

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTOS ACADÊMICOS DE ELETRÔNICA E MECÂNICA  
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

ADEMILSON RODRIGO MOREIRA  
MARCOS CLACEDIR LOSS

**DESENVOLVIMENTO DE UM CONTROLADOR INDUSTRIAL PARA  
USO EM PROCESSOS INDUSTRIAIS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA  
2013

ADEMILSON RODRIGO MOREIRA  
MARCOS CLACEDIR LOSS

## **DESENVOLVIMENTO DE UM CONTROLADOR INDUSTRIAL PARA USO EM PROCESSOS INDUSTRIAIS**

Trabalho de Conclusão de Curso de graduação, apresentado ao Curso Superior de Tecnologia em Mecatrônica Industrial, dos Departamentos Acadêmicos de Eletrônica e Mecânica, da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Sergio Leandro Stebel

CURITIBA  
2013

## **TERMO DE APROVAÇÃO**

ADEMILSON RODRIGO MOREIRA  
MARCOS CLACEDIR LOSS

### **DESENVOLVIMENTO DE UM CONTROLADOR INDUSTRIAL PARA USO EM PROCESSOS INDUSTRIAIS**

Este trabalho de conclusão de curso foi apresentado no dia 05 de Setembro de 2013, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. O aluno foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof. Dr. Milton Luiz Polli  
Coordenador de Curso  
Departamento Acadêmico de Mecânica

---

Prof. Sergio Moribe  
Responsável pelo Trabalho de Conclusão de Curso  
Departamento Acadêmico de Eletrônica

#### **BANCA EXAMINADORA**

---

Prof. Dr. Eduardo Giometti Bertogna  
UTFPR

---

Prof. MSc. Anderson Levati Amoroso  
UTFPR

---

Prof. Dr. Sergio Leandro Stebel  
Orientador - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

## **AGRADECIMENTOS**

Agradecemos nossos familiares pelo apoio e dedicação voltados aos nossos estudos e crescimento acadêmico, profissional e pessoal.

Agradecemos nossos amigos e professores que nos ajudaram direta ou indiretamente na realização deste trabalho de conclusão de curso.

E agradecemos especialmente ao Prof. Dr. Sérgio Leandro Stebel pela sua orientação e dedicação relacionada à esse trabalho.

## RESUMO

Moreira, Ademilson Rodrigo; Loss, Marcos Clacedir. **Desenvolvimento de um controlador industrial para uso em processos industriais**. 2013. 77 p. Trabalho de Conclusão de Curso (Graduação) – Curso de Tecnologia em Mecatrônica Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2013.

Este projeto apresenta o desenvolvimento de um controlador industrial de processos com PID interno, capaz de monitorar e atuar sobre um sistema. Foram determinadas as funções do controlador, similar ao do modelo Novus 1100. Depois de selecionados os componentes necessários, foram projetadas as placas de circuito impresso – observando os requisitos de cada CI através dos respectivos *datasheets* – foi então montado o *hardware*. O sistema é controlado por um DSPic 30F2010 programado em linguagem C. O *hardware* é protegido por uma chassi de metal que foi projetada utilizando-se um *software* CAD. Para a fabricação dos itens foram feitos orçamentos com fornecedores e analisados os processos de fabricação. A utilização do controlador desenvolvido tem o intuito de oferecer um controle eficiente e seguro de baixo custo, garantindo o monitoramento e atuação sobre o sistema.

### Palavras-chave

Controle. PID. Processos.

## ABSTRACT

Moreira, Ademilson Rodrigo; Loss, Marcos Clacedir. **Development of an industrial controller to use in industrial process**. 2013. 77 p. *Monograph (Graduation) – Industrial Mechatronic Technology, UTFPR, Curitiba, 2013.*

This project presents a development of an industrial controller (PID). The controller's functions were based on Novus 1100 controller. The printed circuit boards were designed observing the IC's requirements based on datasheets. The system is controlled by DSPic 30F2010 programmed in C language. The hardware is protected by metal body designed in CAD software. The PID controller presented has low cost.

### **keywords**

Controller. PID. Process.

## LISTA DE FIGURAS

Figura 1 - Diagrama de blocos de um microcontrolador.....	15
Figura 2 - Diagrama de blocos de um controlador digital de sinais. ....	17
Figura 3 - Tipos de furação em chassi. ....	18
Figura 4 - Filtro RC.....	19
Figura 5 - Circuito Snubber. ....	19
Figura 6 - Analógico periódico, sinal digital, analógico não periódico. ....	20
Figura 7 – Representação do sinal analógico e o digital equivalente.....	20
Figura 8 - Diagrama de uma conversão A/D com seus principais processos.....	21
Figura 9 - Resposta do Controle Proporcional a partir de um erro. ....	24
Figura 10 - Resposta do controle Proporcional Integral a partir de um erro.....	25
Figura 11 - Resposta do controle Proporcional Derivativo a partir de um erro. ....	26
Figura 12 – Controle em Malha Fechada. Fonte: Autoria própria.....	27
Figura 13 - Principais sinais no controle de processos. Fonte: Autoria Própria.....	28
Figura 14 – Algoritmo PID série ou interativo.....	29
Figura 15 - Algoritmo PID série. ....	29
Figura 16 – Algoritmo PID paralelo alternativo. ....	29
Figura 17 - Conexões elétricas do controlador N1100 da Novus. ....	31
Figura 18 – Diagrama do Menu.....	32
Figura 19 - Visão geral do controlador. ....	33
Figura 20 - Diagrama em blocos do sistema. ....	34
Figura 21 - Circuito do microcontrolador (CPU). ....	36
Figura 22 - Circuito da interface (I/O digitais e de corrente; PT100; PWM).....	36
Figura 23 - Circuito do PT100 e RCV420 (Entrada de corrente).....	37
Figura 24 - Circuito do PWM.....	38
Figura 25 - Conectores de interface do AD420 (Saída de corrente). ....	38
Figura 26 - Circuito do AD420 (Saída de corrente). ....	39
Figura 27 - Circuito das saídas digitais (Relés).....	39
Figura 28 - Circuito das entradas digitais. ....	40
Figura 29 - Circuito do painel (Placa CPU).....	41
Figura 30 - Circuito do painel (Placa do painel).....	41
Figura 31 – Código responsável pelo menu.....	44
Figura 32 - Trecho onde um ponteiro é declarado. ....	44
Figura 33 - Rotina de inicialização do controle.....	45
Figura 34 - Cálculo do sinal de saída do controle. ....	46
Figura 35 - Chassi metálico do controlador PID. ....	47
Figura 36 - Verificação do funcionamento do hardware no <i>proto-board</i> .....	49
Figura 37 - Placas de circuito impresso.....	50
Figura 38 - Montagem do hardware. ....	50
Figura 39 - Vista frontal e da borneira do controlador. ....	51
Figura 40 - Diagrama em blocos (Controle em malha aberta).....	52
Figura 41 - Resposta ao degrau no controle Proporcional (Controlador). ....	53
Figura 42 - Resposta ao degrau no controle Proporcional (Simulink). ....	53
Figura 43 - Resposta ao degrau no controle Proporcional-Integral (Controlador). ....	54
Figura 44 - Resposta ao degrau no controle Proporcional-Integral (Simulink). ....	54
Figura 45 - Resposta ao sinal de rampa no controle Proporcional-Derivativo (Controlador).....	55

Figura 46 - Resposta ao sinal de rampa no controle Proporcional-Derivativo (Simulink) .....	55
---------------------------------------------------------------------------------------------	----



## LISTA DE ABREVIATURAS

AD	Analógico Digital
BP	Banda Proporcional
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
CI	Circuito Integrado
DA	Digital Analógico
DSC	<i>Digital Signal Controller</i> (Controlador Digital de Sinais)
DSP	<i>Digital Signal Processor</i> (Processador Digital de Sinais)
EMI	<i>Electromagnetic Interference</i> (Interferência Eletromagnética)
EFT	Electrical Fast Transient (Transientes Elétricos Rápidos)
ESD	Electrostatic Discharge (Descarga Eletrostática)
IHM	Interface Homem Máquina
LED	<i>Light Emitting Diode</i> (Diodo Emissor de Luz)
P	Proporcional
PCB	<i>Printed Circuit Boards</i> (Placa de Circuito Impresso)
PD	Proporcional Derivativo
PI	Proporcional Integral
PID	Proporcional Integral Derivativo
PV	Variável Controlada
PWM	<i>Pulse Width Modulation</i> (Modulação por Largura de Pulso)
SP	Set-point

## LISTA DE SIMBOLOS

$e(t)$	Sinal do erro em função do tempo
$K_d$	Ganho derivativo
$K_i$	Ganho integral
$K_p$	Ganho proporcional
$r(t)$	Sinal de referência em função do tempo
$T_d$	Tempo derivativo
$T_i$	Tempo integral
$u(t)$	Sinal de saída do controlador em função do tempo
$y(t)$	Sinal de saída do sistema em função do tempo

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 PROBLEMA.....	13
1.2 JUSTIFICATIVA .....	13
1.3 OBJETIVOS .....	14
1.3.1 Objetivo Geral.....	14
1.3.2 Objetivos Específicos .....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>15</b>
2.1 DSPs E DSCs .....	15
2.3 CONVERSORES AD E DA.....	20
2.4 CONTROLADOR PID.....	22
2.4.1 Ações de controle .....	23
2.4.2 Controle com realimentação .....	27
2.4.3 Arranjo estrutural .....	28
<b>3 MÉTODO DE PESQUISA</b> .....	<b>30</b>
3.1 ANÁLISE DE MERCADO .....	30
<b>4 DETALHAMENTO TÉCNICO</b> .....	<b>32</b>
4.1 VISÃO GERAL DO PROJETO .....	32
4.1.1 Interface com o usuário .....	33
4.1.2 Diagrama em blocos do sistema.....	33
4.1.3 Descrição dos blocos.....	34
4.2 DESCRIÇÃO FUNCIONAL DOS BLOCOS.....	35
4.2.1 Diagrama esquemático .....	35
4.2.2 DSPic (CPU).....	35
4.2.3 Interface.....	36
4.2.4 PT100 e RCV420.....	37
4.2.5 PWM.....	37
4.2.6 AD420 (saída de corrente).....	38
4.2.7 Relés.....	39
4.2.8 Entradas digitais .....	40
4.2.9 Painel.....	40
4.3 PROGRAMAÇÃO .....	42
4.4 CHASSI DO CONTROLADOR PID .....	47
<b>5 DESCRIÇÃO DA MONTAGEM</b> .....	<b>49</b>
<b>6 TESTES E RESULTADOS</b> .....	<b>52</b>
6.1 CONTROLE PROPORCIONAL.....	52
6.2 CONTROLE PROPORCIONAL INTEGRAL.....	53
6.3 CONTROLE PROPORCIONAL DERIVATIVO .....	55
<b>7 CONCLUSÃO</b> .....	<b>56</b>
<b>REFERÊNCIAS</b> .....	<b>57</b>
<b>APÊNDICE A</b> .....	<b>59</b>
<b>APÊNDICE B</b> .....	<b>66</b>
<b>APÊNDICE C</b> .....	<b>68</b>
<b>APÊNDICE D</b> .....	<b>76</b>

# 1 INTRODUÇÃO

As indústrias que utilizam qualquer espécie de controle, tanto de processos discretos como contínuos, necessitam de equipamentos capazes de executar funções automáticas. Por exemplo, as indústrias petrolíferas que abrangem processos físico-químicos de transformação da matéria-prima em derivados, necessitam de uma grande estrutura de automação e controle dos seus processos de fabricação.

Justifica-se a ampla utilização desses controladores e processos automatizados numa indústria de grande porte, pois é necessária uma alta produtividade e com alta qualidade. O faturamento de uma companhia desta magnitude possibilita o investimento em melhoria contínua, adquirindo novos equipamentos e novos sistemas cada vez mais sofisticados.

As pequenas indústrias, por sua vez, permanecem na sua área de negócio com seu produto, geralmente, por satisfazer a necessidade dos clientes. Porém, isto acarreta em um comodismo, pois o fabricante não desenvolve melhorias em seus processos uma vez que o cliente está satisfeito com o produto adquirido.

Devido ao alto custo de mão-de-obra ou consultoria especializada na utilização de um controlador de processos, as empresas encontram barreiras para a correta utilização destes equipamentos.

O primeiro controlador PID data de 1890, baseado num projeto do governo americano para guiar automaticamente um navio. Na sequência, foram criados vários modelos de navios guiados automaticamente. A primeira publicação contendo análise teórica de um controlador PID foi feita por Nicolas Minorsky em 1922 (OGATA, 2003, p.57). Inicialmente os controladores eram pneumáticos, hidráulicos ou mecânicos e posteriormente passaram a ser elétricos (FONSECA, 2004, p.32).

O controlador de processos pode ser entendido como um dispositivo configurável capaz de supervisionar uma variável física (pressão, temperatura, nível, etc.) e controlá-las através de variáveis manipuladas (tensão, corrente, etc.).

## 1.1 PROBLEMA

Para que pequenas e médias empresas passem a implantar controladores de processos em seus sistemas, é necessário apresentar para seus gestores os recursos e benefícios que esses equipamentos oferecem. Uma gama considerável de empresas de diversas áreas utilizam como solução em seus sistemas apenas dispositivos com controle em malha aberta ou, no máximo, com controle *on-off* (liga e desliga o atuador de acordo com limites pré-estabelecidos). Este cenário nos dias de hoje é justificado pelo alto custo da mão-de-obra especializada na implantação correta de controladores industriais de processo, além da falta de conhecimento por parte dos gestores ou técnicos, da existência de controladores PID, que oferecem recursos de sintonização automática ou manual, deixando os sistemas mais rápidos e eficientes.

Assim como em outras áreas de negócio, os valores dos produtos são elevados quando a procura é maior do que a demanda, e a falta de concorrência contribui para que fabricantes mantenham os preços altos. Os controladores de processos não possuem componentes caros no seu sistema eletrônico em geral são componentes simples e de fácil aquisição, tais como o microcontroladores, relés, transformadores e elementos de valor insignificante como resistores e capacitores.

## 1.2 JUSTIFICATIVA

A falta de conhecimento por parte dos gestores e técnicos de uma empresa, impossibilita o desenvolvimento de sistemas realimentados. Muitas vezes existem produtos que têm espaço no mercado por possuir um baixo custo. Um exemplo seria um condicionador de ambiente com controle de temperatura com custo tão baixo quanto um simples aquecedor que utiliza um termostato para desligar o atuador quando a temperatura é atingida. Mas pelos motivos já citados, este produto não possui um controlador *on-off* interno, ou algum tipo de controle mais robusto capaz de tornar o produto mais atraente no mercado.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

Desenvolver um controlador PID capaz de monitorar e atuar em processos industriais e que permita ao usuário configurar as entradas e saídas de controle via uma IHM (Interface Homem-Máquina) com teclado frontal.

### 1.3.2 Objetivos Específicos

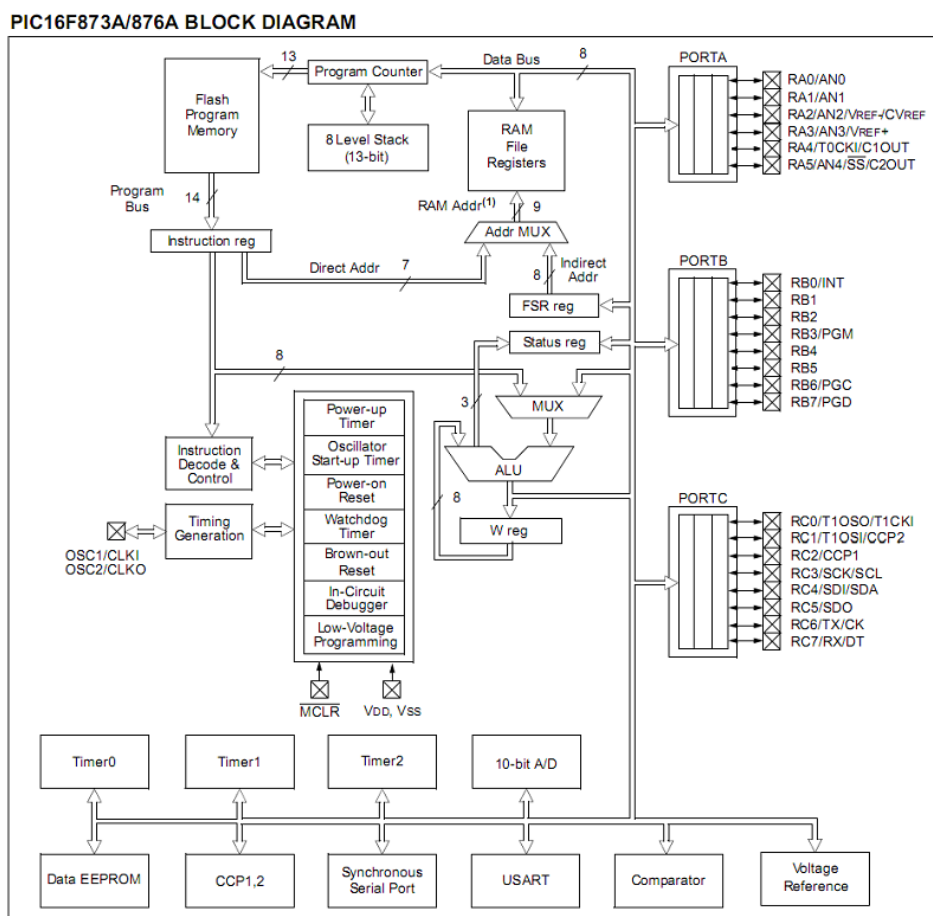
- Definir os componentes que serão utilizados no *hardware*.
- Elaborar os *layouts* das placas de circuito impresso.
- Criar um *hardware* com portas de comunicação, receptores e transmissores de sinais e demais recursos capazes de dar suporte ao *firmware* para que o controlador de sinais digitais possa processar as informações adquiridas pelo sistema.
- Construir a estrutura mecânica capaz de envolver e proteger o *hardware*.
- Desenvolver um *firmware*, escrito na linguagem C, capaz de controlar periféricos de um DSC (Controlador Digital de Sinais) com PID interno.
- Testar e validar o PID em malha aberta.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 DSPs e DSCs

Os microprocessadores, microcontroladores, Processador Digital de Sinais (DSPs - *Digital Signal Processor*) e Controlador Digital de Sinais (DSCs - *Digital Signal Controller*) são os vários termos modernos para os circuitos integrados com poder de processamento de tarefas. Estes componentes possuem algumas diferenças e similaridades bem sutis.

O microprocessador surgiu como sendo o cérebro dos computadores domésticos e possui grande poder de processamento. Trata-se apenas do núcleo de processamento, precisando de periféricos como barramentos de comunicações e portas de entrada e saída para obter, armazenar e devolver os dados processados. Com o surgimento do microcontrolador, os periféricos forma integrados com o microprocessador, conforme ilustrado na Figura 1.



**Figura 1 - Diagrama de blocos de um microcontrolador.**  
**Fonte: MICROCHIP, 2003, p.6.**

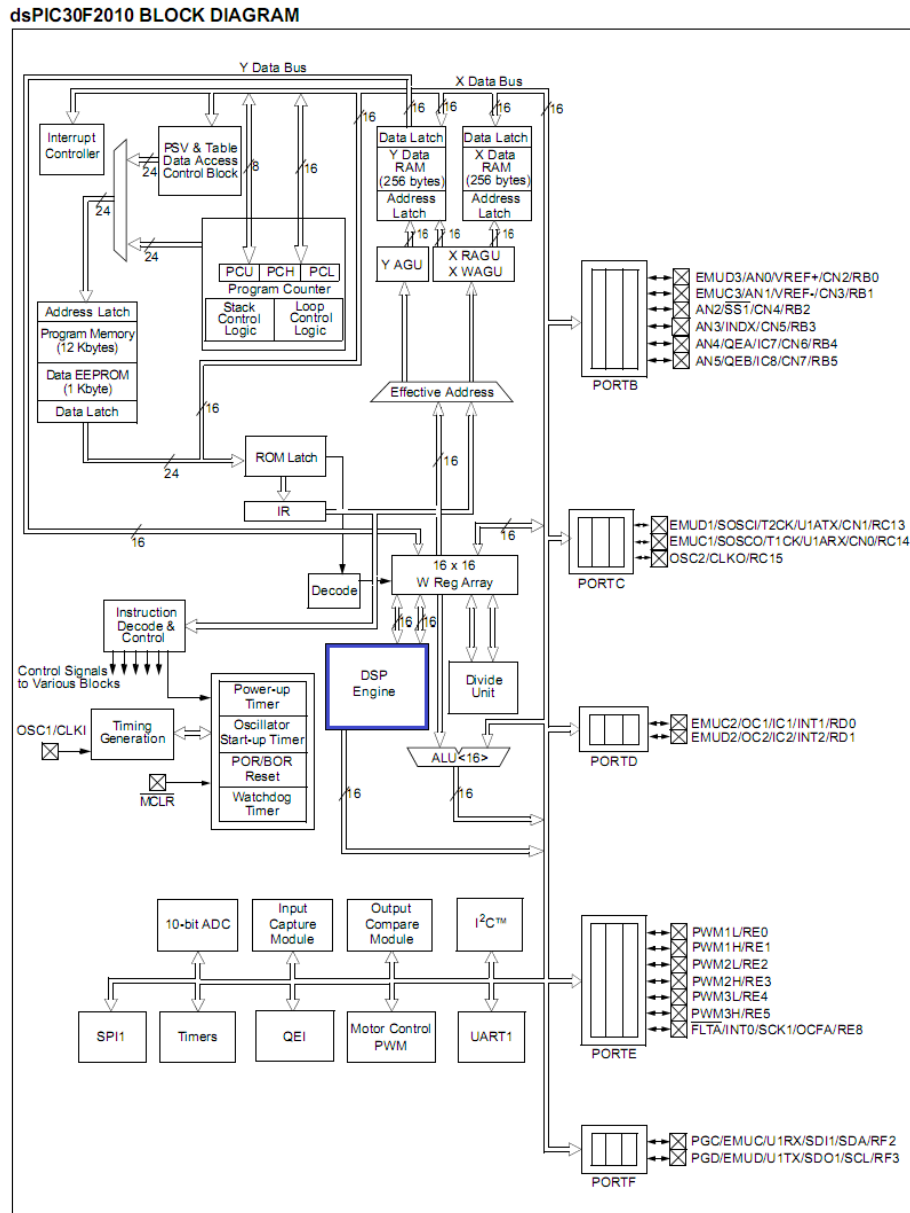
Após a evolução da microeletrônica, alguns equipamentos já miniaturizados precisavam de um pequeno processamento, a união de vários circuitos integrados digitais especiais elevaria o custo do equipamento. A partir desse problema, e também por já existirem os microprocessadores, surgem os microcontroladores, que seriam os equivalentes a um microprocessador rodeado pelos seus periféricos essenciais, mas com um poder de processamento muito menor. Com isso, surgem os primeiros equipamentos automatizados e “inteligentes”.

