

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE MECÂNICA
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

FERNANDA DORNELES MALAQUIAS
KARINA ASSOLARI
MARCOS LIKIO NOGAWA

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE UM
BRAÇO ROBÓTICO DE CINCO EIXOS**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2012

FERNANDA DORNELES MALAQUIAS

KARINA ASSOLARI

MARCOS LIKIO NOGAWA

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE UM BRAÇO
ROBÓTICO DE CINCO EIXOS**

Trabalho de Conclusão de Curso apresentado à disciplina de Trabalho de Diplomação, como requisito parcial para obtenção de grau de Tecnólogo em Mecatrônica Industrial dos Departamentos Acadêmicos de Eletrônica (DAELN) e Mecânica (DAMEC) da Universidade Tecnológica Federal do Paraná – UTFPR.

Orientador: Prof. M. Sc. Juliano Mourão Vieira.

CURITIBA
2012

FERNANDA DORNELES MALAQUIAS

KARINA ASSOLARI

MARCOS LIKIO NOGAWA

**DESENVOLVIMENTO DE UM SISTEMA DE CONTROLE DE UM BRAÇO
ROBÓTICO DE CINCO EIXOS**

Este trabalho de conclusão de curso foi apresentado no dia 09 de julho de 2012, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. O(s) aluno(s) foi(ram) arguïdos(s) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

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

Prof. Décio Estevão do Nascimento
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Luiz Carlos de Abreu Rodrigues

Prof. M. Sc. Juliano Mourão Vieira
Orientador

Prof. Dr. Rubens Alexandre de Faria

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

RESUMO

MALAQUIAS, Fernanda D.; ASSOLARI, Karina; NOGAWA, Marcos L. **Desenvolvimento de um sistema de controle de um braço robótico de cinco eixos**. 2012. 80 f. Trabalho de Conclusão de Curso (Curso de Tecnologia em Mecatrônica Industrial) – Departamentos Acadêmicos de Eletrônica (DAELN) e Mecânica (DAMEC), Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

Este projeto propõe o desenvolvimento de uma luva para controlar um braço robótico didático encontrado no Laboratório de Automação e Sistemas Inteligentes da Manufatura da Universidade Tecnológica Federal do Paraná, no *campus* Curitiba, e de uma nova interface de comunicação entre um computador e este robô, com o objetivo de tornar seu controle mais simples. A nova interface de comunicação foi desenvolvida no ambiente de programação Labview, adicionando novas funções para atender as necessidades dos usuários deste laboratório, tais como a possibilidade de salvar e carregar programas contendo sequência de comandos a serem enviados para o robô e a possibilidade de editar linhas específicas dos programas. Os componentes principais da luva são dois acelerômetros e um microcontrolador PIC. Para o desenvolvimento desta, foi necessário programar o microcontrolador em linguagem C e aplicar conhecimentos de cinemática inversa. Para facilitar a movimentação do robô com a luva foi necessário incluir um pequeno teclado (denominado teclado luva), que define quais juntas serão movimentadas e qual a sensibilidade do movimento. Foi criado também um novo controle (denominado teclado controle) que em conjunto com a luva e o teclado luva substituem o controle “*teaching Box*” original do braço robótico. O teclado controle tem a função de salvar posições e executar programas pré-definidos no robô. Para validação dos protótipos desenvolvidos foram realizados testes práticos destes, comunicando-os com o braço robótico. O robô foi movimentado com a luva auxiliada pelo teclado luva, o teclado controle executou as funções programadas corretamente e a nova interface de comunicação entre o computador e o robô também cumpriu os objetivos desejados. Os resultados obtidos foram considerados satisfatórios pelos membros da equipe, pois atenderam o objetivo geral do trabalho.

Palavras-chave: Sistemas hápticos. Controle de braço robótico. Interfaces de comunicação.

ABSTRACT

MALAQUIAS, Fernanda D.; ASSOLARI, Karina; NOGAWA, Marcos L. **Development of a control system for a five axes robotic arm.** 2012. 80 f. Trabalho de Conclusão de Curso (Tecnologia em Mecatrônica Industrial) – Departamentos Acadêmicos de Eletrônica (DAELN) e Mecânica (DAMEC), Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

This project describes the development of a glove to control a didactic robotic arm located at the Laboratory of Automation and Intelligent Systems for Manufacturing at Universidade Tecnológica Federal do Paraná, *campus* Curitiba, and a new communication interface between a computer and this robot, in order to make its controlling simpler. The new communication interface was developed using Labview, adding new functions to meet the needs of the users of this laboratory, such as the possibility of saving and loading programs with sequences of commands that will be transmitted to the robot and the possibility of editing specific lines of a program. The main components of the glove are two accelerometers and a PIC microcontroller. For its development it was necessary to program the microcontroller in C language and use knowledge of inverse kinematic. To make the robot movement easier using the glove it was necessary to include a small keyboard (named glove keyboard), that defines which joints will be moved and define the movement sensibility. A new controller was created (named controller keyboard) that, together with the glove and the glove keyboard, replaces the original teaching box of the robotic arm. The controller keyboard's purpose is to save positions and run pre-defined programs stored in the robot. To validate the developed prototypes, practical tests were executed, communicating them with the robotic arm. The robot was moved using the glove, aided by the glove keyboard; the controller keyboard executed the programmed functions properly; and the new communication interface between the computer and the robot also met the desired objectives. The results obtained were considered satisfactory by the members of the team, because they met the general objective.

Keywords: Haptic systems. Robotic arm control. Communication interface.

LISTA DE FIGURAS

Figura 1 – Diagrama de blocos da situação atual do laboratório	13
Figura 2 – Diagrama de blocos da sugestão de melhoria para o laboratório	14
Figura 3 – Exemplos de sistemas hápticos	17
Figura 4 – Esquemático de juntas e elos em braços robóticos	19
Figura 5 – Tipos de juntas e movimentação das juntas	20
Figura 6 – Exemplo de robôs com junta deslizante e rotativa	20
Figura 7 – Exemplificação de como se identificar grau de liberdade em juntas deslizantes	21
Figura 8 – Graus de liberdade do robô Mitsubishi RV-M1	22
Figura 9 – Associação do braço robótico com o braço humano	22
Figura 10 – <i>Teaching box</i>	23
Figura 11 – Caixa indicativa	24
Figura 12 – Distâncias físicas do robô	25
Figura 13 – Relação deslocamento/sinal de saída do acelerômetro	26
Figura 14 – Vistas inferior e superior do acelerômetro de modelo MMA7455L	27
Figura 15 – Funcionalidade geral dos microcontroladores	28
Figura 16 – Classificação dos microcontroladores PIC	29
Figura 17 – Exemplo de transmissão de dados pela porta serial	31
Figura 18 – Como ocorre o envio e recebimentos de dados na comunicação serial	32
Figura 19 – Pinagem do conector RS-232	32
Figura 20 – Leitura no protocolo SPI	34
Figura 21 – Escrita no protocolo SPI	34
Figura 22 – Diagrama de blocos da utilização do MAX 232	35
Figura 23 – Diagrama em blocos do projeto	36
Figura 24 – Exemplo de tela do Labview	37
Figura 25 – Tela do programa do Labview finalizado	38
Figura 26 – Diagrama dos temas abordados no tópico de eletrônica	39
Figura 27 – Circuito básico do microcontrolador PIC	40
Figura 28 – Foto do gravador de PIC EDUTECHKITS	40
Figura 29 – Exemplo de tela do <i>software</i> Winpic	41
Figura 30 – Exemplo de tela do <i>software</i> CCS	41
Figura 31 – Gravador PICSTART Plus	42
Figura 32 – Exemplo de tela do <i>software</i> Mplab	42
Figura 33 – Circuito utilizando o CD4053	43
Figura 34 – Funções atribuídas aos microcontroladores	44
Figura 35 – Circuitos eletrônicos que compõem a luva	45
Figura 36 – Fotos do circuito acelerômetro mão finalizado	45
Figura 37 – Fotos do circuito luva finalizado	45
Figura 38 – Circuito divisor de tensão	46
Figura 39 – Circuito amplificador de tensão	46
Figura 40 – Circuito integração PIC e acelerômetro	47
Figura 41 – Foto da fonte	48
Figura 42 – Fotos da placa fonte finalizada	48
Figura 43 – Foto do teclado luva finalizado	49
Figura 44 – Fotos do circuito do teclado luva finalizado	50
Figura 45 – Fotos do teclado controle finalizado	51
Figura 46 – Fotos do circuito teclado controle finalizado	51
Figura 47 – Dados do acelerômetro na aquisição com variação angular	53
Figura 48 – Gráfico da equação da aquisição de dados angulares	54
Figura 49 – Dados do acelerômetro no movimento retilíneo	55
Figura 50 – Movimentação do acelerômetro com ângulo de 45° sobre o eixo Y	55
Figura 51 – Dados do acelerômetro (em ângulo de 45° sobre o eixo Y) na aquisição do movimento retilíneo	56

Figura 52 – Movimentação do acelerômetro com ângulo de 45° sobre o eixo W	56
Figura 53 – Dados do acelerômetro (em ângulo de 45° sobre o eixo W) na aquisição do movimento retilíneo	57
Figura 54 – Foto da luva finalizada	58
Figura 55 – Foto de um usuário com a luva	59
Figura 56 – Fotos da fonte finalizada.....	60
Figura 57 – Foto do movimento pulso-giro.....	62
Figura 58 – Foto do movimento inclinação do braço	63
Figura 59 – Foto do movimento inclinação do pulso em relação ao antebraço.....	64

LISTA DE TABELAS

Tabela 1 – Tabela de polaridades e fases no protocolo SPI	33
Tabela 2 – Tabela de funções do teclado luva	49
Tabela 3 – Tabela de funções do teclado controle	51
Tabela 4 – Tabela comparando vantagens e desvantagens do método <i>Move Position</i> (MP)	61
Tabela 5 – Tabela comparando vantagens e desvantagens do método <i>Move Joint</i> (MJ)	61

LISTA DE SIGLAS

AGV	<i>Automatic Guided Vehicle</i>
ATX	<i>Advanced Technology Extended</i>
CI	<i>Circuito Integrado</i>
CLP	<i>Controlador Lógico Programável</i>
CPU	<i>Central Processing Unit</i>
CS	<i>Chip Select</i>
DAELN	<i>Departamento Acadêmico de Eletrônica</i>
DAQBI	<i>Departamento Acadêmico de Química e Biologia</i>
DCE	<i>Data Communication Equipment</i>
DTE	<i>Data Terminal Equipment</i>
EEPROM	<i>Electrically-erasable Programmable Read-only Memory</i>
IC	<i>Inter-integrated Circuit</i>
IHM	<i>Interfaces homem-máquina</i>
LabVIEW	<i>Laboratory Virtual Instruments Engineering Workbench</i>
LASIM	<i>Laboratório de Automação e Sistemas Inteligentes da Manufatura</i>
LCD	<i>Liquid Crystal Display</i>
LED	<i>Light-emitting Diode</i>
PIC	<i>Peripheral Interface Controller</i>
RAM	<i>Random Access Memory</i>
ROM	<i>Read Only Memory</i>
RS-232	<i>Recommended Standard-232</i>
SCLK	<i>Serial Clock</i>
SDI	<i>Serial Data Input</i>
SDO	<i>Serial Data Output</i>
SI	<i>Serial In</i>
SO	<i>Serial Out</i>
SPI	<i>Serial Peripheral Interface</i>
USART	<i>Universal Synchronous Asynchronous Receiver Transmitter</i>
USB	<i>Universal Serial Bus</i>
UTFPR	<i>Universidade Tecnológica Federal do Paraná</i>

SUMÁRIO

1	INTRODUÇÃO.....	10
1.1	PROBLEMA.....	12
1.2	OBJETIVOS.....	14
1.2.1	Objetivo geral.....	14
1.2.2	Objetivos específicos.....	14
1.3	JUSTIFICATIVA.....	15
2	FUNDAMENTAÇÃO TEÓRICA.....	17
2.1	SISTEMAS HÁPTICOS.....	17
2.2	ROBÔS INDUSTRIAIS MANIPULADORES.....	18
2.3	ROBÔ MITSUBISHI RV-M1.....	21
2.4	CINEMÁTICA INVERSA DO ROBÔ MITSUBISHI RV-M1.....	24
2.5	ACELERÔMETRO.....	26
2.6	ACELERÔMETRO MMA7455L.....	26
2.7	MICROCONTROLADOR.....	28
2.8	MICROCONTROLADOR PIC16F876A.....	29
2.9	COMUNICAÇÃO SERIAL RS-232.....	30
2.10	COMUNICAÇÃO SERIAL SPI (<i>SERIAL PERIPHERAL INTERFACE</i>).....	33
2.11	CIRCUITO INTEGRADO MAX 232.....	34
3	DESENVOLVIMENTO.....	36
3.1	LABVIEW.....	36
3.2	ELETRÔNICA.....	38
3.2.1	Microcontroladores.....	39
3.2.2	Luva.....	44
3.2.2.1	Circuitos eletrônicos que compõem a luva.....	44
3.2.2.2	Integração entre microcontrolador e acelerômetro.....	46
3.2.3	Módulo Fonte.....	48
3.2.3.1	Teclados.....	49
3.3	CONFECÇÃO DE PLACAS DE CIRCUITO IMPRESSO.....	52
3.4	ACELERÔMETRO.....	52
3.5	ESTRUTURA MECÂNICA.....	58
3.6	PROGRAMAÇÃO.....	60
3.6.1	Programação da luva.....	61
3.6.2	Programação do teclado controle.....	64
4	CONSIDERAÇÕES FINAIS.....	66
	REFERÊNCIAS.....	67
	APÊNDICE A – COMANDOS DO ROBÔ RV-M1.....	70
	APÊNDICE B – TABELA DE PINAGEM E DIREÇÃO DE FLUXO DE DADOS NA COMUNICAÇÃO RS-232.....	73
	APÊNDICE C – PROGRAMAÇÃO DO LABVIEW.....	74
	APÊNDICE D – CIRCUITO ESQUEMÁTICO DA PLACA ACELERÔMETRO MÃO.....	75
	APÊNDICE E – CIRCUITO ESQUEMÁTICO DA PLACA LUVA.....	76
	APÊNDICE F – CIRCUITO ESQUEMÁTICO DA PLACA FONTE.....	77
	APÊNDICE G – PINAGEM DOS CONECTORES.....	78
	APÊNDICE H – CIRCUITO ESQUEMÁTICO DA PLACA TECLADO LUVA.....	79
	APÊNDICE I – CIRCUITO ESQUEMÁTICO DA PLACA TECLADO CONTROLE.....	80
	APÊNDICE J – CONFIGURAÇÕES DO ACELERÔMETRO MMA7455L PARA UTILIZAÇÃO COM O MICROCONTROLADOR PIC18F252.....	81
	APÊNDICE K – FLUXOGRAMA DE PROGRAMAÇÃO DA LUVA.....	82
	APÊNDICE L – FLUXOGRAMA DE PROGRAMAÇÃO DO TECLADO CONTROLE.....	84

1 INTRODUÇÃO

A automação industrial está diretamente relacionada com a mecanização de processos e pode ser definida como a tecnologia em que um procedimento é concluído sem intervenção humana (GROOVER, 2008) e, de acordo com Rosário (2005), desde o início dos anos 60 os robôs estão sendo cada vez mais utilizados na indústria.

Automação de sistemas e robótica são duas áreas que estão relacionadas entre si e, em um contexto amplo, pode-se definir automação como uma tecnologia que utiliza certos recursos (mecânicos, computacionais e eletroeletrônicos) para operação e controle da produção (PAZOS, 2002).

De acordo com Pazos (2002), existem cinco razões para a utilização de robôs na produção industrial:

- a. Custo: o investimento inicial é alto, mas menor que o custo total para empregar funcionários (levando em conta que o robô pode trabalhar cerca de 98% da atividade considerada);
- b. Melhora a produtividade: geralmente robôs trabalham mais rápido que humanos e reduzem o desperdício de materiais;
- c. Melhoria da qualidade do produto: velocidades e precisão de posicionamento melhores do que o do trabalho humano;
- d. Capacidade de operar em ambientes hostis ou com materiais perigosos; e
- e. Melhora no gerenciamento da produção: a utilização de robôs facilita o processo de aquisição de dados já que estes podem ser adquiridos instantaneamente, via computador e processados com agilidade.

Algumas situações em que os manipuladores industriais podem substituir o trabalho humano são: situações em que se exige alta repetibilidade, cuidados extremos, situações perigosas, desgaste físico e mental (CARRARA, [19--?]) e ainda, de acordo com Groover (2008), onde houver necessidade de manipulação de peças de difícil manuseio, situações desconfortáveis e na substituição de operadores em mais de um turno (um único robô pode substituir mais de um operador).

De acordo com Laus (2006), na robótica estão presentes as contribuições das áreas de mecânica (estudo das máquinas nas situações estáticas e dinâmicas), matemática (desenvolvimento de ferramentas para descrever os movimentos do robô), teoria de controle

(ferramentas que avaliam os algoritmos adotados para realização de movimentos e aplicação de forças), elétrica (sensores e interfaces) e informática (programar os manipuladores para que desenvolvam a tarefa desejada).

Silveira (2002) classifica a área da robótica como “automação programável”. Esta denominação vem do fato de que os robôs possuem uma importante característica de adaptabilidade ao produto, ou seja, podem ser reprogramados de acordo com a necessidade da linha de produção diante de mudanças ocorridas na mesma.

Um dos tipos de robôs utilizados na indústria são os braços robóticos, que de acordo com, Braga (2009), devem conter os seguintes elementos: controle (que deve obter as informações de movimentação a partir de um teclado, *joystick* ou um computador), atuador (deve movimentar o braço e a garra), sensores (para transmitir informações ao operador) e fonte de alimentação.

Geralmente a programação dos robôs na indústria é feita por aprendizagem (ROBÓTICA INDUSTRIAL, 2002), ou seja, consiste no movimento individual de cada junta. Isto implica que não há necessidade de um conhecimento profundo do robô que está sendo utilizado, apenas é feito o armazenamento de uma sequência de posições intermediárias para determinar o ponto final desejado.

Os robôs industriais são sistemas mecânicos articulados controlados através de programas sequenciais pré-definidos que podem ser definidos e alterados utilizando-se de interfaces. Pazos (2002, p. 79), resume que: “uma interface define a forma como a informação é passada de um sistema para outro”.

Os robôs industriais podem possuir diversas interfaces de comunicação. Isto se exemplifica da seguinte forma (ROBÓTICA INDUSTRIAL, 2002):

- a. Interface homem-máquina: permite a operação e programação através dos operadores;
- b. Interface com sensores e atuadores: permite que os robôs atuem nos processos industriais de acordo com valores obtidos nos sensores; e
- c. Interface com controladores industriais: para que o robô possa interagir com os demais componentes dos sistemas de controle.

As Interfaces homem-máquina (IHM's) são equipamentos que “traduzem” os comandos do homem para a máquina. São exemplos de interfaces para o controle de robôs:

- a. Teclados e *joysticks*: controle através de botões e alavancas que realizam ações programadas dentro do controlador do robô;
- b. Computador: envia comandos para o robô através de programas de comunicação;
- c. Dispositivos que captam impulsos elétricos emitidos pelo cérebro: interpretam e transmitem para o robô;
- d. Processamento de imagens: uma câmera filma uma determinada ação que processa as imagens, identifica os movimentos e transmite para o robô; e
- e. Sistemas hápticos: sistemas mecânicos e eletrônicos capazes de identificar ações através do “toque”, geralmente com dispositivos acoplados ao corpo ou sensíveis ao toque dos operadores.

Com a finalidade de intensificar e tornar a interatividade homem-máquina cada vez mais natural e intuitiva, as pesquisas por sistemas hápticos vêm crescendo (CAETANO, 2008). De acordo com Zelek et al. (2003), háptico refere-se ao estudo do toque e das sensações da pele. Com a evolução da automação, esses sistemas estão cada vez mais complexos e tentam representar com perfeição os movimentos humanos. Caetano (2008) afirma que os sistemas hápticos têm sido utilizados em diversas áreas, como por exemplo, médica, militar, aero-espacial, robótica, submarina e exploração em lugares inóspitos.

1.1 Problema

Atualmente no Laboratório de Automação e Sistemas Inteligentes da Manufatura (LASIM) da Universidade Tecnológica Federal do Paraná (UTFPR) no campus de Curitiba – Paraná (PR) existe um braço robótico, cuja finalidade é manipular peças no sistema de manufatura existente. Nele existem dois tipos de interface homem-máquina (Figura 1). A primeira é um controle manual, onde são configuradas as posições do braço. O segundo tipo é por intermédio do computador utilizando-se um *software* para programar uma sequência de movimentos e em seguida o robô executá-la.

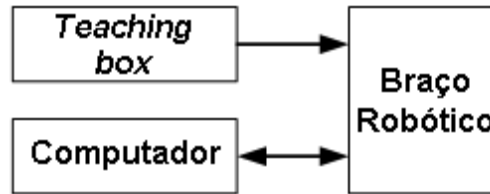


Figura 1 – Diagrama de blocos da situação atual do laboratório
Fonte: Autoria própria.

Há alguns problemas encontrados nessa parte do sistema de manufatura, principalmente com as interfaces atuais. O *teaching box* usado faz com que o trabalho de gravar posições precisas se torne difícil e demorado. A programação utilizada para gravar a sequência de posições do braço robótico com o *software* tem pouca funcionalidade. No sistema atual, é necessário digitar o código de programação, no computador, na ordem que se deseja enviar ao robô. Caso sejam digitadas, por exemplo, dez linhas de comandos e seja necessário adicionar uma linha entre o terceiro e o quarto comando, é preciso reescrever todos os comandos seguintes a partir daquele que deve ser alterado.

O que esse projeto propõe para a solução dos problemas citados é adicionar tanto um novo elemento de controle manual quanto uma interface com o computador utilizando o LabVIEW (*Laboratory Virtual Instruments Engineering Workbench*) e ainda um módulo microcontrolado para que a execução seja possível sem auxílio de computador. O novo elemento de controle sugerido é uma *haptic glove*, cujo objetivo é trazer mais agilidade para o usuário ao posicionar o braço robótico e salvar essas posições. A escolha do *software* LabVIEW foi baseada na integração dos outros componentes do sistema de manufatura. Essa escolha vai proporcionar um tempo menor de processamento, uma maior facilidade de alteração de uma linha de programa sem ter a necessidade de alterar todas as seguintes, salvar o programa e, por último, uma interface mais intuitiva para o usuário. Na figura 2 está a representação em diagrama de blocos da proposta de melhoria para o laboratório, como sugerida neste projeto.

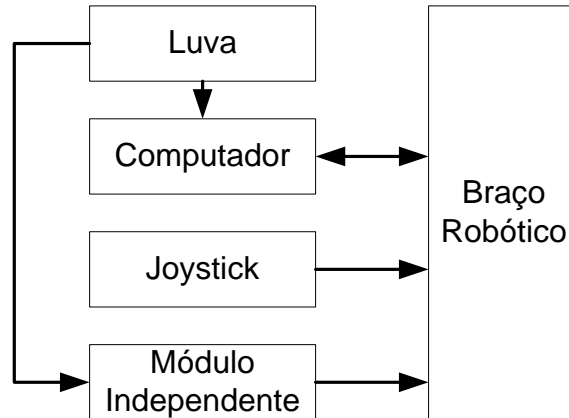


Figura 2 – Diagrama de blocos da sugestão de melhoria para o laboratório
Fonte: Autoria própria.

As soluções propostas estarão delimitadas por outros problemas do robô. São eles: o desgaste, a sensibilidade e a perda de referências do braço robótico. Outros fatores que influenciam no resultado do projeto proposto são: a aquisição de dados e a comunicação que o robô faz com seus periféricos, que nem sempre são concluídas com sucesso.

1.2 Objetivos

1.2.1 Objetivo geral

Implementar uma *haptic glove* para controle de um braço robótico e nova opção de interface de comunicação entre computador e braço robótico.

1.2.2 Objetivos específicos

- a. Estudar a comunicação do braço robótico com o computador;
- b. Elaborar um programa no *software* LabVIEW que cumpra as funções descritas na problemática do projeto;
- c. Estudar e testar os componentes que serão necessários para o desenvolvimento do *hardware* do elemento de controle;
- d. Fazer a parte mecânica da *haptic glove*;
- e. Elaborar a parte eletrônica da luva;
- f. Integrar a parte mecânica com a parte eletrônica da *haptic glove*;
- g. Comunicar o sistema háptico com o *software* LabVIEW;

- h. Elaborar os algoritmos de movimentação da parte robótica juntamente com o *software* LabVIEW;
- i. Integrar o elemento de controle com *software* LabVIEW e com o braço robótico;
- j. Implementar o *hardware* do módulo independente;
- k. Adaptar os algoritmos desenvolvidos para o módulo independente;
- l. Integrar a *haptic glove* com o módulo independente e com o braço robótico;
- m. Realizar testes do funcionamento da *haptic glove* com as duas interfaces de comunicação; e
- n. Finalizar a documentação do projeto de acordo com as normas da UTFPR.

1.3 Justificativa

Um sistema inteligente de manufatura envolve um alto nível de automação, e tem como objetivo melhorar a produção, resultando em uma maior produtividade e redução do preço do produto final. Os robôs manipuladores são utilizados largamente nesse ramo da indústria, para *setup* de ferramentas e posicionamento de peças.

Um modelo desse sistema é encontrado no LASIM da Universidade Tecnológica Federal do Paraná, onde os alunos aprendem a funcionalidade deste. No desenvolvimento da matéria de Projeto Integrador III foi usado este laboratório, e na utilização deste foram encontrados alguns problemas, como os descritos anteriormente (na seção 1.1). Para usar o modelo atual é necessário muito tempo, o que dificulta o acesso ao laboratório e a manipulação do sistema. Com a adição de um novo elemento de controle, é esperada a otimização do tempo gasto para as atividades descritas na problemática do projeto. Além da questão do tempo, a *haptic glove* proporciona um modo mais dinâmico de alunos e professores trabalharem com o braço manipulador. Esse problema foi encontrado na UTFPR, mas a solução dele pode ser aplicada em alguns ramos industriais, como por exemplo, as empresas que tenham um controle precário de braços robóticos.

