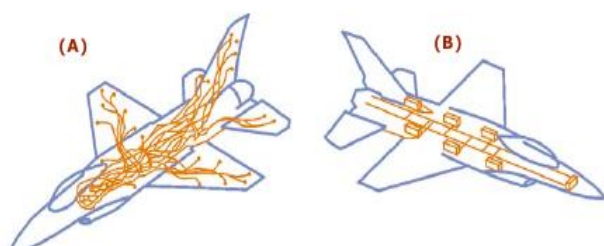


Fig. 8 – Comparativo entre os sistemas comuns que utilizam cabos e conexões ponto a ponto (A) e os sistemas que utilizam barramento e protocolos de comunicação (B)[18].

Observando a figura 8 fica claro que sistemas X-by-wire têm diversas vantagens sobre sistemas comuns, percebendo esta tendência o mercado automotivo também começou o desenvolvimento de sistemas similares com objetivo de aumentar conforto e comodidade, aliando a isto redução nos custos e aumento na eficiência energética (economia de combustível).

As primeiras aplicações by-wire automotivas tiveram início na Fórmula 1 com os sistemas brake-by-wire e drive-by-wire.

O brake-by-wire é o sistema de frenagem eletrônica, ou seja, de acordo com o quanto o piloto pisa no pedal o sistema envia comandos a um atuador.

O drive-by-wire é conhecido como acelerador eletrônico, e substitui o sistema antigo que utilizava um cabo ligado ao pedal para definir o quanto a válvula-borboleta abria e conseqüentemente o quanto de combustível era injetado no motor. Diferente deste modelo, o sistema drive-by-wire utiliza um potenciômetro, que envia sinais para um módulo que define a aceleração de acordo com o quanto o piloto pisa no pedal.

Ambos os sistemas trouxeram diversos benefícios como redução da manutenção, economia de combustível e até mesmo uma melhor dirigibilidade.

VII. CONCLUSÃO

A indústria automobilística busca na padronização de modelos e protocolos de comunicação uma forma de otimizar a comunicação entre os dispositivos eletrônicos. Tais protocolos apresentam diversas vantagens como redução do espaço ocupado, aumento da inteligência associada e redução do consumo, porém é preciso diferenciar quais as

aplicações serão atendidas para escolher de maneira mais adequada possível o protocolo a ser usado, evitando assim falhas ou erros de projeto.

A divisão destes protocolos nas categorias A, B e C de acordo com a finalidade facilita a padronização e o desenvolvimento. Em cada uma destas categorias existem diversos protocolos, e cada um destes apresenta vantagens diferentes. Porém, o padrão mais comum de ser encontrado é o CAN, devido a sua simplicidade e interoperabilidade.

A perspectiva é que a utilização das redes de comunicação automotivas apenas aumente, visto que cada vez mais os protocolos atendem de forma eficiente e segura as aplicações, exemplo disso são os sistemas X-by-wire, as quais executam ações extremamente críticas, que devem dominar o mercado em alguns anos.

REFERÊNCIAS

- [1] Rezende, Daniel C. Felisberto (17 de Julho de 2009) Proposta de especificação formal em SDL de uma rede de comunicação automotiva baseada no protocolo FlexRay com geração automática de código Java. Departamento de Telemática. Unicamp. Disponível em: <https://www.google.com.br/url?sa=t&rcet=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKewjk-7S6ndHNhAhXH2R4KHS2mDv8QFggeMAA&url=http%3A%2F%2Flibdigi.unicamp.br%2Fdocument%2F%3Fcode%3D000467943&usq=AFQjCNEQLhcNYwYtheSgx8OMjmlLkqldkA&sig2=jNSMaTh0t6IyDINfyvtOkg>.
- [2] JUNIOR, Helio Taliani (2012) Estudo dos protocolos de comunicação das arquiteturas eletroeletrônicas automotivas, com foco nas suas características e respectivas aplicações, visando o direcionamento para o uso adequado e customizado em cada categoria de veículo. Instituto Mauá de Tecnologia. Disponível em: <https://www.google.com.br/url?sa=t&rcet=j&q=&esrc=s&source=web&cd=9&cad=rja&uact=8&ved=0ahUKewi7ain9NAhVMKh4KHUX3CKEQFgg0MAg&url=http%3A%2F%2F>.
- [3] Instituto Newton C. Braga LTDA (2016,25 Junho). Como funcionam os Sistemas Eletrônicos do Automóvel (ART567) [Online]. Disponível em: <http://www.newtonbraga.com.br/index.php/como-funciona/4142-art567>.
- [4] Rodrigo V. M. Pereira, Cesar A. Zeferino (17 de Agosto de 2008). Aplicação do protocolo LIN como sub-rede can. UNIVALI -Campus São José – Grupo de Sistemas Embarcados e Distribuídos. Disponível em: https://www.researchgate.net/publication/267372136_APLICACAO_DO_PROTOCOLO_LIN_COMO_SUBREDE_CAN
- [5] Saber Eletrônica (2016,25 Junho). Tecnologias de Redes de Comunicação para Sistemas Automotivo[Online]. Disponível em: <http://www.sabereletronica.com.br/artigos/1733-tecnologias-de-redes-de-comunicacao-para-sistemas-automotivo>
- [6] Whatsl (2016, 25 Junho). UART (Universal Asynchronous Receiver/Transmitter)
- [7] Electric Imp(2016, 25 Junho). UART Explained. Introduction to the standard serial bus. Disponível em: <https://electricimp.com/docs/resources/uart/>
- [8] Embarcados (2016, 25 Junho). Comunicação SPI – Parte 1. Disponível em: <http://www.embarcados.com.br/spi-parte1/>
- [9] Wikipedia (2016, 25 Junho). Serial Peripheral Interface Bus. Disponível em: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
- [10] Rômulo Calado Pantaleão Camara. Eletrônica Digital. Universidade Federal do Vale do São Francisco - Colegiado de Engenharia da Computação. Disponível em: <http://www.univasf.edu.br/~romulo.camara/novo/wpcontent/uploads/2013/11/Protocolo-LIN.pdf>
- [11] MZeditora(2016, 27 Junho). Introdução às Redes Intraveiculares. Disponível em: <http://www.mzeditora.com.br/artigos/veicular.htm>

[12] Alexag(2016, 27 Junho). Can bus barramento controller area network Conceituação. Disponível em: http://www.alexag.com.br/CAN_Bus_Parte_2.html

[13] ICEI. PUC- Minas(2016, 27 Junho). Cabeamento de Rede. Disponível em: <http://www.icei.pucminas.br/professores/marco/cabeamento-de-rede/>

[14] Projeto de Redes(2016, 27 Junho). Disponível em: http://www.projotoderedes.com.br/artigos/artigo_osi_um_modelo_de_referencia.php

[15] Mecatrônica Atual (2016, 27 Junho). X-by-o quê?. Disponível em: <http://www.mecatronicaatual.com.br/educacao/1280-x-by-o-que>.

[17] Bratti Diogo, Augusto A. Fröhlich, Tondo D. Souza (Novembro de 2008). Sistemas Automotivos Embarcados. Disponível: <http://www.lisha.ufsc.br/teaching/os/ine5412-2008-2/work/auto.pdf>

[18] Mecatrônica Atual(2016, 27 Junho). Comunicação Aérea. Disponível em <http://www.mecatronicaatual.com.br/educacao/1829eletcnica-embarcada-e-protocolos-de-comunicacao-na-aviao>

CAN – Control Area Network

Uma breve apresentação das características do protocolo

André Luiz de Sousa Marcondes Reuter

Engenharia de Controle e Automação
UFSC – Universidade Federal de Santa Catarina
Blumenau, Brasil
a.reuter@grad.ufsc.br

Abstract- The CAN (Controller Area Network), was initially created for the automotive sector, it has high flexibility and is capable of work and inter operate with multiple industrial protocols. Supporting real-time applications and higher data security, this protocol is widely used in industrial automation and control of critical systems. This protocol has high economic efficiency and it is very configurable and can work with almost every net equipment needing only a CAN controller and a transceiver. This article presents a brief vision of the structure of the CAN protocol as well as the specifications and details about the messages transmission on the Standard CAN.

Keywords – CAN, industrial network, arbitration, ISO11898.

Resumo— A CAN (Controller Area Network), foi criada inicialmente para o setor automobilístico, tem alta flexibilidade e é capaz de trabalhar e interoperar com vários protocolos industriais. Suportando aplicações em tempo real e com superior segurança de dados, este protocolo é amplamente utilizado em automação industrial e controle de sistemas críticos. O protocolo tem alta eficiência econômica, permite diferentes tipos de configuração e pode trabalhar com quase todos os equipamentos de rede precisando apenas de um controlador CAN e um transceiver. Este artigo apresenta uma breve visão da estrutura do protocolo CAN, bem como as especificações e detalhes sobre a transmissão de mensagens na Standard CAN.

Palavras-chave—CAN, rede industrial, arbitragem, ISO11898.

I. INTRODUÇÃO

A rede CAN (Control Area Network) foi oficialmente publicada em fevereiro de 1986 pela Robert Bosch GmbH e desde então vem sendo utilizada principalmente pela indústria automotiva. Em novembro de 1993 a especificação versão 2.0 da CAN foi padronizada pela ISO 11519-2 que também padronizava uma camada física para taxa de bits de até 1Mbps, no entanto a norma não chegou a ser implementada devido a falhas nos termos de padronização. Isso foi resolvido com um adendo à norma descrevendo o formato estendido da CAN usando um identificador de 29 bits, então em 1995 a ISO 11898 normalizou as redes CAN [1].

A CAN, de acordo com a referência da BOSCH, é um protocolo de comunicação serial que suporta aplicações em tempo real com alto nível de segurança. A área de aplicação varia desde redes para multiplexação de cabos como no caso das redes automotivas, até redes de alta velocidade e controle

de processos em automação industrial [2]. As mensagens possuem tamanho reduzido e alta confiabilidade contando com detecção e correção de erros tornando ideal para comunicação com sensores e atuadores em campo. Em uma rede CAN as mensagens são transmitidas em broadcast o que garante a consistência dos dados em todos os equipamentos. A rede trabalha com um bit de identificador, ou identifier, que provê a ação de prioridade na mensagem e organização da rede. Além disso é um sistema multimaster, ou seja, qualquer nodo da rede pode ser o mestre tornando os outros nodos escravos de acordo com o bit identificador na mensagem que está sendo enviada. As redes CAN funcionam fisicamente em barramento, porém logicamente trabalham como no modelo token-ring [2][3].

Os identifiers também dividem a CAN em dois grandes sistemas: o Standard CAN com identificador de 11 bits e o Extended CAN, com identificador de 29 bits [2]. Este artigo tratará sobre o modelo padrão da CAN abordando a estrutura da rede em comparação com o modelo ISO OSI (Open Systems Interconnection) mostrando detalhadamente o quadro de mensagens da CAN além das funcionalidades e áreas de implementação.

II. PROPRIEDADES ESPECIFICAS DAS REDES CAN

O padrão CAN além de ser multimaster e possuir comunicação broadcast entre o mestre e os escravos possui as seguintes propriedades [2]:

- Priorização das mensagens;
- Garantia de tempos de latência;
- Múltiplos tipos de configuração;
- Consistência de dados por todo o sistema;
- Detecção e correção de erros;
- Retransmissão automática de mensagens corrompidas;
- Distinção entre erros temporários e falhas permanentes, desligando autonomamente nodos comprometidos.

É importante destacar que estas características estão presentes apenas nas revisões mais recentes da CAN (ISO 11898). Este tipo de rede foi desenvolvido para substituir a enorme quantidade de cabos em veículos por uma comunicação em barramento com apenas dois fios, por isso, aplicações em tempo real foram sempre um objetivo do padrão. No entanto, alguns aspectos do padrão são atrativos para

comunicação em ambiente industrial, principalmente no que diz respeito a latência e tamanho reduzido das mensagens. Assim, as redes CAN ganharam popularidade em ambiente industrial [1]. Além disso a rede procura por meios físicos que sejam, da melhor forma possível, imunes a ruído eletromagnético (como a fibra ótica). As redes CAN por serem barramentos de dois cabos e possuírem pequenas distâncias entre o barramento e as estações, são leves e mais baratas, facilitando a instalação e o projeto de layout [4].

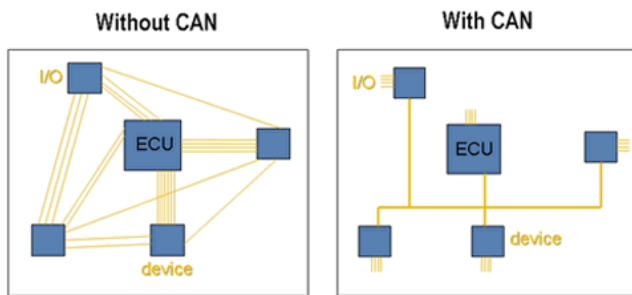


Fig. 1. Comparação entre uma rede sem CAN e com CAN, mostrando a maior quantidade de fios [4].

O padrão CAN segue o modelo de camadas do padrão OSI usando as camadas 1, 2, as funções das camadas 3 a 5 são absorvidas pela camada de enlace de dados, a camada 7 é apenas uma interface para protocolos de mais alto nível [2]. A norma ISO 11898 descreve como as mensagens são passadas de camada a camada. Na Fig. 2. temos um quadro comparando as camadas de rede CAN com o modelo de referência OSI.

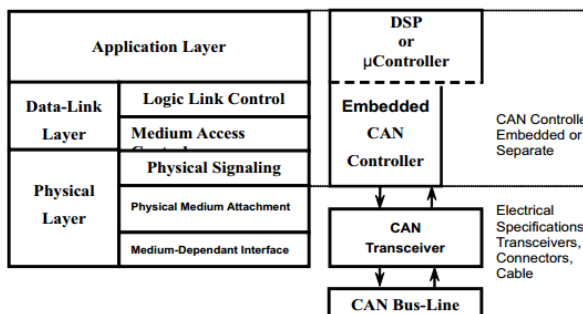


Fig. 2. Comparação entre o modelo de referência OSI e as camadas da rede CAN [3].

As camadas da rede CAN também são conhecidas da seguinte forma: camada de objeto, camada de transferência e camada física. Cada camada possui um conjunto de funções específicas [2]. Como o objetivo das redes industriais é oferecer melhor tempo de resposta (até mesmo aplicações em tempo real como no caso da CAN) e maior confiança nos dados transmitidos, reduzir o número de camadas faz com que todo o processo de comunicação seja mais rápido e com que menos headers sejam adicionados, além da probabilidade de erro de mensagem ser reduzida devido a menor quantidade de “burocracia” envolvida.

III. A CAMADA FÍSICA

As camadas de enlace e física do modelo de referência OSI compõe a camada física das redes CAN. Os parâmetros e conexões, normalmente transparentes ao usuário, estão embutidos em um controlador que implemente a rede CAN. Para conexão com o meio físico é necessário um transceiver. É normalmente usado para trançado ou fibra ótica como meio de comunicação [3][5].

Nas redes CAN o sinal é diferenciado. As duas linhas de sinal são definidas como CANL e CANH, onde um 0 corresponde ao sinal CANH, superior ao sinal CANL, que é denominado bit dominante. De forma contrária um bit 1 corresponde ao sinal CANL, superior ao CAN_H, denominado bit recessivo, daí o nome diferenciado, as denominações dos bits são dadas pelas diferenças entre os sinais das duas linhas. Na Fig. 3. vemos os níveis de sinal das duas linhas da rede CAN [6]. Então quando tivermos o barramento como recessivo, os nodos o enxergam com um bit 1 e vice-versa. Além disso, o sinal é balanceado o que reduz o ruído, bem como a chance de erro de bit e torna as redes CAN robustas para entregar uma alta taxa de transmissão com alta taxa de confiabilidade. A aplicação do sinal diferenciado nas redes CAN confere a capacidade de funcionar mesmo com estações defeituosas ou extremo ruído na linha [3][5].

A camada física é dividida entre a parte de codificação e decodificação que é implementada nos controladores e a conexão com o meio físico que define as especificações dos transceivers [5].

Na Fig. 4. vemos que na verdade a parte de codificação e decodificação em si é dividida em duas subcamadas. A AUI (unidade de interface de fixação) é responsável pela interface com os chips dos transceivers e prove os pinos TxD e RxD. Já a MDI (interface dependente do meio) efetivamente conecta o transceiver ao barramento [5]. Ou seja, a AUI está entre o controlador e o transceiver e a MDI está entre o transceiver e o barramento.

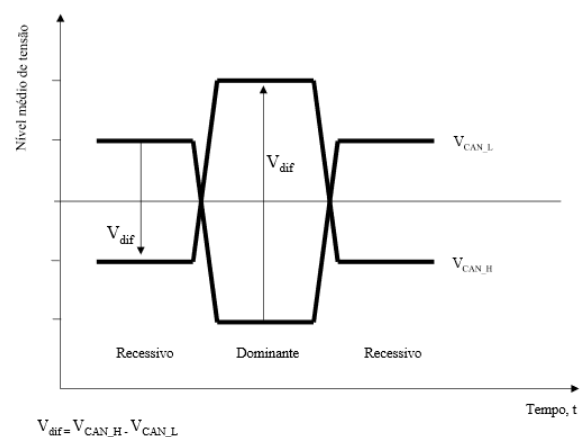


Fig. 3. Nível de sinal das linhas de sinal CANH e CANL [6].

Nas redes CAN é usado codificação NRZ (non return to zero – não retorno a zero), ou seja, os bits não são representados por bordas de descida e subida como ocorre na codificação Manchester. Há duas implementações, para 5V ou 3.3V, os níveis de sinal são constantes durante sua duração sendo dominantes ou recessivos, fazendo com que a largura de banda de transmissão seja melhor aproveitada em relação a outros tipos de codificação. Isto só é possível devido ao sinal balanceado e diferenciado, de outra maneira, se tivéssemos em um barramento um nível constante durante longo tempo isso poderia acarretar em erros de recepção e sincronização [3][5][6].

O tempo nominal do bit é dado pelo inverso da taxa de transmissão de um emissor ideal na ausência de resincronização. As redes CAN possuem tanto hard synchronization quanto soft synchronization. Se o barramento está livre, ou seja, no estado recessivo por mais de três tempos de bit, a primeira borda de descida detectada é usada para sincronização de todos os nodos (hard synchronization), a sincronização local usa as bordas de descida seguintes (soft synchronization) [5][6].

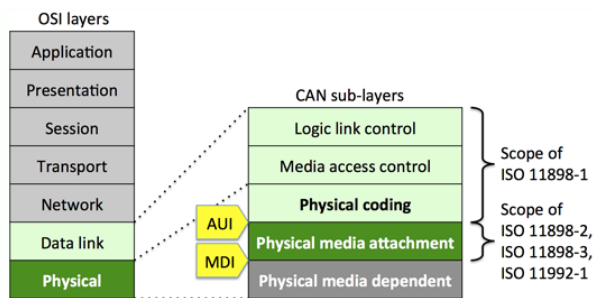


Fig. 4. Comparação entre o modelo de referência OSI e a camada física das redes CAN [5].

As redes CAN estão definidas na ISO 11898 para taxas de até 1Mbps para até 40 m de cabo com um máximo de 30 nodos, a norma também recomenda uma distância máxima entre o nodo e o barramento de 0.3 m. A topologia recomendada pela norma é apresentada na Fig. 5.. As linhas devem possuir link de par trançado blindado ou não blindado de 120-Ω, os resistores em cada fim das linhas também possuem 120-Ω prevenindo a reflexão do sinal [3]. Atualmente também existem barramentos CAN usando fibra ótica para aplicações críticas que estão sujeitas a altos níveis de ruído eletromagnético e possuem severas exigências quanto a baixa latência e integridade dos dados.

Em uma rede CAN todos os nodos se comunicam e podem enviar mensagens quando o barramento estiver livre podendo haver ou não um nodo de controle. A adição de novos nodos pode ser feita inclusive enquanto a rede estiver em operação. Os drivers e transceivers CAN possuem internamente um pull up, então mesmo não havendo entradas, o barramento está sempre recessivo [3]. Isso previne alguns erros no caso do barramento estar em idle por muito tempo e também impede que, por algum ruído ou erro em algum nodo, o barramento pareça ocupado para as estações.

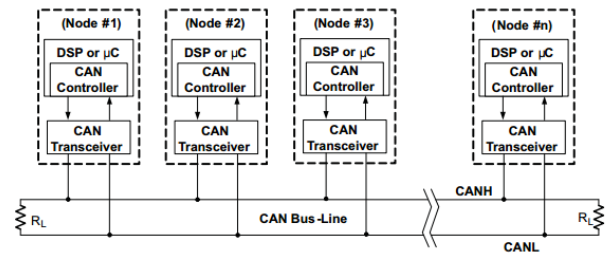


Fig. 5. Topologia de uma rede CAN e posição dos equipamentos de rede [3].

IV. A CAMADA DE TRANSFERÊNCIA

Esta camada é responsável pelas seguintes funções: detecção e correção de erros (validação das mensagens e reconhecimento), definição de prioridade (arbitragem), enquadramento da mensagem, taxa de transferência e tempo [2]. O protocolo de comunicação CAN é um CSMA/CD + NBA (Carrier Sense Multiple Access with Collision Detection + Nondestructive Bit wise Arbitration), ou seja, possui reconhecimento da portadora, comunicação multiponto com detecção de colisão e arbitragem na prioridade das mensagens. As estações podem requisitar ou enviar mensagens a qualquer tempo. As redes CAN possuem a propriedade única de serem capazes de detectar todos os erros de bit único [3].

A. Detecção e correção de erros

Para detecção a rede é monitorada, ou seja, os transmissores comparam o nível do sinal transmitido com o sinal recebido no barramento. Além disso temos as técnicas tradicionais como o CRC (Cyclic Redundancy Check), bit stuffing e checagem do quadro de mensagem. Estes mecanismos de detecção de erros garantem que a probabilidade de uma mensagem com erro não ser identificada seja menor que $4.7 \cdot 10^{-11}$. Nas redes CAN qualquer nodo pode identificar a mensagem corrompida, se um nodo identifica um erro a mensagem é marcada com uma flag e sua comunicação é abortada e será retransmitida automaticamente. O tempo de recuperação entre detecção do erro e retransmissão é de 29 tempos de bit se não houver nenhum erro a mais. A rede CAN ainda é capaz de detectar falhas constantes e desativar nodos defeituosos autonomamente. Todos os nodos que recebem uma mensagem checam a consistência e reconhecem uma mensagem correta assim como as incorretas as marcando com uma flag [2][3][7].

B. Arbitragem

Como o protocolo CAN é multimaster, é possível que mais de uma estação queira transmitir ao mesmo tempo. O protocolo possui um sistema de detecção de colisão que é resolvida com base na prioridade. Os bits de identificação servem para definição de prioridade. Quanto mais baixo for o valor binário no campo identificador da mensagem, mais alta é a prioridade. A prioridade mais alta sempre vence para acesso ao barramento, esse protocolo se assemelha muito com o forcing headers ou contagem regressiva de bits, no entanto aqui temos uma arbitragem orientada a bit e um funcionamento mais simples [2][3][7][9].

Isso é uma grande vantagem para sistemas em que determinadas informações possuem maior relevância em relação a outras. Por exemplo, um sistema que possua um sensor de nível em um tanque que aciona um alarme de nível alto. A informação para acionamento do alarme é mais importante do que as informações de outros sensores e deve ser transmitida imediatamente após a ativação do alarme, para isso basta que o identificador da mensagem enviada pelo nodo do sensor de nível alto seja 0, assim a mensagem terá a prioridade mais alta.

Todas as estações conectadas observam o barramento bit a bit. Como dito anteriormente, o barramento possui dois níveis: dominante e recessivo. Quando uma estação está transmitindo em nível recessivo e o barramento se encontra em nível dominante, essa estação perde o direito de transmitir e muda para o modo ouvinte ou receptor. Isso faz parte do processo de solução de colisão por comparação orientada a bit, ou bit-wise. Após ter o acesso ao barramento negado, a estação pode retransmitir a mensagem quando o barramento estiver livre novamente [3][6][7]. Além disso a estação que transmite com alta prioridade deve esperar por um bit 0 no próximo espaço interframe para transmitir novamente, isso impede que uma estação domine o barramento.

C. Enquadramento da mensagem (framing)

Na Fig. 6. temos a estrutura do quadro de mensagem das redes CAN.



Fig. 6. Quadro de mensagem da rede CAN [3].

SOF – É o bit de início (start of message) usado para sincronização dos nodos. Consiste de um único bit dominante. Uma estação só pode transmitir se o barramento estiver em modo de repouso (idle) e todos os nodos devem ser sincronizados para a comunicação iniciar [3].

Identificador – Este campo compreende os 11 bits de identificação (no caso da Standard CAN) e o bit RTR (Remote Transmission Request). O campo de identificador também diferencia a CAN em Standard CAN (identificador de 11 bits) e Extended CAN (identificador de 29 bits) [2][3].

IDE – Bit de extensão do campo identificador significando que o identificador é da rede Standard CAN e que nenhuma extensão está sendo transmitida [3].

RTR – Requisição de transmissão remota (Remote Transmission Request) é um bit dominante que pode ser usado para requisição do envio de uma mensagem remotamente. Para isso o bit não pode ser recessivo [3].

r0 – Bit extra para implementações futuras [3].

DLC – Código de tamanho de dado (Data Length Code). Campo de quatro bits para indicar o tamanho em bytes da mensagem a ser transferida [3].

Data – Campo que efetivamente contém a mensagem a ser transmitida. A mensagem pode conter de 0 a 8 bytes [3].

CRC – Este campo contém a sequência CRC e é seguido por um delimitador. O polinômio usado no cálculo de detecção dos erros é o que segue em (1) [2]:

$$x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1 \quad (1)$$

ACK – Este campo possui dois bits que são usados pelos nodos que recebem as mensagens. Quando um nodo recebe uma mensagem válida corretamente, é enviado uma confirmação (ACK). A mensagem de confirmação é enviada somente após a verificação do CRC, se estiver correto então a confirmação é enviada. Se a mensagem estiver corrompida ou for incorreta nada é enviado e a mensagem é retransmitida automaticamente pela estação emissora quando o barramento estiver livre novamente [3].

EOF – Fim do quadro (End of Frame). Este campo de sete bits marca o fim do quadro de uma mensagem CAN. O bit stuffing funciona para uma sequência de cinco bits 0, quando detecta uma sequência como essa é inserido um bit 1 [3].

IFS – Espaço interquadro (Inter frame Space). Campo de sete bits contendo o tempo correto para o controlador mover o quadro para sua posição específica no buffer [3].

O bit stuffing funciona para todos os campos do quadro de mensagem, bem como o CRC, isso faz com que a detecção de erros seja mais precisa e, como veremos a seguir, permite uma forma eficiente de compor um quadro de erros e assim corrigi-los. [2][3].

V. OUTROS TIPOS QUADROS

Existem mensagens especiais que servem para organização da rede e possuem estruturas diferentes do quadro de dados mostrado anteriormente. Esses quadros são o quadro remoto, o quadro de erro e o quadro de sobrecarga.

A. Quadro remoto

Como visto anteriormente o quadro de mensagem possui um bit para requisição de mensagem remota (RTR). Quando esse bit é dominante no quadro um nodo agindo como receptor requisita a mensagem de um outro nodo. Todas as estações recebem a requisição mas o identificador especifica qual nodo deve enviar. Como a comunicação é em broadcast todas as estações também recebem a resposta do nodo que requisitou a mensagem e podem usar os dados se houver interesse. Assim os dados são uniformes em toda a rede. No quadro remoto não há campo de dados e o bit RTR deve ser recessivo [2][3].

B. Quadro de erro

Nas redes CAN cada nodo pode identificar um erro em uma mensagem e enviar essa informação para todas as outras. Então a estação que enviou a mensagem original retransmite automaticamente quando o barramento estiver em idle. Um sistema nos controladores previne que um nodo envie repetidamente mensagens de erro armazenando o número de mensagens em um contador [2][7].

Este quadro possui dois campos: o primeiro é a superposição de flags de erro enviadas por diferentes estações, o segundo é um delimitador do quadro. A flag formada pelo quadro de flags pode ser passiva ou ativa. Quando um nodo

ativo ao erro detecta um erro, este quebra as regras da rede quanto a bit stuffing e isso força todas as estações a enviar um quadro de erro. Cada estação detectará um erro diferente de acordo com a mensagem que receber, assim pela superposição de cada mensagem de erro temos o tamanho total do campo de flag (que pode variar de seis a doze bits). Quando uma estação passiva ao erro detecta um erro esta espera por seis bits de mesmo nível chegarem consecutivamente então envia uma flag de erro passiva [2][6][7].

O campo delimitador possui oito bits recessivos. Após o envio de uma flag de erro cada estação envia um bit recessivo e espera pelo barramento até este detectar o bit recessivo, então envia mais sete bits recessivos [2]. Desta forma as estações sabem que o quadro de erro terminou e que podem reiniciar a transmissão de mensagens pois o barramento está em idle.

C. Quadro de sobrecarga

O quadro de sobrecarga é enviado por estações que estão recebendo mensagens além da capacidade de processamento. Essencialmente possui uma estrutura similar ao quadro de erros e força com que o delay entre o envio de novas mensagens para a estação ocupada seja maior. Na verdade o flag enviado pela estação é idêntico ao flag de erro ativo. Existem duas condições para uma estação enviar uma flag de sobrecarga: a) se o primeiro ou segundo bit de ISF for dominante e b) se o sétimo bit de EOF for dominante. Isso ocorre devido a condição da rede, como os quadros são construídos e como os erros são detectados. Se o último bit do campo EOF for dominante, a estação que está enviando a mensagem interpreta o quadro como um erro e retransmite fazendo com que o nodo receptor receba a mensagem duas vezes, pois a mensagem já havia sido aceita por conta dos seis bits anteriores. O envio de uma flag de sobrecarga não altera nenhum aspecto na rede, não incrementa o contador de erros, não exige retransmissão e não faz com que outras estações enviem flags de erros [2][3][7].

D. O intervalo interframe

Como visto anteriormente o intervalo interframe é um campo no frame de mensagem e não uma mensagem em si. Porém este campo é determinante na detecção dos erros pelas estações, sejam de sobrecarga, erros ativos ou passivos. O IFS contem os campos de intermissão, idle e suspensão de transmissão. O campo de intermissão possui três bits recessivos que enquanto estão sendo recebidos não pode haver transmissão de frame de dados ou frame remoto apenas de sobrecarga. Isso garante que a mensagem transmitida seja recebida antes da estação receptora tentar repassá-la. O campo de idle possui tamanho arbitrário e quando possui valor dominante é interpretado como início de um novo frame. A suspensão de transmissão é identificada quando uma estação passiva ao erro, após enviar um frame, envia oito bits recessivos antes de identificar condição de idle. Isso significa que nenhum outro nó pode enviar mensagens durante um tempo específico [2][3][5].

VI. A CAMADA DE APLICAÇÃO

Nas redes CAN os protocolos de alto nível ou de acordo com a referência OSI, os protocolos de aplicação, são

implementados particularmente, isto é, não está no escopo da norma ISO 11898 uma camada de aplicação para as redes CAN. Nas diretivas do modelo CAN a camada de aplicação esta diretamente ligada com as camadas inferiores de rede e enlace. O designer da rede CAN precisa adicionar as capacidades de aplicação se for necessário para o sistema que está sendo implementado. Aplicações comuns são gerenciamento da rede, monitoramento de nodos, etc [8].

A camada de aplicação serve como interface de interoperação com outras redes industriais como a DeviceNet que implementa recursos como segmentação e reagrupamento de dados para camadas mais altas, apesar dessa característica já estar presente na camada de enlace da rede CAN. Essencialmente, os protocolos de alto nível são implementações externas [8].

