

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Campus DIVINÓPOLIS
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

Havilah Lucas Vasconcelos Ramos
William Soares Souza

PLATAFORMA ROBÓTICA REMOTAMENTE CONTROLADA
PARA AQUISIÇÃO DE IMAGENS 360°

Divinópolis
2017

Havilah Lucas Vasconcelos Ramos
William Soares Souza

PLATAFORMA ROBÓTICA REMOTAMENTE CONTROLADA
PARA AQUISIÇÃO DE IMAGENS 360°

Monografia de Trabalho de Conclusão de Curso apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.

Áreas de integração: Mecânica, Eletrônica, Computação.

Orientador: M.e Lucas Silva de Oliveira
Supervisor: Bel. Mateus Nunes Costa

Divinópolis
2017

Havilah Lucas Vasconcelos Ramos
William Soares Souza

PLATAFORMA ROBÓTICA REMOTAMENTE CONTROLADA
PARA AQUISIÇÃO DE IMAGENS 360°

Monografia de Trabalho de Conclusão de Curso apresentada ao Colegiado de Graduação em Engenharia Mecatrônica como parte dos requisitos exigidos para a obtenção do título de Engenheiro Mecatrônico.

Áreas de integração: Mecânica, Eletrônica, Computação.

Comissão Avaliadora:

Prof. M.e Lucas Silva de Oliveira

Departamento de Engenharia Mecatrônica
CEFET-MG *Campus V*

Prof. D.r Wagner Custódio de Oliveira

Departamento de Engenharia Mecatrônica
CEFET-MG *Campus V*

Prof. Bel. Adriano Nogueira D. Lopes

Departamento de Engenharia Mecatrônica
CEFET-MG *Campus V*

Divinópolis
2017

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Campus DIVINÓPOLIS
GRADUAÇÃO EM ENGENHARIA MECATRÔNICA

Monografia intitulada “PLATAFORMA ROBÓTICA REMOTAMENTE CONTROLADA PARA AQUISIÇÃO DE IMAGENS 360°”, de autoria dos graduandos Havilah Lucas Vasconcelos Ramos e William Soares Souza, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Me. Lucas Silva de Oliveira - Orientador
CEFET/MG *Campus V*

Prof. Dr. Wagner Custódio de Oliveira
CEFET/MG *Campus V*

Prof. Bel. Adriano Nogueira D. Lopes
CEFET/MG *Campus V*

Prof. Dr. Lúcio Flávio Santos Patrício
Coordenador do Curso de Engenharia Mecatrônica
CEFET/MG *Campus V*

Divinópolis
2017

AO MEU TIO MARCUS (*In memorian*) SEMPRE AO MEU LADO, PROVENDO FORÇA E RESILIÊNCIA PERANTE AS ADVERSIDADES. E AOS MEUS PAIS PELO APOIO INCONDICIONAL, APESAR DE MINHA AUSÊNCIA CONSTANTE. DESEJO SER MERECEDOR DE SEUS SACRIFÍCIOS E MOTIVO DE ORGULHO EM TODOS OS ASPECTOS DE MINHA VIDA.

AOS MEUS PAIS, POR TODO CARINHO, APOIO E INCENTIVO, SEM OS QUAIS EU NÃO TERIA CHEGADO ATÉ ESSA ETAPA DE MINHA VIDA. E A TODOS MEUS AMIGOS, QUE TORNARAM A CAMINHADA MAIS FÁCIL E AGRADÁVEL.

Agradecimentos

Agradeço,

à minha família por prover a estrutura para que eu pudesse completar o trabalho, bem como minha graduação. Ao AWC, em particular aos nossos coordenadores durante o ano, pelo auxílio, financiamento e mentoria visando o enriquecimento de nosso trabalho e sua aplicação efetiva no mercado imobiliário. Agradeço a todos os companheiros de curso que de alguma forma contribuíram para a execução do projeto, e ao nosso orientador, quem nos direcionou nos momentos de dúvidas.

