

UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS  
DEPTO. DE ENGENHARIA ELÉTRICA E DE  
COMPUTAÇÃO

**Trabalho de Conclusão de Curso**

**Sistema de Aferição de Chute para Futebol de  
Robôs da Categoria RoboCup Small Size League**

**Autor:** Vitor Hugo de Carvalho Gomes, n<sup>o</sup>. USP 8957134

**Orientador:** Prof. Dr. Evandro Luis Linhari Rodrigues

São Carlos

2018

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,  
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS  
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da  
EESC/USP com os dados inseridos pelo(a) autor(a).

G633s            Gomes, Vitor Hugo de Carvalho  
                  Sistema de Aferição de Chute para Futebol de Robôs  
da Categoria RoboCup Small Size Soccer / Vitor Hugo de  
Carvalho Gomes; orientador Evandro Luis Linhari  
Rodrigues. São Carlos, 2018.

Monografia (Graduação em Engenharia Elétrica com  
ênfase em Eletrônica) -- Escola de Engenharia de São  
Carlos da Universidade de São Paulo, 2018.

1. Sistema de Aferição. 2. Medição de Chute. 3.  
Robô Rocup Small Size League. 4. Microcontrolador. 5.  
Fotossensor. I. Título.

# FOLHA DE APROVAÇÃO

Nome: Vitor Hugo de Carvalho Gomes

Título: "Sistema de calibração de chute para futebol de robôs da categoria IEEE Small Size Soccer"

Trabalho de Conclusão de Curso defendido e aprovado  
em 27/11/2018,

com NOTA 9,0 (NOVE, ZERO), pela Comissão Julgadora:

*Prof. Associado Evandro Luís Linhari Rodrigues - Orientador - SEL/EESC/USP*

*Prof. Associado Ivan Nunes da Silva - SEL/EESC/USP*

*Mestre Rafael Guedes Lang - Doutorando - SEL/EESC/USP*

Coordenador da CoC-Engenharia Elétrica - EESC/USP:  
Prof. Associado Rogério Andrade Flauzino



# Agradecimentos

Agradeço imensamente ao Professor Evandro Luis Linhari Rodrigues por toda confiança depositada e aos professores e funcionários da Universidade de São Paulo do campus de São Carlos.

À minha mãe Sueli Catarina de Carvalho pois sem ela eu não teria oportunidade de começar esse trabalho e ao meu irmão Igor de Carvalho Gomes por estar sempre presente.

Aos meus grandes amigos por terem me apoiado durante toda a graduação, me ajudando a concluir meus projetos e ambições. Em particular aos irmãos que a vida me presenteou, William Yonamine, Matheus Ito, Rafael Ribeiro, Ivo Dutra, Ana Laís, Murilo Furlan, João Victor Montanha, Paula Christiny, Arthur Demarchi, Guilherme Segato, Caio Carrasco, Victor Boppré, Luiz Miyazaka, Bruno Bagarini, Gustavo Sandrin, Rafael Albuquerque, Gabriel Ginja, Karina Ono e Akina Yoshioka .

Ao *Warthog Robotics*, e a todos amigos de equipe que fiz durante minha participação no grupo, ao qual foram essenciais para a conclusão deste projeto.



# Resumo

Este documento tem como proposta apresentar uma alternativa de ferramenta para a aferição dos parâmetros de velocidade e aceleração proporcionados no chute de robôs de futebol, feito através da marcação da posição da bola disparada em diferentes instantes por meio de fotossensores em comunicação com um microcontrolador, o qual realiza todo o controle de lógica de leitura de sistema e os cálculos necessários, os resultados obtidos são disponibilizados em um LCD 16x2. Apresenta-se também o desenvolvimento do *firmware* no próprio microcontrolador que permite ao usuário funções como menu de seleção, visualizar os dados de leitura obtidos, guardar os dados desejados no próprio sistema para consultas posteriores e uma interface visual agradável e intuitiva desenvolvida exclusivamente para este projeto.

Palavras-Chave: Sistema de Aferição, Medição de Chute, Robô RoboCup Small Size League, Microcontrolador, Fotossensor.





# Abstract

This paper proposes an alternative tool to measure the parameters of speed and acceleration provided by the kick of soccer robots, achieved through the marking of the shot ball's position in different moments using photosensors in communication with a microcontroller, which controls all the logic of the measuring system and the necessary calculation, the results obtained are available in a 16x2 LCD. It is also presented the firmware developed on the microcontroller itself, allowing the user functions such as menu selection, measured data visualization, desired data saving on the system itself for future consultation and a pleasant and intuitive visual interface developed exclusively to this project.

Keywords: Measuring System, Kick Measuring, RoboCup Small Size League Robot, Microcontroller, Photosensor.



# Lista de Figuras

2.1	<i>Esquema de jogo RoboCup Small Size League.</i> Fonte: Fornecida pela RoboCup. . . .	19
2.2	<i>Símbolo do fotodiodo.</i> Fonte: Elaborada pelo autor. . . . .	21
2.3	<i>Símbolo do fototransistor.</i> Fonte: Elaborada pelo autor. . . . .	22
2.4	<i>Diagrama em blocos de construção básica de um microcontrolador.</i> Fonte: Vandertronic. . . . .	23
2.5	<i>Módulo de display LCD comumente usado em instrumentações simples.</i> Fonte: Elaborada pelo autor. . . . .	24
2.6	<i>Exemplo de aplicação de display LCD, usado no iPhone 7.</i> Fonte: Apple Inc. . . . .	25
2.7	<i>Estrutura básica de funcionamento de temporizadores e contadores.</i> Fonte: Elaborada pelo autor. . . . .	26
3.1	<i>Diagrama da lógica de sensoramento do sistema sem a presença da bola.</i> Fonte: Elaborada pelo autor. . . . .	33
3.2	<i>Diagrama da lógica de sensoramento do sistema com a presença da bola.</i> Fonte: Elaborada pelo autor. . . . .	34
3.3	<i>Esboço da armação mecânica elaborado para o projeto.</i> Fonte: Elaborada pelo autor.	35
3.4	<i>Aletas instaladas nas laterais da estrutura.</i> Fonte: Elaborada pelo autor. . . . .	36
3.5	<i>Fixação da caixa do PCB do túnel de acrílico.</i> Fonte: Elaborada pelo autor. . . . .	37
3.6	<i>PCB soldada - vista por cima.</i> Fonte: Elaborada pelo autor. . . . .	39
3.7	<i>PCB soldada por baixo.</i> Fonte: Elaborada pelo autor. . . . .	39
3.8	<i>PCB vista de perfil.</i> Fonte: Elaborada pelo autor. . . . .	40
3.9	<i>Projeto após integração de mecânica com a eletrônica.</i> Fonte: Elaborada pelo autor.	41
3.10	<i>Primeira mensagem de inicialização.</i> Fonte: Elaborada pelo autor. . . . .	47
3.11	<i>Segunda mensagem de inicialização.</i> Fonte: Elaborada pelo autor. . . . .	48
3.12	<i>Tempo para configuração do microcontrolador e do LCD.</i> Fonte: Elaborada pelo autor.	48
3.13	<i>Mensagem para indicação de disponibilidade de uso.</i> Fonte: Elaborada pelo autor. .	49

4.1	<i>Configuração geral dos testes de robustez mecânica.</i> Fonte: Elaborada pelo autor. . . . .	51
4.2	<i>Bocal de fototransistor danificado pelo chute.</i> Fonte: Elaborada pelo autor. . . . .	52
4.3	<i>Leitura em ambiente com iluminação fraca.</i> Fonte: Elaborada pelo autor. . . . .	53
4.4	<i>Leitura em ambiente fortemente iluminado.</i> Fonte: Elaborada pelo autor. . . . .	53
4.5	<i>Menor velocidade obtida com chutes de um robô SSL.</i> Fonte: Elaborada pelo autor. . . . .	55
4.6	<i>Maior velocidade obtida com chutes de um robô SSL.</i> Fonte: Elaborada pelo autor. . . . .	56
4.7	<i>Lógica de firmware implementado.</i> Fonte: Elaborada pelo autor. . . . .	57
A.1	<i>Schematic desenvolvido no CADSoft EAGLE.</i> Fonte: Elaborada pelo autor. . . . .	63
B.1	<i>Board desenvolvido no CADSoft EAGLE da AUTODESK.</i> Fonte: Elaborada pelo autor. . . . .	65
B.2	<i>Board desenvolvido no CADSoft EAGLE da AUTODESK.</i> Fonte: Elaborada pelo autor. . . . .	66
B.3	<i>Board desenvolvido no CADSoft EAGLE da AUTODESK.</i> Fonte: Elaborada pelo autor. . . . .	67

# Siglas

**LCD** - *Liquid-crystal Display*

**PCB** - *Printed Circuit Board*

**IA** - *Inteligência Artificial*

**FIFA** - *Fédération Internationale de Football Association*

**SSL** - *Small Size League*

**CD** - *Compact Disk*

**LED** - *Light Emitting Diode*

**RAM** - *Random Access Memory*

**I/O** - *Input/Output*

**CPU** - *Central Processing Unit*

**CRT** - *Cathode Ray Tube*

**DVD** - *Digital Video Disc*

**ISR** - *Interrupt Service Routine*

**PIC** - *Programmable Interface Controller*

**ICSP** - *In-circuit Serial Programming*

**USB** - *Universal Serial Bus*

**PLL** - *Phase-locked Loop*

**A/D** - *Analog/Digital*

**CI** - *Circuito Integrado*

**CAD** - *Computer Aided Design*



# Sumário

<b>1</b>	<b>Introdução</b>	<b>15</b>
1.1	Motivação . . . . .	16
1.2	Objetivo . . . . .	16
1.3	Justificativa . . . . .	16
1.4	Organização do Trabalho . . . . .	16
<b>2</b>	<b>Fundamentação Teórica</b>	<b>19</b>
2.1	Futebol de robôs . . . . .	19
2.1.1	Visão Computacional . . . . .	20
2.1.2	Inteligência Artificial . . . . .	20
2.2	Conceitos físicos . . . . .	20
2.2.1	Especificação de velocidade inicial . . . . .	20
2.2.2	Especificação de aceleração . . . . .	20
2.3	Hardware . . . . .	21
2.3.1	Fotossensores . . . . .	21
2.3.2	Microcontrolador . . . . .	22
2.3.3	<i>Liquid-crystal Display</i> . . . . .	23
2.4	Firmware . . . . .	25
2.4.1	Desenvolvimento de firmware embarcado . . . . .	25
2.4.2	Timer e Counters . . . . .	25
2.4.3	Interrupção . . . . .	27
<b>3</b>	<b>Desenvolvimento do Projeto</b>	<b>29</b>
3.1	Material . . . . .	29
3.1.1	Túnel de Acrílico . . . . .	31
3.1.2	Microcontrolador <i>PIC18F4550</i> . . . . .	31
3.1.3	<i>LCD TUXGR_20x2</i> . . . . .	32

3.1.4	Fototransistor <i>TIL78</i> e <i>LED</i> infravermelho <i>TSAL6200</i> . . . . .	32
3.1.5	<i>CADSoft Eagle</i> . . . . .	32
3.1.6	<i>MPLAB X IDE 1.35</i> . . . . .	33
3.1.7	<i>PICKit 3</i> . . . . .	33
3.2	Métodos . . . . .	33
3.2.1	Montagem do Túnel de Acrílico . . . . .	34
3.2.2	Elaboração do PCB ( <i>TunnelBoard</i> ) . . . . .	36
3.2.3	Confecção do PCB ( <i>TunnelBoard</i> ) . . . . .	38
3.2.4	Integração Mecânica e Eletrônica . . . . .	39
3.2.5	Desenvolvimento de <i>Firmware</i> . . . . .	40
<b>4</b>	<b>Resultados e Discussões</b>	<b>51</b>
4.1	Mecânica . . . . .	51
4.1.1	Robustez . . . . .	52
4.1.2	Peso . . . . .	53
4.2	Eletrônica . . . . .	54
4.2.1	<i>PCB</i> . . . . .	54
4.2.2	Fotossensores . . . . .	54
4.2.3	Fiação . . . . .	55
4.3	<i>Firmware</i> . . . . .	56
4.3.1	Inicialização . . . . .	56
4.3.2	Tratamento das leituras . . . . .	56
4.3.3	Representação dos dados . . . . .	57
<b>5</b>	<b>Conclusão</b>	<b>59</b>
	<b>Referências</b>	<b>60</b>
<b>A</b>	<b><i>Schematic do PCB TunnelBoard</i></b>	<b>63</b>
<b>B</b>	<b><i>Layout do PCB TunnelBoard</i></b>	<b>65</b>
<b>C</b>	<b><i>Firmware desenvolvido para a TunnelBoard</i></b>	<b>69</b>
	<b>Anexos</b>	<b>69</b>



# Capítulo 1

## Introdução

Uma das formas mais convenientes para o avanço nos campos de robótica e inteligência artificial é através de competições amistosas, onde montar uma equipe de sucesso com *designs* inteligentes, implementação e integração de diversos sub-componentes de *hardware* e *software* em um robô funcional e robusto torna-se um grande desafio e motivação para a educação assim como a pesquisa. Em uma destas instâncias existe o torneio de robôs que jogam futebol, modalidade difundida internacionalmente nas universidades ligadas à engenharia, incluindo instituições brasileiras como a *Universidade de São Paulo*, que formam times competitivos com o intuito de desenvolver robôs para a pesquisa e extensão.

A primeira concepção concreta de robôs jogadores de futebol foi de *Alan Mackworth*, elaborada em 1993 quando publicada no artigo *On Seeing Robots*, posteriormente incluso no livro *Computer Vision: System, Theory, and Applications*. Simultaneamente ocorria um *workshop* em Tóquio sobre os grandes desafios em *IA*, esse evento levantou as discussões sobre o uso do futebol como modo de motivação ao desenvolvimento de tecnologia. A partir destas iniciativas, foi criada a *Robocup Federation*, uma organização responsável por viabilizar pesquisas nas áreas de robótica e inteligência artificial através da proposta de plataformas fundamentadas em problemas do mundo real com potencial de atrair o público em geral.