VII. A REDE CAN E O PANORAMA INDUSTRIAL

Como visto o protocolo CAN é um das soluções em redes industriais mais versáteis e robustas já implementadas e desde sua criação, sua popularidade em ambiente industrial só cresceu. Com o desenvolvimento de tecnologias na área dos semicondutores, a tendência é que os equipamentos das redes CAN se tornem mais rápidos e possam ser mais inteligentes fazendo com que aplicações com exigências de latência e jitter muito restritos, sejam uma realidade para esse protocolo. Um dos gargalos de hardware atualmente para as redes CAN são os transceivers que fazem a interface direta entre os controladores e a conexão física com os barramentos. O desenvolvimento nessa área é crucial para expansão das redes CAN em direção a aplicações em controles de movimento, robótica industrial e aplicações em tempo real [3].

Apesar de a CAN ser uma das redes mais imunes a erros de bit e erros de mensagem é necessário uma adaptação nas condições de sobrecarga das estações, muitas vezes a própria condição de transmissão de mensagem provoca uma sobrecarga aparente, ou seja, sem haver sobrecarga propriamente dita. Isso poderia ser resolvido transformando dados relativos em dados absolutos referenciados a um contador, por exemplo [7]. No geral, a CAN é um padrão consolidado na indústria que se adapta facilmente as novas necessidades dos setores que esta inserida, o próximo passo seria a expansão da interoperabilidade com redes mais modernas como a ProfiNet.

VIII. CONCLUSÃO

O protocolo CAN se mostra como uma boa opção para uma gama de aplicações desde sua origem em automóveis até as mais recentes aplicações industriais, com a configurabilidade e interoperabilidade necessárias para o dinamismo que há no ambiente industrial atualmente. O que torna a CAN ideal para automação industrial é que foi um protocolo especificamente desenvolvido para aplicações que necessitam de mensagens curtas, alta confiabilidade e baixa latência, além disso oferece grande resistência ao ambiente hostil encontrado no chão de fábrica.

Esse tipo de rede possui atributos essenciais para aplicações de alto risco, pois consegue alta confiabilidade na

transmissão de mensagens detectando e corrigindo os erros de maneira a não comprometer a velocidade de transmissão. Além da capacidade de excluir estações defeituosas autonomamente.

Em vários aspectos há pontos de melhoria, mas para um protocolo relativamente antigo, ainda é possível implementá-lo sem quase nenhuma modificação em muitos processos industriais fazendo com que a CAN persista e se modernize.

REFERÊNCIAS

- [1] History of de CAN technology. Disponível em: <http://www.can-cia.org/can-knowledge/can/can-history/>
- [2] ROBERT BOSCH GmbH, CAN Especificação, Versão 2.0 1991, Stuttgart, Alemanha.
- [3] Corrigan, Steve, "Introduction to the Controller Area Network (CAN)", Relatório de Aplicação Texas Instruments, p. 2-8, Agosto de 2002.
- [4] Controller Area Network (CAN) Overview, Disponível em: <http://www.ni.com/white-paper/2732/en/>
- [5] CAN Physical Layer, Disponível em: <http://www.can-cia.org/can-knowledge/can/systemdesign-can-physicallayer/>
- [6] Projeto de desenvolvimento de software de aplicação para uma rede CAN e sua interligação com uma rede Ethernet, Licenciatura em Engenharia Eletrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, 2002, Disponível em: <https://web.fe.up.pt/~ee99058/projecto/pdf/Can.pdf>
- [7] CAN Data link in some detail, Disponível em: <http://www.can-cia.org/can-knowledge/can/can-data-link-layers/>
- [8] Higher-layer protocols, Disponível em: <http://www.can-cia.org/can-knowledge/hlp/higher-layer-protocols/>
- [9] Tanenbaum, A. S. Redes de Computadores. 5 ed. São Paulo: Perarson Pretice Hall, 2011, p. 272-273.

Redes Veiculares: Princípios e aplicações

Gabriela Alice Krieck

Engenharia de Controle e Automação
Universidade Federal de Santa Catarina
Blumenau, Santa Catarina, Brasil
Email: gabriela.krieck@grad.ufsc.br

Resumo—As redes veiculares são compostas tipicamente por veículos e dispositivos situados nas margens das estradas. Este artigo tem como principais objetivos apresentar os princípios de funcionamento dessas redes, descrevendo as arquiteturas e os padrões de rede utilizados em sua implementação com enfoque nas camadas física, de roteamento e o acesso ao meio. Também são apresentados algumas aplicações e os principais desafios da área.

I. INTRODUÇÃO

Com os avanços em áreas como a eletrônica, a tecnologia da informação e a comunicação, os veículos automotores passaram a incorporar novas tecnologias tais quais computadores de bordo, sistemas de navegação, sensores e sistemas de frenagem que melhoram dia após dia a experiência do condutor e dos passageiros. Essas tecnologias, no entanto, se restringem à comunicação entre o condutor e o veículo.

O próximo passo na linha da sucessão tecnológica é a comunicação e a interação entre veículos e outros equipamentos situados nas margens das estradas. Esses dispositivos irão compor, após a implementação, as chamadas redes veiculares, através das quais se vislumbra que aplicações como a monitoração em tempo real do tráfego, a prevenção a acidentes e o acesso a internet em todo e qualquer lugar [Li e Wang, 2007] possam se tornar reais.

No contexto das redes veiculares, os nós são compostos pelos equipamentos fixos situados nas margens das estradas e por veículos com interfaces de comunicação sem-fio. Desta forma, os nós dessas redes tendem a possuir alta mobilidade e por esse motivo a implementação em grande escala ainda é um desafio. Outras dificuldades que podem ser destacadas são o tempo reduzido de contato entre dois nós e também a perda de conectividade durante a transmissão de dados.

Esse tem como principal objetivo realizar um estudo a respeito dos princípios e de algumas aplicações destas redes e para isso será dividido em duas partes. A primeira englobará os princípios de funcionamento das redes veiculares, através da descrição das arquiteturas e dos padrões de rede utilizados em sua implementação. Na segunda parte, algumas aplicações dessas redes são apresentados e os principais desafios encontrados discutidos.

II. PRINCÍPIOS DE FUNCIONAMENTO DAS REDES VEÍCULARES

A. Arquitetura

A arquitetura de rede determina de que maneira os elementos (nós) se organizam e se comunicam. Atualmente, as redes veiculares podem ser divididas em três principais arquiteturas [Alves et al., 2009]: ad-hoc puro (Veicular Ad hoc NETWORK - VANET), infraestruturada ou híbrida que serão descritas a seguir.

1) *Redes ad-hoc puro*: Nesta arquitetura, os elementos da rede se comunicam sem a presença de qualquer elemento centralizador. Assim, os nós funcionam como roteadores e encaminham os pacotes através de múltiplos saltos. A principal desvantagem deste tipo de rede é que ela depende do padrão de mobilidade dos veículos e também da densidade de tráfego. No contexto das redes veiculares, o modo ad hoc é também referenciado por V2V (*Vehicle-to-Vehicle*).

2) *Arquitetura infraestruturada*: Na arquitetura infraestruturada, nós estáticos, que centralizam todo o tráfego da rede, são distribuídos ao longo das ruas e estradas atuando como nós intermediários nas comunicações. A grande vantagem dessa arquitetura é o aumento da conectividade e conseqüente possibilidade de comunicação com outras redes. No entanto, a conectividade só é eficaz se houver um grande número de elementos fixos, o que eleva os custos de implementação das redes. Esse modo também pode ser referenciado através do termo V2I (*Vehicle-to-Infrastructure*).

3) *Arquitetura híbrida*: A arquitetura híbrida surgiu como uma solução intermediária entre as arquiteturas infraestruturada e ad hoc. Nesse modo, uma infraestrutura mínima é utilizada com a finalidade de aumentar a conectividade da rede e evitar prejuízos. No entanto, existe também a possibilidade dos veículos se comunicarem entre si. Ou seja, essa arquitetura visa a utilização tanto do modo V2V como também do modo V2I.

Na figura 1 é possível visualizar as três arquiteturas em seus devidos contextos. Muitos pesquisadores se referem às redes veiculares de modo geral como VANETs, mesmo quando existe o uso de uma infraestrutura.

B. Padrões de rede

Em 2004, as redes veiculares começaram a ser padronizadas pelo IEEE dentro do padrão IEEE 802.11. De acordo com BARCELOS (2014), esse padrão está em constante

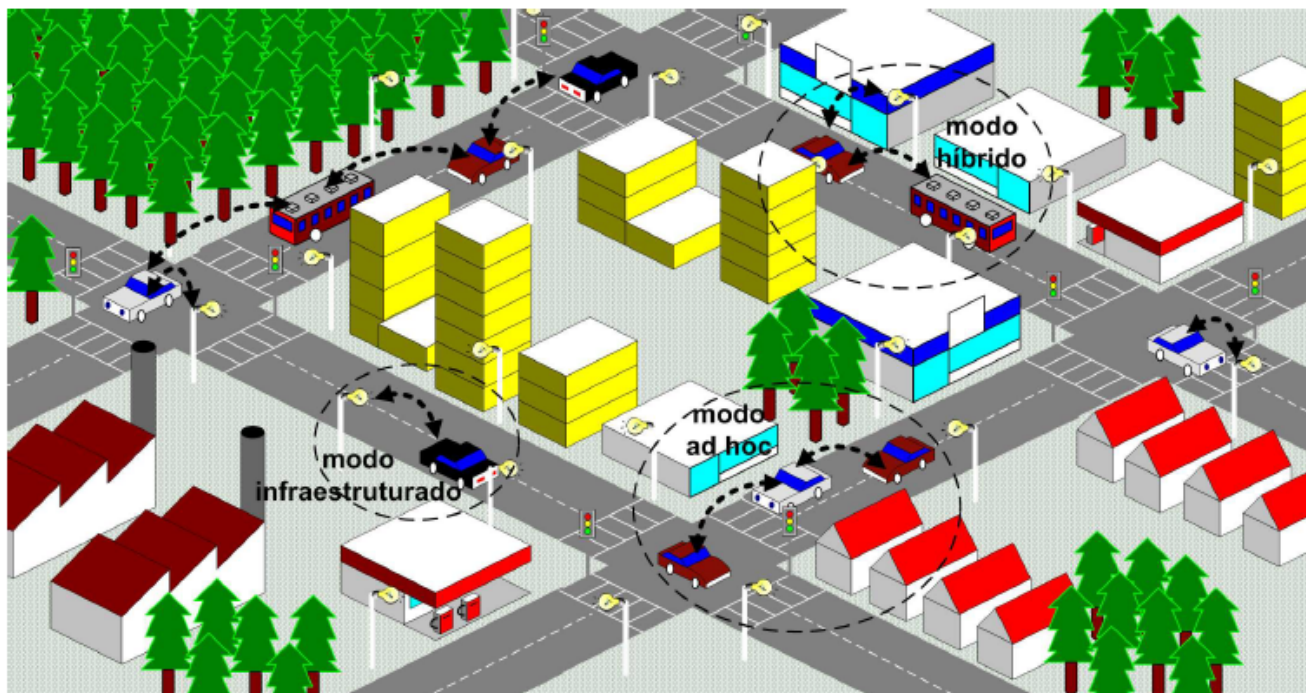


Figura 1. Cenário de uma rede veicular (ALVES, 2005)

desenvolvimento e possui sua última versão publicada em 2010, com a denominação *IEEE 802.11p Wireless Access in Vehicular Environment*, também conhecido como WAVE. Essa arquitetura é regida pelos documentos elaborados pelos grupos IEE 1609 (GROUP, 2003) e IEEE 802.11 (IEEE, 2010) e padroniza as comunicações veiculares. Os padrões IEEE 802.11p, que descreve a camada física e enlace das comunicações veiculares, e os padrões 1609, que definem um conjunto complementar de serviços padronizados e interfaces que viabilizam a comunicação entre veículos e são descritos a seguir.

1) *Padrão 802.11p - WAVE*: O padrão 802.11p opera em uma faixa de frequências entre 5,850GHz e 5,925GHz. Nessa faixa de frequência, os nós comunicam-se a uma distância máxima de 1000 metros, com taxas de transmissão que variam de 1 a 27Mbps. De acordo com ALVES et al., o padrão IEEE 802.11p consegue garantir a transmissão de dados entre veículos que se deslocam entre si a velocidades de até 200km/h.

Nesse padrão, a faixa de frequências é dividida em canais de 10MHz cada. De acordo com BARCELOS, o padrão 802.11p é baseado no padrão 802.11a e possui como diferenças principais a faixa de frequência e a largura dos canais. No padrão 802.11p, existe um canal exclusivo para controle de informações (CCH) e os outros canais são utilizados para diferentes categorias da rede (SCH). O CCH é reservado para a transferência de mensagens curtas de alta prioridade, enquanto que os SCHs transmitem outros tipos de dados.

Ainda acordo com BARCELOS, a cama de acesso ao

meio (MAC) do padrão WAVE é baseado no padrão 802.11e. Esse protocolo usa o método *Enhanced Distributed Channel Access* (EDCA), que diferencia quatro tipos de categorias de aplicação: O tráfego em segundo plano, o tráfego de melhor esforço, o tráfego de voz e o tráfego de vídeo.

Na tabela I é possível visualizar um comparativo entre o padrão 802.11p e WiFi.

Parâmetro	WAVE	WiFi
Faixa de Frequência	5.9GHz	2.4GHz
Largura do canal	10MHz	20MHz
Taxas de Transmissão (Mbps)	3, 4, 5, 6, 9, 12, 18, 24 e 27	6, 9, 12, 18, 24, 36, 48, 54
Modulação	BPSK, QPSK, 16QAM e 64QAM	BPSK, QPSK, 16QAM e 64QAM
Condificação do Canal	1/2, 2/3 e 3/4	1/2, 2/3 e 3/4

Tabela I

COMPARAÇÃO ENTRE OS PADRÕES 802.11P E WiFi

2) *Padrão IEEE 1609*: O padrão IEEE 1609 trata de características específicas das redes veiculares através de uma série de documentos listados a seguir.

i) IEEE 1609.0 – Gerenciamento de recursos: Tem como objetivos fornecer determinados acessos aos recursos de comunicação do sistema. Alguns exemplos de recursos que as aplicações de gerenciamento de recursos podem controlar são o acesso à memória, barramentos especializados e algumas aplicações relacionadas às OBU.

ii) IEEE 1609.2 – Serviços de segurança: Já que as redes veiculares podem ser utilizadas em diversas aplicações, essas redes enfrentam problemas específicos de comunicação. Exemplos são as aplicações emergenciais, que necessitam de baixa latência e também os mecanismos de autenticação das mensagens, que devem ser o mais escalável possível. Garantir um mecanismo flexível, com baixo *overhead* para atender às demandas das aplicações é uma tarefa desafiadora. O padrão IEEE 1609.2 trata desses desafios descrevendo serviços de segurança para a arquitetura WAVE e aplicações que rodam nessas redes.

iii) IEEE 1609.3 – Serviços de rede: Especifica as funções associadas às camadas OSI de enlace, rede e transporte da arquitetura WAVE.

iv) IEEE 1609.4 – Operação Multicanal: Como mencionado anteriormente, a arquitetura WAVE possui aplicações gerais, que devem ser executadas em canais de serviços, e aplicações críticas, que devem ser transmitidas em canais de controle. O padrão IEEE 1609.4 define métodos que visam coordenar e otimizar a alternância de canais que é fundamental para o correto funcionamento da rede.

C. Protocolos de roteamento

O roteamento nas redes veiculares é uma tarefa desafiadora devido à grande mobilidade e à instabilidade dos enlaces das redes sem-fio. Nesse documento nos ateremos a alguns a alguns protocolos de roteamento que podem ser aplicados nas redes *ad hoc* (VANETs).

1) *Ad hoc On-Demand Distance Vector (AODV)*: O Ad-hoc On-Demand Distance Vector (AODV) é um protocolo baseado no *Destination-Sequenced Distance Vector (DSDV)*. Esse protocolo foi projetado para se adaptar a cenários de alta mobilidade, de modo a minimizar o processamento nos nós e evitar o desperdício de banda e pode, desta forma, ser utilizado no contexto das VANETs.

O AODV é um protocolo de roteamento reativo, ou seja, que realiza a descoberta de nós destinos somente quando for necessário fazer uma transmissão. Assim, esse protocolo visa a criação de rotas somente quando for necessário.

Quando um nó deseja enviar uma informação a outro nó, ele primeiro inicia um processo de descobrimento de rota. Desta forma, o nó que deseja transmitir uma informação envia a seus vizinhos um pacote de requisição de rota (*Route Request – RREQ*) via *broadcast*. Os nós que receberam o pacote de requisição o retransmitem até que ele encontre o nó de destino ou um nó que conheça uma rota até o destinatário.

Todos os nós que se envolvem na troca de pacotes RREQ fazem a atualização de suas tabelas de roteamento com base nos dados dos nós vizinhos. Quando um nó reconhece o caminho até o destino ou o destinatário recebe o pacote RREQ, o pacote *Route Reply – RREP* é criado e enviado através da rota reversa ao nó de origem. Quando o remetente recebe o pacote RREP, ele passa a conhecer a rota até o nó destino e atualiza a informação em sua tabela de roteamento.

2) *Optimized Link State Routing Protocol (OSLR)*: O protocolo OSLR é um protocolo pró-ativo implementado para redes

Ad Hoc e se baseia em estados de enlace. Os protocolos pró-ativos realizam periodicamente troca de informações sobre a rede com os nós, a fim de sempre manter suas tabelas de roteamento atualizadas. A grande desvantagem desse protocolo é o aumento do *overhead*, pois as informações de roteamento estão circulando constantemente na rede.

De acordo com BARCELOS (2014), o protocolo OSLR envia periodicamente mensagens de controle aos nós vizinhos a um salto, fornecendo informações do estado de enlace a esse nó com seus nós vizinhos. No entanto, um nó pode receber a mesma mensagem de nós diferentes, o que faz com que o OSLR utilize uma técnica denominada *Multipoint Relaying – MPR* para diminuir o número de mensagens iguais.

Entre os nós da rede, existirão alguns denominados MPR. A escolha dos nós MPR é feita em consenso com os nós vizinho que se localizam a um *hop* de distância. Desta forma, quando uma informação deve ser atualizada na rede, os pacotes enviados por um nó chegarão para toda a vizinhança, mas somente os nós MPR irão retransmitir a informação adiante. Essa metodologia é repetida com todos os nós que recebem os pacotes. Assim, cada nó receberá as informações apenas uma vez, sem que ocorra um *overhead* de informações. A implementação desse modo de roteamento é feito através de um algoritmo.

3) *Better Approach To Mobile Ad-Hoc Networking (BATMAN)*: De acordo com BARCELOS (2014), o protocolo OSLR se mostra muito eficiente para redes Mesh de pequeno e médio porte. Em redes grandes (com mais de cem nós), o cálculo de rotas pode ser considerado uma tarefa complexa para microprocessadores de pequeno porte. O algoritmo BATMAN vem sendo aprimorado desde 2006 e tem como objetivo dividir o conhecimento sobre os melhores caminhos fim-a-fim com todos os nós da rede.

No protocolo BATMAN, cada nó transmite mensagens *broadcasts*, denominadas *Originator Messages (OGM)*, para informar a vizinhança de sua existência. Os nós vizinhos irão retransmitir as OGMs para informar os nós vizinhos da existência do nó que originou a transmissão. Quando um nó recebe uma mensagem OGM, ele atualiza sua tabela de roteamento com essas informações. Quando uma mesma mensagem chega a um nó de destino, ela será descartada, restando apenas a mensagem que chegou mais rápido.

Um nó X aprenderá sobre a existência do nó Y pelo recebimento de suas OGMs, quando as OGMs do nó Y forem reenviadas pelos seus vizinhos próximos. Se X possuir mais de um vizinho, irá detectá-los ao receber mensagens de origem de seus vizinhos de um salto. O algoritmo elege o melhor vizinho de um salto aquele que enviou mais rapidamente a mensagem e assim realiza a configuração da tabela de roteamento.

III. APLICAÇÕES E DESAFIOS

De acordo com ALVES et al. (2009) as aplicações em redes veiculares podem ser divididas em três classes: segurança no trânsito, entretenimento e assistência ao condutor. As aplicações em segurança possuem um caráter preventivo e emergencial, no qual as informações devem trafegar com

rapidez para que o motorista possa reagir rapidamente; a classe de entretenimento inclui aplicações da Internet para as redes veiculares, como sistemas de compartilhamento de conteúdo; as aplicações de assistência ao motorista, por sua vez, envolvem aplicações que auxiliam e automatizam serviços. Nas aplicações de assistência e entretenimento, a latência não é um requisito crítico. Além disso, diferentes aplicações possuem diferentes requisitos e devem ser analisadas caso a caso. A seguir são apresentados algumas aplicações e projetos relacionados às redes veiculares.

A. Segurança

A possibilidade de aumentar a segurança no trânsito é um dos principais propulsores ao desenvolvimento das redes veiculares. De uma maneira geral, aplicações de segurança visam a redução do número e da gravidade dos acidentes através de trocas de informações entre os veículos. As aplicações em nível de segurança possuem requisitos estritos de latência e confiabilidade das mensagens. Além disso, essas aplicações necessitam de segurança e robustez às mensagens falsas e conflitantes.

O CCA (*Cooperative Collision Avoidance*) é um exemplo de aplicação em segurança e objetiva uma abordagem coletiva para evitar colisões entre veículos. De acordo com Biswas et al. (2006), esse Sistema envia mensagens aos condutores avisando a ocorrência de uma situação emergencial. A mesma ideia pode ser utilizada em situações nas quais o condutor possui uma limitação visual (como situações de neblina e chuvas intensas).

Em casos nos quais um acidente já ocorreu, o envio de mensagens com o intuito de informar a ocorrência pode evitar, por exemplo, que veículos próximos também se envolvam no acidente. O envio desse tipo de mensagem pode ainda acelerar o chamado aos serviços de emergência.

De acordo com VSCC (2005), a comunicação entre veículos e equipamentos de infraestrutura podem também reduzir o número de colisões em cruzamentos perigosos. Os automóveis podem avisar a infraestrutura dos cruzamentos a respeito de sua posição e velocidade. Com esses dados, o equipamento de infraestrutura pode calcular uma possível colisão e alertar a respeito do risco iminente. Outra aplicação é o envio de informações pela infraestrutura informando o estado de semáforos, avisando motoristas sobre uma possível violação da sinalização.

Existem ainda outras aplicações onde a troca de informações entre veículos (V2V) ou entre veículo e infraestrutura (V2I) permite a realização segura de manobras no trânsito prevenindo acidentes. Alguns exemplos são situações nas quais os veículos realizam mudanças de faixa, onde mensagens podem ser trocadas entre veículos a fim de evitar colisões laterais ou mesmo ao se aproximar de curvas, onde veículos podem se comunicar com a infraestrutura das estradas, que informaria o condutor se a velocidade utilizada representa ou não perigo, baseando-se em informações como condições do tempo e da estrada e a geometria da curva.

B. Entretenimento

Grande parte das propostas de aplicação das redes veiculares que envolvem questões de entretenimento estão associadas direta ou indiretamente à internet. Os usuários estão cada dia mais dependentes da internet e desejam acessá-la a todo instante e em qualquer lugar. Dentre as aplicações que envolvem as redes veiculares e a internet se destacam a distribuição de áudio e vídeo, a troca de mensagens instantâneas e a troca de músicas e filmes.

Muitas aplicações para as redes veiculares propostas na literatura propõem o uso da arquitetura ad hoc, que dispensa elementos centralizadores. Além disso, no caso de haver uma rede infraestruturada, manter uma rede totalmente conectada requer altos custos. A maioria das aplicações de entretenimento preferem utilizar sistemas par-a-par ao modelo cliente-servidor, que é centralizado. O grande desafio é que muitas dessas aplicações requerem o uso da internet, que só é possível por meio dos pontos fixos de interconexão. No contexto das redes industriais, esses pontos fixos são chamados de *gateways*.

No contexto da internet, uma das principais aplicações é a comunicação par-a-par. Nas redes veiculares, essa ideia é denominada *Car-to-car – C2C*. De acordo com Cohen (2008), a ideia básica é que os veículos troquem pedaços de arquivos desejados entre si, como ocorre em protocolos BitTorrent na internet. A exemplo do que ocorre no BitTorrent, não pode receber pedaços de diferentes nós ao mesmo tempo.

O protocolo Code Torrent é um exemplo de protocolo par-a-par de compartilhamento de conteúdo nas redes veiculares. De acordo com Lee et al. (2006), nesse protocolo um nó escolhe apenas parceiros que estejam a um salto de distância na rede física para evitar a comunicação em múltiplos saltos. O grande problema é que nem sempre vizinhos de um nó estão interessados no mesmo pedaço de arquivo.

O V3 é um sistema de transmissão de vídeo ao vivo veículo-a-veículo (V2V). Ao contrário dos sistemas de compartilhamento de conteúdo, a requisição de vídeo deve ser enviada para um *gateway* localizado às margens da via. Nesta rede, toda a comunicação se dá entre os veículos (V2V). De acordo com Guo et al. (2005), os grandes desafios que esta tecnologia enfrenta são a mobilidade das fontes de vídeo e também a frequência de particionamento da rede. Para contornar esses desafios, o V3 propõe a implementação de uma estratégia de armazenar-transportar-e-encaminhar para transmitir o vídeo em redes particionadas.

Nos dias de hoje, existem inúmeros projetos para o desenvolvimento de aplicações de entretenimento em redes veiculares. Um dos projetos, desenvolvido pela Daimler-Chrysler, busca desenvolver uma aplicação de distribuição de rádio que visa aumentar o número de estações disponíveis para os usuários e também a criação de estações pelos próprios usuários de dentro dos veículos para disponibilizar o conteúdo de seus players para outros usuários [Ramsey, 2009]. De acordo com Ott e Kutscher (2004), outro projeto é o *Drive-thru Internet*, que prevê o acesso à Internet a usuários em

veículos que trafegam em estradas a velocidade que podem atingir 200km/h.

C. Assistência ao motorista

As aplicações de assistência ao motorista visam auxiliar o condutor do veículo através do fornecimento de informações úteis, fornecidas em momentos oportunos. Dentre alguns exemplos, pode-se destacar o aviso de estacionamentos, a condução conjunta de veículos, a localização em mapas, o aumento da visibilidade e veículos sem condutor humano.

Atualmente, as aplicações de indicação de vagas de estacionamento vem recebendo bastante atenção. Um dos argumentos é que além de ser conveniente aos condutores do veículo, ela pode reduzir muito o congestionamento das cidades. O trabalho realizado em Caliskan et al. (2006) revelou que 44 % do trânsito em um distrito da cidade de Munique, na Alemanha se deve somente a busca de lugares para estacionar. Com base nisso, Panayappan et al. (2007) desenvolveram uma proposta de solução que visa a implantação de um sistema de controle e a divulgação de vagas de estacionamento. Uma área onde há vagas de estacionamento foi dividida em pequenas zonas, de forma que cada zona foi gerenciada por uma Unidade de Acostamento (RSU). Cada RSU controla a posição e o estado de vagas nas zonas nas quais o veículo se encontra e verifica se existe alguma vaga livre próxima ao destino pretendido pelo motorista e, em caso afirmativo, informa o motorista. Quando um veículo se encaminha para alguma vaga, a RSU verifica a presença de veículos ocupando outras vagas. Quando um veículo sai de uma vaga, ele informa a RSU da liberação.

Outra aplicação importante para auxílio ao motorista são as informações sobre as condições das vias. Essas aplicações tem como finalidade reduzir o tempo de espera de motoristas em congestionamentos, apresentando rotas alternativas. Nessa linha, o TrafficView é um sistema que propõe a disseminação de dados de tráfego de veículos. De acordo com Nadeem et al. (2004), nessa rede cada automóvel mantém informações próprias sobre a posição e a velocidade, além das informações recebidas por outros nós da rede.

O auxílio em cruzamentos também é considerado uma aplicação importante e interessante. Nesse contexto, as redes veiculares podem obter dados como a posição dos veículos e realizar a comunicação com elementos como os semáforos. Através das informações do veículo e do semáforo, a aplicação pode calcular, por exemplo, a velocidade ideal para que o motorista alcance o cruzamento sem precisar aguardar o semáforo abrir ou outro veículo passar.

Aplicações de condução conjunta de veículos são utilizadas para veículos que viajam juntos. Essas aplicações realizam a troca de informações de posição e velocidade dos veículos do conjunto, permitindo aos condutores um controle rápido e preciso da distância entre veículos.

As aplicações de aumento de visibilidade mostram aos condutores a posição de veículos, construções e objetos na pista em situações de pouca visão como em neblinas ou tempestades. Nos dias de hoje, já existem automóveis que utilizam esse tipo de mecanismos para a visão noturna. As

redes veiculares podem contribuir muito para aplicações desse tipo por acrescentar informações além do campo de visão do motorista.

A condução de veículos sem a intervenção humana é uma aplicação bastante sofisticada. Esses sistemas obtêm informações pela rede a respeito das condições de trânsito e, a partir dessas percepções, tomam as melhores decisões para alcançar seus destinos. Apesar de já existirem diversas propostas de desenvolvimento desses tipos de sistemas, o pleno funcionamento dessas aplicações ainda está muito distante.

IV. CONCLUSÃO

As redes veiculares abrem possibilidades para aplicações que tendem a tornar a experiência de dirigir mais segura, eficaz e agradáveis. No entanto, características que estas redes apresentam, como a alta mobilidade dos nós, ainda impõem desafios enormes em sua implementação. Esse cenário demonstra ainda a fragilidade e limitação destas redes e conseqüentemente a necessidade de implementação novos protocolos que atendam essas demandas.

É importante destacar que além dos trabalhos apresentados por esse documento, existem inúmeros outros nessa área e que apesar do grande número de trabalhos na área, ainda há muito o que se fazer em pesquisa até que as redes veiculares possam atender satisfatoriamente as aplicações.

REFERÊNCIAS

- [1] B. Cohen *The BitTorrent protocol specification*. http://www.bittorrent.org/beps/bep_0003.html. Acessado em 01 de julho de 2016. 2008.
- [2] B. Paul, M. Ibrahim e M. A. N. Bikas, *VANET Routing Protocols: Pros and Cons*, International Journal of Computer Applications, Nº 3, 2011, pp. 11-34.
- [3] F. Li e Y. Wang, *Routing in vehicular ad hoc networks: A survey*. IEEE Vehicular Technology Magazine, 2007, p.12–22.
- [4] J. F. B. Gonçalves, *Arquitetura Baseada em Serviços para Redes Veículo-a-Veículo*, tese de mestrado, Universidade do Porto, Porto, Depto. de Engenharias, 2009.
- [5] J. Ott e D. Kutscher *Drive-thru internet: IEEE 802.11b for "automobile" users*. Em IEEE Conference on Computer Communications (INFOCOM), 2004.
- [6] M. Guo, M. H. Ammar e E. W. Zegura, *V3: a vehicle-to-vehicle live video streaming architecture*. Em IEEE International Conference on Pervasive Computing and Communications (PerCom), 2005, p. 171–180
- [7] R. S. Alves, I. V. do Campbell, R. S. Couto et al, *Redes Veiculares: Princípios, Aplicações e Desafios*, Livro Texto dos Minicursos do XXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2009, pp. 199-254.
- [8] S. Biswas, R. Tatchikou, F. R. Dion, F. *Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety*. IEEE Communications Magazine, 2006, p.74–82.
- [9] T. Nadeem, S. Dashtinezhad, C. Liao e L. Iftode (2004). *Traffic-View: traffic data dissemination using car-to-car communication*. ACM SIGMOBILE Mobile Computing and Communications Review, 2004, p. 6–19.
- [10] U. Lee, J. S. Park, J. Yeh, G. Pau e M. Gerla (2006). *CodeTorrent: content distribution using network coding in VANET*. Em International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare), 2009, p. 1-5.
- [11] VSCC. *Vehicle safety communications project task 3 final report: Identify intelligent vehicle safety applications enabled by DSRC* Relatório Técnico DOT HS 809 859, National Highway Traffic Safety Administration, 2005.
- [12] V. P. Barcelos, *Análise e experimentação do padrão IEEE 802.11p em Redes Veiculares Híbridas*, dissertação de doutorado, Universidade Federal de Lavras, Lavras-MG, Depto. Ciências da Computação, 2014.