O projeto, além de proporcionar novos desafios, integra as áreas estudadas durante o curso de Tecnologia em Mecatrônica Industrial, agrega novas tecnologias e ajudará os usuários do LASIM. Os conhecimentos utilizados da área de eletrônica estão na implementação do *hardware* do módulo independente de computador e aplicação dos acelerômetros. A disciplina de robótica é utilizada nos cálculos de posições do braço manipulador. A programação do *software* do módulo independente e do LabVIEW exigem conhecimentos da área de informática. Os conhecimentos adquiridos são nas três principais

áreas do curso: eletrônica, mecânica e programação. Entre esses conhecimentos estão o funcionamento e aplicação de acelerômetros, a programação em *software* LabVIEW e cálculos complexos de posicionamento de robôs.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Sistemas Hápticos

Um sistema háptico é aquele que interage com o usuário de maneira física (força, vibrações e/ou movimento). É composto de *hardware* e *software* que transmitem informações táteis que permitem ao usuário interagir com objetos a longa distância ou na realidade virtual através de um retorno sensorial. De acordo com Gomes [20--?], esse retorno sensorial pode ser de dois tipos: tátil (sensível ao toque) e cinestésico (sensível a movimentos). O retorno tátil proporciona ao usuário sentir temperatura e rugosidade. No ser humano a percepção dessas propriedades ocorre, principalmente, na região dos dedos e mãos. O retorno cinestésico utiliza-se de forças para representar movimentos e basicamente copia a função de ossos e músculos (GOMES, [20--?]). Estas forças podem ser aceleração, peso e compressão, por exemplo.

A evolução desses sistemas caminha juntamente com a evolução da automação. De acordo com Gomes ([20--?]), os primeiros sistemas operados a longa distância foram construídos em 1950, no laboratório de Argonne (Estados Unidos da América), pelo Dr. Raymond C. Goertz, para segurar remotamente substâncias radioativas. Atualmente essa tecnologia está presente em várias áreas: celulares, jogos, medicina e indústrias. Na figura 3 há vários exemplos de sistemas hápticos.

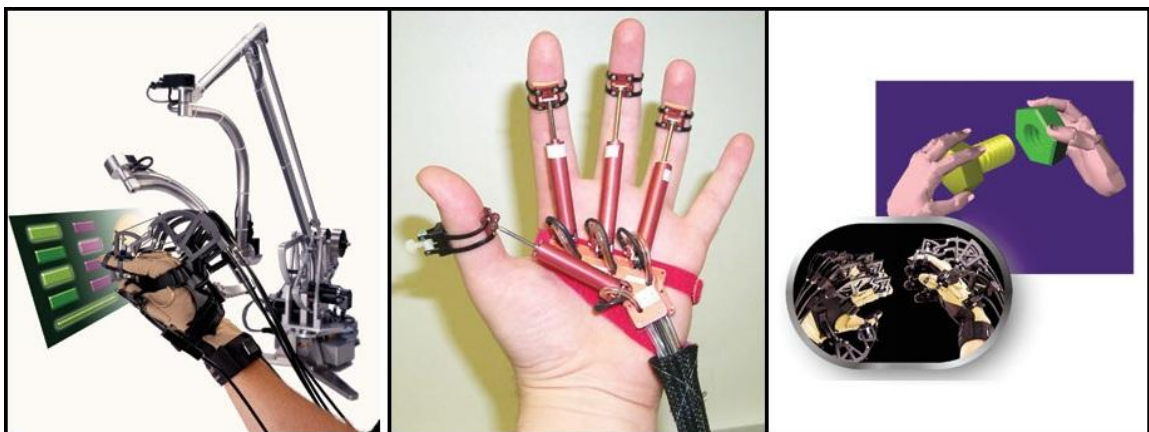


Figura 3 – Exemplos de sistemas hápticos
Fonte: Thilmany (2003).

Ao longo dos anos são desenvolvidas, cada vez mais, interfaces das quais o usuário pode se aproximar dos recursos computacionais de modo intuitivo utilizando aplicações como

se estivesse atuando no mundo real (KIRNER; SISCOOTTO, 2007). O laboratório de pesquisas avançadas da empresa *Hitachi* (situado em Hatoyama no Japão) “vem desenvolvendo uma interface cérebro-máquina que permite aos usuários ligar e desligar interruptores com a mente” (CAETANO, 2008, p. 8).

2.2 Robôs Industriais Manipuladores

Os robôs podem ser construídos de acordo com a aplicação específica que se é desejada. Com este princípio, Silveira (2002) classifica os robôs em quatro grupos:

- a. Manipuladores: aqueles que possuem articulações de encaixe em suas terminações. Estes são usados como integradores, transportadores ou posicionadores de objetos;
- b. Transporte: contam com sistemas de locomoção para sua movimentação. Um exemplo desta classificação são os veículos teleguiados chamados de AGV (*Automatic Guided Vehicle*);
- c. Medição: utilizam sensores de alta precisão, pois são destinados a fazer comparações em escalas muito pequenas; e
- d. Interventores: são destinados a executar tarefas que ofereçam alto risco aos operários ou em locais de acesso difícil.

De acordo com Silva *et al.* (2006), o tipo de robô mais utilizado é o manipulador que tem sua utilização “direcionada à execução de tarefas com certo grau de risco onde características como precisão, eficiência e repetibilidade são características imprescindíveis”.

Os braços robóticos são exemplo dos robôs classificados como manipuladores. Rosário (2005, p. 148), descreve que para a execução das tarefas requeridas pelo usuário, o braço robótico “move partes, objetos, ferramentas e dispositivos especiais segundo movimentos e pontos pré-programados”.

Os braços robóticos são basicamente compostos de juntas e elos (também chamados de vínculos) que podem ser observados na figura 4 (Carrara, [19--?]). Lopes (2002) complementa que uma das extremidades do robô fica fixa a uma base e a outra extremidade oposta dá suporte a um órgão terminal que pode movimentar-se livremente no espaço. Esta extremidade livre pode ser, por exemplo, uma garra ou uma ferramenta.

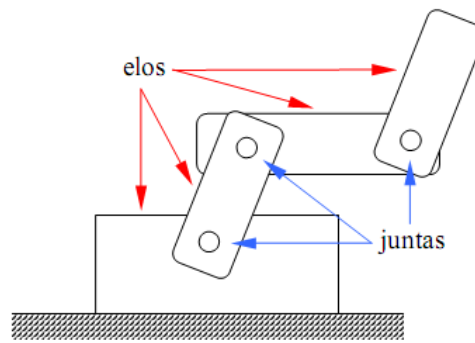


Figura 4 – Esquemático de juntas e elos em braços robóticos
Fonte: Carrara [19--?], página 14.

De acordo com Rosário (2005), existem três tipos de juntas encontradas nos manipuladores industriais:

- a. Deslizantes: permite o movimento linear entre dois vínculos (elos). Estas juntas são compostas de dois elos alinhados um dentro do outro, sendo que o elo interno escorrega pelo externo desenvolvendo um movimento linear;
- b. Rotativas: a conexão existente permite a rotação entre dois elos unidos por uma dobradiça comum dando a liberdade de uma poder se mover em movimento cadenciado em relação à outra; e
- c. Bola-e-encaixe: este tipo de conexão se comporta como se fosse a união de três juntas de rotação, o que permite movimentação em torno dos três eixos.

Na figura 5 podem ser observados os três tipos de juntas mencionadas anteriormente: deslizante (figura 5a), rotativa (figura 5b) e bola-e-encaixe (figura 5c). As setas ao lado de cada junta na, figura 5, indicam o sentido de movimentação das mesmas.

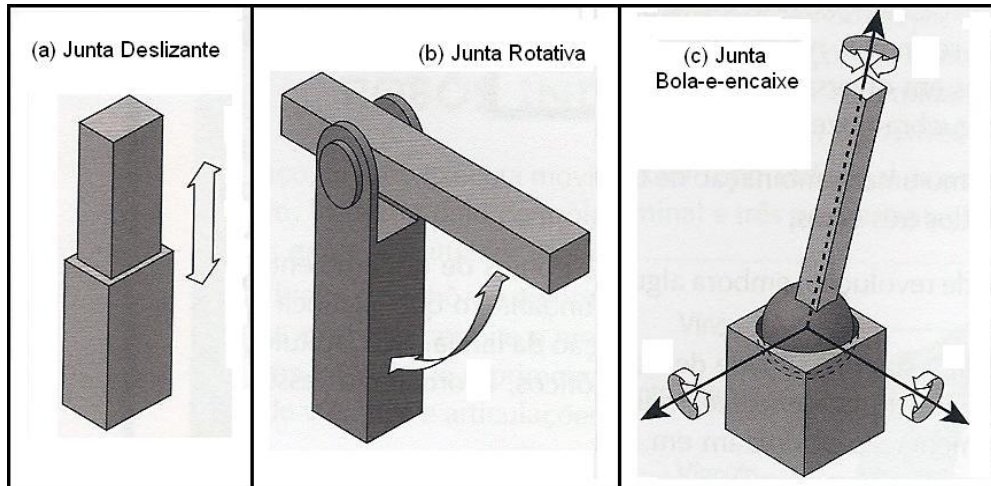


Figura 5 – Tipos de juntas e movimentação das juntas
Fonte: Rosário (2005), página 156.

A figura 6 apresenta dois exemplos de aplicações para as juntas apresentadas anteriormente: deslizante e rotativa, respectivamente. O exemplo para junta deslizante é um robô cartesiano e para a junta rotativa um braço robótico.

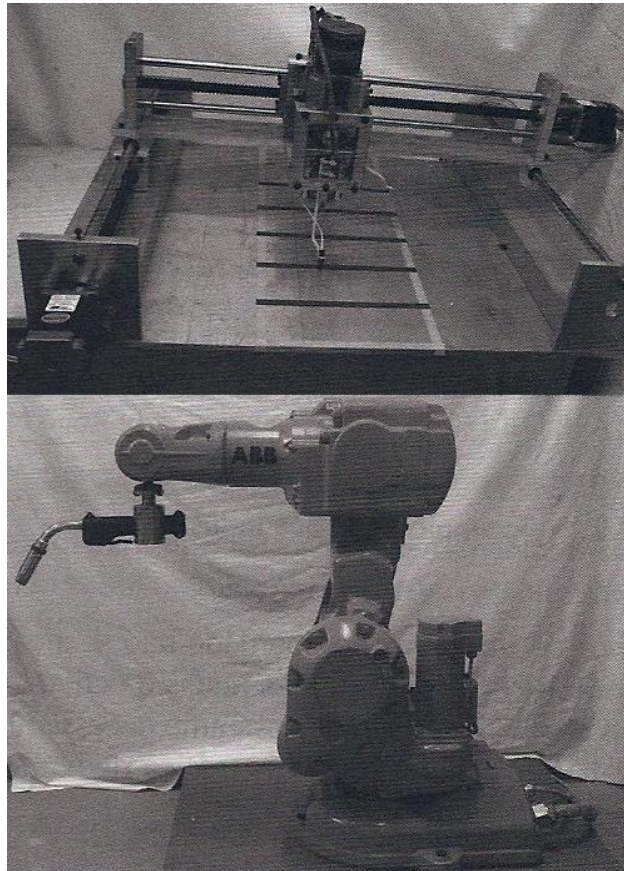


Figura 6 – Exemplo de robôs com junta deslizante e rotativa
Fonte: Rosário (2005), página 156.

Laus (2006) define que grau de liberdade de um manipulador é o número de variáveis de posição independentes que teriam que ser especificadas para localizar todas as partes do mecanismo. Nos braços robóticos o grau de liberdade é cada um dos movimentos que uma junta pode realizar de forma independente em relação à junta anterior (SILVA *et al.*, 2006).

Na figura 7 estão dois exemplos esquemáticos de braços robóticos compostos por duas juntas lineares. Pode-se observar que na parte A são possíveis dois movimentos distintos (em planos cartesianos diferentes), portanto este manipulador apresenta dois graus de liberdade. Na parte B as juntas tem movimentos independentes porem no mesmo plano cartesiano o que origina apenas um grau de liberdade.

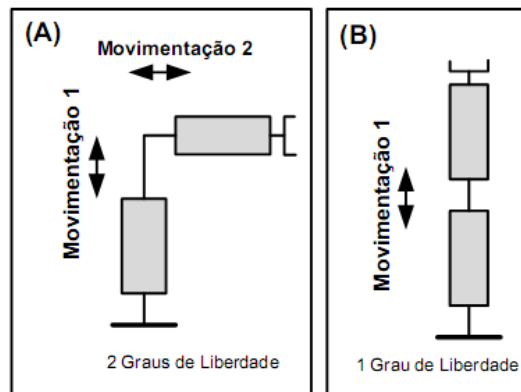


Figura 7 – Exemplificação de como se identificar grau de liberdade em juntas deslizantes
Fonte: Autoria própria.

Tipicamente um braço robótico possui seis graus de liberdade: três deles destinados ao posicionamento da ferramenta terminal e os outros três para orientação. Este número de graus de liberdade é utilizado quando o robô necessita alcançar todos os pontos do ambiente de trabalho e, no caso específico dos braços robóticos, o número de juntas geralmente é igual ao número de graus de liberdade (ROSÁRIO, 2005).

2.3 Robô Mitsubishi RV-M1

O robô RV-M1 é um braço robótico com cinco graus de liberdade (eixos rotacionais) produzido pela empresa Mitsubishi. Os eixos podem ser denominados cintura, ombro, cotovelo, ângulo de afastamento e ângulo de rolamento, como mostrados na figura 8.

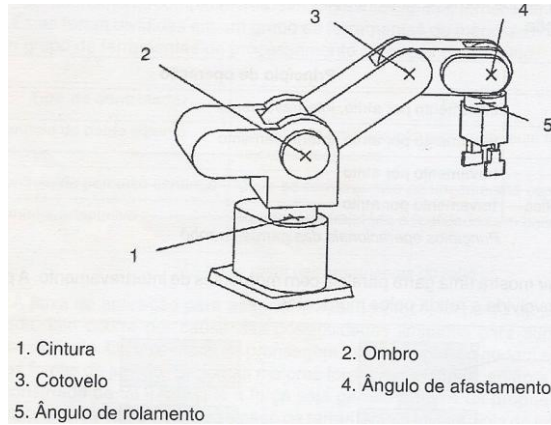


Figura 8 – Graus de liberdade do robô Mitsubishi RV-M1
Fonte: Dahlhoff (1993), página 54.

Cada um dos cinco eixos deste braço robótico pode ser associado a um determinado movimento do corpo humano, conforme indicado na figura 9:

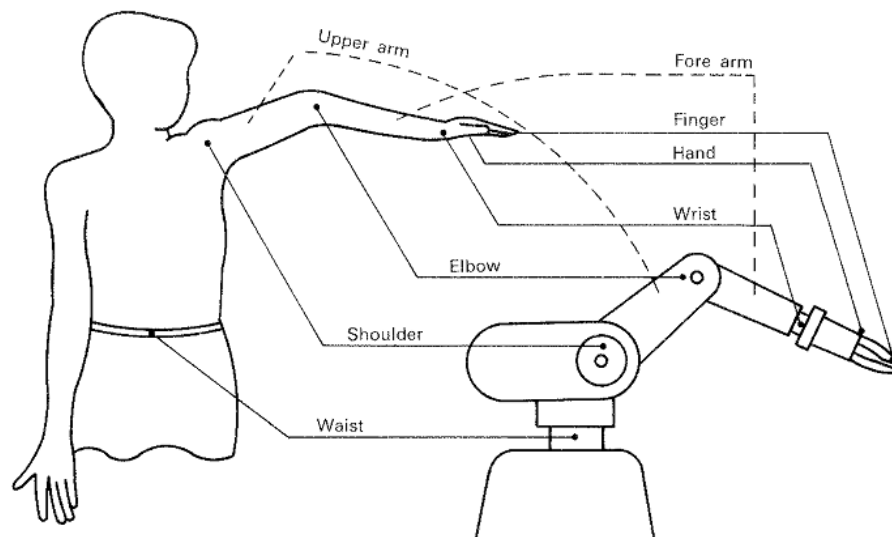


Figura 9 – Associação do braço robótico com o braço humano
Fonte: Mitsubishi [19--?], página 1-21.

As seguir são apresentadas algumas informações sobre este manipulador (DAHLHOFF, 1993, p. 55):

- a. Velocidade: 1 mm/s em relação à peça de trabalho;
- b. Carga nominal: 1,2 kg;
- c. Carga útil: 0,6 kg; e
- d. Precisão de repetição: 0,3 mm.

De acordo com o fabricante (ROBOTICS, 2008), este robô disponibiliza uma fácil programação, confiabilidade, boa velocidade e alto grau de repetibilidade, podendo ser utilizado para pegar e movimentar componentes, por exemplo. A sua programação pode ser

realizada de maneira simples através de posições escolhidas pelo usuário (por um controle manual, pelo computador ou até mesmo um controlador lógico programável – CLP).

A programação do robô pode ser feita de duas formas: por um controle (*teaching box*), como pode ser visualizado na figura 10, ou pelo computador através da porta de comunicação serial RS-232 (*Recommended Standard-232*) (MITSUBISHI, [19--?]).

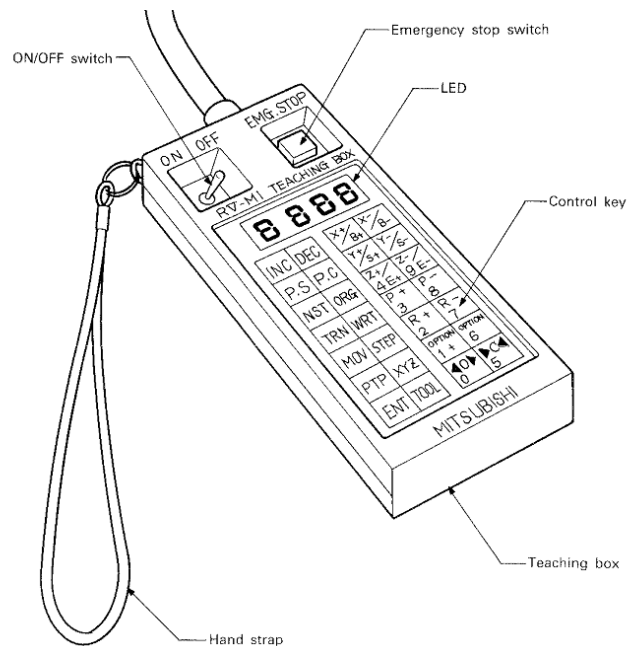


Figura 10 – Teaching box
Fonte: Mitsubishi [19--?], página 1-14.

O fabricante explica que através do *teaching box* é possível incrementar ou decrementar as posições das juntas do robô e assim salvar posições específicas. Após salvar as posições intermediárias o usuário, através do computador, pode editar a sequência que o robô deverá realizar. Para este braço robótico podem ser encontrados comandos de programação específicos no seu manual que, assim que transferidos ao robô, serão interpretados de forma adequada e automática (MITSUBISHI, [19--?]), por exemplo:

- a. **MO 1:** O robô automaticamente se move para a posição 1 (predeterminada pelo usuário com o auxílio do *teaching box*);
- b. **SP 2:** Muda a velocidade de movimentação do robô para 2 (o robô apresenta 7 velocidades de movimentação); e
- c. **GC:** Iniciais das palavras inglesas *grip close* que pode ser interpretado como “fechar garra”.

No apêndice A encontra-se uma tabela adaptada do manual do robô com os comandos mais utilizados.

O braço robótico está ligado a uma caixa que apresenta algumas indicações luminosas importantes (figura 11) como, por exemplo, se o robô está ligado, se foram detectados erros e se o equipamento encontra-se executando comandos (itens indicados, respectivamente, pelos números 1, 3 e 4 na figura 11). Nesta caixa também existem os botões de emergência (aciona a parada imediata do braço robótico), o botão de início, parada, e *reset* (apaga comandos). Estes botões são indicados, respectivamente, pelos números 2, 5, 6 e 7 na figura 11.

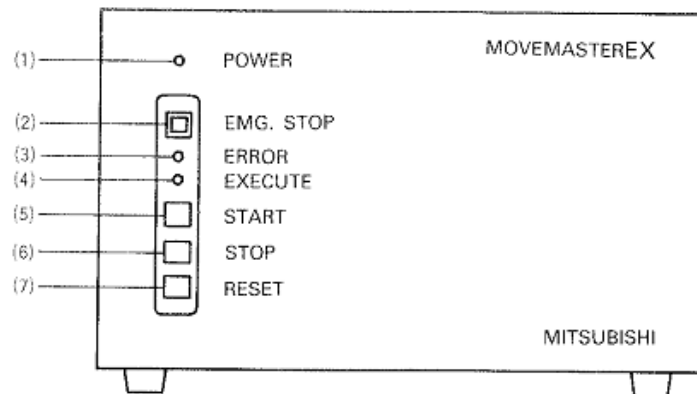


Figura 11 – Caixa indicativa
Fonte: Mitsubishi [19--?], página 2-13.

2.4 Cinemática Inversa do Robô Mitsubishi RV-M1

A cinemática trabalha com aspectos de movimentação, desprezando suas causas, considerando parâmetros como posição, deslocamento, velocidade, aceleração e tempo (SANDOR, 1983).

De acordo com Niku (2001), a cinemática pode ser abordada de duas maneiras: direta ou inversa. A cinemática direta permite que seja determinada a posição do manipulador quando todas as posições das juntas do robô são conhecidas. Com a cinemática inversa é possível determinar as posições das juntas do robô sendo conhecidas as coordenadas cartesianas do manipulador.

De acordo com Carrara (19--?, p. 41), “independentemente da geometria do manipulador, a solução da cinemática requer conhecimento de geometria, trigonometria e cálculo vetorial”. Este autor ainda destaca que cada braço robótico deve ser analisado individualmente, pois não há uma regra geral para se determinar as equações cinemáticas.

Laus (2006) destaca que a cinemática inversa, além de produzir as equações das variáveis de junta, é usada para se obter um algoritmo que permita calcular de forma computacional as variáveis de junta respeitando os limites físicos do robô.

Para se determinar o algoritmo desejado é necessário considerar a cinemática direta, a multiplicidade de soluções e a influência que uma junta causa nas outras (LAUS, 2006).

Foram estudadas as seguintes equações, elaboradas por Laus (2006), para serem utilizadas especificamente com o robô Mitsubishi RV-M1:

$$\theta_1 = \arctan 2(p_x, p_y) \quad (1)$$

$$\theta_3 = -\arccos(((p_z - d_1 - d_6 * \sin 234)^2 + (p_x / \sin 1 - d_6 * \cos 234)^2 - a_3^2 - a_2^2) / (2 * a_2 * a_3)) \quad (2)$$

$$\theta_2 = \arctan 2((p_z - d_1 - d_6 * \sin 234) * (a_2 + a_3 * \cos 3) - (p_x / \sin 1 - d_6 * \cos 234) * (a_3 * \sin 3), \quad (3)$$

$$(p_x / \sin 1 - d_6 * \cos 234) * (a_2 + a_3 * \cos 3) + (p_z - d_1 - d_6 * \sin 234) * (a_3 * \sin 3))$$

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad (4)$$

As deduções das equações apresentadas podem ser encontradas em Laus (2006).

Os ângulos θ (de 1 a 4) estão associados aos graus de liberdade mostrados na figura 8. As variáveis p_x , p_y e p_z são as coordenadas cartesianas do manipulador e θ_{234} é o ângulo do manipulador em relação ao plano de referência (conforme indicado na figura 12). As constantes a_2 , a_3 , d_1 e d_6 são distâncias fornecidas por Mitsubishi (19--?), conforme a figura 12. Existe ainda um ângulo θ_5 que indica a rotação do manipulador e não é necessário o uso de fórmulas para obtê-lo.

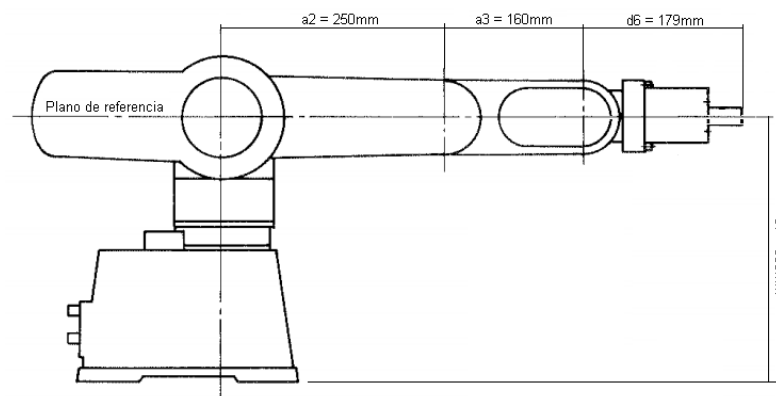


Figura 12 – Distâncias físicas do robô
Fonte: Mitsubishi [19--?], página 1-22.

2.5 Acelerômetro

O acelerômetro tem a finalidade de medir a aceleração de uma peça ou sistema (PAZOS, 2002). Este dispositivo consiste em um objeto com um grau de liberdade em relação ao seu eixo. Quando esse objeto sofre uma aceleração, por efeito de inércia, tende a manter sua posição e conseqüentemente desloca-se em relação ao seu eixo. Dependendo do grau de aceleração, a magnitude do deslocamento é maior ou menor, podendo assim, através da medição deste deslocamento, ser obtido o grau de aceleração sofrido.

Os acelerômetros podem medir a aceleração de 1 até 3 eixos distintos (X, Y e Z) e transmitir um sinal de saída equivalente que será utilizado na aplicação desejada. Quando o acelerômetro está na posição de repouso referente a um eixo, o sinal obtido é equivalente à metade do valor máximo (Figura 13 a). Quando ocorre movimentação em uma direção (de P1 para P2), o sinal aumenta proporcionalmente à aceleração (Figura 13 b) e se ocorrer movimentação em direção contrária (de P3 para P4), o sinal diminui (Figura 13 c) variando entre o valor máximo e mínimo da porta de saída. Esses valores de saída são enviados para um microcontrolador que processa este sinal de acordo com a programação feita.