Na categoria *Small Size League*, a capacidade de chute do robô é um dos maiores diferenciais para a equipe em campo, permitindo passes para jogadores próximos até chutes a gol, além disso os campeonatos desta modalidade possuem regras quanto a velocidade máxima de um chute, denotando uma necessidade do controle de intensidade do impulso de bola. Aliado ao fato de durante o curso de engenharia elétrica o aluno ser incentivado a desenvolver e melhorar dispositivos e processos através de problemas motivacionais, torna-se evidente a importância destes pois, além de se apresentarem como problemas práticos, são realmente encorajadores, instigando o aluno a procurar e desenvolver

habilidades multidisciplinares. Este projeto apresenta uma solução de um desses problemas.

Para estas competições, existe o seguinte estado da arte:

*"Em meados de século 21, uma equipe totalmente autônoma de robôs humanoides jogadores de futebol deve vencer um jogo contra o time de humanos campeão da última Copa do Mundo da FIFA, utilizando as regras da FIFA."*

*-RoboCup Federation*

## 1.1 Motivação

Tem-se como motivação deste projeto o desenvolvimento de uma ferramenta portátil e simples para a medição de velocidade e aceleração de bolas de golfe via fotossensores que disponibilize os parâmetros aferidos ao usuário como proposta de aferição do chute de um robô *SSL*.

## 1.2 Objetivo

O objetivo desta monografia consiste na concepção de um sistema capaz de extrair os parâmetros de velocidade e aceleração de uma bola de golfe através de fotossensores e, utilizando um microcontrolador com *firmware* apropriado, dispor os dados coletados em um *LCD* onde estão tratados e conformados para disponibilidade do usuário.

## 1.3 Justificativa

Geralmente a aferição do chute de um *SSL* envolve o sistema de câmeras usado nos jogos, onde verifica-se a posição da bola ao longo do tempo e determina-se sua velocidade. Este método é significativamente laborioso pois a montagem da configuração necessária requer estruturas com de 4 metros de altura para a instalação de 4 câmeras que fazem a cobertura do campo, além disso a medida é consideravelmente imprecisa, devido as distorções da visão causadas pela lente das câmeras. Outro ponto, é que a intensidade dos chutes é assunto de grande disputa entre equipes durante a competição do *SSL* pois chutes com velocidade maior do que 8m/s são proibidos no regulamento, e é bastante comum jogos paralisarem por longos períodos por acusações dúbias de infrações no chute do adversário.

## 1.4 Organização do Trabalho

Esta monografia está organizada conforme segue: Capítulo 2 trata da fundamentação teórica do projeto, explicando conceitos do *hardware* e *firmware* envolvidos utilizados no Capítulo 3, o qual apre-

senta detalhadamente o material e os métodos adotados para sua execução. O Capítulo 4 mostra e discute os resultados obtidos com o projeto final, concluindo-se o trabalho com o Capítulo 5, expondo as considerações finais acerca do desenvolvimento do projeto, além de apontar os trabalhos futuros que podem melhorar o resultado final.



## Capítulo 2

# Fundamentação Teórica

Neste capítulo são apresentadas as conceituações necessárias para o bom andamento do projeto obtidas em literatura relacionada. De forma sintetizada, são abordados os embasamentos relacionados a contextualização de futebol de robôs, aos conceitos físicos envolvidos, ao *hardware* do projeto associado as componentes eletrônicas que deseja-se implementar, além de tratar das concepções de desenvolvimento do *firmware* necessárias para a criação da lógica de controle e obtenção de resultado.

### 2.1 Futebol de robôs

Dentre as várias categorias, a que esse trabalho se insere é a SSL, uma categoria bastante disputada no Brasil e no mundo. Ela é uma classe de robôs autônomos, o que significa que não há interação com o ser humano durante as partidas.

Ela é composta por três grandes blocos: os robôs enquanto estruturas físicas, a visão computacional e a inteligência artificial.

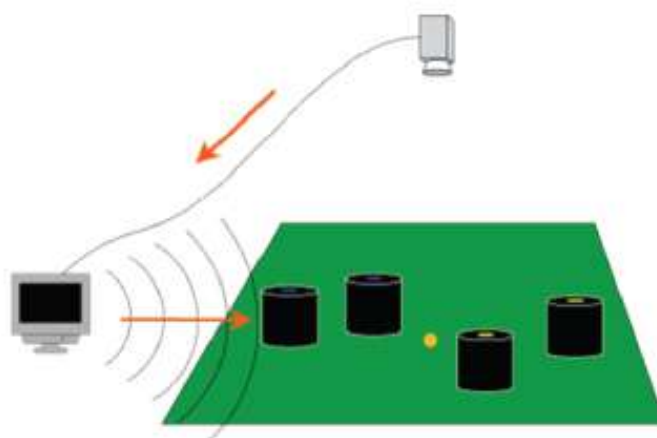


Figura 2.1: Esquema de jogo RoboCup Small Size League. Fonte: Fornecida pela RoboCup.

### 2.1.1 Visão Computacional

A visão computacional é responsável por captar o jogo propriamente dito, ela tem uma visão superior do jogo, e será processada afim de se obter o posicionamento dos seus jogadores, dos jogadores adversários e da bola em campo.

### 2.1.2 Inteligência Artificial

A inteligência artificial recebe da rede as posições que necessita para preparar sua jogada. É necessário ainda que saiba lidar com imprevistos como possível falta de jogador ou atrasos na comunicação.

## 2.2 Conceitos físicos

A capacidade de um jogador de futebol de produzir variadas ações de alta velocidade é creditada em influenciar em partidas [1]. Enquanto que atos de alta velocidade correspondem a apenas 11% da distância total coberta em um jogo, estas movimentações contribuem para os momentos mais cruciais e que mais pesam diretamente na marcação de pontos para o time com posseção da bola [2]. Para o caso do chute, tem-se dois parâmetros de interesse, a velocidade inicial e aceleração da bola após a ação.

### 2.2.1 Especificação de velocidade inicial

A velocidade inicial de uma bola indica a distância percorrida pela mesma em uma dada variação de tempo imediatamente após a realização do chute.

$$Velocidade = \frac{\Delta Espaco}{\Delta Tempo}$$

O que possibilita, em conjunto com o conhecimento da aceleração, a estimação da distância máxima que a bola consegue percorrer antes de uma parada total. É uma componente de extrema importância para a tomada de decisão se é válido fazer um chute a gol ou fazer o toque para um companheiro que encontra-se muito longe ou muito perto.

### 2.2.2 Especificação de aceleração

A aceleração é a taxa de mudança em velocidade da bola, ou seja, a mudança no valor de velocidade de um objeto após uma certa variação de tempo.

$$Aceleracao = \frac{\Delta Velocidade}{\Delta Tempo}$$

Assim, permite-se prever a velocidade da bola em qualquer ponto de sua trajetória logo após o chute, se sua velocidade inicial for conhecida. Outro parâmetro essencial para a estratégia de chutes em um time de futebol.

## 2.3 Hardware

Nesta seção trata-se sobre o funcionamento dos principais componentes presentes no *hardware* deste projeto, para facilitação do entendimento de sua construção.

### 2.3.1 Fotossensores

Fotossensores são dispositivos semicondutores que convertem a luz que recebem em corrente elétrica. Esta corrente é reproduzida quando fótons são absorvidos no componente; entretanto há uma corrente residual mesmo com a ausência de luz incidente [3].

#### Tipos

No âmbito dos fotodetectores, pode-se relatar sobre dois tipos de componentes que são os mais aplicados e difundidos em uso prático, são eles o fotodiodo e o fototransistor.

No caso do fotodiodo, este é composto de uma junção  $P-N$  designada para responder a uma estímulo de luz. Possuem um trecho constituído de fibra ótica, responsável por deixar a luz passar e incidir na parte sensível do dispositivo. Sem este trecho citado pode ser usado para detectar raios ultravioleta ou raios X. Devido a sua simplicidade de construção possui tempo de resposta menor que o fototransistor [4].

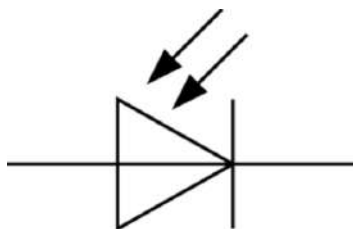


Figura 2.2: *Símbolo do fotodiodo*. Fonte: Elaborada pelo autor.

Enquanto que em um fototransistor, existem dois terminais, correspondendo ao coletor e emissor do transistor. A base é ativada pela luz; quando uma quantidade suficiente de luz é captada, o transistor conduz, permitindo a passagem de corrente do coletor para o emissor. Sem a luz, o transistor não conduz e coletor e emissor ficam isolados. Possui uma sensibilidade muito maior à luz, pois os elétrons que são gerados pelos fótons na junção da base-coletora são aplicados dentro da base, a sua corrente é

então amplificada pela operação do transistor [5].

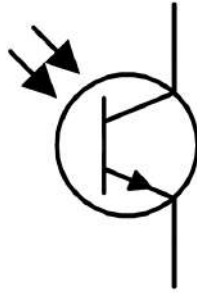


Figura 2.3: *Símbolo do fototransistor*. Fonte: Elaborada pelo autor.

### Aplicações

Estes dispositivos são geralmente empregados para gerar uma saída que depende da iluminação para fins analógicos; para medição e afins ou para alterar o estado do circuito de forma digital; para controle e comutação, ou processamento de sinal digital.

Tratando-se de fotodiodos, estes são utilizados em leitores de CD, detectores de fumaça, certos dispositivos médicos [6] e em controles remotos infravermelhos.

Qualquer tipo de fotossensor pode ser usado para medição de luz como feito em câmeras, ou para responder a níveis de luz, como ligar a iluminação pública após anoitecer.

Fotossensores de todos os tipos podem ser usados para responder à luz incidente ou a uma fonte de luz que faça parte do mesmo sistema. Frequentemente combina-se o dispositivo em um único conjunto com um emissor de luz, geralmente um LED, tanto para detectar a presença de uma obstrução mecânica ao feixe.

### 2.3.2 Microcontrolador

Um microcontrolador é um componente capaz de processar variados dados em um único circuito integrado. Na terminologia moderna, possui funcionamento semelhante, mas menos sofisticado, que um sistema em um chip como feito nos computadores comerciais; por exemplo, um destes chips pode incluir um microcontrolador como um de seus componentes. Um microcontrolador contém um ou mais processadores, juntamente com periféricos de entrada e saída de memória e programáveis. A memória de programa é frequentemente incluída no dispositivo, bem como uma pequena quantidade de RAM. Microcontroladores são projetados para aplicações embarcadas, em contraste com os microprocessadores usados em computadores pessoais ou outros aplicativos de uso geral que consistem em vários chips discretos [7].



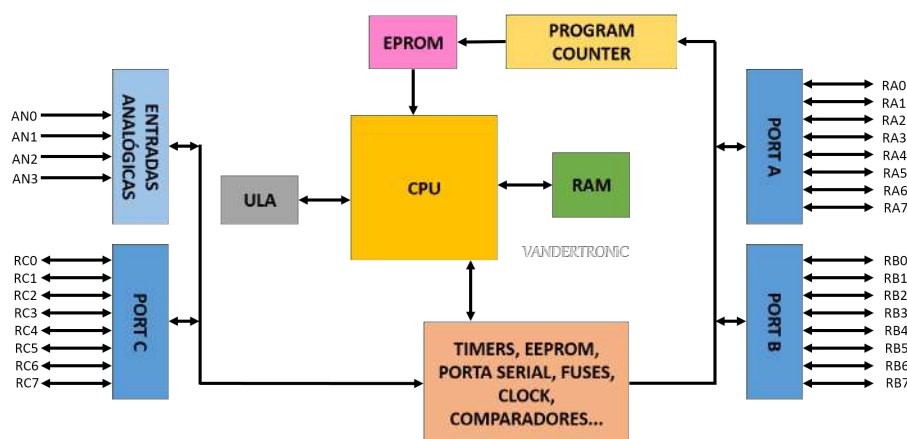


Figura 2.4: Diagrama em blocos de construção básica de um microcontrolador. Fonte: Vandertronic.

### Aplicações

Os microcontroladores são usados em produtos e dispositivos controlados automaticamente, como sistemas de controle de motores de automóveis, dispositivos médicos implantáveis, controles remotos, máquinas de escritório, eletrodomésticos, ferramentas elétricas, brinquedos e outros sistemas embarcados. Ao reduzir o tamanho e o custo em comparação com um projeto que usa um microprocessador separado, memória e dispositivos de I/O, os microcontroladores tornam econômico o controle digital de ainda mais dispositivos e processos. Microcontroladores de sinais mistos são comuns, integrando componentes analógicos necessários para controlar sistemas eletrônicos que não são digitais. No contexto da internet das coisas, os microcontroladores são um meio econômico e popular de coleta de dados, detectando e atuando no mundo físico como dispositivos periféricos [8].

Alguns microcontroladores podem usar palavras de quatro bits e operar em frequências tão baixas quanto 4 kHz, para baixo consumo de energia (miliwatts de um dígito ou microwatts). Eles geralmente têm a capacidade de manter a funcionalidade enquanto aguardam por um evento, como um pressionamento de botão ou outra interrupção; consumo de energia durante o desligamento do relógio da CPU e da maioria dos periféricos pode ser apenas nanowatts, tornando muitos deles adequados para aplicações de bateria de longa duração. Outros microcontroladores podem servir a funções de desempenho crítico, onde eles podem precisar agir mais como um processador de sinal digital, com velocidades de *clock* e consumo de energia [8].

### 2.3.3 Liquid-crystal Display

Um *display* de cristal líquido, ou LCD do inglês *liquid-crystal display*, é um monitor de tela plana ou outro dispositivo óptico modulado eletronicamente que usa as propriedades moduladoras de luz de cristais líquidos. Os cristais líquidos não emitem luz diretamente, em vez disso, usam uma luz

de fundo para produzir imagens coloridas ou monocromáticas [9]. Os LCDs estão disponíveis para exibir imagens dinâmicas mais detalhadas ou imagens fixas com baixo conteúdo de informação, que podem ser exibidas ou ocultadas, como palavras predefinidas, dígitos e exibições de sete segmentos, como em um relógio digital . Eles usam a mesma tecnologia básica, exceto que imagens arbitrárias são compostas de um grande número de pixels pequenos, enquanto outras exibem elementos maiores.



Figura 2.5: *Módulo de display LCD comumente usado em instrumentações simples.* Fonte: Elaborada pelo autor.

As telas de LCD substituíram as telas de tubo de raios catódicos volumosas e pesadas em quase todas as aplicações. As telas de LCD estão disponíveis em uma ampla gama de tamanhos de tela do que os monitores CRT e de plasma, com telas de LCD disponíveis em tamanhos que variam de minúsculos relógios digitais a receptores de televisão muito grandes.