- [13] Y. J. Li. *Quality, reliability, security and robustness in heterogeneous networks*. An overview of the dscr/wave technology, New York, 2012, p. 544-558.

AS-INTERFACE

Ronaldo William de Oliveira Baggio
 UNIVERSIDADE FEDERAL DE SANTA CATARINA – CAMPUS BLUMENAU
 Coordenação de Engenharia de Controle e Automação -ECA
 Rua João Pessoa, N.2750, Sede Acadêmica
 Blumenau/SC, Brasil, 89036-256
 rwobaggio@hotmail.com

Resumo – Nesse artigo descreveremos a utilização da rede industrial AS-Interface (AS-i), indicada para atuar nos níveis mais baixos da automação de processo. Além de descrever as diversas metodologias de comunicação, topologias e dispositivos utilizados. Mostrando que sua solução é muito simples dentro das redes industriais, com custo relativamente baixo, com um sistema flexível e de desempenho muito eficiente. A rede AS-i tem muitas vantagens em relações a outras redes que tem a mesma finalidade.

Palavras-chave—sistemas de automação; redes industriais; rede AS-I;

I. INTRODUÇÃO

Este trabalho será desenvolvido em torno da rede AS-Interface (AS-i), Actuator-Sensor Interface, e é um dos vários protocolos existentes para a comunicação entre sistemas automatizados na indústria. Por ser uma solução mais simples em uma rede de automação, e é a ideal para sensores e atuadores trabalhar em rede num sistema de automação. [1]

A rede AS-Interface tem um baixo custo e é uma alternativa eficiente para conectar todos os dispositivos ao controlador usando apenas um par de fios. [1][2]

AS-Interface é uma tecnologia de ‘open’ e há um grande número de fornecedores de equipamentos para a automação. Existem cerca de 28 milhões de dispositivos de campo AS-Interface instalados globalmente, de acordo com AS-International Association. [1][2]

E é uma ótima alternativa de rede para a fiação de dispositivos de campo. Ele pode ser utilizado juntamente com redes de campo de níveis superiores, como o Profibus, DeviceNet, Interbus e Ethernet industrial, pois fornece um baixo custo remoto para soluções I/O. [1][2]

A eficiência da rede AS-Interface pode ser comprovada através dos milhares de produtos e aplicações disponíveis. [1]

A especificação AS-Interface é gerido pela AS-International, uma organização sem fins lucrativos membro financiado, localizado em Gelnhausen/Alemanha. Existem várias organizações internacionais filha em todo o mundo. [2]

II. VANTAGENS E BENEFÍCIOS

Um sistema industrial formado por redes AS-i é considerado como o mais econômico e ideal para comunicação entre atuadores e sensores. Podemos ver na fig.1 fatores que influenciam a escolha da rede AS-i.[4]



Fig. 1 – Análise na escolha da rede AS-i [4]

Vamos destacar uns dos principais benefícios de se usar uma rede AS-i:

- **Simplicidade:** é muito fácil implementar uma rede AS-i, pois um único cabo pode conectar diversos módulos de entradas e saídas para diversos fabricantes, e ela não exige conhecimentos avançados em protocolos e sistemas industriais pelo usuário, além de as fabricantes não precisarem elaborar catálogos aprofundados na descrição de equipamentos. A simplicidade é seu ponto forte. [3]
- **Desempenho:** Os sistemas AS-i, são altamente eficientes e muito rápidos, podendo muito bem substituir sistemas mais avançados, grandes e caros. Sendo realizado de forma simples e confiável. [3]
- **Flexibilidade:** A sua expansibilidade é extremamente simples. Apenas deve-se conectar os módulos, endereça-los e conectar o cabo a rede. A rede AS-i suporta qualquer tipo de topologia, com até 100m de cabeamento, e se adicionar repetidores pode atingir até 300m. [3]
- **Custo:** Tipicamente a rede AS-i foi projetada para reduzir o custo de cabeamento e instalação até pela metade, se comparada a outras redes convencionais. A utilização de um único cabo para a conexão com os dispositivos de campo discretos, reduz a necessidade de gabinete, conduítes e bandejas, além de reduzir significativamente o custo com comissionamento de técnicos e engenheiros. [3]

III. EVOLUÇÃO E VERSÕES

A. Versão original (1994, versão 2.04)

Nas primeiras redes, os módulos (escravos) de interligação dos elementos finais permitiam a conexão de quatro entradas digitais e quatro saídas digitais, resultando no total de 124 entradas e 124 saídas em uma única rede (especificação AS-i 2.0 ou AS-i 1). Porém, nessa arquitetura, o número máximo de escravos é limitado a 31. [4]

Suas principais características estão relacionadas com a substituição automática de um módulo na rede e o tempo de atualização era facilmente calculado pela multiplicação do número de módulos de entrada e saída pelo tempo determinístico de atualização da rede para cada nó (aproximadamente 150 μ s). Este cálculo simplificado não inclui a fase de gerenciamento o qual pode ser desprezado para instalações típicas da rede. [4]

B. Versão com melhorias (1998, Versão 2.14)

Após o seu lançamento os usuários rapidamente adotaram a tecnologia e introduziram novos requisitos com relação à versão. Desta forma, foi publicada a especificação para a rede AS-i 2.1 (ou AS-i 2). [4]

As novas funcionalidades acrescentadas na versão 2.1 são:

- Ampliação do número de escravos de 31 para 62. A capacidade máxima do barramento foi ampliada para 248 + 186 E/S, mas o tempo de ciclo passou para 10ms. [4]
- Um bit adicional no registro de status é utilizado para sinalizar erros de periféricos. A indicação de status de funcionamento dos escravos foi padronizada e ampliada. [4]
- O número de perfis de escravos foi ampliado de 15 para 225 com a adição de novos ID codes. [4]
- Melhor tratamento de sinais analógicos, ampliando o espectro de atuação das redes AS-i. [4]

Os chips para a versão 2.1 da rede AS-i são produzidos por dois consórcios distintos: Siemens e Festo desenvolveram em conjunto o chip SAP4.1, pino a pino compatível com o chip SAP4, e o consórcio de oito outros membros (Bosch, Hirschmann, ifm electronic, Leuze, Lumberg, Klockner Moeller, Pepperl+Fuchs and Schneider Electric) desenvolveu o chip A2SI. Ambos os chips proporcionam as funcionalidades da versão 2.1. [4]

C. Versão mais moderna (2005/2007, Versão 3.0)

Até o ano de 2005 o sucesso mundial da rede AS-i, com aproximadamente 10 milhões de nós em operação, promoveu a introdução de novos requisitos para a rede. Além disto, o crescente uso da Ethernet em protocolos industriais demandou soluções de baixo nível que superassem as inerentes falhas da Ethernet (por exemplo: topologia limitada, grandes pacotes de dados, alto custo no uso de roteadores, entre outros). Essa especificação atende aos usuários de forma a definir novos perfis para dados discretos e analógicos além da introdução de um perfil de transmissão de dados serial (especificação 3.0 ou AS-i 3). [4]

- Nós de entradas e saídas discretas suportando endereçamento estendido (A/B) com 4 entradas e 4 saídas; [4]
- Nós de entradas e saídas discretas suportando endereçamento estendido (A/B) com 8 entradas e 8 saídas; [4]
- Canal analógico configurável (8, 12 ou 16 bits);
- Canal de dados discreto com comunicação serial full-duplex. [4]

Com essas novas características, a rede AS-i torna-se uma parceira ideal para quaisquer outros protocolos industriais baseados na comunicação Ethernet. Gateways para EtherNet/IP, PROFINET, Modbus/TCP e outros estão disponíveis. Alguns especialistas da área de controle dizem que nos próximos 10 anos não haverá mais soluções intermediárias entre a rede AS-i e a Ethernet para novas instalações. [4]

IV. CARACTERÍSTICAS

O nome Actuator Sensor Interface representa o seu próprio conceito. É uma solução simples para a integração de sensores e atuadores discretos em sistema de controle de processos. Esta rede possui uma série de características que são apresentadas na fig.2 a seguir. [4]

Topologia	Estrutura em árvore
Meio físico	Cabo não blindado com dois fios para dados e energia (24VDC/8A)
Comprimento máximo do cabo	100 m
Número de dispositivos escravos	31 máx
Número de pontos	Até 4 sensores e 4 atuadores por escravo (máx 31 * 4 = 124 bi direcional, máx 31 * 8 = 248 sinais binários)
Endereçamento	Cada escravo possui um endereço determinado. O endereço é definido pelo mestre ou ferramenta de programação.
Mensagens	A mensagem parte do mestre para um endereço único com resposta imediata do escravo
Bit rate	Transmite 4bits/slave/mensagem. Todos os escravos são chamados sequencialmente pelo mestre e recebem 4 bits de dados. Cada escravo responde imediatamente com 4 bits de dados.
Tempo de ciclo com 31 escravos	5 ms
Deteção de erros	As mensagens incorretas são identificadas e retransmitidas
Serviços do mestre	Inicialização da rede Identificação dos participantes Definição assíncrona dos parâmetros para os escravos Diagnóstico do barramento e dos escravos Mensagens de erro para o computador host Definição de endereços em escravos substituídos
Operação do mestre	Realiza <i>polling</i> cíclico de todos os participantes. Realiza a transmissão cíclica de dados para o computador <i>host</i> ou para os escravos.

Fig. 2 - Principais características [5]

V. CONECTIVIDADE

A rede AS-i pode se conectar ao nível de controle principal de duas formas. A primeira forma é a conexão direta (Fig.3). Neste caso, o mestre é parte de um CLP ou PC

sendo executado dentro dos tempos de ciclos determinados por esses dispositivos. Um mestre AS-i pode ser construído por qualquer fabricante uma vez que se trata de um padrão aberto. [4]

A segunda maneira de se conectar é através de um acoplador (gateway) entre uma rede de mais alto nível e a rede AS-i. Hoje existem acopladores para outras redes de campo, tais como: Profibus, Interbus, FIP, DeviceNet, CAN, etc. [5]

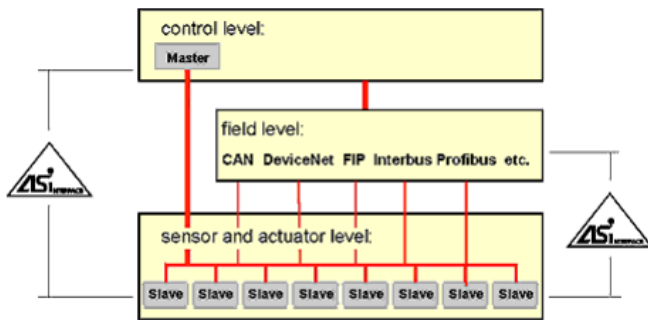


Fig. 3 - Interligação das outras redes [5]

VI. CONCEITOS GERAIS

A rede AS-Interface propicia a interligação de sensores e atuadores, via uma rede de baixo custo, e que pode operar no ambiente industrial poluído eletromagneticamente. O sistema foi elaborado com comprometimento de vários fabricantes, que se propuseram a desenvolver uma rede de comunicação de baixo custo, para o nível de 'chão de fábrica' da automação de campo. O layout principal é a interligação de todos os sensores / atuadores por apenas um par de fio aos controladores lógicos (CLP), isso evita a poluição visual e a desorganização dos fios. [1][2]

Tradicionalmente essas conexões são feitas por pares de fios que conectam um a um os atuadores e sensores ao controlador correspondente. [4]

O sistema AS-i é configurado e controlado por um mestre, o qual programa a interface entre um controlador e o sistema AS-i. Esse mestre troca informações continuamente com todos os sensores e atuadores ligados ao barramento AS-i de forma pré-determinada e cíclica. [4]

A Fig.4 ilustra o sistema AS-i como um todo, evidenciando os seus principais componentes: cabo, fonte AS-i com seu circuito de desacoplamento, o mestre e o escravo AS-i. [4]

- Interface 1: entre o escravo e os sensores e atuadores;
- Interface 2: entre os equipamentos (fonte, mestre e escravo) e o meio de transmissão;
- Interface 3: entre o mestre e o host, ou seja, uma entidade qualquer que acessa a rede AS-i de um nível superior.

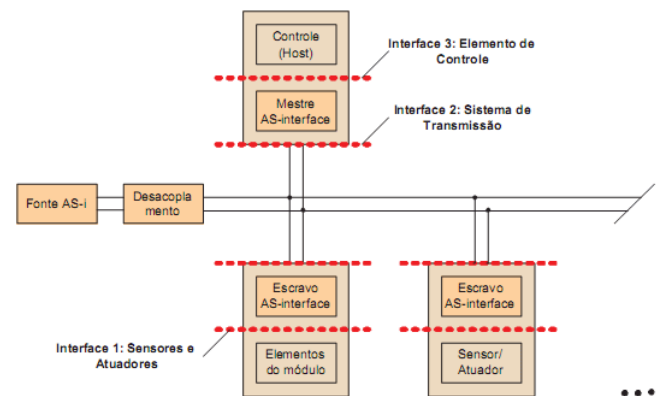


Fig. 4 – Interface [4]

AS-Interface é um sistema que requer 4 componentes básicos:

- Mestre da rede, normalmente sob a forma de um gateway para uma rede industrial de nível superior. [3]
- Escravos de rede, na maioria dos casos módulos de entrada e saída. [3]
- Fonte de energia, utilizada para fazer a alimentação dos escravos da rede e comunicação com o mestre da rede. [3]
- Infraestrutura de cabeamento e topologia, majoritariamente é utilizado o cabo plano amarelo. [3]

A. Mestre da rede (interface 3)

O mestre AS-i faz a ligação entre um controlador ou um sistema Fieldbus (Host) os sensores e atuadores da rede AS-i. Ele gerencia o tráfego de mensagens no barramento de forma independente, de forma que para o Host os sensores e atuadores são vistos como na arquitetura tradicional. Além disso, o mestre provê funções adicionais de configuração e diagnóstico da rede AS-i. [4]

A especificação do AS-i divide o mestre em três camadas, que o descreve desde o cabo de transmissão até a interface 3. [4]

- Camada mestre;
- Controle de execução;
- Controle de transmissão.

A transmissão física das requisições do mestre é especificada pela interface 1 e possui basicamente as mesmas características do escravo. [4]

A camada lógica mais baixa é a camada de transmissão, que é responsável pela transmissão e recepção de telegramas individuais. A repetição automática de telegramas quando uma resposta do escravo falha é possível, garantindo integridade para as camadas superiores. [4]

O controle de sequência, ou controle de execução, está logo acima da camada de transmissão e passa requisições de transmissão de dados para ela. A função dessa camada é controlar a sequência em que os telegramas são enviados, e também processar as funções chamadas pela camada superior, gerando os telegramas corretamente e na sequência requerida. [4]

A camada mais alta é chamada de camada mestre, e realiza a interface com o Host.

B. Escravo da rede (interface 1)

O escravo AS-i, como visto na Fig.4 faz a conexão entre o sistema de transmissão e a interface 1, na qual estão conectados os sensores e atuadores. O escravo os alimenta e gerencia sua comunicação com o mestre. Desde o início do desenvolvimento da especificação do AS-i esteve claro que o escravo deveria ser pequeno e compacto para poder ser integrado diretamente aos sensores e atuadores. Isso foi atingido graças ao uso de circuitos integrados, dando origem ao famoso chip AS-i. [4]

O chip escravo AS-i permite que sensores, atuadores e ou módulos possam ser ligados ao barramento AS-i como um dispositivo escravo, reconhecendo os comandos de saída do mestre e enviando dados de resposta. Um grande número de sensores e atuadores utilizados ultimamente na automação, permitem um baixo custo por conexão em dispositivos escravos AS-i. No caso de dispositivos analógicos, os dados ultrapassam os 4 bits de informação útil por ciclo. Neste caso, os dados são divididos e enviados em vários ciclos. Existem duas possibilidades de uso de um chip escravo AS-i: [4]

- O chip AS-i pode ser embutido em sensores ou atuadores, ou;
- Embutidos em módulos no qual pode-se conectar sensores e atuadores convencionais.

Independente da forma como for implementado o escravo, ele terá uma arquitetura em que as conexões com a interface 1 podem ser físicas ou lógicas e as com a interface 2 necessariamente são físicas.

C. Comunicação Mestre / Escravo

A comunicação mestre / escravo, possui um mestre para gerenciar a comunicação, e tem como função solicitar e receber os dados e comandos. Os outros participantes da rede conhecidos como escravos, que nunca iniciam uma comunicação e respondem com dados para o mestre, que mantém uma lista de todos os escravos presentes na rede e rotineiramente solicita para cada escravo a troca de dados. Para melhor entendimento observe a fig.5 a seguir: [1]

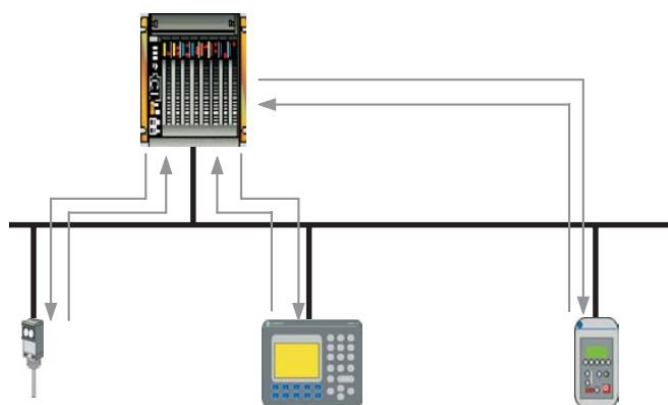


Fig. 5 - Comunicação mestre / escravo. [1]

D. Fonte de alimentação AS-Interface

Os dados e a alimentação trafegam pelo mesmo cabo. A fonte de alimentação utilizada normalmente é de corrente contínua de 30V. Uma corrente alternada é modulada sobre essa corrente contínua como um transmissor de dados. [1]

E. Topologias

A rede AS-Interface pode ser montada como instalações elétricas usuais. Por ser robusta não há nenhuma restrição quanto a estrutura, apenas respeitar o limite máximo de 100m de comprimento para o cabo. Na necessidade de conexões com comprimentos maiores, devem-se usar repetidores para aumentar o alcance da rede. Os módulos AS-i podem ser instalados em forma linear, estrela, árvore e anel (Fig.6). [1][4].

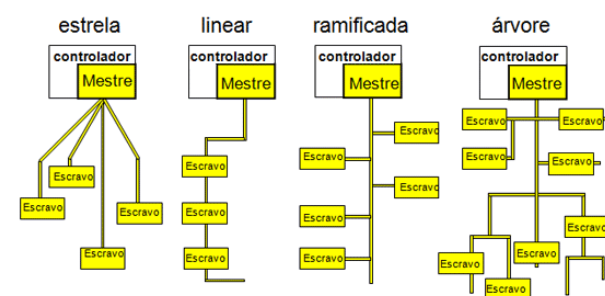


Fig. 6 – Topologia física da rede. [4]

F. Cabeamento

A rede AS-i substitui o tradicional arranjo de cabos múltiplos, caixas de passagem, canaletas, dutos de cabos por um simples cabo especialmente desenvolvido para rede AS-i, esse cabo se caracteriza por ter somente um par de fios, e caminham juntos a alimentação dos sensores ou atuadores em 24Vcc. [4]

AS-i proporciona uma redução nos custos de instalação e manutenção. Agora, um cabo padronizado com 2 fios habilita a troca de informações e ao mesmo tempo a alimentação dos equipamentos. Escravos são conectados diretamente no barramento sem a necessidade de interligação adicional. O cabo de flexível de duas vias mostrado na fig.7 é considerado o padrão para a rede AS-i. Mas também existe o cabo circular PP com as mesmas características elétricas, mas só deve ser usado se for especificado pela fabricante.

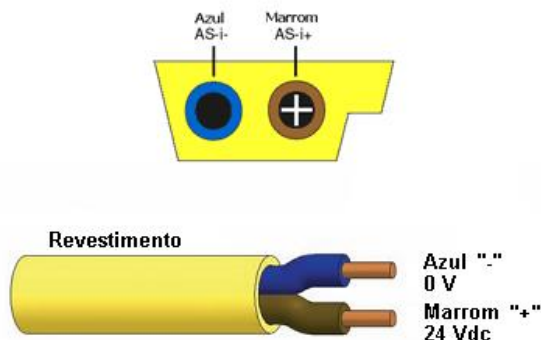


Fig. 7 - Cabo flexível padrão [4]

O comprimento máximo de um cabo do barramento AS-i é de 100 m sem o uso de repetidores. No caso de serem

utilizados os repetidores um comprimento máximo de 300 m é permitido. [4]

G. Modulação

A modulação adequada para a rede AS-i levou vários aspectos em consideração. E isso levou a criação de um novo procedimento de modulação, conhecido por Modulação de Pulsos Alternados (APM - Alternating Pulse Modulation). Podemos destacar alguns requisitos: [1][4]

- O sinal de modulação deve ser sobreposto ao sinal de alimentação. [1][4]
- O processo de transmissão deve ser simples e barato para poder ser integrado no escravo. [1][4]
- O sinal deve estar concentrado em uma banda estreita para não ser afetado por interferência eletromagnética induzida no cabo (que não possui blindagem). [1][4]
- Níveis altos de radiação de ruído são também inaceitáveis. [1][4]

A modulação APM é um procedimento para transmissão serial em banda base e pode ser observada na Fig.8. A sequência de bits é inicialmente codificada de maneira que uma mudança de fase ocorre sempre que o sinal se modifica (codificação Manchester). [4]

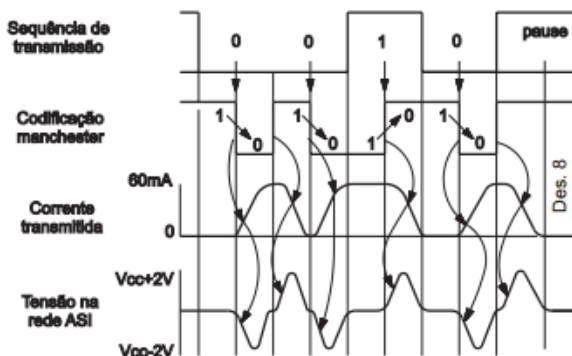


Fig. 8 Modulação APM. [1]

A corrente de transmissão é gerada em conjunto com indutores presentes na linha, que em caso de aumento de corrente provoca um pulso negativo, e em decréscimo da corrente gera um pulso positivo de tensão na linha.

H. Tratamento de erros

O reconhecimento confiável de erros é de grande importância para comunicação sem falhas através do cabo AS-i, que geralmente não é blindado. Devido ao fato de os telegramas AS-i envolvidos nas transações serem bastante curtos, a detecção de erros é diferente daquela aplicada em geral a outras redes de campo. A requisição do mestre possui apenas 11 bits de dados a serem checados, e a resposta do escravo, 4 bits. Adicionar bits para verificação de erros nas mensagens faria com que a taxa de transmissão caísse drasticamente. Ao invés disso, o AS-i faz uso do tamanho conhecido das mensagens, dos padrões de bits especificados e da modulação de pulsos alternados para ser capaz de distinguir os seguintes erros: [4]

- Erro de Start Bit;
- Erro de alternância;

- Erro de pausa;
- Erro de Informação;
- Erro de paridade;
- Erro de End Bit e;
- Erro de tamanho do telegrama.

Cada requisição do mestre e cada resposta do escravo é submetida a uma verificação desses 7 tipos de erro. Se algum deles é detectado, a requisição (ou resposta) é considerada inválida. [4]

I. Acesso ao meio

O mestre envia um telegrama que é recebido pelo escravo em um endereço particular e o escravo contactado nesse endereço responde dentro do tempo previsto. Essa operação é designada como uma transação. O sistema de transmissão permite a conexão ao barramento de um mestre e de até 62 escravos. [1][2][4] Esse procedimento escolhido permite a construção de escravos muito simples, com um custo baixo, provê flexibilidade e integridade. No caso de pequenas perturbações na rede, o mestre pode, por exemplo, repetir telegramas para o endereço do qual não recebeu resposta, ou reposta inválida. As mensagens AS-i, portanto, são de dois tipos: as que são enviadas pelo mestre e as repostas do escravo. A Fig.9 mostra uma transação envolvida. [4]

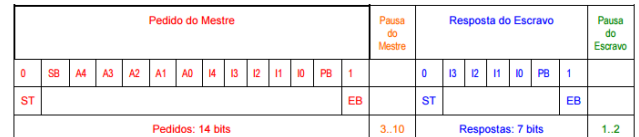


Fig. 9 - Estrutura de uma mensagem AS-i. [5]

Podemos reparar que a transação é constituída da requisição do mestre; uma pausa; resposta do escravo e; pausa do escravo. [4][5]

Uma requisição do mestre consiste de:

- Start Bit (SB). Identifica o início de uma requisição do mestre no barramento, com valor sempre em 0.
- Control Bit (CB). Identifica o tipo de requisição: 0 para requisição de dados, parâmetros ou atribuição de endereço e 1 para chamadas de comando.
- Address (A4.A0). Endereço do escravo é requisitado.
- Informação (I4.I0). Esses 5 bits compõem, dependendo da requisição a ser feita, os dados passados ao escravo.
- Bit de Paridade (PB). O número de bits com valor 1 numa requisição, excetuando o end bit, deve ser par.
- End Bit (EB). Identifica o final de uma requisição do mestre. Sempre possui valor 1. [4][5]

E uma resposta de qualquer escravo consiste de:

- Start Bit (SB). Identifica o início de uma resposta do escravo. Possui sempre valor 0.

- Informação (I3.I0). Esses bits compõem a informação propriamente passada ao mestre.
- Bit de Paridade (PB). Assim como uma requisição, uma resposta também deve possuir paridade par.
- End Bit (EB). Sempre de valor 1, assinala o final de uma resposta do escravo. [4][5]

VII. DESVANTAGENS

O AS-i foi conscientemente construído e otimizado para uso em aplicações abaixo dos fieldbuses. Sendo assim, algumas capacidades dos fieldbuses de alto nível não podem ser realizadas em AS-i. Algumas limitações são expostas logo a baixo: [4]

- Os dados transmitidos em AS-i são limitados a 4 bits por escravos que podem ser trocados a cada ciclo. Mensagens longas podem ser transmitidas dividindo-as em vários ciclos. Isto pode ser usado em processos de dinâmicas lentas, como pressão ou temperatura (valores analógicos). [4]
- AS-i é estritamente mestre-escravo, com varredura cíclica por escravos. Isto impede a transmissão assíncrona pelos sensores e atuadores. Os escravos devem aguardar 10 ms (no caso de uma rede com 62 escravos) até ser chamado novamente. [4]
- A transferência de dados de escravo para escravo só é possível via mestre. [4]
- A limitação de comprimento do cabo é de 100m sem o uso de repetidores. Esta limitação física se deve a manutenção de outros critérios como o tempo de ciclo da rede, tipo de topologia livre e a não exigência de resistores de terminação. [4]

VIII. PROTOCOLOS

Os protocolos regulamentam os componentes e suas instalações. Abaixo podemos ver alguns desses protocolos atualizados:

- Statutory Obligations (including the Electricity Ordinance, Chapter 406); [2]
- IEC 364: Electrical installations of buildings; [2]
- IEC 60947-1: Low-voltage switchgear and controlgear – Part 1: General rules; [2]
- IEC 529: Degrees of protection provided by enclosures (IP Code); [2]
- IEC 439: Low-voltage switchgear and control gear assemblies; [2]
- IEC 62026-1: Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs); [2]

- IEC 62026-2: Actuator sensor interface (AS-i); [2]
- IEC 61131-2: Programmable controllers – Part 2: Equipment requirements and tests; [2]
- IEC 61076-2-101: Connectors for electronic equipment: Part 2-101 Circular connectors (M8 & M12); [2]
- EN 954-1: Safety of machinery. Safety related parts of control systems. General principles for design; [2]
- CISPR 11: Industrial, scientific and medical (ISM) radio-frequency equipment – Electromagnetic disturbance characteristics – Limits and methods of measurement. [2]

IX. CONCLUSÃO

A rede de comunicação AS-I é um sistema bastante versátil devido à construção de seu hardware. A disponibilização dos dispositivos e fontes a qualquer ponto do barramento, o tráfego da alimentação com os dados no mesmo par de fios são um diferencial, além de possuir um cabeamento robusto ideal para o “chão de fábrica”.

Apesar de ter essas características que barateiam o custo devido à redução de materiais e mão de obra, se houver o rompimento de algum desses condutores, pode haver a perda de comunicação com todos os dispositivos provocando a parada do sistema.

Existem diversas redes industriais, com os mais variados tipos de metodologias empregadas. Desta forma não se pode falar em um sistema bom ou ruim, mas sim no que melhor se adéqua a situação proposta para a aplicação.

AGRADECIMENTOS

Este artigo foi elaborado a partir do conhecimento adquirido através de pesquisa e de ensinamentos do professor Dr. Adão Boava durante as aulas. Partes desse documento devem inclusive ser creditadas a autores desconhecidos que escreveram os artigos, catálogos ou documentos sobre o tema, e que serviram de referência para a minha pesquisa através da internet. Agradeço ao professor Dr. Adão Boava, a alguns colegas de turma por lerem e prestarem sua opinião sobre o artigo escrito, em especial Gabriel, Jorge, Diogo, Enrique e Caique.

REFERÊNCIAS

- [1] <http://www.sense.com.br/arquivos/produtos/arq0/3000000093A.pdf> . Em 27/06/2016, 14:20h.
- [2] <http://www.as-interface.net/> . Em 27/06/2016, 15:30h.
- [3] <http://www.smar.com/newsletter/marketing/index150.html> . Em 28/06/2016, 20:30h.
- [4] <http://www.smar.com/brasil/asi> . Em 28/06/2016, 21:30h.
- [5] <http://www.cpdee.ufmg.br/~seixas/PaginaSDA/Download/DownloadFiles/Asi.pdf> . Em 29/06/2016, 14:30h.

P-Net - Process Network

Guilherme Moser Manerichi
 Universidade Federal de Santa Catarina
 Blumenau, Brasil
 Email: guilherme.manerichi@grad.ufsc.br

Resumo — P-Net ou Process Network (Rede de Processo) é um padrão fieldbus para redes industriais e foi desenvolvido na Dinamarca pela PROCES-DATA.

No P-Net é utilizado um padrão multi-mestre baseado no Token-Ring Virtual, onde cada mestre solicita a mensagem para um escravo endereçado e o mesmo deve responde-lo no mesmo momento. Utiliza o padrão RS-485 e suas conexões são de par-trançado blindado diminuindo os erros de transmissão. É possível interligar equipamentos com uma distância de até 1200m sem a necessidade de repetidores.

Abstract – P-Net or Process Network is a fieldbus standard to industrial network and was developed at Denmark by PROCES-DATA.

P-Net uses a multimaster standard based in virtual-token-ring, VTP, wheres the network master who has the token request a message to slave and the slave must to request the master faster as possible. Uses RS-485 standard and your connections cables are a pair twisted shielded. Connect devices until 1200 without repeaters in the best topology.

Index Terms— P-Net, Multi-Mestre, Multi-Net, Token Ring Virtual.