Agradeço,

à minha família por todo o apoio e incentivo durante toda a graduação e o desenvolvimento desse projeto. Ao instituto Tim, ao núcleo de empreendedorismo da USP e todos os envolvidos no programa AWC, pelo financiamento do projeto e mentoria, que não apenas viabilizaram a implementação de nossa ideia, como nos proporcionaram ensinamentos importantes e nos guiaram a fim de transformar a ideia inicial em um produto atrativo e mercadologicamente viável. Agradeço ao meu parceiro de projeto e ao nosso orientador, que foram fundamentais para a obtenção do resultado final, além de vários de nossos amigos que também contribuíram com seus conhecimentos, dicas e opiniões durante o desenvolvimento do projeto.

Estamos todos na sarjeta, mas alguns de nós estão
olhando para as estrelas...

Oscar Wilde

Não se preocupe com o que todos os outros
vão fazer, a melhor forma de prever o futuro é
inventando-o.

Alan Kay

Resumo

Atualmente, ambientes internos são apresentados por consultores imobiliários e rede hoteleira ao público por meio de fotografias comuns. Esse procedimento comum do setor, apesar de apresentar o ambiente ao cliente em potencial, não facilita a compreensão de características relevantes do espaço, como noção de profundidade, dimensões e espaçamento, prejudicando, assim, a avaliação do ambiente. A solução mais comum até então, consiste na visita do interessado ao local, o que resulta em despesas com o deslocamento e tempo despendido. A fim de minimizar essas despesas e proporcionar mais conforto ao cliente durante o reconhecimento do ambiente, foi proposto o desenvolvimento de um sistema robótico, controlado remotamente, que possibilita a aquisição de dados e imagens para construção de um *tour* virtual do ambiente. Para tanto, objetivou-se o projeto e construção de um robô diferencial, ao qual foi acoplado uma câmera de aquisição de imagens 360°, um sistema de transmissão de dados sem fio teleoperado pelo usuário. As aplicações do equipamento são consideravelmente amplas, sendo útil em processos em que se deseja apresentar uma melhor representação do espaço físico ao cliente. Logo, o dispositivo desenvolvido realiza a substituição de tripés de câmeras e homem/hora para deslocar entre as posições das fotografias, sendo necessária para sua implementação a integração das quatro áreas da Engenharia Mecatrônica: mecânica, eletrônica, programação e controle. A mecânica está presente no projeto e construção da plataforma do robô. A eletrônica, referente à toda instrumentação, sistemas microcontrolados e transmissão *wireless* de dados. A computação, encontra-se em todos os aspectos, mas principalmente no processamento de imagens e criação dos *softwares* de controle e armazenamento, além da programação do sistema robótico. E por último, o controle para facilitar a manipulação do veículo por parte do operador, garantindo a precisão dos movimentos e que os atuadores funcionem de forma desejável. Ao final do projeto foi obtido um protótipo funcional de fácil operação capaz de capturar imagens 360°, que possibilitam a criação de um *tour* virtual proporcionando uma experiência visual orgânica e imersiva. Dessa forma, as escolhas dos clientes dos mercados imobiliário e de turismo poderão ser simplificadas, melhorando também as vendas e atratividade destes negócios perante seu público.

Palavras-chave: Robótica Móvel, Mapeamento Tridimensional, *Tour* Virtual.