### **Aplicações**

Os LCDs são usados em uma ampla gama de aplicações, mais notoriamente por televisores LCD e monitores de computador, porém também tem aplicação em painéis de instrumentos, através de pequenas telas de LCD. São comuns em dispositivos portáteis de consumo, como câmeras digitais, relógios, calculadoras e smartphones. As telas de LCD também são usadas em produtos eletrônicos de consumo, como DVD players e dispositivos de videogame.



Figura 2.6: Exemplo de aplicação de display LCD, usado no iPhone 7. Fonte: Apple Inc.

## 2.4 Firmware

Nesta seção trata-se sobre a lógica e recursos empregados no *firmware* deste projeto, para facilitação do entendimento de sua implementação.

### 2.4.1 Desenvolvimento de firmware embarcado

Com a evolução e popularização do uso de sistemas embarcados nas últimas décadas, surge-se uma necessidade de códigos acompanhantes cada vez mais complexos e extensos para a realização de certas tarefas e requisitos. Seguindo o aumento das capacidades do *hardware* destes sistemas, ficou possível o porte e processamento de *firmwares* mais elaborados, entretanto para facilitar a sua programação que inicialmente era realizada através da linguagem *assembly*, que dificulta a lógica e construção de códigos extensos, foram criados interpretadores de linguagens como a C e C++, para que estas possam ser compiladas e facilitar o desenvolvimento do programador [10].

### 2.4.2 Timer e Counters

Timers e counters, ou temporizadores e contadores, são usados em todo projeto para fornecer uma série de tempo ou contar eventos relacionados dentro do sistema com o mínimo de sobrecarga de processador e software.

O tempo gerado é derivado de uma entrada de *clock*. Este *clock* pode ser interno ao temporizador/contador ou ser externo e assim conectado através de um pino separado. O *clock* pode ser dividido usando um divisor simples que pode fornecer divisão limitada normalmente baseada em uma potência de dois ou através de um pré-escalar que efetivamente dimensiona para baixo ou divide o *clock* pelo valor que está contido registro do pré-escalar. O *clock* dividido é então passado para um contador que geralmente é configurado em uma operação de contagem regressiva, ou seja, ele é carregado com um

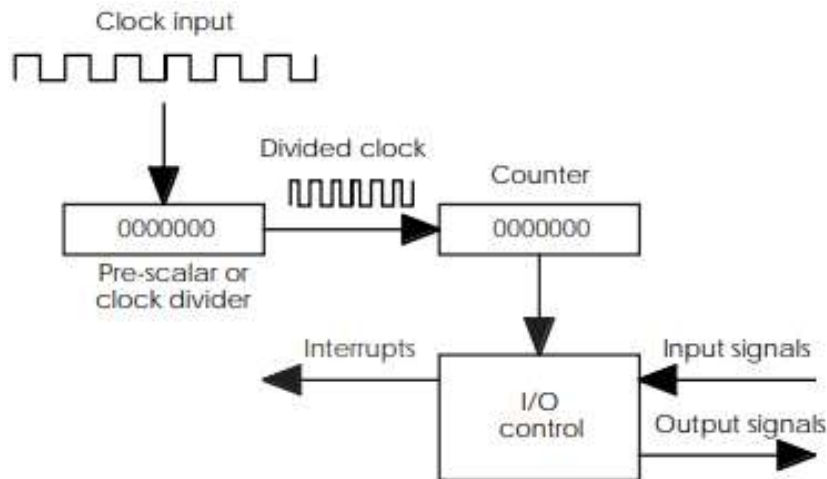


Figura 2.7: Estrutura básica de funcionamento de temporizadores e contadores. Fonte: Elaborada pelo autor.

valor pré-definido que é reduzido para zero. Quando uma contagem zero é atingida, isso faz com que um evento ocorra, como a interrupção de um estado de mudança de linha externa. O bloco final é vagamente descrito como um bloco de controle de I/O, mas pode ser mais sofisticado se necessário. Ele gera interrupções e pode controlar contadores baseados em sinais externos, podendo bloquear a contagem regressiva e fornecer um controle adicional [11].

## Tipos

*Timers* e *counters* normalmente são definidos em termos do tamanho de contagem que podem fornecer. Geralmente vêm em variantes de 8, 16 e 24 bits. O tamanho do bit determina duas propriedades fundamentais:

- O valor pré-escalar  $e$ , portanto, a frequência dos mais baixos *clocks* que podem ser gerados a partir de uma determinada entrada de clock.
- O tamanho do contador determina o valor máximo do período de contra-derivação e quando utilizado com um *clock*, a resolução máxima ou a medição de um evento baseado em tempo.

Essas duas propriedades frequentemente determinam a adequação de um dispositivo para a aplicação pretendida.

## Aplicações

A maioria dos sistemas embarcados tem um componente de tempo interno como referência temporal para as sequências de controle, para fornecer os passos para o sistema operacional e até mesmo a geração de formas de onda para a taxa de transmissão serial.

### 2.4.3 Interrupção

Uma interrupção é, como o próprio nome sugere, uma maneira de parar a *thread* que o processador está executando no momento, passando para uma rotina de software diferente, executando-a antes de restaurar o status do processador para antes da interrupção, continuando o processamento anterior.

Interrupções podem ocorrer de forma assíncrona à operação e podem assim serem usadas de forma mais eficiente com sistemas que funcionam através de eventos ao invés de dependerem de tempos. No entanto, eles podem ser usados para sistemas acionados por tempo sem ter que recorrer a temporizadores baseados em software.

Seu funcionamento mais a fundo acontece fazendo-se com que o código da porta de dados seja removido do plano de fundo software de análise. O código de transferência de dados é guardado como parte do rotina de serviço de interrupção associada à interrupção gerada pelo hardware da porta de dados. Quando a porta recebe um byte de dados, gera uma interrupção. Isso ativa o ISR que processa os dados antes de entregar a execução de volta ao segundo plano de tarefa. O fundo tarefa pode ser modificado independentemente do código da porta de dados e que todo o ordenamento de leitura do sistema agora é movido da forma dependente em intervalos de pesquisa anterior para um modo em que tão rapidamente os dados podem ser acessado e processado [11].

#### Tipos

Os tipos de interrupção mais importante para a fundamentação deste documento são as interrupções interna e externa.

Interrupções internas são aquelas que são geradas pelo próprio microcontrolador através de periféricos como portas seriais e paralelas. Com um periférico externo, o dispositivo normalmente possui um pino que está conectado a um pino de interrupção no processador. Com os periféricos internos, essa conexão já foi feita. Alguns integrados permitem alguma flexibilidade em relação a essas conexões e possibilitam que o nível de prioridade seja ajustado ou mesmo mascarado e até desativado por completo.

Interrupções externas são o método comum de conexão de periféricos externos ao processador. Eles geralmente são fornecidos através de pinos que são conectados aos periféricos e que são assegurados pelo periférico. Por exemplo, uma porta serial pode ter um pino que é declarado quando há dados presentes em seus *buffers*. O pino pode ser conectado a interrupção do processador para quando o processador ver o sinal de dados pronto como uma interrupção, a rotina de serviço de interrupção correspondente buscaria os dados do periférico antes de restaurar o processamento anterior.

**Aplicações**

A vantagem de uma interrupção é que ela inclui um mecanismo para alterar o fluxo do programa e em algum processador arquiteturas, para mudar para um estado mais protegido. Isso significa que uma interrupção poderia ser usada para fornecer uma interface para outros software, como um sistema operacional. Esta é a função que é fornecida pela interrupção do software. É tipicamente uma instrução ou conjunto de instruções que permite que um software atualmente em execução sequencia para alterar o fluxo e retornar usando o mais normal mecanismo de interrupção.

## Capítulo 3

# Desenvolvimento do Projeto

Neste capítulo descreve-se o material utilizado durante a execução do projeto, seja ele *hardware* ou *software*, explicando também os métodos empregados durante todo o desenvolvimento afim de tornar possível a replicação dos resultados.

### 3.1 Material

Para a realização deste projeto, foi desenvolvido uma estrutura mecânica de acrílico montada com cantoneiras, parafusos e porcas M5, esta comporta os sensores e toda a eletrônica necessária, servindo também como um túnel para acomodar o robô *SSL* de teste e limitar o trajeto da bola. Devido a sua importância, informações a respeito da armação projetada estão apresentadas na Subseção 3.1.1.

Em seguida ao material mecânico, arquitetou-se a porção eletrônica do projeto. Neste segmento, o microcontrolador utilizado é o *PIC18F4550*; para disponibilizar os resultados e fazer a interface de usuário foi selecionado o *LCD TUXGR\_20X2*; acompanhando um circuito de regulação de tensão, compostos pelos integrados *LM7810* e *LM7805*; e no *reset* do *PIC* usa-se um *push-button*. A Subseção 3.1.2 discursa detalhadamente o microcontrolador escolhido enquanto a Subseção 3.1.3 apresenta o *LCD* usado.

Os sensores que realizam a leitura do chute são os fototransistores *TIL78* e os *LEDs* infravermelhos *TSAL6200*, especificados na Subseção 3.1.4. Os *LEDs* são protegidos por bocais e todo o cabeamento dos fotosensores é conduzido por aletas nas laterais da estrutura mecânica em direção ao microcontrolador.

Outros circuitos eletrônicos presentes no sistema incluem o *pin-head* para gravação e leitura do *firmware* no *PIC*; os *LEDs* para *debug* junto do *driver* 4066D; e os conectores *MOLEX* para comunicação dos sensores com o microcontrolador. A alimentação de todos os circuitos mencionados tem origem de um *Jack DCJ-202*. Toda a eletrônica concentra-se em uma PCB desenvolvido no *CADSoft*

*Eagle* (Subseção 3.1.5), localizada em uma caixa de plástico para proteção.

O *firmware* utilizado para o controle deste módulos foi desenvolvido e programado no *PIC* com o auxílio do ambiente *MPLAB X IDE 1.35*, usando-se exclusivamente a linguagem C na codificação. Para programação via *ICSP* utilizou-se o gravador *PICKit 3*.

A lista de componentes utilizados:

### **Mecânica**

- 3 Placas de acrílico de 500 x 150 x 5 mm
- 4 Aletas de plástico de 500 mm
- 8 bocais de LED 5mm
- 12 parafusos e 12 porcas M3, 6 parafusos e 8 porcas M5
- 4 cantoneiras de 500 mm
- 1 caixa de plásticos 125 x 100 x 45 mm com abertura para periféricos

### **Eletrônica**

- 1 Microcontrolador *PIC18F4550*
- 1 *LCD TUXGR\_20x2*
- 4 Fototransistores *TIL78* e 4 *LEDs* infravermelho *TSAL6200*
- 1 *Push-button*
- 4 Resistores de 5M $\Omega$ , 3 resistores de 10K $\Omega$ , 4 resistores de 1K $\Omega$ , 4 resistores de 270 $\Omega$  e 2 resistores de 22 $\Omega$
- 4 Capacitores de 100nF, 2 capacitores de 330nF e 1 capacitor de 1uF
- 1 Regulador de tensão 10V *LM7810* e 1 Regulador de tensão 5V *LM7805*
- 1 Driver de *LED 4066D*
- 1 *Jack DCJ-202*
- 1 *pinhead 1x5*, 1 *pinhead 1x2*, 1 *MOLEX 1x5* e 1 *MOLEX 1x4*
- 4 *LEDs* 5mm
- 1 conector USB



### 3.1.1 Túnel de Acrílico

A escolha do acrílico entre outros polímeros foi decidida por suas características materiais, entre elas: resistência ao impacto 10 vezes maior que a do vidro; densidade 2 vezes menor que o vidro, menor peso para o mesmo objeto; bloqueia luz infravermelha acima de 25.000nm; e 92% de transmitância luminosa (transparência);

O túnel de acrílico compõe o corpo principal do sistema, é composto por 3 paredes retangulares e com um comprimento de 500 mm. Este tem a função de comportar todos os outros módulos do projeto, além de também definir o trajeto a qual a bola deve percorrer. A composição da estrutura escolhida, o acrílico, deve-se ao baixo peso, robustez, disponibilidade e praticidade de manipulação do material, a transparência característica também permite a observação do trajeto da bola dentro do túnel, podendo ser notado quaisquer colisões dentro da estrutura ou angulações no chute do robô.

São dispostos bocais ao longo das paredes em intervalos de espaço iguais para os fotossensores, perfurando a placa na altura do centro de uma bola de golfe, A mm, ao todo são 8 bocais sendo 4 de cada lado. A fiação gerada pelos fotossensores são contidas em aletas localizadas nas laterais externas do túnel, escondendo e protegendo-a.

Na tampa superior do aparato em uma de suas extremidades encontra-se a caixa de plástico onde é mantido toda a eletrônica do sistema, este reservatório fica parafusado na placa do túnel em que está repousado.

### 3.1.2 Microcontrolador *PIC18F4550*

É o principal componente eletrônico do sistema, o *PIC18F4550* faz a leitura dos sensores e periféricos de interface, imprimindo as mensagens e resultados no *LCD* fornecido, assim realiza todo o controle de lógica para a medição dos parâmetros de chute, e ainda faz o reconhecimento dos comandos do usuário.

Algumas funcionalidades deste microcontrolador podem ser conferidas na seguinte lista:

- 16kB de memória de programação
- 2048 *bytes* de *RAM*
- Programável por *ICSP*
- Oscilador interno *PLL* de 31kHz até 48MHz
- Interface *USB* pronta
- 13 Conversor *A/D* de 10 *bits*

- 34 Pinos *I/O*
- 3 *Timers*
- Múltiplas Interrupções com função de prioridade

A seleção deste microcontrolador remete-se a familiaridade do autor do projeto com a família *PIC* da *Microchip*, por serem baratos, amplamente disponíveis e de fácil implementação, são os CI's ideais para aplicações simples e amadoras. O modelo *PIC18F4550* foi selecionado em específico por possuir um número adequado de pinos *I/O* e funcionalidades como interface USB, oscilador interno e memória de programação, conforme estimado nas elaborações de esquemático eletrônico.

### 3.1.3 *LCD TUXGR\_20x2*

Este é o *display* responsável por representar os dados e mensagens enviados pelo *PIC*, logo é o componente que faz a interface homem-máquina, permitindo que o usuário consiga visualizar sua interação com o sistema. É um *LCD* 20x2, ou seja, é capaz de representar até 20 caracteres em cada uma de suas 2 linhas em uma única mensagem. Consegue representar caracteres alfanuméricos, japoneses, entre outros símbolos.