I. INTRODUÇÃO

P-Net é um padrão *fieldbus* (Barramento de Campo) internacional para redes industriais razoavelmente antigo, foi desenvolvido na Dinamarca pela PROCES-DATA e sua primeira versão *fieldbus* é datada de 1984. Neste mesmo período outros protocolos foram surgindo na Europa com a necessidade da intercomunicação entre dispositivos de fábrica. Na França tivemos o surgimento do FIP em 1982 e na Alemanha tivemos o PROFIBUS 1984.

Foi desenvolvido para interconectar componentes de um processo como computadores, sensores inteligentes, atuadores, CLP's, módulos I/O's entre outros dispositivos.

Segundo o P-Net Standard, as aplicações com o P-Net são caracterizadas por seu baixo custo para pequenos sistemas, e conforme o sistema aumenta o custo sobre linearmente.[1]

Possui alguns pontos interessantes que fazem com que sua reduzida velocidade de transmissão não o torne tão inferior quanto à protocolos como PROFIBUS.[2]

No P-Net é utilizado um padrão multi-mestre baseado no Token-Ring Virtual, onde cada mestre solicita a mensagem para um escravo endereçado e o mesmo deve responde-lo no mesmo momento. É possível interligar equipamentos com uma distância de até 1200m sem a necessidade de repetidores.

Os dados são enviados de forma assíncrona usando o NRZ, Non-Returning-Zero e sua velocidade de transmissão máxima é de 76800bps. [2]

II. CARACTERÍSTICAS ELÉTRICAS

Utiliza-se par-trançado blindado, desta forma evitando-se muito dos ruídos comuns de chão de fábrica que causam um enfraquecimento do sinal até mesmo a perda total do mesmo.

Aliado ao padrão RS-485 desenvolvido pela *Electronics Industry Association* – EIA [4], padrão desenvolvido justamente para o ambiente industrial, garante-se um total de 1200m de espaçamento entre dois equipamentos ainda mantendo um sinal adequado para o uso, isso significa a não necessidade de repetidores neste intervalo. O padrão RS-485 substitui padrões como RS-232 e RS-424 que não possibilitavam intercomunicação barramento.

As interfaces do P-Net (Interligações) são galvanicamente isoladas e permitem que até 125 dispositivos sejam conectados a cada segmento.

A P-Net Standard especifica que o cabo de par trançado único pode ser suficiente para a comunicação, entretanto percebeu-se que o uso de par trançado duplo otimizava muito a imunidade ao ruído e a velocidade na instalação não era tão afetada, logo chegou-se a conclusão que a diferença no custo compensaria pelo benefício. [3]

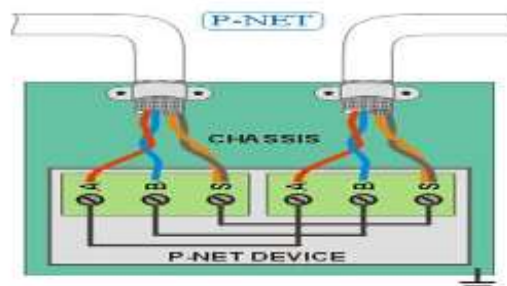


Figura 1: Par trançado blindado duplo [3]

Conforme a distância necessária para a interligação na rede, os cabos 24AWG(0–100m), 22AWG(100-400m) ou 20AWG(400-1200m).

A ligação com par-trançado único acontece da mesma forma, porém no lugar dos dois cabos ligados ao 'S', a blindagem do cabo será ligado ao 'S'. 'S' é a referência.

III. TOPOLOGIAS P-NET

Como dito na introdução, o P-Net é um protocolo multi-mestre que opera com um Token-Ring-Virtual. Para a instalação o usuário pode escolher entre topologia em anel, barramento ou 4-Wire P-NET, sendo o mais comum a topologia anel. Como ilustrado nas figuras abaixo:

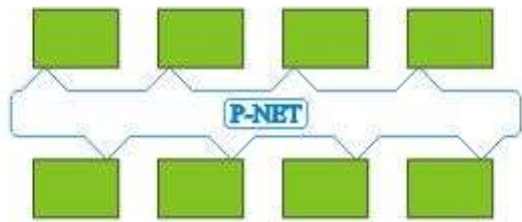


Figura 2: Topologia Anel [3]

Na topologia em anel como visto na figura 2, é a ligação tradicional para o P-Net. O barramento é interconectado na forma de um anel físico entre as estações. Como a comunicação é bidirecional, caso aconteça algo para alguma estação, a comunicação não será interrompida no anel o alcance da rede é de 1200m.

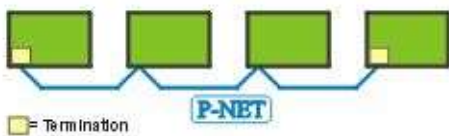


Figura 3: Topologia Barramento [3]

Na figura 3 temos a topologia em barramento, porém neste caso o alcance máximo da rede é de 600m sem o uso de repetidores.

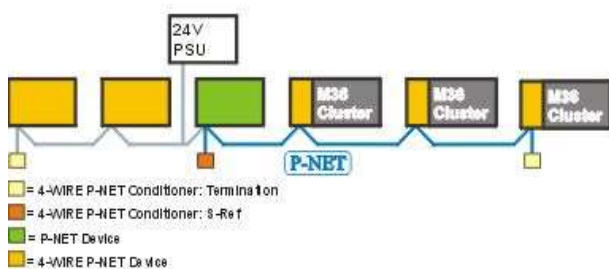


Figura 4: Barramento 4 Cabos [3]

Esta topologia é a mesma que a anterior, utiliza basicamente RS-485 como as duas anteriores porém neste, obrigatoriamente o cabeamento serão dois cabos de par trançado blindados, pois além de um dos pares prover a comunicação o outro proverá a alimentação do circuito, ou seja, os dispositivos não precisariam de fontes externas. A diferença primordial deste para os demais, pois a referência do sinal agora é dada pela

fonte, o que garante que todos os dispositivos terão o mesmo valor de referência, mesmo que ocorram quedas de tensão ou variações pelo barramento. Porém a distância para a rede continua a mesma, 600 metros sem usar repetidores.

IV. MAC – CONTROLE DE ACESSO AO MEIO

No P-NET pode-se ter até 32 mestres em um mesmo barramento, por este motivo é conhecido por multi-mestre. O Token passará por cada um desses mestres que poderá pegar e solicitar informações aos seus escravos. Depois de realizada a operação o Token será passado para o próximo mestre da rede.

Aqui diferentemente do PROFIBUS, o Token utilizado por um período de tempo, isso quer dizer que o Token estará disponível para o mestre por um intervalo de tempo, logo após será repassado para o próximo mestre.

No PROFIBUS há a necessidade de o Token ser enviado em forma de uma mensagem o que gera tráfego e aumenta a probabilidade de algum erro de transferência de Token.

Um mestre na rede P-NET é conhecido também como controlador, ele será o responsável pelas solicitações de informações aos escravos. Um mestre também pode assumir o papel de um escravo, assim os outros mestres poderam solicitar informações à ele.

Os escravos normalmente são dispositivos conectados à sensores e/ou atuadores e são dotados de certa inteligência para que possam ler e fazer determinadas funções.

Os módulos normalmente não trabalham somente com um sensor ou um atuador, mas sim com alguns sensores e atuadores juntos, pois estes dispositivos escravos tem uma quantidade de inputs e outputs.

A taxa de velocidade padrão do P-Net é de 76.8Kbps que foi obtida de uma equilíbrio entre o transporte da alta velocidade exigido pela rede e restrições de distância ou tipo de cabo utilizado na rede.

Um quadro possui cinco campos sendo eles representados na tabela abaixo:

Tabela Quadro do P-Net [5]

Endereço 2 bytes	Controle/ Status 1 byte	Informação de Comprimento 1 byte	Dados 0 – 63 bytes	Detecção de Erros 1-2 bytes
---------------------	-------------------------------	---	--------------------------	--------------------------------------

No P-Net cada byte do quadro corresponde na realidade por 11 bits, então se levarmos em consideração que o maior tamanho para o quadro será de 69 bytes teremos um total de $(69 \times 11) = 759$ bits são transmitidos, só de informação serão 693 bits que podem ser dados em geral sensores analógicos ou binários.

No pior dos casos, onde o quadro é o maior possível nós obteremos um tempo de resposta do escravo a uma velocidade de 76800bps de 20.15ms.

V. FUNCIONAMENTO TOKEN VIRTUAL

Para a comunicação utilizam-se dois contadores regulam sincronização da rede determinística.

O primeiro contador trata de coordenar o retentor do Token.

O contador armazena o endereço do mestre que possui o Token no momento. Após o mestre ter realizado o que pretendia fazer, ou no caso ter realizado a requisição e o barramento ter ficado ocioso por alguns períodos de bit, mais precisamente 40 períodos de bit à uma velocidade de 76800bps que resulta em um tempo de 520us, os contadores são incrementados, isso significa que o Token estará disponível para o próximo mestre. Dessa forma garantimos que o barramento estará disponível para uma próxima requisição. [5]

Depois do Token chegar ao último mestre este contador é resetado, e o Token volta para o início do ciclo realizando-o novamente.

Para que se saiba o tempo que o barramento está ocioso é necessário outro contador, que é incrementado por cada período de bit ocioso do barramento. Logo quando chega-se aos 40 períodos de bit acontece a passagem do Token. É reiniciado após a ocorrência de um checkpoint.

Caso o mestre não tenha nada para transmitir o barramento vai continuar inativo e continuará a contagem e a cada 10 períodos de bit, ou no caso 130us, o primeiro contador de todos os mestres serão incrementados, para que os outros mestres possam utilizar o barramento.

Enquanto ninguém utilizar o barramento o segundo contador continuará a contar e a cada 130us o Token será oferecido ao próximo mestre até que um mestre o pegue.

Pela padronização o protocolo permite até um ciclo de mensagem por Token requisitado, para que o Token permaneça em um mestre somente para sempre.

Após a requisição do Token por parte do mestre que pode levar entre 2 e 7 bits de período no pior dos casos, o mestre envia a requisição para o escravo. Após 11 ou 30 bits de período o barramento está livre para o escravo utilizar tem um período máximo de 30 bits de período, 390us, para utiliza. Depois do mestre ter recebido a mensagem do escravo encerrasse o ciclo da mensagem, então se passarão os 40 bits de período do segundo contador e então completa-se o tempo que o mestre permanece com o Token, assim o Token vai para o próximo mestre. [2]

A figura 5 ilustra os dois contadores do processo de troca de Token virtual.

O quadro em azul significa que o barramento está em uso, e podemos ver que o contador coordenador de Token está no valor três, ou seja o mestre três tem o Token, depois de usado o barramento começa a contagem do contador de barramento ocioso, que ao chegar em 40 bits de período como pode ser visto na figura o Token passou para o mestre 4, ele não utilizou o Token, passou 10 bits de período, o Token vai para o mestre 1, mestre 1 também não usa, vai para o mestre 2, para o 3 e por fim o mestre 4 pega o Token e o barramento começa a ser usado, representado pelo retângulo em azul. [5]

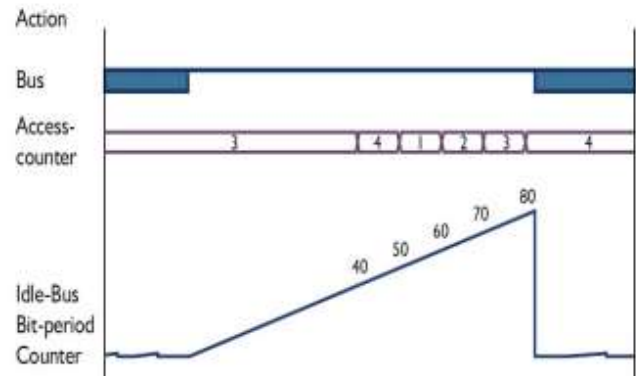


Figura 5: Funcionamento Token Virtual [2] [5]

VI. ESTRUTURA MULTI-NET

No P-Net um mestre que possui mais de uma porta P-Net pode atuar como uma ponte entre dois segmentos, ou seja, no novo segmento pode possuir mais 125 dispositivos. Neste outro segmento outro mestre pode possuir duas portas P-Net e assim por diante. Desta forma torna-se a rede escalável de uma forma eficiente. [2]

O P-Net tem uma estrutura Multi-Net, onde tem-se por objetivo dividir a rede, em varias células para que o controle seja feito de forma mais eficiente.

Quando a rede é dividida em células como por exemplo por seções da planta, é possível desligar aquela célula em questão sem afetar nenhuma outra célula. Caso ocorra um erro em uma célula as outras também não serão prejudicadas.

O P-Net permite o endereçamento direto entre diversos segmentos do barramento, e está especificado como parte do protocolo.

Uma vantagem em se dividir a rede em células é que não há a necessidade de uma estrutura hierárquica para os segmentos dos barramentos e isso facilita muito na expansão da rede.

Na figura 6 podemos ver bem como funciona o multi-net no P-NET, existem diversas células, onde um mestre que possui duas interfaces P-NET faz a intercomunicação entre a célula e o barramento, e dentro da célula tem-se outros dispositivos e até mesmo outros mestre que comporão outras células e por aí adiante.

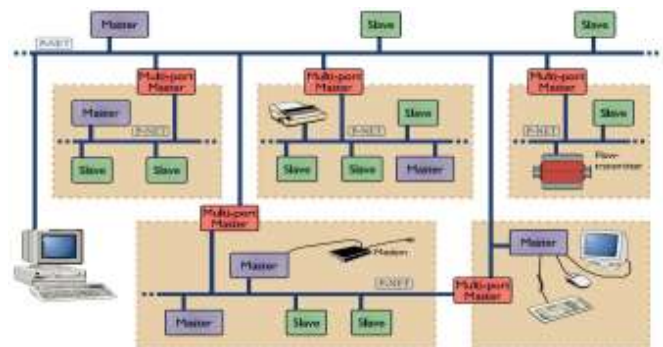


Figura 6: Multi-Net [2]

VII. ARQUITETURA P-NET

Considerando o modelo OSI – Open System Interconnection, o P-Net opera nas camadas 1, 2, 3, 4 e 7. As camadas 3 e 4 são por conta da sua estrutura Multi-Net. [2]

Na primeira camada é onde acontece a parte física da comunicação, onde se determina o tipo de cabeamento, como os ‘0’ e ‘1’ serão representados no barramento, a velocidade que a comunicação será estabelecida e a tensão de nível lógico. Aqui também é onde acontece a codificação de Non-Returning-Zero, NRZ. [2]

Na segunda camada, cuida-se da parte do token para o multi-mestre, forma os quadros que serão enviados juntamente com os endereços de destino e de envio. Nesta camada também ocorre o tratamento dos erros acontece por meio do código da distância de Hamming que é um método de correção de erros um pouco mais complexo e que apresenta bons resultados. [2] [3]

Na terceira camada é o organizador das mensagens, pois ele recebe e envia os quadros conforme o endereço de destino. Esta camada também realiza a conversão de endereçamento caso em algum momento seja necessário. Nesta camada já trata-se a informação como pacote. Talvez uma mensagem precise ser direcionada para outra porta P-Net, e essa camada realiza-o, ou seja ela atua como um gateway. [2]

Para entendermos melhor a quarta camada vamos fazer uma breve abordagem dos canais no P-NET.

Os canais são nada mais nada menos que um conjunto de dados de um determinado sinal do processo (single process signal), assim como uma entrada ou uma saída, que estão estruturadas e definidas de forma lógica.[2] Cada sinal provenientes do processo tem mais informações associadas a ele e não somente o valor escalar que ele representa. Estes dados são sobre temporizadores, mensagens de erro entre outras informações. Um canal é estruturado como 16 registradores armazenadores de informação e essas informações podem ser números inteiros, reais, bytes entre outros. A grande vantagem dos canais é que qualquer mestre presente na rede pode ter acesso ao canal, ou seja, aos dados deste módulo sem ao menos ter um conhecimento interno do sensor. [2] [3]

Cada Registrador do canal tem um endereço lógico que é chamado por SOFTWARE Number. Que pode ser acessado de outros lugares da rede.

Na figura 8 podemos ver uma exemplificação de um canal.

A quarta camada trabalha com uma tarefa que envolve a lista SOFTWARE, que é uma tabela que contém as informações de forma estruturada das variáveis globais do processo. Ela oferece um serviço que pode ler ou escrever dados presentes na lista SOFTWARE. [2]

E por fim na sétima camada, que é a camada de aplicação onde por meio de programas de aplicação se tem acesso as variáveis do processo. Esse acesso às variáveis é realizado por uma solicitação à lista SOFTWARE, pois nela estão armazenadas dados importantes dos nodos da rede.

Na figura abaixo vemos a divisão das camadas presentes no P-NET, 1, 2, 3, 4 e 7:

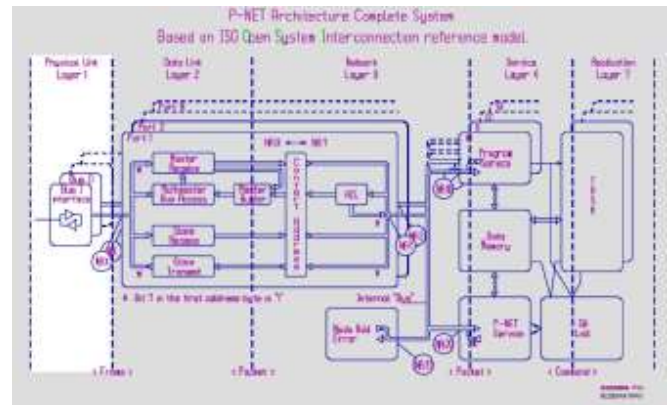


Figura 7: Arquitetura completa P-Net [3]

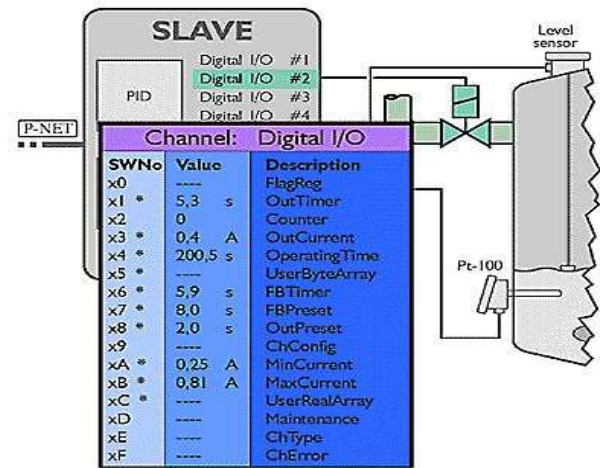


Figura 8: Exemplo Canal [2]

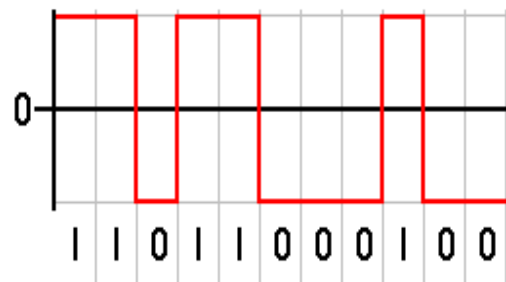


Figura 9: NRZ [6]

VIII. COMPUTADORES NA REDE

No P-Net conecta-se um computador à um segmento da rede como um dos mestres da rede. Este computador desempenha um papel bastante importante na rede. Ele pode ser usado para monitorar e configurar dispositivos.

Pode-se usar também como um sistema SCADA, que proporciona dados para o usuário de toda a planta gerando muitos gráficos.

Deve-se ter um software instalado à ele, que é o *VIGO Fieldbus Management System*. VIGO é um sistema de gerenciamento de barramento criado pelo P-NET, é importante ressaltar que está presente somente para computadores com Microsoft Windows. [7]

Além disso o VIGO desempenha o importante papel de compilar, debugar e programar os mestres da rede através do *Process Pascal*. [3]

Como o P-Net não sofreu muitas alterações em seu layout, a representação não é muito interessante para os padrões atuais, porém esta figura 10 abaixo é de uma planta e alguns valores associados recolhidos pela rede. A figura 11 mostra a janela disponível para o usuário. [7]

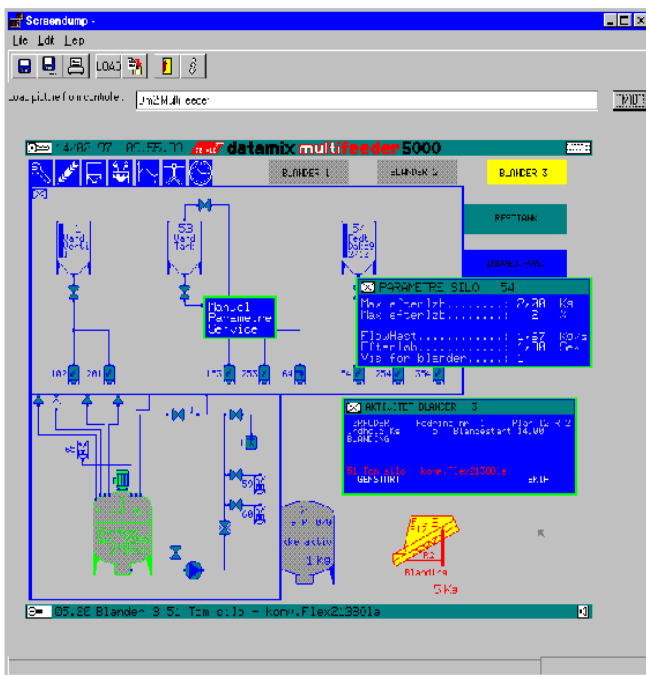


Figura 10: Planta de um processo [9]



Figura 11: Supervisor com acesso aos segmentos e módulos

IX. CONCLUSÃO

O P-Net é um padrão de rede industrial da década de 80 surgido juntamente com outros importantes como FIP e PROFIBUS. Porém por conta da documentação e a falta de informações sobre o protocolo pode ter ideia que o P-Net não é mais tão popular no dia de hoje.

Para o seu tempo apresenta ideias muito interessantes principalmente a passagem do token, a forma como o token é trabalhado na rede torna-a muito mais dinâmica e rápida, que é o necessário para uma rede.

Nas pesquisas realizadas para este trabalho dizia que o P-Net é a única rede com o esquema Multi-Net, onde a rede é dividida em diversas células, além do que a expansão da rede é muito prática.

A distância capaz da rede se comunicar sem a necessidade de repetidores também é algo muito interessante pois 1200m é uma considerável.

Uma desvantagem do P-NET em relação aos outros protocolos disponíveis na atualidade é o tempo de resposta, que no pior dos casos é 20ms. Obviamente que com pacotes menores o tempo diminuirá em consequência, porém mesmo assim não atinge valores como o Profibus e o Profinet em questões de microssegundos. Parte disso também é devida a sua baixa velocidade que opera em no máximo 76800bps, muito abaixo do Gigabit ethernet disponível para o Profinet.

Mesmo que atualmente não haja muitas implementações no mundo, ou até nenhuma, o P-NET continua sendo uma opção mesmo que restrita por não ter atualizado seu protocolo. Em aplicações críticas de processo o P-Net não poderia ser uma opção, existem protocolos mais eficiente. Para se adequar ao Real-time que exige tempos menores de 1ms deve-se restringir os quadros à um número menor de bits para talvez ser uma solução.

X. REFERÊNCIAS

- [1] P-NET – the European Fieldbus - Standard EN50170, Volume 1.
- [2] Thomesse, Jean-Pierre. “Fieldbus Technology in Industrial Automation”. Invited Paper. 2005.
- [3] Process Data. [Online]. Disponível: <https://www.proces-data.com/>
- [4] UCPEL [Online]. Disponível: http://olaria.ucpel.tche.br/autubi/lib/exe/fetch.php?media=padrao_rs485.pdf.
- [5] Tovar, Eduardo Emanuel de Médicis. “Supporting Real-Time Communications with Standard Factory-Floor Networks. July, 1999.
- [6] Wikipedia [Online]. Disponível: <https://en.wikipedia.org/wiki/Non-return-to-zero>.
- [7] Jenkin, Christopher G. P-NET as a European Fieldbus Standard EN50170 vol. 1. 1997.

HART e WirelessHART

Redes industriais

Fellipe Eduardo Gonçalves da Silva
Engenharia de Controle e Automação
Universidade Federal de Santa Catarina, Campus Blumenau
Blumenau, Brasil
fellipegoncalves@gmail.com

Resumo — O sistema industrial necessita de uma rede diferente de uma rede comercial, ou de escritório. A rede industrial necessita integrar todos os dispositivos e equipamentos de uma fábrica, além de garantir um comportamento temporal e uma confiabilidade de seus dados. Nessa perspectiva surgiu vários métodos para a integração total, desde o chão de fábrica ao nível de planejamento. Este trabalho tem o objetivo de mostrar um dos métodos bastante utilizado, o protocolo HART, e o método sem fio, Wireless HART.

Palavras Chave – Redes industriais, HART, WirelessHART.

I. INTRODUÇÃO

As redes industriais surgiram com o objetivo de integrar todos os níveis do processo industrial, desde o chão de fábrica até o nível de planejamento.

Dentre essa necessidade foi criado diversos tipos de rede para implementar no setor industrial. E nesse processo várias empresas foi criando seu padrão para sua aplicação ou para melhorar algum requisito da rede.

O sistema industrial procura sempre o melhor método para implementar no seu setor, e dentro os diversos protocolos se destaca o HART, que uniu o meio físico que já era utilizado no setor com a integração da rede necessitada, assim, garantindo um custo inferior aos demais que necessitavam de um custo inicial elevado.

O protocolo HART introduziu a comunicação digital em um processo que acontecia analogicamente sem modificar o sistema original, por isso o sucesso que o protocolo possui até hoje. Assim poderemos analisar como é esse protocolo detalhadamente, além de ver sua evolução para o sistema de redes sem fio.

II. HART

A. Introdução

A tecnologia HART surgiu em 1980, com base na mesma tecnologia que trouxe o identificador de chamadas para a telefonia analógica. O protocolo HART (Highway Addressable Remote Transducer, ou Via de Dados Endereçável por Transdutor Remoto) é um padrão para envio e recebimento de informações digitais por cabos analógicos[1].

HART é um protocolo de comunicação que permite o acesso aos dados entre instrumentos de campo inteligentes e sistemas host. HART foi o primeiro protocolo de comunicação bidirecional que não afetava o sinal analógico de controle.

A tecnologia HART é o maior de todos os protocolos de comunicação, instalado em mais de 30 milhões de dispositivos em todo mundo[2].

O protocolo HART tem como características ter projetos simples, ser de fácil operação e manutenção, é compatível com a instrumentação analógica, 500ms de tempo de resposta (com até duas transações)[9].

B. Funcionamento dos dispositivos do protocolo HART

A comunicação de campo usada durante muito tempo pelos equipamentos de controle de processos tem sido o sinal analógico de corrente, na maioria das aplicações, variando dentro da faixa de 4-20mA (miliampére).

Em um circuito o sinal do transmissor varia a corrente que passa por ele de acordo com o processo de medição, o controlador identifica a variação de corrente através de um resistor sensor de corrente. Podemos ver o modelo pela Fig. 1[3].

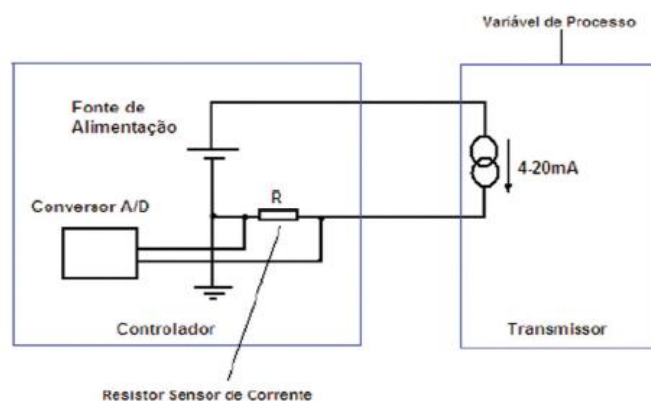


Figura 1. Modelo 4-20mA

No modelo HART (Fig. 2) ambas as terminações do loop possuem um modem e um amplificador de recepção, sendo que possuem alta impedância para não carregar o loop de corrente. O transmissor possui uma fonte de corrente AC (corrente

alternada) e o controlador uma fonte de tensão AC. A chave em série com a fonte de tensão do controlador HART fica aberta normalmente. O sinal analógico não é afetado, uma vez que os componentes adicionados são acoplados em AC.

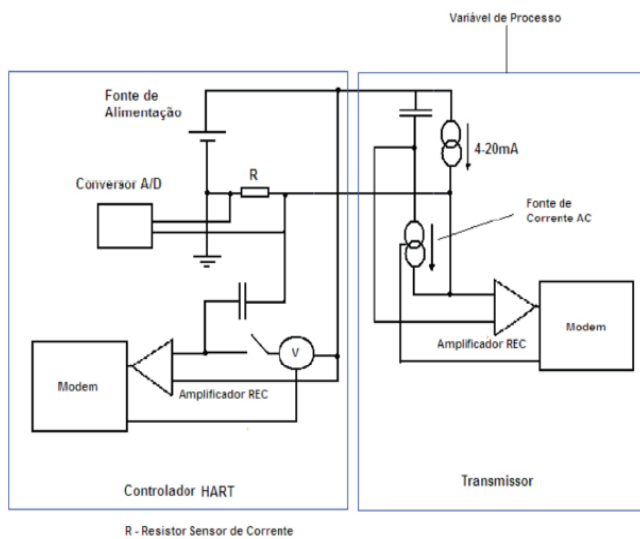


Figura 2. Modelo do circuito HART

Para enviar uma mensagem, o transmissor, ligando sua fonte de corrente, irá sobrepor o sinal de corrente de 1mA pico-a-pico com uma frequência alta em relação ao sinal analógico da corrente de saída. O resistor no controlador converterá o sinal de tensão e será amplificado no receptor chegando até o modem (demodulador do controlador). Devemos notar que o mestre transmite como fonte de tensão e o escravo como fonte de corrente.

O protocolo HART utiliza o padrão Bell 202, de chaveamento por deslocamento de frequência (Frequency Shift Keying – FSK), para sobrepor os sinais de comunicação digital ao de 4-20mA. A frequência de 1200 Hz (hertz) representa o 1 binário e a frequência de 2200 Hz representa o 0 binário. A mensagem se propaga a uma taxa de 1200 bits por segundo. Os sinais analógicos normalmente são usados com frequência entre 0 e 10 Hz, uma frequência bem baixa em relação a frequência usada pelo protocolo HART, de tal forma que não há interferência entre elas. O sinal FSK possibilita a comunicação bidirecional, o que torna possível a transmissão e recepção de informações adicionais, além da normal do meio. O modelo do sinal em FSK para o protocolo HART pode ser visto na Fig. 3[3].

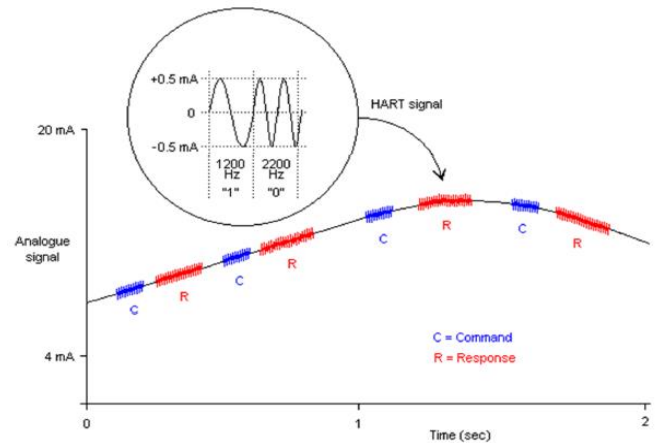


Figura 3. Sinal em FSK

C. Modelo do protocolo HART segundo o modelo OSI

O protocolo HART possui apenas as camadas 1, 2 e 7 do modelo OSI, especificado na Tabela 1.