Os DSPs são um pouco diferentes na sua arquitetura, *hardware*, *software* e no conjunto de instruções. Otimizados em relação aos processadores comuns, os DSPs necessitam ter um ótimo processamento de sinais digitais. Esse tipo de tratamento exige um alto desempenho para aplicações numéricas em tempo real (termo que define aplicações aparentemente simultâneas quando um evento é disparado, por exemplo, a sensação de uma lâmpada acender imediatamente depois de pressionado seu interruptor elétrico), tarefas que seriam impossíveis serem desempenhadas por circuitos analógicos (OLIVEIRA, 2006, p.34).

Estes dispositivos são projetados para trabalhar com várias manipulações e transformações matemáticas, a partir de informações adquiridas do mundo real (analógico). Para isso utiliza-se o conversor analógico-digital (A/D), que captura os sinais contínuos e os converte em dados digitais. Uma vez que estes dados digitais estejam dentro do sistema, efetuam-se operações e cálculos que levarão a um novo dado digital que passará por um conversor digital-analógico (D/A) e será entregue ao meio em tempo real (OLIVEIRA, 2006, p.35).

Este processamento rápido foi capaz graças à incorporação de um módulo conhecido como MAC. Trata-se de um *hardware* (interno ao DSP *Engine*, conforme destacado na Figura 2) capaz de multiplicar e acumular os valores operados em apenas um ciclo de máquina, o que era impossível na abordagem antiga dos processadores, pois a multiplicação era realizada por somas e deslocamentos de bits sucessivos (OLIVEIRA, 2006, p.36).





**Figura 2 - Diagrama de blocos de um controlador digital de sinais.**  
**Fonte: MICROCHIP, 2008, p.8.**

Os Controladores Digitais de Sinais (DSCs), muitas vezes chamados apenas de DSPs, se relacionam com os Processadores Digitais de Sinais, assim como os microcontroladores estão relacionados com os microprocessadores. Em outras palavras, um DSC une o baixo custo e a variedade de periféricos internos com a arquitetura voltada para o processamento de sinais em tempo real, como os DSPs (OLIVEIRA, 2006, p.34 e p.36).

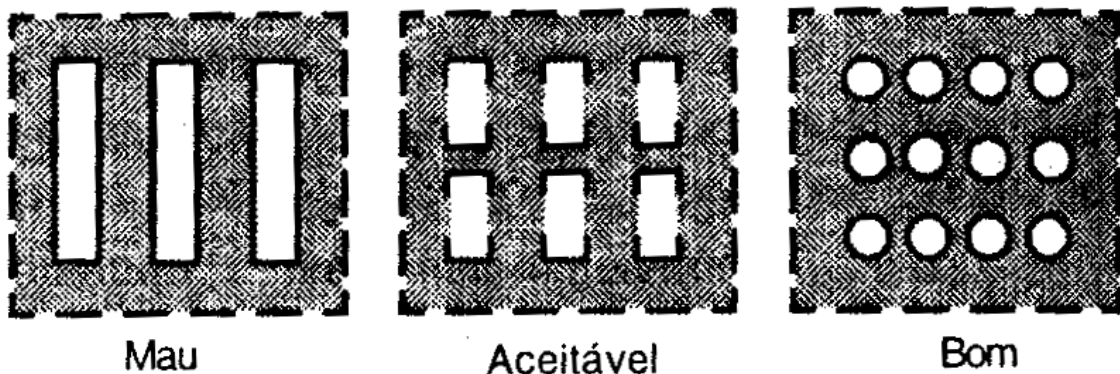
Isto tornou possível a implementação barata e confiável de equipamentos de comunicações (áudio e vídeo) e precisão de medidas em controle industrial.

## 2.2 INTERFERÊNCIA ELETROMAGNÉTICA

Um fator causador de mau funcionamento em muitos aparelhos eletrônicos é a interferência magnética. Entretanto, outros tipos de interferências também prejudicam o funcionamento correto desses aparelhos.

Estas interferências conhecidas (ou principais) são: EFT – Transientes Elétricos Rápidos (picos de pulso positivo ou negativo); ESD – Descarga Eletrostática (geralmente transferida pelos dedos); EMI – Interferência Eletromagnética (conduzida ou radiada) (MICROCHIP, 2013).

É possível proteger uma placa de circuito impresso contra interferências eletromagnéticas, por exemplo, utilizando-se caixas metálicas (de preferência sem grandes descontinuidades, conforme mostra a Figura 3) ao redor da mesma. Isto é a chamada *Gaiola de Faraday* (KOUYOUMDJIAN, 1998, p.118). Pode ser uma proteção individualizada, ou seja, uma por PCB no sistema (facilita o projeto, porém é mais cara), ou uma blindagem integral em todo o sistema (solução barata, mas nem sempre possível).



**Figura 3 - Tipos de furação em chassi.**  
**Fonte: KOUYOUMDJIAN, 1998, p.120.**

No entanto, se as interferências ou ruídos estão sendo gerados em alguma placa ou circuito do próprio sistema, a simples blindagem não resolverá o problema.

Para proteger os circuitos contra ESD, algumas configurações de resistores e capacitores em pinos sensíveis ou que possuam alguma forma de contato com o usuário, mesmo que indireto, é o suficiente para eliminar o problema. Com isso os EFTs são eliminados. Basicamente esta última configuração, ilustrada na Figura 4, são circuitos de filtros RC (Resistor e Capacitor) (SANCHES, 2003, p.92).

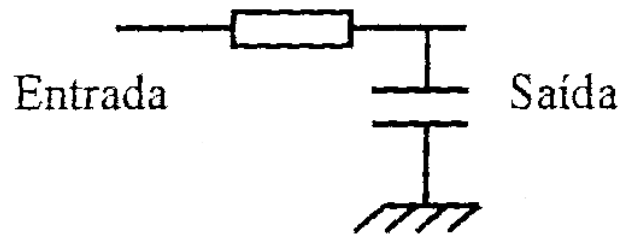


Figura 4 - Filtro RC.  
Fonte: SANCHES, 2003, p.92

Um dos maiores causadores de problemas nas aplicações são os chaveamentos de indutores. Para isso algumas precauções devem ser tomadas. Por exemplo, utilizar um circuito Snubber, destacado na Figura 5, (resistor e capacitor em série) acoplado em paralelo com o circuito chaveador (SANCHES, 2003, p.93), utilizar filtros RLC e ferrites nas entradas e saídas do sistema, utilizar filtros RC em chaveamentos digitais, utilizar desacopladores e “reserva de carga” nas alimentações de circuitos digitais (capacitores cerâmicos e eletrolíticos).

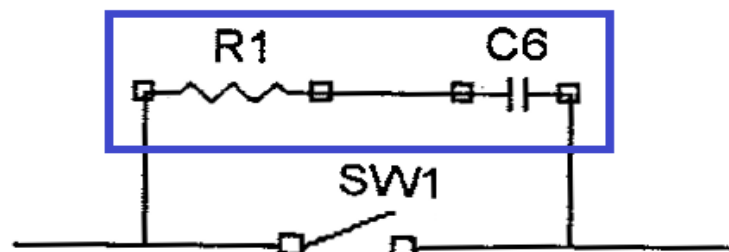


Figura 5 - Circuito Snubber.  
Fonte: SANCHES, 2003, p.6.

Apesar destes métodos de proteção, um bom *layout* de circuito impresso evita a geração de *EMI* que é a interferência com maior peso no surgimento de problemas de funcionamento.

Muitos métodos são aplicados no desenho de um circuito impresso. Basicamente deve-se ter cuidado com a alimentação, separando as alimentações analógicas das digitais. Também é uma boa prática utilizar capacitores cerâmicos próximos aos terminais de alimentações de cada CI para desacoplamento de ruídos (SANCHES, 2003, p.40).

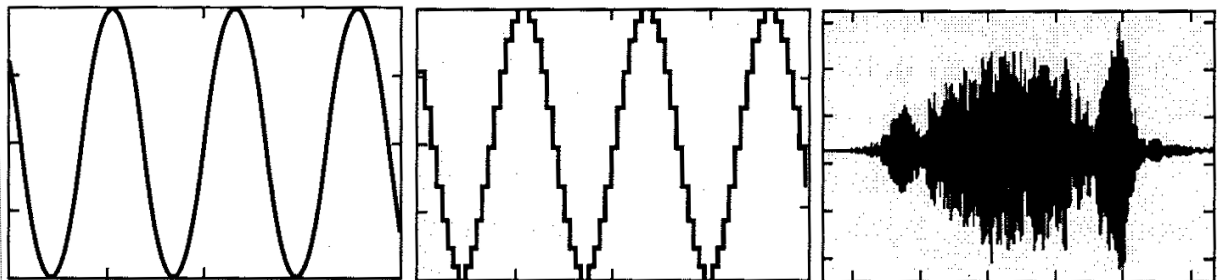
Existem muitos outros aspectos com vários detalhes que devem ser considerados no roteamento das trilhas de uma *PCB*, como os métodos de

aterramento, métodos de alimentação e de roteamento da alimentação de cada bloco funcional do sistema, métodos para traçados de trilhas condutoras de sinais, elaboração do plano ou malha de aterramento no circuito.

Com todas estas precauções, o sistema estará livre de algumas interferências externas e geração e irradiação de ruídos.

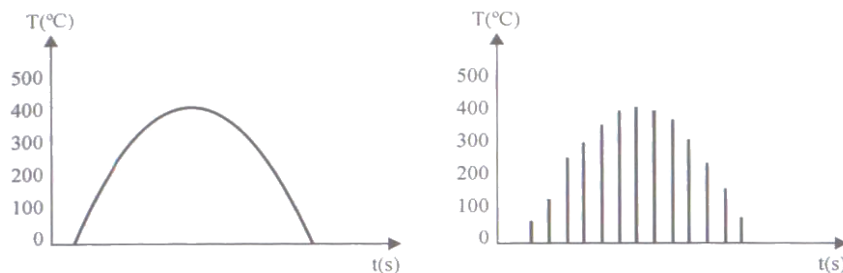
### 2.3 CONVERSORES AD E DA

De um modo geral, os sinais encontrados na natureza são analógicos, pois variam de forma contínua no tempo, como exemplo: os sons e ruídos, intensidade luminosa, etc. (OLIVEIRA, 2006, p.98). Os sinais analógicos, além de contínuos, podem ser periódicos ou não periódicos. Já os sinais digitais, por mais que possam variar no tempo, haverá um pequeno intervalo onde esta variação não será de forma contínua (Figura 6).



**Figura 6 - Analógico periódico, sinal digital, analógico não periódico.**  
 Fonte: OLIVEIRA, 2006, p.93.

Já a Figura 7 mostra um sinal analógico e seu equivalente amostrado.

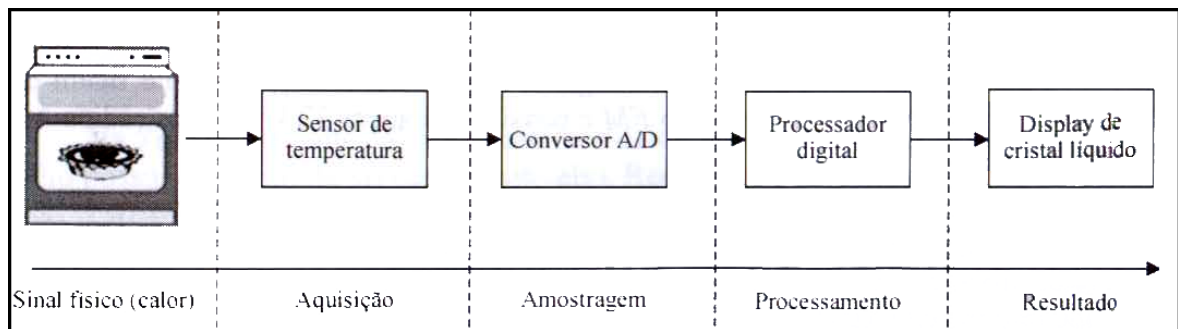


**Figura 7 – Representação do sinal analógico e o digital equivalente.**  
 Fonte: OLIVEIRA, 2006, p.99.

Os sinais que os sistemas modernos, como computadores, trabalham são os digitais. Para isso usa-se a conversão analógico-digital (A/D) para representar os dados analógicos nos sistemas digitais. Isso torna viável o uso dessa representação do sinal original para cálculos, operações ou simplesmente armazenar esses valores (OLIVEIRA, 2006, p.98). O processo inverso chama-se conversão digital-analógico (D/A), e consiste na conversão do sinal oriundo do sistema digital para um sinal analógico capaz de reproduzir o sinal físico inicial, mas com alguma espécie de processamento (eliminação de ruídos, correção de falhas).

Estes circuitos conversores podem tanto vir internamente nos processadores e controladores de sinais digitais como em circuitos integrados dedicados para estas funções. Resumidamente, os conversores são blocos com portas de entrada e saída. A entrada num conversor A/D recebe sinais elétricos de forma contínua e operam numa faixa de tensão de entrada máxima. Um exemplo são os conversores A/D de um microcontrolador que funciona alimentado com uma tensão de 5V. Neste caso, este A/D opera na faixa de [-5V a +5V] (OLIVEIRA, 2006, p.98).

A Figura 8, a seguir, ilustra um sistema que utiliza um bloco de conversão A/D para mostrar a temperatura em um display.



**Figura 8 - Diagrama de uma conversão A/D com seus principais processos.**  
**Fonte: OLIVEIRA, 2006, p.98.**

Alguns termos devem ser analisados para se trabalhar com circuitos conversores A/D e D/A. Estes são: a *amostragem* e a *quantização*.

Na saída de um conversor A/D, por exemplo, é disponibilizado um determinado valor que representa o sinal analógico naquele momento (quantização) por certo intervalo de tempo fixo (determinado pela frequência de amostragem). A informação digital difere da informação original contínua devido a estes dois termos, ou seja, é *amostrada* porque se baseia em amostragens (leituras num intervalo de

tempo do sinal contínuo) e é *quantizada* porque é atribuído um valor proporcional a cada amostra (OLIVEIRA, 2006, p.99).

As três etapas mais importante desse processo são: aquisição, amostragem e processamento. A aquisição consiste na captura do sinal analógico através de algum sensor. A amostragem foi abordada anteriormente e o processamento pode ser entendido como o programa que executará as várias operações com o sinal já representado digitalmente.

A taxa de conversão (também chamada de frequência de amostragem) é o número de amostras obtidas por segundo. Esta frequência é dada em *Hertz* (Hz) e é adequada quando se pode reconstituir um sinal analógico razoável a partir das amostras resultantes da conversão (OLIVEIRA, 2006, p.100).

Esta taxa é muito importante no processamento de sinais reais. Para tanto, deve adequar-se ao teorema da amostragem, conhecido também como teorema de *Nyquist* ou *Shannon* (OLIVEIRA, 2006, p.100).

Este teorema basicamente diz que um sinal contínuo corretamente amostrado não pode conter frequências acima de um valor máximo em seu espectro de frequências (banda limitada). Outro aspecto importante é que a taxa de amostragem deve ser pelo menos duas vezes maior que a frequência máxima (limite de banda), por exemplo, para representar um sinal de áudio de no máximo 5 kHz, a taxa de amostragem do conversor deve ser no mínimo de 10 kHz (OLIVEIRA, 2006, p.100).

## 2.4 CONTROLADOR PID

No atual mercado competitivo, as empresas são obrigadas a melhorar continuamente a produtividade das suas plantas industriais. Uma das áreas tecnológicas fundamentais para se aumentar a rentabilidade das unidades é a de controle, automação e otimização de processos (CAMPOS, 2006, p.3).

O controle de um processo pode ser de malha aberta ou malha fechada. Os sistemas de controle com realimentação são, com frequência, denominados também sistemas de controle de malha fechada. O sinal de erro atuante, que é a diferença entre o sinal de entrada e o sinal de realimentação, realimenta o controlador, de modo que minimize o erro e corrige a saída do sistema ao valor desejado. Os

sistemas de controle de malha aberta são aqueles em que o sinal de saída não exerce nenhuma ação de controle no sistema. Assim, a cada entrada de referência corresponde uma condição fixa de operação (OGATA, 1998, p.5).

O controlador do tipo Proporcional Integral Derivativo (*PID*) é o mais usado em sistemas de malha fechada na área industrial (CAMPOS, 2006, p.7).

#### 2.4.1 Ações de controle

O algoritmo *PID* usa o erro em três módulos distintos para produzir a sua saída ou variável manipulada: o termo proporcional (P), o integral (I) e o derivativo (D) (CAMPOS, 2006, p.23).

**P (Proporcional)** – Esta ação de controle gera a sua saída proporcionalmente ao erro ( $e(t)$ ). O fator multiplicativo ( $K_p$ ) é conhecido como ganho do controlador (CAMPOS, 2006, 23)

A saída do controlador  $u(t)$ , com ganho proporcional é dado pela equação (1).

$$u(t) = K_p \times e(t) + P_0 \quad (1)$$

Onde:

$u(t)$ : Sinal de saída do controlador

$K_p$ : Ganho proporcional

$e(t)$ : Sinal do erro  $e(t) = SP - PV$  ou  $e(t) = SP + PV$

$P_0$ : Polarização inicial

Caso o ganho proporcional seja muito alto o sistema pode se tornar instável. O sistema pode se tornar muito sensível ou pouco sensível dependendo do valor do ganho, ou seja, uma saída muito alta dada por um erro muito pequeno ou vice-versa.

Alguns fabricantes de controladores industriais usam a banda proporcional (*BP*), em vez do ganho, que é definida como sendo 100% divididos pelo ganho ( $BP=100/K_p$ ). O erro do controlador também pode ser definido como o “*set-point*”

menos a variável controlada ( $SP - PV$ ) ou vice-versa como:  $PV - SP$  (CAMPOS, 2006, p.24)

A equação (2) mostra a definição do erro para ação direta e a equação (3) para ação reversa:

$$e(t) = PV - SP \quad (2)$$

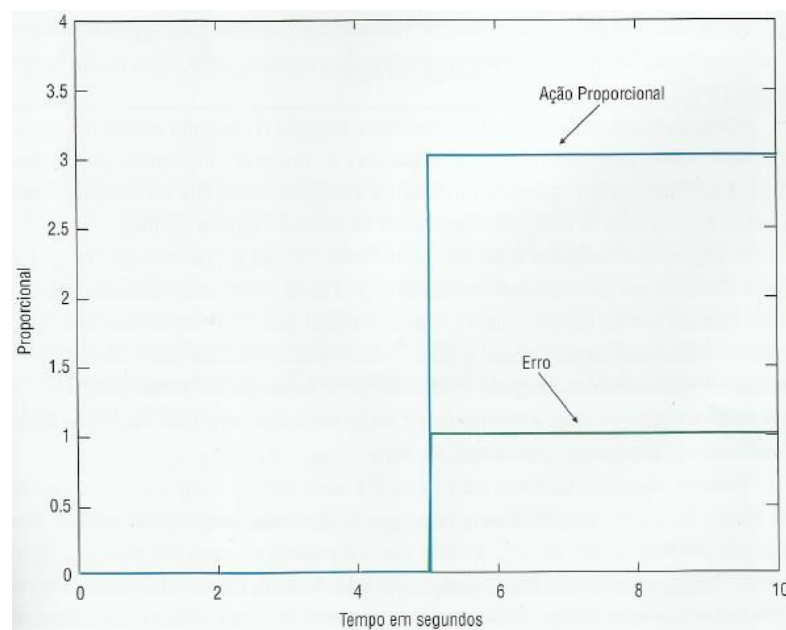
$$e(t) = SP - PV \quad (3)$$

Onde:

$SP = Set-point$

$PV = Variável controlada$

A Figura 9 mostra a ação do controlador proporcional quando ocorre um erro em degrau.



**Figura 9 - Resposta do Controle Proporcional a partir de um erro.**

Fonte: CAMPOS, 2006, p.25.

**PI (Proporcional Integral)** – Nesta ação a saída do controlador é ajustada somando-se o ganho proporcional ( $K_p$ ) e o ganho integral ( $K_i$ ) multiplicado pelo erro acumulado num certo período de tempo. Portanto a contribuição da integração é proporcional à magnitude e a duração do erro.



A saída do controlador  $u(t)$ , com ações de controle PI são dados pela equação (4)

$$u(t) = K_p \times e(t) + K_p \times 1/T_i \times \int e(t)dt + P_0 \quad (4)$$

Onde:

$u(t)$ : Sinal de saída do controlador

$T_i$ : Tempo integrativo

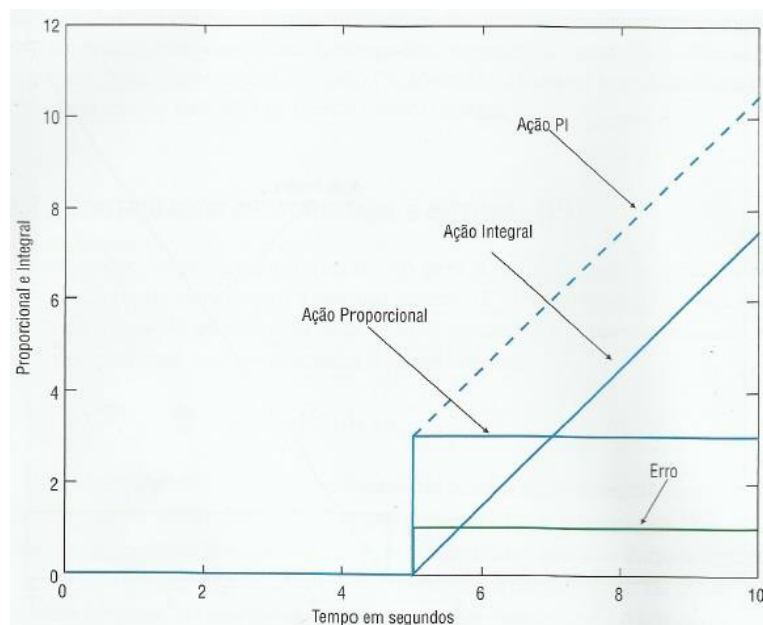
$e(t)$ : Sinal do erro  $e(t) = SP - PV$  ou  $e(t) = SP + PV$

$P_0$ : Polarização inicial

$K_p$ : Ganho proporcional

O termo ( $T_i$ ) é o tempo integral. Alguns fabricantes preferem que o termo da ação integral a ser ajustado durante a sintonia seja o tempo integral ( $T_i$ ) em segundos ou minutos por repetição, enquanto outros escolhem o “reset” ou repetição por segundo ou por minuto, que é o inverso do tempo integral ( $1/T_i$ ). A ação integral continua mudando a saída enquanto existir erro. O controlador PI não irá aceitar um erro em regime permanente entre o valor desejado (*set-point*) e a variável a ser controlada (PV) (CAMPOS, 2006, p.26)

A Figura 10 mostra a ação do controlador proporcional integral quando ocorre um erro em degrau.



**Figura 10 - Resposta do controle Proporcional Integral a partir de um erro.**  
 Fonte: CAMPOS, 2006, p.28.

**PD (Proporcional Derivativo)** – Multiplicando a declividade do erro ao longo do tempo (derivada primeira) pelo ganho derivativo ( $K_d$ ) se obtém a taxa de variação do erro de processo.