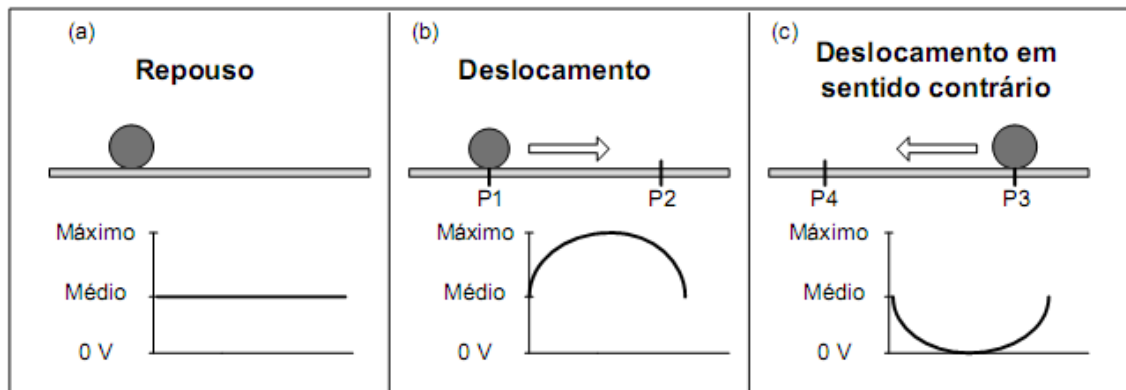


Figura 13 – Relação deslocamento/sinal de saída do acelerômetro
Fonte: Autoria própria.

2.6 Acelerômetro MMA7455L

O acelerômetro de modelo MMA7455L é projetado e produzido pela empresa americana Freescale Semiconductor. A seguir estão listadas algumas das características deste acelerômetro (FREESCALE SEMICONDUCTOR, 2007):

- a. Saída Digital (utilização de protocolo I²C – *Inter-integrated Circuit* e SPI – *Serial Peripheral Interface*);
- b. Baixo consumo de corrente (400 mA);
- c. Auto teste para o eixo Z;
- d. Baixa tensão de operação (entre 2,4V e 3,6V);
- e. Sensibilidade selecionável (três opções);
- f. Estrutura robusta resistente a choques;
- g. Baixo custo;
- h. Capacidade de reconhecimento (no nível do sinal de saída) de choques mecânicos e vibrações; e

Este dispositivo pode ser utilizado, por exemplo, em telefones celulares, detecção de queda-livre, anti-roubo em computadores portáteis, jogos e detectores de posição.

A imagem a seguir (figura 14) é a foto do acelerômetro, nas vistas inferior e superior, respectivamente.

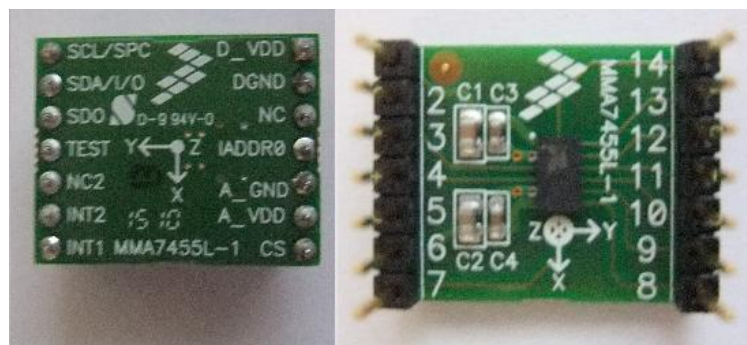


Figura 14 – Vistas inferior e superior do acelerômetro de modelo MMA7455L
Fonte: Autoria própria.

Este dispositivo consiste em um encapsulamento microusinado com células de sensoriamento capacitivo. Estas células são estruturas mecânicas formadas por materiais semicondutores. O funcionamento destas células pode ser explicado da seguinte forma: existe um conjunto de vigas ligadas a uma massa central móvel que se movimenta entre vigas fixas. Quando esta massa móvel se desvia da posição de repouso, a distância entre a massa e as vigas muda e com isto é possível obter a aceleração do sistema (FREESCALE SEMICONDUCTOR, 2007). Esta explicação do funcionamento é complementada pelo desenho esquemático da figura 13.

2.7 Microcontrolador

Souza (2007) define microcontrolador como um pequeno dispositivo eletrônico dotado de inteligência programável composto de um processador, memória, pinos de entradas e saídas e outros componentes auxiliares. Os microcontroladores são utilizados para fazer controle de processos, ou seja, controle de periféricos ligados a ele, como por exemplo, *displays*, botões, sensores, motores, resistências e relés (SOUZA, 2007).

De acordo com Pereira (2011), os microcontroladores possuem a vantagem de possuir em um único encapsulamento a CPU (*Central Processing Unit*), memórias (RAM – *Random Access Memory* e ROM – *Read Only Memory*) e circuitos de interface. Em sistemas onde não é utilizado o microcontrolador, há implementação separada dos elementos mencionados (CPU, RAM e ROM) e a integração deles pode ser trabalhosa.

A seguir (figura 15), está a representação em blocos da funcionalidade geral dos microcontroladores em aplicações mecatrônicas, baseada na explicação da empresa Microchip, fabricante de microcontroladores. A figura 15 mostra os cinco componentes presentes nos circuitos que utilizam microcontroladores: entrada, preparação do sinal de entrada, controle, preparação do sinal de saída e saída. Ao lado de cada um destes componentes estão alguns exemplos usuais a cada uma das partes destacadas.

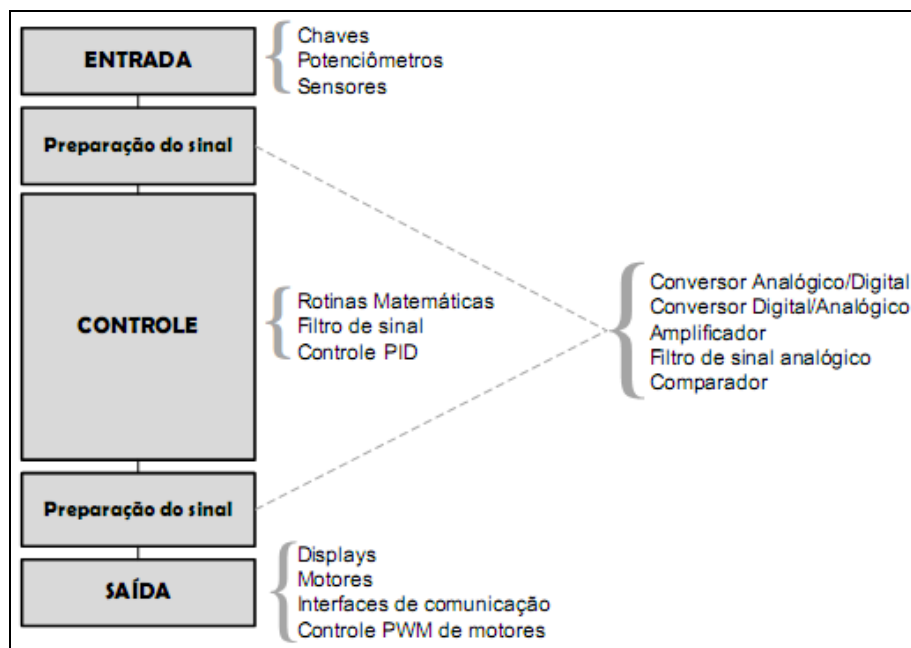


Figura 15 – Funcionalidade geral dos microcontroladores
Fonte: Autoria própria.

Utilizando como referência as entradas do microcontrolador e um programa interno, que contém a lógica de operação do processo, é feito o acionamento ou desacionamento das suas saídas. Toda vez que o microcontrolador for iniciado, o programa que está salvo dentro da memória deste será executado.

2.8 Microcontrolador PIC16F876A

O microcontrolador de modelo PIC16F876a (*Peripheral Interface Controller*) é fabricado pela empresa Microchip. A empresa fabrica três famílias de microcontroladores denominadas: 8-bit, 16-bit e 32-bit. As famílias são distribuídas de acordo com suas funcionalidades e desempenho, como pode ser observado na figura 16 (MICROCHIP, 2011). Todas as famílias apresentam em comum a plataforma de programação MPLab.

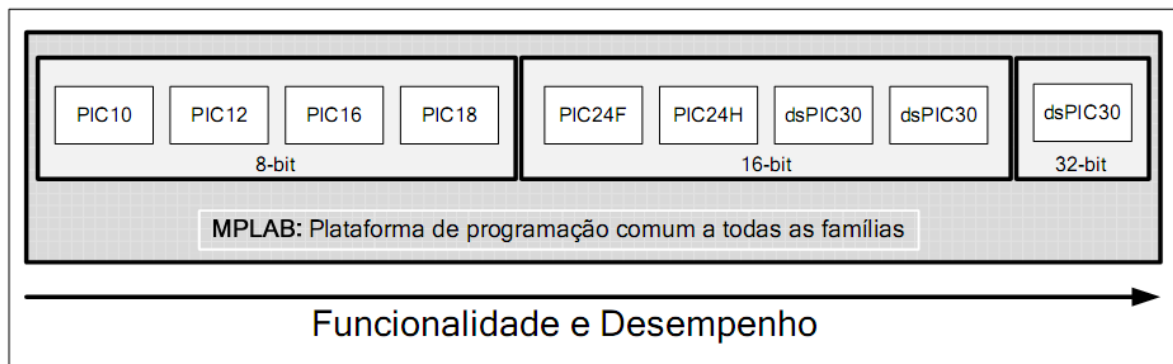


Figura 16 – Classificação dos microcontroladores PIC
Fonte: Autoria própria.

O microcontrolador de modelo PIC16F876a apresenta as seguintes características (MICROCHIP TECHNOLOGY, 2003):

- a. 3 *Timers*: dois *timers* de 8 bits e um *timer* de 16 bits;
- b. Controle para motores;
- c. Permite comunicação serial;
- d. USART (*Universal Synchronous Asynchronous Receiver Transmitter*);
- e. Permite comunicação paralela;
- f. Conversores de sinal: analógico para digital e digital para analógico;
- g. Alimentação simples de 5V suportando uma ampla faixa de tensão operacional (2.0V a 5.5V);

- h. Permite programação de código de proteção;
- i. Economia de energia automática quando não está sendo utilizado;
- j. Opções de osciladores;
- k. Baixo consumo de energia;
- l. Memória *flash* de alta velocidade;
- m. Memória de programa (*FLASH*) de 8Kb (*words*) de 14 bits;
- n. Cada instrução é codificada em uma *word* de 14 bits;
- o. Memória de dados RAM de 368 bytes;

Memória de dados EEPROM (*Electrically-erasable Programmable Read-only Memory*) de 256 bytes;

O MPLab é um programa fornecido pela empresa Microchip que apresenta diversas funções, num mesmo ambiente, para se trabalhar com qualquer das famílias do PIC. As funções desse programa são: gerenciamento de projetos, compilação, simulação, emulação e a gravação do chip (SOUZA, 2007).

2.9 Comunicação Serial RS-232

De acordo com Soares (2008), a porta de comunicação serial (RS-232) está presente na comunicação de vários tipos de máquinas e seus periféricos e este tipo de comunicação permanecerá nas áreas de automação industrial pelas próximas décadas.

A RS-232 permite comunicação bidirecional entre dois componentes, onde o dado é transmitido em forma de um byte, bit por bit ou em uma sequência pré-determinada pelo programador e chega ao receptor que interpreta e processa os dados recebidos (NICOLSI, 2002).

Para que a comunicação na forma serial seja padronizada, nas diversas aplicações que são possíveis, foram estipuladas algumas normas, citadas a seguir (SOARES, 2008):

- a. Existem apenas dois níveis lógicos possíveis: “0” lógico e “1” lógico (sem níveis intermediários);
- b. As tensões são sempre em corrente contínua;
- c. A tensão correspondente ao nível lógico “0” deve estar compreendida entre -3V e -15V;

- d. A tensão correspondente ao nível lógico “1” deve estar compreendida entre +3V e +15V;
- e. Os níveis de tensão entre -3V e +3V devem ser evitados, pois são níveis de tensão considerados indefinidos;

Na figura 17 pode ser observada a representação da transmissão do byte 10000010 em níveis lógicos “1” e “0” associados, respectivamente, às tensões de +15V e -15V. Na figura 17 também estão presentes o bit início e o bit final, que são importantes para que os receptores das informações transmitidas serialmente possam interpretar onde começa e termina o byte.

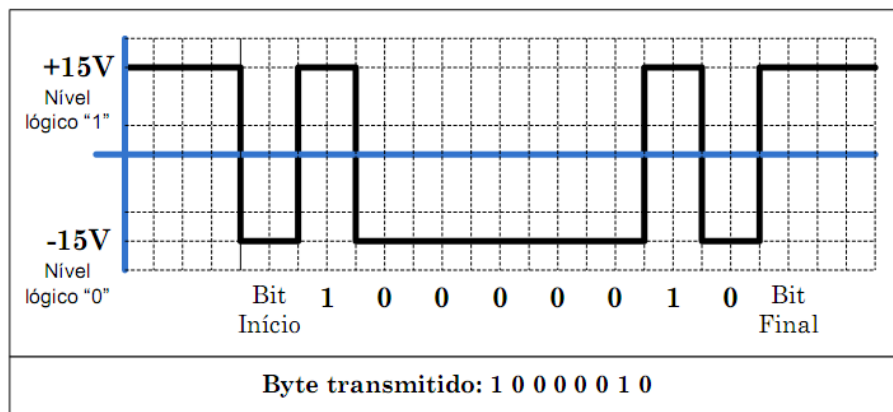


Figura 17 – Exemplo de transmissão de dados pela porta serial
Fonte: Autoria própria.

De acordo com Soares (2008), a porta de comunicação serial RS-232, transforma dados paralelos em seriais em sua transmissão e faz o processo contrário na recepção. Pode ser observado na figura 18 que um byte (dado em forma paralela) é transmitido para a porta serial transmissora (chamada de DTE – *Data Terminal Equipment*), onde é transformado em uma sequência de bits (dado em forma serial). Esta sequência de bits é enviada à porta serial do equipamento receptor (chamada de DCE – *Data Communications Equipment*) que “monta” novamente o byte de origem.

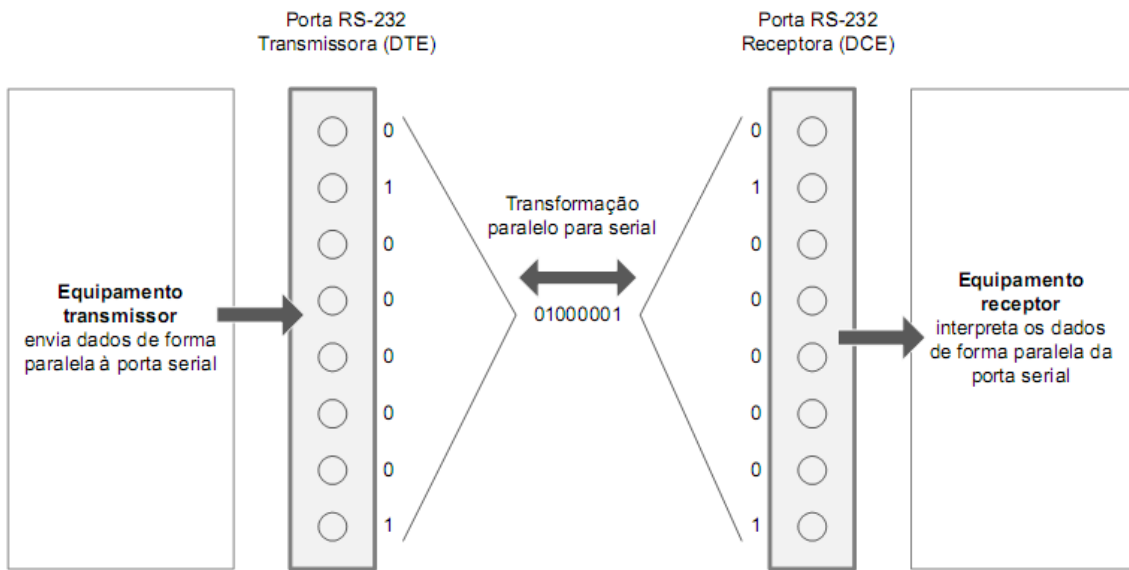


Figura 18 – Como ocorre o envio e recebimentos de dados na comunicação serial
Fonte: Aatoria própria.

Na figura 19 está o desenho das duas portas RS-232 existentes. Elas apresentam diferentes tamanhos (mais pinos ou menos pinos) que funcionam da mesma forma podendo ser escolhidas de acordo com a aplicação desejada.

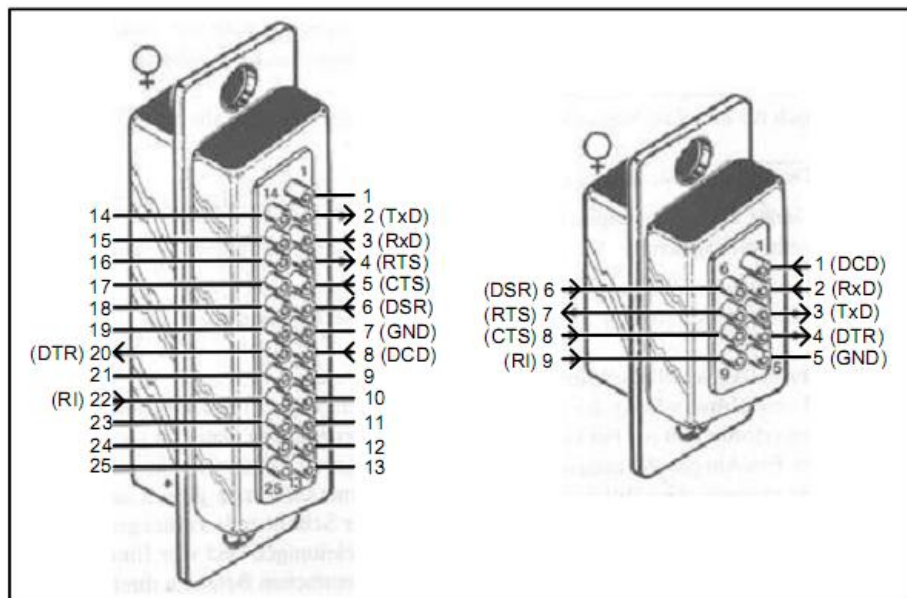


Figura 19 – Pinagem do conector RS-232
Fonte: Filho (2011), página 1.

No apêndice B encontra-se uma tabela com o significado da pinagem utilizada na figura 19, tradução do mesmo e direção do fluxo de dados.

2.10 Comunicação Serial SPI (*Serial Peripheral Interface*)

O SPI é um protocolo de comunicação síncrono (ou seja, quando um componente envia dados o outro recebe) composto de quatro sinais que não permite endereçamento, ou seja, a comunicação é feita diretamente entre dois pontos (SIEMEINTCOSKI; WACKERHAGEN, 2008).

De acordo com Pereira (2003, p. 268), é um protocolo de simples implementação já que “não existe facilidade de software para convivência de múltiplos dispositivos simultaneamente no mesmo barramento” e é composto fisicamente por quatro linhas de sinais:

- a. Seleção de dispositivo (CS – *chip select*): utilizada para selecionar qual dispositivo que deseja se estabelecer comunicação. É o elemento físico que prevê a utilização de múltiplos dispositivos no sistema;
- b. *Clock* (SCLK): utilizada para estabelecer sincronização entre os dispositivos mestres e escravos. O dispositivo que gera o sinal de *clock* é o mestre e o que recebe é o escravo;
- c. Entrada Serial (SI – *serial in*): recepção de dados; e
- d. Saída Serial (SO – *serial out*): saída de dados.

Existem quatro modos de operação; cada modo apresenta polaridade (CPOL) e fase (CPHA) de *clock* distintas, conforme a tabela 1 (PEREIRA, 2003):

Tabela 1 – Tabela de polaridades e fases no protocolo SPI

MODO	CPOL	CPHA	DESCRIÇÃO
0	0	0	O dado é armazenado na borda de subida do <i>clock</i>
1	0	1	O dado é armazenado na borda de descida do <i>clock</i>
2	1	0	O dado é armazenado na borda de descida do <i>clock</i>
3	1	1	O dado é armazenado na borda de subida do <i>clock</i>

Fonte: Autoria própria.

Pereira (2003) ressalta que todas as operações executadas que utilizam esse protocolo são precedidas de oito bits (que descrevem a natureza da operação) e observa que é importante verificar em cada dispositivo utilizado as especificações do fabricante para este tipo de comunicação, já que não há um padrão muito rígido e universal.

Os comandos de leitura e escrita são descritos a seguir Pereira (2003):

- Leitura: é precedida por um comando de leitura (*READ* = 0000 0011) seguido pelos 16 bits do endereço de memória. Após o último bit de endereço o dado passa a ser transmitido serialmente e ao término do dado trabalha-se com o valor de CS para aguardar nova leitura (figura 20).
- Escrita: é precedida por um comando de escrita (*WRITE* = 0000 0010) seguido pelos 16 bits do endereço de memória onde será escrito o dado. Após o último bit de endereçamento são enviados pelo dispositivo mestre os oito bits do dado a ser escrito (figura 21).

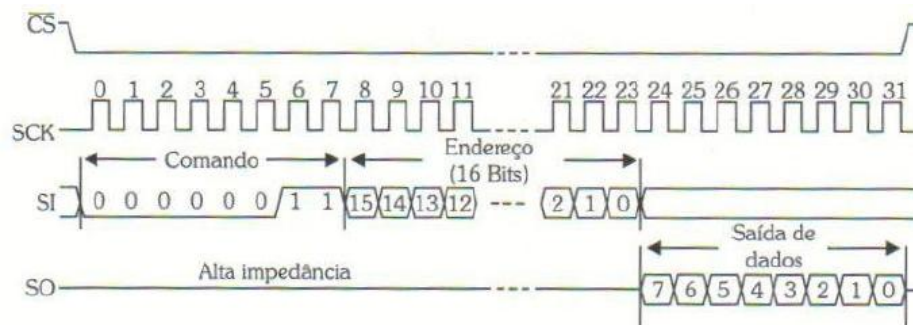


Figura 20 – Leitura no protocolo SPI
Fonte: Pereira (2003), página 270.

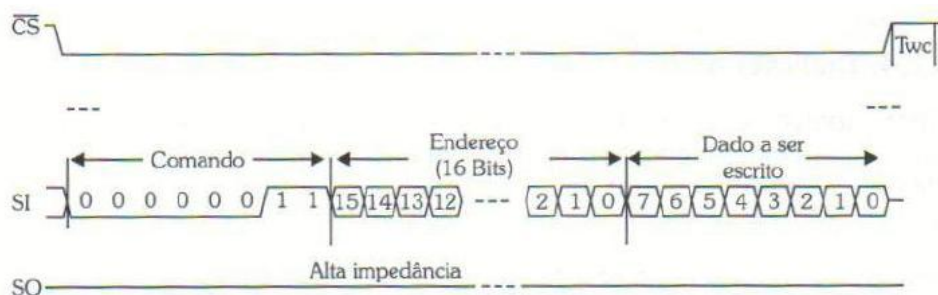


Figura 21 – Escrita no protocolo SPI
Fonte: Pereira (2003), página 271.

2.11 Circuito Integrado MAX 232

O circuito integrado (CI) MAX 232 é um transmissor/receptor que desempenha o papel de conversor de nível em circuitos que utilizam a comunicação serial RS-232 e microcontroladores. Este CI tem a capacidade de gerar sinais de $\pm 12V$ (necessários na comunicação RS-232) apenas com os 5V que são fornecidos a ele. A utilização do MAX 232

facilita o projeto dos circuitos já que não há necessidade de criar um circuito secundário com fonte de alimentação de ± 12 V para converter o sinal mandado pelo microcontrolador de 5V (TEXAS INSTRUMENTS, 2002).

Na figura 22 está um diagrama de blocos que representa o papel desempenhado pelo MAX 232 em um circuito utilizando comunicação serial e microcontrolador.

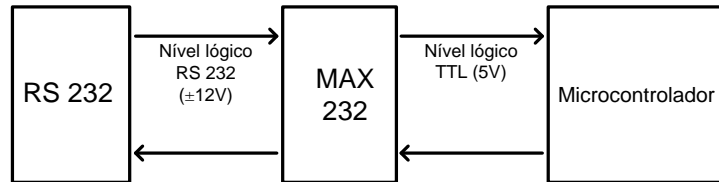


Figura 22 – Diagrama de blocos da utilização do MAX 232

Fonte: Autoria própria.

3 DESENVOLVIMENTO

Neste capítulo são detalhados todos os procedimentos e etapas realizados para a execução e conclusão deste projeto. Na figura 23 está o diagrama em blocos do projeto.

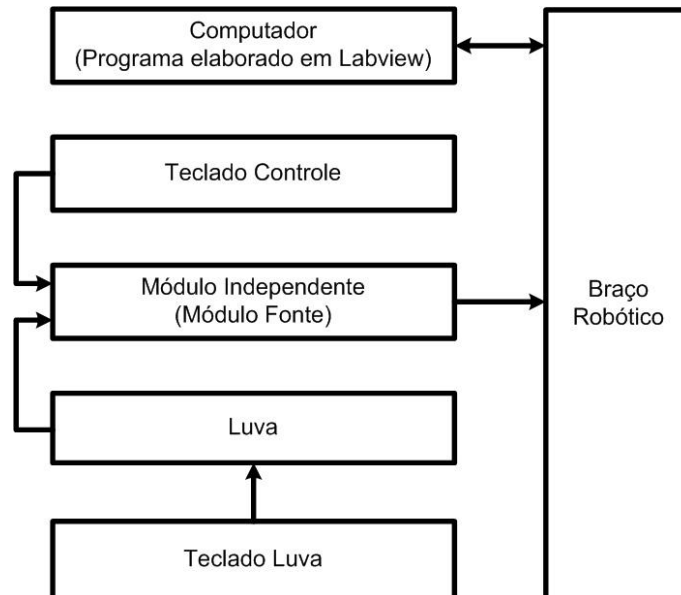


Figura 23 – Diagrama em blocos do projeto
Fonte: Autoria própria.

3.1 Labview

O Labview é um ambiente de desenvolvimento gráfico que utiliza ícones para programação ao invés de linhas de código. Na figura 24, está um exemplo de tela do Labview.