Tabela 1. Relação do modelo HART e o modelo OSI

OSI Layers	HART layers
Application	HART commands
Presentation	
Session	
Transport	
Network	
Data Link	HART protocol rules
Physical Layer	Bell 202

• Camada Física

Os dispositivos HART possuem o processo de instalação igual ao utilizado para os processos de 4-20mA convencional. O cabeamento deve ser par trançado blindado, aterrado em um único ponto, além de garantir uma alimentação de energia elétrica de acordo com a especificação dos equipamentos utilizados.

O cabo mais utilizado é o par trançado individualmente blindado, podendo ser único ou múltiplos pares. O diâmetro mínimo do condutor é 0,51mm (milímetros) (AWG 24) para distâncias de até 1500m e 0,81mm (AWG 20) para distâncias longas[3][4].

As características elétricas do cabo e o número de dispositivos conectados podem interferir no comprimento máximo permitido do cabo, é possível ver essa relação na Tabela 2.

Tabela 2. Relação de dispositivos e a capacitância do cabo

Nº de Dispositivos em Rede	Capacitância do Cabo – pF/pé (pF/m)			
	20 pF/pé (65 pF/m)	30 pF/pé (95 pF/m)	50 pF/pé (160 pF/m)	70 pF/pé (225 pF/m)
1	9000 pés (2769 m)	6500 pés (2000 m)	4200 pés (1792 m)	3200 pés (985 m)
5	8000 pés (2462 m)	5900 pés (1815 m)	3700 pés (1138 m)	2900 pés (892 m)
10	7000 pés (2154 m)	5200 pés (1600 m)	3300 pés (1015 m)	2500 pés (769 m)
15	6000 pés (1846 m)	4600 pés (1415 m)	2900 pés (892 m)	2300 pés (708 m)

Em relação a topologia utilizada no protocolo HART, pode ser operado em dois modos, ponto-a-ponto e multiponto (com até dois mestres, sendo o segundo de configuração e calibração, instrumento de campo, como notebook, equipamento móvel), a diferença entre os métodos é vista na Fig. 4 e o esquema de multipontos com o segundo mestre na Fig. 5.

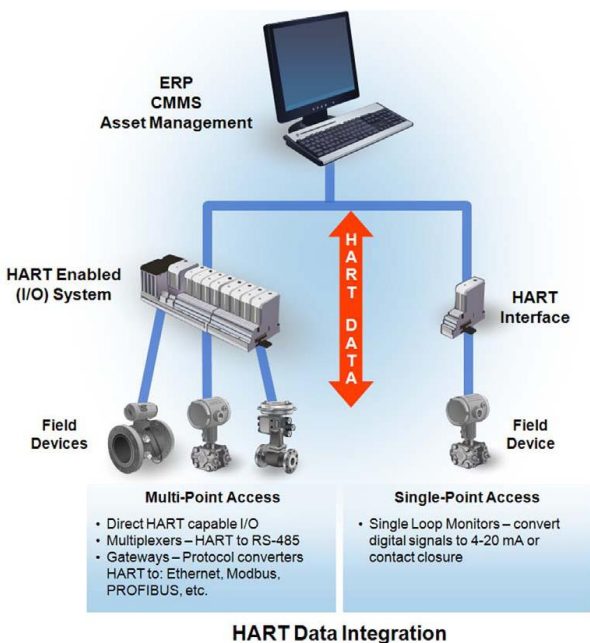


Figura 4. Topologias do protocolo HART

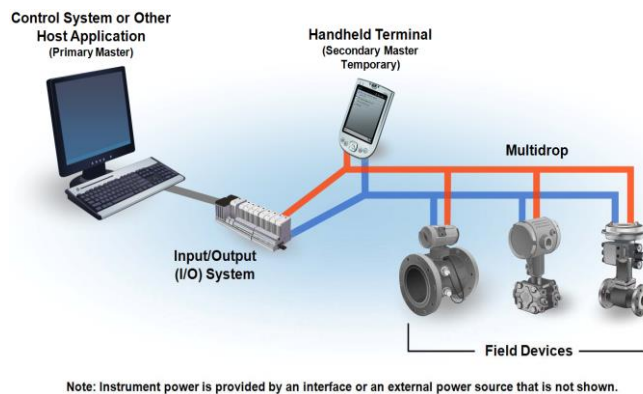


Figura 5. HART em multipontos

• Camada de Enlace

O protocolo HART opera de acordo com o padrão Mestre-Escravo, onde o escravo só transmite uma mensagem se o mestre requisitar. Toda comunicação é iniciada pelo mestre e o escravo só vai responder se houver um pedido na linha. O protocolo pode usar diferentes modos de comunicação, sendo ele o mestre/escravo, modo burst ou broadcast, e o modo multidrop (multipontos).

O modo mestre/escravo é o mais comum, o qual permite que a informação digital oriunda do escravo seja atualizada duas vezes por segundo no mestre. Esquema do processo na Fig. 6.

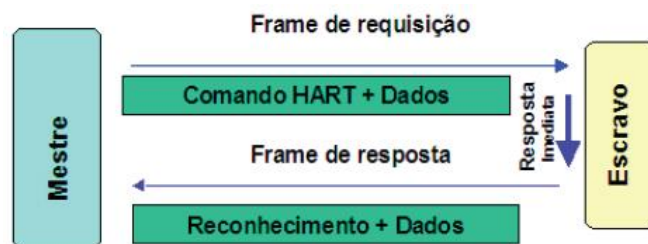
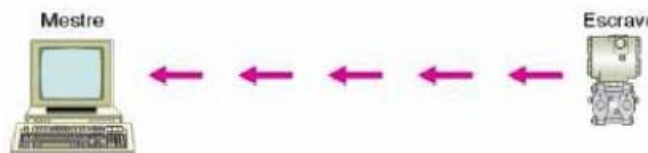


Figura 6. Modo de comunicação mestre/escravo

O modo burst permite que um único escravo publique continuamente uma mensagem de resposta padrão HART. Esse modo evita que o mestre fique repetindo um comando de solicitação para atualizar a informação da variável de processo. O modo burst só pode ser usado quando existe um único equipamento escravo na rede. O esquema desse modo é visto na Fig. 7.



Modo de Comunicação Digital Puro, Resposta continua de uma variável padrão selecionada como a PV. As janelas existentes entre cada resposta permitem ao Mestre mudar o comando ou o modo de comunicação. 3 a 4 atualizações por segundo tipicamente.

Figura 7. Modo de comunicação burst

O modo multidrop é usado na topologia multipontos, onde a rede suporta até 15 instrumentos de campo. O grande problema desse modo é o tempo de resposta, onde o tempo de ciclo de cada equipamento leva em média meio segundo, então a leitura de 15 equipamentos pode levar um tempo bastante desagradável dependendo do processo. O esquema desse modo pode ser visto na Fig. 9.

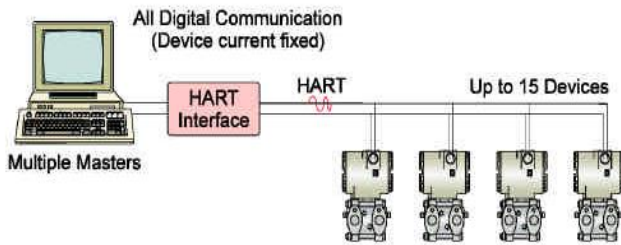


Figura 8. Modo multidrop

O frame padrão do HART é representado pela Fig. 9.

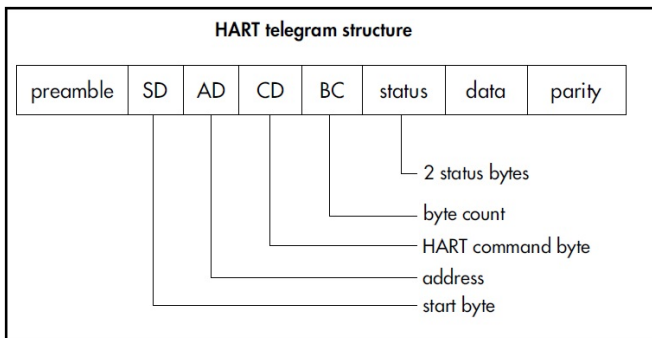


Figura 9. Frame padrão do protocolo HART

Onde PR é o preâmbulo, pode ser 3 ou mais bytes de sincronismo dos sinais da mensagem.

SD é o byte que indica quem está enviando a mensagem, se é mestre ou escravo (além de informar se está no modo burst ou escravo normal), e informa se está no formato short (curto) ou long (longo).

AD é o campo de endereço, onde com formato de frame curto possuirá apenas 1 byte com 1 bit servindo para distinção dos 2 mestres e 1 bit para indicar o modo burst. São usados 4 bits para endereçar (0 à 15) o campo dos dispositivos. No formato longo o frame possui 5 bytes, e o campo de dispositivos é definido utilizando 38 bits. A diferença dos frames pode ser vista na Fig. 10[8].

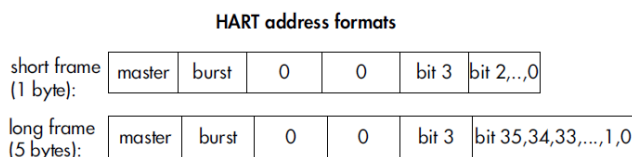


Figura 10. Formato de endereço do HART

Os comandos do campo HART podem ser divididos em classes (universal, comum, de acordo com o fabricante) e o

campo CD determina o tipo, o que vai depender da camada de aplicação.

BC indica o comprimento da mensagem, no HART o comprimento máximo é 25 bytes.

STATUS – são usados dois bytes para indicar a condição do equipamento.

No campo DATA é onde está localizado os dados transmitidos.

No campo PARITY é o byte de checksum que contém a paridade longitudinal de todos os bytes do pacote.

- *Campo de Aplicação*

A camada de aplicação determina o que pode ser transmitido entre os dispositivos, além de especificar os comandos a serem utilizados na transferência de dados. Os comandos podem ser universais ou podem ser específicos para cada dispositivo. Podemos ver uma forma simplificada dos comandos na Tabela 3.

Tabela 3. Comandos básicos do protocolo HART

Comandos universais	Comandos específicos do dispositivo
Leitura de variáveis	Funções específicas do modelo
Mudança de limite inferior e superior	Opções especiais de calibração
Ajuste de zero e span	Iniciar, parar e resetar totalizador
Inicia auto teste	Selecionar variável primária
Número de série	Habilitar PID, mudar Set Point
Valores de constantes de tempo	Ajustar parâmetros de sintonia

III. WIRELESSHART

O setor industrial está cada vez mais interessado no uso da comunicação sem fio, e o que as empresas buscam aliar a segurança do tráfego dos dados à flexibilidade de acesso a eles.

Em uma rede industrial sem fio os principais objetivos são:

- Dados não podem ser perdidos ou alterados durante a comunicação;
- Dados não podem ser acessados sem autorização;
- A rede deve ser compatível ao ambiente, mesmo com pouca infraestrutura;
- Fácil instalação e baixo custo;
- Atraso mínimo na transmissão da informação (tempo de latência).

Nos últimos anos a tecnologia de redes sem fio obteve grandes avanços tecnológicos, o que proporcionaram segurança, confiabilidade, estabilidade, auto-organização, baixo consumo e baterias de longa vida.

As principais redes sem fio industrial são ISA SP100 e a WirelessHART. Iremos destacar a WirelessHART.

Durante os anos, o protocolo HART precisou se modificar para atender as novas áreas tecnológicas. A evolução do HART chegou até o HART 7, onde o WirelessHART está inserido. A Fig. 11 mostra graficamente a evolução do protocolo HART[4].

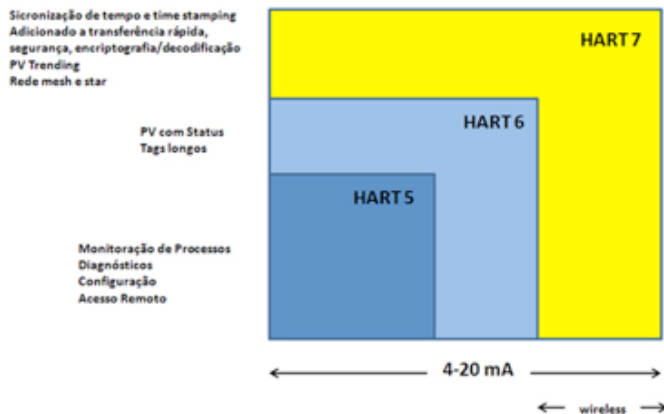


Figura 11. Evolução do protocolo HART

O HART 7 possui as seguintes características:

- Comandos especiais para Wireless;
- Equipamentos de campo podem ser alimentados por uma bateria de longa vida ou energia solar;
- Aplicação de monitoração, controle de processos que não precise de alta taxa de atualização.

WirelessHART é um protocolo de comunicação de rede em malha sem fio, onde aplica os recursos sem fio no protocolo HART, além de garantir uma compatibilidade com os dispositivos e comandos do HART existentes (interoperabilidade)[3][10].

A rede WirelessHART possui três elementos principais:

- Instrumentos de campo sem fio – são instrumentos de campo WirelessHART puro, atuam na rede sem fio diretamente;
- Gateways – realizam a comunicação entre os instrumentos e os hosts de outra rede de comunicação existente ou de um backbone de alta velocidade;
- Gerente de rede – responsável por configurar a rede, programar as comunicações entre os instrumentos, gerenciar rotas de mensagens e monitorar a situação da rede.

Podemos ver a estrutura da rede WirelessHART na Fig. 12 e na Fig. 13.

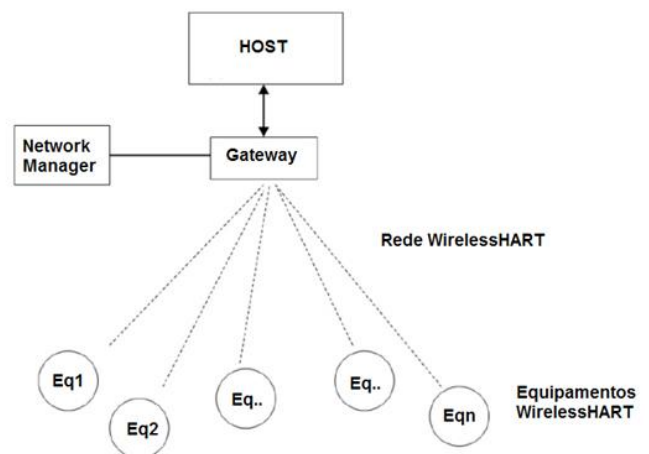


Figura 12. Estrutura da rede WirelessHART(1)

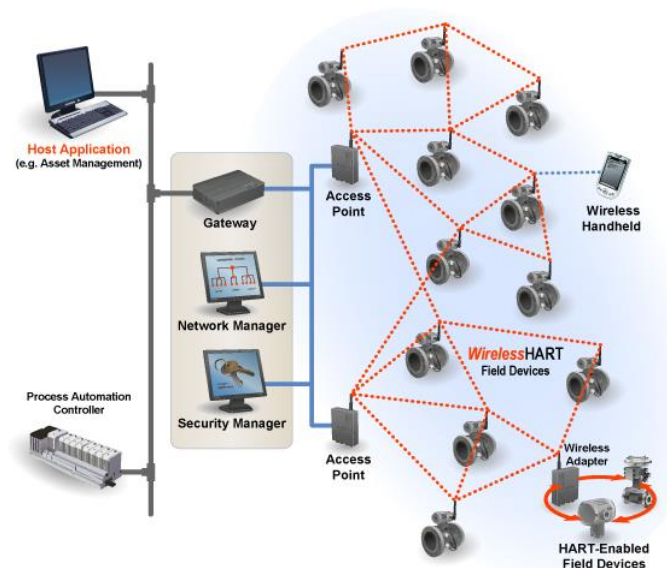


Figura 13. Estrutura da rede WirelessHART(2)

A rede WirelessHART é compatível com o padrão 802.15.4 do IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos), operando na banda ISM (Industrial, Científica e Médica) de 2,4 GHz. Na rede é utilizada a tecnologia de sequência direta de espalhamento espectral (DSSS) e salto de frequência (FHSS) garantindo maior segurança e confiabilidade dos dados. As comunicações TDMA (Time Division Multiple Access, ou Acesso Múltiplo por Divisão de Tempo) são utilizadas entre os dispositivos na rede[4][3].

A rede WirelessHART utiliza a rede mesh (malha) o que garante que todo instrumento pode funcionar como um roteador de mensagem de outros equipamentos. Assim, não precisa necessariamente uma comunicação direta com o gateway, pode ser encontrado um instrumento mais próximo. Com isso é possível ampliar o alcance da rede e proporciona diferentes rotas de comunicação para aumentar a confiabilidade. Esse tipo de rede facilita a mudança do local de instalação de instrumentos ou a adição de algum deles.

Para garantir a melhor rota, o gerente de rede determina a partir dos fatores de latência, eficiência e confiabilidade. Se a mensagem não consegue alcançar seu destino por um caminho, ela é automaticamente redirecionada para um outro caminho, sem perda de dados.

Outros componentes do WirelessHART que devemos mencionar são o processo, o repetidor, o gerente de segurança, o adaptador e o terminal portátil.

O processo são as instrumentações habilitadas para HART.

O gerente de segurança determina quem poderá “ler” o dado transmitido ou recebido, além de manter a lista de dispositivos autorizados a entrar na rede.

O repetidor apenas estende o alcance da rede, não possui conexão própria.

O adaptador é um dispositivo que conecta a um instrumento habilitado para HART e o permite transmitir ou receptor em WirelessHART.

O terminal portátil pode ser uma unidade do HART padrão de configuração por FSK ou ser um suporte sem fio que pode introduzir um instrumento de campo.

A rede WirelessHART utiliza as camadas 1, 2, 3, 4 e 7 do modelo OSI. A relação da rede com o modelo OSI pode ser vista na Fig. 14 e ainda ver a comparação com o modelo HART padrão[4].

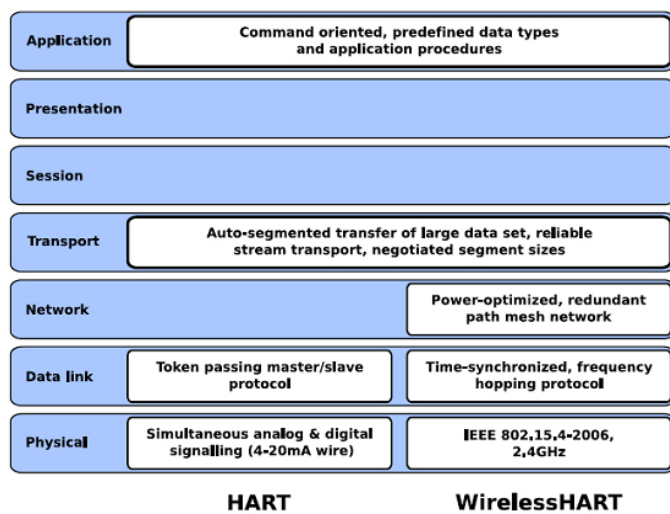


Figura 14. Relação dos protocolos HART com o modelo OSI

Podemos destacar que a rede WirelessHART é bastante segura, tanto a camada MAC quanto a camada de rede possuem serviços de segurança, como CRC (Cyclic Redundancy Check) e MIC (Message Integrity Code)[4].

IV. CONCLUSÃO

O protocolo HART foi introduzido para evitar a monitoração manual das variáveis industriais. O protocolo permitiu em uma instalação já existente uma forma mais rápida e eficaz da monitoração do processo, tornando o risco menor e reduzindo o tempo de visualização do estado das máquinas.

A evolução da tecnologia proporcionou o HART 7, oferecendo a rede sem fio do HART, Wireless HART. A rede proporciona uma rede sem fio aberta, no qual, possui uma alta confiabilidade, custo baixo, e garante uma compatibilidade com os equipamentos do HART padrão. A rede sem fio proporciona um avanço comercial, proporcionando um aumento da produtividade, e uma redução da variabilidade dos processos, além de reduzir o tempo de resposta. A rede wireless ampliou a área da rede e reduziu o tempo de cada ciclo. A coisa mais importante do Wireless HART é de o sistema sem fio ser compatível com o sistema padrão, o que permite a utilização dos equipamentos antigos.

Assim, entende-se que o protocolo HART é bastante popular por oferecer um serviço barato e eficiente, o que é requerido das redes industriais, e ao avançar as tecnologias, vimos que há uma tendência do sistema se adaptar para melhorar o funcionamento. Porém, o sistema industrial necessita de uma rede padrão que não a deixa restrita para equipamentos da sua empresa. O HART não permite uma interoperabilidade total, ele só aceita os equipamentos da sua plataforma. No futuro, é possível que o protocolo seja ultrapassado por algum padrão que permita uma implementação com diferentes plataformas e não deixa restrita a apenas um fabricante, ou empresa.

V. REFERÊNCIAS

- [1] Filho, Constantino S., “Introdução ao Protocolo HART”.
- [2] HART Communications Foundation, <http://www.hartcomm.org/index.html>.
- [3] Cassiolato, César, “O protocolo digital HART”.
- [4] www.smar.com.br/
- [5] Comer, Douglas E. Redes de computadores e internet. Porto Alegre: BOOKMAN, 2007.
- [6] Brandão, Dennis e Pantoni, Rodrigo P. e Bonifácio, Tatiana G., “Análise tecnológica de redes industriais wireless”.
- [7] Lugli, Alexandre B. e Sobrinho, Darlan G., “Tecnologias wireless para automação industrial: wireless_hart, bluetooth, wisa, wi-fi, zigbee e SP-100”.
- [8] SAMSON, HART Communication – Technical Information” – Capítulo 4.
- [9] <http://www.mecatronicaatual.com.br/educacao/1467-tecnologia-hart-na-industria-parte-1-estrutura-do-protocolo>
- [10] <http://www.automacaoindustrial.info/o-protocolo-wirelesshart-parte-1/>
- [11] <http://www.automacaoindustrial.info/o-protocolo-wirelesshart-parte-2/>

Rede Sinec

Comunicação Industrial

Diogo Luiz Demarchi

Engenharia de Controle e Automação
UFSC – Universidade Federal de Santa Catarina
Blumenau, Brasil
diogo.demarchi@grad.ufsc.br

Resumo – Este artigo contempla a rede de comunicação Sinec e suas vertentes, bem como sua origem, propósito de existência e seu estado atual de desenvolvimento nas áreas onde é utilizada. Também há comparação com outras redes similares e as diferenças com suas versões. Para um melhor entendimento do assunto abordado, também é feito uma breve explicação do início das redes e qual a importância de se ter diferentes tipos delas.

Palavras-chave — rede; indústria; comunicação; sinec; siemens; interface; protocolo;

I. INTRODUÇÃO

Para se falar em redes de comunicação, é necessário pensar, antes de tudo, em como foi estabelecida toda a computação moderna. Com a chegada do pensamento digital e a busca incessante de conquistar novos horizontes informatizados, o ser humano sempre tentou achar novas formas melhores e otimizadas para se pensar a nível lógico. Primeiro, vieram os computadores mais arcaicos, que mal conseguiam “pensar” em bits 0 e 1. Hoje, não muito tempo depois, vemos uma revolução avassaladora, com computadores ultrarrápidos e que conseguem realizar processos a taxas absurdas. Entretanto, nada disso adianta se não for possível realizar a comunicação entre dois ou mais equipamentos. É neste ponto que entram as redes de comunicação, tanto domésticas, para uso pessoal, ou industriais, para interligação entre diferentes equipamentos dentro do processo fabril. Neste artigo, particularmente, será tratada mais a questão industrial das redes de comunicação, considerando, mais especificamente, a rede Sinec.

A rede Sinec, em sua essência, consiste em um conjunto de soluções para a comunicação em uma rede de automação. Foi proposto pela Siemens e consegue abranger uma grande gama dos níveis hierárquicos da produção de uma indústria ou processo que necessite do uso da rede Sinec.

Existem soluções para redes Ethernet, com o Sinec H1, para MAP Broadband e Carrierband, com o Sinec H2, em FieldBus por meio do Sinec L2 e também para redes proprietárias de baixo custo, utilizando o Sinec L1. O sistema também conta com equipamentos para a realização da interface da rede no processo como, por exemplo, CNC, CLP, outros controladores, etc. Outro ponto importante é que é possível fazer a interconexão com outras redes.

II. SINEC

A. Base Geral

A rede Sinec é um protocolo de comunicação industrial criado pela empresa Siemens para atender procedimentos de automação e controle de processos industriais. Sua principal característica é ter uma largura de banda elevada e fornecer o papel da camada de transporte equivalente ao modelo OSI. Suas características são bastante propícias a um tráfego elevado de dados.

O principal foco deste tipo de protocolo é a comunicação aberta entre diversos tipos de unidades de automação. Pode-se tomar como exemplo de uso do Sinec sistemas como Sicomp, Sinumerik e Simatic (Fig. 1).

Ele também é utilizado no LHC (Grande Colisor de Hádrons) (Fig. 2), bem como na organização detentora dele, a CERN (Organização Européia de Pesquisa Nuclear) e, por isso, tem grande importância para o desenvolvimento científico mundial.

Algo importante a ser citado, é que os programas individuais têm acesso transparente ao sistema de barramento de transporte usado, que pode ser PROFIBUS ou Ethernet. Isso significa que o transporte não é ligado diretamente à aplicação.

O desenvolvimento dos sistemas de barramento estão bem avançados até o presente momento. Pelo que se pode observar pelas tendências mundiais, sistemas abertos têm grande



Figura 1 - Simatic



Figura 2 – LHC (Large Hadron Collider)

potencial para conseguir cada vez uma fatia maior de mercado se comparado à soluções fechadas.

A rede de comunicação Sinec possui algumas vertentes e este artigo tentará abordar de forma clara e simples cada uma delas. Existem vários tipos de redes Sinec, porém, neste artigo, serão abordadas apenas as principais que são a Sinec H1, H2, L1 e L2.

III. SINEC H1

A primeira vertente da rede da Siemens a ser estudada aqui é a rede Sinec H1. Ela usa o padrão IEEE 802.3 (Ethernet). O sinec H1 é capaz de fornecer dois tipos de serviço de comunicação, cada um com seu protocolo específico. São eles:

- Serviço de transferência de dados entre pares (Peer Data Transfer), que é baseado na camada 4 (camada de transporte) e é usado geralmente para uma eficiente troca de dados entre um nó de controlador para outro.
- Serviço de função de tecnologia (Technology Function ou TF) é baseado na camada 7 (camada de aplicação) e é tipicamente utilizado entre um sistema supervisor e um nó de um CLP (controlador lógico programável), do qual está obtendo ou escrevendo dados.

Quando um link lógico entre dos nós H1 se dá através da camada 4 (transporte), o modo peer data transfer toma o papel principal na comunicação. Diz-se, assim, que o link de comunicação está em conexão, porém pode também ser referido como conexão da camada de transporte. Esta ligação funciona normalmente em entre dois CLPs em uma rede de sistemas.

No caso da ocorrência de uma ligação lógica entre dois nós H1 através da camada de aplicação (camada 7), quem toma o papel da comunicação é o modo technology function. À isso dá-se o nome de associação. Esta ligação tipicamente ocorre entre um host de supervisão e um CLP.

A. Peer Data Transfer

Para criar um sistema com peer data transfer, é necessário configurar um módulo H1 em cada CLP do sistema e implementar um código apropriado em Relay Ladder Logic para o CLP ativo fazendo, assim, com que o sistema realize o serviço necessário.

Quando um CLP está se comunicando por meio de um módulo de comunicação Sinec H1 no modo de transferência de dado entre pares, ele pode estar funcionando como cliente (parceiro ativo ou inicializador) ou como servidor (parceiro passivo).

Com o serviço de parceiros, um CLP é designado como estação ativa enquanto o outro é designado como estação passiva.

- Ativo – a estação ativa requer uma lógica de retransmissão em ladder (Relay Ladder Logic ou RLL) para iniciar ou autorizar a conexão e a transferência de dados (trabalho).
- Passivo – a estação passiva não requer o uso de qualquer código em ladder do usuário para iniciar a transmissão de dados. Em vez disso, é configurada para estar pronta permanentemente para a troca de dados.

Neste modo de operação, é possível observar vários tipos de serviço para a transferência de dados. Exemplos de tipos de serviço são send; receive; write active; write passive; read active; read passive, cada um com sua respectiva função.

Para que haja uma comunicação efetiva, este modo faz uso de um sistema chamado “handshake” (aperto de mão). Ele consiste em pegar o primeiro bit das três primeiras palavras designadas no módulo de comunicação e no CLP e identificar a que tipo de serviço corresponde.

O bit de começo de serviço (Job Start) é selecionado pelo usuário em uma das palavras de saída disponíveis (WY) designadas ao módulo H1 e usada no código RLL para acionar o serviço de transferência de dados. O módulo detecta a borda de subida do Job Start Bit para começar o serviço.

Já o bit de de serviço ativo (Job Active) corresponde ao bit na palavra de entrada (WX), definido pelo módulo H1 para sinalizar que um pedido de transferência de dados está ocorrendo. Quando há uma borda de descida, o CLP sabe que o serviço está completo e pode esperar por um outro bit para início de serviço.

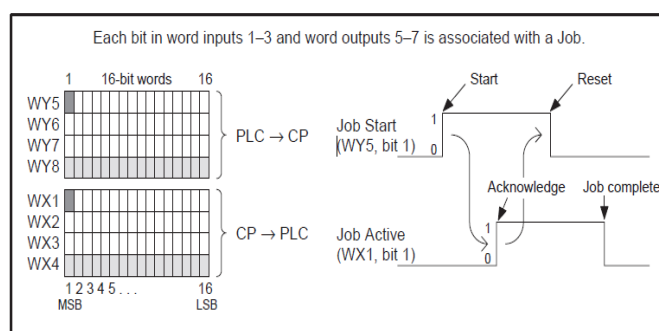


Figura 3 – Comunicação com Handshake entre CLP e módulo H1

B. Technology Function

O serviço de technology function usualmente é usado entre um host supervisor e um CLP para aquisição de dados e controle. Em uma rede H1 típica, um ou mais hosts podem ser conectados a um ou mais CLPs utilizando de módulos H1 CP.

Quando o CLP está se comunicando no mesmo módulo de comunicação H1, porém no modo de technology function, ele sempre estará no modo de servidor (passivo).

O TF possui basicamente três métodos de endereçamento de dados contidos no CLP:

- Endereçamento de Nome Codificado – Este método consiste em um conjunto de nomes predefinidos codificados que são usados para endereçar os dados contidos no CLP. Nenhuma ação do usuário é necessitada para a realização deste tipo de endereçamento.
- Variáveis nomeáveis – É possível definir nomes de variáveis relacionados à aplicação utilizada. Quando define-se uma variável nomeável, especifica-se os dados do CLP associado a tal variável.
- Variáveis sem restrições – É um programa hospedeiro que pode emitir um pedido de leitura ou escrita em endereços especificados no formato binário. Não é necessária ação do usuário no CLP ou no próprio módulo H1 para que o host faça o uso do endereçamento por meio de variáveis sem restrições.

TABELA 1

Modo de Operação	Serviço TF	Serviço de Pares
Cliente		X
Servidor	X	X

Modos de operação do módulo H1

IV. SINEC H2

Ao contrário do tipo de rede H1, que faz sua comunicação utilizando o padrão Ethernet, a Sinec H2 é compatível com o padrão MAP (Manufacturing Automation Protocol) Broadband e CarrierBand.

Este tipo de Sinec atua nos níveis 1 a 4 do modelo de referência OSI.