O ganho derivativo é dado por:

$$u(t) = K_p \times e(t) + K_p \times T_d \times de/dt(t) + P_0 \quad (5)$$

Onde:

$u(t)$ : Sinal de saída do controlador

$e(t)$ : Sinal do erro  $e(t) = SP - PV$  ou  $e(t) = SP + PV$

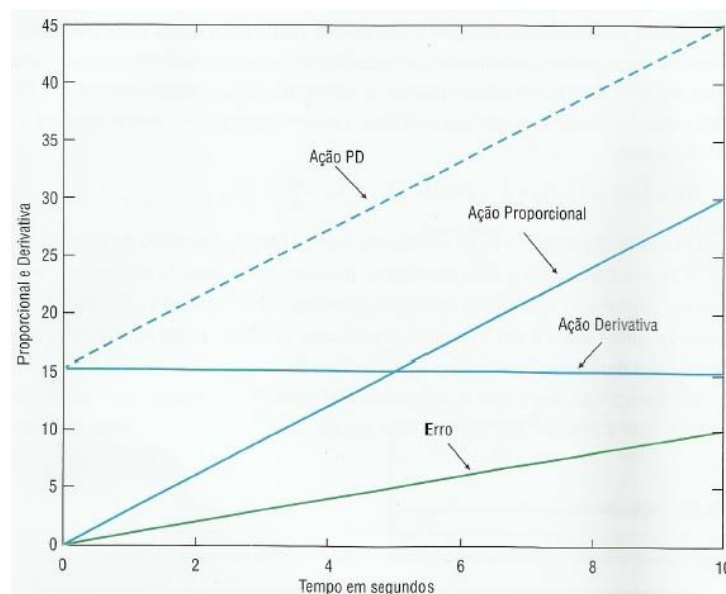
$P_0$ : Polarização inicial

$K_p$ : Ganho proporcional

$T_d$ : Tempo derivativo

O termo derivativo é útil para diminuir os efeitos do termo integrativo, como o *Overshoot*, e melhora a estabilidade do processo. No entanto este termo é muito sensível ao ruído, portanto, dependendo da magnitude do ganho derivativo o sistema pode tornar-se instável. (OGATA, 1998, p.181).

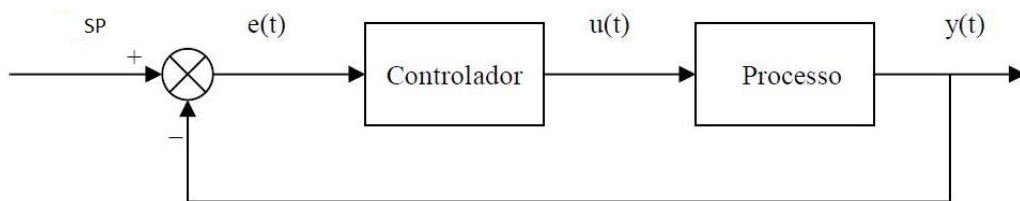
A Figura 11 mostra a ação do controlador proporcional derivativo quando ocorre um erro em rampa.



**Figura 11 - Resposta do controle Proporcional Derivativo a partir de um erro.**  
**Fonte: CAMPOS, 2006, p.30.**

### 2.4.2 Controle com realimentação

O diagrama em blocos, mostrado na Figura 12, permite avaliar o comportamento do sistema com controle em malha fechada.

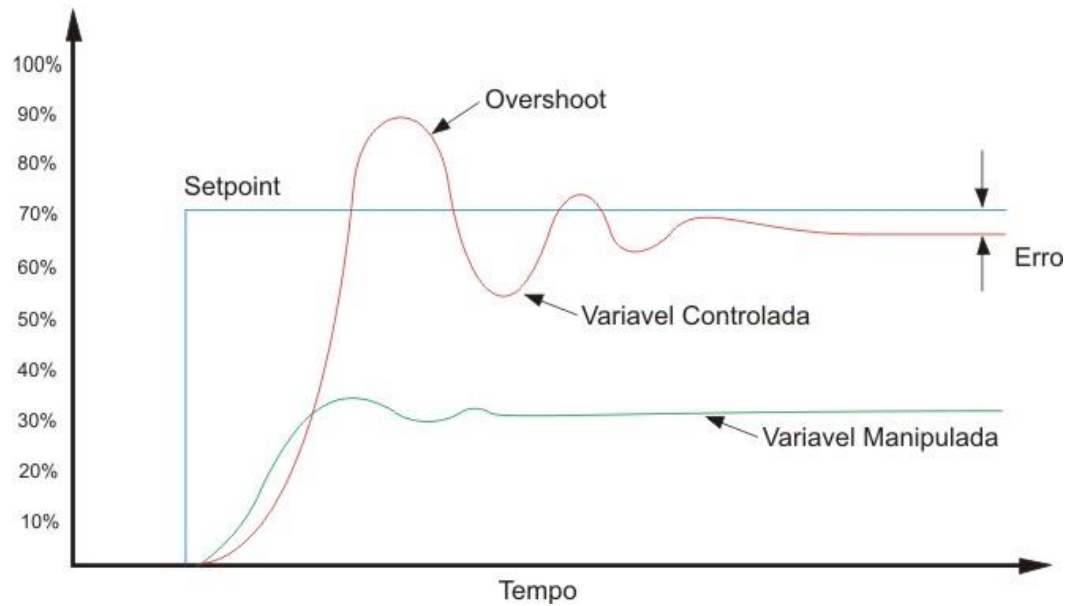


**Figura 12 – Controle em Malha Fechada.**  
Fonte: Autoria própria.

A fim de tornar o sistema mais preciso o sinal de saída  $y(t)$  é comparado com um sinal de referência  $SP$  e o desvio entre estes dois sinais é utilizado para determinar o sinal de controle  $u(t)$  que deve efetivamente ser aplicado ao processo. Assim, o sinal de controle é determinado de forma a corrigir este desvio entre a saída e o sinal de referência. Portanto, a diferença existente entre o sinal de referência  $SP$  e o sinal de saída  $y(t)$  multiplicada pela ação de controle, gera em sua saída um sinal de controle  $u(t)$  que é capaz de corrigir e se possível anular tal diferença. No caso específico do controlador PID, a lei de controle descrita pelo bloco do controlador é composta de três termos.

Ajustando os parâmetros de controle, consegue-se sintonizar um *loop* de controle, porém sistemas diferentes têm comportamentos diferentes e muitos processos têm um grau de não-linearidade. Assim parâmetros que funcionam bem em determinada faixa não funcionam tão bem em outras faixas (OGATA, 2003, p.545).

O gráfico da Figura 13 mostra algumas informações significativas para a análise de controle.



**Figura 13 - Principais sinais no controle de processos.**  
**Fonte: Autoria Própria.**

Na Figura 13 tem-se o *set-point*, que representa o valor desejado para a variável controlada. A variável controlada muda seu valor através da ação de outra variável denominada variável manipulada. Dependendo do tempo de resposta o sistema pode ultrapassar o *set-point*. Este fenômeno é conhecido como *overshoot*. Quando no estado estacionário o valor da variável controlada não corresponde ao valor do *set-point* o sistema apresenta um erro, denominado erro de estado estacionário.

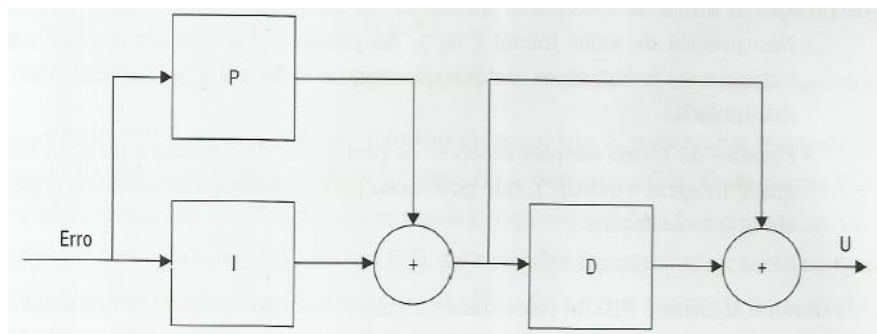
#### 2.4.3 Arranjo estrutural

Os controladores PID podem ter sua estrutura classificada como série, paralelo ou misto. Muitas vezes um equipamento pode oferecer mais de uma estrutura interna de controle e a escolha da estrutura será definida de acordo com a necessidade. As equações dos controladores são diferentes, afetando no desempenho e dinâmica do processo (IWASSE, 2009, p.33).

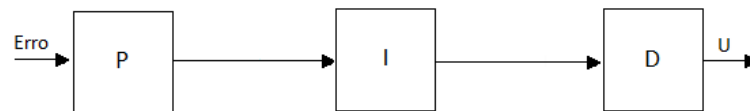
Conforme o arranjo, os ganhos podem estar relacionados e serem dependentes uns dos outros ou serem independentes. Um exemplo é o arranjo em série ou interativo (Figura 14), neste o termo PI é calculado em paralelo e em seguida o resultado deste termo passa pelo termo derivativo para calcular a saída do

PID (CAMPOS, 2006, p.32). A Figura 15 ilustra um arranjo onde os três termos são calculados em série.

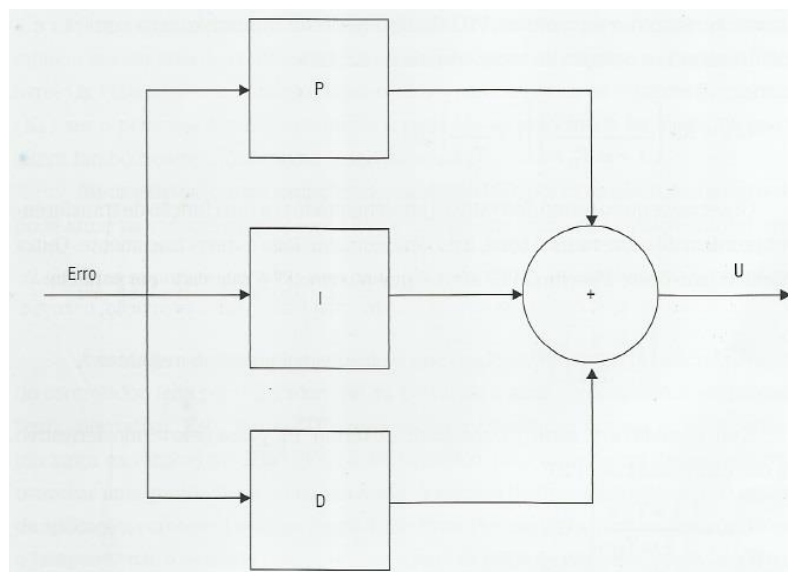
Um exemplo de arranjo onde os ganhos são independentes é o paralelo alternativo (Figura 16). Ele é dito paralelo porque as suas ações: proporcional (P), integral (I) e derivativo (D) são calculadas em paralelo e em seguida somadas (CAMPOS, 2006, p.31).



**Figura 14 – Algoritmo PID série ou iterativo.**  
Fonte: CAMPOS, 2006, p.33.



**Figura 15 - Algoritmo PID série.**  
Fonte: Autoria Própria.



**Figura 16 – Algoritmo PID paralelo alternativo.**  
Fonte: CAMPOS, 2006, p.31.

### 3 MÉTODO DE PESQUISA

Partindo do princípio de que o controlador PID é microcontrolado, o trabalho foi iniciado através de uma pesquisa que verificou qual o DSC mais indicado. Para isso foi analisado os *datasheets* dos DSCs de cada fabricante como a *Microchip*, *Texas Instruments*, entre outros. Nesta etapa foi imprescindível analisar se o DSC possui parâmetros PID interno, se estes estão dispostos em série ou paralelo, se possui conversores AD e DA internamente. Após serem identificados os recursos oferecidos pelo DSC, foram implementadas as rotinas de programação baseadas na teoria de controle.

Em seguida foi desenvolvido o *hardware*. Nesta fase do projeto é importante ser levantado todos os componentes necessários e quais pinos do DSC estarão comprometidos, bem como projetar um *layout* imune a ruídos. Inicialmente o controlador teria recursos equivalentes ao controlador N1100 da NOVUS (NOVUS, 2013), por esta razão foi analisado a ficha técnica deste modelo. O *software* usado para desenvolver o *layout* da placa de circuito impresso foi o ALTIUM DESIGNER 6 (ALTIUM, 2013), pois oferece muitas ferramentas para edição e é de fácil manuseio.

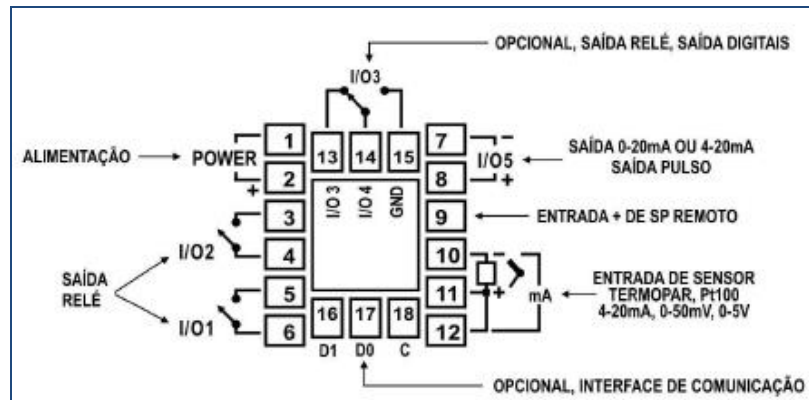
O *hardware* determinou o tamanho necessário da caixa que envolve o equipamento. Os pontos de fixação foram determinados através do *layout* da placa,. Após isto, o projeto da caixa teve suas partes funcionais praticamente definidas: torres de fixação da placa, visor para o *display*, posição dos botões de configuração, posição dos conectores e *plugs*. Para o desenvolvimento da caixa foi utilizado o *software* Solidworks (SOLIDWORKS, 2013) de fácil modelagem, pois oferece recursos para se identificar interferências e facilita a criação do detalhamento técnico dos desenhos e levantamento da lista de materiais.

#### 3.1 ANÁLISE DE MERCADO

Como referência, foi utilizado o controlador N1100 do fabricante NOVUS. Foram analisadas as principais funções, essenciais para um controlador PID. Recursos como: entrada e saída de corrente (4-20mA), sensor de temperatura (PT100), PWM, entradas e saídas digitais e sinalizadores de alarme, foram consideradas e implementadas. Contudo, recursos como comunicação serial

(RS485), oferecido pelo N1100, não foi implementada, mas estão disponíveis no hardware pinos para esta comunicação, podendo ser desenvolvido futuramente um protocolo de comunicação.

Na Figura 17 podemos verificar as conexões elétricas do controlador N1100 e os I/Os disponíveis



**Figura 17 - Conexões elétricas do controlador N1100 da Novus.**  
**Fonte: NOVUS, 2013, p.4.**

## 4 DETALHAMENTO TÉCNICO

### 4.1 VISÃO GERAL DO PROJETO

Este projeto utiliza conceitos básicos de controle, sistemas analógicos, digitais, microcontrolador, programação e processos mecânicos.

Para realização das funções propostas, o usuário determina os parâmetros do processo (dados de entrada, saída e ganhos), conecta os periféricos no controlador (sensores, atuadores e transmissores) e inicia o sistema para monitoramento e controle do processo.

O diagrama na Figura 18 representa o menu e as funcionalidades do controlador.

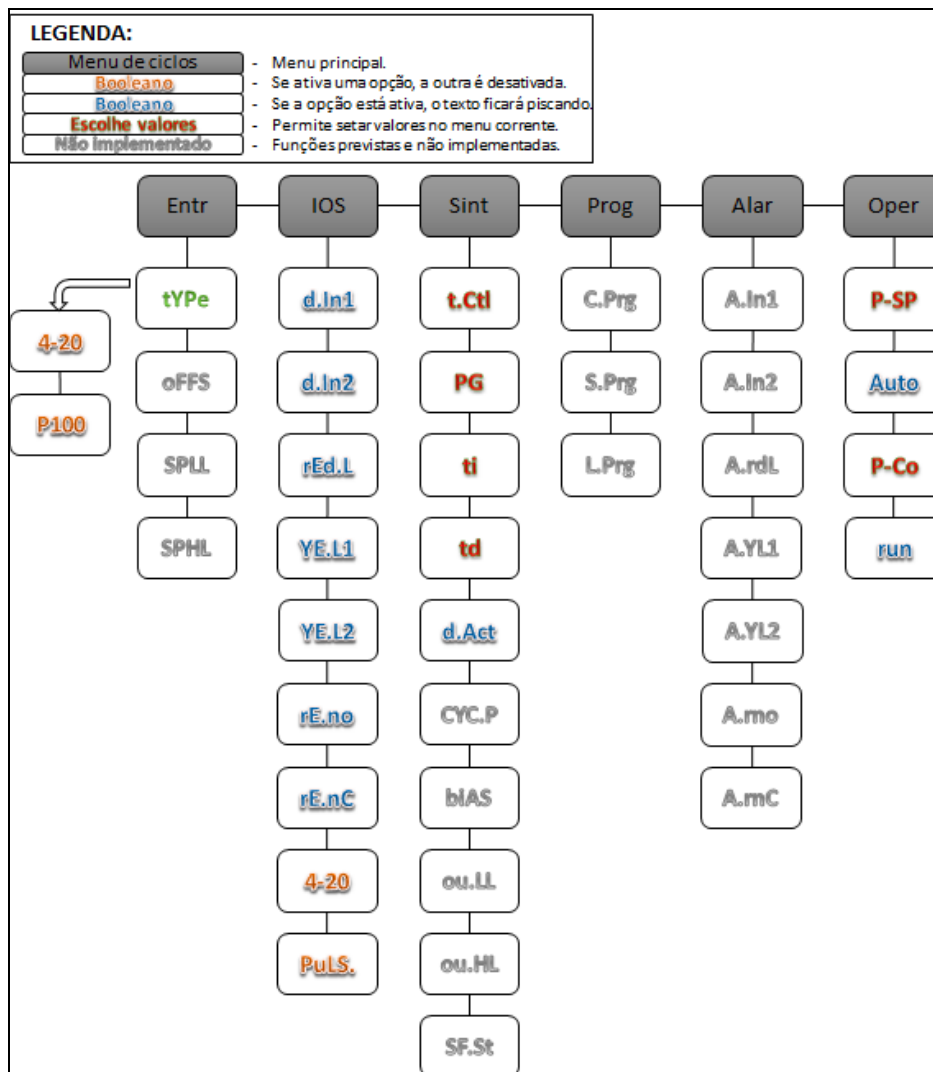


Figura 18 – Diagrama do Menu  
Fonte: Autoria Própria.

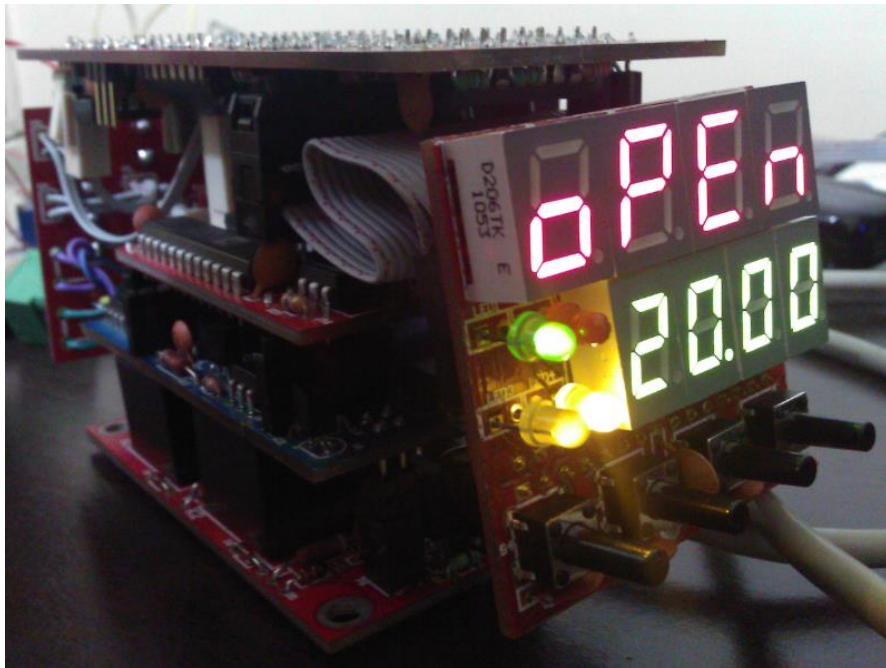


#### 4.1.1 Interface com o usuário

Ao usuário são disponibilizados dois visores, compostos por oito *displays* de sete segmentos, quatro vermelhos e quatro verdes, e quatro botões que possibilitam a interação com as rotinas programadas no controlador e verificação dos dados de controle.

Para as conexões físicas do controlador com a planta, ou sistema, são utilizados Terminais da Metaltex modelos BR8.

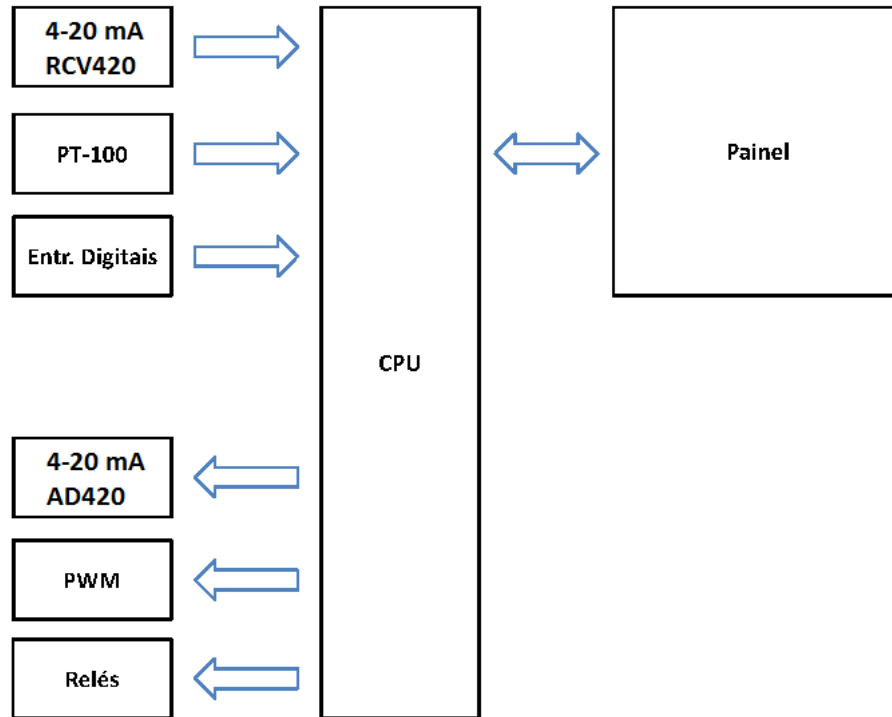
Através da Figura 19 é possível ter uma visão geral do controlador.