Possui alimentação de 5V e consome até 2.5mA durante seu funcionamento, além deste modelo possuir 19 comandos diferentes para uma configuração personalizada do *display*, a qual o usuário pode fazê-lo através de um microcontrolador.

### 3.1.4 **Fototransistor *TIL78* e *LED* infravermelho *TSAL6200***

Para compor os sensores de presença da bola de golfe foram selecionados o fototransistor *TIL78* e *LED* infravermelho *TSAL6200*, ambos possuem encapsulamento convencional de *LED* de 5mm. No caso do *TIL78*, possui capacidade de dissipação de até 50mW, não havendo restrição de tensão reversa de diodo, desde que a potência dissipada esteja adequada.

### 3.1.5 *CADSoft Eagle*

Atualmente gerenciado pela *AUTODESK*, o *software CAD Eagle* é uma ferramenta de desenvolvimento de *PCBs*, neste programa é possível elaborar os *schematics*, que são esquemáticos com as representações das conexões elétricas entre os componentes de um circuito; e também os *boards* que apresentam o layout físico do *PCB* a partir do esquemático feito anteriormente. Por padrão, possui bibliotecas de diversos componentes, principalmente os mais comuns e amplamente utilizados, inclusive todos os componentes usados no projeto já estavam inclusos nestas bibliotecas do software)

### 3.1.6 MPLAB X IDE 1.35

O *MPLAB X IDE 1.35* é o ambiente de trabalho próprio para a programação de microcontroladores, possuindo a configuração pronta de centenas de modelos diferentes. Este foi utilizado para codificação inteira do *firmware* além da submissão do código via *ICSP* para o *PIC18F4550*.

### 3.1.7 PICKit 3

É a ferramenta utilizada para gravar o *firmware* desenvolvido no microcontrolador, o *PICKit 3* é próprio para os componentes da família *PIC*, no caso este suporta os microcontroladores das linhas *PIC16F*, *PIC18F*, *dsPIC33F*, *PIC24* e *PIC32*. Funciona através de conexão *In Circuit Serial Programming* (ICSP), também possui função de *debug*, onde é possível verificar em tempo real as condições e configurações do equipamento.

## 3.2 Métodos

O desenvolvimento do projeto pode ser dividido em três grandes partes: a estrutura mecânica do sistema; a eletrônica geral, incluindo o PCB, os sensores e o cabeamento; e o *firmware* que faz o controle do funcionamento dos módulos.

Antes de iniciar a manufatura e outras tarefas laborais, foi estabelecido como seria pautada sua lógica de funcionamento, sendo assim elaborado o esquemático da Figura 3.1 ao qual define o funcionamento básico da composição dos fotossensores.

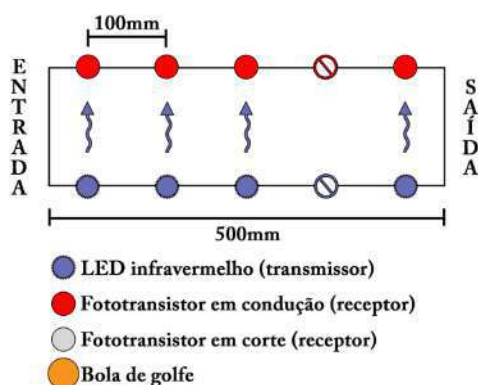


Figura 3.1: Diagrama da lógica de sensoriamento do sistema sem a presença da bola. Fonte: Elaborada pelo autor.

Enquanto que na Figura 3.2 fica evidenciado o funcionamento do registro de presença da bola, a interrupção da luz infravermelha corta a condução do fototransistor obstruído. Nota-se que o quarto par de fotossensores não foi incluído, essa decisão foi tomada com intuito de usar o espaço liberado para

aprimorar o cálculo de desaceleração próximo da saída do túnel, evitando de pegar apenas acelerações muito próximas.

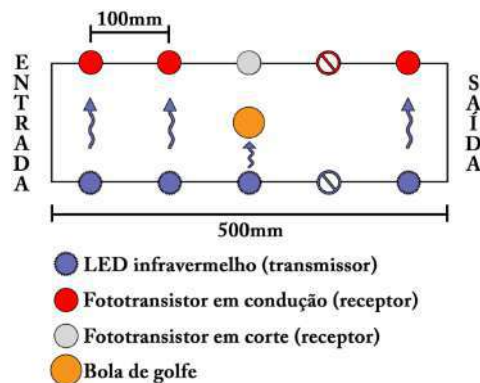


Figura 3.2: Diagrama da lógica de sensoriamento do sistema com a presença da bola. Fonte: Elaborada pelo autor.

Definido o fundamento do projeto, foi começado o processo de construção da ferramenta começando pela estrutura mecânica para portar as outras demais funcionalidades, em seguida produziu-se a porção eletrônica para possibilitação das leituras e, finalmente, a codificação do *firmware*.

Para a armação mecânica, foi necessário construir um túnel de acrílico (Subseção 3.2.1) para portar todos os componentes de forma robusta e de fácil manuseio.

Em seguida, para o desenvolvimento da eletrônica do projeto foi usado a estrutura mecânica elaborada para definir o formato do PCB (Subseção 3.2.2), além de concentrá-la em uma caixa de plástico para proteção e praticidade, a placa foi manufaturada de acordo com a Subseção 3.2.3, processo que finalizou-se na integração dos módulos criados assim como descrito na Subseção 3.2.4.

Em relação ao *firmware*, foi desenvolvido um código único que realiza todas as funções previstas no projeto, a configuração do LCD, opção para salvamento de dados, comunicação USB e a interface de usuário. O desenvolvimento do *firmware* como um todo pode ser vista na Subseção 3.2.5

É importante ressaltar que o fornecimento de todos os materiais do projeto veio do grupo *Warthog Robotics*, enquanto que a manufatura foi realizada aproveitando-se das oficinas e recursos do *Campus* de São Carlos da Universidade de São Paulo.

### 3.2.1 Montagem do Túnel de Acrílico

Após as definições das dimensões do sistema, e principalmente da estrutura mecânica, começou-se a manufaturar o túnel de acrílico. Comprou-se 3 placas de acrílico de 500 x 150 mm e com 5mm de espessura, valores que permitem a disposição e instalação dos sensores adequadamente ao longo das laterais. As paredes são fixadas com o auxílio de 12 parafusos e porcas M3 perfurados em intervalos

de 70mm nas extremidades de junção das placas de acrílico, além de mais 2 parafusos e 8 porcas M5 para evitar que as paredes perpendiculares ao solo entortem para dentro ou fora. Um esboço deste esquema pode ser visto na Figura 3.5.

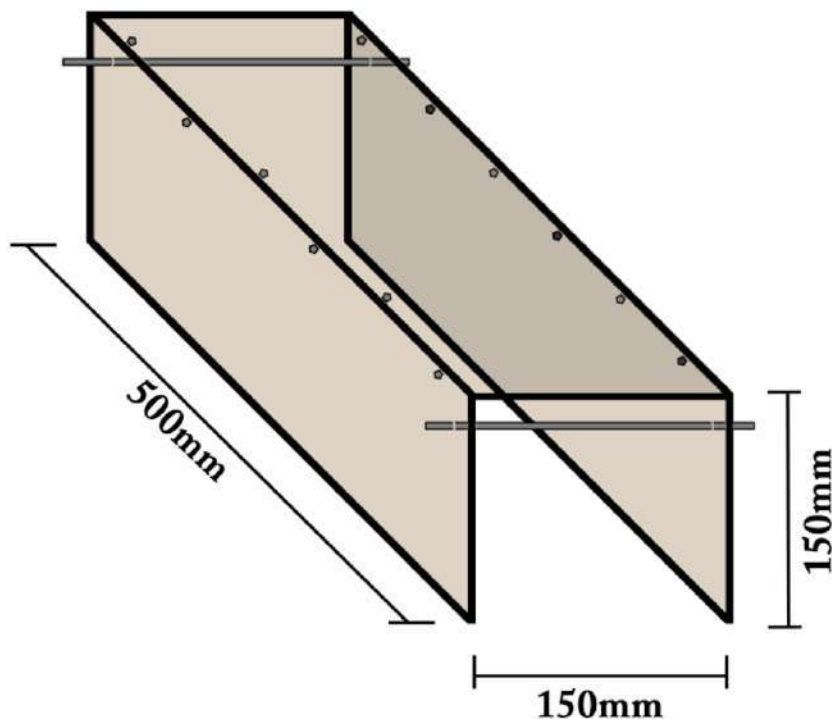


Figura 3.3: *Esboço da armação mecânica elaborado para o projeto.* Fonte: Elaborada pelo autor.

Para instalar os fotossensores no túnel elaborado, foram feitos furos de 5mm de diâmetro em intervalos de 80mm que receberam bocais próprios para LEDs de 5mm, estes fazem a proteção dos sensores de colisões acidentais com a bola após os chutes. Os bocais desembocam em aletas acrescentadas nas laterais externas do túnel, percorrendo todo o comprimento da estrutura.

Com os espaços para os sensores montados, fixa-se a caixa do PCB acima do túnel em uma das suas extremidades, para tal são feitos 4 furos de 5mm, 1 para cada canto da caixa, no teto da estrutura de acrílico, estas perfurações batem com os buracos pré-existent na caixa comprada logo com 4 parafusos M5 de tamanho suficientemente grande conseguem firmar o envólucro no túnel. A Figura 3.5 demonstra o acoplamento realizado.

Por último na montagem da estrutura, para melhorar a estabilização da armação e evitar deslocamento nas paredes durante o seu uso foram implementadas cantoneiras no projeto, 2 nas bases do túnel para mantê-lo estável quando repousado no solo e 2 ao longo das junções das paredes de acrílico, todas com 500mm de comprimento.



Figura 3.4: *Aletas instaladas nas laterais da estrutura.* Fonte: Elaborada pelo autor.

### 3.2.2 Elaboração do PCB (*TunnelBoard*)

A elaboração do PCB do projeto ocorreu em paralelo com a definição da estrutura mecânica, usando o espaço disponível nos esboços do túnel para encaixar a porção eletrônica desenvolvida. Com o tamanho necessário em mente, definiu-se as funcionalidades desejadas pelo sistema e separou-se os componentes principais que cumprissem os requisitos estabelecidos. Os componentes selecionados determinam, então, os periféricos e circuitos menores necessários para seu funcionamento, no caso do *PIC18F4550* tem os capacitores de tanque, o circuito de *reset*, de *ICSP* e para o *USB*; o *LCD TUXGR\_20x2* demanda apenas um resistor para definição de contraste; os fotossensores *TIL78* e *TSAL6200* também precisam apenas de alguns resistores, embora sejam para limitação de corrente; os demais circuitos como os *LEDs* de *debug* e regulação de tensão surgem de necessidades gerais do PCB.

Os dimensionamentos da alimentação foram definidos segundo os cálculos:

#### **Corrente máxima total calculada: 228mA**

- Fotossensores - 80mA (LED infravermelho);
- LCD - 3mA
- Microcontrolador - 140mA (100mA para I/O + 40mA geral);
- LED alimentação - 5mA;

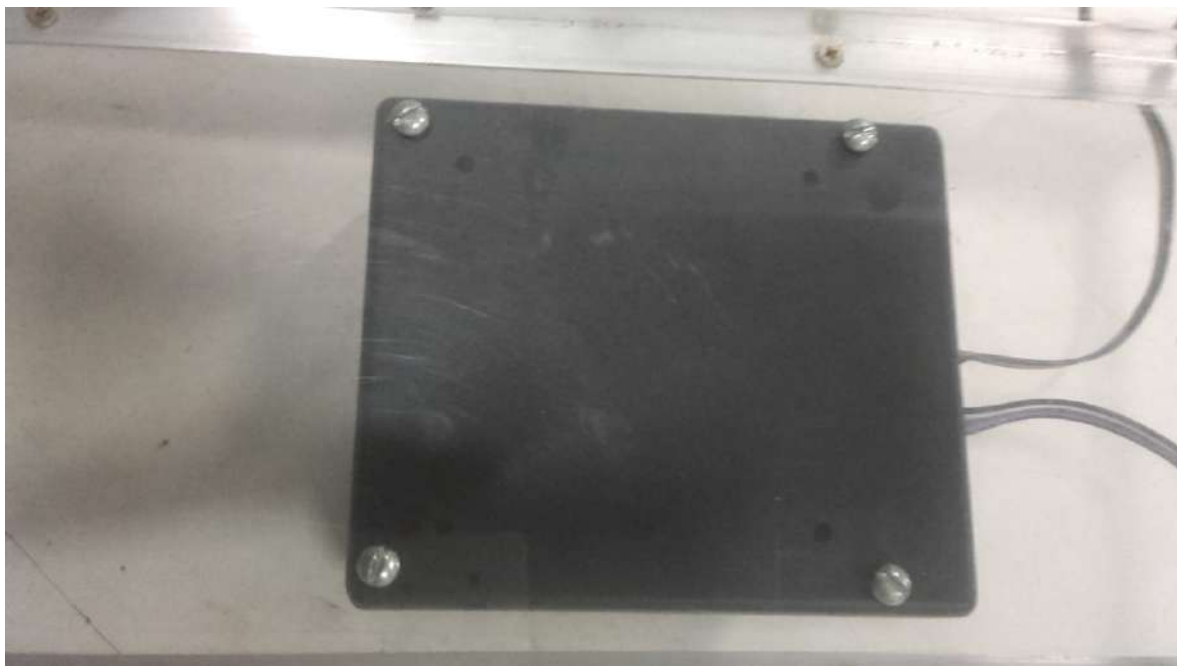


Figura 3.5: Fixação da caixa do PCB do túnel de acrílico. Fonte: Elaborada pelo autor.

#### **Limites de tensão de alimentação: 12V a 18,77V**

- MÍNIMO => Dropout mínimo do LM7810: 2V
- MÁXIMO => Dissipação máximo do LM7810: 2W ->  $P = V \cdot I \Rightarrow 2 = V \cdot 0,228 \Rightarrow V = 8,77$   
V

#### ***Schematic***

Na realização do esquemático foi utilizado o *CADSoft Eagle 9.0.1*, neste fez-se uso de apenas um *sheet* para todo o circuito, visto que separar em módulos faria com que *sheets* de certas porções do projeto ficassem muito simples. O *datasheet* dos principais componentes foram constantemente consultados para replicar circuitos essenciais para o funcionamento do dispositivo. A alimentação foi a última parte realizada, para que atendesse aos níveis demandados pelos demais componentes implementados.