A. MAP

O Manufacturing Automation Protocol foi uma iniciativa da General Motors na década de 1980, tendo como objetivo criar uma rede para suprir as necessidades da manufatura que conseguisse realizar uma comunicação compatível com padrões de outros fabricantes de, por exemplo, controladores lógico programáveis. A ideia original foi baseada no modelo de referência OSI.

Também foram criadas vertentes do projeto MAP. Por exemplo, quando a variável tempo é crucial para o bom funcionamento do sistema, existe a versão MAP/EPA (Enhanced Performance Architecture). Este modelo apresenta duas pilhas de camada, sendo uma completa com todas as 7 camadas do modelo OSI e outra simplificada, apenas com as camadas 1, 2 e 7.

Uma outra versão mais simplificada do MAP também foi criada. Esta, por sua vez, utiliza apenas as camadas 1,2 e 7 do modelo de referência.

Os módulos MAP possuem duas portas: uma em broadband e outra em carrier-band. A porta faz a comunicação em broadband usa interface IEEE 802.4G para um modem de banda larga externo. A outra, em carrier-band, é dirigida por um modem externo que realiza a transmissão em banda portadora. Redes redundantes são possíveis utilizando vários módulos.

V. SINEC L1

É a vertente que utiliza um sistema fieldbus proprietário da Siemens. Consiste em uma rede local que permite que controladores lógicos consigam se comunicar em toda a faixa que abrange o barramento.

O Sinec L1 opera com o princípio de mestre-escravo e pode ser utilizado com os modos tanto de mestre como de escravo. A taxa de transmissão normal para este método é de 9600 Bit/s.

Quando um dispositivo é utilizado em modo mestre, coordena e monitora todo o tráfego de dados na sua rede, bem como o roteamento que está ocorrendo e fazendo o monitoramento das funções programáveis dentro do barramento. Pode-se ter apenas um mestre por vez neste tipo de rede. Já no modo escravo, ele apenas fica esperando por ações do mestre e envia dados que eventualmente forem requisitados.

Existem dois tipos de transmissão de pacotes de dados em uma rede Sinec L1: de um nó para outro ou de um nó simultaneamente para todos os outros nós conectados à mesma rede (broadcasting). Os dados contidos nos pacotes de dados podem ser informação de estados das entradas, das saídas, flags, etc

Para se estabelecer a conexão entre um mestre e um escravo, é necessário, primeiramente, que haja uma conexão estabelecida para ter certeza de que os dados cheguem no parceiro correto.

O estabelecimento da conexão sempre é introduzido com um “break”, que não é nada mais do que um sinal zerado. Isto faz com que qualquer conexão previamente existente seja interrompida. Depois do break, o mestre envia o número correspondente ao escravo no qual se deseja a comunicação e espera um ACK (acknowledgement), este que contém o número do próprio escravo para garantir que o ACK veio do escravo correto.

Quando a conexão é estabelecida entre o mestre e o escravo, ela tem uso exclusivo do barramento. Caso o escravo mande dados de volta para o mestre, também é necessário que se faça o uso do ACK. Depois disso, uma nova conexão só pode ser estabelecida quando haja outro sinal de break.

Quem controla a troca de dados na rede, neste caso, é o protocolo de barramento. Além do puro uso de dados, há também um controle e detecção de erro no barramento, o que não afeta diretamente o usuário mas pode ser avaliada pelo mesmo.

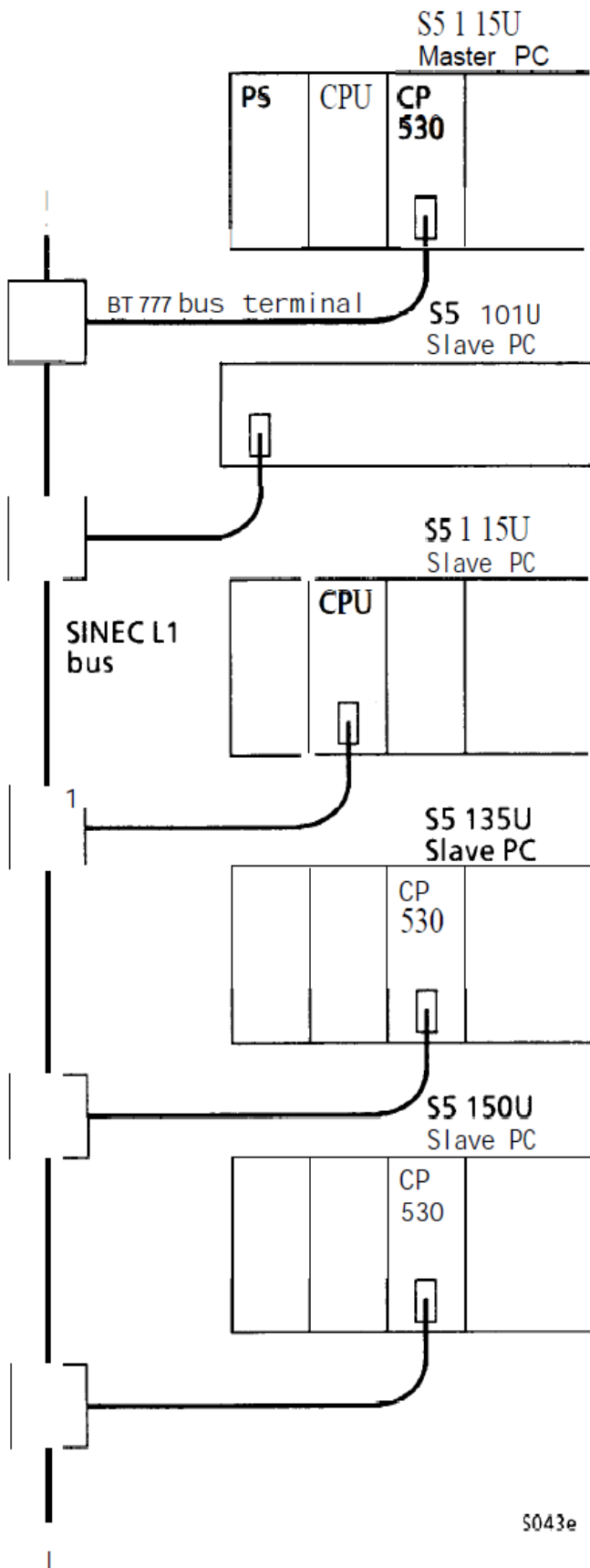


Figura 4 – Barramento típico de Sinec L1

VI. SINEC L2

É uma rede em FieldBus que é compatível com a norma alemã Profibus. Nela, podemos observar três versões distintas: DP, que faz a interligação entre equipamentos remotos de entrada e saída, bem como CLPs, sensores e atuadores; FMS, para interligação entre equipamentos inteligentes autônomos, como CNCs, CLPs, PCs, etc.; e PA, para interligação de equipamentos de instrumentação.

A. Profibus

O Profibus (Process Field Bus) é o processo alemão e padrão de barramento de campo que é definido no DIN 19245. Este padrão designa características funcionais, elétricas e mecânicas em uma rede de campo bit-serial. O propósito desta padronização é ter a capacidade de interligar por meio de uma rede dispositivos de vários fabricantes sem a necessidade do uso de adaptadores muito caros.

Deste modo, é possível que se misture e combine componentes de vários fabricantes e que todos eles se comuniquem por meio da rede Sinec L2, desde que todos os componentes sigam o padrão Profibus.

B. Métodos de acesso

O sistema L2 possui dois tipos de estações com diferentes direitos de acesso: estações ativas e estações passivas.

- Estações ativas – possuem permissão para transmitir dados para outras estações e também são permitidas de requisitar que dados sejam enviados de outras estações.
- Estações passivas – possuem permissão de fazer a troca de dados com alguma estação ativa somente após a ação ser requisitada à isso pela estação ativa.

O que define uma estação de ser passiva ou ativa depende apenas da unidade e qual seu propósito na rede. Unidades de campo simples, como unidades de controle de motores são passivas naturalmente. Por outro lado, unidades inteligentes, como CLPs, são necessariamente estações ativas, pois precisam frequentemente precisam fazer requisições para que dispositivos de campo enviem informações sobre seu estado para o próprio CLP para que, assim seja feito o controle. Do mesmo modo, precisa também poder enviar dados livremente pois, após receber as informações do dispositivo de campo, é feito uma leitura e, conseqüentemente, poderá haver ou não uma ação a ser realizada sobre o mesmo.

Existem unidades também que podem ser configuradas para serem tanto passivas como ativas, depende apenas da finalidade que elas vão ter na rede.

C. Sinec L2-DP

O Sinec L2-DP é uma das três vertentes existentes da versão L2 do Sinec. Ele foi desenvolvido para locais onde sejam necessárias respostas rápidas e/ou sistemas remotos de entrada e saída, assim como CLPs ligados a sensores e atuadores.

Faz uso do padrão RS485 ou fibra ótica para a comunicação na camada 1 (física). Consegue obter uma taxa de

5043e

transmissão, usando RS485, de até 12 Mbps até 100 metros de distância e 93,75 Kbps quando se distancia até 1200 metros.

Suporta até 127 estações com 4 segmentos de rede que são ligados por repetidores. Também faz uso do sistema master-slave com apenas um mestre e vários escravos. Tem serviços sem conexão e sem reconhecimento, o que constitui o LLC tipo 1. Por fim, também há serviços de aplicação para leitura e escrita de variáveis remotas, o que justifica o seu uso.

D. Sinec L2-FMS

Já nesta vertente do L2, tem-se a utilização principalmente para realizar a troca de dados entre sistemas autônomos mais complexos e inteligentes na manufatura, bem como CNCs, CLPs, RCs, PCs, etc.

Também faz o uso do padrão RS485 ou da fibra ótica na camada física. Entretanto, como há a possibilidade da existência de estações autônomas, é utilizado o MAC Token-Passing com sistema master-slave, o que permite a função de multi-mestre.

Este tipo de rede suporta até 127 estações com no máximo 4 segmentos de rede, utilizando os serviços LLC dos tipos 1 e 3. Já os serviços de aplicação seguem o padrão FMS (Fieldbus Message Services).

E. Sinec L2-PA

Este tipo de L2 permite a interligação dos instrumentos da área de processos unitários para um sistema de comunicação. Adota o padrão IEC 1158-2 na camada física, permitindo usar o mesmo cabo de transmissão de dados para fazer a energização dos dispositivos de campo.

Possui uma taxa de transmissão relativamente baixa, com cerca de 31,25 Kbps, pois é utilizado em áreas de segurança intrínseca, ou seja, áreas com riscos de explosão, incêndio ou outras adversidades que possam vir a acontecer com mais frequência nesses locais.

O comprimento máximo do cabo depende do número de estações conectadas e varia conforme do consumo de energia. 32 estações são suportadas em cada segmento sem repetidores. Assim como a L2-DP, utiliza o protocolo master-slave com LLC do tipo 1 e seus serviços de aplicação acabam sendo semelhantes.

VII. CONCLUSÃO

Por ser uma rede relativamente antiga, sendo do início da década de 1980, é difícil ver o seu uso muito abrangente nos dias de hoje. Entretanto, por ser uma rede em que é possível realizar a interconexão com diferentes tipos de rede, ainda é uma opção viável para certos tipos de sistemas de comunicação específicos.

Com o seu número alto de versões e variações, a rede Sinec se destaca por se encaixar em quase qualquer lugar, podendo servir para processos básicos com propósitos apenas industriais como para processos mais avançados e complexos, a exemplo do colisor de hádrons.

REFERÊNCIAS

- [1] DIN ISO 7498, modelo de referência OSI.
- [2] Bray et al. (1995) "The Control and Data acquisition of the LHC Test String". In proceedings of International Conference on Accelerator and Large experimental Physics Control Systems, Chicago, IL, USA, 30 Oct – 3 Nov 1995. <https://cds.cern.ch/record/293232?ln=no>. Acesso em Junho de 2016.
- [3] ALVARES, A. J. Tendências e Aplicações Especiais. In: Rede de Automação da Manufatura - RECOPE/MANET. (Org.). Robótica Industrial - Aplicação na Indústria de Manufatura e de Processos. 1ed.São Paulo: Ed. Edgar Blücher: Editora Edgar Blücher LTDA, 2002, v. I, p. 197-215.
- [4] Alvares, A. J.. Aspectos de Comunicação de Dados e LAN. In: Apostila LAN. ftp://graco.unb.br/pub/alvares/sim/0_apostila_alvares/lan.pdf. Acesso em Junho de 2016.
- [5] Becker, K. and Flaschka, E., "Sinec Concept and Implementation" in Sensors and Actuators A: Physical 25: 125, 1990.
- [6] T Kersting, R Sommer, H Stier, "SINEC H1, ein Bussystem fuer die Prozesskommunikation", 1985. Pp 152-156.
- [7] Ninin et al. (1997) "Industrial Control Systems for the Technical Infrastructure at CERN". In Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems. <http://www.aps.anl.gov/News/Conferences/1997/icalepcs/paper97/p133.pdf>. Acesso em Junho de 2016.
- [8] SIMATIC TI505, SINEC H1, Communication Processor User Manual
- [9] SNL2-E-SINEC-L1 (Serial Network Link for SINEC-L1) Installation Instructions.
- [10] SIMATIC S5, SINEC LI Local Area Network User Manual
- [11] SIMATIC S5, SINEC L2 Interface of the S5-95U Programmable Controller User Manual.
- [12] STEMMER, M. R.. Redes Locais Industriais - A integração da produção através das redes de comunicação. 1. ed. Florianópolis: EdUFSC, 2010. v. 1. 272p.
- [13] Turner, P. Programmable Logic Controllers – The complete Guide to the Technology. 1. Ed. Atlanta, USA 1996. https://books.google.com.br/books?id=AuMzoz90j10C&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. Acesso em Junho de 2016.
- [14] http://www.aotewell.com/wp-content/uploads/2012/04/SIMATIC_TDC.jpg. 2016.
- [15] http://www.aotewell.com/wp-content/uploads/2012/04/SIMATIC_TDC.jpg. 2016.

Protocolo de Redes Industriais PROFIBUS

Rômulo Pacher Júnior

Universidade Federal de Santa Catarina

Email: romulo.pacher.j@grad.ufsc.br

Resumo—PROFIBUS é um dos protocolos de barramento de campo mais amplamente usado, realizando comunicação desde a camada de dispositivos até a camada de célula em redes industriais através de um único cabo, enquanto também permite interoperabilidade com outros protocolos de comunicação atuando na mesma camada hierárquica ou em camadas mais altas e oferecendo uma ampla gama de aplicações. Este artigo apresenta uma visão geral do protocolo e suas características, mostrando de forma sucinta sua arquitetura e comentando sobre suas aplicações e os perfis empregados na padronização das estações que compõe a rede.

Keywords—Automação, Barramento de Campo, PROFIBUS, Redes Industriais

I. INTRODUÇÃO

Com o uso de redes industriais é possível realizar de forma remota o controle de manufaturas e processos automatizados, fazer aquisição de dados para sistemas de controle da produção e planejamento de recursos. De forma a reduzir custos, melhorar o desempenho e eficiência além de facilitar a execução de diagnósticos. Vendo grande oportunidade neste campo, diversos modelos foram desenvolvidos por companhias e instituições, mas ao decorrer do tempo muitos modelos acabaram sendo abandonados por não serem eficientes, por não acompanharem a constante mudança das tecnologias da automação ou por não serem flexíveis, exigindo que todos os componentes da rede fossem do mesmo fabricante e não permitindo ligações com outros protocolos atuando em níveis hierárquicos mais altos ou baixos da rede industrial.

Este artigo tem como objetivo apresentar o modelo de redes industriais PROFIBUS de forma geral, falando um pouco sobre sua história, suas características, arquitetura e aplicações.

O PROFIBUS (Process Field Bus), um dos modelos mais amplamente usados atualmente, com mais de 35 milhões de nós instalados, é independente de fabricantes, padronizado pelas normas EN50170, EN50254 e abertamente publicado como parte de IEC 61158 e IEC 61784. Realiza a ligação de sistemas de controle com dispositivos de campo, operando em níveis do nível de componentes até o nível de células, por meio de um único cabo e permite ligação consistente com sistemas de comunicação de mais alto nível. [1] [2] [3]

II. ORIGEM E HISTÓRIA

O PROFIBUS teve origem no ano de 1986 quando 21 instituições e companhias se juntaram a pedido do Ministério Federal Alemão de Pesquisa e Tecnologia para um projeto chamado fieldbus cujo objetivo era a implementação e disseminação de um barramento de campo bit-serial baseando se nos requisitos das interfaces de dispositivos de campo. [2] [3] Uma

parcela das companhias e instituições envolvidas desenvolveu o PROFIBUS FMS (Field bus Message Specification) que veio a ser a primeira versão do padrão. [2] [3] Posteriormente no ano de 1993 foi concluída a especificação do protocolo PROFIBUS DP (Decentralised Peripherals) que era mais simples e rápido que seu antecessor. [2] [3] Outro protocolo, PROFIBUS PA (Process Automation), foi criado no ano de 1996 para atender a instrumentos de controle de processos e todas as aplicações relacionadas. [2] [3]

Atualmente a versão FSM caiu em desuso devido ao fato de ser mais complexa e lenta que as versões mais novas. O protocolo PROFIBUS PD é a versão predominantemente usada em aplicações do modelo enquanto a versão PA é utilizada em aplicações mais específicas de automação de processos. [2]

III. CARACTERÍSTICAS E VERSÕES

Uma característica importante do PROFIBUS é sua estrutura modular e o uso de perfis de aplicação normatizados que garantem o comportamento uniforme dos dispositivos independentemente do fabricante. Por meio do uso de módulos específicos para aplicação integrados com o protocolo central e tendo todos os dispositivos da rede encaixados dentro de algum dos perfis, é assegurada consistência com grande variedade de aplicações. [4]

Outra característica importante é a possibilidade de interoperar com outros modelos de rede como CAN, SINEC, HART e PROFINET. Isso é possível através do uso de dispositivos como *bridges* e *gateways* que realizam a conversão das mensagens transmitidas entre as redes.

O PROFIBUS faz uso da topologia de barramento e os dispositivos que participam da rede são separados entre mestres e escravos, sendo que o protocolo permite redes multimestre. Os mestres controlam o uso do meio, determinando quem e quando deve realizar o envio de mensagens. O controle de acesso ao meio dos mestres é efetuado pela passagem de um *token (token-bus)*, que garante direito de acesso ao barramento à estação mestre que o possui. [1]

Como mencionado anteriormente, o PROFIBUS possui duas versões distintas, sendo a DP de aplicação mais geral e a PA especializada em automação de processos. Pelo fato das versões serem compatíveis entre si, é possível se realizar automação híbrida evitando o uso de duas tecnologias diferentes em um único ambiente industrial.

A. PROFIBUS DP

A versão DP é projetada para realizar a transmissão de dados entre sistemas de controle e dispositivos I/O, sendo

otimizada para operar em tempos críticos e permitindo conexões de baixo custo. Pode ser aplicada tanto na automação de processos discretos quanto de processos analógicos, podendo transmitir sinais de 4 a 20 mA, mas possui algumas restrições para uso nesta última devido a parametrização. Considerando uma rede com 32 estações, o PROFIBUS DP pode enviar 512 bits em 6 ms a velocidade de 1,5 Mbit/s, ou em menos de 2 ms no caso de uma conexão de 12 Mbit/s, distribuídos por todas as estações. Esta versão possui uma gama de diagnósticos que ajuda a identificar com rapidez falhas que venham a ocorrer na rede. [1]

São definidos 3 classes de dispositivo:

- Mestre DP classe 1 (DPM1): dispositivos com esta denominação são controladores centrais, responsáveis por realizar a comunicação com os periféricos (escravos DP);
- Mestre DP classe 2 (DPM2): são equipamentos que normalmente fazem parte de terminais de engenharia e são empregados no monitoramento e configuração, tanto da rede quanto dos escravos DP;
- Escravo DP: estações de comunicação passivas que transmitem dados somente quando recebem uma requisição de um mestre. Escravos DP são em sua maioria dispositivos periféricos de entrada e saída, podendo ser compactos ou modulares. Dispositivos modulares são compostos por uma interface fieldbus na qual são ligados diversos módulos, permitindo que um único nó da rede adquira múltiplas entradas e aplique várias saídas.

B. PROFIBUS PA

O PROFIBUS PA foi projetado para implementação em automação de processos analógicos, para transmissão de sinais de 4 a 20 mA ou HART. Seu uso é muito menos frequente quando comparado com a versão DP e seu custo é maior. Sua velocidade de operação é de 31,25 Kbit/s. A versão PA faz o uso dos mesmos protocolos da versão DP, sendo que para seu uso é necessária uma conexão com uma rede DP que atua como um *backbone* para transmissão de sinais. Uma das vantagens do PROFIBUS PA em relação ao DP é que sua instalação pode ser feita em áreas com risco de explosão, pois a camada física pode fazer a alimentação dos dispositivos conectados à rede enquanto limita o fluxo de corrente, evitando condições para ocorrência de explosão mesmo em caso de defeito. Isso acaba limitando o número de dispositivos que podem ser ligados à rede. Outra vantagem é a possibilidade do uso de FDT (*Field Device Tool*) ou DTM (*Device Type Manager*), que são tecnologias mais avançadas para análise de falhas, diagnósticos e parâmetros.

IV. ARQUITETURA

A arquitetura do PROFIBUS é baseada no modelo de referência OSI (Open System Interconnection), criado pela ISO, composto por sete camadas. No PROFIBUS são utilizadas somente as camadas 1, 2 e 7. [4] [5] O motivo disso é para tornar o protocolo mais simples, diminuindo o tempo de processamento e aumentando a velocidade de operação.

User program		Application profiles	
7	Application Layer	PROFIBUS DP Protocol (DP-V0, DP-V1, DP-V2)	
6	Presentation Layer	Not used	
5	Session Layer		
4	Transport Layer		
3	Network Layer	Fieldbus Data Link (FDL): Master Slave principle Token principle	
2	Data link Layer		
1	Physical Layer	Transmission technology	
OSI Layer Model		OSI implementation at PROFIBUS	

Figura 1. Camadas do modelo de referência OSI implementadas no PROFIBUS [4]

A. Camada Física

A camada física do PROFIBUS DP pode usar tanto RS485, quanto fibra óptica como tecnologia de transmissão. O RS485 é um par trançado blindado comumente utilizado por ser custo eficiente e permitir conexões de alta velocidade de transmissão. Existem duas variações do cabo, uma com duas vias que transporta somente os sinais de comunicação, que é composto por um fio verde e outro vermelho, e um cabo de 4 vias que além de transportar sinais também possui um fio marrom e um branco, além do verde e vermelho, para realizar a alimentação dos equipamentos. Ambos os tipos possuem isolamento externo de cor roxa de acordo com a norma. [1] A fibra óptica é usada para conexões a distâncias que excedem o limite de alcance do RS485 ou em áreas com grande interferência magnética. O método de codificação NRZ (Não Retorna Zero) é utilizado para comunicação em ambos os meios e a taxa de transmissão pode ser 9,6 Kbit/s até 12 Mbit/s. Na Tabela I pode-se observar as taxas de transmissão e seus alcances respectivos levando em consideração cabos do tipo A com as seguintes propriedades: impedância de 135 a 165Ω, capacitância menor ou igual a 30pF/m, resistência do cabo menor ou igual a 110Ω/m, diâmetro do condutor interno maior que 0,64 mm e área da seção transversal do condutor interno maior que 0,34 mm². [4]

Tabela I
TAXA DE TRANSMISSÃO E SEUS LIMITES DE ALCANCE [4]

Taxa de Transmissão (Kbit/s)	Limite de Alcance por Segmento (m)	Meio Físico
9,6	1200	RS485
19,2	1000	RS485
45,45	400	RS485
93,75	200	RS485
3000	100	RS485
6000	1900	MBP
12000		
31,25		

A camada física do PROFIBUS PA também pode usar a tecnologia RS485 com codificação NRZ ou, alternativamente, pode usar a tecnologia MBP (*Manchester coded, Bus Powered*) usando codificação Manchester com taxa de transmissão de 31,25 Kbit/s. Quando é efetuada instalação em zonas de risco de explosão, são utilizadas as variações intrinsecamente

Tabela II
INFORMAÇÕES GERAIS DA CAMADA FÍSICA [4]

	RS485	RS485-SI	MBP	MBP-SI	Fibra Óptica
Transmissão de Dados	digital; sinal diferencial acc. Para RS485, NRZ	digital; sinal diferencial acc. Para RS485, NRZ	digital, síncrona, Manchester	bit-síncrona, codificação Manchester	Óptico, digital, NRZ
Taxa de Transmissão	9.6 a 12000 Kbit/s	9.6 a 12000 Kbit/s	31,25 Kbit/s	31,25 Kbit/s	9,6 a 12000 Kbit/s
Segurança dos Dados	HD = 4; bit de paridade; delimitador de início/fim	HD = 4; bit de paridade; delimitador de início/fim	preâmbulo, contra delimitador de início/fim	seguro contra falhas de falhas de início/fim	HD = 4; bit de paridade; delimitador de início/fim
Cabo	par trançado blindado tipo A	par trançado blindado tipo A	par trançado blindado tipo A	par trançado blindado tipo A	fibra de vidro multi e monomodo; PFC; fibra plástica
Alimentação Remota	possível usando núcleos adicionais	possível usando núcleos adicionais	opcional usando os núcleos de sinal	opcional usando os núcleos de sinal	possível usando cabo híbrido
Proteção contra Ignição	nenhuma	segurança intrínseca Ex	nenhuma	segurança intrínseca Ex ia/ib	nenhuma
Topologia	topologia de linha com terminação	topologia de linha com terminação	topologia de linha com terminação	topologia de linha com terminação; combinada também	tipicamente estrela e anel; linha possível
Número de Estações	até 32 nós por segmento. máximo de 126 por rede	até 32 nós por segmento. máximo de 126 por rede	até 32 nós por segmento. máximo de 126 por rede	até 32 nós por segmento. máximo de 126 por rede	Até 126 nós por rede
Número de Repetidores	até 9 com signal refreshing	até 9 com signal refreshing	até 4 com signal refreshing	até 4 com signal refreshing	ilimitado com signal refreshing; notar atraso de propagação do sinal

seguras RS485-IS ou MBP-IS. [1] [4] [2] A Tabela 2 apresenta as informações gerais da camada física.

O PROFIBUS PA precisa estar conectado a uma rede DP para operar. Para realizar a conexão entre DP e PA (Fig. 2) deve ser usado um acoplador ou um *link*. O acoplador apenas conecta os meios físicos das redes, sem causar alterações no endereçamento e fazendo com que os módulos PA operem como DP do ponto de vista do mestre. O *link* funciona como um escravo na rede DP e mestre na PA, atuando como um *gateway*. [1]

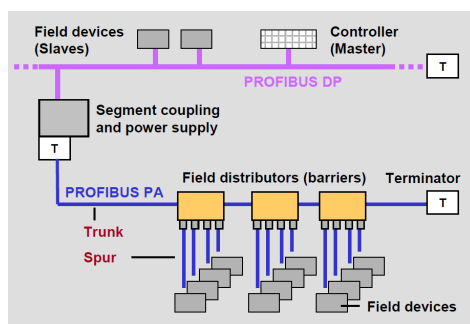


Figura 2. Acoplamento de segmento PA em rede DP. [4]

Há a possibilidade de se utilizar meios físicos *wireless* como IEEE 802.11, apesar da integração destes não ser especificada pela PI, e conseguir estabelecer comunicação com estabilidade aceitável. [4] Para fazer a conexão das estações *wireless* ao resto da rede pode se usar um *bridge* que realiza a conversão

do meio físico, sem alterar o funcionamento da rede, ou empregar um protocolo para a subcamada MAC especialmente projetado para realizar a comunicação sem fio que atua paralelamente com o protocolo PROFIBUS padrão e oferece a mesma interface de camada de enlace. A integração pode ainda ser feita diretamente na camada de aplicação ou usando uma combinação dos métodos citados. [6] Ainda assim, o PROFIBUS apresenta problemas de instabilidade caso aja alta taxa de erro no meio e seu desempenho é superado por outros protocolos. [7]

A camada física oferece duas primitivas de serviço diferentes para a camada de enlace de dados: a requisição de envio de informações `PHY_DATA.request` e uma indicação da presença de informação no barramento `PHY_DATA.indication`. [1]

B. Camada de Enlace de Dados

A camada de enlace, ou FDL (*Fieldbus Data Link*), do protocolo é composta por duas subcamadas, sendo elas MAC (*Medium Access Control*), responsável pelo controle de acesso ao meio físico e LLC (*Logical Link Control*), responsável pelo controle da ligação lógica. [1]

A subcamada MAC utiliza uma combinação dos métodos *token passing* e mestre-escravo. Os escravos, também chamados de estações passivas, tem seu acesso ao meio controlado pelos mestres, ou estações ativas, de forma que para um escravo fazer uso do meio ele precisa receber uma requisição do mestre para transmitir dados. Para o controle do acesso ao meio dos mestres é utilizado o método *token bus*, baseado na norma IEEE 802.4, em que um *token*, que

representa o direito de acesso ao barramento, é passado ao longo de um anel lógico formado pelas estações mestres. Para a passagem do *token*, uma estação A deve transmitir um quadro de *token* para seu sucessor B. Quando B recebe o *token*, deve começar a transmitir dentro de um intervalo de tempo pré definido, chamado tempo de *slot*, caso contrario a estação A repetira o envio do *token*. A estação A realiza até três tentativas para passar o *token* para B, caso não tenha sucesso B é considerada inativa e A então envia um quadro de *token* para a estação sucessora de B. Os mecanismos de manutenção do anel são responsáveis por incluir estações no anel, sejam elas novas ou estações que perderam conexão com o anel. Quando uma estação percebe que foi considerada como inativa ela elimina todas as informações armazenadas sobre o anel lógico e se comporta como uma estação recém ativada. Uma estação recém ativada passa por um período de preparação para se incluir no anel em que ela ouve as transmissões de *tokens* na rede até terem se completado dois ciclos completos da passagem do *token*, para formar uma lista contendo os endereços das estações que compõe o anel e a ordem de passagem do *token*. Nesse período o *token* pode ser oferecido a ela por alguma outra estação, caso ela já tenha ouvido os dois ciclos completos do anel lógico ela aceita o *token* e se inclui na lista de estações e marcando o endereço da estação que lhe enviou o *token* como sua antecessora. Caso ela não tenha ouvido os dois ciclos completos do *token* ela responde que ainda não esta pronta. Quando o meio fica inativo por muito tempo, as estações começa a contagem de um timer. Caso ocorra alguma transmissão o timer é reiniciado. Caso não ocorram transmissões antes do timer chegar a zero a estação se considera sozinha no anel e envia um *token* para si mesma. [8] Este método apresenta problemas quando há alta taxa de erro no meio físico, devido à possibilidade de corrupção do quadro do *token*, podendo causar níveis inaceitáveis de instabilidade. [9]

É definida uma série de telegramas (Fig. 3, Fig. 4, Fig. 5, Fig. 6) que são utilizados na comunicação.

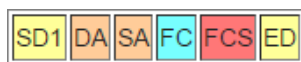


Figura 3. Estrutura de telegrama sem campo de dados. [2]

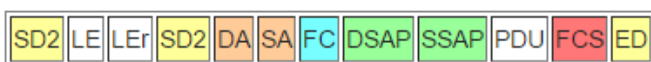


Figura 4. Estrutura de telegrama com campo de dados de comprimento variável. [2]



Figura 5. Estrutura de telegrama com campo de dados de comprimento fixo. [2]

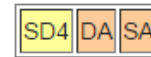


Figura 6. Estrutura de telegrama de transmissão de token. [2]

Estes são divididos em dois grupos, telegramas longos e telegramas curtos. Telegramas longos são usados na comunicação entre equipamentos mais intrincados como os mestres. Telegramas curtos são usados na comunicação com dispositivos escravos.