**Figura 19 - Visão geral do controlador.  
Fonte: Autoria Própria.**

#### 4.1.2 Diagrama em blocos do sistema

A Figura 20 apresenta um diagrama em blocos do sistema proposto, que passa a ser descrito a seguir:



**Figura 20 - Diagrama em blocos do sistema.**  
**Fonte: Autoria Própria.**

#### 4.1.3 Descrição dos blocos

- MICROCONTROLADOR (CPU): Tem a função de controlar o processo, tratando os dados obtidos pelas entradas, de acordo com os parâmetros determinados pelo usuário, e enviado para as saídas os dados de controle.
- PT100: Sensor de temperatura responsável pela aquisição desta grandeza.
- RCV420: Conversor digital de corrente para tensão, responsável pela aquisição de dados oriundos de receptores que podem ser conectados no controlador.
- PWM: Sinal de saída para controle de potência apenas de cargas CC.
- AD420: Conversor digital de dados (16 bits) para corrente numa faixa de 4 a 20 mA. Responsável por repassar o sinal para transmissores de corrente.

- RELÉS: Saída digital de uso geral (Alarmes sonoros e/ou luminosos, controle on/off, etc...)
- ENTRADAS DIGITAIS: Entradas de dados de propósitos gerais. Como sinalização de alarme ou botão de emergência.
- PAINEL: Interface entre homem/máquina que consiste em botões, *display* e *LEDs* indicadores.

## 4.2 DESCRIÇÃO FUNCIONAL DOS BLOCOS

### 4.2.1 Diagrama esquemático

Os diagramas esquemáticos estão no **APÊNDICE A**.

### 4.2.2 DSPic (CPU)

O microcontrolador modelo DSPic 30F2010 possui 28 pinos, dos quais 20 são de entradas/saídas. Desses vinte pinos, quatro tem funções compartilhadas: os pinos 17 e 18 são as entradas para gravação do *firmware* e também funcionam como entradas digitais (MICROCHIP, 2008, p.9,10), e os pinos 5 e 6 são utilizados tanto para comunicação com o AD420 (CI saída de corrente) quanto para o MCP3553 (CI conversor A/D utilizado tanto para entrada de corrente quanto PT100).

O DSPic é responsável por gerenciar o sistema, tratando as informações adquiridas pelos pinos de entrada através de rotinas programadas em sua memória de programa interna, em seguida envia dados para os periféricos conectados nos pinos de saída, os quais atuam sobre o sistema. Todos os sinais de entradas e saídas da CPU são eletricamente isolados dos periféricos através de optoacopladores, e a fonte de alimentação da CPU é independente da alimentação dos periféricos evitando interferências elétricas, conforme Figura 21.

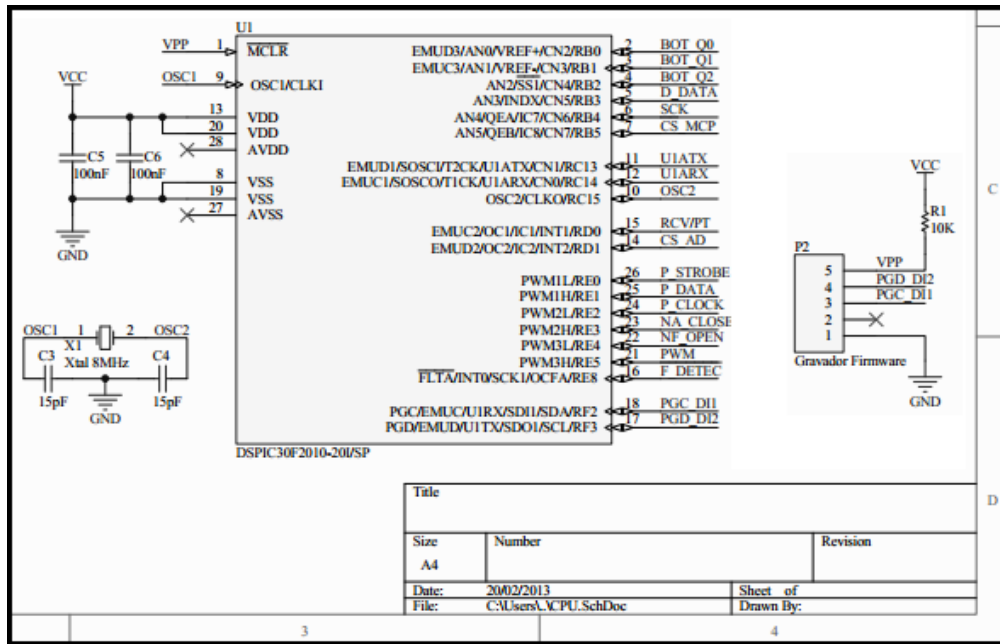


Figura 21 - Circuito do microcontrolador (CPU).  
Fonte: Autoria Própria.

#### 4.2.3 Interface

Este circuito tem a função de realizar a gravação do *firmware* e a comunicação entre os periféricos (relés, sensor PT100, driver de PWM) e a CPU. Por exemplo, a leitura de um sinal do sensor PT100 e o envio de sinais para o PWM, conforme Figura 22.

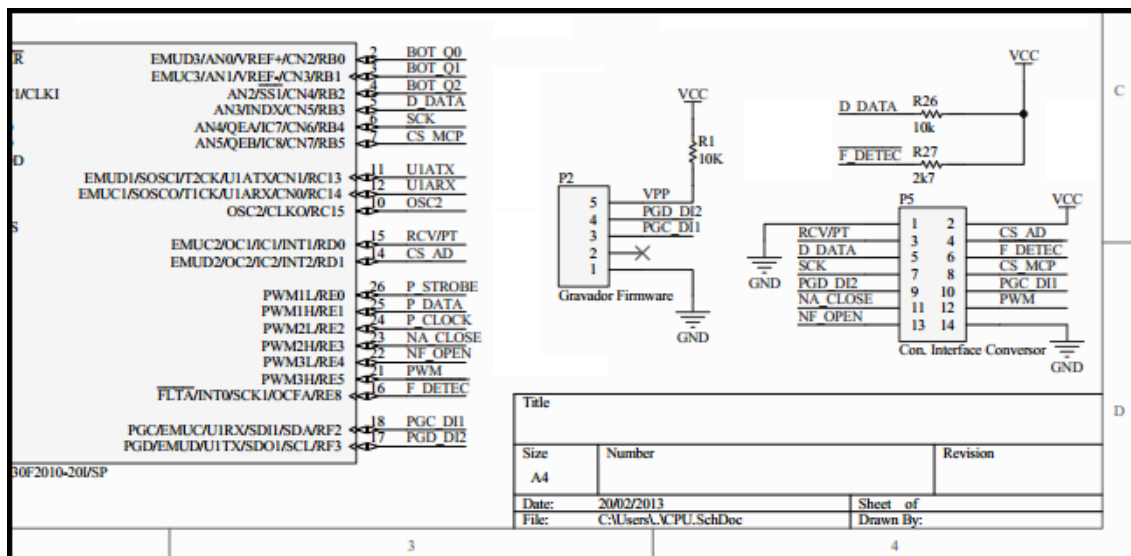


Figura 22 - Circuito da interface (I/O digitais e de corrente; PT100; PWM).  
Fonte: Autoria Própria.

#### 4.2.4 PT100 e RCV420

O circuito da Figura 23 é compartilhado entre o PT100 e a entrada de corrente através do CI RCV420, pois ambos disponibilizam uma tensão em suas saídas e esta tensão é lida pelo conversor A/D (MCP3553). Para selecionar o sinal a ser lido é utilizado um relé como roteador. O sinal convertido é lido e processado pela CPU.

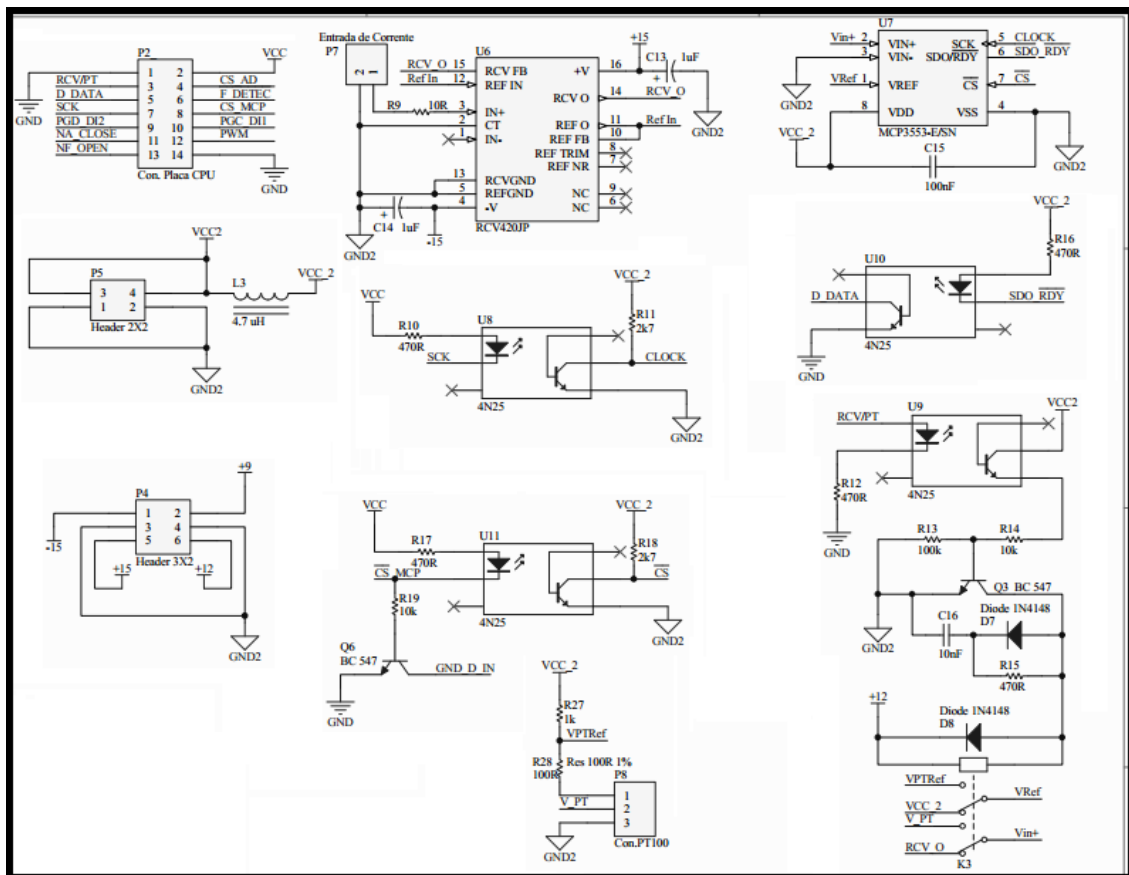
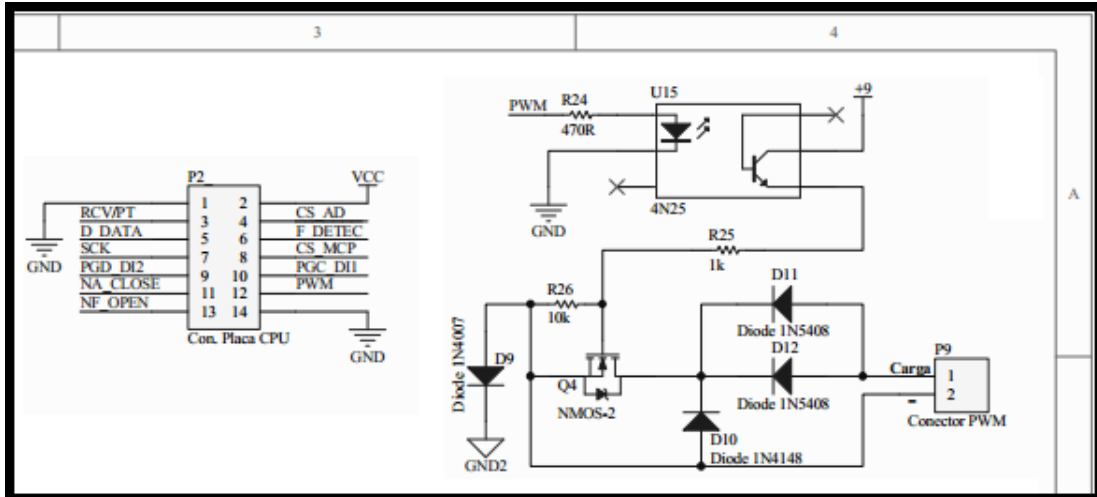


Figura 23 - Circuito do PT100 e RCV420 (Entrada de corrente)  
Fonte: Autoria Própria.

#### 4.2.5 PWM

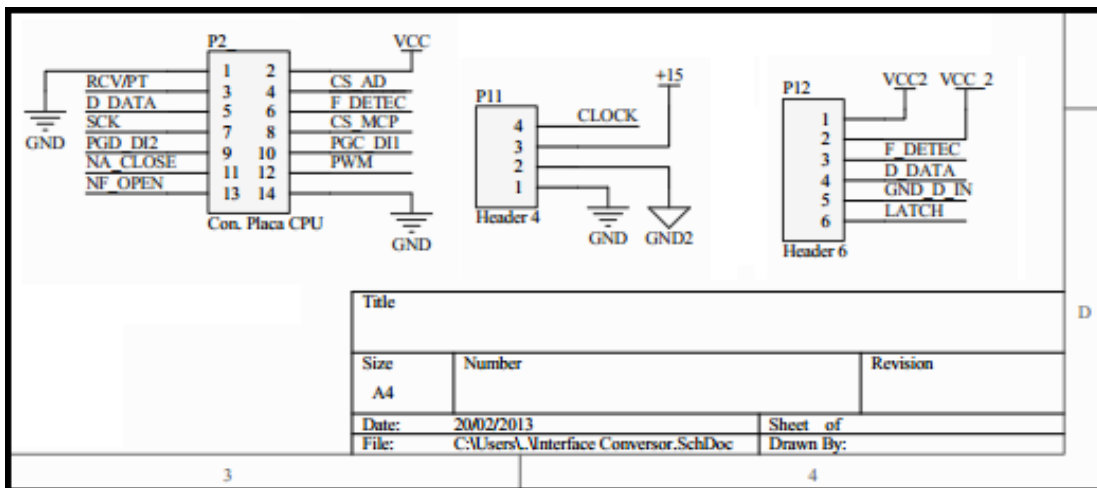
O sinal de PWM é oriundo da CPU. Este sinal poderá controlar uma carga de alta potência através do *driver* mostrado no circuito, conforme Figura 24.



**Figura 24 - Circuito do PWM**  
 Fonte: Autoria Própria.

4.2.6 AD420 (saída de corrente)

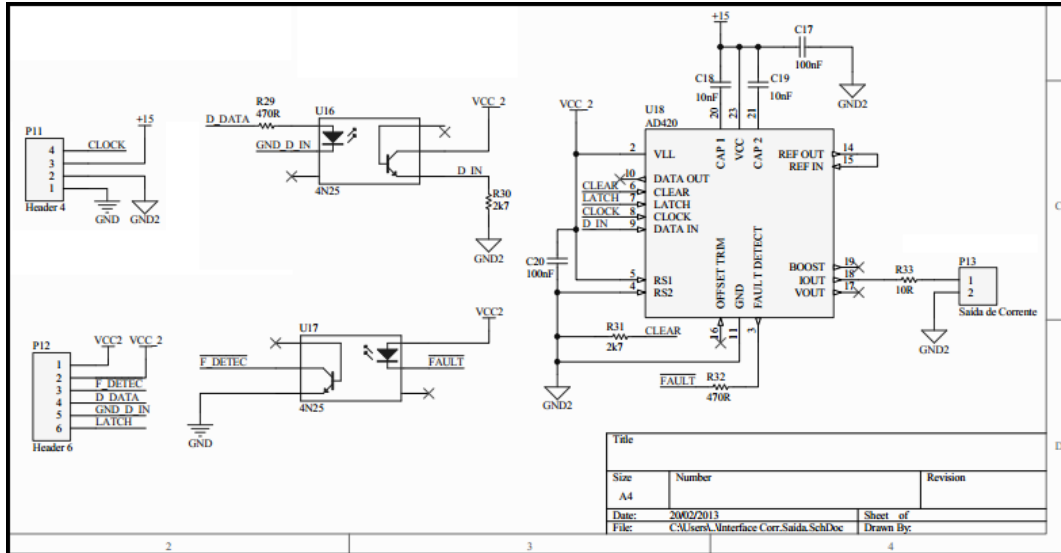
Os conectores representados nesta região do esquemático são apenas interfaces de comunicação entre as placas, conforme Figura 25.



**Figura 25 - Conectores de interface do AD420 (Saída de corrente).**  
 Fonte: Autoria Própria.

O circuito do AD420 da Figura 26 possui um sinalizador de falha que indica quando há uma interrupção de comunicação entre este periférico e o transmissor conectado, ou quando a tensão gerada (corrente de saída x impedância do transmissor) na saída do AD420 for maior que a tensão de alimentação menos 2,5V

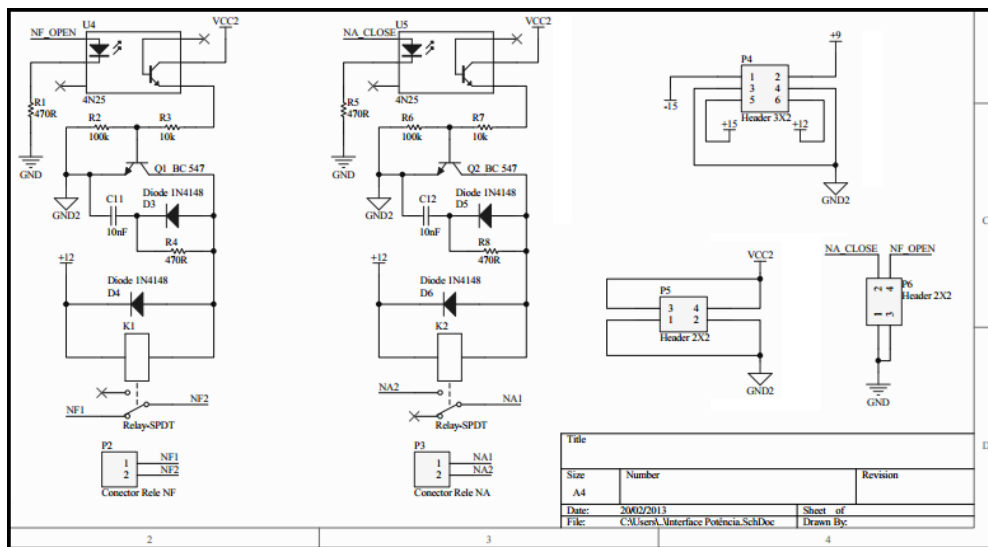
(Vcc – 2,5V). Este CI comunica-se com a CPU serialmente através dos pinos LATCH, CLOCK e DATA IN, pinos 7, 8 e 9 respectivamente.



**Figura 26 - Circuito do AD420 (Saída de corrente).**  
**Fonte: Autoria Própria.**

4.2.7 Relés

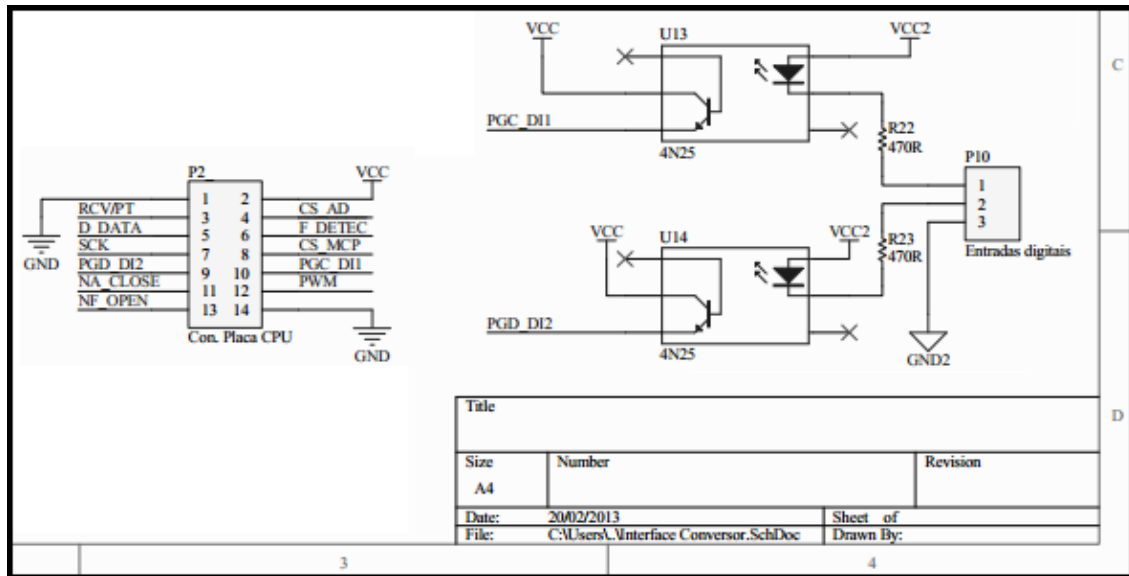
O circuito mostrado na Figura 27 é referente a dois relés acionados por sinais (pulso ou permanente) enviados do CPU. Um dos relés utiliza contato normalmente aberto e outro normalmente fechado. Seus propósitos são para acionamentos de sirenes ou sinalizadores de alarme.



**Figura 27 - Circuito das saídas digitais (Relés).**  
**Fonte: Autoria Própria.**

#### 4.2.8 Entradas digitais

O circuito de entradas digitais da Figura 28 possui duas entradas de sinais, de pulso ou permanente, de 5V. Utilizado para parada de emergência ou pausar e continuar o programa.