Abstract

Nowadays, indoor environments are presented by real estate consultants and hotel chains to its public through the use of common pictures. This procedure, typical in the sector, despite presenting the environment to the potential customer, does not facilitate the comprehension relevant features in the space, such as the notion of depth, dimensions and spacing, and thus, impairing the understanding of the place of interest. The most usual solution until then, consists of the visitation of the site by the individual, which results on expenses with logistics and wasted time. In order to minimize these costs and provide more confort to the customer during the environment's recognition, the development of a remotely controlled robotic system was proposed, which allows the acquisition of data and images to the construction of virtual tours of the ambient. Therefore, it was objectified the project and construction of a differential mobile robot, which was coupled with a 360° camera, a wireless data transmitting system, and being teleoperated by the user. The applications of this kind of equipment are considerably vast, being useful on processes that a better presentation of the physical space to the client is desired. So, the developed device substitutes the need of tripods and men/hours of moving from on position to another between the pictures, needing the integration of the four areas of Mechatronics Engineering: mechanics, electronics, programming and control. The mechanics portion regards the project and construction of the robot's platform. The electronics fraction refers to all the instrumentation present, microcontrolled systems and wireless data transmission. Computation is present in all aspects, but mainly on imaging processing, control and storage software development and on the robotic operational system programming. Lastly, the controle portion is used to facilitate the vehicule's handling for the operator, granting a satisfactory movement precision and desirable actuators' performance. By the end of the project, a functional prototype with easy operation was obtained, being capable of taking high quality 360° pictures that allows the creation of virtual tours, providing an organic and immersive visual experience. This way, client's choices in the real estate and tourism markets will be simplified, also increasing sales and attractiveness of these business towards its public.

Key-words: Mobile Robotics, Three-dimensional Mapping Virtual Tour.

Sumário

Lista de Figuras	xviii
Lista de Tabelas	xix
Lista de Acrônimos e Notação	xx
1 Introdução	1
1.1 Definição do Problema	3
1.2 Motivação	4
1.3 Objetivos do Trabalho	4
1.3.1 Objetivo geral	4
1.3.2 Objetivos específicos	5
1.4 Estado da arte	5
1.5 Revisão de Literatura	9
1.6 Escopo do trabalho	12
2 Fundamentação Teórica	13
2.1 Locomoção de robôs móveis	13
2.1.1 Principais aspectos da locomoção	16
2.2 Morfologia de robôs móveis com rodas	17
2.2.1 Rodas	17
2.2.2 Topologia de robôs móveis	18
2.3 Atuadores	20
2.4 Mecanismo <i>Pan-Tilt</i>	22
2.5 Microcontroladores e Microprocessadores	23
2.6 <i>System on a Chip</i> — SoC	24
2.7 Modulação por Largura de Pulsos — PWM	24
2.8 Ponte H	26
2.9 Encoder de Quadratura	27
2.9.1 Conversão da leitura do <i>encoder</i> em movimento do robô	28
2.10 Regulador de tensão	30
2.11 Sensor Ultrassônico	33
2.12 Comunicação I ² C	34
2.13 Servidor Web	36
2.14 XMLHttpRequest	36
2.15 <i>Common Gateway Interface</i>	37

3	Projeto	39
3.1	Especificações de Projeto	39
3.2	Mecânica	40
3.2.1	Topologia do robô	40
3.2.2	Dimensionamento do motor	41
3.2.3	Suporte e Fixação	42
3.3	Tubo telescópico	44
3.4	Esquemático do <i>Hardware</i>	46
3.5	Sensores	47
3.5.1	Câmera de Aquisição de Imagens 360°	47
3.5.2	Sensor Ultrassônico	50
3.6	Hardware	52
3.6.1	Opções de Mercado	52
3.6.2	Raspberry Pi	53
3.6.3	Células de Energia	56
3.7	Circuitos Eletrônicos	57
3.7.1	Regulador de Tensão	58
3.7.2	Acelerômetro	60
3.7.3	Comunicação I ² C	62
3.7.4	Placa do Microcontrolador	64
3.7.5	Sensores Ultrassom	65
3.7.6	Placa para os Motores	66
3.8	Programação	67
3.8.1	Servidor Web	67
3.8.2	Página Web	67
3.8.3	CGI e XMLHttpRequest	68
3.8.4	Movimentação do Robô	70
3.8.5	Código do Microcontrolador	71
4	Resultados	73
4.1	Topologia do robô	73
4.2	Plataforma Mecânica	73
4.3	Sensores	77
4.3.1	Câmera 360°	77
4.3.2	<i>Encoders</i>	78
4.3.3	Ultrassom	79
4.4	Motores	80
4.4.1	Controle dos Motores	82
4.5	Células de Energia	83
4.6	Circuitos Eletrônicos	84
4.6.1	Simplificações para a parte eletrônica	84
4.6.2	Placas eletrônicas	88
4.7	Programação	90
4.8	Integração dos Subsistemas	92
4.9	Plataforma de Visitas Virtuais	92
5	Conclusões	96
5.1	Considerações Finais	97
5.2	Propostas de Trabalhos Futuros	99