Os telegramas são formados por vários campos, cada um com sua função específica:

- SYN: sincronismo para início da comunicação; [1]
- SD2 (*Start Delimiter*): determina o início do telegrama; [1] [2]
- LE e LER (*Length*): indica o tamanho da informação no campo DU; [1] [2]
- DA (*Destination Address*): contém o endereço do destino do telegrama; [1] [2]
- SA (*Source Address*): contém o endereço da origem do telegrama; [1] [2]
- FC (*Frame Control*): controle de telegrama; [1] [2]
- DU (*Informações*): campo contendo DSAP, SSAP e PDU, pode variar de 1 a 244 bytes por escravo na rede PROFIBUS; [1]
- DSAP (*Destination Service Access Point*): identificação de network endpoint do destino do telegrama; [2]
- SSAP (*Source Service Access Point*): identificação de network endpoint da fonte do telegrama; [2]
- PDU (*Protocol Data Unit*): dados a serem transmitidas pelo telegrama; [2]
- FCS (*Frame Check Sequence*): sequência de verificação de erros; [1] [2]
- ED (*End Delimiter*): determina o fim do telegrama. [1] [2]

Os serviços oferecidos a camada de aplicação pela camada de enlace são definidos pela subcamada FDL. Entre os serviços estão transmissão ou requisição de informações, gerenciamento do token e atualização das estações. As primitivas de serviço do protocolo FDL podem ser observadas na Tabela 3.

C. Camada de Aplicação

A camada de aplicação no protocolo PROFIBUS faz a interface com os processos de aplicação que se localizam acima dela. [4] É composta por três subcamadas: ALI (*Application Layer Interface*), LLI (*Lower Layer Interface*) e FMS (*Fieldbus Message Specification*). A subcamada LLI e ALI são encarregadas, respectivamente, de realizar a interface com a camada de enlace de dados e realizar a interface com os processos de aplicação do usuário. [1] A subcamada FMS contém o protocolo em si. [1]

Na camada de aplicação estão definidos os três perfis de performance: DP-V0, DP-V1 e DP-V2. A versão DP-V0 é usada para comunicação cíclica e realização de diagnósticos. A versão DP-V1 é utilizada para transmissões acíclicas como

Tabela III
SERVIÇOS OFERECIDOS PELO PROTOCOLO FDL [1]

Classe	Primitiva de Serviço	Comentários
SDN (<i>send data with no acknowledge</i>)	FDL_DATA	envio de dados sem reconhecimento
SDA (<i>send data with acknowledge</i>)	FDL_DATA_ACK	envio de dados com reconhecimento
RDR (<i>request data with reply</i>)	FDL_REPLY FDL_REPLY_UPDATE	requisição de dados com reconhecimento
CRDR (<i>cyclic data with reply</i>)	FDL_CYC_REPLY FDL_CYC_DEACT FDL_REPLY FDL_REPLY_UPDATE	estação local requisita dados ao usuário remoto ciclicamente
CSRDR (<i>cyclic send and request data</i>)	FDL_SEND_UPDATE FDL_CYC_DATA_REPLY FDL_CYC_DEACT FDL_DATA_REPLY FDL_DATA_UPDATE	Estação local envia ciclicamente e requisita, simultaneamente, dados de respostas
SRD (<i>send and request data</i>)	FDL_DATA_REPLY FDL_REPLY_UPDATE	leitura e escrita de variáveis contidas em dispositivos servidores

monitoramento, operação, parametrização e controle de alarmes, além do acesso de nodos por ferramentas de engenharia. A versão DP-V2 é empregada para realização de comunicação em modo isócrono, contendo funções para comunicação entre escravos, sincronização de ciclos e marcação de tempo. [4]

V. PERFIS DE APLICAÇÃO

Para que possa haver uma comunicação estável entre as estações em uma rede é necessária uma padronização dos serviços, conceitos e formatos básicos usados pelas estações. Para garantir essa uniformidade o PROFIBUS define perfis que relacionam famílias de dispositivos e especificam características que os dispositivos associados ao perfil devem apresentar. [4]

Existem características específicas à classe de dispositivos e características apresentadas por diferentes classes de dispositivos. Na Tabela IV são listados os perfis de aplicação gerais. Na Tabela V são listados os perfis de aplicação específicos.

VI. REDUNDÂNCIA

Existem aplicações que exigem alta disponibilidade do sistema de controle, como processos contínuos por exemplo. Para estes casos são implementadas redundâncias, que pode se estender a todos os componentes da rede. [4] É possível definir alguns conceitos diferentes de redundância que podem ser usados com o PROFIBUS:

- Redundância de Mestre: o sistema de controle ou controlador é projetado redundantemente (Fig. 7).
- Redundância de Meio: o cabeamento é projetado redundantemente (Fig. 7).
- Redundância em Anel: fazendo o uso de acopladores redundantes e distribuidores de campo ativos é possível se implementar redundância em anel (Fig. 8), de forma que se alguma estação para de funcionar, houver uma

Tabela IV
PERFIS DE APLICAÇÃO GERAIS PROFIBUS [3]

Nome do Perfil	Comentário	Diretriz PNO
Identificação e manutenção	Especifica o conceito de dispositivo PROFIBUS e acesso pela internet à informações específicas do dispositivo	3.502
iPar-Server	Define a preservação de i-parâmetros no controlador e o retorno a leitura de i-parâmetros após a substituição de dispositivos	3.532
PROFI-safe	Define comunicação segura de dispositivos relacionados a segurança (paradas de emergência, etc.) no PROFIBUS	3.092
Redundância	Especifica os mecanismos para dispositivos de campo com comportamento de comunicação redundante	2.212
Marcação de Tempo	Define o tempo preciso da atribuição de certos eventos e ações por marcação de tempo	3.522

Tabela V
PERFIS DE APLICAÇÃO ESPECÍFICOS PROFIBUS [3]

Nome do Perfil	Comentário	Diretriz PNO
Dosing / Weighing	Descreve o uso de sistemas de pesagem e dosagem no PROFIBUS	3.182a; 3.182b
Encoder	Descreve a conexão de encoders lineares, angulares e rotacionais com resolução de volta única ou múltipla	3.062
Fluid Power	descreve o controle de drives hidráulicos via PROFIBUS (cooperação com VDMA)	3.112
HART on PROFIBUS	Define a integração de dispositivos HART em sistemas PROFIBUS	3.102
Ident Systems	Descreve a comunicação entre dispositivos de comunicação (leitor de código de barras, transponder)	3.142
LabDevices	Especifica as propriedades de dispositivos de automação laboratorial no PROFIBUS	2.412
Liquid Pumps	Define o uso de bombas hidráulicas no PROFIBUS (cooperação com VDMA)	2.422
Low Voltage Switchgear	define troca de dados para aparelhagem de baixa voltagem no PROFIBUS	3.122
PA Devices	Especifica as propriedades de dispositivos de automação de processos	3.042
PROFI-drive	descreve o comportamento de dispositivo e comportamento de acesso de dados para drives elétricos de velocidade variável no PROFIBUS	3.172; 3.272
Remote I/O	Define a permutabilidade de dispositivos I/O remotos em processos de automação	3.132
SEMI	Descreve propriedades de dispositivos de produção de semicondutores no PROFIBUS (cooperação com SEMI)	3.152

ruptura no cabo ou um acoplador parar apresentar defeito, os subsegmentos restantes continuam operando como barramentos.

- Redundância de Escravo: Os dispositivos de campo são

projetados com redundância.

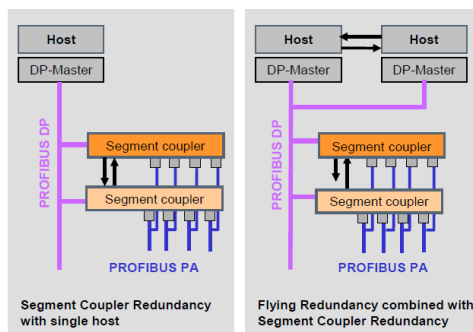


Figura 7. Redundância de meio e redundância de mestre. [4]

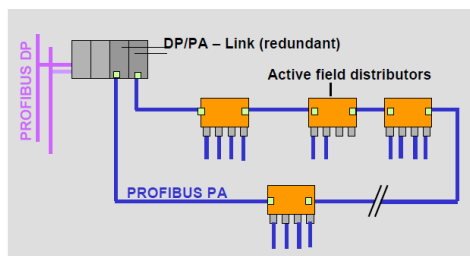


Figura 8. Redundância de anel. [4]

VII. CONCLUSÃO

Como pudemos ver, o PROFIBUS é um modelo de rede industrial versátil com ampla gama de dispositivos disponíveis, produzidos por um grande número de fornecedores devido a independência de fabricantes, que possui vários atributos úteis como sua capacidade de interoperabilidade e aplicações híbridas de automação de processos e de produção, mas não é livre de problemas, como seu desempenho instável em meios com alta taxa de erro devido à corrupção ou perda do *token*.

Ainda assim o PROFIBUS permite comunicação rápida, robusta, de baixo custo e fácil instalação com ampla variedade de aplicações.

REFERÊNCIAS

- [1] A. B. Lugli, *Redes Industriais: características, padrões e aplicações*, 1st ed. São Paulo, Brasil: Érica, 2014.
- [2] "Wikipedia the free encyclopedia profibus," <https://en.wikipedia.org/wiki/Profibus>.
- [3] "Pi north america: Profibus, fieldbus for networks in industrial automation," <http://us.profinet.com/technology/profibus/>.
- [4] P. N. e.V. (PNO), "Profibus system description - technology and application," Haid und Neu-Str. 7 76313 Karlsruhe Germany, November 2010.
- [5] K. C. Lee, S. Lee, and M. H. Lee, "Remote fuzzy logic control of networked control system via profibus-dp," *IEEE transactions on Industrial Electronics*, vol. 50, no. 4, pp. 784–792, 2003.
- [6] M. Alves, E. Tovar, F. Vasques, G. Hammer, and K. Rother, "Real-time communications over hybrid wired/wireless profibus-based networks," in *Real-Time Systems, 2002. Proceedings. 14th Euromicro Conference on*. IEEE, 2002, pp. 142–151.
- [7] A. Willig, "Polling-based mac protocols for improving real-time performance in a wireless profibus," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 806–817, 2003.

- [8] E. Tovar and F. Vasques, "Real-time fieldbus communications using profibus networks," *IEEE transactions on Industrial Electronics*, vol. 46, no. 6, pp. 1241–1251, 1999.
- [9] A. Willig and A. Wolisz, "Ring stability of the profibus token-passing protocol over error-prone links," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 5, pp. 1025–1033, 2001.

Protocolo BITBUS

Redes Industriais

Caique Sanches

Engenharia de Controle e Automação
Universidade Federal de Santa Catarina
Blumenau, Brasil
caique.sanches@grad.ufsc.br

Resumo—As redes industriais são bastante utilizadas atualmente devido ao aumento da demanda de comunicação no chão de fábrica. Estas redes devem ser resistentes a distúrbios do ambiente como interferência elétrica, magnética e térmica. Este artigo explica sobre o protocolo BITBUS que é bastante implementado nas redes industriais e suas especificações de como deve ser implementado. Utiliza uma topologia mestre/escravo combinada com token-passing, podendo ter até 250 nós em seu barramento. Possui pacotes de informação enxutos e pode operar em diferentes velocidades conforme varia seu número de nós ou o tipo de conexão.

Keywords—BITBUS; protocolo; redes industriais; mestre/escravo;

I. INTRODUÇÃO

As redes industriais são definidas como uma estrutura digital que permite o envio de informações entre vários equipamentos como computadores, sensores e atuadores, colocados em ambiente industrial. Existem diversos tipos de protocolos feitos para resistir estes tipos de ambientes, entre eles está o BITBUS.

O BITBUS foi desenvolvido em 1983 pela Intel e promovido a padrão em 1990. Sua tecnologia é difundida principalmente pela BEUG (BITBUS European Users Group), uma empresa sem fins lucrativos para ajudar usuários do protocolo e criando blogs para troca de informação e experiências.

Comparado com o modelo OSI de referência de redes, o BITBUS possui apenas as camadas 1, 2 e 7, ou seja, camada física, de enlace e de aplicação. As funções das camadas 3 a 6 estão embutidas principalmente na camada 2. A função disto é dar maior desempenho ao protocolo.

O BITBUS possui uma estrutura de Mestre/Escravo combinada com “Token-Passing”, utilizando um barramento RS-485 sobre par trançado ou fibra ótica.

O seu barramento possui diferentes elementos como um mestre, escravos e suas extensões e repetidores. Nele pode haver até 32 escravos por segmento ou 250 utilizando repetidores.

Sua principal utilização é para controle de processos e em módulos de entradas e saídas inteligentes.

II. ESTRUTURA

O BITBUS utiliza uma estrutura hierárquica permitindo uma alta velocidade. Ele atua no barramento de campo utilizando o modelo de mestre escravo. Onde apenas um mestre se comunica com vários escravos. Estes escravos podem ser simples entradas e saídas como sensores e atuadores ou também controladores.

O modelo observado na figura 1 é chamado de estrutura hierárquica multinível, onde são interligados vários barramentos. Cada escravo do nível superior pode se comportar como mestre porem de um segundo barramento, ou seja, de um nível a baixo, assim tendo a possibilidade de ligar vários barramentos através dos escravos.

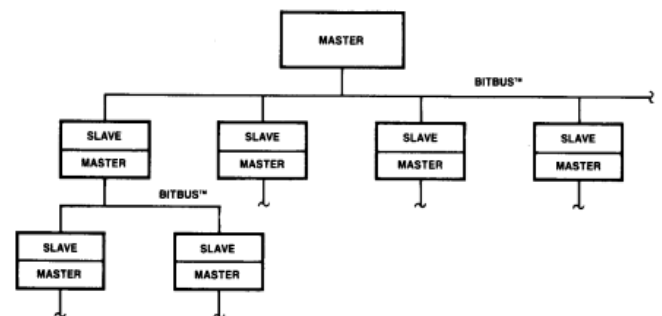


Figura 1. Estrutura hierárquica multinível.

III. ELEMENTOS DE UM BARRAMENTO

O elemento básico no barramento do sistema BITBUS é um nó. Um nó pode ser constituído de um dispositivo, um dispositivo e uma extensão, ou um repetidor. Os dispositivos podem ser mestres ou escravos com suas respectivas extensões.

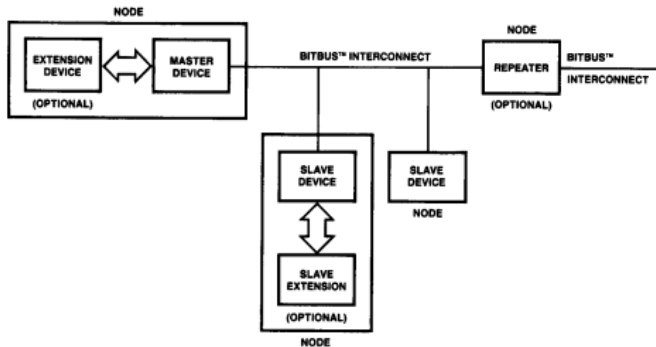


Figura 2. Elementos de um barramento.

Nas conexões BITBUS é possível ter apenas um dispositivo mestre, o qual controla todas as interconexões do sistema. Ele inicia a comunicação enviando uma mensagem de ordem para um escravo e, posteriormente, fica verificando se algum dos escravos deseja enviar uma mensagem de resposta.

Durante a operação normal cada mensagem enviada pelo mestre é identificada pelo endereço do escravo destino. Se não há uma resposta imediata disponível então é devolvida uma mensagem para identificar que foi recebida com sucesso. Isto faz com que o mestre possa enviar outras mensagens para qualquer outro escravo ou verificar se algum deles quer enviar uma resposta, maximizando o uso da largura de banda. [1]

A função de mestre pode ser estacionária ou pode ser passado para os outros nós similar ao funcionamento do token (simulação de uma ficha para indicar prioridade). Sua utilização tem seus aspectos positivos e negativos, dependendo da aplicação na qual se deseja utilizar o BITBUS. O processo de passar a função do mestre reduz a confiabilidade do sistema devido à possibilidade de perda no token, porém em alguns casos, como quando se deseja um mestre redundante, ajuda a diminuir o tempo entre falhas. Esta função não pode ser passada por repetidores então pode ser ainda mais útil se os possíveis mestres estão isolados dos escravos por um repetidor. [1]

Os escravos são os dispositivos que respondem ao mestre, podendo haver até 32 por segmento ou até 250 utilizando repetidores no sistema, endereçados do 1 ao 250. Eles não podem iniciar a comunicação, apenas dar respostas ao mestre e isto faz com que o protocolo seja mais simples e que tenham um baixo consumo de energia. [2]

As extensões atuam em um plano secundário dentro de um nó e a interface entre elas e o mestre ou escravo não fazem parte do mesmo protocolo, porém em alguns casos também é utilizado o BITBUS para formar hierarquias de vários níveis.

Um repetidor é um nó que é utilizado para regenerar os sinais da interface BITBUS. Também podem aumentar a distância ou aumentar o número de nós que podem ser conectados ao sistema.

IV. MODOS DE OPERAÇÃO

O protocolo BITBUS pode operar em dois modos: síncrono e assíncrono. O propósito de operar em diferentes

modos é possibilitar a utilização em várias faixas de alcances e desempenho para uma variedade de aplicações

O modo de operação síncrono é otimizado para um maior desempenho sobre uma pequena distância. Podem ser conectados até 28 nós em 30 metros com uma velocidade de transmissão entre 500 Kbps e 2,4 Mbps. São utilizadas duas conexões com um par de sinais cada, um para os dados e outro para o "clock". A vantagem em relação ao outro modo de operação é que sua velocidade de transmissão de dado é maior. O sistema pode ser observado na figura 3. [1]

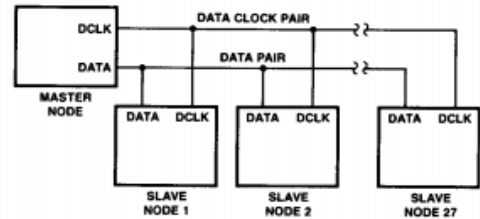


Figura 3. Conexão no modo síncrono.

O modo assíncrono é utilizado para operação em maiores distâncias. Este modo pode funcionar a 375kbps em até 300 metros ou 62,5kbps para até 1200 metros. A sua desvantagem em relação à velocidade é compensada com um maior alcance dos dados, graças à utilização de repetidores em seu sistema como é possível observar na figura 4.

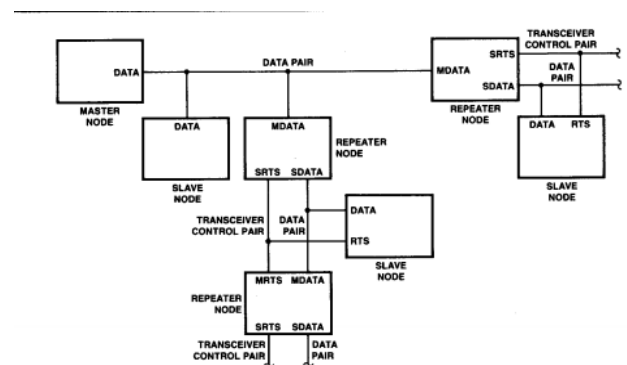


Figura 4. Conexão do modo assíncrono.

Neste, um par de sinais é utilizado para os transceptores dos repetidores. O sinal de dados é codificado com o método NRZI (Non-Return to Zero Inverted), que combina o "clock" e os dados no mesmo sinal. Quando nenhum escravo está transmitindo, uma combinação de resistores configuram os transceptores para que todos possam receber informações do mestre e quando um deles quer responder, a direção de todos os que estejam no caminho é invertida. [1]

V. CAMADA FÍSICA

O protocolo BITBUS utiliza o RS-485 sobre o barramento de par trançado ou fibra ótica. O RS-485 é um padrão de conexão elétrica do tipo multiponto, sendo muito efetivo para

longas distâncias e ambientes com grandes perturbações. Este barramento permite até 32 nós sem o uso de repetidores. A velocidade do barramento varia de acordo com o meio físico utilizado e o número de repetidores. [3]

A conexão deste protocolo pode ser feita por meio de cabos de par trançado, o que possibilita um menor custo de instalação, ou, atualmente, de fibra ótica, aumentando a velocidade de transmissão e a distância alcançada pelo sinal. Para a escolha do tipo de conexão a ser utilizada deve ser considerado a impedância, atenuação e resistência do respectivo.

A tabela 1 mostra que com até dois repetidores, a velocidade pode ser de 375kb/s e com até dez, 62,5kb/s sobre o par trançado. Utilizando fibra ótica a velocidade é de 1,5Mbits/s. [2]

Os conectores utilizados nas conexões devem ter 9 pinos com o padrão sub-D9 como mostra a figura 5.

VI. CAMADA DE ENLACE

O protocolo BITBUS segue o modelo de FieldBus que possui somente as camadas 1, 2 e 7, portanto as funções que se encontravam entre as camadas 3 a 6 foram distribuídas para as outras camadas, principalmente para a de enlace. Nela se encontram especificações de interface elétrica e enlace de dados como formato do quadro e da mensagem.



Figura 5. Conector sub-D9

TABELA 1. VELOCIDADE E NÚMERO DE REPETIDORES.

Velocidade (kbit/s)	Número máximo de repetidores entre o mestre e o escravo
375	2
62,5	10

As especificações da interface elétrica definem a codificação de bits que serão transmitidos nos diferentes modos de operação.

Na codificação de dados do modo síncrono um zero digital é representado pelo valor negativo de tensão (V-) e o um pelo positivo (V+). A codificação dos dados deve haver a inserção ou eliminação do bit zero, ou seja, modificar alguns padrões dos dados para que não sejam confundidos com os flags. Estes

são representados por 01111110, então cada vez que o transmissor detectar uma sequência de cinco bits um, ele adiciona um zero independente do próximo bit. Desta maneira o receptor não irá confundir uma parte da mensagem com um flag. [1]

No modo assíncrono é utilizado o método NRZI para a codificação e também a inserção ou eliminação do bit zero, assim como no modo síncrono. Neste, um bit zero representa uma mudança de nível (de V- para V+, ou de V+ para V-), e o bit um faz com que o nível seja mantido.

Baseado no modelo SDLC (Controle Síncrono de Enlace de Dados), o enlace de dados do BITBUS utiliza um único mestre conectando com vários escravos em uma topologia multiponto. Ele possui a função de formar e controlar os quadros (ou frames) utilizando as mensagens que serão transmitidas. Os quadros tem o formato mostrado na figura 6, com o número de bytes respectivos de cada campo.

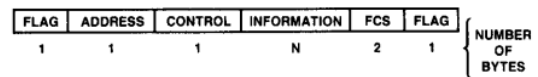


Figura 6. Quadro de transmissão do BITBUS.

O bit menos significativo de cada byte é transmitido primeiro, exceto no caso do FCS (verificação de sequência do quadro), em que o bit mais significativo é transmitido primeiro.

Os flags são utilizados para delimitar o quadro e ambos possuem a sequência 01111110. Para garantir que não há nenhuma outra sequência igual dentro de quadro é utilizada a inserção e eliminação de um bit zero.

O campo de endereçamento indica o endereço do escravo envolvido na comunicação. Quando o mestre envia a mensagem, está identificado o escravo destino e, quando um escravo envia a mensagem, é mostrado qual o escravo que enviou a mensagem. O campo contém um byte com valores de 0 a 255, cujos 0 e de 251 a 255 são reservados pela Intel e os restantes são de uso livre. Escravo devem ignorar os quadros que não possuem seu endereço. [1]

O campo de controle é usado para três operações: sincronização, supervisão e transferência de mensagem. Para uma sequência de mensagens transmitidas para o escravo é necessária uma sincronização, isto é feito através do envio do byte de controle observado na figura 7.

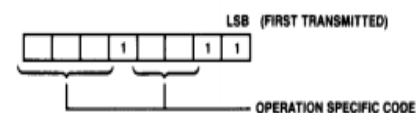


Figura 7. Byte de controle para sincronização.

Após a sincronização, quando não é enviada nenhuma mensagem, é realizada uma supervisão. Pode ser utilizada quando o mestre interpola os escravos (RR se o receptor está pronto para receber ou RNR se não pode receber mensagem no momento), ou para um escravo informar que foi recebido

um pacote com sucesso, identificando sua sequência (N). Para essa supervisão o byte deve ser no formato da figura 8.

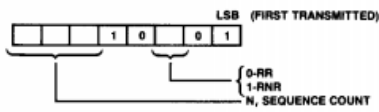


Figura 8. Byte de controle para supervisão.

No caso de transferência de mensagem, apenas é informado que será transmitida uma mensagem e a sua sequência para que o receptor possa deixá-las em ordem, representado na figura 9.

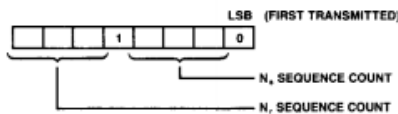


Figura 9. Byte de controle para transferência.

O FCS é a verificação de erro de mais baixo nível, contendo um CRC de 16 bits. O nó transmissor gera este campo de acordo com o polinômio $X^{16} + X^{12} + X^5 + 1$ e o envia, se o receptor o verifica e se não estiver correto então ignora o quadro. [1]

Os erros que podem ocorrer no enlace de dados são: contagem da sequência, campo de controle inválido ou o tempo de verificação é esgotado. Se o mestre não recebe uma confirmação de recebimento do escravo em 10ms é detectado como um erro, então é feita uma segunda tentativa de retransmitir o pacote. Se o erro acontecer pela segunda vez consecutiva é enviado um protocolo de erro ao usuário.

No campo de informação são transmitidas as mensagens, porém no caso da sincronização ou supervisão não é utilizado. A mensagem deve seguir um padrão mostrado na figura 10.

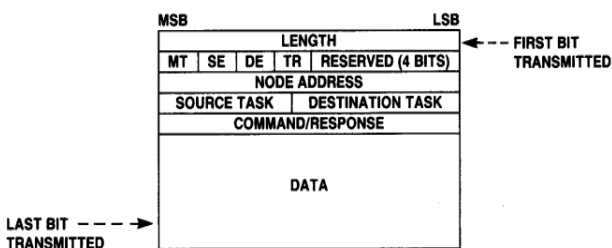


Figura 10. Quadro de mensagem.

Onde cada campo é definido por:

A. Length

Contém oito bits especifica o tamanho total da mensagem, podendo ter valores de 7 a 255. Mostra o número de bytes no campo de dados mais 7, facilitando a adição de dois bytes para

manipulação de mensagens locais como filas ou controle de buffers, caso necessário.

B. MT

Representa o tipo de mensagem, ou seja, se é uma ordem ou uma resposta. O mestre sempre enviará uma ordem então deixará este campo como 0 e o escravo sempre uma resposta, então o bit será setado.

C. SE

Indica se a fonte de uma ordem, ou o destino de uma resposta, é o mestre ou sua extensão. Se o bit é setado, indica que é referido a extensão, se for 0, representa que a mensagem ou a resposta é referente ao mestre. O bit não é alterado entre a ordem e sua respectiva resposta.

D. DE

Possui a mesma função que o SE, porém referente ao escravo. É setado se for referente à extensão ou 0 se for ao escravo. Este bit também não é alterado entre a ordem e sua resposta.

E. TR

Track Field, utilizado para controle em um nó, mestre ou escravo, por algumas aplicações. O bit é 0 quando está sendo enviada uma mensagem e 1 quando está sendo recebida.

F. Reserved

Possui quatro bits para possíveis melhorias a serem aplicadas. Quando uma mensagem é enviada devem ser colocados em 0 e quando está sendo recebida pode ser qualquer valor.

G. Node Address

Este campo especifica o endereço destino nas ordens e origem nas respostas. Este byte possui a mesma informação que o campo de endereço.

H. Source Task

Campo de quatro bits que identifica a tarefa que o mestre gerou ou se esta por receber uma resposta.

I. Destination Task

Identifica a tarefa que irá receber uma ordem ou gerar uma resposta. Complementa o campo anterior para os escravos.

J. Command/Response

Nas condições normais este campo é apenas utilizado pelas tarefas do usuário e somente usado pelo protocolo de mensagem para reportar erros. A tabela 2 mostra os usos para as respostas ou erros.

Tabela 2. RESPOSTAS DO PROTOCOLO DE MENSAGEM.

Condição	Erro
Sem erro	00H
Definido pelo usuário	01H-7FH
Tarefa sem destinatário	80H
Erro de protocolo	91H
Sem dispositivo de destino	93H
Reservado pela Intel	81H-90H,92H,94H-0FFH

K. Data

É definido pelo conteúdo de comando e pode ter de 13 até 248 bytes de extensão. Nela é inserida a informação que se deseja transmitir.

VII. CAMADA DE APLICAÇÃO

A camada de aplicação possui um controle de acesso ao meio (RAC) que é definido por uma série de comandos e respostas de alto nível para gerar operações em um nó escravo. Ele pode ler e escrever nas entradas e saídas, baixar ou atualizar a memória e controlar e monitorar as tarefas feitas por um escravo. Basicamente é uma especificação que se aplica para os tipos de mensagens. Esse diferentes tipos podem gerar comandos como: criar ou eliminar uma tarefa, resetar um escravo, baixar ou atualizar a memória, ler, escrever, atualizar ou realizar operações lógicas com as estradas e saídas e também ler e definir estados dos escravos. A tabela 3 mostra alguns dos comandos do RAC que podem ser utilizados. [1]

Tabela 3. COMANDOS DO RAC.

COMMAND	ACCESS	CONTROL	VALUE
Reset Slave		✓	00H
Create Task		✓	01H
Delete Task		✓	02H
Get Function ID		✓	03H
RAC Protect		✓	04H
Read I/O	✓		05H
Write I/O	✓		06H
Update I/O	✓		07H
Upload Memory	✓		08H
Download Memory	✓		09H
OR I/O	✓		0AH
AND I/O	✓		0BH
XOR I/O	✓		0CH
Status Read	✓		0DH
Status Write	✓		0EH
Reserved			0FH-0BFH
User Defined			0C0H-0FFH

O BAPI é uma interface para programadores de aplicativos que utilizam o protocolo BITBUS, feito para poder ser programado em várias linguagens como Pascal, Java ou, principalmente, em C. Os distribuidores do protocolo e suas placas devem suportar as bibliotecas do BAPI no topo dos seus drivers, podendo utilizar seus comandos. Esta arquitetura também permite com que várias aplicações utilizem o BITBUS ao mesmo tempo.[5]

VIII. CONCLUSÃO

O BITBUS é um protocolo que pode ser implementado e utilizado com baixo custo de manutenção devido à facilidade de conexão dos nós.

Comparada com outras redes, sua taxa de transmissão não é elevada, porém devido a que os pacotes utilizados no protocolo são enxutos, utilizam poucos bytes, é possível obter uma alta eficiência.

Suas conexões fornecem uma alta robustez para a implementação em ambientes industriais com aplicações como leitura de sensores e controle de atuadores.

IX. REFERÊNCIAS

- [1] Intel, "The BITBUS Interconnect Serial Control Bus Specification" Intel Corporation, <http://www.bitbus.org/dnl/BITBUS%20Interconnect%20Specification.PDF> (Acessado em: 30 de junho 2016).
- [2] G. Fernandes, M. Correia, S. Almeida, "Sistema de Instrumentação distribuído Multi-sensorial", <https://web.fe.up.pt/~ee96112/arquivo/relatorio.pdf>, (Acessado em: 30 de junho 2016).
- [3] Kunbus, "BITBUS - a field bus with enormous expansion possibilities", Kunbus Industrial Communication, <https://www.kunbus.com/bitbus.html> (Acessado em 27 de junho 2016).
- [4] J. Trethewey, "BITBUS, distribution control modules.", <http://www.alfirin.net/flamer/bitbus/> (Acessado em: 3 de julho 2016).
- [5] BEUG, "BITBUS, Application programmers interface (BAPI)", <http://www.bitbus.org/dnl/bapi.pdf> (Acessado em: 3 de julho 2016).