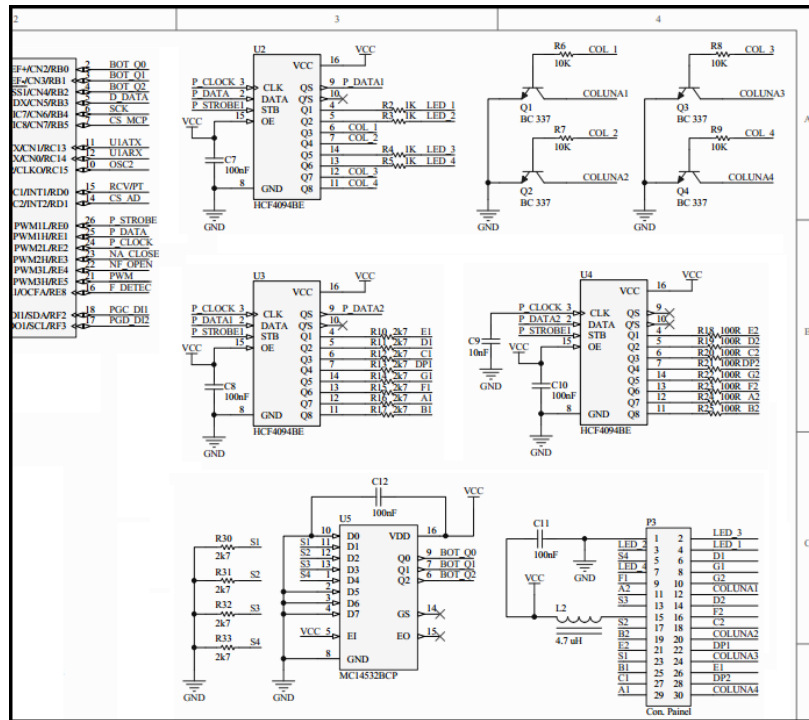
**Figura 28 - Circuito das entradas digitais.**  
**Fonte: Autoria Própria.**

#### 4.2.9 Painel

O circuito do painel presente na placa do CPU (Figura 29), controla a interface homem-máquina do sistema. Este circuito refere-se à leitura do teclado e ao controle serial dos *displays* e *LEDs*. As teclas são lidas pelo CI 4532, um decodificador de prioridade superior, que disponibiliza os dados para o DSPic através dos pinos RB0, RB1 e RB2. Os *displays* e os *LEDs* são controlados por uma informação de 24 bits enviada serialmente pelo pino RE1 do DSPic para os CIs HC4094, um *shift register*, onde é distribuída da seguinte maneira:

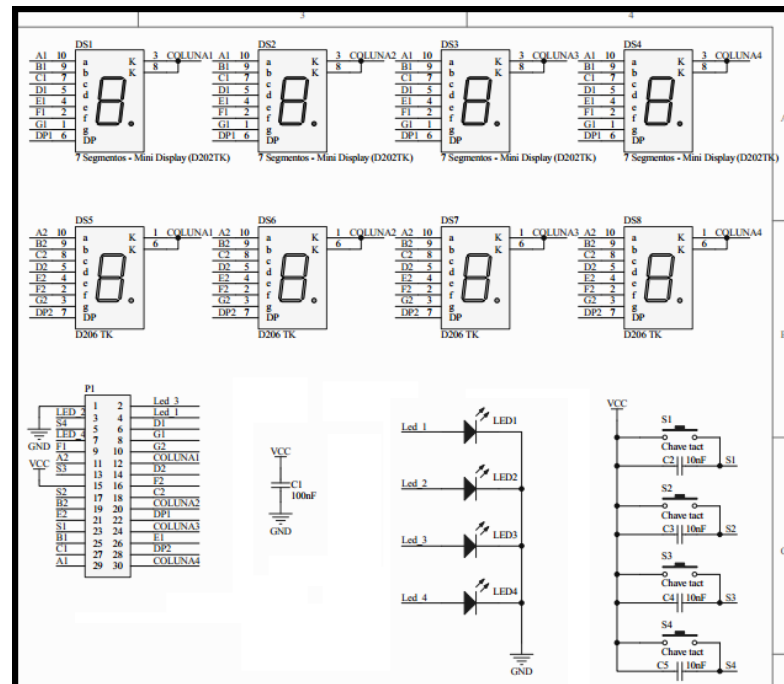
- 8 bits para o *display* vermelho da coluna ativa no momento;
- 8 bits para o *display* verde da mesma coluna ativa no *display* vermelho;
- 4 bits para acionar a coluna que estará ativa em ambos os *displays*;
- 4 bits para acender os *LEDs* indicadores individualmente.





**Figura 29 - Circuito do painel (Placa CPU).**  
**Fonte: Autoria Própria.**

O circuito placa do painel, mostrado na Figura 30, representa apenas os periféricos que interagem com o usuário (*displays*, botões e *LEDs* indicadores).



**Figura 30 - Circuito do painel (Placa do painel).**  
**Fonte: Autoria Própria.**

### 4.3 PROGRAMAÇÃO

A linguagem de programação C foi utilizada para a implementação do *firmware* (*software* para sistema embarcado) através do compilador C30 da Microchip. A IDE (*Integrated Development Environment*) utilizada foi o MPLab também da Microchip.

O funcionamento do programa consiste em disponibilizar na IHM as opções e configurações pertinentes ao funcionamento do controlador. Estas configurações definem o comportamento do controlador em relação aos sinais obtidos na entrada e aos sinais disponibilizados na saída, ou seja, demonstram o controle em si.

Para atender todas as funções oferecidas pelo controlador foi necessário otimizar os recursos disponíveis no dsPIC utilizado. As rotinas foram modeladas adequando a quantidade de memória de programa oferecida e também a quantidade de memória de dados disponível. Além dos recursos de armazenamento, foi necessário compartilhar alguns pinos para poder atender todas as funções do controlador.

O funcionamento da IHM, descrito no item 4.2.9, foi otimizado para funcionar com seis pinos, sendo três para a leitura do teclado e outros três para a escrita das informações no painel.

Outra característica, não visível para o usuário final, é o compartilhamento dos pinos de dados e de *clock* (economizando dois pinos) que tanto lêem sinais do conversor AD (MCP3553) quanto escrevem dados no CI usado para disponibilizar a saída de 4-20mA (AD420). Com isso, apenas mais dois pinos de *chip select* (CS), necessários para cada CI, foram utilizados. Através desses pinos, escolhe-se qual CI está sendo utilizado no momento. Ainda há um segundo compartilhamento no uso do conversor analógico-digital (MCP3553), que é responsável por ler dois sinais, o sinal de 0-5 volts convertido da saída do CI receptor de corrente (RCV420) ou o sinal do divisor de tensão (resistores) que estão em série com a entrada do sinal do PT-100. Esta seleção é feita através do relé K3 mostrado na Figura 20. Devido a estes compartilhamentos de recursos, a escolha da saída de corrente não possibilita a escolha de um período muito baixo de leitura do sinal, pois nesta situação os pinos de dado e *clock* estarão sendo utilizados a todo momento.

Os circuitos do Painel e CPU possuem uma fonte de alimentação de 5 V (através de um conector na parte de trás do chassi) eletricamente isolada da fonte de alimentação dos circuitos de potência, circuitos de leitura e escrita de dados de comunicação com o exterior do aparelho (sensores e atuadores). Este isolamento é feito através de opto-acopladores. Com isso, o acionamento dos relés ou leitura das entradas digitais não interfere nos circuitos da parte lógica, o que poderia gerar ruídos no dsPIC.

Os dois relés de saída (um NA e outro NF), bem como o *driver* do PWM precisam de uma terceira fonte externa para que os elementos ali conectados funcionem corretamente. O *driver* da saída PWM pode gerenciar correntes de até 5 ampere devido ao uso do mosfet de canal N IRF 540N e à construção da placa e das trilhas.

As duas entradas de sinais digitais podem ser utilizadas para acionamento de botões ou de sensores que sinalizem alarmes ou outra situação de emergência. Estas entradas não poderão ser utilizadas se o controlador estiver tendo seu *software* depurado, pois os pinos necessários para a gravação e depuração do programa no dsPIC são compartilhados com estas entradas digitais. O CI responsável pela saída de corrente, AD420, também gera um sinal de alarme quando a resistência do transmissor acoplado é muito baixa, relação entre a tensão de alimentação do CI (15V) e a corrente de saída (máximo de 20mA), ou quando há uma interrupção do fluxo desta corrente de saída.

O *software* embarcado gerencia as opções escolhidas através do menu (tipo de entrada, tipo de saída, parâmetros PID, ação de controle, *offsets*, zona morta do PWM, alarmes utilizados) bem como faz o controle propriamente dito. Para isto foi utilizada a equação PID discreta,  $a(n) = a(n-1) + (K_p + K_D / T + K_I T) e(n) + (-K_p - 2K_D / T) e(n-1) + (K_D / T) e(n-2)$  (RAMOS, 1997, p.202).

A Figura 31 apresenta o código responsável pela troca dos textos de menu nos *displays*. O funcionamento do menu é baseado em uma estrutura com ponteiros para a própria estrutura (menu superior e menus inferiores) e um ponteiro para *strings* (opções do menu atual).

```

#####
/**
/** FUNÇÃO DE ENTRADA DE SELEÇÃO DOS MENUS
/**
#####
void managerMainMenu(unsigned char _keyPressed)
{
    switch (_keyPressed)
    {
        case KEY4:
            if (lastMenu[menuLevel])
            {
                currentMenu = lastMenu[menuLevel--];
            }
            break;

        case KEY3:
            (currentMenu->currentOption > 0) ? currentMenu->currentOption--
            : (currentMenu->currentOption = currentMenu->limitOption);
            break;

        case KEY2:
            (currentMenu->currentOption < currentMenu->limitOption) ? currentMenu->currentOption++
            : (currentMenu->currentOption = 0);
            break;

        case KEY1:
            if (currentMenu->nextMenu[currentMenu->currentOption])
            {
                lastMenu[++menuLevel] = currentMenu;
                currentMenu = currentMenu->nextMenu[currentMenu->currentOption];
            }
            break;
    }

    valueToDisp (DISP_VAR_CONTR, (char*)currentMenu->currentMenu[currentMenu->currentOption]);
    valueToDisp (DISP_SETPOINTI, (char*)currentMenu->lastMenu);
}

```

**Figura 31 – Código responsável pelo menu.**  
**Fonte: Autoria própria.**

Se as teclas 2 e 3, no painel, são pressionadas o texto no visor muda para a *string* posterior ou anterior, respectivamente, fazendo com que as várias opções do menu atual sejam mostradas como se fossem colunas sendo trocadas no visor. Na Figura 32 temos o trecho que declara o ponteiro de *string* para o menu inicial relacionado com a troca das opções dos ciclos do controlador.

```

const char * stringCicl[] = {"Entr", " IOS", "Sint", "Prog", "Alar", "Oper"};

```

**Figura 32 - Trecho onde um ponteiro é declarado.**  
**Fonte: Autoria própria.**

Cada menu (estrutura de dados em C) é iniciado com o endereço de ponteiro para o menu ao qual pertence (para poder voltar à tela inicial de opções). Com

ponteiros para os menus das opções inferiores (entrar em uma dada opção, como escolher qual será a saída do controlador ou qual serão os parâmetros PID por exemplo) e ponteiro para as *strings* que representam as opções no visor.

Se as teclas 1 ou 4 são pressionadas, a variável que guarda o menu corrente é atualizada para guardar o endereço do menu superior ou inferior, respectivamente, fazendo com que as opções exibidas no visor sejam o menu “pai” ou “filho”.

Para um melhor entendimento do funcionamento do menu é possível analisar o fluxograma genérico do menu no **APÊNDICE B**.

O controle PID discreto é inicializado com seus respectivos parâmetros ao selecionar a função “Run” no menu, fazendo com que o controle propriamente dito se inicie. A Figura 33 apresenta a imagem com a rotina de inicialização do controle.

```
#include <p30f2010.h>
#include "adcMCP3553.h"
#include "main.h"
#include "PIDControl.h"
#include "transmitterCurrent.h"

float T, K, Ti, Td, Ki, Kp, Kd;
float erro = 0.0, erro1 = 0.0, erro2 = 0.0;
float uk = 0.0, uk1 = 0.0, value;
unsigned char isDirectControl;

#####
/**
**# FUNÇÃO DE INICIALIZAÇÃO DOS PARÂMETROS DO CONTROLE PID.
**#
**#
**#
#####
void initParamPID(unsigned int _timerPeriod, float _K, float _Ti, float _Td, unsigned char isDirectControl)
{
    T = _timerPeriod * 0.001; // Transforma de mS para S.
    K = _K;
    Ti = _Ti;
    Td = _Td;
    isDirectControl = _isDirectControl;

    Ki = (K * T) / Ti;
    Kp = K - (Ki / 2);
    Kd = (K * Td) / T;
}
```

**Figura 33 - Rotina de inicialização do controle.**  
**Fonte: Autoria própria.**

Enquanto o controle PID estiver rodando, a rotina mostrada na Figura 34 será executada a cada período “T” setado na inicialização do controle. Ainda na Figura 34, a equação PID discretizada completa (considerando os três parâmetros de controle) é destacada.

```

//#####
//#                                     #
//# FUNÇÃO DE CÁLCULO DO CONTROLE PID.   #
//#                                     #
//#####
void processPID(float _setPoint)
{
    value = getPercValue();

    if (isDirectControl)
        erro = value - _setPoint; // Ação direta.
    else
        erro = _setPoint - value; // Ação reversa.

    uk = uk1 + ((Kp + Ki + Kd) * erro) - ((Kp + (2 * Kd)) * erro1) + (Kd * erro2);

    // Controla os limites superior e inferior para que a saída não fique saturada.
    uk = controlLimitValues(uk);

    uk1 = uk;
    erro2 = erro1;
    erro1 = erro;

    setPercValue(uk);
}

```

**Figura 34 - Cálculo do sinal de saída do controle.**  
**Fonte: Autoria própria.**

Após o cálculo da saída do controlador ser realizado, este valor é verificado para impedir que a saída sature muito além dos limites inferior ou superior configurados no menu adequado. Com isto, o controle tem a capacidade de recuperar o sinal de saída, para valores entre 0% e 100%, conforme as variações dos sinais de entrada.

Outra característica presente no controlador é a capacidade de salvar e carregar, da memória EEPROM interna do dsPIC, 5 configurações escolhidas no menu, facilitando assim a demonstração de controles previamente testados.

Através da substituição do dsPIC – em que 95% da memória de programa foi utilizada – por outro de mesmo tamanho físico e com memórias de programa e de dados com maiores capacidades, é possível implementar mais características ao aparelho, como mais opções no menu, possibilidade de salvar mais configurações, utilizar mais arranjos de controle (paralelo, misto) bem como criar funcionalidades para comunicação com um PC. Embora esta última característica não exista nesta versão do aparelho, há um conector serial – denominado Com. Serial P4 no diagrama e placa da CPU – destinado a esta possibilidade.

## 4.4 CHASSI DO CONTROLADOR PID

Todo o *hardware* precisou ser envolvido por um *chassi* protetor, conforme Figura 35, feito de chapas de aço para proteção e manuseio.



**Figura 35 - Chassi metálico do controlador PID.**  
Fonte: Aatoria Própria.

Para o projeto do *chassis* foi utilizado o *software SolidWorks* da *Autodesk*, o qual possui o módulo *sheet-metal* que oferece recursos avançados para modelamento 3D de peças fabricadas com chapas de aço. Os processos envolvidos na fabricação do *chassi* são: corte a laser, dobras, solda e pintura.

### 4.4.1 Corte a laser

O corte a laser tem grande precisão, na casa dos décimos de milímetros, garantindo assim as tolerâncias de dimensão, posição e montagem. Não possui um custo muito elevado e dá um bom acabamento, não deixa rebarbas.

#### 4.4.2 Dobra

Para as dobras foram usadas dobradeiras convencionais, sem necessidade de ferramentas.

#### 4.4.3 Solda

Foram usados dois tipos de solda, capacitiva e por arco elétrico (MIG). A solda capacitiva foi utilizada para soldar as barras roscadas. Além disso foi utilizada uma ferramenta de solda tipo pistola que garante a orientação das barras.

#### 4.4.4 Pintura

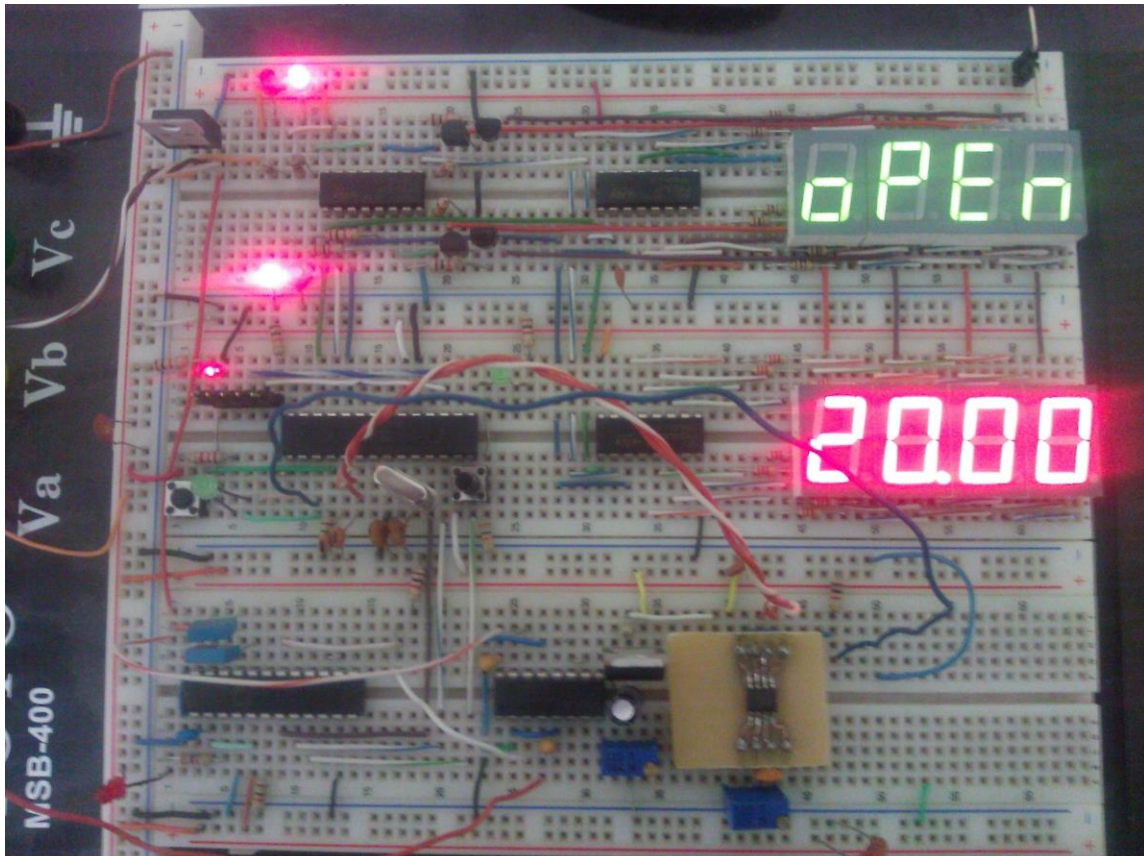
A pintura utilizada é baseada em tinta em pó Epóxi, aplicada eletrostaticamente e cura sob ação do calor. A tinta escolhida é de cor preta e microtexturizada.

Para a fabricação do *chassi* foram realizados desenhos, detalhamento técnico, com medidas e informações necessárias, ver **APÊNDICE C**.



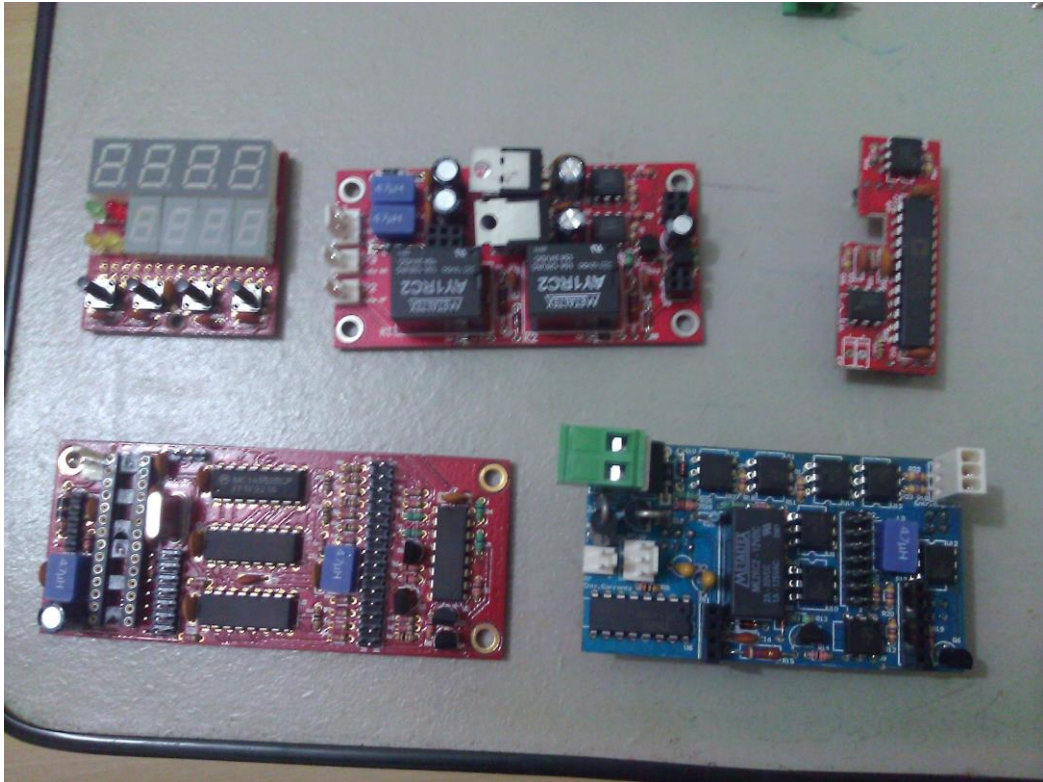
## 5 DESCRIÇÃO DA MONTAGEM

Inicialmente, foram testados no *protoboard* o funcionamento do painel (*displays*, botões e *LEDs*) dos CIs responsáveis pela aquisição e envio de dados (RCV420, MCP3553, AD420) e do PWM, como ilustrado na Figura 36.

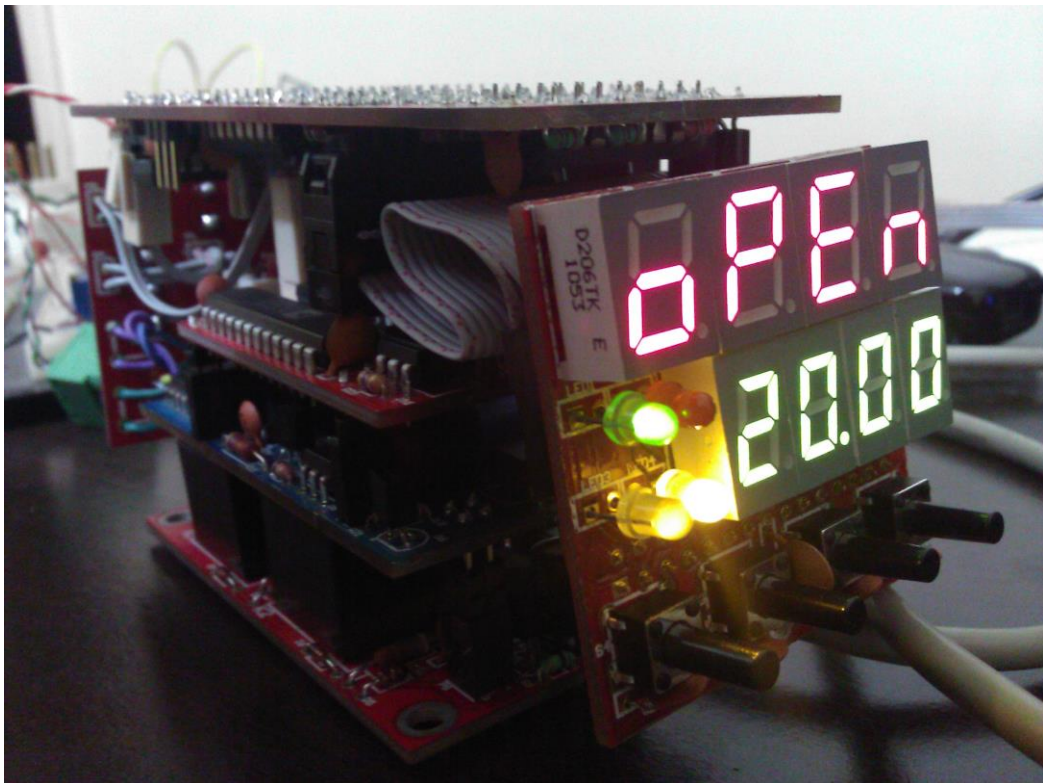


**Figura 36 - Verificação do funcionamento do hardware no *protoboard*.**  
Fonte: Autoria Própria.

Após a verificação do funcionamento do *hardware*, foram projetados os *layouts* das placas de circuito impresso utilizando o *software* Altium Designer 6. As placas foram produzidas no Laboratório de Microeletrônica da UTFPR. Elas possuem furos metalizados e circuito em ambas as faces, conforme a Figura 37. O processo utilizado na fabricação tem boa precisão e possibilitou uma rápida montagem final do *hardware*, conforme a Figura 38.