A	Códigos	100
A.1	Código dos Sensores	100
A.2	Código de movimentação retilínea	101
A.3	Código do Arduino	102
B	Desenhos técnicos	106
	Referências	109

Lista de Figuras

1.1	Amazon Prime Air (AMAZON (2016)).	6
1.2	Alguns dos robôs da Boston Dynamics.	7
1.3	As eMotionButterflies da Festo (FESTO (2015)).	7
1.4	O carro autônomo da Tesla (MEIO BIT (2016)).	8
1.5	Robô Elsie (HOLLAND (2003))	10
1.6	Robô Curiosity desenvolvido pela NASA (NASA (2015)).	11
1.7	Robô Asimo desenvolvido pela Honda (RIDERS (2014))	12
2.1	Mecanismos de locomoção usados em sistemas biológicos (SIEGWART; NOURBAKSH; SCARAMUZZA (2011)).	14
2.2	Robô Athlete da NASA (NASA (2017)).	15
2.3	Energia <i>versus</i> velocidade obtida para vários mecanismos de locomoção. Adaptado. (SIEGWART; NOURBAKSH; SCARAMUZZA (2011)).	16
2.4	Os quatro tipos básicos de rodas.	17
2.5	Configurações das rodas em robôs móveis; Adaptado. (SIEGWART; NOUR- BAKSH; SCARAMUZZA (2011)).	19
2.6	Esquema de uma câmera <i>pan-tilt</i> e seus eixos de rotação. (KIKUCHI (2007))	23
2.7	Variação do <i>duty cycle</i> de um sinal PWM. (ARDUINO (2008))	25
2.8	Polarização do motor através da Ponte H. (MAGAR (2016))	26
2.9	Encoder e seus principais componentes. (NATIONAL INSTRUMENTS (2013))	27
2.10	Saídas de um encoder de quadratura. (UNO ROBÓTICA (2012))	28
2.11	Rotação em torno do eixo do robô. (UNO ROBÓTICA (2012))	29
2.12	Transistor de chaveamento no período ativo. Adaptado. (COATES (2017))	31
2.13	Transistor de chaveamento no período desativado. Adaptado. (COATES (2017))	31
2.14	Comportamento das tensões e correntes de um buck ideal operando em modo contínuo (WOW (2017)).	32
2.15	Comportamento das tensões e correntes de um buck ideal operando em modo descontínuo (WOW (2017)).	33
2.16	Módulo HC-SR04. Adaptado. (FREAKS (2017)).	33
2.17	Distribuição prática do alcance do módulo HC-SR04. (SAPKOTA (2012)).	34
2.18	Estrutura de mensagem da comunicação I ² C. Adaptado. (BASICS (2017)).	35
2.19	Ilustração do acesso de um servidor Web por um navegador. Adaptado. (MOZILLA (2017)).	36