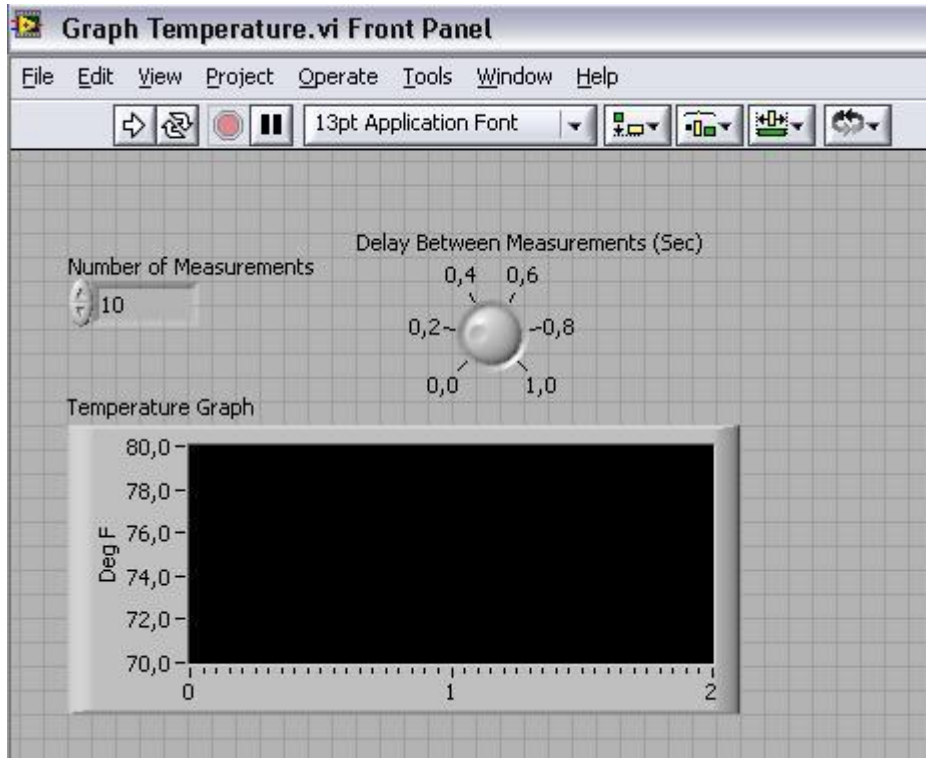


Figura 24 – Exemplo de tela do Labview
Fonte: Autoria própria.

Primeiramente foi estudada a programação da comunicação serial dentro do Labview. Foi alterado um exemplo genérico fornecido pelo próprio *software*.

O primeiro teste foi enviar, via comunicação serial, linhas de programa para o robô. Esta etapa não apresentou dificuldades, pois apenas enviar os caracteres (sem nenhum tipo de conversão) já é o suficiente para que o robô interprete o comando.

Após o teste de comunicação foi incluída a possibilidade de escrever o programa que será enviado ao robô em um arquivo de bloco de notas. Dentro do Labview foi colocado um comando que abre uma janela onde o usuário seleciona o arquivo de bloco de notas já criado. Após isto o *software* mostra na tela do Labview o código contido no arquivo selecionado e que quando solicitado pelo usuário, é enviado ao braço robótico.

O primeiro teste não foi totalmente eficaz, pois apesar do envio das linhas de programa ao braço robótico funcionar corretamente, a janela de solicitação do endereço do arquivo do bloco de notas abria mesmo quando não era solicitada. Depois de solucionados alguns problemas na programação a situação foi resolvida.

A tela do programa final (figura 25) possui os seguintes elementos, indicados pelas letras de A até I: (A) botão para escolha entre envio do programa de extensão “txt” ou comando digitado; (B) caminho de um arquivo com extensão “txt”; (C) janela do programa carregado indicado no caminho; (D) janela de escrita para envio de comando ao robô; (E)

janela de leitura do último comando enviado ao robô; (F) botão de envio de comando ou programa ao robô; (G) janela de leitura da serial do robô; (H) quantidade de bits recebidos pela serial; (I) botão de parada do programa. Para iniciar o programa, é necessário abrir o arquivo executável Labview_ROBOT.exe. A tela de programação do Labview encontra-se no apêndice C.

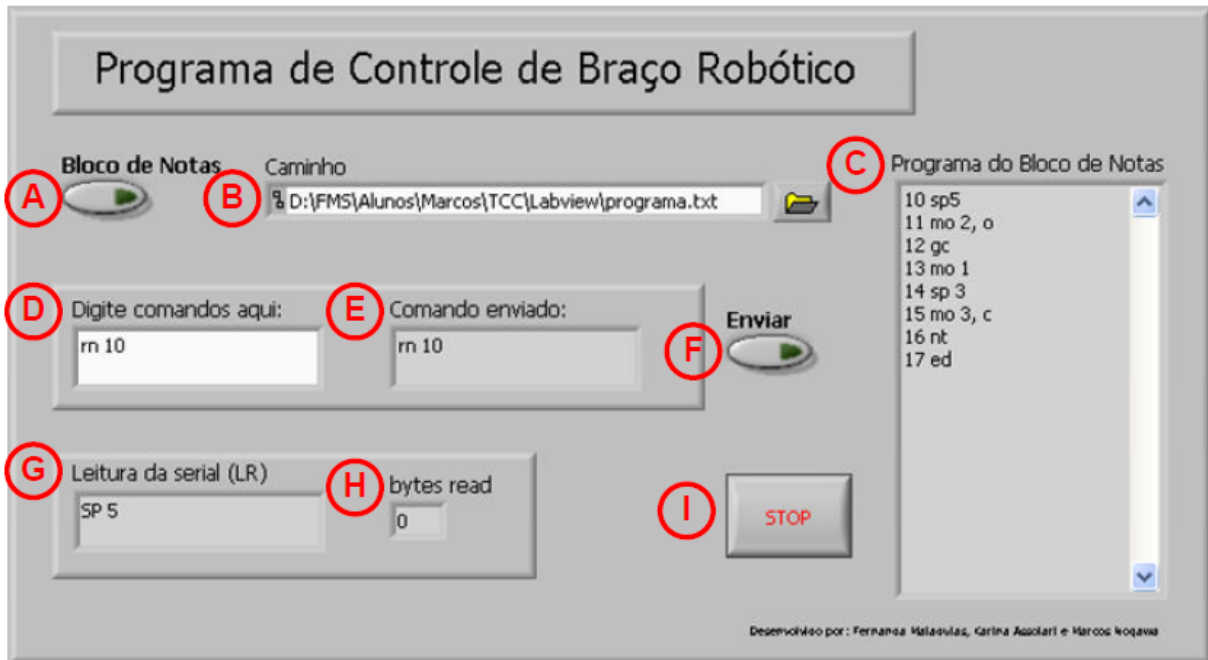


Figura 25 – Tela do programa do Labview finalizado
Fonte: Autoria própria.

3.2 Eletrônica

O conteúdo abordado nesta seção foi subdividido em: microcontroladores, luva, módulo fonte e teclados. Na figura 26 está um diagrama que representa estas subdivisões resume o que é tratado em cada tema.

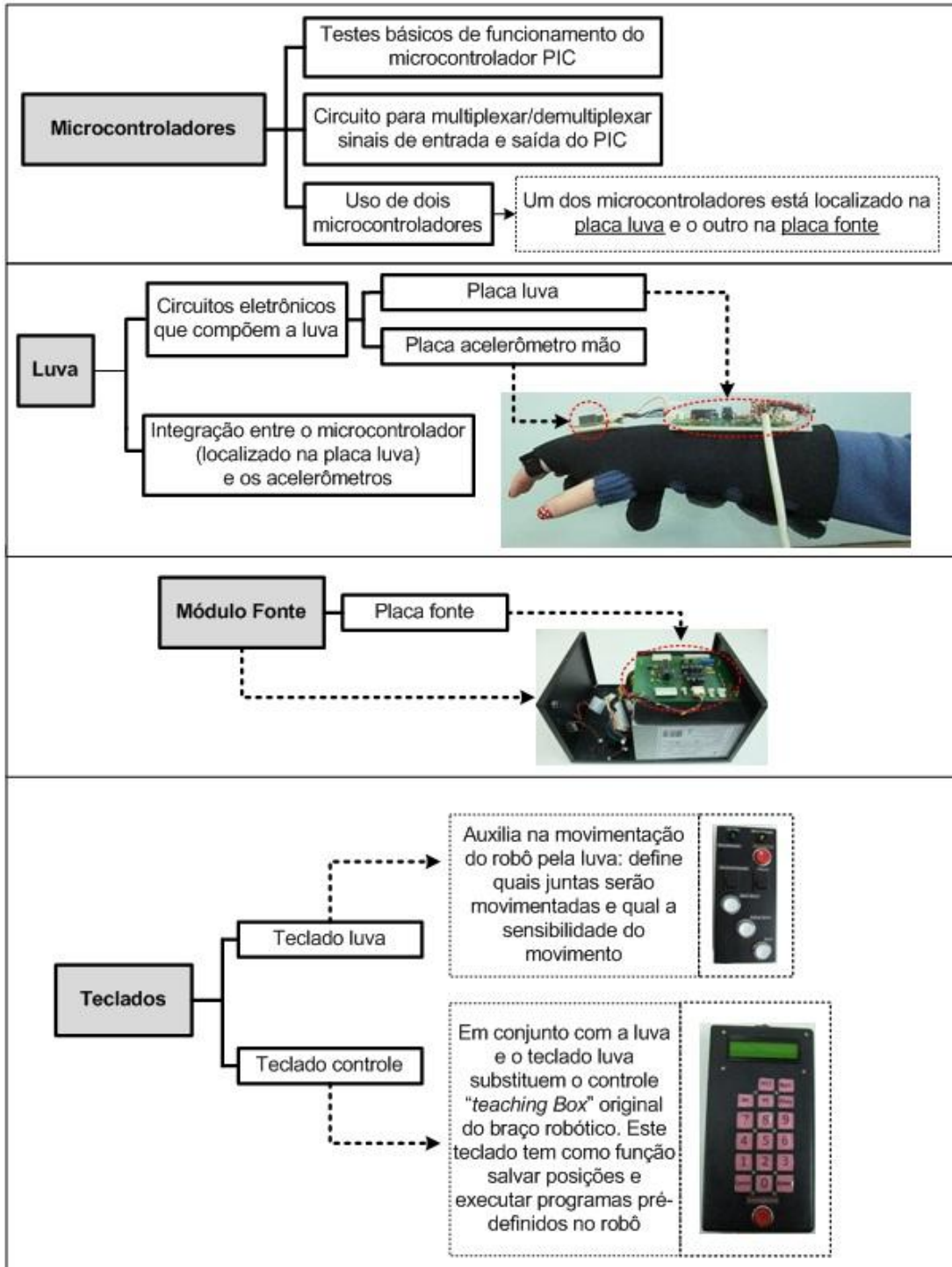


Figura 26 – Diagrama dos temas abordados no tópico de eletrônica
Fonte: Autoria própria.

3.2.1 Microcontroladores

O passo inicial foi montar o circuito básico do microcontrolador PIC16F873a, em *proto board*, de acordo com o *datasheet* (MICROCHIP TECHNOLOGY, 2003) fornecido

pelo fabricante (figura 27), para em seguida fazer programas simples de teste, transferir para o PIC e visualizar os resultados.

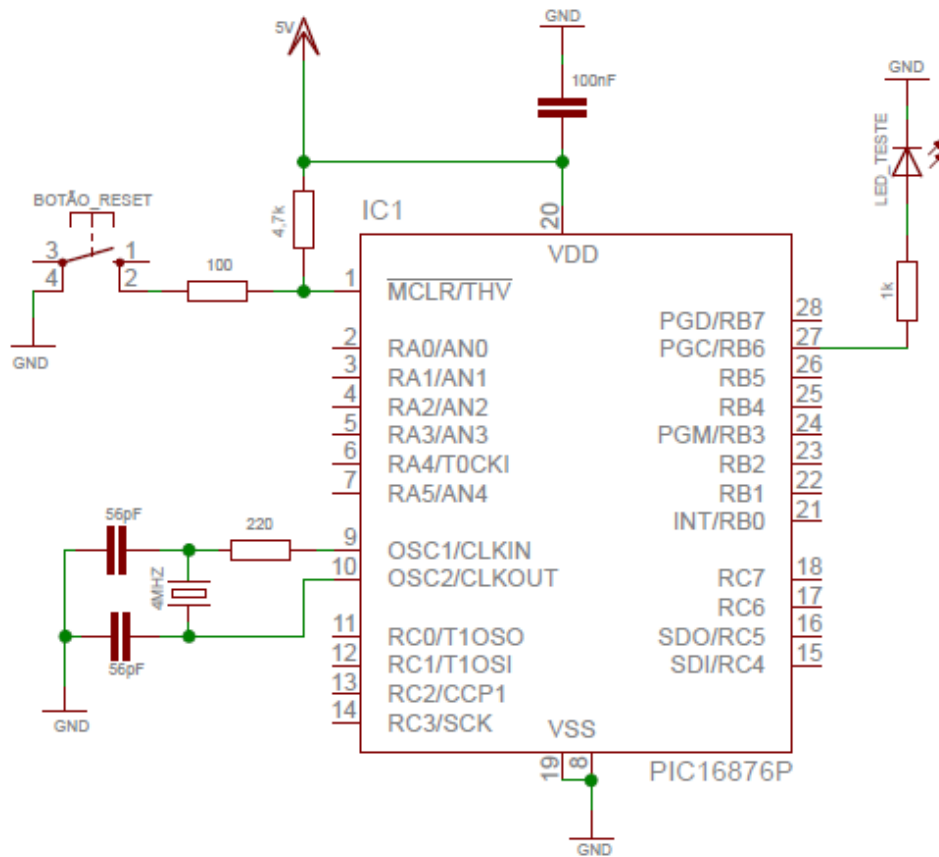


Figura 27 – Circuito básico do microcontrolador PIC
Fonte: Autoria própria.

A princípio foi utilizado um gravador de PIC pertencente à equipe (figura 28). Nessa situação trabalha-se com os *softwares* Winpic (figura 29) e CCS (figura 30). O *software* Winpic é utilizado para transferir o programa, desenvolvido e compilado no *software* CCS, para o microcontrolador através da porta serial.



Figura 28 – Foto do gravador de PIC EDUTECHKITS
Fonte: Autoria própria.

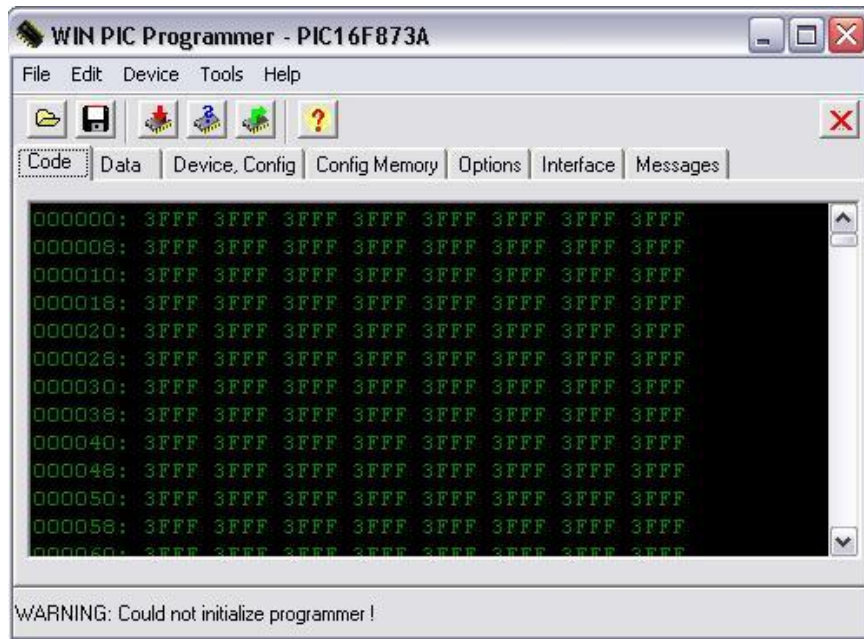


Figura 29 – Exemplo de tela do software Winpic
Fonte: Autoria própria.

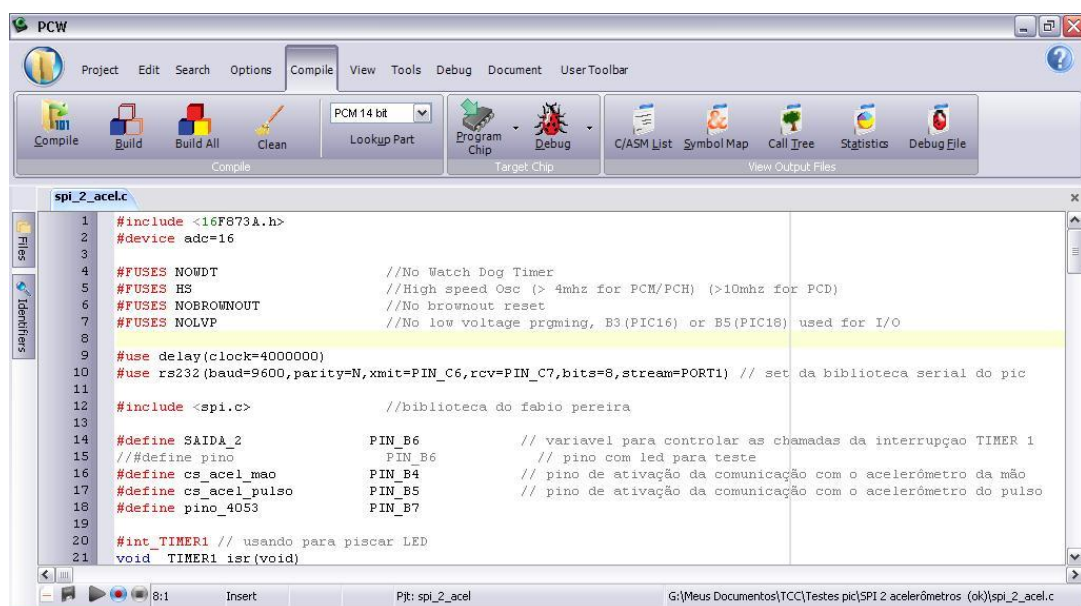


Figura 30 – Exemplo de tela do software CCS
Fonte: Autoria própria.

A prioridade da equipe foi trabalhar no LASIM. Devido à incompatibilidade de horários entre o laboratório e os integrantes da equipe, algumas vezes foi necessário utilizar o laboratório livre do Departamento Acadêmico de Eletrônica (DAELN) da UTFPR. A tentativa inicial foi utilizar o gravador de PIC citado anteriormente. No LASIM este equipamento é utilizado diretamente na porta serial dos computadores disponíveis neste laboratório. No laboratório disponível para trabalhar com os circuitos eletrônicos no DAELN não há computadores disponíveis, sendo necessário o uso de um *notebook* levado pelo usuário. Neste

não existe porta serial; assim foi testado o gravador com um conversor USB (*Universal Serial Bus*)/serial. Infelizmente a corrente mínima necessária para o funcionamento do gravador não é fornecida pelo conversor.

Como alternativa foi utilizado o gravador PICSTART Plus (figura 31) emprestado pelo DAELN que funciona corretamente com o conversor.



Figura 31 – Gravador PICSTART Plus
Fonte: Autoria própria.

Com este gravador são utilizados os seguintes *softwares*: Mplab (figura 32) e CCS (explicado anteriormente). O *software* Mplab exerce a função de transferir para o microcontrolador o código elaborado no CCS.

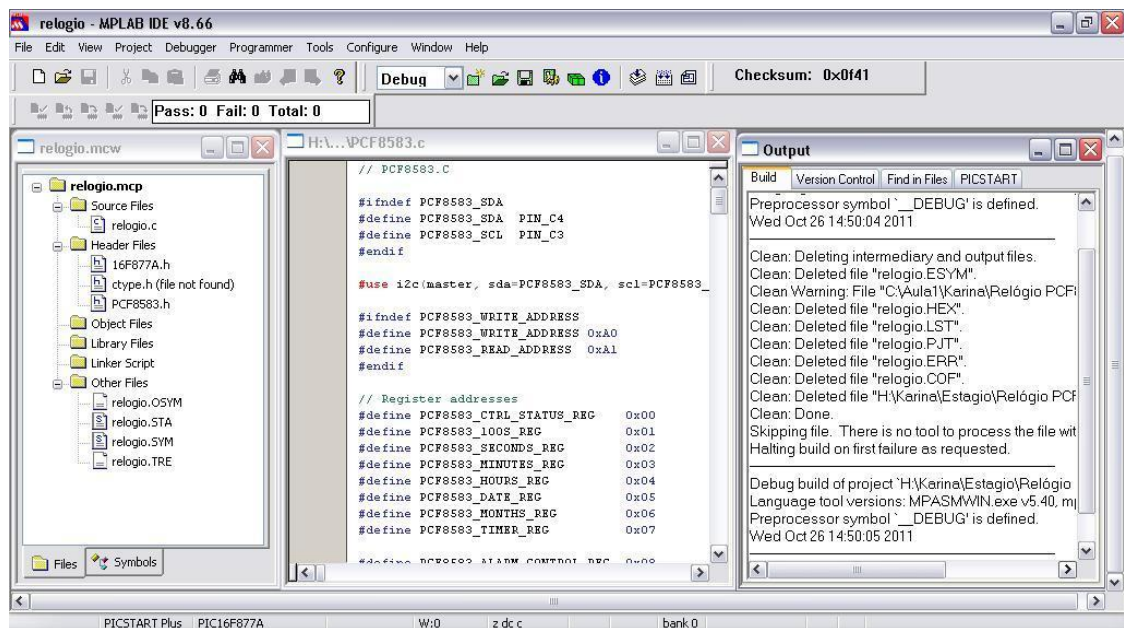


Figura 32 – Exemplo de tela do *software* Mplab
Fonte: Autoria própria.

O segundo teste realizado teve como objetivo montar um circuito que permitisse a comunicação simultânea entre o microcontrolador e computador ou microcontrolador e robô (através da porta serial RS-232). Para isto foi utilizado o componente CD4053 para multiplexar/demultiplexar os sinais de entrada e saída do PIC para duas portas seriais RS-232.

Na primeira tentativa de utilização do CD4053, seguindo somente o *datasheet* (FAIRCHILD SEMICONDUCTOR CORPORATION, 2000), o sinal não passava sempre pelo componente. Seguindo um circuito (figura 33) apresentado por Keen (2000), a divisão do sinal do PIC para as duas portas seriais foi bem sucedida.

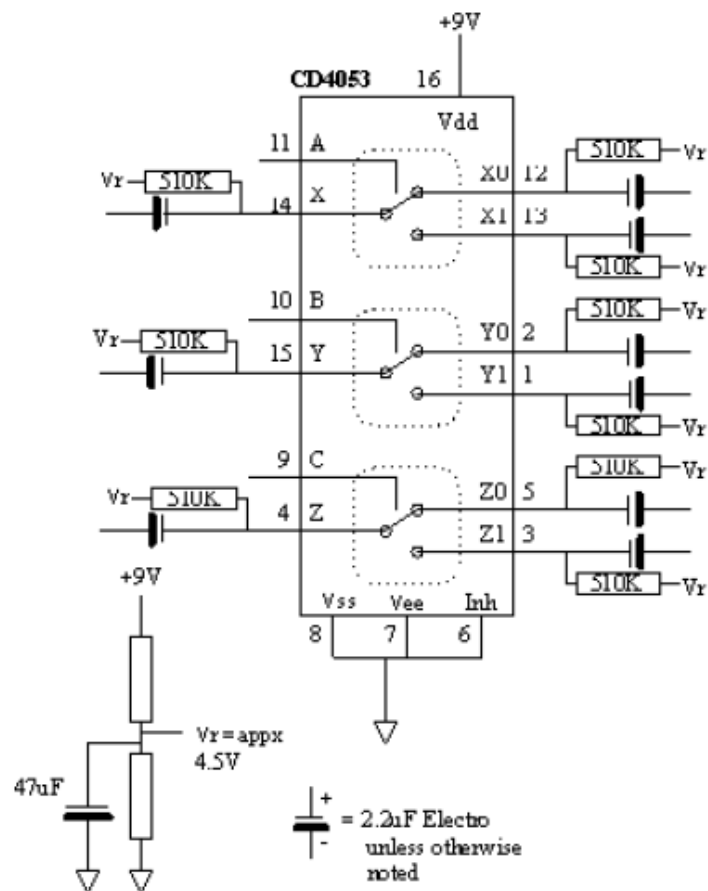


Figura 33 – Circuito utilizando o CD4053
Fonte: KEEN (2000), página 1.

Apesar deste circuito atender o objetivo de estabelecer comunicação simultânea entre os componentes, foi observado que esta era uma etapa desnecessária ao trabalho já que todo o controle foi desenvolvido para que o microcontrolador processe os dados recebidos pelos periféricos e controle o robô sem necessidade do computador.

Devido à grande quantidade de portas digitais utilizadas e também de algoritmos de programação do PIC, tornou-se necessária a utilização de dois microcontroladores. Um dos

microcontroladores fica na placa fonte (seção 3.2.3) e o outro está localizado na placa luva (seção 3.2.2.1). A descrição das funções atribuídas a cada microcontrolador pode ser observada na figura 34.

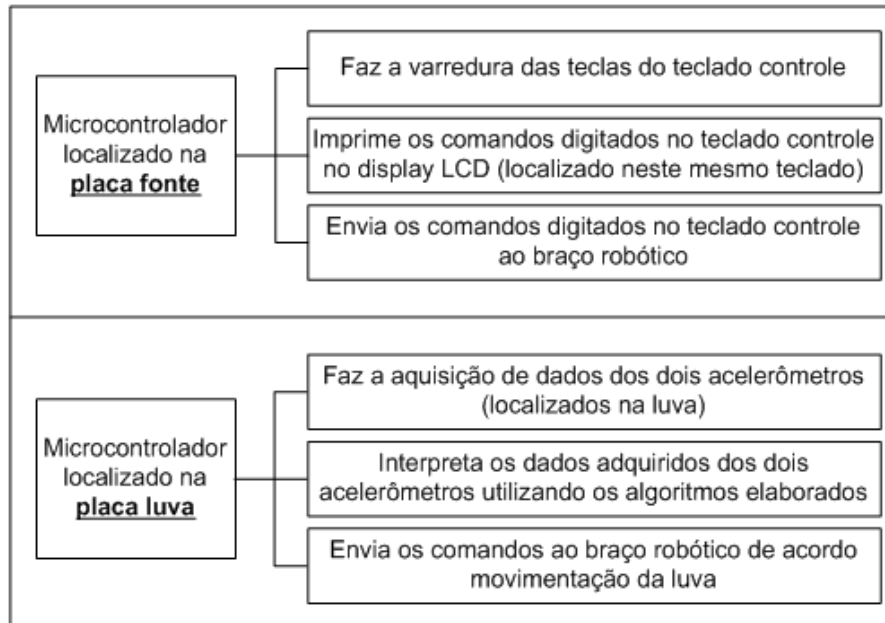


Figura 34 – Funções atribuídas aos microcontroladores
Fonte: Autoria própria.

A descrição da programação dos microcontroladores está na seção 3.6.