O esquemático gerado em detalhe pode ser visto no Apêndice A, este foi separado em módulos de funcionalidades semelhantes para uma leitura mais intuitiva e simples do documento.

#### ***Board***

Com o espaço disponível da caixa de proteção em consideração, definiu-se os limites da placa projetada desejando-se vagar o máximo de área possível para manejar os componentes no *layout*. Também atentou-se a posição dos conectores para que pudessem ter saída adequada para o cabeamento neces-

sário, outro caso foi os parafusos que passam pela placa para fixação, sendo necessário liberar este espaço com margens de segurança. O *LCD* também influenciou a decisão do desenvolvimento do desenho, visto que apenas encaixava dentro do envólucro em uma orientação.

O layout físico do PCB é representado no Apêndice B, comentários quanto as nuances da *board* podem ser vistos no arquivo. O *board* criado utiliza as duas faces de uma placa de cobre.

### 3.2.3 Confecção do PCB (*TunnelBoard*)

Para o processo de manufatura foi adotado a transferência térmica devido aos equipamentos disponíveis no *Campus* para tal. Começou-se com a placa de cobre separada para manufatura, esta foi cortada para um tamanho adequado de 115 x 110mm e limpa com álcool usando uma palha de aço, retirando quaisquer resquícios de impurezas na superfície.

Logo depois, o circuito da parte superior do PCB foi impresso à laser em papel apropriado e fixado em uma das faces da placa, certificando-se da firmeza do papel a placa passa pela prensa térmica em 180°C por 90 segundos. Espera-se a placa esfriar na retirada do equipamento, faz-se os furos guias e o circuito da parte inferior é agora fixado na face restante da placa atentando-se ao alinhamento dos lados, em seguida repete-se o processo com a prensa.

Feito a impressão nas duas faces, é iniciado a corrosão do cobre não coberto pelo desenho do circuito, para tal a placa é totalmente submersa em uma bacia contendo  $\text{FeCl}_3$  por cerca de 25 minutos até que todas as regiões desejadas estejam visivelmente corroídas. Limpa-se a placa com água para retirar os excessos da química utilizada e o PCB novamente é varrido com álcool para remover a tinta de impressão e outras impurezas. Após, é checado para identificar quaisquer falhas no desenho final em cobre que possam ser consertados.

Antes de soldar os componentes, resta apenas fazer os furos necessários da placa, estes são feitos com as furadeiras de bancada da oficina do departamento de elétrica, terminado o processo começa-se o processo de soldagem. Separa-se os componentes comprados e solda inicia na alimentação do circuito, sempre testando o funcionamento e as conexões realizadas, pasta térmica é colocada no regulador de 5V; depois é soldado os periféricos do microcontrolador, componentes novamente verificados atentadamente; e, por último, solda-se o soquete *PIC* e o *pin-header* do *LCD*, nesta etapa são feitos testes de continuidade para averiguação de funcionamento. A placa finalizada pode ser observadas nas Figuras 3.6, 3.7 e 3.13.



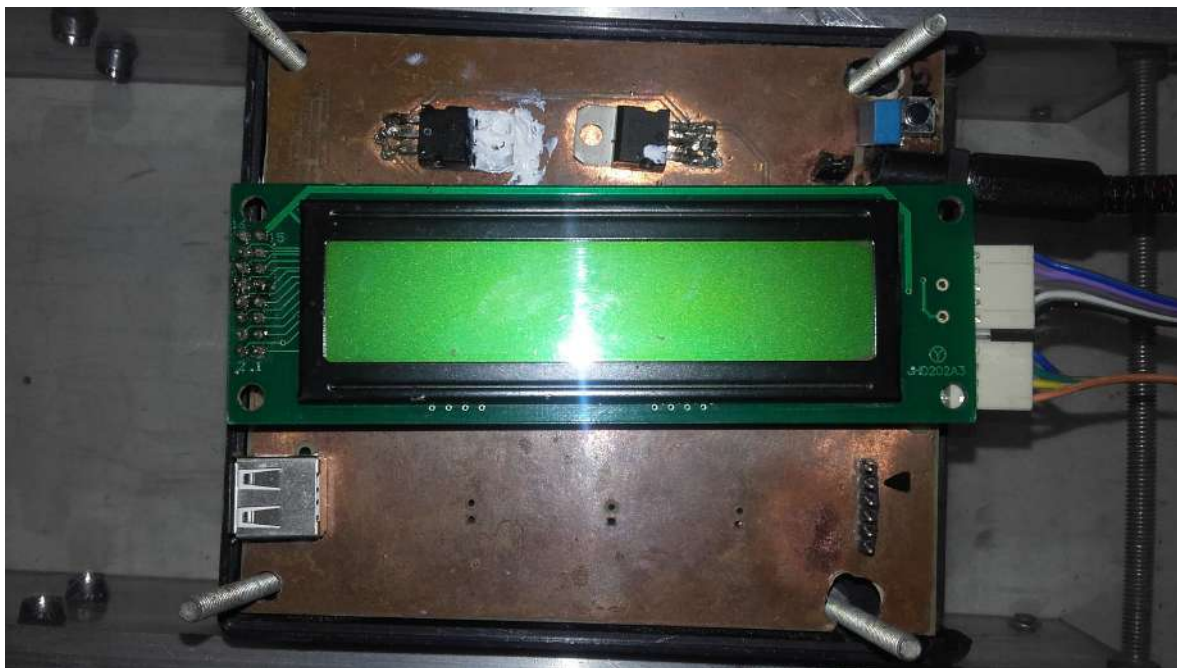


Figura 3.6: *PCB soldada - vista por cima.* Fonte: Elaborada pelo autor.



Figura 3.7: *PCB soldada por baixo.* Fonte: Elaborada pelo autor.

### 3.2.4 Integração Mecânica e Eletrônica

Com o PCB pronto, foi instalado a placa em sua caixa anexada ao corpo do túnel de acrílico, ajustando para que a fiação futuramente em comunicação como circuito possa adentrar o envólucro. Introduziu-se um calço de espuma na proteção eletrônica para amenizar choques na estrutura que possam afetar

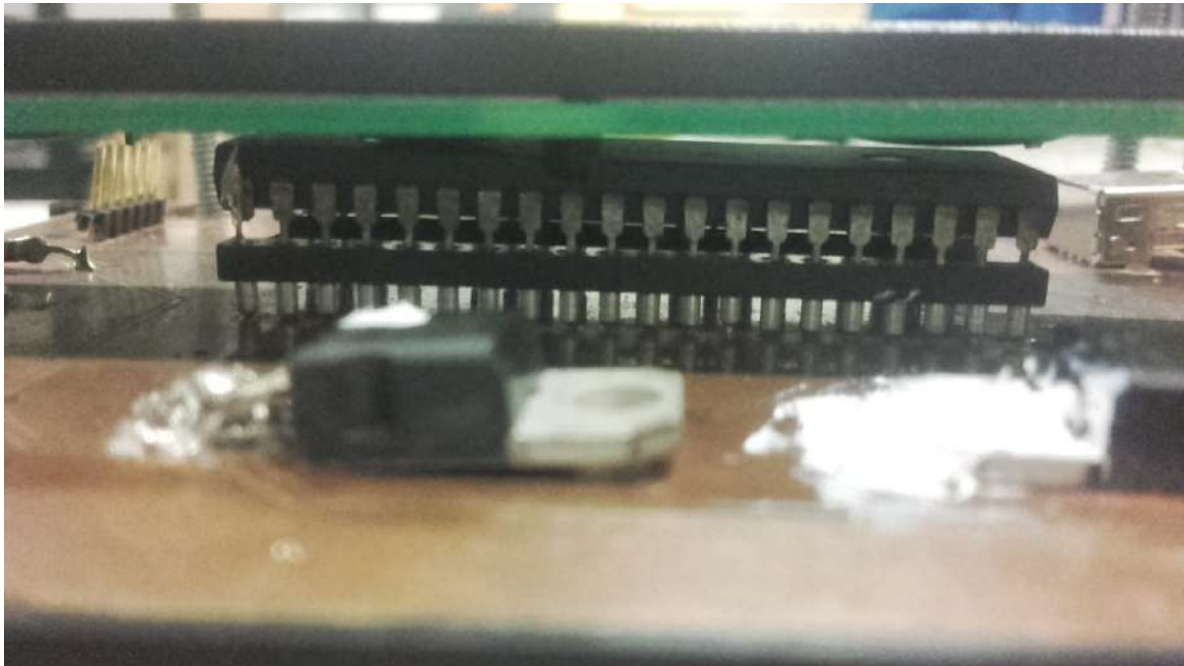


Figura 3.8: *PCB vista de perfil*. Fonte: Elaborada pelo autor.

a placa.

Mudando o foco para os sensores, os fototransistores são instalados no lado esquerdo da parede do túnel, nos bocais previamente posicionados, enquanto os *LEDs* de infravermelho são colocados no lado direito, também inseridos nos bocais ali presentes.

A fiação dos fotossensores são elaboradas sob medida para percorrer o percurso da aleta desembocando na extremidade da caixa do PCB e finalizando em dois *MOLEX*, um para o receptor e outro para transmissor, conectados à placa pela abertura disponibilizada.

### 3.2.5 Desenvolvimento de *Firmware*

Para o *firmware*, o primeiro passo foi definir as funcionalidades e um esboço de organização da codificação. O código desenvolvido em detalhe pode ser visto no Apêndice C, este foi separado em módulos de funcionalidades semelhantes para uma leitura mais intuitiva e simples do documento.

#### Configuração de *PIC18F4550*

Considerando o planejamento feito, iniciou-se a configuração do microcontrolador *PIC18F4550* para cumprimento dos requisitos estabelecidos, entabulando com os registradores gerais do *PIC* conforme visto no trecho de código comentado a seguir:

```
1 // CONFIGURACAO REGISTRADORES
2
```

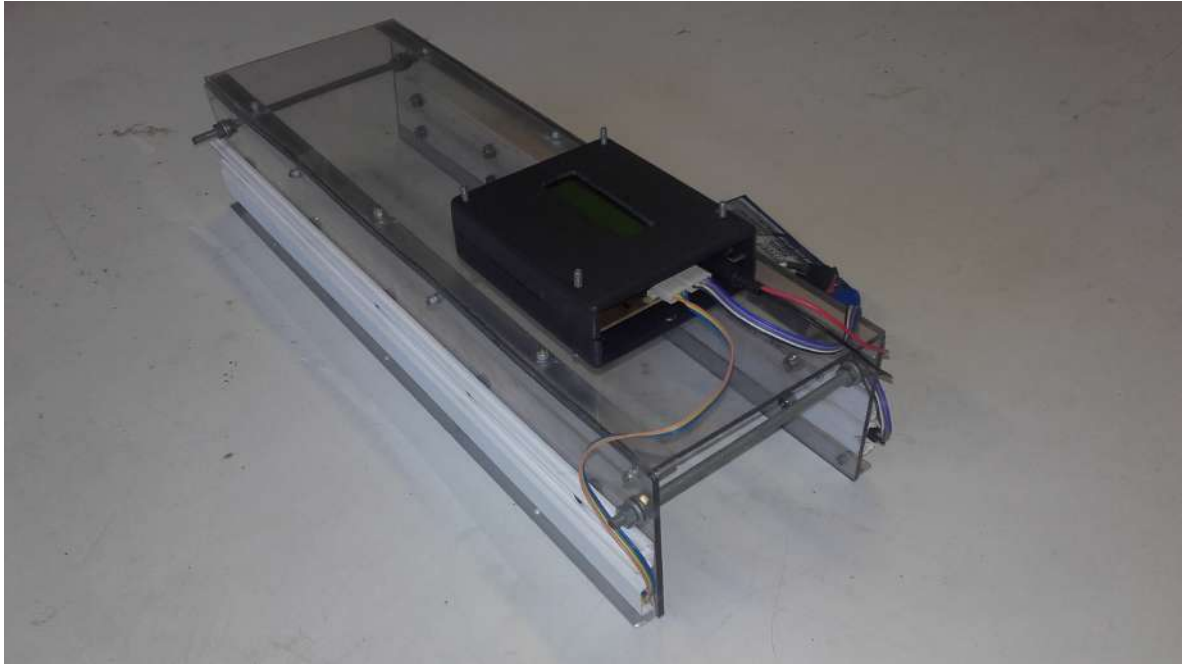


Figura 3.9: Projeto após integração de mecânica com a eletrônica. Fonte: Elaborada pelo autor.

```

3 #pragma config FOSC = INTOSC_EC //internal oscillator 4MHz
4 #pragma config PLLDIV = 1 //divide PLL
5 #pragma config FCMEN = OFF //no fail safe clock monitor
6 #pragma config IESO = ON //oscillator switchover disabled
7 #pragma config VREGEN = OFF
8 #pragma config PWRT = OFF //oscillator power up timer enabled (release version
   only)
9 #pragma config BOR = OFF //hardware brown out reset
10 #pragma config WDT = OFF //watchdog timer disabled
11 #pragma config MCLRE = OFF //MCLR pin enabled
12 //#pragma config LPT1OSC = ON //timer1 low power operation
13 #pragma config PBADEN = OFF //portB 0to 4 digital – not analogue
14 #pragma config LVP = OFF //low voltage programming disabled
15 #pragma config STVREN = ON //Stack full/underflow will not cause Reset.
16 #pragma config CCP2MX = OFF //CCP2 input/output is multiplexed with RB3.