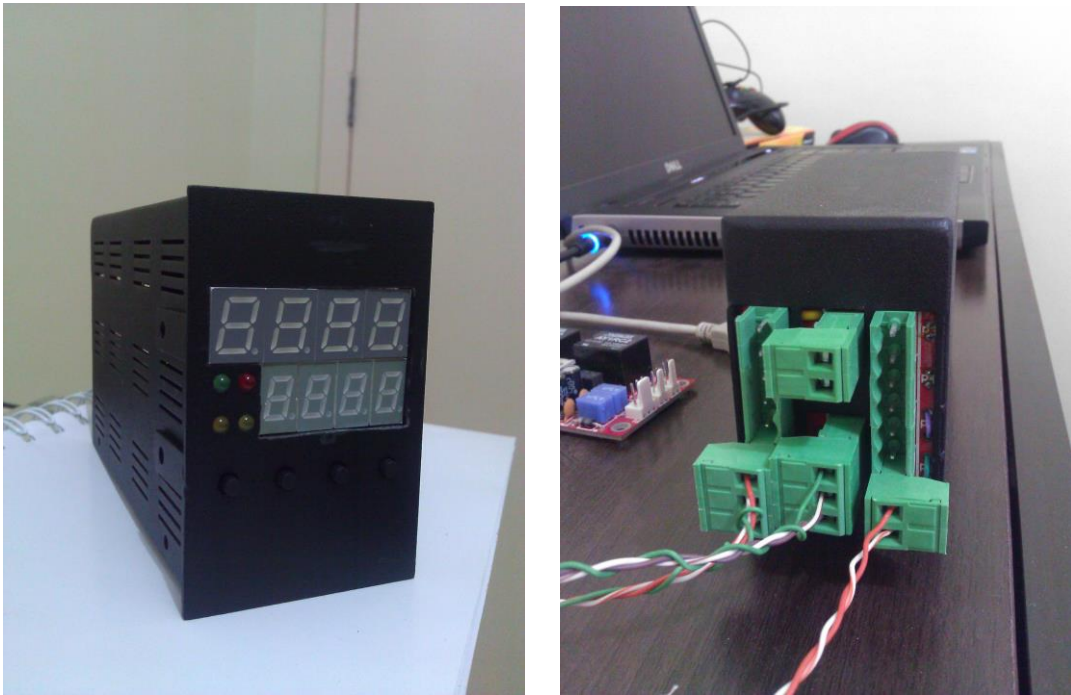
**Figura 37 - Placas de circuito impresso.**  
**Fonte: Autoria Própria.**



**Figura 38 - Montagem do hardware.**  
**Fonte: Autoria Própria.**



Após a montagem do *hardware*, foram feitos testes de funcionamento dos periféricos através de algumas rotinas programadas em linguagem C no microcontrolador. Concluído os testes, foram definidas as dimensões do *chassi*, o qual foi fabricado em uma indústria mecânica, ilustrado na Figura 39.



**Figura 39 - Vista frontal e da borneira do controlador.**  
**Fonte: Aatoria Própria.**

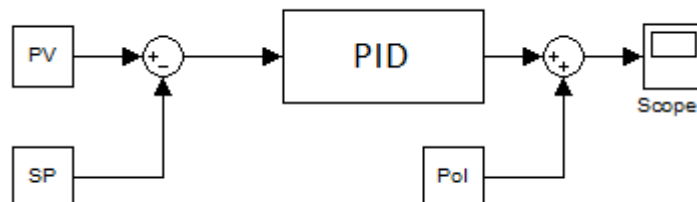
O custo referente à fabricação de dois protótipos completos esta discriminado no **APÊNDICE D**.

## 6 TESTES E RESULTADOS

Afim de verificar e validar o funcionamento do controlador foram realizados alguns testes. Para a realização destes testes foram simulados no Matlab três sistemas em malha aberta, um com controle somente proporcional, um com controle proporcional e integral, outro sistema com controle proporcional derivativo. Foram gerados os gráficos da resposta do sistema ao longo do tempo para analisar o comportamento deste de acordo com os valores de parâmetros estipulados, por exemplo, degraus, ganhos, tempo integrativo e tempo derivativo. Em seguida, estes valores foram incluídos nos parâmetros do controlador desenvolvido, e então foram executados os testes analisando os sinais através do osciloscópio, para então comparar os resultados obtidos pelo controlador com os resultados obtidos pelo MatLAB.

### 6.1 CONTROLE PROPORCIONAL

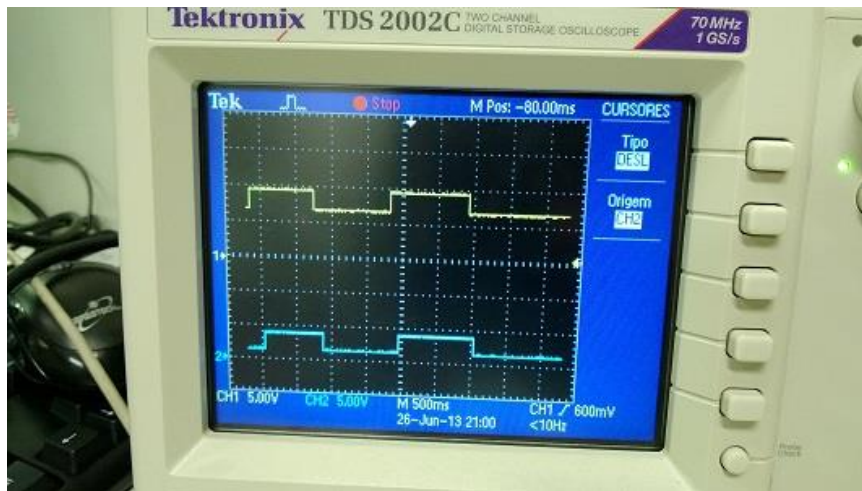
O diagrama em blocos ilustrado na Figura 40 representa o controle em malha aberta e seus principais parâmetros.



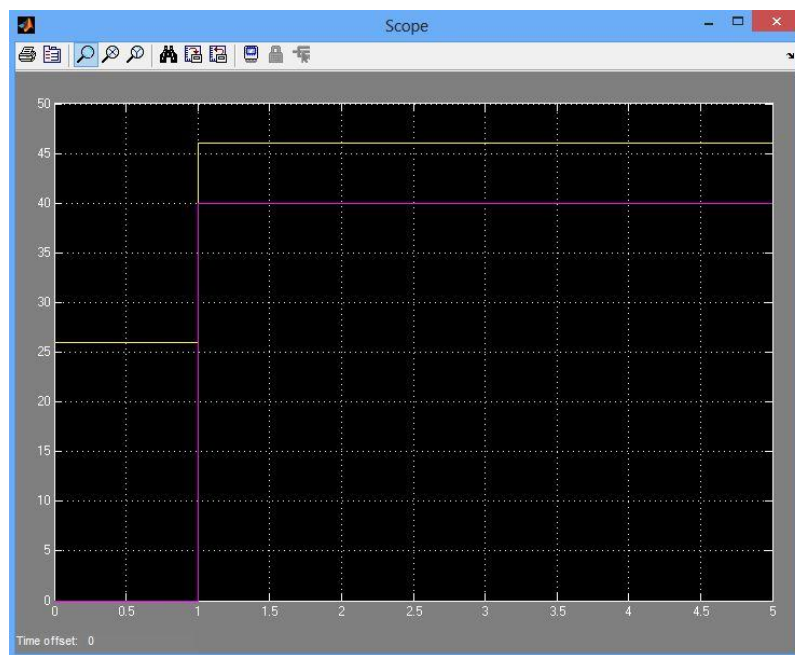
**Figura 40 - Diagrama em blocos (Controle em malha aberta).**  
**Fonte: Autoria Própria.**

O primeiro teste realizado foi do controle proporcional. Os parâmetros adotados na simulação foram: Ganho proporcional ( $K_P$ ) = 2,0; Período ( $T$ ) = 100ms. O *setpoint* (SP) e variável controlada (PV) foram ajustados para 26% do range, PID com erro zero. A variável controlada foi alterada de 26% para 46%, gerando um erro de 20% no sistema. Foi verificado o comportamento da variável manipulada, comprovando a atuação do controle proporcional que aumentou a saída em 40%,

devido ao ganho estipulado. As Figuras 41 e 42 mostram os resultados obtidos nas simulações práticas e teóricas, respectivamente.



**Figura 41 - Resposta ao degrau no controle Proporcional (Controlador).**  
Fonte: Autoria Própria.

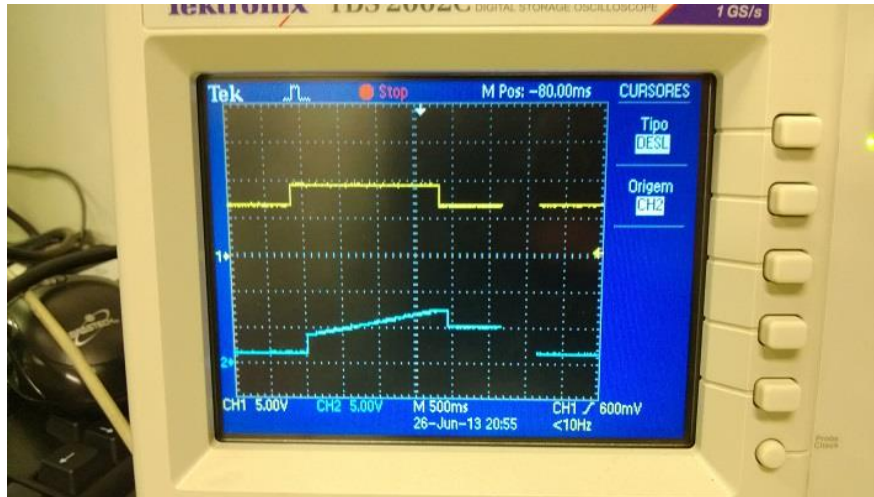


**Figura 42 - Resposta ao degrau no controle Proporcional (Simulink).**  
Fonte: Autoria Própria.

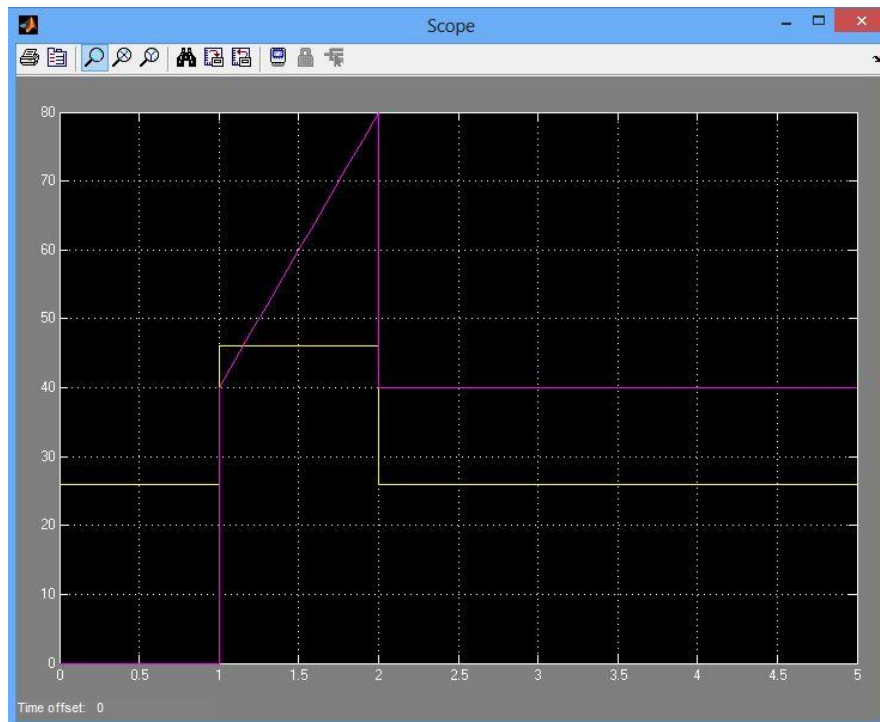
## 6.2 CONTROLE PROPORCIONAL INTEGRAL

O segundo teste realizado foi do controle proporcional integral. Os parâmetros adotados na simulação foram: Ganho proporcional ( $K_P$ ) = 2,0; Tempo integral ( $T_i$ ) = 1,0 s; Período ( $T$ ) = 100 ms. O SP e a PV foram ajustados para 26% do range, o PID

com erro zero. A variável controlada foi alterada de 26% para 46%, gerando um erro de 20% no sistema. Foi verificado o comportamento da variável manipulada comprovando a atuação do controle proporcional-integral que aumentou a saída devido ao ganho estipulado. As Figuras 43 e 44 mostram os resultados obtidos nas simulações práticas e teóricas, respectivamente.



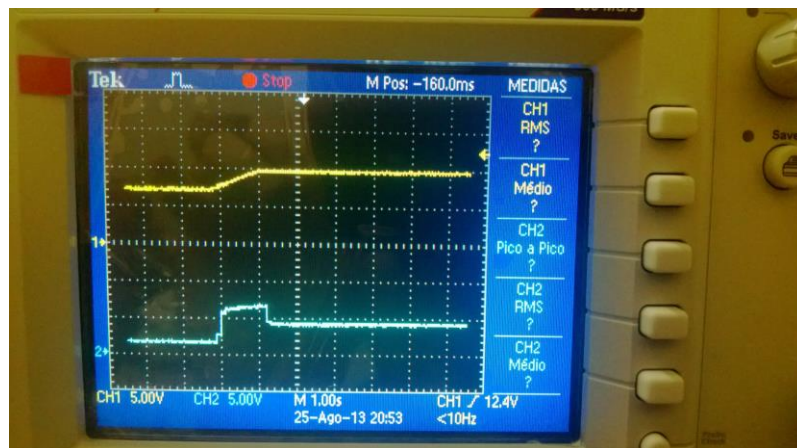
**Figura 43 - Resposta ao degrau no controle Proporcional-Integral (Controlador).**  
Fonte: Autoria Própria.



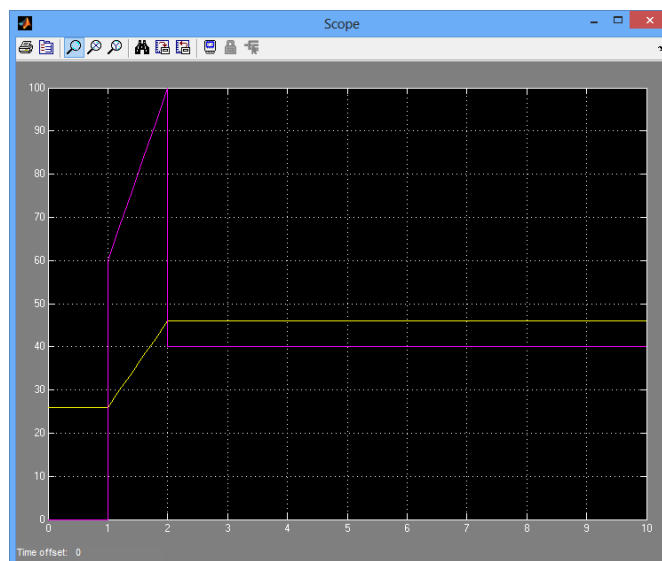
**Figura 44 - Resposta ao degrau no controle Proporcional-Integral (Simulink).**  
Fonte: Autoria Própria.

### 6.3 CONTROLE PROPORCIONAL DERIVATIVO

O terceiro teste realizado foi do controle proporcional derivativo. Os parâmetros adotados foram: Ganho proporcional ( $K_P$ ) = 2,0; Tempo derivativo ( $T_d$ ) = 1,5 s; Período ( $T$ ) = 100 ms. O SP e a PV foram ajustados para 26% do *range*. O PID foi iniciado com erro zero. A variável controlada foi alterado de 26% para 46%, gerando um erro de 20% no sistema. Foi verificado o comportamento da variável manipulada comprovando-se a atuação do controle proporcional derivativo. As Figuras 45 e 46 mostram os resultados obtidos nas simulações práticas e teóricas respectivamente.



**Figura 45 - Resposta ao sinal de rampa no controle Proporcional-Derivativo (Controlador)**  
 Fonte: Autoria Própria.



**Figura 46 - Resposta ao sinal de rampa no controle Proporcional-Derivativo (Simulink)**  
 Fonte: Autoria Própria.

## 7 CONCLUSÃO

Este trabalho surgiu após a equipe observar a utilização de alguns controladores mais simples, sem controle PID e com poucos periféricos, por empresas de pequeno porte. Deste modo resolveu desenvolver um controlador que oferecesse mais funções e que fosse de baixo custo. Nossos protótipos tiveram custos elevados devido a pouca quantidade fabricada. É preciso realizar cotações com os diversos fornecedores para avaliar o custo real do produto.

Inicialmente foi feito um *benchmarking*, através da internet, analisando as melhores práticas na indústria, verificando os recursos oferecidos pelos produtos disponíveis no mercado. Após o levantamento dos recursos que nosso controlador ofereceria, foi feita uma busca através dos sites de fabricantes de componentes eletrônicos para determinar os componentes necessários. A escolha de cada componente foi criteriosa, dependendo algumas vezes do custo e também da resolução dos CIs, pois queríamos uma boa precisão no controle. Foi necessário importar alguns componentes devido a algumas funcionalidades específicas (AD420, MCP3553, RCV420).

As atividades realizadas até esta etapa podem ser relacionadas com os departamentos de *marketing* e compras de uma empresa. Esta observação nos mostra que para desenvolver um produto é preciso envolver outras áreas e outros conhecimentos, além da área técnica.

A implementação do *hardware* e *firmware* foi realizada paralelamente, testando e observando o comportamento de cada componente. Vários problemas ocorreram na transmissão de informações, devido ao ruído, causado principalmente pelos relés. Além de perda de dados relacionadas as rotinas de programação. Essas dificuldades fez a equipe se aprofundar em assuntos referentes a interferências eletromagnéticas e aprender boas práticas de programação.

Para o desenvolvimento do controle PID foi implementada uma equação discreta. Esta equação possibilitou a análise de cada termo durante a implementação e tornou fácil as simulações, que por sua vez foram comparadas com as análises realizadas no Matlab. Neste trabalho de conclusão de curso foram aplicados conhecimentos das três áreas de estudo no curso de Tecnologia em Mecatrônica Industrial; mecânica, eletrônica e informática.



## REFERÊNCIAS

ALTIUM Limited. Disponível em:

<<http://www.altium.com>>. Acessado em: 13 ago. 2013

CAMPOS, Mário Cesar M. Massa; TEIXEIRA, Herbert Campos Gonçalves. **Controles típicos de equipamentos e processos industriais**. Editora Edgard Blucher, São Paulo – SP. 2006.

FONSECA, M. O. Constantino, TORRES, S.. **Avaliação de desempenho e auditoria de malhas de controle**. Intech Brasil 63, 2004.

IWASSE, Felipe. 2009. **Análise dos Arranjos Estruturais de Controladores Comerciais – Monografia – Especialização**, Curitiba – PR

KOUYOUMDJIAN, Ara. 1998. **A Compatibilidade Eletromagnética**, Editora MM – 1ª Edição, São Paulo - SP.

MICROCHIP TECHNOLOGY. Datasheet: DS70118H dsPIC30F2010. Electronic Publication, 2008.

MICROCHIP TECHNOLOGY. Datasheet: DS39582B Pic16F873. Electronic Publication, 2003.

MICROCHIP TECHNOLOGY, Disponível em:

<<http://www.scribd.com/doc/22529943/Tecnicas-de-Robustez-em-Projetos-para-Reducao-de-EMI-e-maior-Imunidade-dos-Circuitos-contr-ESD>>. Acessado em: 13 ago. 2013

NOVUS Produtos Eletrônicos Ltda. Disponível em:

<<http://www.novus.com.br>>. Acessado em: 13 ago. 2013

OGATA, Katsuhiko **Engenharia de Controle Moderno**. 4a Edição.

Editora Prentice Hall. 2003.

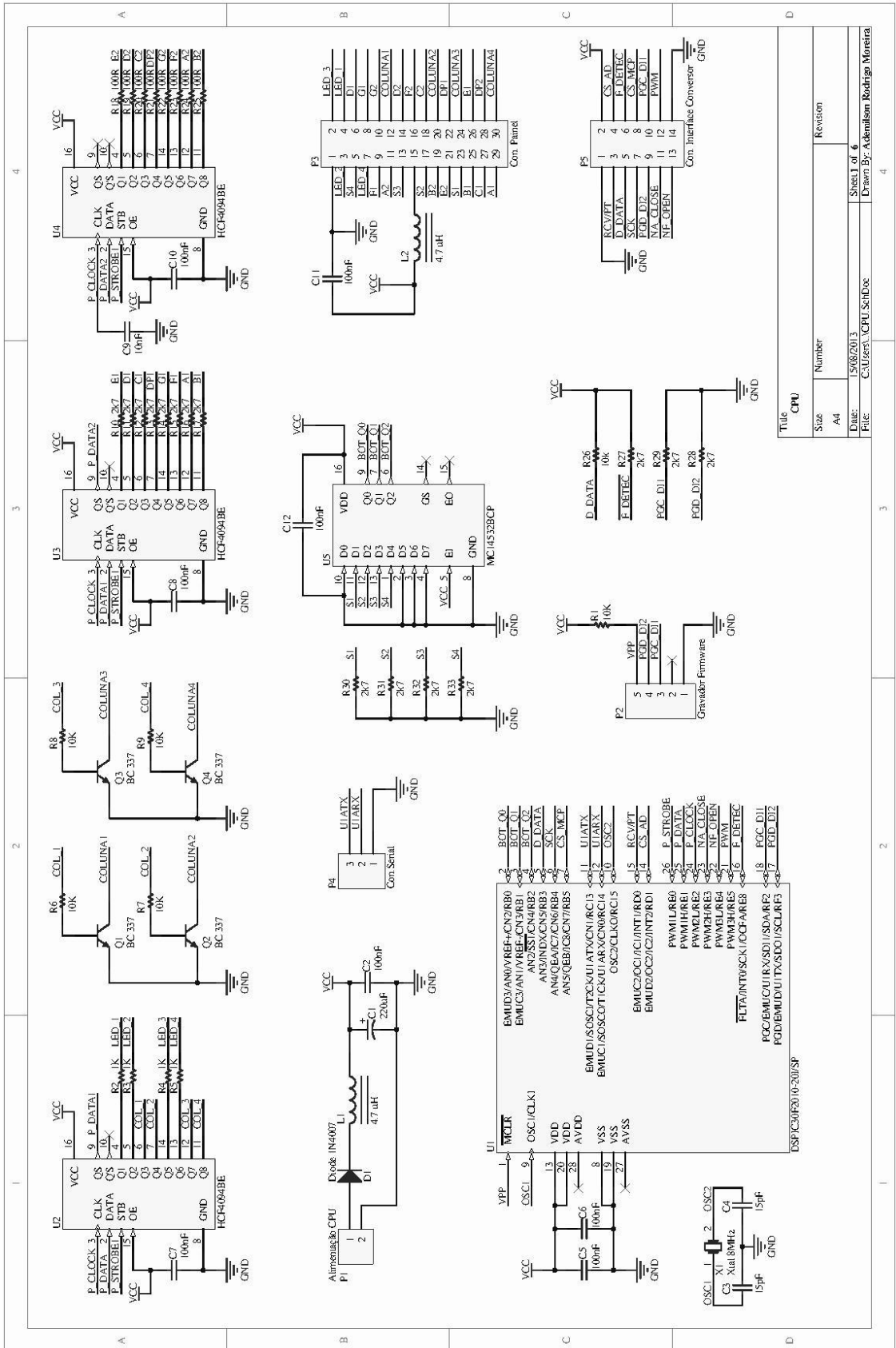
OLIVEIRA, André Schneider; ANDRADE, Fernando Souza. 2006. **Sistemas Embarcados – Hardware e Firmware na Prática**, Editora Érika – 1ª Edição, São Paulo - SP.