Interbus

Redes Industriais

Enrique Adami Geske

Engenharia de Controle e Automação
Universidade Federal de Santa Catarina - UFSC
Blumenau, Santa Catarina - Brasil
enrique_geske@hotmail.com

Resumo—No mundo atual, as redes industriais vieram para ficar. Com o intuito de facilitar e otimizar o processo fabril, hoje contamos com uma larga variedade de redes, das quais certas se destacam. A Interbus, nesses quase 30 anos de existência, dominou grande fatia desse mercado, devido a sua facilidade de instalação, manutenção e eficiência. Veremos nesse artigo várias características importantes acerca do funcionamento desse sistema e compararemos com outros também consolidados nesse mercado.

Palavras-Chave—Interbus; redes industriais; comunicação; barramento; dispositivo;

Abstract—In today's world, the industrial networks came to stay. Meaning to facilitate and optimize the manufacturing process, we now have a large variety of networks, of which, some stand above. Interbus in these almost 30 years of existence, dominated a big market share, due to its easy installation, maintenance and efficiency. In this article, we will see many important operation features of this system comparing to others that are also consolidated in this market.

Keywords—Interbus; industrial networks; communication; bus; device;

I. INTRODUÇÃO

O sistema Interbus foi apresentado em 1987 e tem como princípios a facilidade de instalação e manutenção e elevada eficiência de protocolo. É um sistema rápido, universal e aberto de bus sensor/atuador com topologia mestre-escravo. Foi idealizado e desenvolvido pela Phoenix Contact e, nesses quase 30 anos, conta com mais de 600 fabricantes de componentes e mais de 10 milhões de nós instalados. Se consolidou primariamente na indústria automobilística, mas hoje é utilizado em diversas aplicações de pequeno e grande porte. [1][2]

II. INTERBUS

A. Características Principais

Atualmente, a rede em chão de fábrica, tendo evoluído significativamente ao longo dos anos, conta com 8192 pontos de entrada e saída, com taxa de transmissão bruta de 500kbps no cobre e 2Mbps na fibra e, além disso, vários recursos de diagnóstico. Utiliza conectores 9-pin D-Shell e 23mm DIN circular e cabeamento de fibra óptica, infravermelho, par trançado, anel *slip* ou conexões SMG. [2][3]

A sua eficiência é devida a diversos fatores, sua arquitetura mestre-escravo e o método *one total frame*, de mensagem única, contribuem intensamente para isso, além de ter uma estrutura em anel, fazendo com que ele possa mandar e receber mensagens simultaneamente. [2][4]

Desde o mestre, todos os dispositivos são conectados ao barramento, cada um deles tem duas linhas distintas para transmissão de informação, uma que transmite para o dispositivo seguinte e uma para o anterior, fazendo com que seja eliminada a necessidade de uma linha de retorno do último ao primeiro dispositivo, algo que seria necessário em uma estrutura anel comum, por isso a Interbus é considerada, de um ponto de vista de instalação, uma rede com estrutura em árvore, pois há apenas um cabo que liga um dispositivo ao seguinte. [4]

O hardware pode ser facilmente especificado para a instalação de uma rede Interbus de acordo com as seguintes nomenclaturas: Local Bus (LB), Remote Bus (RB) e Remote Bus Branch (BA). [2]

B. Remote Bus

O nome Remote Bus é referente ao cabo que liga o mestre (controlador) até o último participante da rede, sendo a distância entre os participantes máxima de 400m em cabo de cobre e 3km em fibra óptica e um número máximo de 254 dispositivos, assim totalizando um comprimento máximo total de 12,8 e 80 quilômetros para os respectivos materiais (informações detalhadas estão tabeladas). [4]

Na rede Interbus, há um componente conhecido como *acoplador de rede (BK)*, por qual toda a conexão passa e o trecho entre dois acopladores é conhecido como *segmento de bus*. Essa definição contribui para a eficiência do reparo de erros na rede devido ao fato de que, se em uma instalação, com n participantes, houver uma falha na conexão entre a ilha 5 e 6, por exemplo, a comunicação da falha virá com um identificador de número 6, assim comunicando onde se encontra a falha. [2]

Outra característica interessante da rede é que, além de poderem ser atribuídos identificadores aos integrantes da rede, automaticamente, o primeiro integrante da rede é identificado como 1 e os demais seguem a contagem, facilitando a organização dos dispositivos conectados. [2]

C. Local Bus

Local Bus é o nome dado à ligação física entre os participantes da rede e os módulos de entrada e saída utilizados pelos mesmos, sendo que, na rede Interbus, o número máximo é de 63 módulos. [2]

Da mesma maneira que no Remote Bus, há uma identificação sequencial dos módulos conectados, sendo assim, o sétimo módulo do décimo quinto participante da rede terá a identificação 15.7, assim adicionando informações a própria identificação do RB e assistindo à identificação de erros, visto que se houver alguma falha em um módulo, além de a rede não precisar parar, haverá uma informação PF 15.8, ou seja, falha do periférico 15.8. [2]

D. Remote Bus Branch

É uma ramificação de remote bus. A ramificação é conectada ao remote por meio de um acoplador especial. Essa especificação de conexão permite utilizar a rede principal como um backbone e trata-la como uma rede em árvore, podendo chegar até 16 sub-derivações, ou seja, ramificações dentro de ramificações e assim sucessivamente. [2][4]

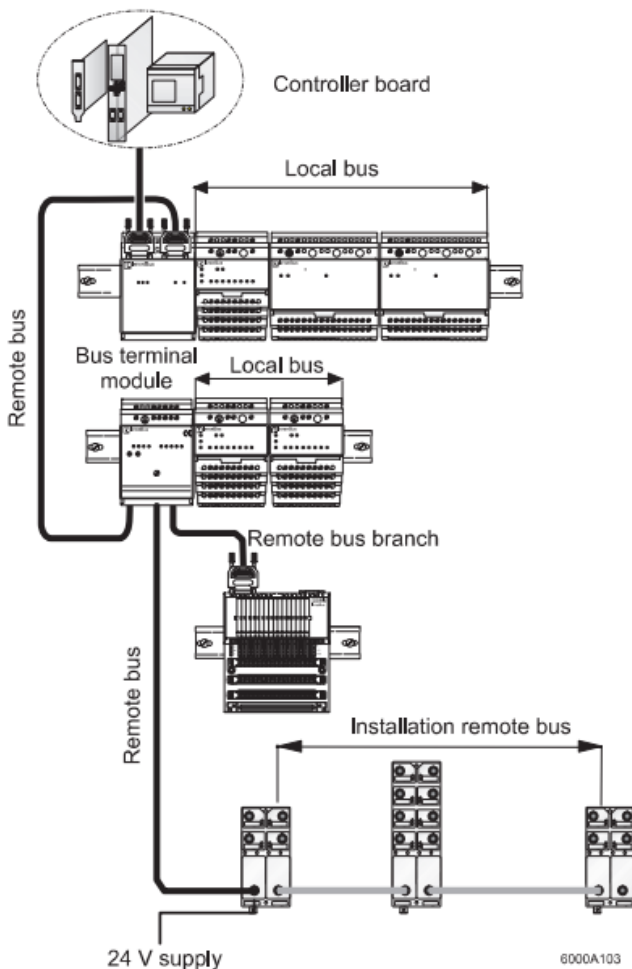


Fig. 1. Estrutura da rede. [5]

III. TRANSMISSÃO DE DADOS

O sistema Interbus utiliza comunicação serial para transmissão de dados, sendo que pode utilizar alguns materiais para tal fim, cobre, fibra, slip rings ou infravermelho. [4]

A. Protocolo Elétrico

A comunicação na rede Interbus foi baseada no protocolo elétrico RS 485 para o cobre e utiliza 5 fios conectados ponta a ponta entre o mestre e o primeiro participante e seu subsequente até o último elemento do RB. [2]

Desses, 4 fios são utilizados para a transmissão de dados, full-duplex (envia e recebe informações de modo síncrono) de forma diferencial e o outro como referência de potencial, operando em torno de 2,5 V. [2]

B. Cabeamento

O tipo padrão de cabeamento utilizado é o par-trançado de cobre, mas a rede Interbus pode utilizar outros tipos de meios, como fibra óptica, infravermelho e slip rings, fazendo com que a rede alcance lugares não acessíveis com o par trançado de cobre. [2,4]

O cobre pode alcançar até 400m por segmento de bus, e é utilizado principalmente para instalações indoor, subterrâneas e para aplicações flexíveis e/ou com partes móveis. [4]

A fibra é dividida em 3 tipos:

- Fibra de Polímero: Mais fácil de montar, opera com distâncias pequenas entre participantes;
- Cabo HCS: Híbrido entre fibra de vidro e polímero, difícil de ser montada e opera com distâncias médias;
- Fibra de Vidro: Muito complicada de montar, opera a grandes distâncias entre acopladores.

Os slip rings tem a função de transmitir dados para peças girantes e os transmissores infravermelhos convertem sinais elétricos em luz infravermelha eliminando a necessidade de cabeamento entre participantes da rede. [4]

TABELA 1: DISTÂNCIAS MÁXIMAS PARA CABEAMENTO [4]

Distâncias	
Do controlador ao último dispositivo	12,8 km, max. (Cobre)
	80 km, max. (Fibra de Vidro)
Do controlador ao primeiro dispositivo	400 m, max. (Cobre)
	50 m, max. (Fibra de Polímero)
	300 m, max. (Fibra HCS)
	3000 m, max. (Fibra de Vidro)
Entre dois dispositivos remote bus	400 m, max. (Cobre)
	50 m, max. (Fibra de Polímero)
	300 m, max. (Fibra HCS)
	3000 m, max. (Fibra de Vidro)

C. Multi-port Memory

É o nome dado ao espaço da memória principal do mestre no qual o programa irá realizar o processo de leitura dos sinais de entrada e escrita dos sinais de saída e, logo após, informa que pode ser iniciado o ciclo de transferência desses dados pelo princípio *shift register*. [2]

D. One-Total-Frame

O One-Total-Frame ou Summation-Frame Protocol tem como princípio somar, ou seja, juntar as informações de todos os dispositivos (sensores e atuadores) conectados à rede em uma única mensagem, fazendo com que os headers sejam utilizados apenas uma vez, assim aumentando de forma considerável o espaço útil de dados, visto que as informações recebidas de tais dispositivos têm tamanho muitas vezes menor que os overheads. [5]

Contudo, pode-se perceber que a eficiência desse protocolo aumenta com o número de entradas e saídas sendo utilizadas na rede, que, no caso da área industrial, é geralmente alto, fazendo com que haja uma combinação de uma baixa transmissão de dados com um throughput considerável. [5]

Com um simples exemplo, podemos demonstrar a eficiência desse método. Suponhamos uma rede com 32 dispositivos conectados cada qual representando 8 bits de dados a processar. Em um sistema simples orientado à mensagem, cada um desses 8 bits necessitaria de um overhead de em torno de 200 bits, fazendo com que apenas 4% da transmissão seja utilizado por dados enquanto no caso do summation-frame pode chegar à mais de 60%, combinando isso com uma taxa de transmissão de dados na ordem de Mbps, temos um sistema muito mais eficiente que a transmissão de mensagens únicas para cada dispositivo. [5]

E. Shift Register

Cada participante da rede Interbus tem dois registradores internos, um de informação e um de dados. A cada ciclo de bus, os novos dados “empurram” os dados antigos registro a registro para que cada dispositivo mande e receba informações a cada ciclo. [4]

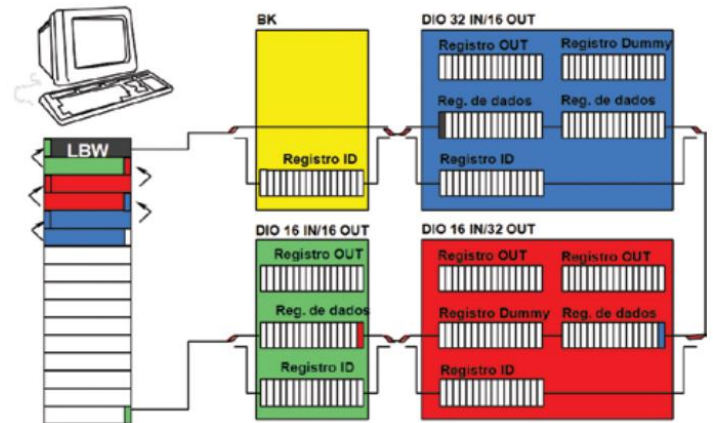


Fig. 3. Mecanismo Shift Register. [2]

Simplificando, o shift register transmite bit a bit os dados de um ponto a outro, ou seja, o bit do primeiro sinal passará registro a registro até chegar ao MPM, tendo como referência a Loop Back Word (LBW). [2]

F. Loop Back Word

Todo participante da rede Interbus, tem um ID code, nele é identificado o tipo do módulo e também qual o espaço a ser alocado na área MPM do mestre para realizar a troca de dados. [4]

No primeiro ciclo, logo quando a rede é energizada, será gerada uma palavra de 16 bits que será única em todo o ciclo de clock. Sendo assim, quando o gerenciador MPM identificar a Loop Back Word, ele dirá ao controlador que a transferência de dados pode ser iniciada, pois todos os dados estão prontos para alteração. [2][4]

G. PCP (Peripherals Communication Protocol)

Alguns dispositivos conectados à rede Interbus necessitam de alguma parametrização específica, tais como inversores de frequência, módulos de comunicação, etc. Quando isso acontecer, a rede utiliza o recurso de PCP, ou seja, protocolo de comunicação de periféricos, onde os pacotes de dados parametrizados são quebrados em pacotes menores e enviados aos periféricos durante o ciclo de identificação, fazendo assim, com que haja uma cópia tanto no dispositivo quanto no controlador, para que, quando haja necessidade de substituição do dispositivo ele não tenha que ser configurado novamente. [2]

H. Registradores

A rede Interbus conta com registradores específicos que guardam diagnósticos da rede e podem ser lidos e manipulados de acordo com certa necessidade, como local do erro, descrição do erro, performance e estatísticas, assim facilitando uma série de soluções a serem implementadas por alguém com certo conhecimento na rede Interbus. [2]

IV. MECANISMOS E RECURSOS

A. Grupos

Um dos recursos do Interbus é a criação de grupos lógicos de acopladores ou segmentos de bus, facilitando a configuração de certos parâmetros comuns entre participantes do grupo. Para

One-Total-Frame Protocol

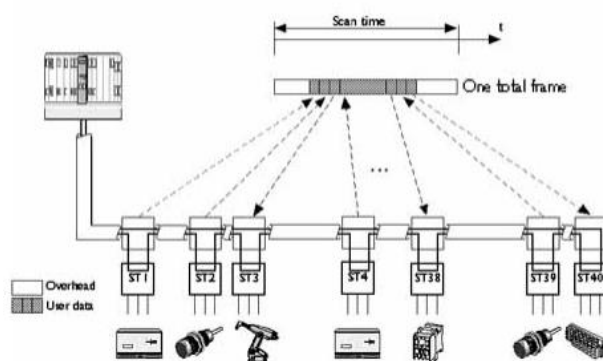


Fig. 2. Protocolo One-Total-Frame. [5]

tais, os acopladores ou segmentos podem estar próximos fisicamente ou não. O controlador pode ativar e desativar esses grupos de acordo com a necessidade, ou configuração inicial, sem interferir no funcionamento/execução da rede de forma alguma. [2]

Há também o conceito de grupos alternativos, esses à nível de acoplador, onde um pode ter mais de uma configuração de entradas/saídas, fazendo com que, de acordo com alguma necessidade específica, o conjunto de I/O pode ser trocado, fazendo com que outro local bus assuma fisicamente o acoplador. Esse processo ocorre, por exemplo, quando um robô necessita de diferentes ferramentas para certa operação. Essa troca ocorre também sem interferir no funcionamento da rede. [2]

B. CMD (Configuring Monitoring and Diagnostics)

No Interbus é possível desconectar os módulos da rede conectados a um acoplador logicamente. Em caso de defeito de algum módulo específico, esse recurso poder vir a ser muito interessante, visto que pode ser feito a manutenção dos módulos sem interferir no restante do processo, desde que a operação realizada por tal módulo não seja vital para o mesmo, porém, essa desconexão deve ser prevista na concepção da rede. E o programa de parametrização que realiza essa desconexão é chamado de CMD. [2]

É permitida também a operação de bypass, onde um trecho inteiro do bus, ou seja, mais de um acoplador é desconectado. Primeiramente o CMD desconecta logicamente um acoplador ligando o anterior diretamente no seguinte, fazendo com que todo o trecho desconectado seja interpretado como um cabo direto. Essa operação é especificamente muito interessante para a manutenção da rede em situações que ela não pode ser feita de imediato. [2]

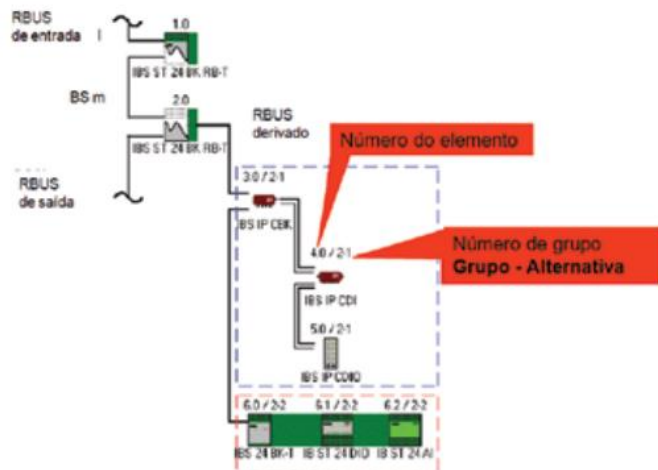


Fig. 4. Grupos de Acopladores. [2]

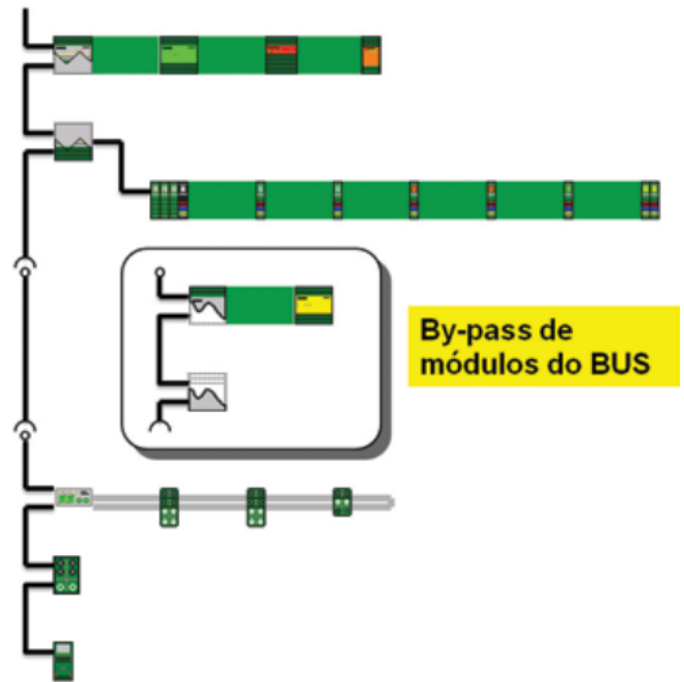


Fig. 5. Bypass de módulos do bus. [2]

V. COMUNICAÇÃO ENTRE REDES

A rede Interbus é intrinsecamente preparada para a conectividade com outras redes. A comunicação com as mesmas pode acontecer em qualquer direção, ou seja, entre mestres Interbus, realizando a troca de informações paralelas, entre um mestre Interbus e um As-i, de cima para baixo, ou de baixo para cima, no caso da troca de informações de com uma rede Ethernet.

A. Comunicação entre Redes com Mestres Distintos

A troca entre rede com mestres distintos, ou paralela, é quando um mestre InterBus troca informações com outra rede que possui um mestre, por exemplo Profibus (Interbus-Profibus), ou mesmo outra Interbus (Interbus-Interbus). A comunicação é feita por um módulo que faz a ligação entre os mestres, que pode ser interpretado como um hardware que é um lado escravo Profibus (por exemplo) e do outro escravo Interbus. É necessário também um espaço de memória comum previamente reservado durante a configuração de ambas as redes, assim os mestres receberão a mesma informação para leitura e escrita nos diversos periféricos. Isso é possível mesmo devido a diferença entre as redes, mas necessário mesmo com uma ligação Interbus-Interbus. [2]

Sendo o Profibus uma rede semelhante e principal concorrente ao Interbus, podemos listar certas diferenças entre as mesmas, tanto nas topologias quanto no desempenho e eficiência de cada uma delas. [3]

Segue uma tabela com as principais diferenças:

TABELA 2: DIFERENÇAS ENTRE INTERBUS E PROFIBUS [3][4]

Característica	Interbus	Profibus
Topologia da Rede	Segmentada com "T" drops	Linha, Estrela e Anel
Meio Físico	Par trançado, Fibra, slip-ring e Infra-vermelho	Par trançado ou Fibra
Max. Nós	256 nós	127 nós (124 escravos - 4 seg, 3 repetidores) + 3 mestres
Max. Distancia	12,8 km no cobre, 400m entre segmentos	100m entre segmentos, 24km na fibra
Métodos de Comunicação	Mestre/Escravo com One-Total-Frame	Mestre/Escravo, Ponto a Ponto
Propriedades da Transmissão	500 kbps, full duplex	DP: 9.6 kbps à 12Mbps PA: 31.25 kbps
Tamanho de Transferência de Dados	1-64 Bytes dados 246 Bytes Parâmetros 512 Bytes h.s., bloco ilimitado	0-244 Bytes
Checkagem de Erros	16-bit CRC	HD4 CRC

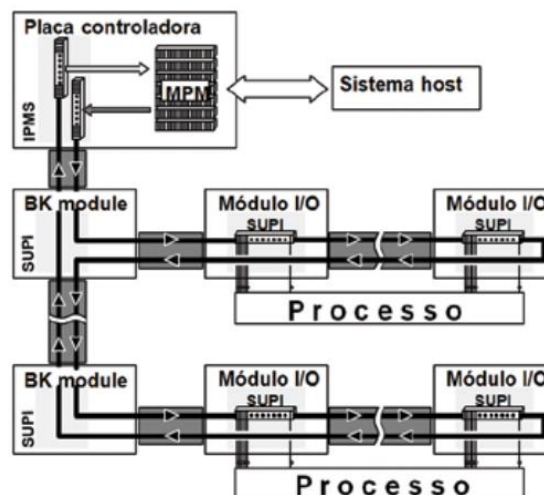


Fig. 6. Esquema de comunicação com outras redes. [2]

VI. CONCLUSÃO

Hoje em dia a variedade de redes industriais oferecidas no mercado é consideravelmente maior que tempo atrás. Com o avanço da tecnologia buscou-se melhorar o processo fabril em diversos aspectos. Dentre as redes existentes, destaca-se o Interbus, que com quase 30 anos, ainda detém uma vasta fatia desse mercado de redes.

O Interbus é um protocolo eficiente, rápido, com uma grande flexibilidade em vários quesitos, seja ele a mídia, já que há a possibilidade de, em uma mesma rede, haver fibra, cobre, infravermelho e *slip-ring*, seja na construção da rede ou até a comunicação com outras redes, além dos recursos que tornam a conexão dos módulos e a comunicação muito mais simples, como o Shift Register e o endereçamento sequencial dos módulos. Por esses e outros fatores a rede é considerada uma das mais eficazes para instalação, manutenção e utilização nos dias de hoje.

B. Comunicação entre Interbus e As-i

Quando para certa aplicação, já existe o protocolo As-i configurado nos dispositivos que serão utilizados, é simples a comunicação das informações para o mestre Interbus. O que é feito é a conexão de um mestre As-i ao acoplador Interbus e o mesmo é configurado como um módulo que lê os agregados elétricos e manda as informações indexadas à uma área reservada da memória interna do módulo. [2]

Como normalmente na rede Interbus, a transferência das informações para o mestre Interbus é controlada pelo PCP (Peripherals Communication Protocol), garantindo sincronia das mesmas quando mandadas e quando recebidas. Essa comunicação se trata de uma comunicação de cima para baixo. [2]

C. Comunicação entre Interbus e Ethernet

Temos também outra possibilidade de comunicação entre diferentes redes, essa baseada na troca de informações verticais, onde uma base de dados, como o Ethernet, que é orientada a bit, compõe um frame, baseado nas informações disponibilizadas na área de memória de um hardware comumente conhecido como Proxy de Interbus para Ethernet. [2]

Uma das principais diferenças entre essa comunicação e a anterior é o sincronismo. Nessa o tempo de ciclo da rede Interbus é mantido, mas é independente da taxa de transferência do pacote de dados para a rede Ethernet, o qual depende do comportamento da mesma. [2]

REFERÊNCIAS

- [1] PHOENIX CONTACT | INTERBUS [Site]
https://www.phoenixcontact.com/online/portal/br?1dmy&urilc=wcm:pat h:brpt/web/main/products/subcategory_pages/Interbus_p-04-11/a20f6e69-4460-457b-9267-2aea300acd7c
Acessado em: Jun, 23, 2016
- [2] Marcio M. Vaz, "20 anos de Interbus e 10 milhões de nós depois!" [Site]
http://pd7.com.br/v2/images/stories/Artigos_tcnicos/br_CII139.pdf
Acessado em: Jun, 23, 2016
- [3] J. M. Barbosa, F.A. Silveira, "Comparação entre as Redes Industriais: Profibus e Interbus" [Site]
Acessado em: Jun, 23, 2016
- [4] INTERBUS | Installation Manual [Site]
https://www.phoenixcontact.com/assets/downloads_ed/global/web_dwl_technical_info/Interbus.pdf
Acessado em: Jun, 24, 2016
- [5] Interbus System Technology Basics - Steven Engineering [Site]
http://stevenengineering.com/Tech_Support/PDFs/67SYSTECH.PDF
Acessado em: Jun, 27, 2016

Índice de Autores

Akerman, Marco	1
Alfredo Nack, Felipe	24
Almeida, Leonardo	112
Antero, Nicolas	18
August Roehrig Domingues Dos Santos, Stephan	75
Azzolini Vieira, Lucas	24
Baggio, Ronaldo	158
Battiston, Tiago	13
Belck, Lukas	18
Bergmann, Erick	1
Bloemer, Martha	41
Bubeck, Bruno	30
Cogliatti, Pedro Arthur	94, 119
Correa de Borba, Christian	82, 98
Da Costa Neto, Júlio Joel	119
Davanço, Gabriel	36
Demarchi, Diogo Luiz	79, 170
Dilly, Guillermo	13
Dos Santos, Stephan August R. D.	90
Fernandes, Cecili Duwe	53
Ferraz, Gabriel	18
Ferreira, João Victor	18
Floriani, Rodolfo	6
Freitag Binsfeld, Allyson	24
Geske, Enrique	79, 186
Geske, Guilherme	30
Godoy, João Gabriel	112
Guimarães, Janaina Gonçalves	82
Hammes, Nikolas	125
Hammes, Nikolas Sbaraini	106
Hasse, Bruno Leone	47
Heil, Allan	30
Hermes Anschau, Martin	98

Kriek, Gabriela	157
Lange, Henrique	1
Lauth, Artur Henrique	106
Liao, Douglas	41
Lobato, Rafael	36
Luiz Granemann, André	98
Luz, Gustavo	13
Mafra, Gabriel	41
Mailer, Christian	47
Manerichi, Guilherme Moser	90, 164
Martins, Maria	13
Masson, Juliano E. N.	69, 130
Masson, Juliano Emir Nunes	90
Menon, Bianca	82
Milheiro, Arthur	13
Moser Manerichi, Guilherme	64
Novelletto, Carlos	1
Obenaus, André Filipe	47, 119
Outa Umezawa, Ciro	94
Pacher Júnior, Rômulo	86
Pacher, Rômulo	75, 175
Paloschi, Gustavo	24
Pinto, Bruna	41
Reuter, Andre Luiz De Sousa Marcondes	69, 151
Reuter, André Luiz De Sousa Marcondes	86
Ribeiro, Carlos	30
Rodrigo, Alvaro	18
Sanches, Caique	79, 94, 181
Santana, Jorge	137
Santos, Jeane Almeida	53
Schiochet, Letícia Terezinha Farias	53
Schwank, Herman	1
Siegel, Arthur Phillip B.	106
Silva, Elton	36
Silva, Fellipe E G	169

Silvestre, Gabriel A.	58, 144
Sonohara, Eduardo	112
Vanelli, Brunno	64, 86, 136
Vieira de Azevedo, Nathã	24
Weber, Rodrigo	125
Wippel, Arthur	30
Zimmer, Gabriel	41

Índice de Palavras

Acidentes	1
Ad-hoc	157
Amplificador	75
Aprender fazendo	119
Arbitration	151
Arduino	1, 6, 13, 18, 24, 30, 36, 41, 53
Audio Mixer	58
Automatizado	53
Automação	13, 24, 41, 98, 112, 130, 175
Banco de dados Mysql.	18
Barramento de Campo	175
Batidas	13
BITBUS	181
bluetooth	137
Buck converters	69
Bus	186
CAN	151
Choque	79
CI555	58
Circuito	79
Circuito Impresso	75
Clipping	64
CLP	98, 106, 119
CNC	6
Communication	186
Comunicação	170
Comunicação	181
Controle	47, 119, 125
Coração	36
Cruzamento Controlado	86
Defesa	79
Device	186
Diagrama de estados	106
Distorção	64
Domótica	13, 24

Electronic Piano	58
Eletrônica	79
Embarcados	36
Ensino prático	119
Estufa	47
Ethernet Industrial	130
Fabricação de refrigerante	112
Fazendas urbanas	47
Fechadura	13
Fechaduras inteligentes.	18
Ferroviário	90
FPGA	82
Funcionamento	6
HART	169
IEEE 802.15.1	137
IEEE 802.15.4	136
Industrial Network	151, 186
Indústria	170
Informática Industrial	112
Integrated circuit	82
Interbus	186
Interface	170
ISO11898	151
KA7808	69
ky039	36
LM2676-ADJ	69
Líquidos	98
MEGA	36
Microcontrolador	24
Misturador	98
Modelagem	90
Modelagem	94, 119
Monitoramento Cardíaco	36
Motor	125
Multi-Mestre	164
Multi-Net	164
Neuron network	82

NRF24101+	36
Overdrive	64
P-Net	164
Passeio dos Filósofos	94
Plantio urbano	47
Porta	13, 18, 41
Problema de sincronização	94
PROFIBUS	175
PROFINET	130
Programação	36, 106
Protocolo	170
Protocolo de comunicação	144
Prototipagem	1
Protótipo	30
Realimentado	125
Rede	170
Rede AS-I	158
Redes Automotivas	144
Redes de Petri	86, 90, 94
Redes Industriais	130, 158, 169, 175, 181
Redes Veiculares	157
RSSF	136
Saúde	36
Segurança	1, 24, 30
Sensores	24, 41, 98, 106
Siemens	170
Sinec	170
Sistema de Irrigação	53
Sistema de Nível	119
Sistemas de Automação	158
Sistemas Intraveiculares	144
Som Stereo	75
Stun gun	79
TDA 2050	75
TINA	86
Tina	94
Token Ring Virtual	164
Tráfego	1

UNO	36
VANET	157
Voltage regulator	69
voltagem	79
wireless network	137
WIRELESSHART	169
Zigbee	136
Áudio	75