```

Também é caracterizado o oscilador do projeto, conforme visto anteriormente o oscilador usado é o interno, no código seguinte estabelece que este *clock* interno apresentará frequência de 1MHz para realização das operações.

```

1 void config_osc () {
2     OSCCONbits.IRCF0=0;
3     OSCCONbits.IRCF1=0; //1MHz = XTAL_FREQ
4     OSCCONbits.IRCF2=1;

```

```

5 | OSCCONbits.IOFS=1; //INTOSC frequency is stable
6 | OSCCONbits.SCS1=1; //Internal oscillator
7 | }

```

Em seguida configurou-se as portas *I/O* do dispositivo, os pinos ligados ao *LCD* foram definidos como saídas lógicas, para que fique possível escrever no *display*, e no caso das portas de leitura dos fototransistores foram instituídas como entradas para permitir a identificação do estado do sensor.

```

1 | void config_io() {
2 |     // CONFIGURAO I/O
3 |     //1 - INPUT / 0 - OUTPUT
4 |     TRISCbits.TRISC0 = 0; //E s a da
5 |     TRISCbits.TRISC1 = 0; //RW s a da
6 |     TRISCbits.TRISC2 = 0; //RS s a da
7 |     //DADOS -> D0-D7 s a das
8 |     TRISAbits.TRISA1 = 0;
9 |     TRISAbits.TRISA2 = 0;
10 |    TRISAbits.TRISA3 = 0;
11 |    TRISAbits.TRISA4 = 0;
12 |    TRISAbits.TRISA5 = 0;
13 |    TRISEbits.TRISE0 = 0;
14 |    TRISEbits.TRISE1 = 0;
15 |    TRISEbits.TRISE2 = 0;
16 |
17 |    RW=0;
18 |    RS=0;
19 |    E=0;
20 |    D7=0;
21 |    D6=0;
22 |    D5=0;
23 |    D4=0;
24 |    D3=0;
25 |    D2=0;
26 |    D1=0;
27 |    D0=0;
28 | }

```

A interrupção do *firmware* cumpre a função de marcação de presença da bola quando um sensor percebe sua passagem, logo foi definida como sendo externa e de alta prioridade.

```

1 | void config_int() {
2 |     // CONFIGURAO INTERRUPO
3 |     RCONbits.IPEN = 1; //Enable priority levels on interrupts
4 |

```

```

5  INTCONbits.RBIE = 0; //Disables the RB port change interrupt
6  INTCON2bits.NOT_RBPU = 1; //All PORTB pull-ups are disabled
7
8  INTCONbits.GIE_GIEH = 1; //Enables all high-priority interrupts
9  INTCONbits.PEIE_GIEL = 1; //Enables all low-priority peripheral interrupts
10
11 INTCONbits.INT0IE = 0; //disables the INT0 external interrupt
12 INTCON3bits.INT1IE = 0; //disables the INT1 external interrupt
13 INTCON3bits.INT2IE = 0; //disables the INT2 external interrupt
14
15 INTCON2bits.INTEDG0 = 0; //Interrupt on falling edge
16 INTCON2bits.INTEDG1 = 0; //Interrupt on falling edge
17 INTCON2bits.INTEDG2 = 1; //Interrupt on rising edge
18
19 INTCON3bits.INT1IP = 1; //High priority
20 INTCON3bits.INT2IP = 1; //High priority
21 INTCON2bits.RBIP = 1; //High priority
22
23 PIE1 = 0x00; //disables all peripheral1 interrupts
24 PIE2 = 0x00; //disables all peripheral2 interrupts
25 }

```

E, por último, ajustou-se o *timer0* do *PIC* como um temporizador de *16-bits* para ser usado como o contador de tempo entre as marcações de presença da bola que percorre o túnel, estes instantes são guardados para uso nos cálculos posteriores.

```

1 void config_tmr() {
2     // CONFIGURA O TIMER0
3     TOCONbits.TMR0ON = 0; //timer0 module enabled
4     TOCONbits.T08BIT = 0; //16-bit timer
5     TOCONbits.T0CS = 0; //temporizador - clock source - Internal instruction cycle
        clock (CLKO)
6     TOCONbits.T0SE = 1; //Increment on high-to-low transition on T0CKI pin
7
8     TOCONbits.PSA = 1; //Timer0 prescaler is NOT assigned. Timer0 clock input
        DOESN'T come from prescaler output.
9     TOCONbits.T0PS2 = 0;
10    TOCONbits.T0PS1 = 1; //1:16 -> 62,5kHz
11    TOCONbits.T0PS0 = 1;
12
13    TMR0L = 0; // Zera bits menos sign timer 0
14    TMR0H = 0; // Zera bits mais sign timer 0
15

```

```

16 // CONFIGURA O INTERRUPTO DE TIMERS
17 INTCONbits.TMR0IE = 1; // Enables the TMR0 overflow interrupt
18 PIE1bits.TMR1IE = 0; // Disables the TMR1 overflow interrupt
19 PIE1bits.TMR2IE = 0; // Disables the TMR2 to PR2 match interrupt
20 PIE2bits.TMR3IE = 0; // Disables the TMR3 overflow interrupt
21
22 INTCON2bits.TMR0IP = 1; // High priority
23 IPR1bits.TMR1IP = 0; // Low priority
24 IPR2bits.TMR3IP = 0; // Low priority
25 }

```

### Configuração de LCD

Em seguida, trabalhou-se no funcionamento do *LCD*, consultando a tabela de comandos para inicializar o dispositivo corretamente, a configuração escolhida para a representação foi a de 8-bits e 2 linhas. O visor é energizado em seguida, juntamente do cursor que é posicionado na primeira posição de caractere do *display*.

```

1 void config_lcd() {
2     set_inst(0,0,1,1,1,0,0,0); // 8bit, 2 linhas, fonte 5*8
3     set_inst(0,0,0,0,1,1,0,0); // liga LCD, cursor ligado, pisca ligado
4     set_inst(0,0,0,0,0,1,1,0); // AC address 0 ddram
5     // D7=0;
6     // D6=0;
7     // D5=0;
8     // D4=0;
9     // D3=0;
10    // D2=0;
11    // D1=0;
12    // D0=0;
13    __delay_ms(100);
14    DELAY_L;
15 }

```

### Cálculo dos parâmetros

Para os cálculos de velocidade e aceleração desejados, trabalha-se com os instantes fornecidos pelo temporizador do *PIC18F4550* após um registro de bola, com estes valores aplica-se as seguintes equações:

$$V_{media(n)} = \frac{S_n - S_{n-1}}{T_n - T_{n-1}}$$

Onde  $V_{media(n)}$  é a velocidade registrada de uma dado sensor para o seguinte;  $n$  é a ordem de marcação dos sensores, considerando que todos os fotossensores são interrompidos na sequência de posicionamento;  $S_n$  é a distância entre 2 fotossensores, o valor é de 100mm para entre sensores vizinhos; e  $T_n$  é o instante de marcação do sensor  $n$  em segundos.

$$V_{total} = \frac{V_{media(1)} + V_{media(2)} + V_{media(3)}}{3}$$

Nesta equação,  $V_{total}$  é a média aritmética de velocidade das 3 velocidades  $V_{media(n)}$  medidas, este é o valor reproduzido no *LCD*.

$$A_{media(n)} = \frac{V_{media(n)} - V_{media(n-1)}}{T_n - T_{n-1}}$$

Neste caso,  $A_{media(n)}$  é a aceleração registrada de um par de sensores para o par seguinte;  $n$  é a ordem de marcação dos sensores, começando em 2 pois é neste momento que há a velocidade entre 2 sensores, também é considerando que todos os fotossensores são interrompidos na sequência de posicionamento; e  $T_n$  é o instante de marcação do sensor  $n$  em segundos.

$$A_{total} = \frac{A_{media(2)} + A_{media(3)}}{2}$$

Similarmente a equação de velocidade,  $A_{total}$  é a média aritmética de aceleração das 2 acelerações  $A_{media(n)}$  medidas, este é o valor reproduzido no *LCD*.

Na implementação da lógica nota-se que a posição do quarto sensor não é utilizado, logo a medição de  $V_{media(3)}$  usa  $S_3 - S_2 = 200\text{mm}$ . Uma adequação para a representação do resultado no *LCD* requer que uma multiplicação por 10000000000. As traduções destas equações e nuances para o *firmware* podem ser vistos no trecho de código apresentado a seguir.

```

1 void vel_med () {
2     // CLCULO VELOCIDADE
3     int i;
4     for (i=1;i<4;i++) {
5         if (i == 3) {
6             vel_data[i] = 100000000000/(float) time_data[i+1];
7             vel_data[i] = 200*vel_data[i];
8         }
9     else {
10        vel_data[i] = 100000000000/(float) time_data[i+1];
11        vel_data[i] = 100*vel_data[i];

```

```

12     }
13 }
14 for (i=1;i<4;i++) {
15     vel_tot = vel_tot + vel_data[i];
16 }
17 velocidade_med = vel_tot/3;
18 }
19
20 void acel_med() {
21     // CLCULO ACELERA O
22     int i;
23     for (i=1;i<3;i++) {
24         acel_data[i] = vel_data[i]-vel_data[i+1];
25         acel_data[i] = 1000000*acel_data[i];
26         acel_data[i] = acel_data[i]/(float)time_data[i+2];
27         acel_data[i] = 100*acel_data[i];
28     }
29     for (i=1;i<3;i++) {
30         acel_tot = acel_tot + acel_data[i];
31     }
32     aceleracao_med = acel_tot/2;
33 }

```

## Interface de Usuário

A interface de usuário foi implementada para que o usuário da ferramenta pude-se visualizar e interagir com o funcionamento do projeto. Para facilitar a escrita de mensagens foi desenvolvido a função *write\_string* conforme o trecho na sequência.

```

1 void write_string(unsigned char string []) {
2     int i;
3     for (i=0; string[i]!='\0'; i++)
4         set_data(string[i]);
5 }

```

Com esta função fica simples o envio de mensagens para o *LCD*, com isso as mensagens de inicialização abaixo foram elaboradas para sinalizar o usuário sobre o versionamento do *firmware* e sobre o tempo dos processos iniciais. Toda vez que o microcontrolador tiver seu *reset* ativado ou a alimentação interrompida e restaurada, estas são as mensagens apresentadas.

```

1 void msg_init() {
2     clear_scr();

```



```
3   write_string("_WARTHOG_ROBOTICS");
4   jump_line();
5   write_string("_EESC_-_USP");
6   delay_sec(2);
7   clear_scr();
8   write_string("_WRTUNNEL_4.0");
9   jump_line();
10  write_string("_PROTOTIPO");
11  delay_sec(2);
12  clear_scr();
13  write_string("Configurando");
14  delay_sec(1);
15  for(int i=0;i<6;i++) {
16      write_string(".");
17      delay_sec(0.25);
18      if(i==2){
19          clear_scr();
20          write_string("Configurando");
21      }
22  }
23 }
```



Figura 3.10: Primeira mensagem de inicialização. Fonte: Elaborada pelo autor.



Figura 3.11: *Segunda mensagem de inicialização.* Fonte: Elaborada pelo autor.



Figura 3.12: *Tempo para configuração do microcontrolador e do LCD.* Fonte: Elaborada pelo autor.



Figura 3.13: Mensagem para indicação de disponibilidade de uso. Fonte: Elaborada pelo autor.



## Capítulo 4

# Resultados e Discussões

Neste capítulo são exibidos e discutidos os resultados dos métodos apresentados na Seção 3.2 com o material explicitado na Seção 3.1, comentando-se também as decorrências que não estavam previstas e desfechos errôneos. Ao final comenta-se sobre o cronograma estimado e real do projeto.

### 4.1 Mecânica

Para testar o desempenho da estrutura mecânica, observou-se o comportamento apresentado após várias baterias de testes por diferentes pessoas em um espaço de 1 mês, incluindo usos para calibrações de chute legítimos e verificações com o intuito de testar a robustez do túnel, ambos utilizando o chute de um robô *SSL*.

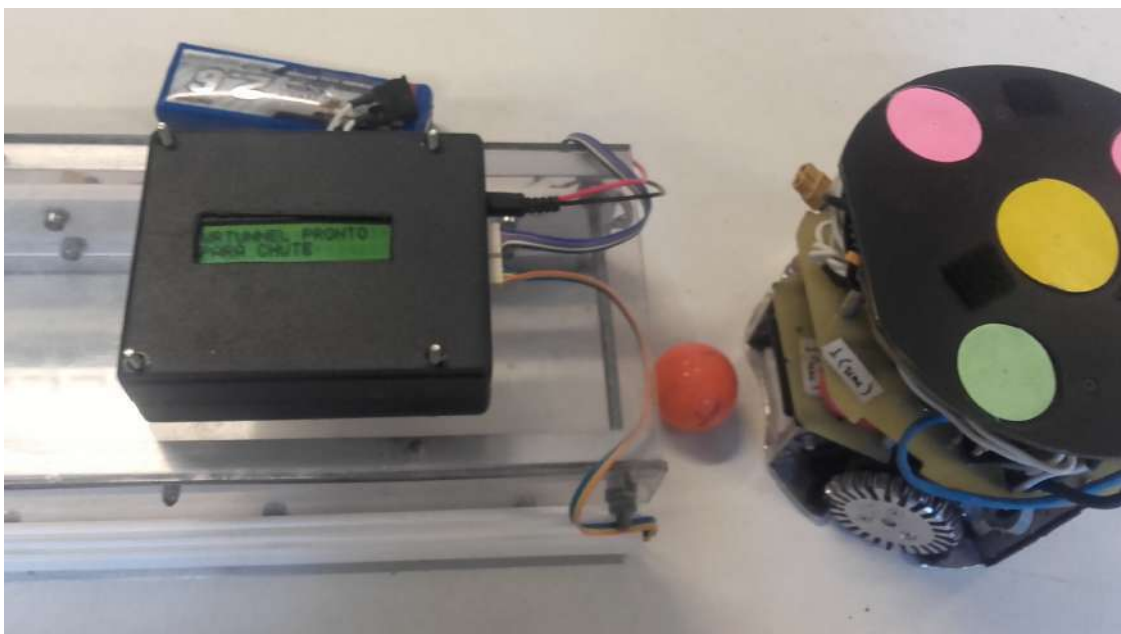


Figura 4.1: *Configuração geral dos testes de robustez mecânica.* Fonte: Elaborada pelo autor.

### 4.1.1 Robustez

Em testes específicos para a robustez da estrutura, utilizou-se chutes de velocidade conhecida para a leitura nos mais diversos ambientes como método de controle para averiguar seu desempenho. Primeiro foi checado sua resistência, chutes fortes em colisão de até 12m/s com as paredes não apresentaram qualquer danos visíveis, quando em impacto com os bocais dos fotossensores notou-se que estas ficam ligeiramente lascadas, como evidenciado na Figura 4.7. O funcionamento dos sensores foi afilido apenas em um momento devido a uma lasca do bocal que cobriu um dos *LEDs* infravermelhos, após retirado a obstrução voltou a fazer a leitura.



Figura 4.2: *Bocal de fototransistor danificado pelo chute.* Fonte: Elaborada pelo autor.