A utilização de dois microcontroladores tornou necessário multiplexar os sinais de saída destes para a entrada do robô. O circuito para realizar esse processo foi o do CD4053, descrito anteriormente. Foi utilizado um botão do teclado controle para a escolha de qual dos microcontroladores PIC irá se comunicar com o robô.

3.2.2 Luva

3.2.2.1 Circuitos eletrônicos que compõem a luva

A luva é composta por dois circuitos eletrônicos denominados de placa acelerômetro mão e placa luva. Na figura 35 está uma foto que indica a localização destas duas placas na luva.

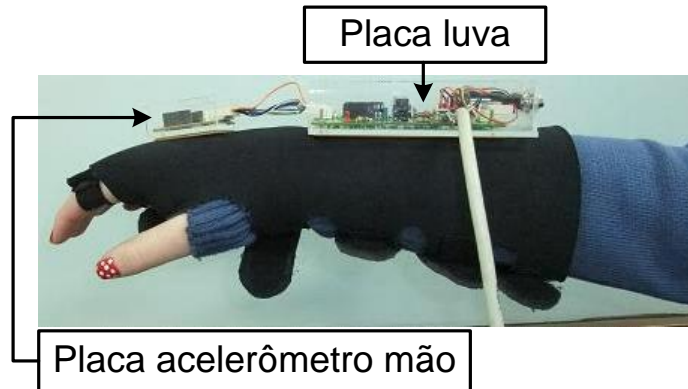


Figura 35 – Circuitos eletrônicos que compõem a luva
Fonte: Autoria própria.

O circuito acelerômetro mão (apêndice D) é composto por um acelerômetro e um TL082. O uso do TL082 é explicado na seção 3.2.2.2. Na figura 36 encontra-se a foto da placa finalizada deste circuito, em vistas superior e inferior.



Figura 36 – Fotos do circuito acelerômetro mão finalizado
Fonte: Autoria própria.

Os principais elementos que compõem a placa luva (apêndice E) são: um microcontrolador, um acelerômetro e um TL082. A foto desta placa finalizada pode ser observada na figura 37.

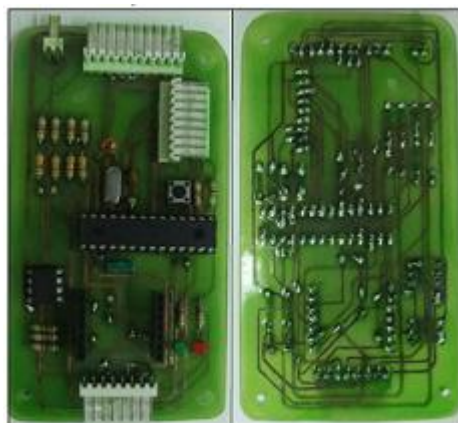


Figura 37 – Fotos do circuito luva finalizado
Fonte: Autoria própria.

3.2.2.2 Integração entre microcontrolador e acelerômetro

O processo de integração entre PIC e acelerômetro exige que seja projetado um circuito conversor de sinal, pois o microcontrolador trabalha com tensão de 5V e o acelerômetro com 3V. A solução está no uso de dois circuitos: um divisor de tensão e um amplificador de tensão.

O circuito divisor de tensão (figura 38) é um circuito simples que consiste em trabalhar com valores de corrente e resistores para, neste caso, diminuir a tensão vinda do microcontrolador PIC fazendo com que o sinal chegue em 3V até o acelerômetro (ACEL).

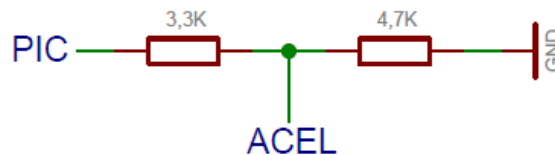


Figura 38 – Circuito divisor de tensão
Fonte: Autoria própria.

O circuito amplificador (figura 39) executa o papel contrário do circuito divisor de tensão, ou seja, amplifica o sinal de saída do acelerômetro ACEL (SDO – *Serial Data Output*) para que chegue até o PIC (SDI – *Serial Data Input*) com sinal de 5V. Este circuito utiliza o amplificador operacional TL082 que é alimentado simetricamente.

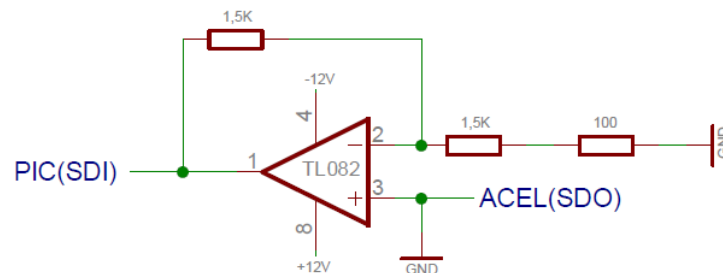


Figura 39 – Circuito amplificador de tensão
Fonte: Autoria própria.

Pode ser observado na figura 40 o circuito de integração entre PIC e um dos acelerômetros.

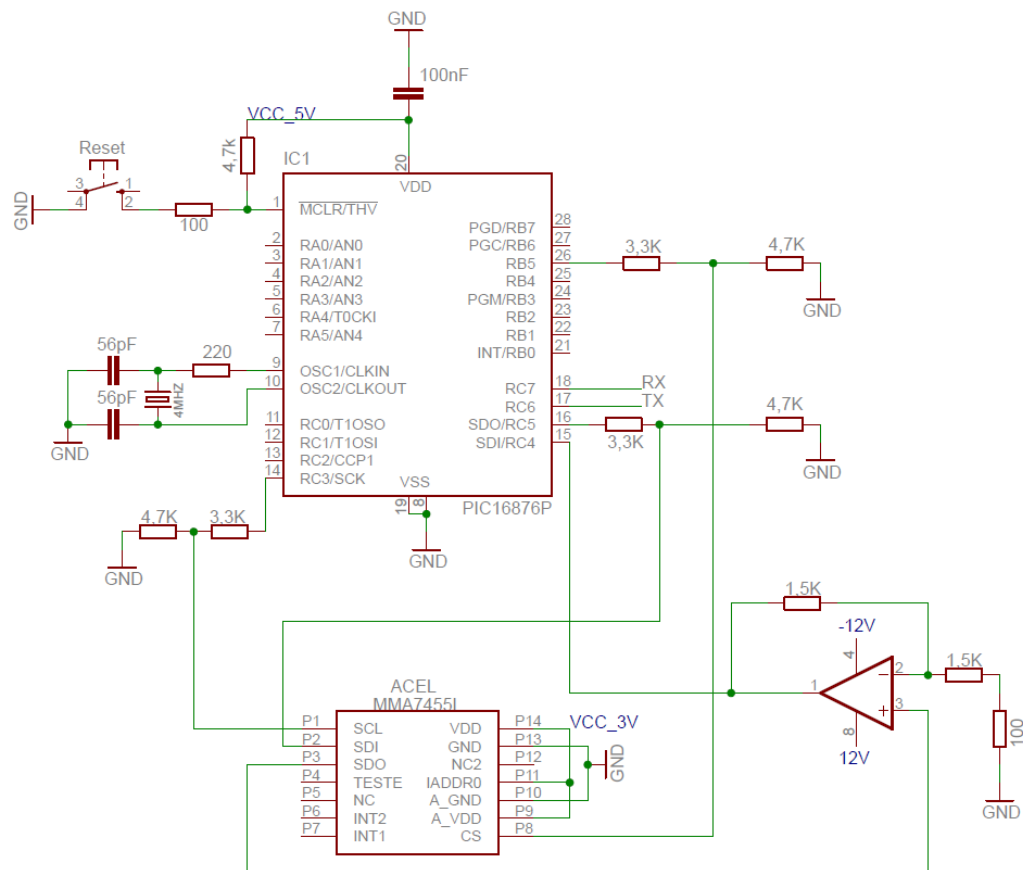


Figura 40 – Circuito integração PIC e acelerômetro
Fonte: Autoria própria.

Até então haviam sido feitos testes com apenas um acelerômetro. O próximo passo foi integrar o segundo acelerômetro. São interligados entre os acelerômetros os sinais de *CLOCK* e PIC (SDO). Foi adicionado na programação do microcontrolador um modo de seleccionar com qual acelerômetro será estabelecida a comunicação (pino CS), pois é utilizado apenas um acelerômetro por vez.

Nos testes em *protoboard* foi detectado um sinal de ruído na entrada do PIC. A solução encontrada foi colocar um capacitor para filtrar este sinal. Na placa de circuito impresso verifica-se que não há este ruído, tornando-se desnecessário utilizar este capacitor de filtro. O ruído provavelmente foi causado pelo *protoboard*.

É recomendado que o acelerômetro fique o mais próximo possível do controlador (FREESCALE SEMICONDUCTOR, 2007) porque a corrente do pino ACEL (SDO) é muito pequena. Caso o acelerômetro encontre-se distante do controlador, ocorre falha na comunicação entre eles. Como pode ser observado na seção 3.2.2.1, o acelerômetro localizado na placa acelerômetro não está há uma distancia maior do microcontrolador do que o

segundo acelerômetro localizado na placa luva. Este problema foi solucionado com o uso do amplificador operacional TL082.

3.2.3 Módulo Fonte

O módulo fonte é composto por uma fonte de alimentação e um circuito eletrônico denominado de placa fonte, conforme indicado na figura 41.

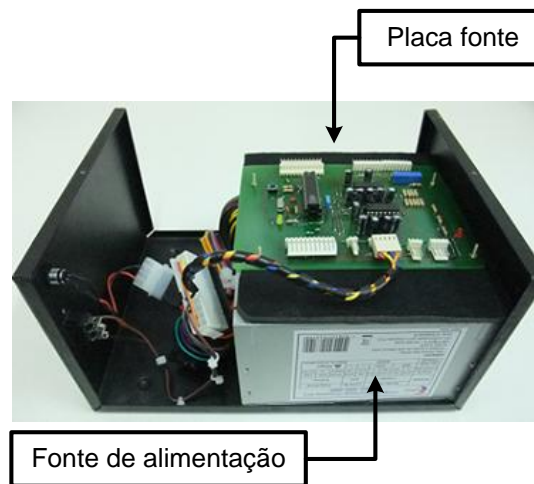


Figura 41 – Foto da fonte
Fonte: Autoria própria.

A fonte de alimentação foi adaptada de uma fonte padrão ATX (*Advanced Technology Extended*) de computador.

A placa fonte (apêndice F) tem como elementos principais um microcontrolador PIC, conectores, CD4053 e MAX232. A pinagem dos conectores encontra-se no apêndice G. Na figura 42 pode ser observada a foto desta placa finalizada.

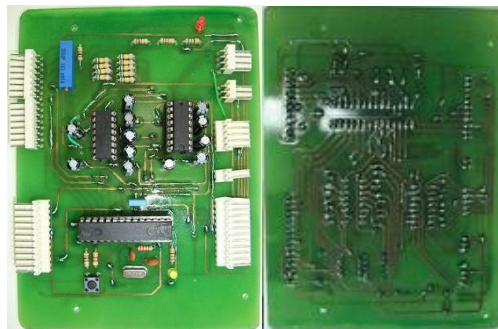


Figura 42 – Fotos da placa fonte finalizada
Fonte: Autoria própria.

3.2.3.1 Teclados

Foram elaborados dois teclados denominados de teclado luva e teclado controle. O primeiro auxilia o uso da luva na movimentação do robô e o segundo, em conjunto com a luva e o teclado luva, substituem o controle original do robô.

Conforme foram sendo realizados testes com a luva foi verificada a necessidade de um controle de sensibilidade para a movimentação do robô para, por exemplo, quando o usuário desejar fazer um ajuste fino na posição do robô. Esse ajuste de sensibilidade é feito através de um potenciômetro encontrado no teclado luva. Este componente está ligado diretamente a uma porta analógica do PIC. As portas analógicas do microcontrolador convertem o sinal analógico (0-5V) em um sinal digital de 10 bits.

Na figura 43 está a foto do teclado luva finalizado e a tabela 2 explica as funções dos seus botões.



Figura 43 – Foto do teclado luva finalizado
Fonte: Autoria própria.

Tabela 2 – Tabela de funções do teclado luva

Tecla	Função
Pausa	Pausa a luva
Punho/Cotovelo	Alterna a junta a ser controlada
Nest	Envia o robô para a origem
Abrir Garra	Muda estado da garra (abrir)
Fechar Garra	Muda estado da garra (fechar)
Sensibilidade	Ajusta a sensibilidade de movimentação do robô
Emergência	Para o sistema

Fonte: Autoria própria.

O circuito esquemático do teclado luva encontra-se no apêndice H.

As fotos do circuito finalizado do teclado luva (vistas superior e inferior, respectivamente) podem ser observadas na figura 44.

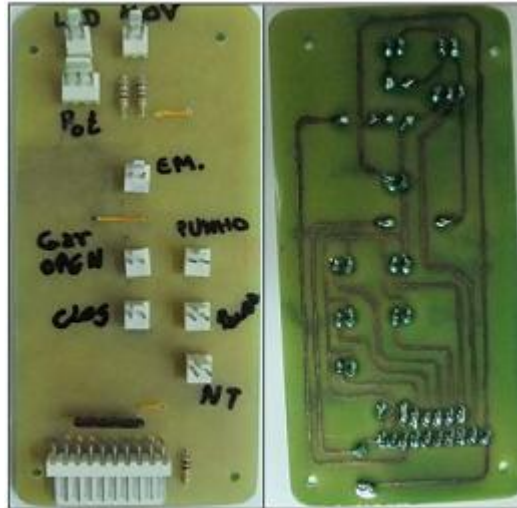


Figura 44 – Fotos do circuito do teclado luva finalizado
Fonte: Autoria própria.

Para a confecção do teclado controle foi utilizado um circuito de teclado matricial composto de seis linhas e três colunas, visando otimizar a utilização das portas de entrada e saída digitais do PIC. O circuito esquemático deste teclado encontra-se no apêndice I.

Para a visualização dos comandos digitados no teclado controle foi utilizado um *display* LCD (*Liquid Crystal Display*) WH1602A.

Foi criada uma placa de testes, um programa de varredura das portas e feita a interligação com o *display* LCD. Nos primeiros testes verificou-se erro e, a princípio, acreditou-se que o problema estava no circuito. Após algumas modificações e mais testes chegou-se à conclusão de que o problema estava na lógica de varredura das portas. Feita a modificação do circuito novamente e correção da lógica de varredura, os testes foram bem sucedidos.

Na figura 45 está a foto do teclado controle finalizado e a tabela 3 explica as funções dos seus botões.

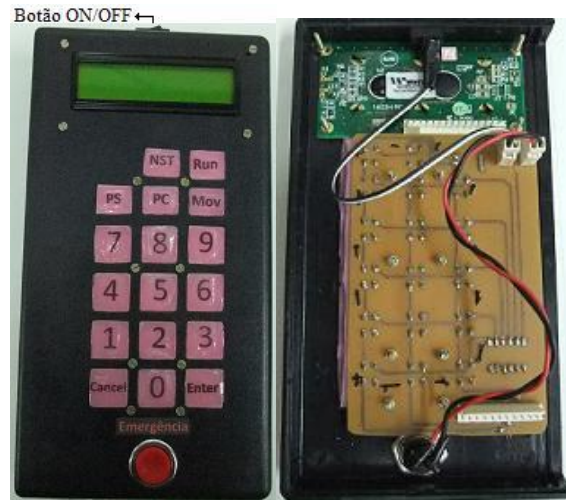


Figura 45 – Fotos do teclado controle finalizado
Fonte: Autoria própria.

Tabela 3 – Tabela de funções do teclado controle

Tecla	Função
NST (<i>Nest</i>)	Posição de descanso do robô
Run	Roda o programa que está armazenado no robô
PS (<i>Position Save</i>)	Salva a posição
PC (<i>Position Clear</i>)	Limpa a posição
Mov	Move para a posição selecionada
Numeração 0 a 9	Determina a numeração da linha em que será salva a posição
Cancel	Cancela a última operação
Enter	Envia o comando digitado
ON/OFF	Liga e desliga o teclado (localizado na parte superior)
Emergência	Para o sistema (botão vermelho localizado na parte inferior)

Fonte: Autoria própria.

Na figura 46 estão as fotos da placa do teclado controle finalizado (vistas superior e inferior, respectivamente).



Figura 46 – Fotos do circuito teclado controle finalizado
Fonte: Autoria própria.

Além dos botões de emergência localizados nos teclados, existe ainda um terceiro localizado na caixa do módulo fonte. Eles estão interligados e ligados diretamente na porta

RB0 do PIC. Esse pino é o de interrupção externa do microcontrolador. Quando acionado qualquer um dos botões de emergência, o sistema para de funcionar.

3.3 Confeção de placas de circuito impresso

A equipe optou por fabricar artesanalmente as placas de circuito impresso devido ao menor custo. Foi utilizado um método básico de fabricação que consiste nos seguintes passos:

- a. Imprimir o circuito (das trilhas) em folhas de transparência ou papel *couché*;
- b. Transferir para placa de fibra de vidro ou fenolite utilizando ferro de passar roupas;
- c. Conferir se as trilhas ficaram certas e reparar o que for necessário;
- d. Efetuar, por processo de imersão, a corrosão do cobre da placa utilizando perclorato de ferro;
- e. Perfurar a placa com furadeira manual;
- f. Soldar componentes; e
- g. Testar a placa.

Na fabricação das duas primeiras placas foi encontrado um problema: quando se transfere (com o ferro) o desenho impresso do circuito para a placa ocorre redução nas dimensões do circuito devido ao calor. A solução é imprimir o desenho um pouco maior.

A princípio a equipe encontrou problemas de mau contato nas placas fabricadas, mas com auxílio de multímetro foram detectados os pontos críticos e reforçadas as soldas até que os problemas foram corrigidos.

3.4 Acelerômetro

Na etapa de integração do PIC com o acelerômetro foram estudados os protocolos de comunicação SPI e I2C. Foi escolhido o primeiro por ser mais simples de se trabalhar.

Para estudar os dados do acelerômetro foram configurados os protocolos de comunicação com o PIC (SPI) e do PIC com o computador (RS-232). Foi escolhido o Labview para visualizar a resposta do acelerômetro.

Após testes na placa do PIC (descritos na seção 3.2.1) foi feita a integração do PIC com o Labview (via comunicação serial). O primeiro teste foi feito diretamente entre PIC e

Labview, mas a comunicação falhou. Foi feita então a comunicação entre PIC e hiperterminal e a comunicação ocorreu normalmente, levando à conclusão que o problema estava na configuração do Labview. Feitas as devidas configurações, a comunicação foi estabelecida conforme esperado.

Inicialmente, com o acelerômetro em um local plano e em repouso, obtivemos um valor fixo em cada eixo indicando a posição de referência. Os resultados obtidos foram coerentes com as explicações das seções 2.5 e 2.6.

Foram feitos três testes para estudar mais a fundo o acelerômetro: aquisição de dados com variação angular, movimento retilíneo e movimento retilíneo com o acelerômetro em ângulo de aproximadamente 45° em relação ao plano do chão.

A aquisição de dados com variação angular do acelerômetro foi feita com o auxílio de um transferidor. Com esse instrumento foi possível encontrar uma relação entre ângulos e a resposta do acelerômetro. A figura 47 indica a variação dos dados do eixo Y do acelerômetro adquiridos num período de 4,3 segundos em uma movimentação de 0° a 90° com retorno a 0° em relação ao plano do chão.

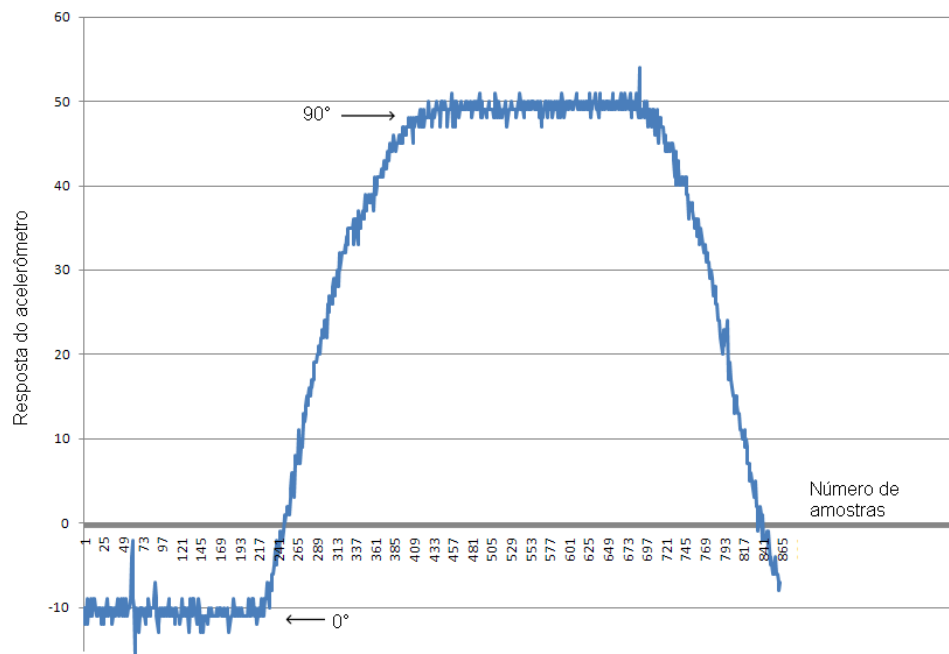


Figura 47 – Dados do acelerômetro na aquisição com variação angular
Fonte: Autoria própria.

Esses valores foram inseridos numa planilha do *software* Microsoft Excel com o qual foi produzida uma equação de segundo grau (figura 48) que estima a resposta do acelerômetro de acordo com a variação do ângulo em relação ao plano chão. Essa equação tornou possível

calcular a posição angular do acelerômetro com precisão suficiente, e assim identificar pequenas variações de movimento do acelerômetro.

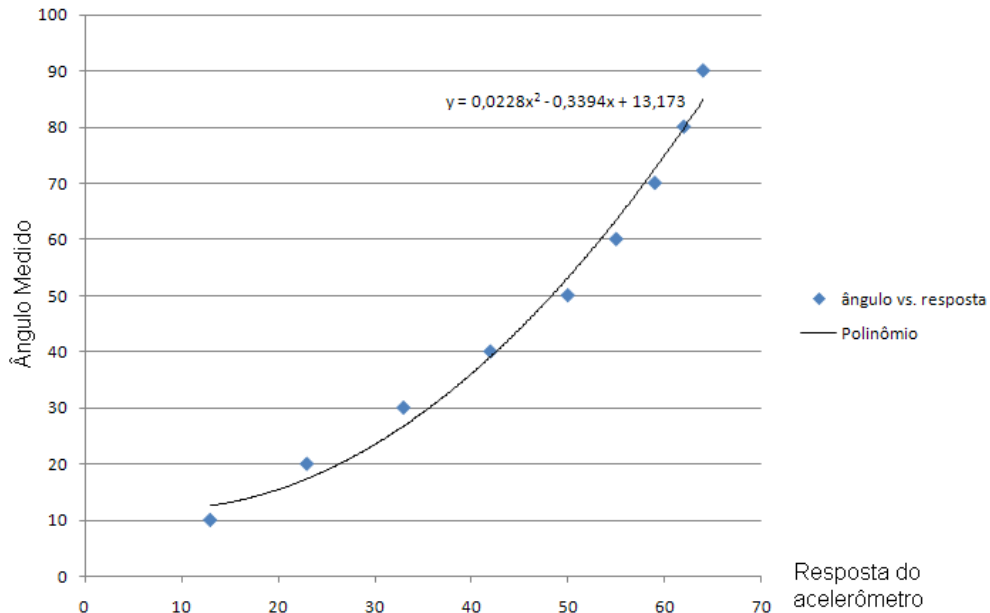


Figura 48 – Gráfico da equação da aquisição de dados angulares
Fonte: Autoria própria.

A aquisição de dados de movimento retilíneo foi feita de duas maneiras: com o movimento de um braço de um integrante da equipe e utilizando a esteira do laboratório LASIM. O primeiro teste foi inconclusivo, pois o integrante da equipe não conseguiu repetir o movimento com a precisão necessária em todos os testes. A solução encontrada foi utilizar uma esteira rolante, disponível no LASIM, que realiza movimento uniforme todas as vezes em que é acionada, garantindo que a movimentação seja a mais parecida possível em todos os testes.

Na figura 49 pode ser observado o sinal obtido do acelerômetro no teste realizado com a esteira (movimentação num período de 9,925 segundos). Este teste começa com a esteira parada, sendo em seguida dada a partida e, após o tempo indicado, a esteira para novamente.

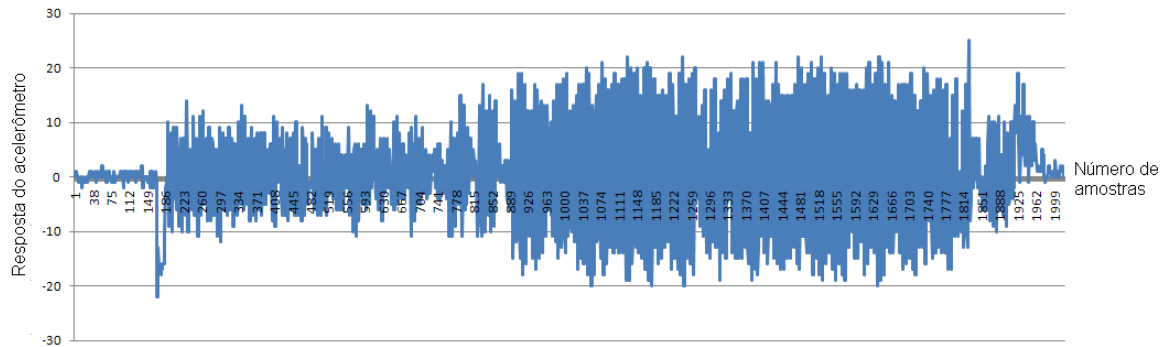


Figura 49 – Dados do acelerômetro no movimento retilíneo
Fonte: Autoria própria.

A aquisição de dados do movimento retilíneo com o acelerômetro em ângulo de aproximadamente 45° em relação ao plano do chão foi realizada pela movimentação do braço de um integrante da equipe. O acelerômetro foi inclinado a 45° de uma superfície plana e movimentado linearmente sobre o eixo Y, conforme indicado na figura 50.