RAMOS, Carlos João Rodrigues Costa. 1997. **Sistema de Recuperação da Energia de Deslizamento Aplicado ao Aproveitamento da Energia das Ondas Marítimas – Dissertação – Mestrado**, Porto

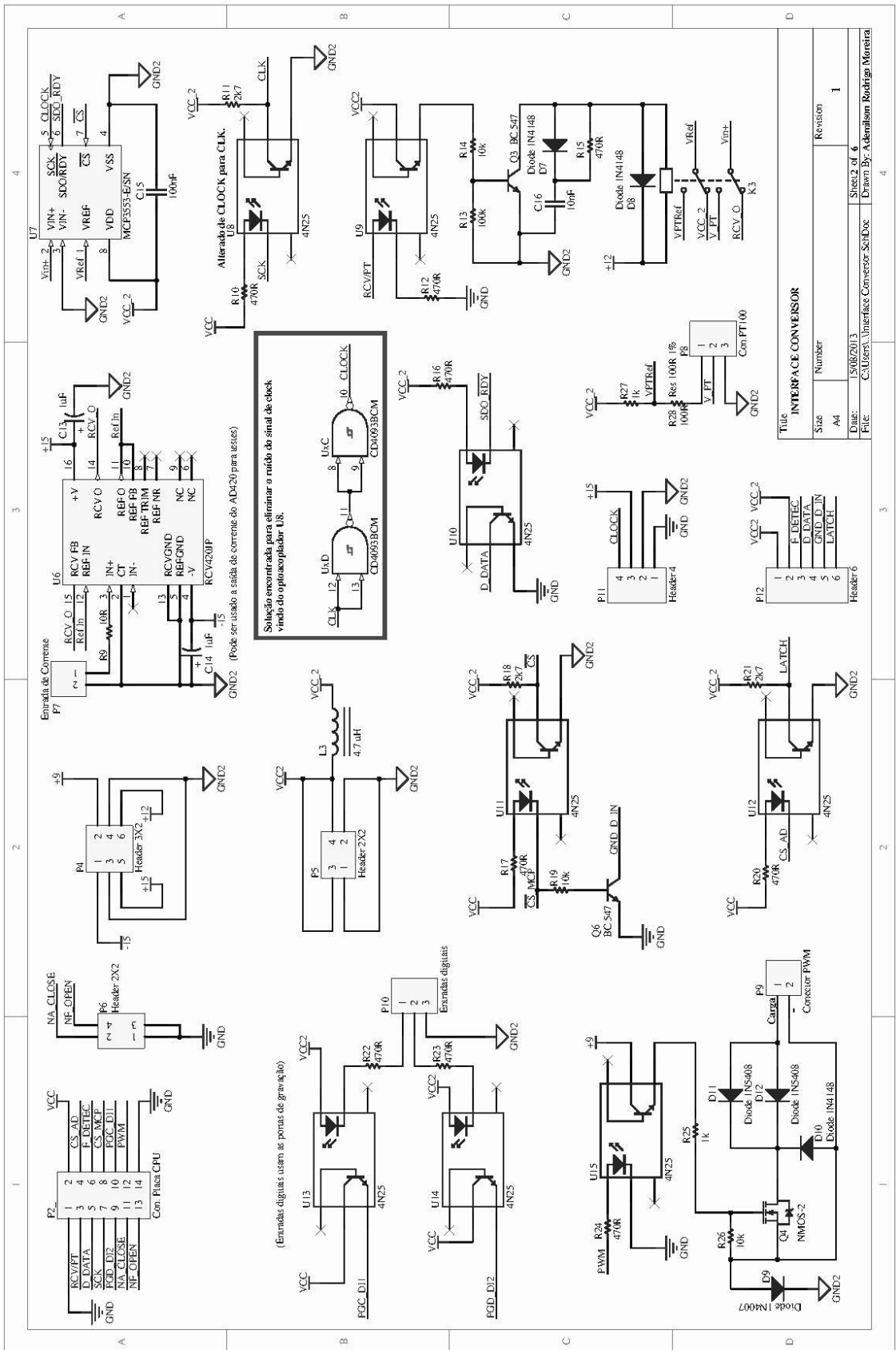
SANCHES, Durval. 2003. **Interferência Eletromagnética**, Editora Interciência – 1ª Edição, Rio de Janeiro - RJ.

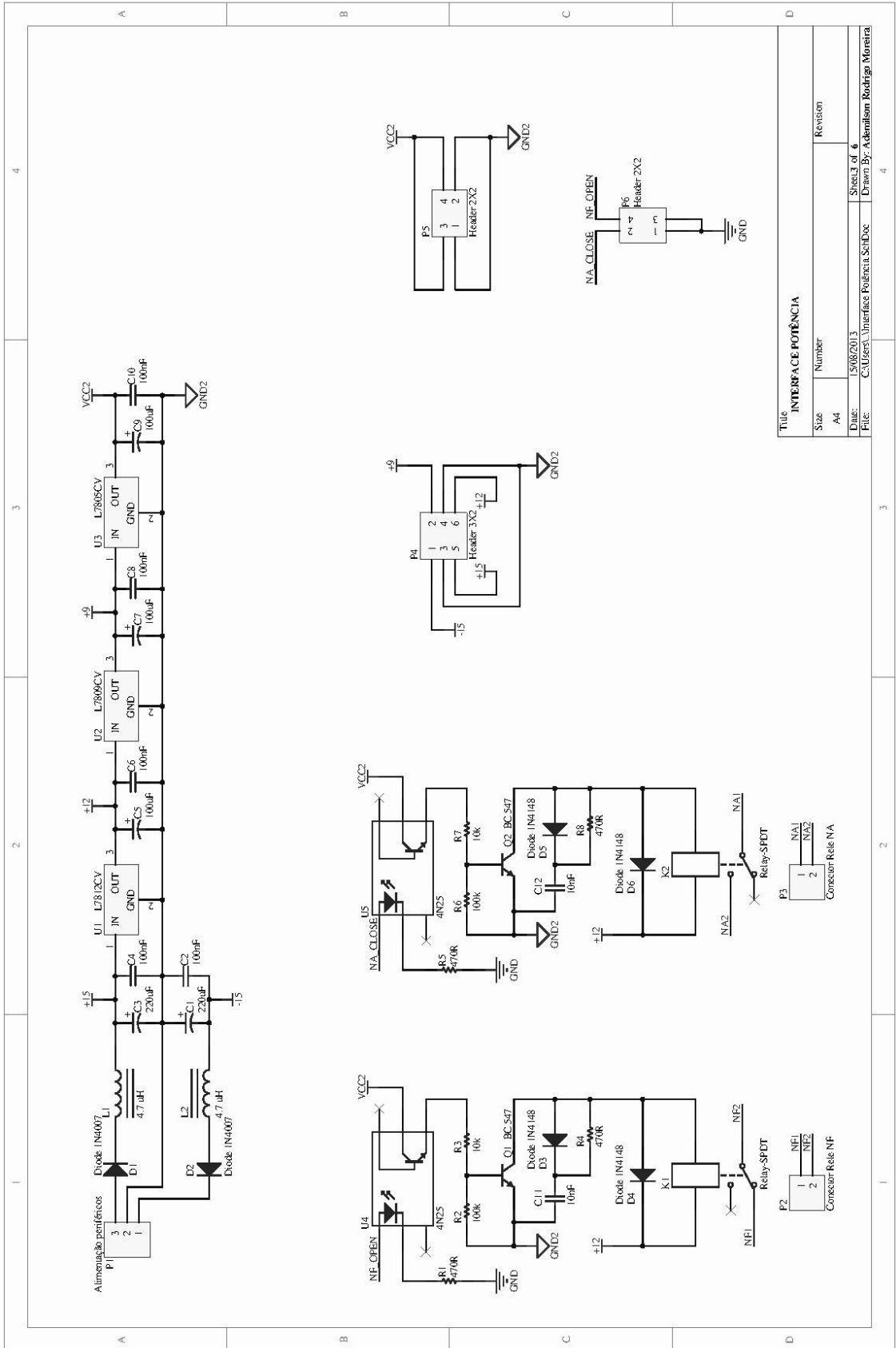
SOLIDWORKS Corp.. Disponível em:  
<<http://www.solidworks.com>>. Acessado em: 13 ago. 2013

## **APÊNDICE A**



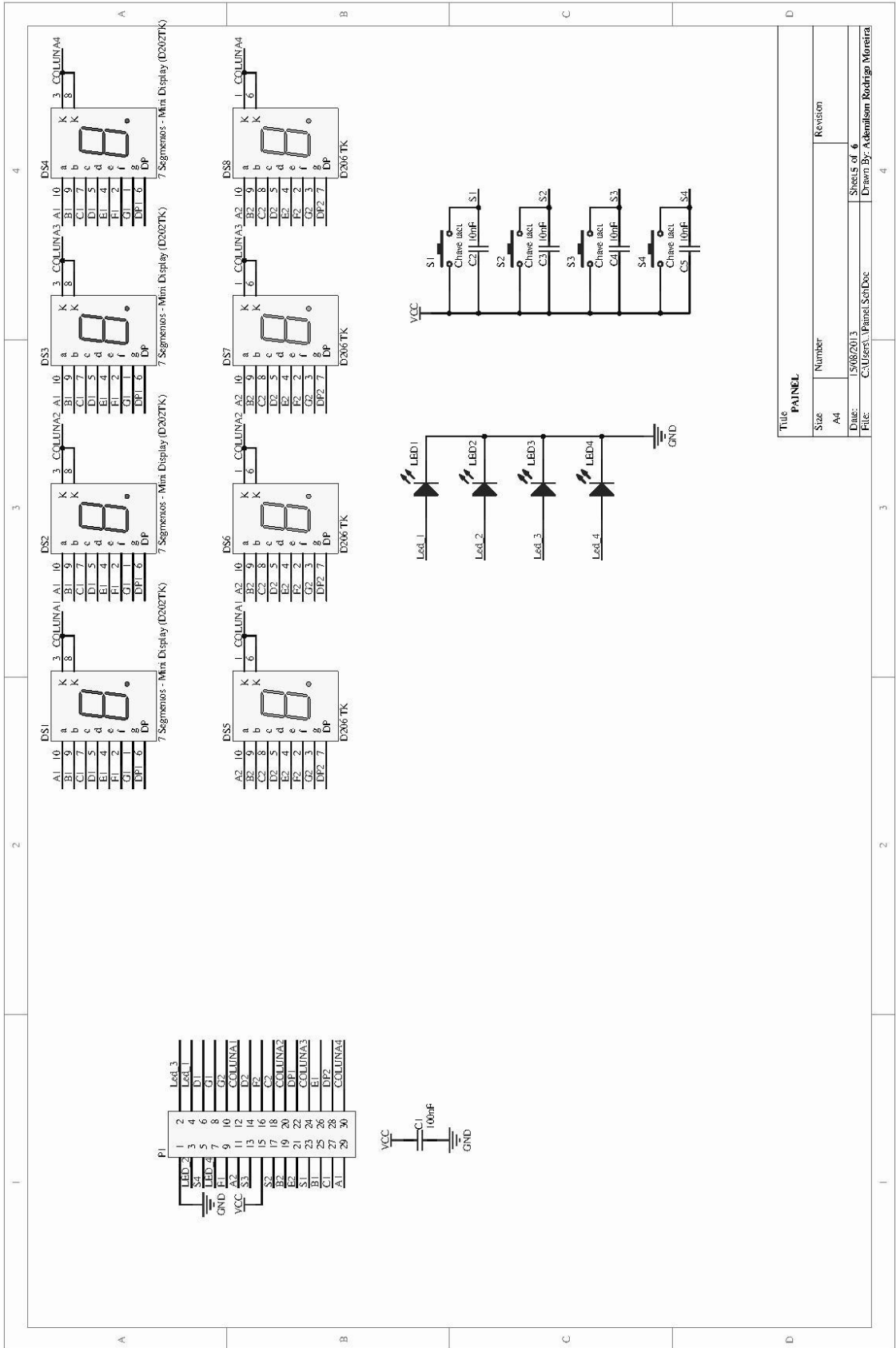
Title	Size	Number	Revision
CPU	A4		
Date:	15/08/2013		Sheet 1 of 6
File:	C:\Users\CPUI\SchDoc		Drawn By: Ademilson Rodrigues Monteiro



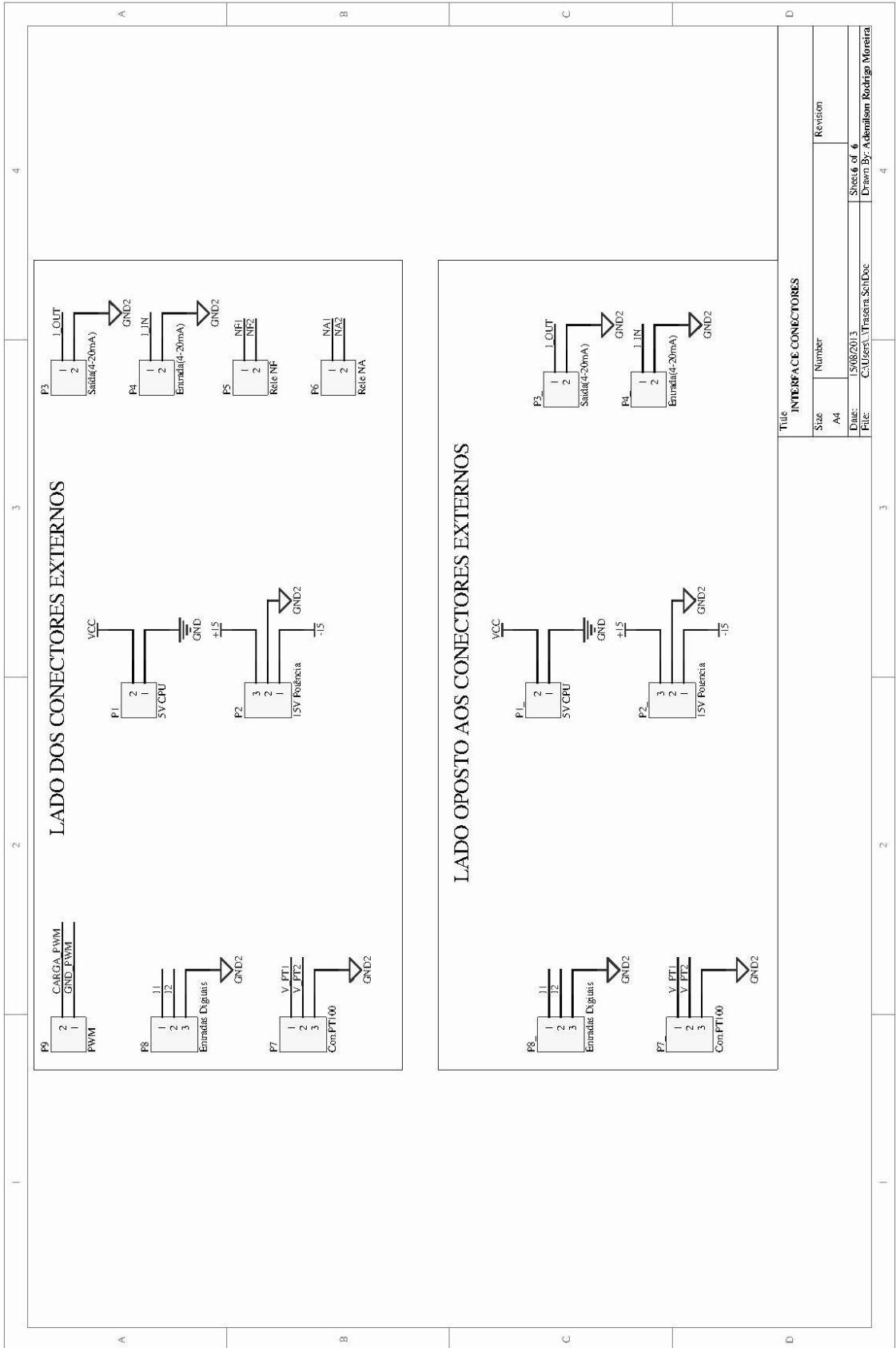


Título		
INTERFAÇE POTÊNCIA		
Size	Number	Revision
A4		
Date:	15/06/2013	
File:	C:\Users\Interface Potência\Src\Doc	
Sheet of 6		Drawn By: Ademilson Rodrigo Moreira

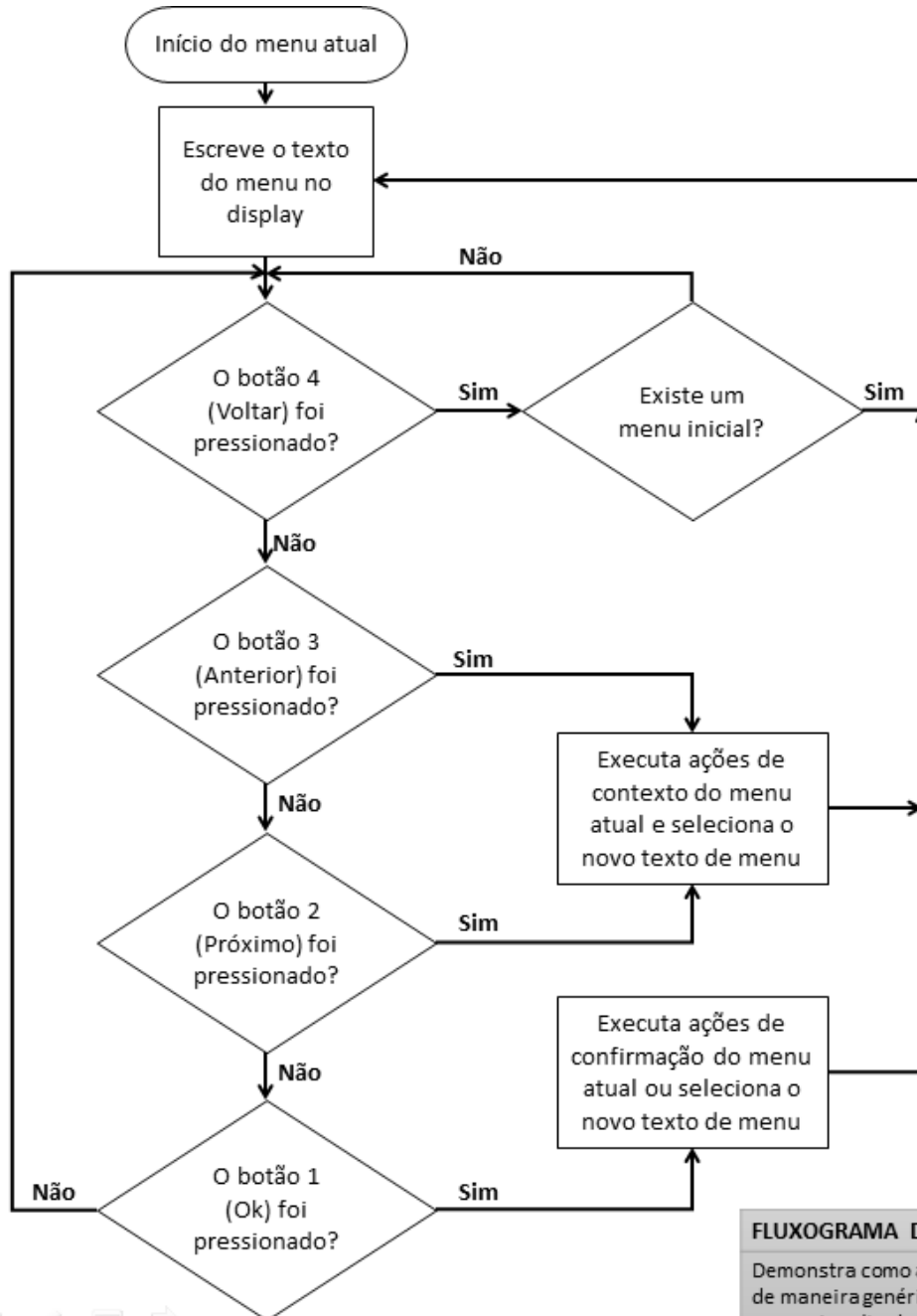








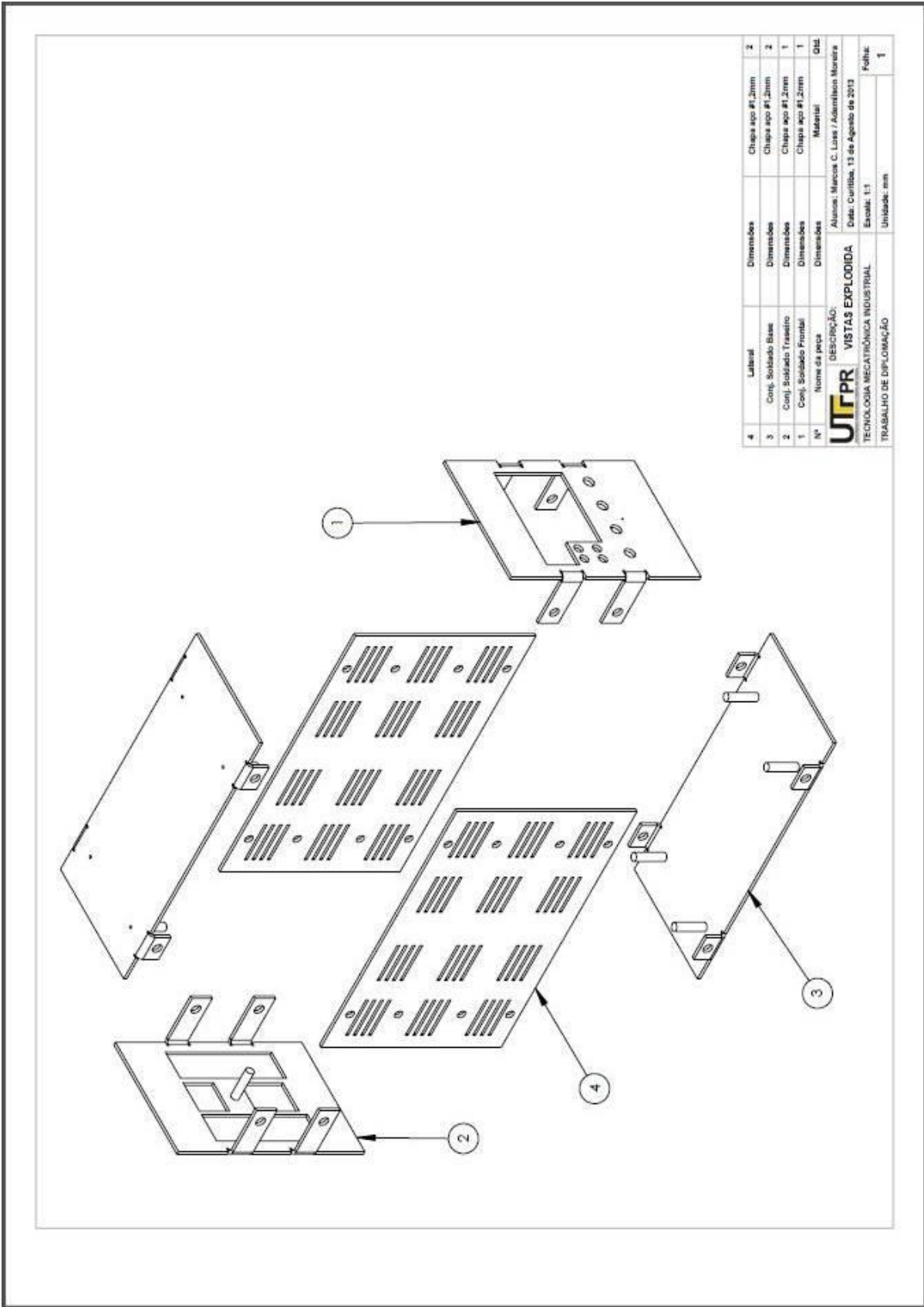
## **APÊNDICE B**



#### FLUXOGRAMA DA TROCA DOS MENUS

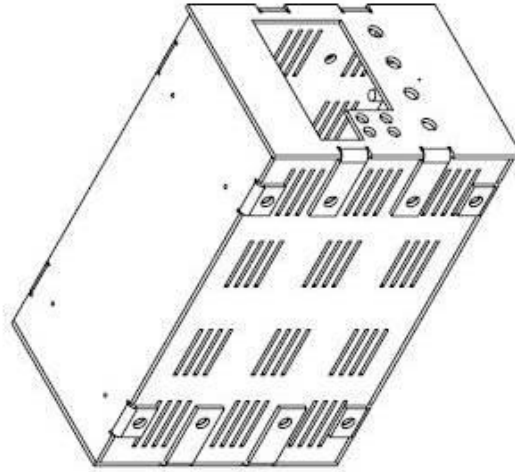
Demonstra como a troca dos menus é feita de maneira genérica, não importando qual o menu visualizado no momento.