Quanto a iluminação, o acrílico transparente não parece influenciar as leituras dos fototransistores receptores mesmo em ambientes de forte iluminação, duas instâncias dos mesmos parâmetros de chute estão registrados nas Figuras 4.3 e 4.4, onde tem-se um leitura em ambiente com pouca iluminação e outra num local com bastante luminosidade. O único caso de interferência constatado ocorreu com a radiação deliberada de luz infravermelha nas imediações do túnel, entretanto apenas quando esta fonte de luz não estava por fora do acrílico e a cerca de 200mm de um dos receptores.



Figura 4.3: *Leitura em ambiente com iluminação fraca.* Fonte: Elaborada pelo autor.



Figura 4.4: *Leitura em ambiente fortemente iluminado.* Fonte: Elaborada pelo autor.

#### 4.1.2 Peso

No manuseio das partes do projeto, ficou evidente que a armação mecânica compõe a maior parte do peso do sistema, utilizando um balança de tração o peso total do conjunto foi de 1,36kg aproximadamente, cerca de 1,15kg provenientes da estrutura mecânica em si incluindo a caixa do PCB vazia e

sem os fotossensores com sua fiação. O peso do túnel acrílico e adições não apresentaram qualquer empecilho no funcionamento do projeto, inclusive fez com que a estrutura mantivesse sua posição mesmo com a batida da bola em suas partes internas.

## 4.2 Eletrônica

A eletrônica do projeto é considerada como o PCB, fotossensores e a fiação geral, a configuração de teste do desempenho eletrônico foi similar ao mecânico, apresentado na Figura 4.1, usando-se um robô *SSL* com chutes de velocidade conhecida para facilitar a análise dos resultados.

### 4.2.1 PCB

Ao longo do processo de manufatura o funcionamento dos módulos do *PCB* projetado foram verificados rigorosamente, com a confirmação da operação dos componentes centrais, começou-se o testes definitivos com chutes de um robô *SSL*. A *PCB* manteve desempenho constante durante diversos chutes, não apresentando qualquer avariação seja por diferentes velocidades de chute ou por impactos da bola com a estrutura mecânica. Estava fixada na caixa plástica sem qualquer possibilidade de movimentação, os periféricos com necessidade de abertura encaixaram com as brechas feitas no envólucro.

### 4.2.2 Fotossensores

Os fotossensores foram testados tanto no desempenho quanto na robustez, com chutes de maior e menor velocidades possíveis e algumas leituras com o objetos que não fossem a bola, em especial bloqueando os sensores usando-se o braço.

Para a robustez, notou-se que os bocais utilizados impediam qualquer danos aos *LEDs*, mesmo sendo lascados pelas colisões, permitindo um funcionamento normal salvo um caso na qual uma lasca de bocal bloqueou a luz infravermelho de um dos pares de sensores.

No desempenho dos fotossensores, usou-se os chutes mais fracos e os mais fortes para verificar o comportamento da leitura, o chute mais fraco registrado foi de aproximadamente 1,43m/s como visto na Figura 4.5 e para a mais alta foi de cerca de 12,35m/s como na Figura 4.6. O caso mais lento onde esperava-se 10 segundos para marcar cada *LED* obtê-se cerca de 0,01m/s, em tentativas de bloquear todos os fotossensores ao mesmo tempo atingiu-se velocidades próximas de 438m/s. Entretanto os limites teóricos do sistema encontram-se a seguir:

**Velocidade: 0,01 a 568,18 m/s**



Mínimo - Tempo máximo de espera de leitura entre pares de sensores: 10s  $\Rightarrow v = 0,1/10 \Rightarrow v = 0,01$ ;

Máximo - Simulador do MPLAB: 352 instruções entre leituras de velocidade (500ns/instrução)  
 $\Rightarrow v = 0,1/(352*500n) \Rightarrow v = 568,18m/s$ ;

**Aceleração: -9999999 a 9999999 m/s**

Calculado em cima das velocidades aferidas (apenas 7 caracteres são exibidos);



Figura 4.5: Menor velocidade obtida com chutes de um robô SSL. Fonte: Elaborada pelo autor.

### 4.2.3 Fiação

Testando-se a fiação junto nas baterias de teste dos sensores e do PCB, não foi verificado quaisquer mal-contatos ou falhas de comunicação. Os cabos também não apresentaram qualquer empecilho para a leitura ou na configuração montada.



Figura 4.6: *Maior velocidade obtida com chutes de um robô SSL.* Fonte: Elaborada pelo autor.

### 4.3 Firmware

Com o intuito de verificar a integridade dos códigos do *firmware* foi utilizado o *LCD* do *PCB* logo quando se obteve a confirmação de seu funcionamento, para imprimir as trajetórias, estados e valores presentes no código desenvolvido, permitindo uma noção sólida do comportamento que o *firmware* apresentou.

#### 4.3.1 Inicialização

O microcontrolador *PIC18F4550* inicializou conforme as configurações determinadas, no caso apresentou o *clock* interno de 1MHz, os pinos *I/O* para o *LCD* foram definidos como saídas e a comunicação com os fototransistores como entradas e a interrupção *timer0* se comportou conforme um temporizador de *16-bits*. O *LCD* também teve suas configurações definidas expressas, os caracteres mostrados estavam no modo 2 linhas, demonstrando as mensagens de início definidas ao se ligar o sistema.

#### 4.3.2 Tratamento das leituras

As únicas leituras desempenhadas pelo *PIC* são as dos fototransistores conectados nas portas definidas como *input*, considerando-se com as interrupções ocorrendo conforme o esperado verifica-se os valores obtidos. Os resultados após o tratamento impressos no *LCD* aparentam possuir grandeza con-

dizente com os chutes feitos, ao usar-se velocidades conhecidas o valores mostram-se extremamente similares ao esperado.

### 4.3.3 Representação dos dados

A impressão das mensagens e dos dados finais no *LCD* ocorreu exatamente como codificado no microcontrolador, mantendo a ordem de grandeza e as casas decimais definidas.

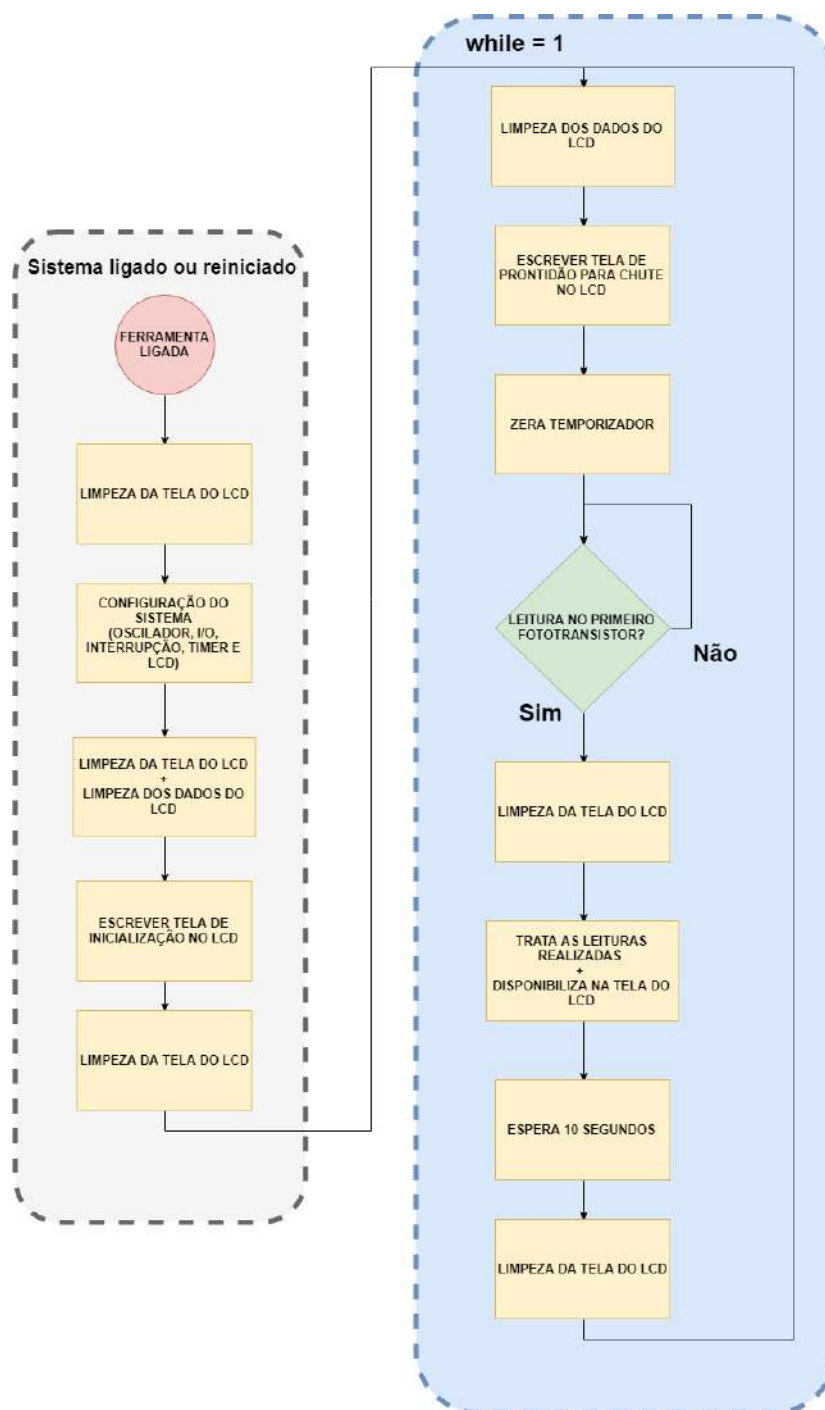


Figura 4.7: Lógica de firmware implementado. Fonte: Elaborada pelo autor.



## Capítulo 5

# Conclusão

Ao final dos testes e análises realizados, é seguro afirmar que o projeto em seu estado vigente é funcional e já permite aplicação em seu objetivo principal - disponibilizar para o usuário as leituras de velocidade e aceleração da bola chutada por um robô SSL. Embora ainda existam funcionalidades para serem implementadas e outras utilidades que podem ser desenvolvidas ou corrigidas, o projeto já está consolidado o suficiente para sua viabilização.

Na questão da mecânica, o sistema final demonstrou-se completamente satisfatório e confiável. Possuindo peso e robustez suficiente para aguentar os impactos de chutes fortes e ao mesmo tempo sendo prático e viável de ser carregado e portado pelo usuário. O *design* simples e bem manufaturado também facilita e incrementa na experiência de uso da ferramenta.

A parte de realização de medidas pelos fotossensores é altamente precisa e robusta, não havendo quaisquer desvios de operação nos testes e validações finais, independente de iluminações ou ângulos de chute. Única desvantagem percebida é o alto consumo de energia pelos LEDs emissores, podendo-se selecionar componentes com menos intensidade de emissão em atualizações futuras.

Quanto ao *hardware* eletrônico, o circuito mostrou-se capaz de realizar as funcionalidades requeridas sem empecilhos, e somado a sua simplicidade e custo de manufatura barato, determinou-se que possui atuação satisfatória. Sua manufatura e integração com a mecânica permitiu a garantia de robustez e praticidade final do projeto. A porta USB adicionada, embora não possua funcionalidade implementada, poderia ser utilizada para captura de medidas em comunicação com um PC.

Em relação ao *firmware*, o essencial foi concluído, permitindo que o usuário acesse o valor de suas medições sem quaisquer dificuldades. No entanto, este valor não pode ser salvo, nem exportado via USB, embora a estrutura para desenvolver tais utilidades exista no código.

É notável a evolução que o trabalho provocou, notando-se algumas incertezas quanto ao desenvolvimento do projeto, em especial em *firmware*, foi possível chegar a resultados que pareciam ser

difíceis de se obter inicialmente. Com todo o conhecimento desenvolvido e experiência adquirida para execução de um projeto em engenharia, é possível realizar um projeto deste porte novamente em um período de tempo menor e com procedimentos mais eficazes, conhecidos durante a tentativa e erro na execução do projeto presente nesta monografia.

Uma proposta para futuras iterações é a de desenvolver um sistema de fotossensores mais econômicos energeticamente, visto que os presentes no sistema possuem intensidade de brilho superdimensionada. O *hardware* eletrônico pode ser incrementado introduzindo-se botões e LEDs de sinalização ligados ao microcontrolador, permitindo ao usuário uma interação ainda maior com a ferramenta. Também seria necessário fazer os testes de medida para identificar a variação do tempo de ciclo de instrução em função da imprecisão do oscilador interno, assim como fazer uma análise em relação a temperatura de operação que foi medida para o procedimento de aferição

Para o *firmware* existem diversas melhorias e incrementações possíveis, como a criação de perfis para salvar medidas de chute e exportação de dados via *USB*, o que deixa necessária a criação de menus para navegação e seleção de opções. As melhorias observadas para a mecânica são a de instalação de alças para o manuseio mais ergonômico do projeto e a troca dos bocais dos fotossensores para materiais mais resistentes aos impactos da bola, com maior vida útil.