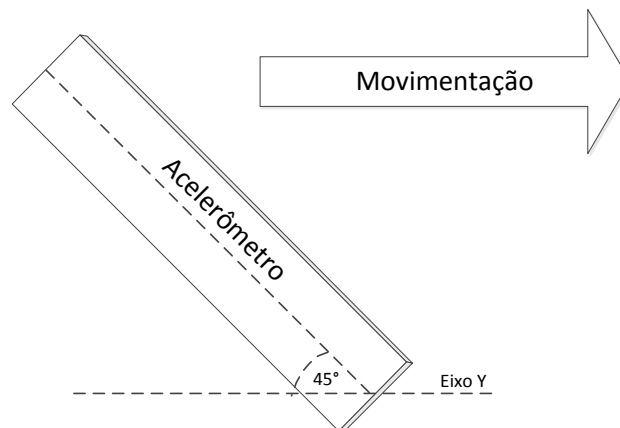


Figura 50 – Movimentação do acelerômetro com ângulo de 45° sobre o eixo Y
Fonte: Autoria própria.

Com isso foi possível obter dados do movimento retilíneo com acelerômetro em ângulo de 45° . Na figura 51 encontram-se os resultados obtidos.

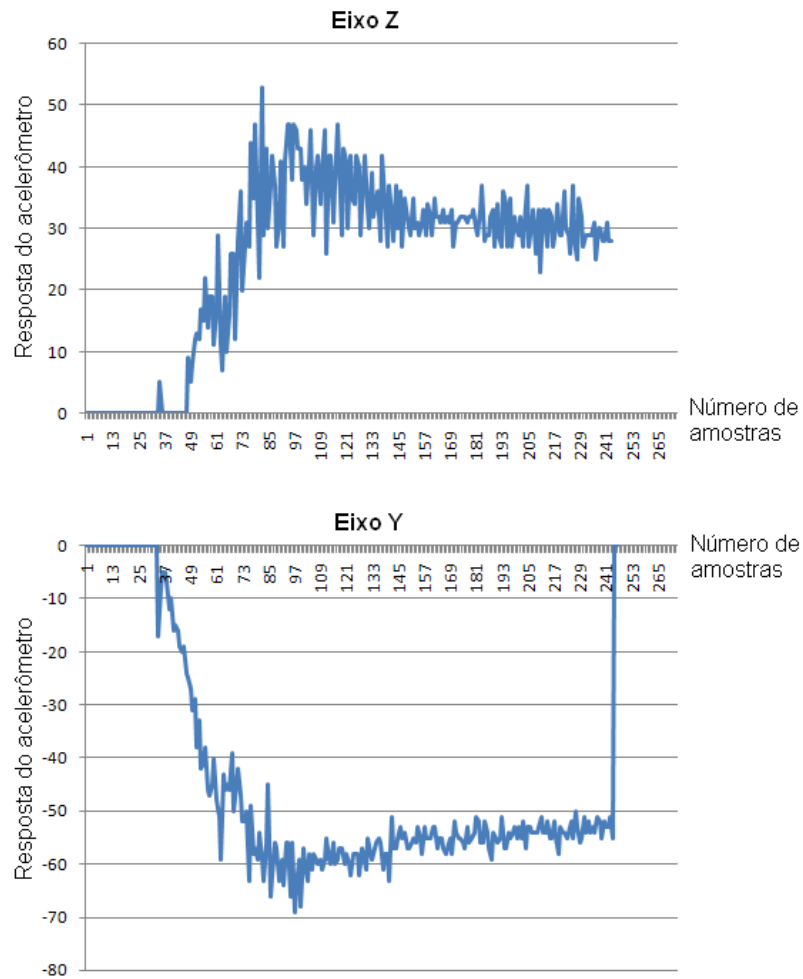


Figura 51 – Dados do acelerômetro (em ângulo de 45° sobre o eixo Y) na aquisição do movimento retilíneo
Fonte: Autoria própria.

Após o teste descrito anteriormente, o acelerômetro foi inclinado a 45° de uma superfície plana e movimentado no eixo W, conforme indicado na figura 52. O eixo W corresponde ao eixo Z do acelerômetro.

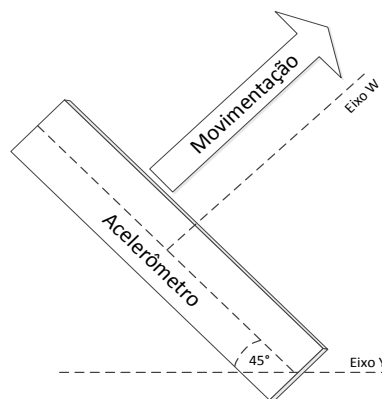


Figura 52 – Movimentação do acelerômetro com ângulo de 45° sobre o eixo W
Fonte: Autoria própria.

Os dados obtidos com esse tipo de movimento podem ser observados na figura 53.

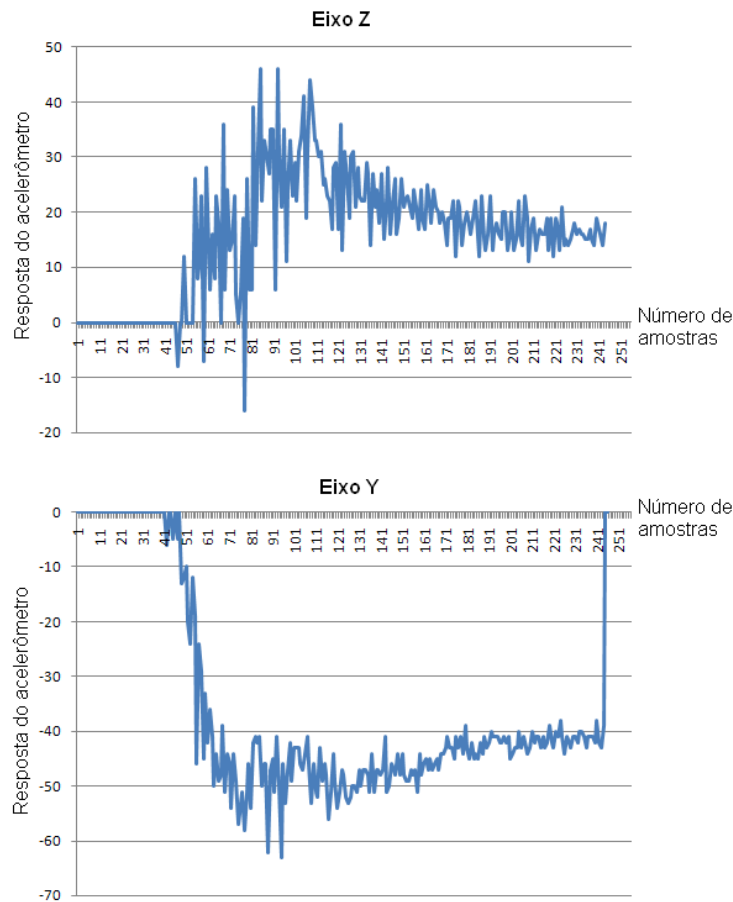


Figura 53 – Dados do acelerômetro (em ângulo de 45° sobre o eixo W) na aquisição do movimento retilíneo
Fonte: Autoria própria.

Após os testes chegou-se a duas conclusões. A primeira é que o processador escolhido para o projeto não tem memória suficiente para interpretar os sinais e realizar os outros cálculos necessários para a movimentação do robô: além de ter que armazenar uma quantidade mínima de amostras de dados do acelerômetro, são necessários cálculos para interpretar esses sinais e identificar os movimentos do usuário. A segunda conclusão baseou-se no fato de que existem vários movimentos diferentes que podem ser executados pelo usuário indicando respostas finais similares, o que dificulta a diferenciação destes movimentos. Um exemplo é fazer a comparação dos dados apresentados nas figuras 51 e 53 apresentadas anteriormente. Como o tratamento desses sinais seria complexo, decidiu-se por uma alternativa mais simples. Com essas conclusões decidiu-se utilizar apenas a aquisição de dados angular para controlar o braço robótico. Dessa forma, o controle se dá apenas pelos ângulos dos acelerômetros com referência aos eixos correspondentes.

3.5 Estrutura Mecânica

A escolha do material da luva se baseou em uma pesquisa de um tecido com alta resistência e boa maleabilidade. A primeira característica está ligada a necessidade de fixação do circuito eletrônico na luva. A segunda característica deve se a necessidade de adaptação da luva aos diversos tamanhos de mãos dos usuários, e também para facilitar no momento do usuário vestir a luva. O material escolhido, em pesquisa em campo, foi o *neoprene*. Esse material é utilizado na fabricação de roupas para mergulho, *surf* e capa de *notebook*.

A luva foi projetada com o objetivo de se adaptar em diversos tamanhos de mãos. Foi elaborado o encaixe dos dedos (exceto o polegar) com elástico, lembrando uma luva convencional “sem dedos”. Para ajustes de tamanho foi usado um conector consistido em ganchos e voltas usado para conectar objetos (Velcro®). Na luva foram fixadas as placas acelerômetro mão e placa luva.

Foi elaborada uma caixa de proteção para os circuitos da luva. O material escolhido foi o policarbonato compacto, devido à transparência e à relativa resistência à flexão e ao impacto. Para usinagem do material, foi utilizada a fresa da minifábrica de máquinas convencionais da UTFPR e para colagem do mesmo foi necessário o uso do clorofórmio. Essa substância foi concedida pelo Departamento Acadêmico de Química e Biologia (DAQBI) da UTFPR. Na figura 54 estão indicados os encaixes para os dedos, os ajustes com Velcro® e as caixas de proteção dos circuitos na luva finalizada.

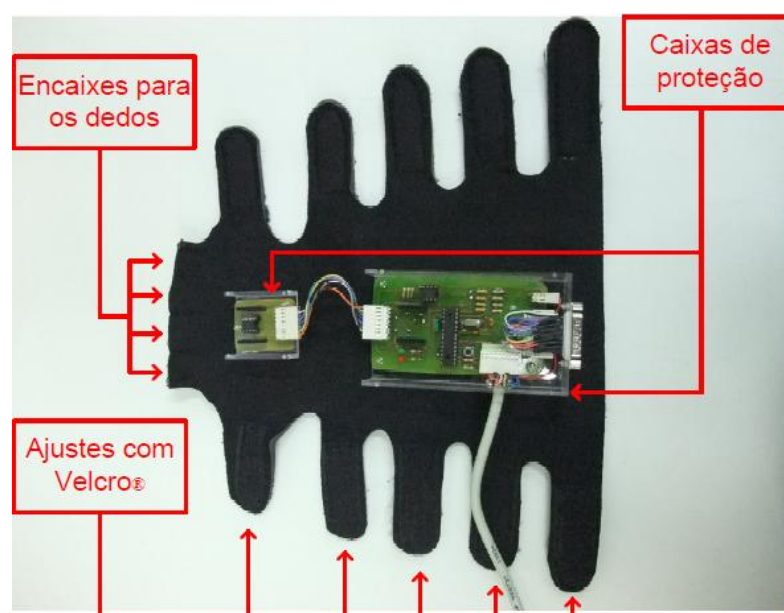


Figura 54 – Foto da luva finalizada
Fonte: Autoria própria.

Na figura 55 está a foto de um usuário utilizando a luva.



Figura 55 – Foto de um usuário com a luva
Fonte: Autoria própria.

Os teclados (teclado luva e teclado fonte) foram adaptados à caixas compradas prontas. Essas caixas foram perfuradas na minifábrica de máquinas convencionais da UTFPR para adaptação dos botões. No teclado fonte foi utilizada uma membrana de silicone adaptada de um teclado convencional de computador.

O módulo fonte consiste em uma fonte de alimentação padrão ATX adaptada de computador, a placa fonte e uma caixa de ferro. A caixa de ferro (comprada pronta) foi perfurada na minifábrica de máquinas convencionais da UTFPR para adaptar a fonte, os conectores, os botões (emergência e *ON/OFF*) e o LED (*Light-emitting Diode*) que indica o estado da fonte. Na figura 56 está a foto do módulo fonte em diversas vistas.



Figura 56 – Fotos da fonte finalizada
Fonte: Autoria própria.

3.6 Programação

Foram analisadas duas possibilidades para o desenvolvimento do algoritmo de movimentação do robô através dos comandos MP ou MJ (conforme explicados no apêndice A). A seguir, nas tabelas 4 e 5, são apresentadas as vantagens e desvantagens dos dois métodos estudados *Move Position* (MP) e *Move Joint* (MJ), respectivamente.

Tabela 4 – Tabela comparando vantagens e desvantagens do método *Move Position* (MP)

Vantagens	Desvantagens
Mais intuitivo para o usuário	Mapear o plano cartesiano
Limites físicos do robô menos complexos para definir	Implementação complexa na programação
	Tratamento de sinal do acelerômetro complexo
	Programa ocupa mais memória do PIC

Fonte: autoria própria.

Tabela 5 – Tabela comparando vantagens e desvantagens do método *Move Joint* (MJ)

Vantagens	Desvantagens
Implementação mais fácil na programação	Menos intuitivo para o usuário
Programa ocupa menos memória do PIC	Limites físicos do robô difíceis de definir
Tratamento de sinal do acelerômetro menos complexo	

Fonte: Autoria própria.

Analisando as características de cada método, a primeira escolha foi o *Move Position* por ser mais intuitivo para o usuário. Porém devido à dificuldade para a diferenciação entre os movimentos lineares e da combinação entre movimento linear e angular (conforme descrito na seção 3.4), a equipe optou pelo desenvolvimento através do *Move Joint*.

Existem botões de emergência interligados conectados aos pinos de interrupção externa dos microcontroladores. Quando um dos botões é acionado ambos os programas são desativados e é cortada a comunicação deles com o braço robótico.

3.6.1 Programação da luva

Pereira (2003) apresenta em seu livro uma biblioteca de comunicação SPI para o PIC. Esta biblioteca foi escolhida por conter um código genérico que pode ser utilizado com qualquer PIC e é simples de ser entendida e utilizada. Para esta decisão foram estudadas a biblioteca encontrada no *software* CCS e a fornecida por Pereira (2003).

O programa interpreta os sinais dos acelerômetros e de acordo com outros parâmetros definidos no teclado luva envia comandos para a movimentação do robô. Existem as seguintes condições de funcionamento: o botão “pausa” do teclado luva e o botão “ON/OFF” do teclado fonte devem estar na posição ‘0’.

O programa configura as interrupções internas e externas do microcontrolador, habilita a comunicação com robô e configura os acelerômetros. Essas configurações foram feitas de acordo com o *datasheet* fornecido pelo fabricante (FREESCALE SEMICONDUCTOR, 2007), para a utilização com o microcontrolador PIC18F252. Estas se encontram no apêndice J. Usando os registros \$10, \$12 e \$14, do acelerômetro, foi possível

zerar o valor de referência dos eixos X, Y e Z respectivamente. Deste modo pode-se trabalhar diferenciando os dados do acelerômetro em positivo e negativo, de acordo com a direção do movimento.

De acordo com a leitura dos eixos x e y do acelerômetro da mão e do eixo x do acelerômetro do pulso, é possível uma diferenciação dos movimentos feitos pelo usuário. Na figura 57 é demonstrado o movimento de giro do pulso do usuário; essa ação modifica o valor do eixo y do acelerômetro da mão. Com esse dado é feita a movimentação da base (ângulo de movimento θ_1) ou do giro do punho do braço robótico (ângulo de movimento θ_5), regido pelo estado do botão “punho/cotovelo” localizado no teclado luva.

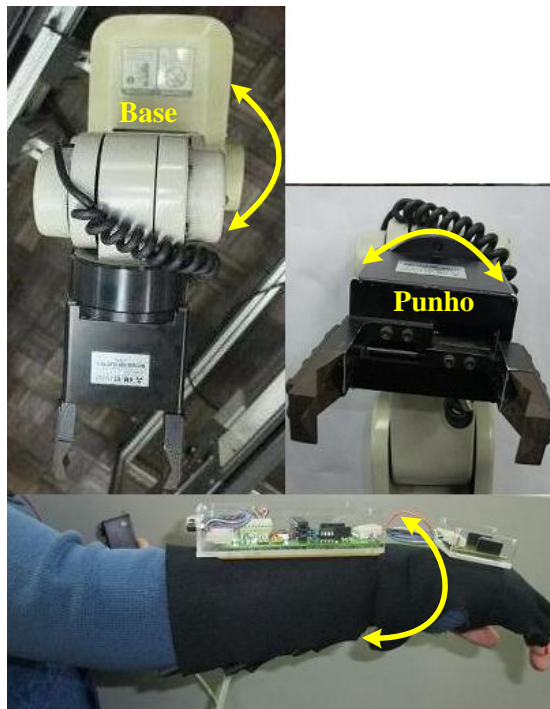


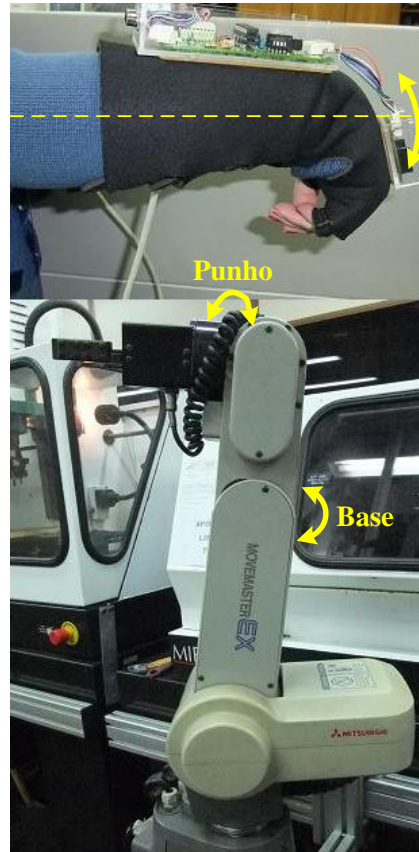
Figura 57 – Foto do movimento pulso-giro
Fonte: Autoria própria.

Na figura 58 o movimento realizado pelo usuário é a inclinação do braço para cima ou para baixo, essa ação modifica o valor do eixo x do acelerômetro do pulso. Com a modificação desse eixo é possível movimentar o ‘ombro’ do braço robótico (ângulo de movimento θ_2). O movimento somente será realizado se o estado do botão “punho/cotovelo” estiver em “cotovelo”.



Figura 58 – Foto do movimento inclinação do braço
Fonte: Autoria própria.

A figura 59 mostra o movimento de inclinação do pulso em relação ao antebraço do usuário, que modifica o valor do eixo x do acelerômetro da mão. A alteração do eixo x do acelerômetro da mão permite a movimentação do ‘cotovelo’ (ângulo de movimento θ_3) ou da ‘munheca’ do braço robótico (ângulo de movimento θ_4), regido pelo estado do botão “punho/cotovelo”.



**Figura 59 – Foto do movimento inclinação do pulso em relação ao antebraço
Fonte: Autoria própria.**

Para realizar o movimento do robô deve se assegurar que este não ultrapasse os limites físicos de cada junta. O microcontrolador PIC envia o comando WH (*where*), via comunicação serial RS232, para que o robô responda com as coordenadas da ponta da garra no eixo cartesiano. Com essas coordenadas torna-se possível calcular os ângulos de cada junta através da cinemática inversa. Esses cálculos foram feitos com as equações descritas na seção 2.4. Aos ângulos obtidos são adicionados os valores de variação de movimento (sensibilidade imposta pelo usuário no teclado luva), sendo esses novos valores comparados com o limite máximo de cada junta; se o limite não for ultrapassado, será enviado ao robô o comando para a movimentação deste.

O fluxograma que explica o funcionamento deste programa encontra-se no apêndice K.

3.6.2 Programação do teclado controle

O PIC que realiza a varredura do teclado matricial está localizado na placa fonte.

A programação do display contido no teclado fonte utiliza como base a biblioteca de LCD elaborada por Pereira (2003). Essa biblioteca foi escolhida pela facilidade de utilização. A peculiaridade desta é utilizar quatro bits de comunicação ao invés dos oito usuais.

Para que o teclado funcione é necessário que o botão “ON/OFF” esteja no estado “ON” (para que a comunicação da luva com o robô seja desativada). A função do programa do teclado controle é fazer a varredura dos comandos, se necessário identificar a posição, e quando solicitado enviar o comando ao robô.

O fluxograma que explica o funcionamento deste programa encontra-se no apêndice L.

4 CONSIDERAÇÕES FINAIS

Com este projeto foi possível adquirir novos conhecimentos sobre acelerômetros (funcionamento e aplicação), sistemas hápticos, confecção de placas de circuito impresso, programação de microcontroladores PIC e programação em ambiente Labview. Foi possível também aprimorar alguns conhecimentos vistos durante o curso, como eletrônica básica, comunicação serial, cinemática, usinagem e *hardware* e *software* do robô Mitsubishi RV-M1.

O objetivo geral deste projeto foi alcançado, pois resolveu os problemas citados no início deste trabalho. A nova interface para programação do robô facilita a sua programação, a luva melhora a dinâmica para controlar o braço robótico e atualmente é possível rodar um programa, que esteja salvo na memória do robô, sem a necessidade do uso de um computador.

Quando é feita uma proposta de projeto nem sempre é possível determinar o melhor método para o desenvolvimento deste, levando em conta conhecimentos ainda a adquirir e a complexidade de execução. Devido a isso, a proposta inicial deste projeto foi alterada em dois pontos: o controle do robô pela luva através do computador foi abandonado; e a comunicação simultânea do microcontrolador da luva com ambos computador e o robô se tornou desnecessária. A primeira modificação do projeto se deu pela complexidade de aprendizagem em programar em ambiente Labview. Infelizmente os conhecimentos adquiridos sobre este ambiente não foram suficientes para programar o controle da luva. A segunda adaptação está ligada à primeira, pois como o computador não foi utilizado para o controle da luva não houve necessidade de implementação da comunicação entre eles.

São sugeridas as seguintes melhorias para este projeto:

- a. Integrar o programa feito no Labview com os outros elementos do sistema de manufatura do laboratório LASIM;
- b. Fazer o programa de controle da luva em ambiente Labview;
- c. Desenvolver um circuito para comunicação sem fio da luva com o computador e com o robô;
- d. Fazer a movimentação do braço robótico pela ponta da garra para que o sistema fique mais intuitivo para o usuário; e
- e. Desenvolver um novo *driver* para o braço robótico utilizando a luva como interface para melhorar a resposta do robô.

REFERÊNCIAS

- BRAGA, Newton. **Fundamentos de robótica e mecatrônica**. Disponível em: <www.newtonbraga.com.br/index.php/artigosmec/777-fundamentos-de-robotica-e-mecatronica-mec001>. Acesso em: 8 ago. 2010.
- CAETANO, Alexandra C. M. **Interfaces hápticas - dispositivos não convencionais de interação**. VII ENCONTRO INTERNACIONAL DE ARTE E TECNOLOGIA. 2008. Brasília. Disponível em: <<http://arte.unb.br/7art/textos/AlexandraCMCaetano.pdf>>. Acesso em: 21 jun. 2011.
- CARRARA, Valdemir. **Apostila de robótica**. Universidade de Braz Cubas. [19--?].
- DAHLHOFF, H.. **Fundamentos da robótica: BSP70**. São Paulo: Festo Didactic, 1993. 120 p.
- FILHO, Roberto A. **Padrão serial RS-232**. Disponível em: <<http://www2.eletronica.org/artigos/eletronica-digital/padrao-serial-rs-232>>. Acesso em: 29 jun. 2011.
- FAIRCHILD SEMICONDUCTOR CORPORATION. **CD4053BC - triple 2-Channel analog multiplexer/demultiplexer**. San Jose: Fairchild Semiconductor Corporation, 2000. 14 p.
- FREESCALE SEMICONDUCTOR. **MMA7455L**. Tempe: Freescale Semiconductor, 2007. 27 p.
- GOMES, André C. **Calibração de propriedades materiais para incorporação de toque em sistemas de realidade virtual**. Disponível em: <http://www.abs-tech.com/admin/modulos/portal/upload/arquivos/17/calibrador_final.pdf> Acesso em: 2 set. 2011.
- GROOVER, Mikell P.. **Automation, production systems, and computer integrated manufacturing**. 3.ed. Pearson Education. 2008.
- KEEN, R.G. **Bypassing and switching with the CD4053 CMOS analog MUX**. 2000. Disponível em: <http://www.geofex.com/article_folders/cd4053/cd4053.htm>. Acesso em: 30 out. 2011.
- KIRNER, Claudio; SISCOOTTO, Robson. **Realidade virtual e aumentada – conceitos projetos e aplicações**. 1.ed. Porto Alegre - RS: Sociedade Brasileira de Computação - SBC, 2007. v. 1. 292 p.
- LAUS, Luís Paulo. **Introdução à robótica**. Universidade Tecnológica Federal do Paraná. 2006.
- LOPES, António Mendes. **Modelação cinemática e dinâmica de manipuladores e estruturas em série**. 2002. 101 f. Dissertação (Mestrado em automação, instrumentação e controle). Faculdade de engenharia, Universidade do Porto. 2002.

MICROCHIP. **Scaling the PIC MCU & dsPIC DSC families**. Disponível em: <http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2551>. Acesso em: 31 set. 2011.

MICROCHIP TECHNOLOGY. **PIC16F87XA data sheet**. Chandler: Microchip Technology, 2003. 234p.

MITSUBISHI. **Industrial micro-robot system RV-M1**. Nagoya: Mitsubishi. [19--]. 260p.

NICOLOSI, Denys Emílio Campion. **Microcontrolador 8051 detalhado**. 3. ed. São Paulo: Érica, 2002.

NIKU, Saeed B.. **Introduction to robotics analysis, systems, applications**. Upper Saddle River, NJ.: Prentice-Hall, 2001. 349 p.

PAZOS, Fernando. **Automação de sistemas & robótica**. Rio de Janeiro: Axcel, 2002. 377 p.

PEREIRA, Fábio. **Microcontroladores PIC linguagem C**. São Paulo: Érica, 2003. 358 p.

PEREIRA, Filipe. Circuitos de Interface para Microcontroladores. **Revista Saber Eletrônica**, São Paulo, mar. 2011. Disponível em: <<http://www.sabereletronica.com.br/secoes/leitura/1714>>. Acesso em: 21 jun. 2011.

ROBOTICS Exchange. **Mitsubishi RV-M1**. Disponível em: <www.roboex.com/s19>. Acesso em: 10 ago. 2010.

ROBÓTICA industrial: aplicação na indústria de manufatura e de processos. São Paulo: Edgard Blücher, 2002. 256p.

RODRIGUES, Carlos E. **Um dispositivo háptico de auxílio à navegação para deficientes**. Disponível em: <<http://www.cin.ufpe.br/~tg/2005-2/cemr.pdf>> Acesso em: 2 set. 2010.