2.20	Ilustração do funcionamento de um XMLHttpRequest. Adaptado. (OVERFLOW (2012)).	37
2.21	Visão geral de um sistema Web utilizando uma CGI. (OLIVEIRA C. NEVES E. (2000)).	38
3.1	Rodas de 100 mm de diâmetro (POLULU (2017)).	40
3.2	Vista explodida da montagem do adaptador de fixação. (POLULU (2017))	44
3.3	Simulação da montagem do motor, roda e suportes.	44
3.4	Diagrama esquemático da base do robô.	45
3.5	Manfrotto Compact Extreme (MANFROTTO (2016)).	45
3.6	Esquemático do <i>Hardware</i> do robô.	46
3.7	Ricoh Theta S (RICOH (2017)).	48
3.8	Samsung Gear 360 (SAMSUNG (2016)).	48
3.9	Panono Camera (PANONO (2017)).	49
3.10	Câmera Insta360 4K (SIXTY (2016)).	50
3.11	Esquemático de posicionamento e alcance do sensor ultrassônico.	51
3.12	Esquemático de posicionamento e alcance dos sensores laterais.	51
3.13	Configuração final dos sensores ultrassom evidenciando o alcance dianteiro.	52
3.14	Comparativo entre embarcados.	53
3.15	Features do Raspberry Pi 3 (ELEMENT14 (2016)).	54
3.16	Raspberry Pi 3 GPIO Header (ELEMENT14 (2015)).	55
3.17	Modelo de bateria Gens Ace 5200 mAh, 15C, 3S LiPo (MERCADO LIVRE (2017a)).	57
3.18	Circuito regulador de tensão proposto com o XL4015.	58
3.19	Regulador de tensão XL4015 (MERCADO LIVRE (2017b)).	59
3.20	Esquemático do circuito com o regulador de tensão.	59
3.21	O sistema de ângulos de rotação <i>Roll-Pitch-Yaw</i> (QUORA (2015)).	60
3.22	Modelo físico simplificado do acelerômetro capacitivo (NXP (2008)).	61
3.23	Acelerômetro MMA7361 (ELETROGATE (2011)).	61
3.24	Esquemático do circuito com o conversor A/D MCP3008.	62
3.25	Circuito para a comunicação <i>I²C</i> entre o Raspberry Pi e o PIC.	63
3.26	Esquemático da placa do microcontrolador <i>PIC16F876A</i>	64
3.27	Circuito eletrônico para os sensores ultrassom.	65
3.28	PCB do circuito eletrônico para os sensores ultrassom.	65
3.29	Circuito eletrônico para acoplamento dos cabos dos motores.	66
3.30	PCB do circuito eletrônico para acoplamento dos cabos dos motores.	66
3.31	Função que pode ser vinculada a um botão para abrir um arquivo CGI.	69
4.1	Protótipo virtual do robô proposto	74
4.2	Análise do centro de massa do sistema robótico com placas de acrílico de alto impacto.	75
4.3	Análise do centro de massa do sistema robótico com placas de alumínio de alto impacto.	76
4.4	Placas de acrílico cortadas e fixadas no sistema robótico.	77
4.5	Sensor ultrassom montado em seu suporte fixado à base do robô.	79
4.6	Vista superior da base do robô evidenciando a configuração espacial dos sensores ultrassom.	80
4.7	Montagem final do mecanismo de acoplamento do motor à roda.	81
4.8	Fixação do motor e seu suporte à plataforma inferior do robô.	81