## Referências

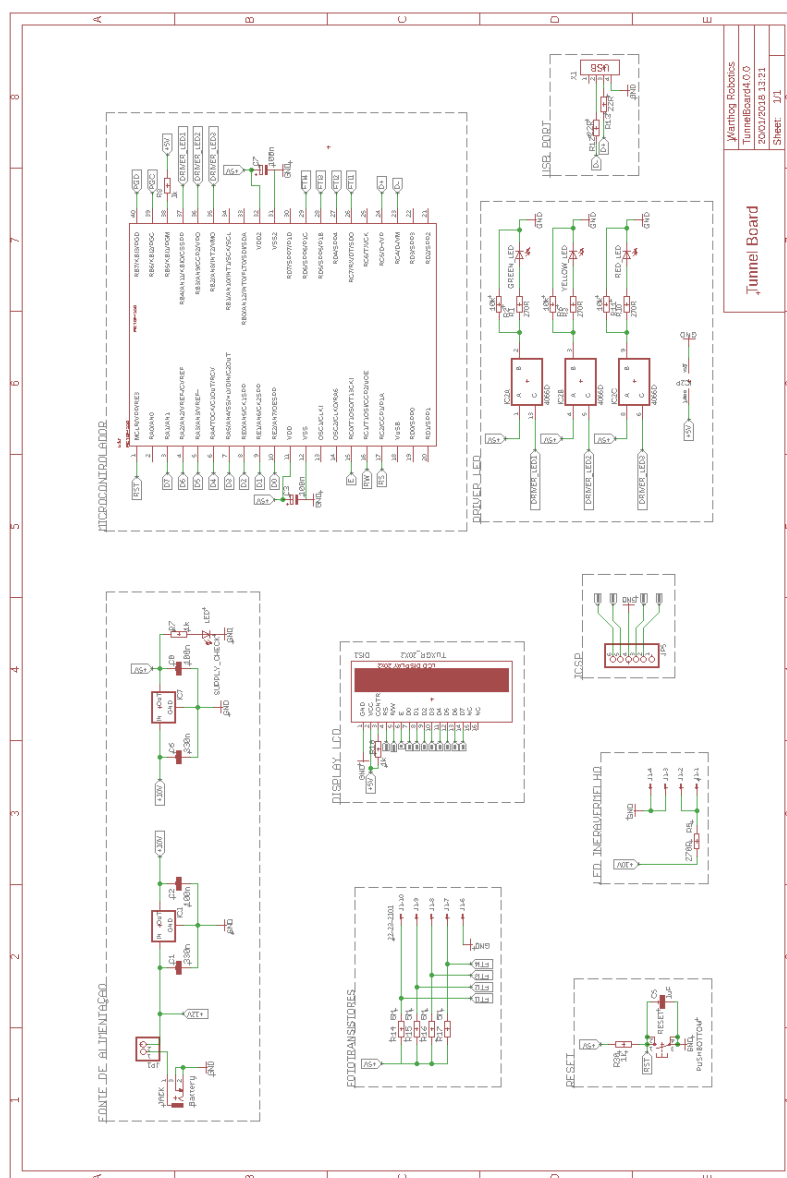
- [1] Luhtanen P. *Handbook of Sports Medicine and Science*. Oxford: Blackwell Scientific Publications, 2000.
- [2] Bangsbo J. Energy demands in competitive soccer. *Journal of Sports Sciences*, 1994.
- [3] James F Cox. *Fundamentals of linear electronics: integrated and discrete*. Cengage Learning, 2002.
- [4] Gerd Keiser. *Optical fiber communications*. McGraw-Hill, 2008.
- [5] Carmen Gonzalez and Antoine Marty. Phototransistors. *Optoelectronic Sensors*, 2009.
- [6] Eduardo Aguilar Pelaez and Esther Rodriguez Villegas. Led power reduction trade-offs for ambulatory pulse oximetry. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. IEEE, 2007.
- [7] Muhammad Ali Mazidi, Rolin D McKinlay, Danny Causey, and PIC Microcontroller. *Embedded Systems*. Pearson, New Jersey, 2008.
- [8] Luigi Carro and Flávio Rech Wagner. Sistemas computacionais embarcados. *Jornadas de atualização em informática. Campinas: UNICAMP*, 2003.
- [9] Shunsuke Kobayashi, Shigeo Mikoshiba, and Sungkyoo Lim. *LCD backlights*. John Wiley & Sons, 2009.
- [10] Ed Sutter. *Embedded systems firmware demystified*. CMP books, 2002.
- [11] Steve Heath. *Embedded systems design*. Elsevier, 2002.





# Apêndice A

## Schematic do PCB TunnelBoard





## Apêndice B

### *Layout do PCB TunnelBoard*

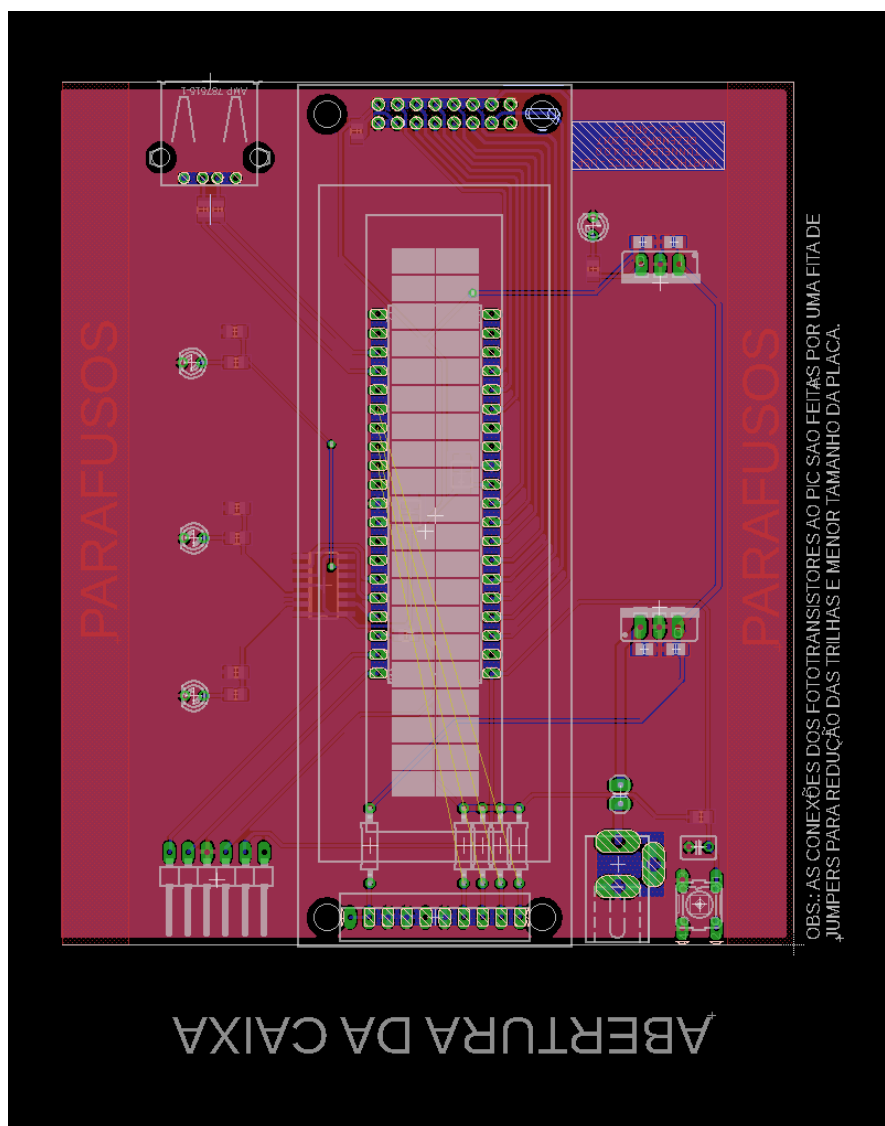


Figura B.1: Board desenvolvido no CADSoft EAGLE da AUTODESK. Fonte: Elaborada pelo autor.

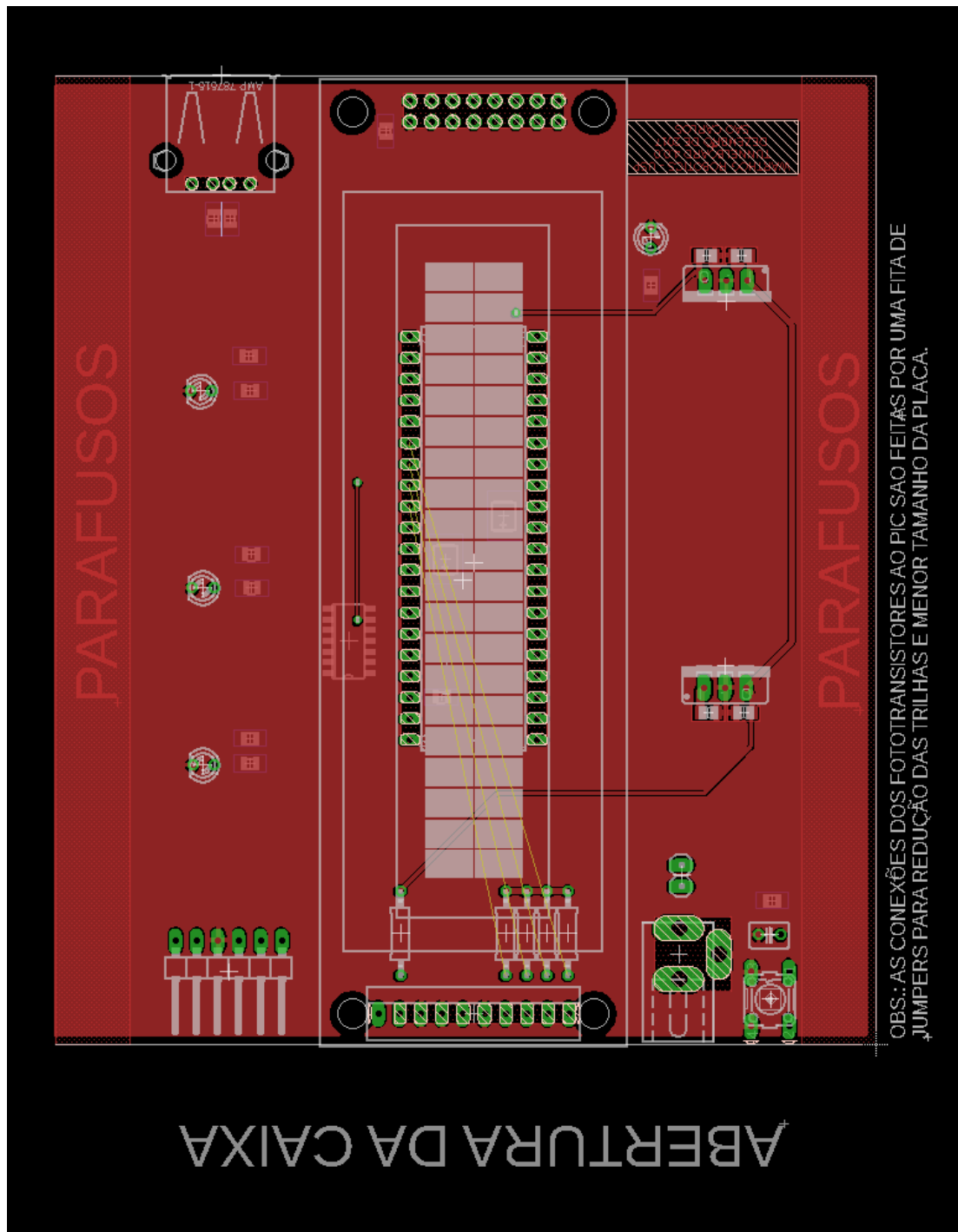


Figura B.2: Board desenvolvido no CADSoft EAGLE da AUTODESK. Fonte: Elaborada pelo autor.

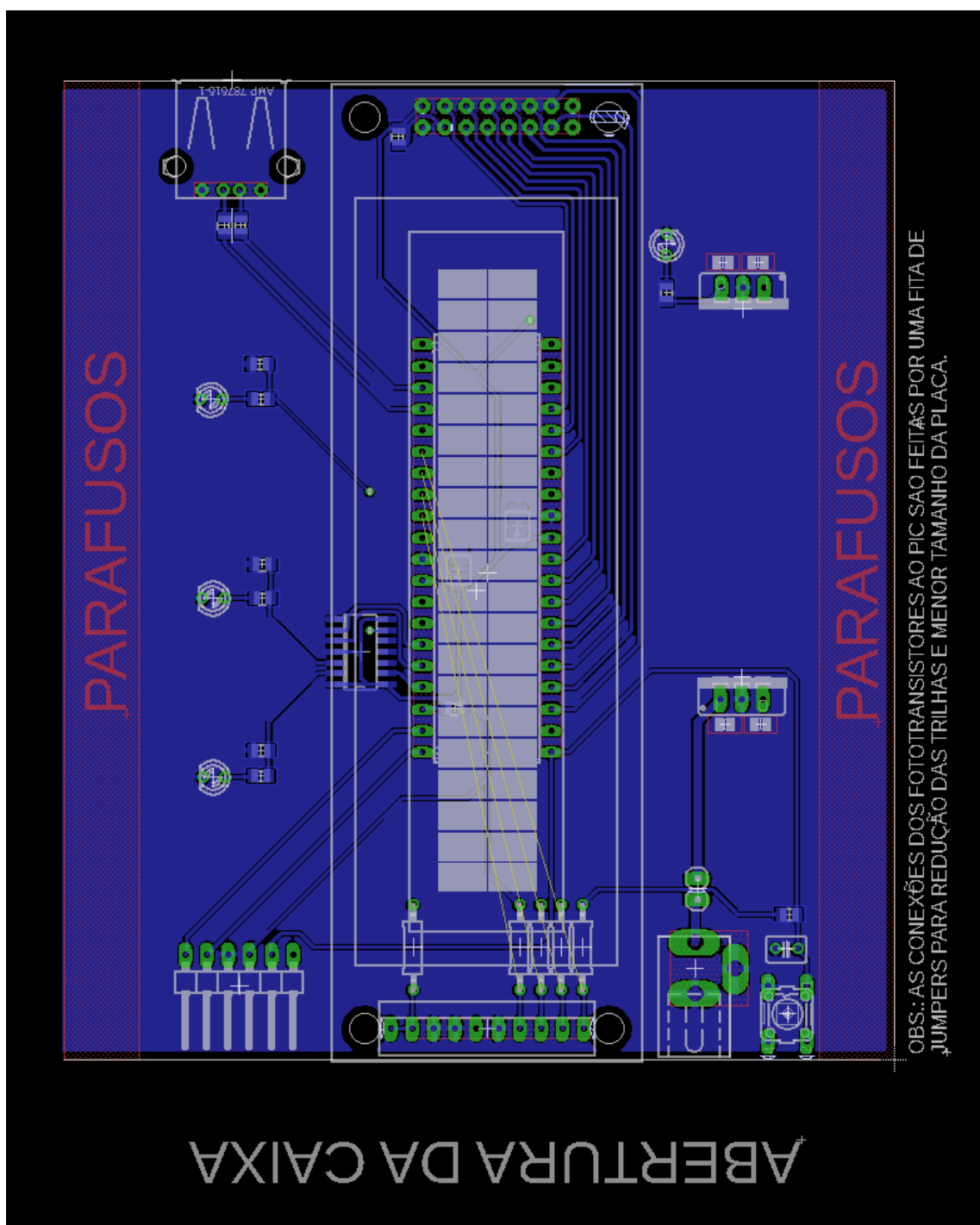


Figura B.3: Board desenvolvido no CADSoft EAGLE da AUTODESK. Fonte: Elaborada pelo autor.



## Apêndice C

# Firmware desenvolvido para a *TunnelBoard*

Devido ao extenso tamanho do código desenvolvido para o projeto, deixa-se indicado o repositório aonde o *firmware* da ferramenta pode ser consultado: <https://github.com/vitor-gomes96/tunnelboard>