ROSÁRIO, João Maurício. **Princípios de mecatrônica**. São Paulo: Prentice-Hall, 2005. 356p.

SANDOR, B. I. **Engineering mechanics: design**. Englewood Cliffs, N.J.: Prentice-Hall, 1983. 499p.

SIEMEINTCOSKI, Michael; WACKERHAGEN, Guilherme. Comunicação Serial: Protocolo SPI e periférico USART para microcontroladores- Parte 2. **Revista saber eletrônica**, São Paulo, jan. 2008. Disponível em: <<http://www.sabereletronica.com.br/secoes/leitura/279>>. Acesso em: 21 jun. 2011.

SILVEIRA, Paulo Rogério da; SANTOS, Winderson Eugenio dos. **Automação e controle discreto**. 4. ed. São Paulo: Érica, 2002 229 p.

SILVA, João C. S.; LINDER, Marcelo; IRMÃO, Marcos A. da S.; SILVA, Ricardo P. da. **Modelagem cinemática de um robô manipulador**. CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA. 2006. Passo Fundo. Disponível em: <http://www.dee.ufma.br/~fsouza/anais/arquivos/1_66_333.pdf>. Acesso em: 21 jun. 2011.

SOARES, Marcio J. Comunicação RS-232 - Noções básicas - Parte 1. **Revista Saber Eletrônica**, São Paulo, mai. 2008. Disponível em: <<http://www.sabereletronica.com.br/secoes/leitura/774>>. Acesso em: 21 jun. 2011.

SOUZA, David José de. **Desbravando o PIC : ampliado e atualizado para PIC 16F628A**. 12 ed. São Paulo: Érica, 2007.

TEXAS INSTRUMENTS. **MAX232, MAX232I DUAL EIA-232 drivers/receivers**. Dallas: Texas Instruments, 2002. 9p.

THILMANY, Jean. A Touching Sensation. **Mechanical Engineering Magazine**, nov. 2003. Disponível em: <http://memagazine.asme.org/Articles/2003/November/Touching_Sensation.cfm>. Acesso em: 20 out. 2012.

ZELEK, John S.; BROMLEY Sam; ASMAR Daniel; THOMPSON David. **A haptic glove as a tactile-vision sensory substitution for wayfinding**, Nova Iorque, 2003, p. 621-632, out. 2003.

APÊNDICE A – Comandos do robô RV-M1

Comando	Descrição	Formato
DP (<i>Decrement Position</i>)	Move o robô para posição anterior a atual	DP
DW (<i>Draw</i>)	Move o final da mão para uma posição distante da atual respeitando as distâncias especificadas em cada eixo	DW <DistânciaX>, <DistânciaY>, <DistânciaZ>
HE (<i>Here</i>)	Grava a posição atual do robô (coordenadas) em uma posição numérica	HE <Número da Posição>
HO (<i>Home</i>)	Estabelece uma posição de referência no plano de coordenadas cartesianas	HO
IP (<i>Increment Position</i>)	Move o robô para a posição seguinte a atual	IP
MA (<i>Move Approach</i>)	Move o final da mão para uma posição distante da atual respeitando os incrementos especificados entre duas posições indicadas	MA <Posição número (a)>, <Posição número (b)>, <O ou C>
MC (<i>Move Continuous</i>)	Move o robô continuamente entre dois pontos predefinidos	MC <Posição número (a)>, <Posição número (b)>
MJ (<i>Move Joint</i>)	Movimenta cada junta de acordo com os ângulos especificados a partir da posição atual.	MJ <Ângulo da cintura>, <Ângulo do ombro>, <Ângulo do cotovelo>, <Ângulo do pulso>, <Ângulo de giro>
MO (<i>Move</i>)	Move o final da mão para uma posição especificada	MO<Número da Posição>, <O ou C>
MP (<i>Move Position</i>)	Move o final da mão para uma posição especificada a partir de coordenadas e ângulos	MP <Coordenada X>, <Coordenada Y>, <Coordenada Z>, <Ângulo do pulso>, <Ângulo de giro>
MS (<i>Move Straight</i>)	Move o robô para uma posição especificada com a quantidade de pontos intermediários que devem ser seguidos em linha reta	MS <Número da posição>, <Quantidade de pontos intermediários>, <O ou C> A quantidade de pontos deve ser entre 1 e 99
MT (<i>Move Tool</i>)	Move o final da mão em linha reta a partir da posição atual com uma distancia predefinida	MT <Número da posição>, <Distância>, <C ou O>
NT (<i>Nest</i>)	Retorna o robô para sua posição de origem mecânica	NT
OG (<i>Origin</i>)	Move o robô para a posição de origem do plano cartesiano	OG
PL (<i>Position Load</i>)	Atribui as coordenadas da posição B à posição A.	PL <Número da posição A>, <Número da posição B>

PX (<i>Position Exchange</i>)	Troca as coordenadas da posição B com as da posição A	PX <Número da posição A>, <Número da posição B>
SF (<i>Shift</i>)	Adiciona as coordenadas da posição B às coordenadas da posição A	SF <Posição A>, <Posição B>
SP (<i>Speed</i>)	Define a velocidade, aceleração (H) ou desaceleração (L) na aproximação do ponto de destino	SP <Velocidade>, <H ou L> Onde velocidade varia entre 0 e 9.
TI (<i>Timer</i>)	Para o movimento do robô por um período de tempo determinado	TI <Tempo> Onde tempo pode variar de 0.1 a 32767 segundos
TL (<i>Tool</i>)	Estabelece uma distancia entre o final do braço robótico e a superfície que será trabalhada (tamanho de ferramenta)	TL <Distância> Onde a distância pode variar entre 0 e 300 mm. O padrão é 0.
DL (<i>Delete Line</i>)	Deleta o conteúdo de um determinado intervalo de linhas	DL <Linha A>, <Linha B> Deleta inclusive o conteúdo das linhas A e B.
GC (<i>Grip Close</i>)	Fecha e mantém a garra fechada	GC
GO (<i>Grip Open</i>)	Abre e mantém a garra aberta	GO
ED (<i>End</i>)	Encerra o programa	ED
GT (<i>Go To</i>)	Envia o robô para uma linha específica do programa (não condicional)	GT <Linha do programa>
ER (<i>Error Read</i>)	Lê o estado de erro usando RS-232	ER
LR (<i>Line Read</i>)	Lê a linha especificada do programa	LR <Linha do programa>
PR (<i>Position Read</i>)	Lê as coordenadas de uma posição específica do robô	PR <Número da posição>
WH (<i>Where</i>)	Lê as coordenadas da posição atual do robô	WH
EQ (<i>If Equal</i>)	Compara um valor de um registro interno com um valor especificado e se igual vai para uma linha especificada de programa	EQ <Valor a ser comparado>, <Linha do programa>
NE (<i>If Not Equal</i>)	Compara um valor de um registro interno com um valor especificado e se não igual vai para uma linha especificada de programa	NE <Valor a ser comparado>, <Linha do programa>
RC (<i>Repeat Cycle</i>)	Repete um ciclo (delimitado pelo comando NX) por um número de vezes especificado	RC <Número de repetições>
NX (<i>Next</i>)	Especifica o final do ciclo de repetição iniciado pelo comando RC	NX

RN (<i>Run</i>)	Executa as linhas do programa de linha A até B (ou até ED)	RN <Linha A>, <Linha B>
RS (<i>Reset</i>)	Reseta o programa e as condições de erro	RS
' (<i>Coment</i>)	Adiciona comentário ao programa	' <Comentário>

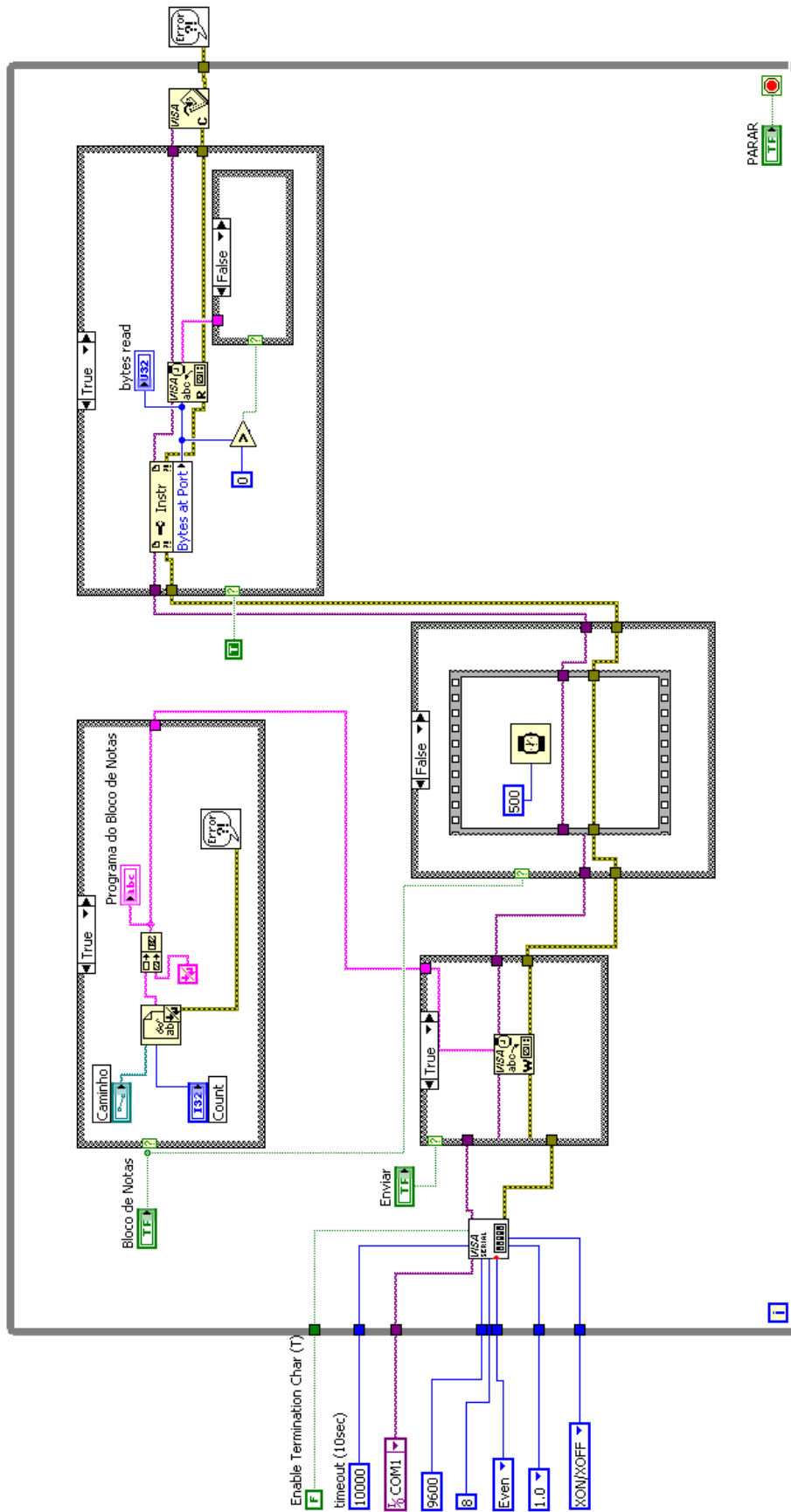
Observações:

- O número da posição deve ser entre 1 e 629;
- Utiliza-se “O” para a garra aberta ou “C” para a garra fechada;
- A linha do programa deve estar entre 1 e 2048;
- O número de repetições deve estar entre 1 e 32767;
- Os valores a serem comparados devem estar entre -32767 (decimal) e +32767 (decimal);

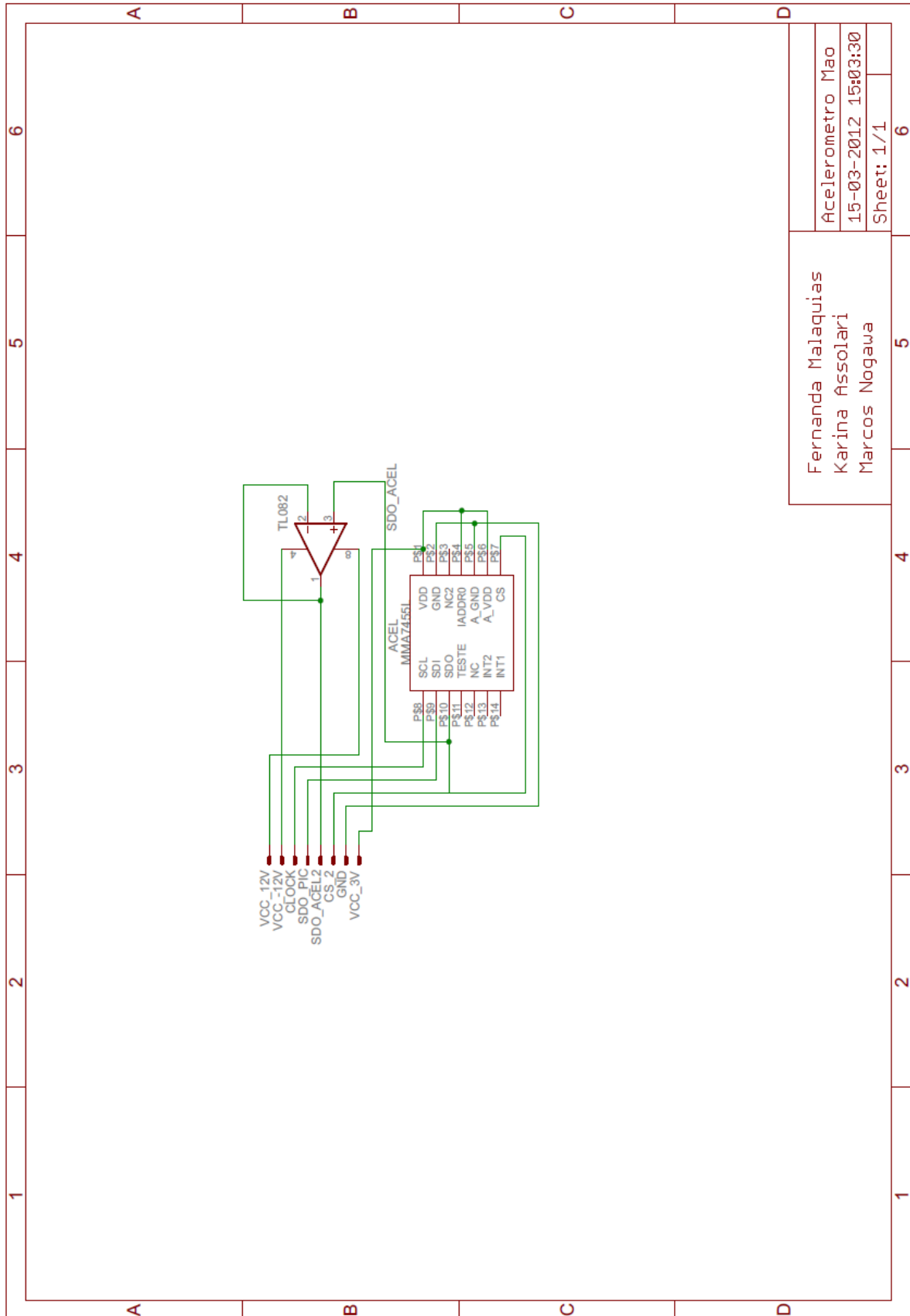
APÊNDICE B – Tabela de pinagem e direção de fluxo de dados na comunicação RS-232

Sigla	Significado	Tradução	Fluxo de Dados
CTS	Clear To Send	Pronto para enviar dados	DCE para DTE
DCD	Data Carrier Detected	Portadora detectada	DCE para DTE
DSR	Data Set Ready	Conjunto de dados pronto	DCE para DTE
DTR	Data Terminal Ready	Terminal de dados pronto	DTE para DCE
RI	Ring Indicator	Indicador de toque/campainha	
RTS	Ready To Send	Pronto para enviar (computador)	DTE para DCE
RxD	Received Data	Recepção de dados	DCE para DTE
GND	Signal Ground	Sinal de terra	
TxD	Transmitted Data	Transmissão de dados	DTE para DTE

APÊNDICE C – Programação do Labview

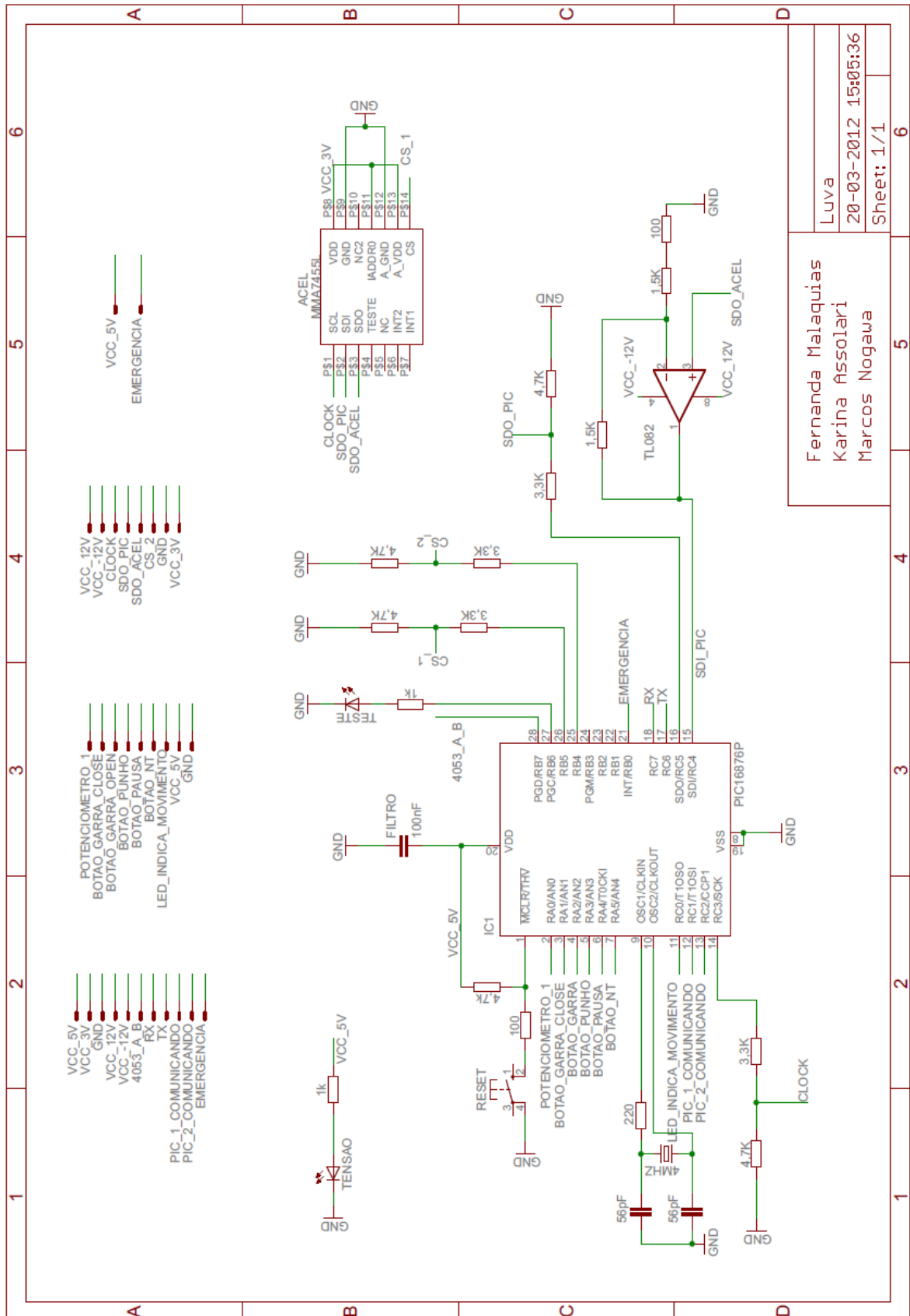


APÊNDICE D – Circuito esquemático da placa acelerômetro mão

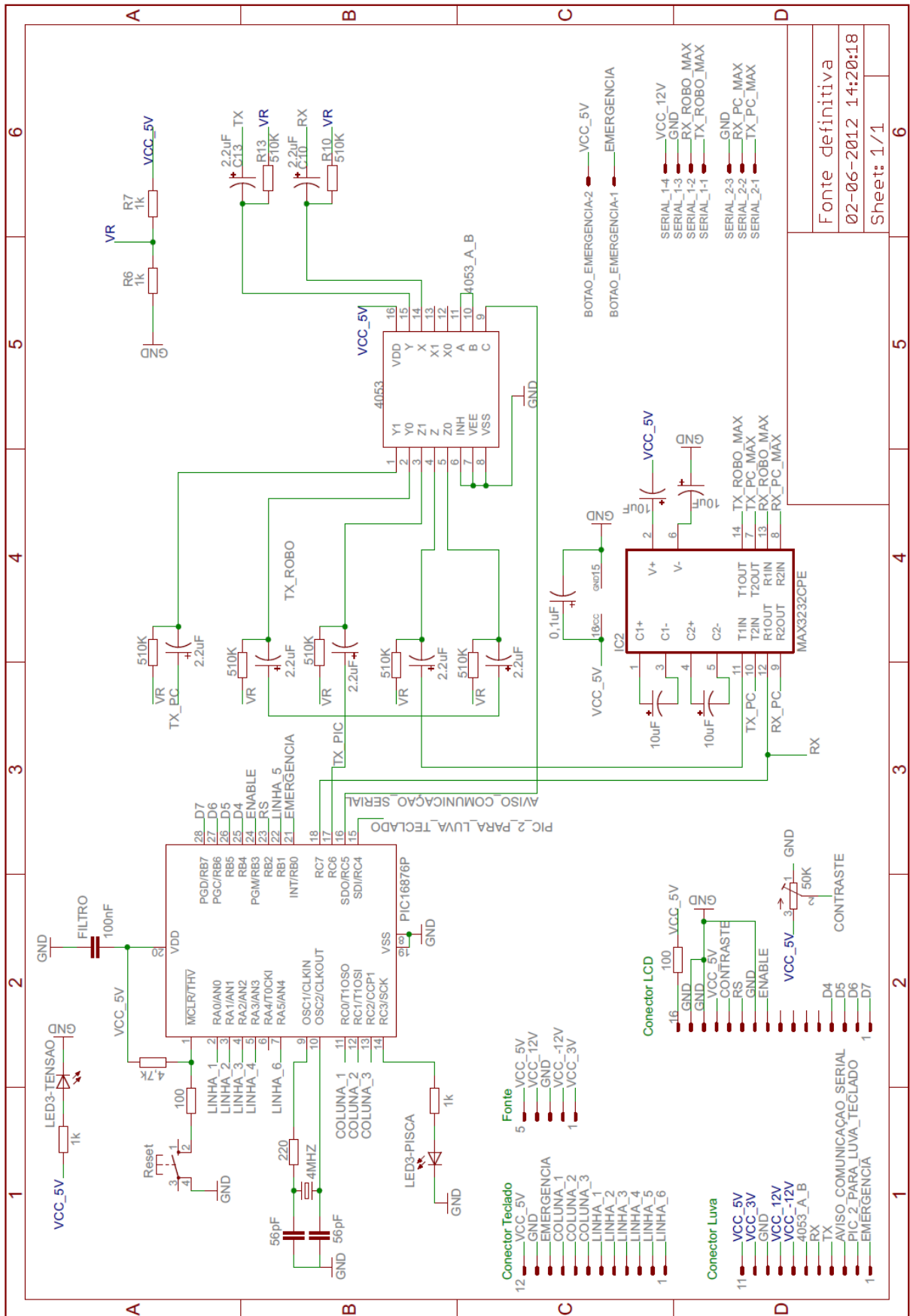


Fernanda Malaquias	Acelerometro Mao
Karina Assolari	15-03-2012 15:03:30
Marcos Nogawa	Sheet: 1/1

APÊNDICE E – Circuito esquemático da placa luva

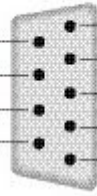
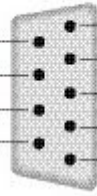
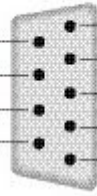
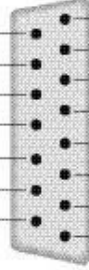
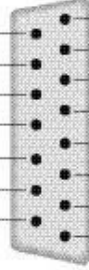
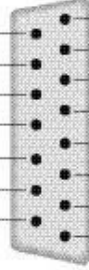
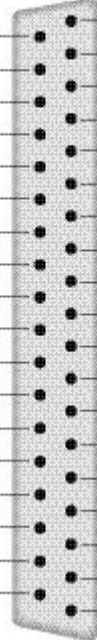
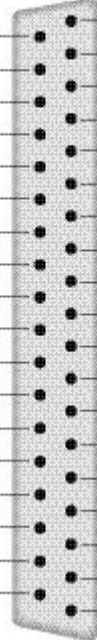
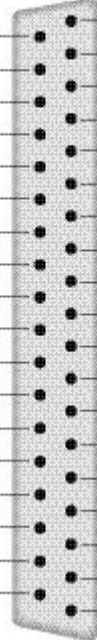