## **APÊNDICE C**

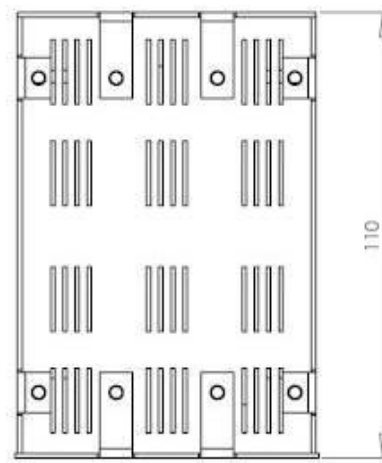
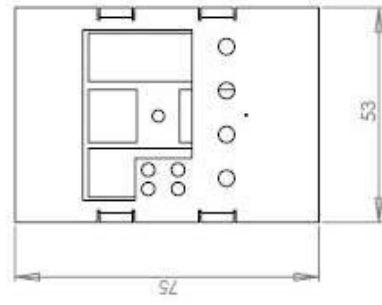


4	Lateral	Dimensões	Chapa aço #1,2mm	2
3	Comp. Soldado Base	Dimensões	Chapa aço #1,2mm	2
2	Comp. Soldado Traseiro	Dimensões	Chapa aço #1,2mm	1
1	Comp. Soldado Frontal	Dimensões	Chapa aço #1,2mm	1
Nº	Nome da peça	Dimensões	Material	Qtd.

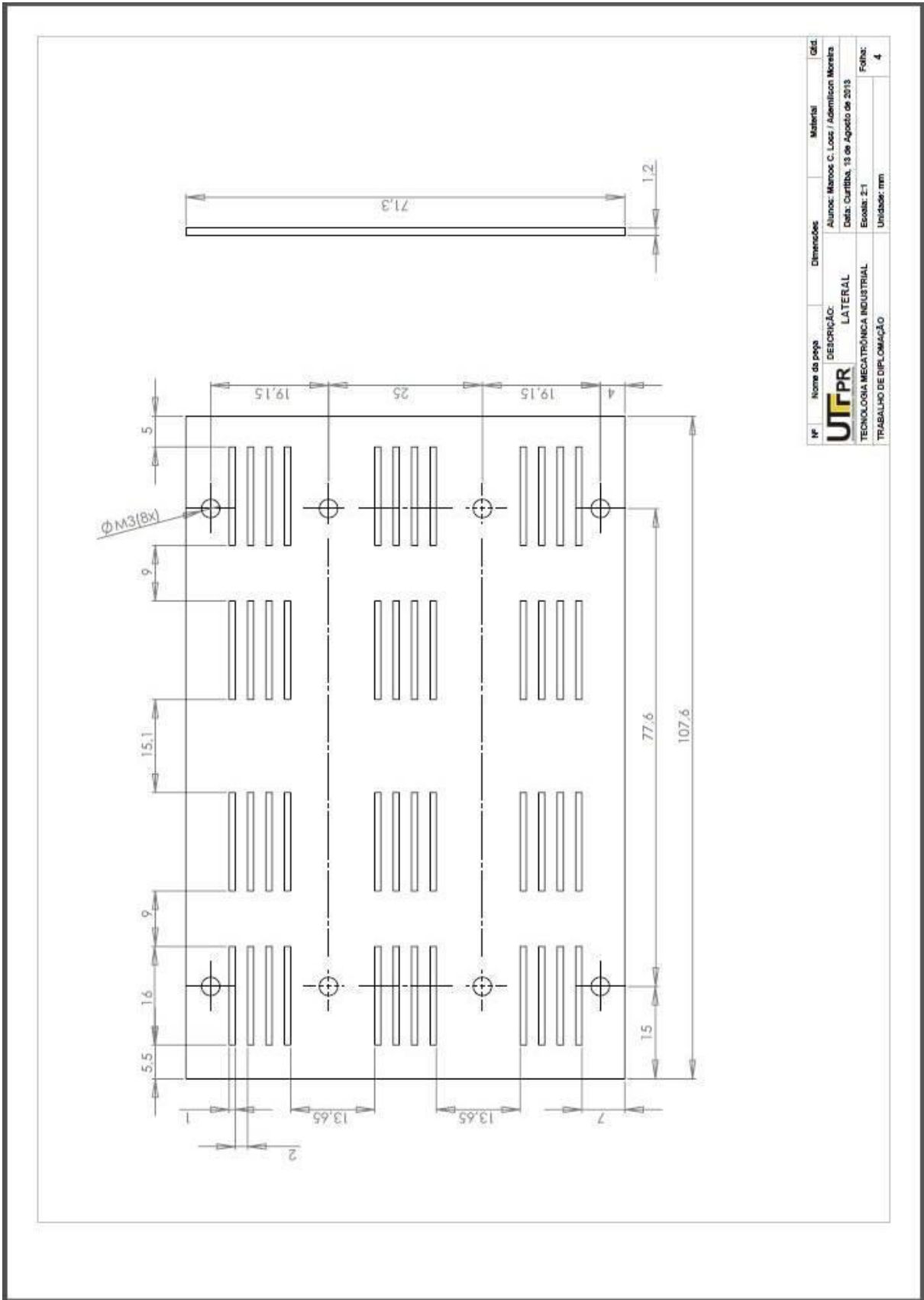
**UTPR** VISTAS EXPLODIDA  
 Alunos: Marcos C. Lora / Admilson Moreira  
 TECNOLOGIA MECATRÔNICA INDUSTRIAL Data: Curitiba, 13 de Agosto de 2013  
 Escala: 1:1  
 TRABALHO DE DIPLOMAÇÃO Unidade: mm  
 Folha: 1



Nº	Nome da peça	Dimensões	Material	Qtd.
	UTPR			
DESCRIÇÃO: PERSPECTIVA				
ALUNO: MARCOS C. LIMA / ADMISTRAÇÃO MARCINA				
DATA: CURITIBA, 15 de Agosto de 2013				
TECNOLOGIA MECATRÔNICA INDUSTRIAL				
ESCALA: 2:1				
FOLHA:				
TRABALHO DE DIPLOMAÇÃO				2
				Unidade: mm

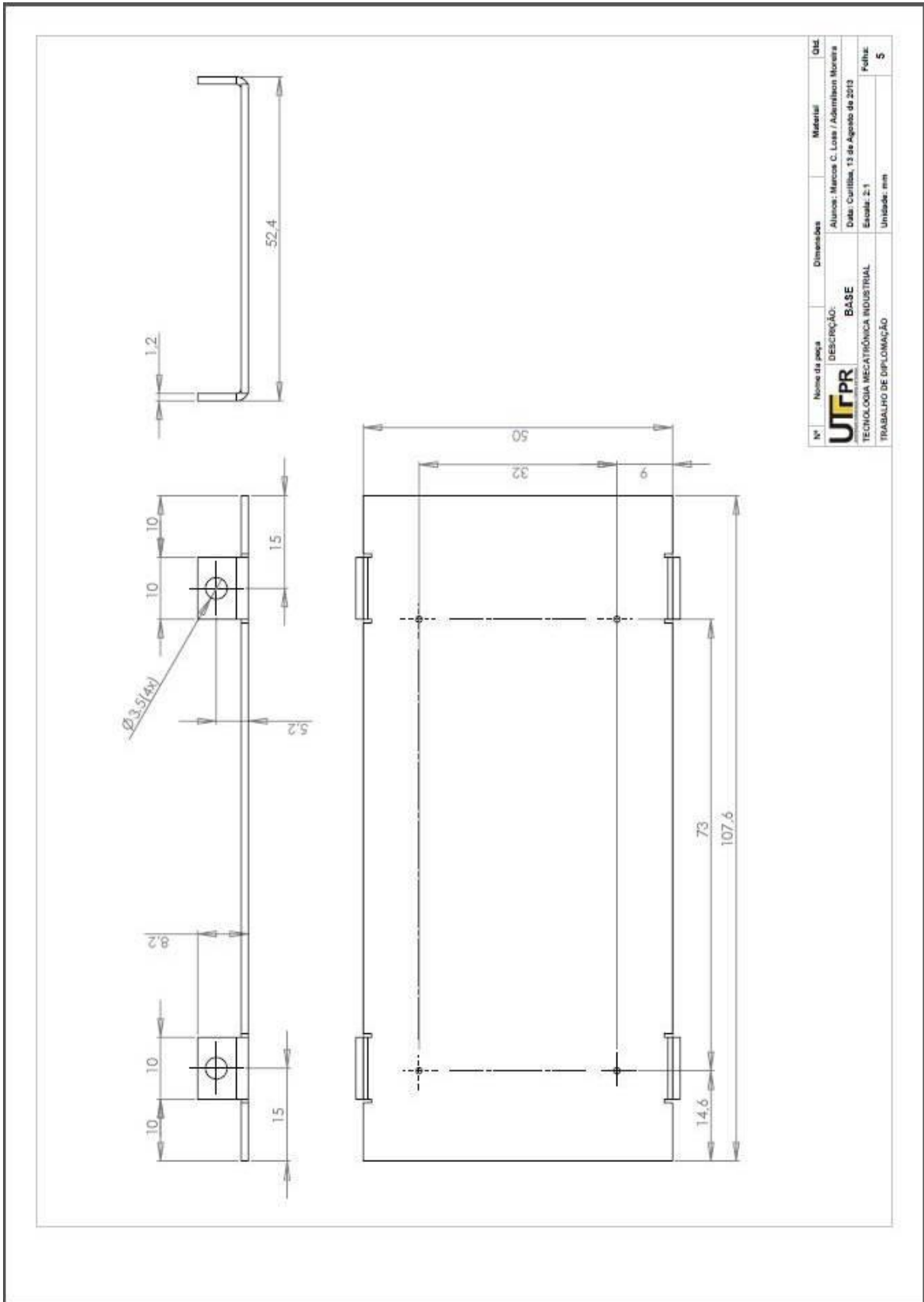


Nº	Nome da peça	Dimensões	MATERIAL	Qtd.
	DESCRIÇÃO: <b>UTPR</b> VISTAS ORTOGONAIS		Aluon: Marcas C. Ltda / Admissão Manobra	
	TECNOLOGIA MECATRÔNICA INDUSTRIAL		Data: Curitiba, 15 de Agosto de 2013	
	TRABALHO DE DIPLOMAÇÃO		Escala: 1:1	Folha:
			Unidade: mm	3

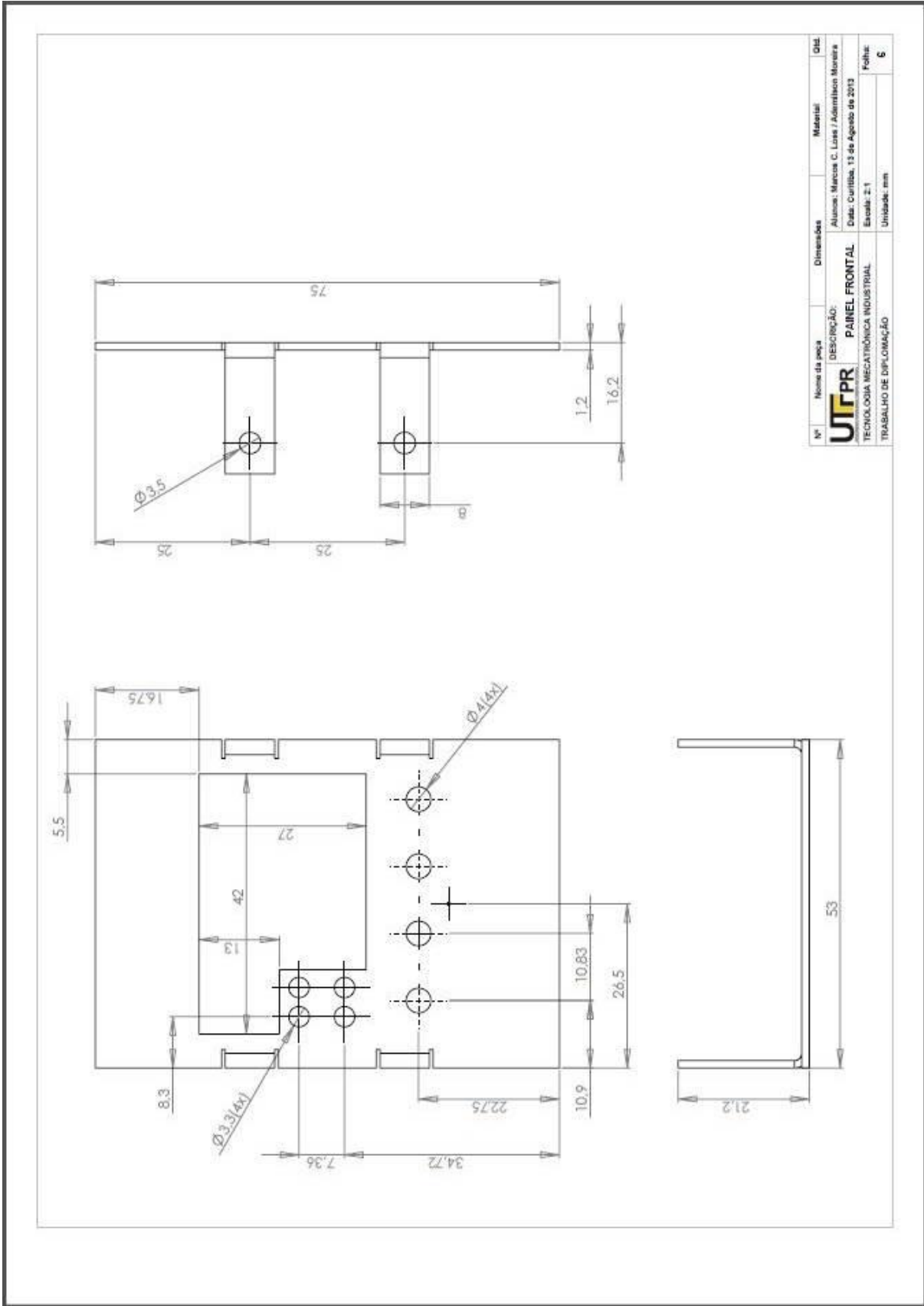


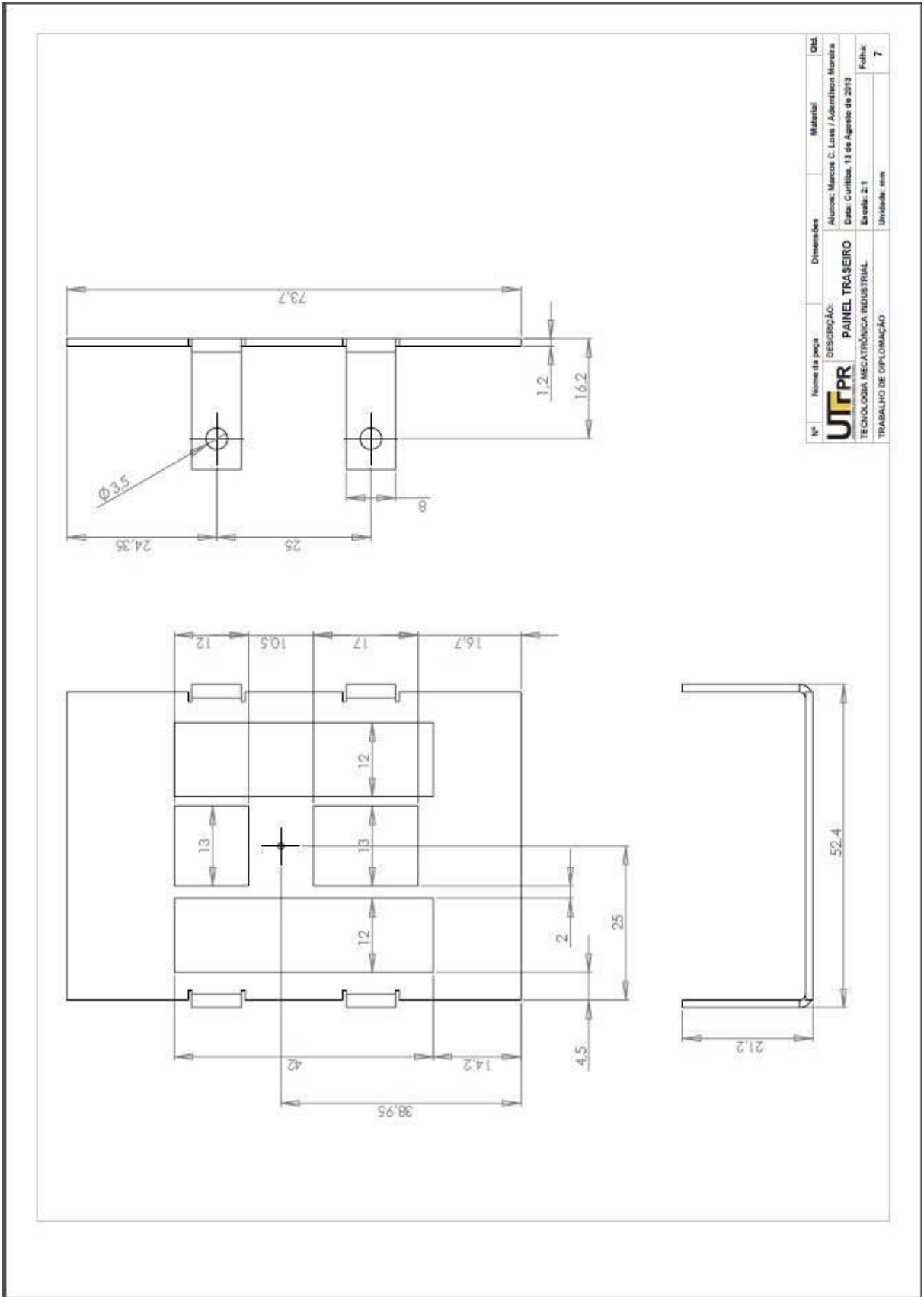
Nº	Nome da peça	Dimensões	Material	Qtd.
	DESCRIÇÃO: <b>UTPR</b> LATERAL		Aluno: Marcos C. Loos / Admissão Moreira	
	TECNOLOGIA MECATRÔNICA INDUSTRIAL		Data: Curitiba, 19 de Agosto de 2019	
	TRABALHO DE DIPLOMAÇÃO		Escola: ZT	Folha: 4
			Unidade: mm	





Nº	Nome da peça	Dimensões	MATERIAL	Qtd.
	UTPR			
	DESCRIÇÃO:			
	BASE			
	TECNOLOGIA MECATRÔNICA INDUSTRIAL			
	TRABALHO DE DIPLOMAÇÃO			
	Aluno: Marcos C. Lobo / Admilson Moreira			
	Data: Curitiba, 12 de Agosto de 2013			
	Escala: 2:1			
	Unidade: mm			
	Folha:			5





Nº	Nome da peça	Dimensões	Material	Qtd.
	DESIGNAÇÃO: <b>PAINEL TRASEIRO</b>			
	Alunos: Marcos C. Lopes / Ademilson Moreira			
	Data: Curitiba, 13 de Agosto de 2013			
	TECNOLOGIA MECATRÔNICA INDUSTRIAL	Escola: 2.1		Folha:
	TRABALHO DE DIPLOMAÇÃO	Unidade: mm		7

## **APÊNDICE D**

## CUSTOS

- Componentes

Fornecedores: Digi-Key; Croma

Frete + Taxas = R\$40 + R\$60,00 + R\$24,00

Itens = R\$70 + R\$366,00 + R\$160,00

Valor total = R\$596,00

- Placas de circuito impresso

Fornecedor: Laboratório UTFPR

Itens = R\$380,00

- Mecânica

Fornecedores: Maxestampo; Bruson

Corte e Dobra = R\$330,00

Pintura = R\$100,00

Valor total = R\$430,00

- Aquisições

- PT100 = R\$60,00

- Osciloscópio = R\$1080,00

- Fonte de tensão = R\$680,00