APÊNDICE F – Circuito esquemático da placa fonte

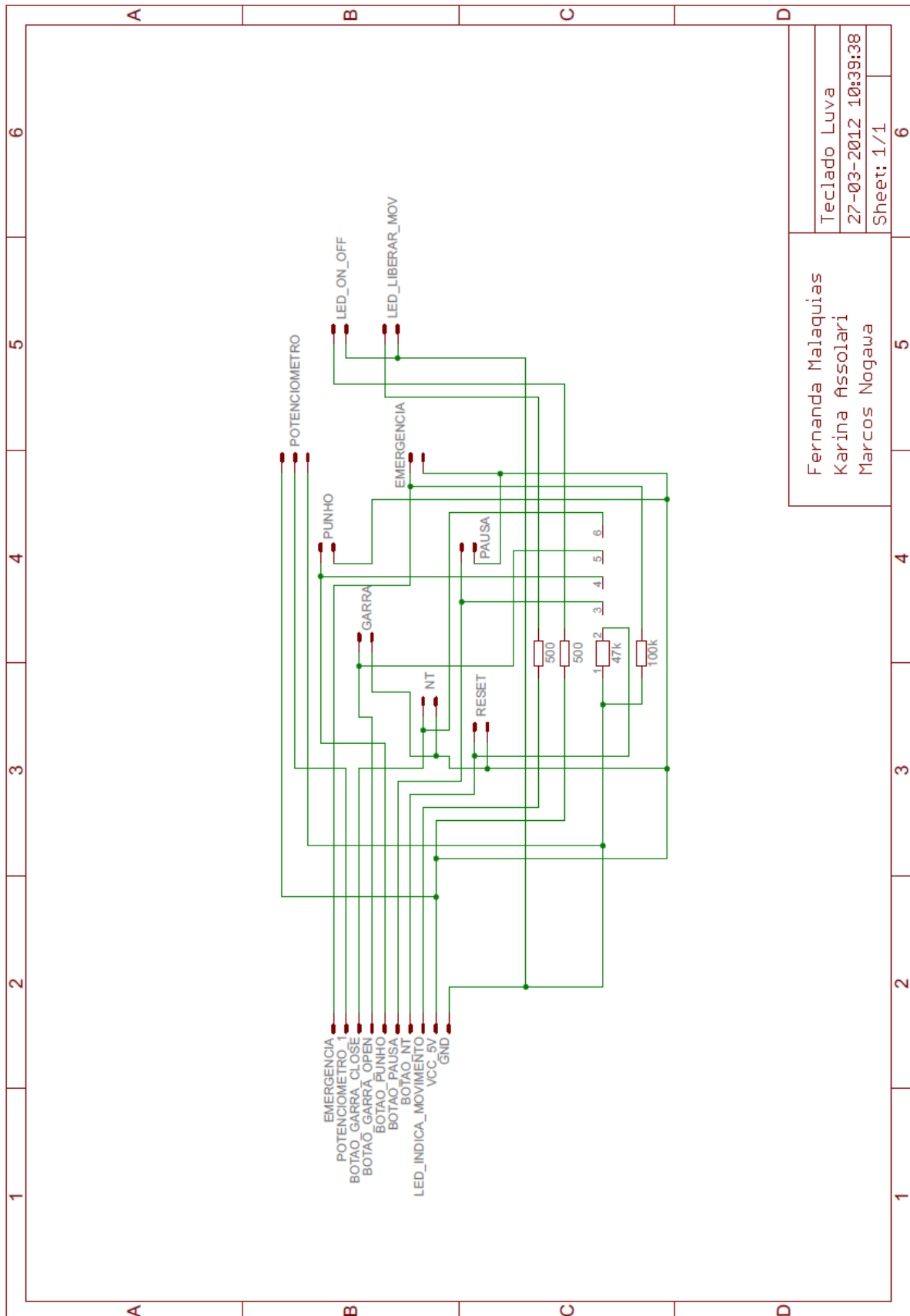


Fonte definitiva
 02-06-2012 14:20:18
 Sheet: 1/1

APÊNDICE G – Pinagem dos conectores

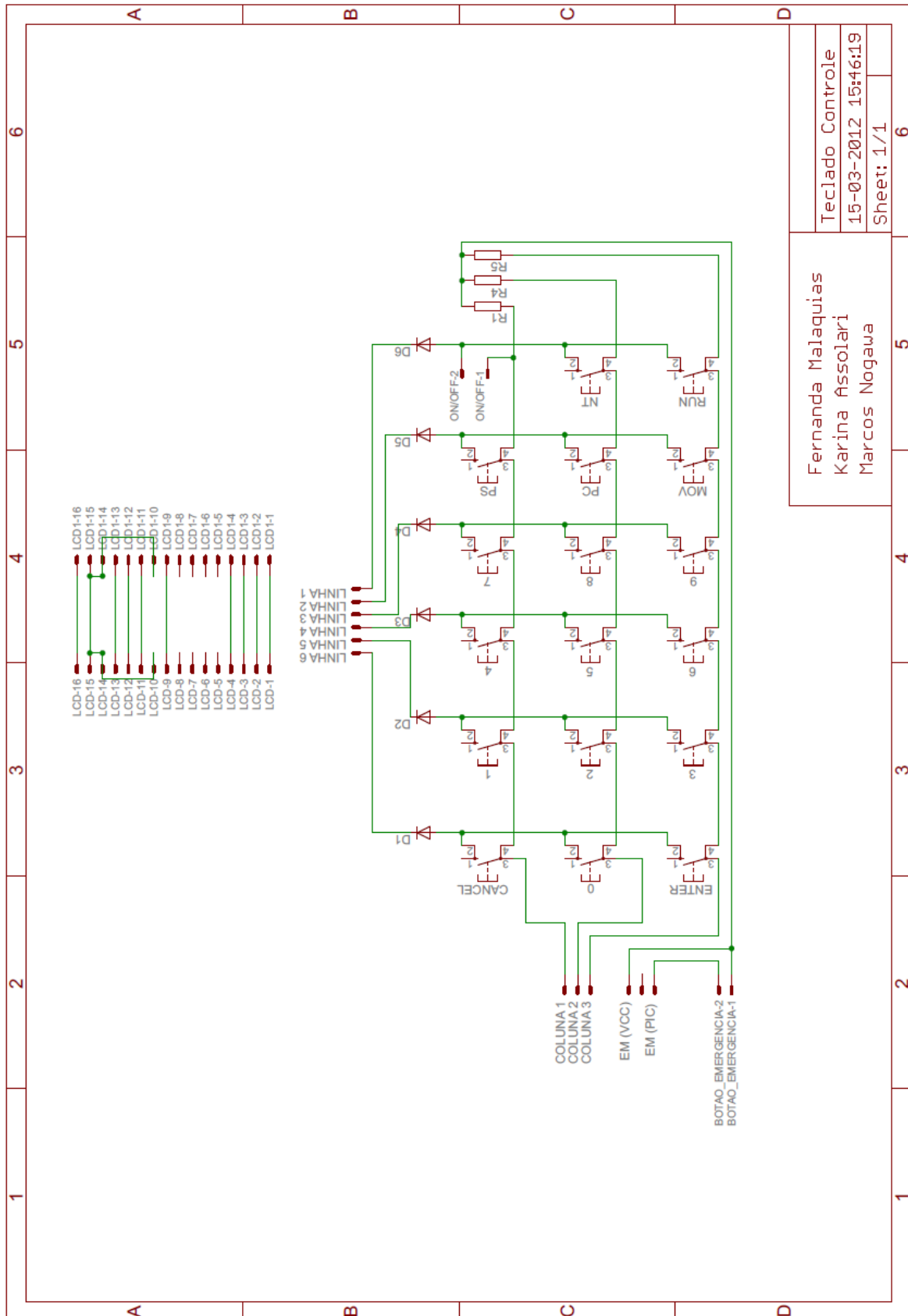
DB9	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">FUNÇÃO</th> <th style="width: 10%;">Nº dos Pinos</th> <th style="width: 10%;"></th> <th style="width: 10%;">Nº dos Pinos</th> <th style="width: 20%;">FUNÇÃO</th> </tr> </thead> <tbody> <tr> <td>NC</td> <td>6</td> <td rowspan="5" style="text-align: center;">  </td> <td>1</td> <td>NC</td> </tr> <tr> <td>NC</td> <td>7</td> <td>2</td> <td>TX</td> </tr> <tr> <td>NC</td> <td>8</td> <td>3</td> <td>RX</td> </tr> <tr> <td>NC</td> <td>9</td> <td>4</td> <td>NC</td> </tr> <tr> <td>NC</td> <td></td> <td>5</td> <td>GND</td> </tr> </tbody> </table>	FUNÇÃO	Nº dos Pinos		Nº dos Pinos	FUNÇÃO	NC	6		1	NC	NC	7	2	TX	NC	8	3	RX	NC	9	4	NC	NC		5	GND																																																								
FUNÇÃO	Nº dos Pinos		Nº dos Pinos	FUNÇÃO																																																																															
NC	6		1	NC																																																																															
NC	7		2	TX																																																																															
NC	8		3	RX																																																																															
NC	9		4	NC																																																																															
NC			5	GND																																																																															
DB15	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">FUNÇÃO</th> <th style="width: 10%;">Nº dos Pinos</th> <th style="width: 10%;"></th> <th style="width: 10%;">Nº dos Pinos</th> <th style="width: 20%;">FUNÇÃO</th> </tr> </thead> <tbody> <tr> <td>AVISO DE COMUNICAÇÃO</td> <td>9</td> <td rowspan="8" style="text-align: center;">  </td> <td>1</td> <td>VCC 5V</td> </tr> <tr> <td>PIC_2_PARA_LUVA_TECLADO</td> <td>10</td> <td>2</td> <td>VCC 3V</td> </tr> <tr> <td>EMERGÊNCIA</td> <td>11</td> <td>3</td> <td>GND</td> </tr> <tr> <td></td> <td>12</td> <td>4</td> <td>VCC 12V</td> </tr> <tr> <td></td> <td>13</td> <td>5</td> <td>VCC -12V</td> </tr> <tr> <td></td> <td>14</td> <td>6</td> <td>SELEÇÃO 4053</td> </tr> <tr> <td></td> <td>15</td> <td>7</td> <td>RX</td> </tr> <tr> <td></td> <td></td> <td>8</td> <td>TX</td> </tr> </tbody> </table>	FUNÇÃO	Nº dos Pinos		Nº dos Pinos	FUNÇÃO	AVISO DE COMUNICAÇÃO	9		1	VCC 5V	PIC_2_PARA_LUVA_TECLADO	10	2	VCC 3V	EMERGÊNCIA	11	3	GND		12	4	VCC 12V		13	5	VCC -12V		14	6	SELEÇÃO 4053		15	7	RX			8	TX																																												
FUNÇÃO	Nº dos Pinos		Nº dos Pinos	FUNÇÃO																																																																															
AVISO DE COMUNICAÇÃO	9		1	VCC 5V																																																																															
PIC_2_PARA_LUVA_TECLADO	10		2	VCC 3V																																																																															
EMERGÊNCIA	11		3	GND																																																																															
	12		4	VCC 12V																																																																															
	13		5	VCC -12V																																																																															
	14		6	SELEÇÃO 4053																																																																															
	15		7	RX																																																																															
			8	TX																																																																															
DB37	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">FUNÇÃO TECLADO</th> <th style="width: 10%;">NÚMERO DE PINOS</th> <th style="width: 10%;"></th> <th style="width: 10%;">NÚMERO DE PINOS</th> <th style="width: 20%;">FUNÇÃO LCD</th> </tr> </thead> <tbody> <tr> <td>VCC 5V</td> <td>20</td> <td rowspan="19" style="text-align: center;">  </td> <td>1</td> <td>VCC 5V</td> </tr> <tr> <td>GND</td> <td>21</td> <td>2</td> <td>GND</td> </tr> <tr> <td>EMERGÊNCIA</td> <td>22</td> <td>3</td> <td>GND</td> </tr> <tr> <td>COLONA 1</td> <td>23</td> <td>4</td> <td>VCC 5V</td> </tr> <tr> <td>COLONA 2</td> <td>24</td> <td>5</td> <td>CONTRASTE</td> </tr> <tr> <td>COLONA 3</td> <td>25</td> <td>6</td> <td>RS</td> </tr> <tr> <td>LINHA 1</td> <td>26</td> <td>7</td> <td>GND</td> </tr> <tr> <td>LINHA 2</td> <td>27</td> <td>8</td> <td>EMERGÊNCIA</td> </tr> <tr> <td>LINHA 3</td> <td>28</td> <td>9</td> <td>NC</td> </tr> <tr> <td>LINHA 4</td> <td>29</td> <td>10</td> <td>NC</td> </tr> <tr> <td>LINHA 5</td> <td>30</td> <td>11</td> <td>NC</td> </tr> <tr> <td>LINHA 6</td> <td>31</td> <td>12</td> <td>NC</td> </tr> <tr> <td>NC</td> <td>32</td> <td>13</td> <td>D4</td> </tr> <tr> <td>NC</td> <td>33</td> <td>14</td> <td>D5</td> </tr> <tr> <td>NC</td> <td>34</td> <td>15</td> <td>D6</td> </tr> <tr> <td>NC</td> <td>35</td> <td>16</td> <td>D7</td> </tr> <tr> <td>NC</td> <td>36</td> <td>17</td> <td>NC</td> </tr> <tr> <td>NC</td> <td>37</td> <td>18</td> <td>NC</td> </tr> <tr> <td>NC</td> <td></td> <td>19</td> <td>NC</td> </tr> </tbody> </table>	FUNÇÃO TECLADO	NÚMERO DE PINOS		NÚMERO DE PINOS	FUNÇÃO LCD	VCC 5V	20		1	VCC 5V	GND	21	2	GND	EMERGÊNCIA	22	3	GND	COLONA 1	23	4	VCC 5V	COLONA 2	24	5	CONTRASTE	COLONA 3	25	6	RS	LINHA 1	26	7	GND	LINHA 2	27	8	EMERGÊNCIA	LINHA 3	28	9	NC	LINHA 4	29	10	NC	LINHA 5	30	11	NC	LINHA 6	31	12	NC	NC	32	13	D4	NC	33	14	D5	NC	34	15	D6	NC	35	16	D7	NC	36	17	NC	NC	37	18	NC	NC		19	NC
FUNÇÃO TECLADO	NÚMERO DE PINOS		NÚMERO DE PINOS	FUNÇÃO LCD																																																																															
VCC 5V	20		1	VCC 5V																																																																															
GND	21		2	GND																																																																															
EMERGÊNCIA	22		3	GND																																																																															
COLONA 1	23		4	VCC 5V																																																																															
COLONA 2	24		5	CONTRASTE																																																																															
COLONA 3	25		6	RS																																																																															
LINHA 1	26		7	GND																																																																															
LINHA 2	27		8	EMERGÊNCIA																																																																															
LINHA 3	28		9	NC																																																																															
LINHA 4	29		10	NC																																																																															
LINHA 5	30		11	NC																																																																															
LINHA 6	31		12	NC																																																																															
NC	32		13	D4																																																																															
NC	33		14	D5																																																																															
NC	34		15	D6																																																																															
NC	35		16	D7																																																																															
NC	36		17	NC																																																																															
NC	37		18	NC																																																																															
NC			19	NC																																																																															

APÊNDICE H – Circuito esquemático da placa teclado luva



Fernanda Malaquias		Teclado Luva	
Karina Assolari		27-03-2012 10:39:38	
Marcos Nogueira		Sheet: 1/1	

APÊNDICE I – Circuito esquemático da placa teclado controle

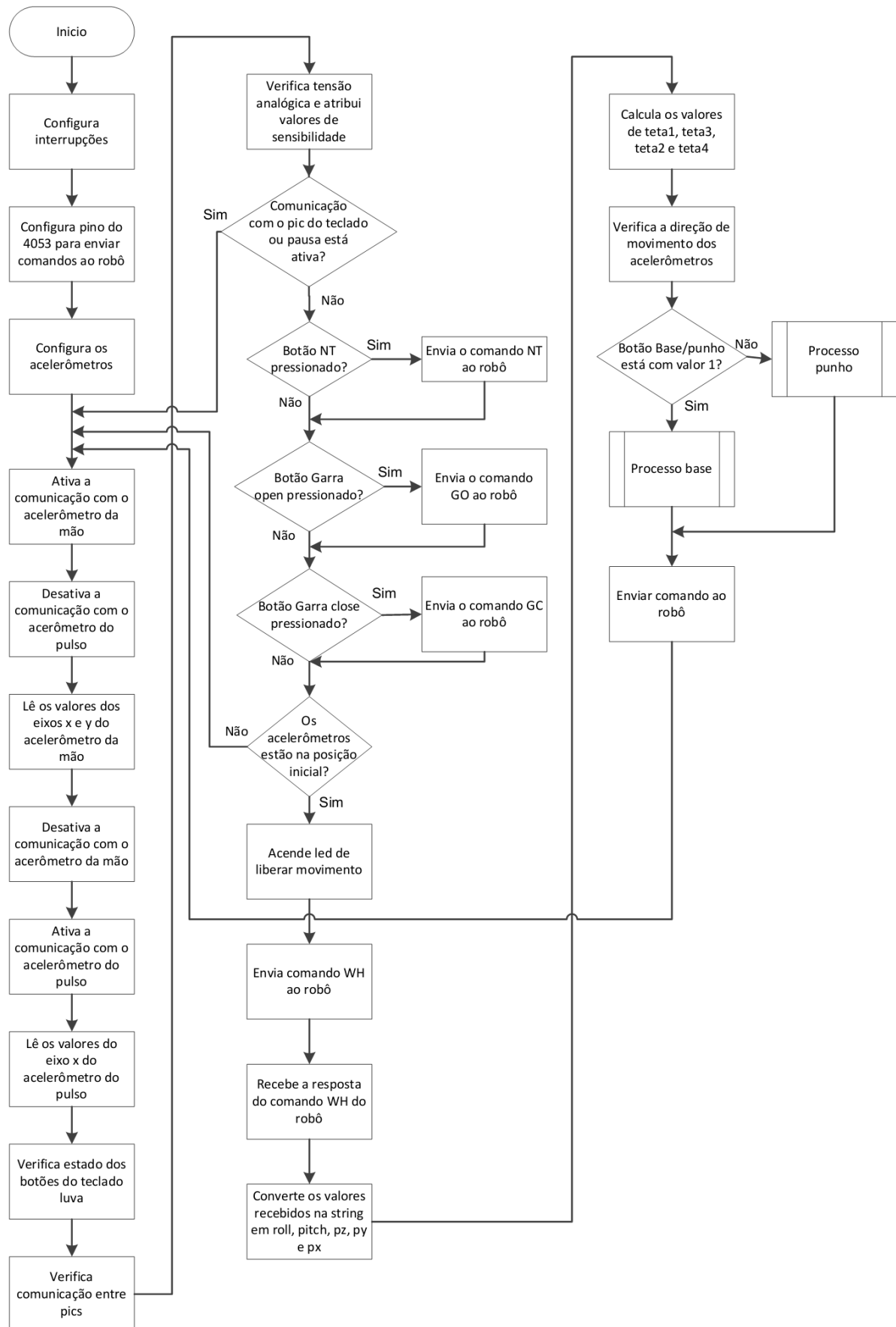


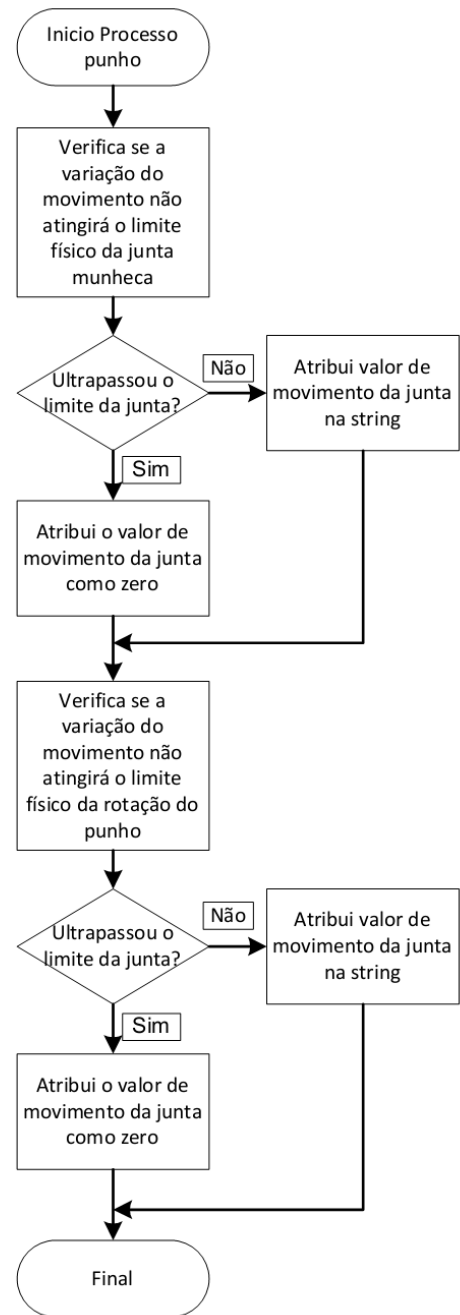
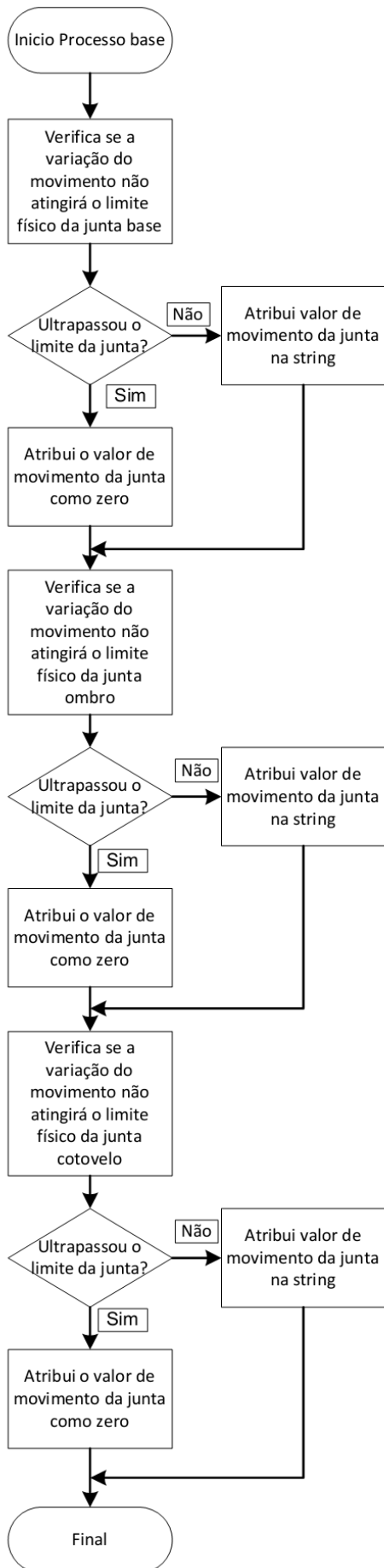
Fernanda Malaquias		Teclado Controle	
Karina Assolari		15-03-2012 15:46:19	
Marcos Nagawa		Sheet: 1/1	

**APÊNDICE J – Configurações do acelerômetro MMA7455L para utilização com o
microcontrolador PIC18F252**

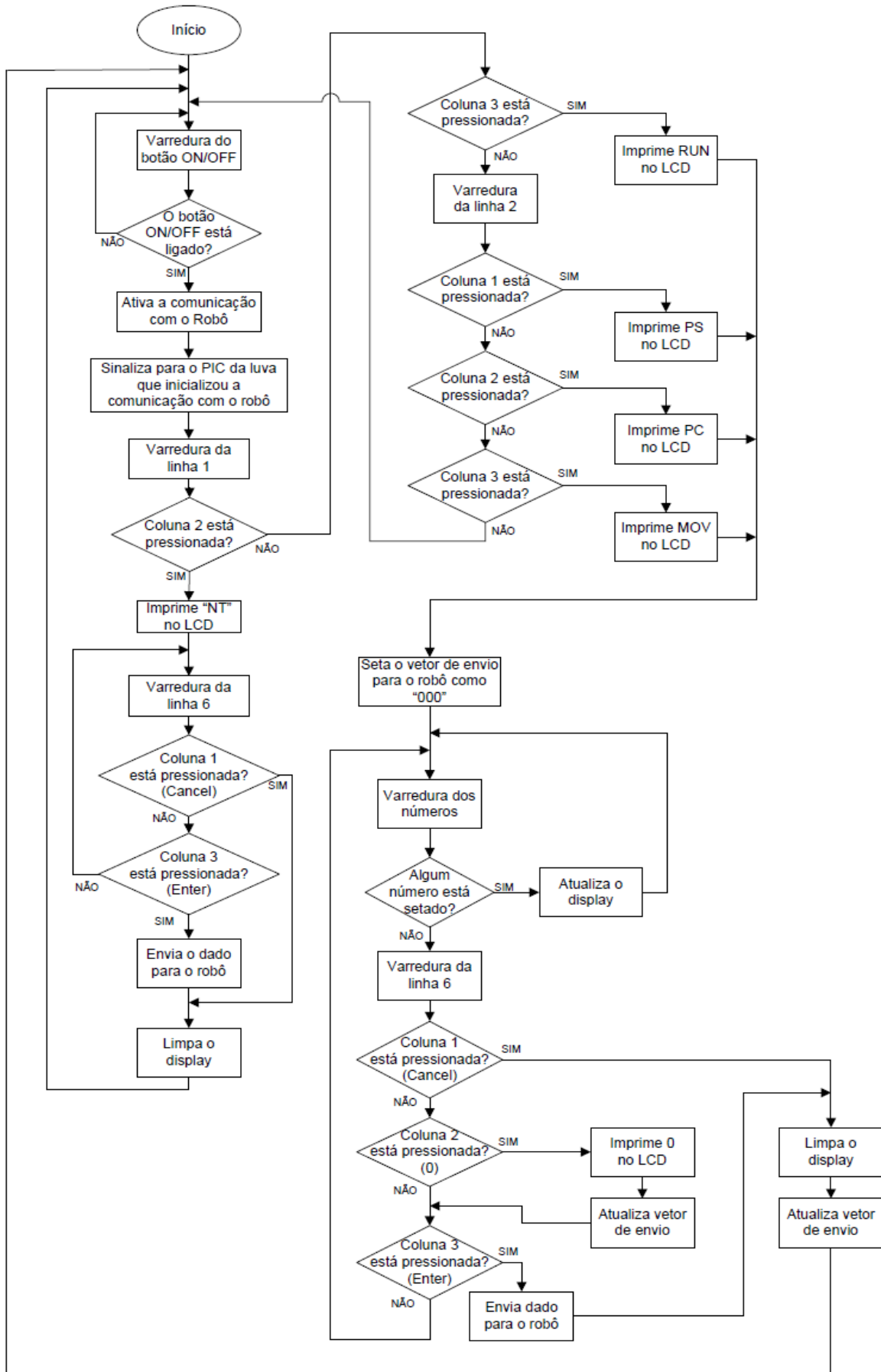
Registro	Nome	Definição
\$16	MCTL	Modo de controle (sensibilidade e modo medição)
\$0D	I2CAD	Endereço do protocolo de comunicação I2C (Usado para desabilitar a I2C)
\$10	XOFFL	<i>Offset</i> do eixo X (bit menos significativo)
\$12	YOFFL	<i>Offset</i> do eixo Y (bit menos significativo)
\$14	ZOFFL	<i>Offset</i> do eixo Z (bit menos significativo)

APÊNDICE K – Fluxograma de programação da luva





APÊNDICE L – Fluxograma de programação do teclado controle



Fluxograma de varredura dos números do teclado controle