4.9	Montagem final da roda acoplada ao motor através de seus adaptadores.	82
4.10	Leitura dos <i>encoders</i> avaliadas durante teste de deslocamento.	83
4.11	Modelo de bateria de LiPo 3S 5000mAh 30C utilizada (FABMODELISMO (2017)).	84
4.12	Acelerômetro de 3 Eixos MMA8452Q (ROBOCORE (2016)).	85
4.13	Conversor de lógica bidirecional (ROBOCORE (2015)).	85
4.14	Placa Arduino Uno (ARDUINO (2015)).	86
4.15	Esquemático do circuito eletrônico simplificado.	87
4.16	PCB do circuito eletrônico simplificado.	87
4.17	88
4.18	Placa eletrônica para os sensores ultrassom	89
4.19	Placa eletrônica para acoplamento dos cabos dos motores.	89
4.20	Página web para <i>notebooks</i>	90
4.21	Página web para dispositivos móveis.	91
4.22	Montagem final do robô.	92
4.23	Exibição da tela inicial da plataforma de visitas virtuais evidenciando imóveis cadastrados.	93
4.24	Ferramenta “Desenhar” da plataforma de visitas virtuais.	93
4.25	<i>Pop-up</i> do Google Street View para visualizar a vizinhança.	94
4.26	Exemplo de fotografia padrão para imóveis sem suporte do Google Street View.	94
4.27	Interface de navegação dos ambientes internos mapeados.	95
B.1	Modelo do motor especificado (POLULU (2014a)).	106
B.2	Suporte para motor (POLULU (2014b)).	107
B.3	Adaptador para fixação do motor à roda - parte 1. (POLULU (2014c))	107
B.4	Adaptador para fixação do motor à roda - parte 2. (POLULU (2014c))	108

Lista de Tabelas

2.1	Principais características da comunicação I ² C.	35
3.1	Peso dos componentes	41
3.2	Parâmetros para avaliação das características dos motores em função da velocidade de deslocamento e condições ambientais.	42
3.3	Consumo energético dos principais componentes.	57

Lista de Acrônimos e Notação

3D	Três Dimensões
A/D	Analógico-Digital
ARM	Advanced RISC Machine (Máquina RISC Avançada)
AUV	Autonomous Underwater Vehicles (Veículo subaquático autônomo)
API	Application Programming Interface (Interface de Programação de Aplicações)
AWC	Academic Working Capital
B2B	Business-to-Business (negócios entre duas empresas)
BLE	Bluetooth Low Energy (bluetooth de baixo consumo de energia)
CA	Corrente Alternada
CC	Corrente Contínua
CGI	Common Gateway Interface (Interface Comum de Porta de Entrada)
CI	Circuito Integrado
CMOS	Complementary Metal-Oxide-Semiconductor (Semicondutor de Metal-Óxido Complementar)
CNC	Computer Numerical Control (Controle Numérico Computadorizado)
D/A	Digital-Analógico
DNS	Domain Name System (Sistema de Nomes de Domínios)
EEPROM	Electrically Erasable Programmable Read-Only Memory (Memória ROM Eletricamente Apagável e Programável)
EUA	Estados Unidos da América
ExoMars	Exobiology on Mars
fps	frames per second (quadros por segundo)
GND	Ground (Terra)
GPIO	General Purpose Input/Output (Entrada/Saída de Propósito Geral)
GPU	Graphics Processing Unit (Unidade de Processamento Gráfico)
HD	High-Definition (Alta Definição)
HDMI	High-Definition Multimedia Interface (Interface Multimídia de Alta Definição)
HDR	High Dynamic Range (Grande Alcance Dinâmico)
HTML	HyperText Markup Language (Linguagem de Marcação de Hipertexto)
HTTP	HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto)
I ² C	Inter-Integrated Circuit (Circuito Inter-Integrado)
IBGE	Instituto Brasileiro de Geografia e Estatística

IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IFR	International Federation of Robotics (Federação Internacional de Robótica)
IoT	Internet of Things (Internet das coisas)
ISO	Sensibilidade fotográfica
LAAS	Laboratoire d'Architecture et d'Analyse de Systèmes (Laboratório de Arquitetura e Análise de Sistemas)
LCD	Liquid Crystal Display (Display de Cristal Líquido)
LED	Light Emitting Diode (Diodo Emissor de Luz)
LiPo	Lithium Polymer (Polímero de Lítio)
MAX-C	Mars Astrobiology Explorer-Cacher (Explorador-Armazenador de Astrobiologia de Marte)
MOSFET	Metal Oxide Semiconductor Field Effect Transistor (Transistor de Efeito de Campo Metal-Óxido-Semicondutor)
NASA	National Aeronautics and Space Administration (Administração Nacional da Aeronáutica e Espaço)
NAVLAB	Navigation Laboratory (Laboratório de Navegação)
NFC	Near Field Communication (Comunicação por Campo de Proximidade)
PCB	Printed Circuit Board (Placa de Circuito Impresso)
PIC	Programmable Interface Controller (Controlador de Interface Programável)
PSEIKI	Production System Environment for Integrating Knowledge with Images (Ambiente do Sistema de Produção para Integração de Conhecimento com Imagens)
PWM	Pulse Width Modulation (Modulação por Largura de Pulsos)
RAM	Random Access Memory (Memória de Acesso Aleatório)
RIA	Robot Institute of America (Instituto americano de robótica)
ROM	Read-Only Memory (memória que permite apenas a leitura)
RPM	Rotações por minuto
RX	Receptor
SoC	System on a Chip (Sistema em um Chip)
SPI	Serial Peripheral Interface (Interface Periférica Serial)
TBJ	Transistor Bipolar de Junção
TTL	Transistor-Transistor Logic (Lógica Transistor-Transistor)
TX	Transmissor
UAV	Unmanned Aerial Vehicle (Veículo Aéreo Não Tripulado)
URL	Uniform Resource Locator (Localizador Uniforme de Recursos)
USB	Universal Serial Bus (Barramento Serial Universal)
VANTs	Veículos Aéreos Não Tripulados
Wi-fi	Wireless Fidelity (fidelidade sem fio)

A	Ampères
g	Gramas
GB	GigaByte
GHz	Gigahertz
Kg	Quilogramas
m	Metros
m/s	Metros por segundo
m/s ²	Metros por segundo ao quadrado
mA	Miliampères
MHz	Megahertz
mm	Milímetros
MP	MegaPixels
N.m	Newton por metro
V	Volts
π	Letra grega pi, equivalente à aproximadamente 3,1416
θ	Letra grega theta, tipicamente utilizada para indicar um ângulo

Introdução

A robótica móvel vem se desenvolvendo consideravelmente nas últimas décadas, possibilitando que diversos exemplos desses dispositivos estejam presentes no cotidiano de indivíduos comuns (SECCHI (2008)). Foi um grande passo desde os primeiros equipamentos do tipo, como a *Elsie* a tartaruga da década de 1940, e o *Shakey* dos anos 60, até os veículos aéreos não tripulados do inglês *Unmanned Aerial Vehicle* — *UAV*, aspiradores de pó autônomos e os *rovers*¹ em Marte (CORKE (2011)). Segundo SOUZA PIO; CASTRO; CASTRO JÚNIOR (2006), um robô móvel autônomo pode ser descrito como, um agente autônomo capaz de extrair informações do ambiente e utilizar esse conhecimento do ambiente para deslocar-se com segurança de modo significativo e intencional, atuando e executando tarefas. Apesar dos grandes avanços em inteligência artificial, nem todos os veículos robóticos são completamente autônomos (FAGGELLA (2016)).

Os robôs móveis são mais versáteis que robôs industriais convencionais, pois, apresentam mobilidade em seu ambiente de trabalho, o que possibilita a expansão da área de atuação e diversificação das atividades. O sistema de locomoção de um robô móvel pode ser implementado de modo a atender as condições ambientais, podendo ser implementado pelo uso de pernas, rodas, esteiras, entre outros (SOUZA PIO; CASTRO; CASTRO JÚNIOR (2006)).

Existe um campo da robótica, denominada telerrobótica, focada no controle de robôs semiautônomos à distância, através de métodos variados de comunicação, como *Wi-fi*, *bluetooth*, via satélite ou mesmo cabos (SCHMIDT; LANDIS; OLESON (2011)). Tais tecnologias proporcionam vantagens óbvias, das quais a grande maioria gira em torno do fato de possibilitar-se o acesso e intervenção em ambientes sem expor o operador humano aos perigos e custos inerentes a estar naquele local (SECCHI (2008)). Adicionalmente, expandiram a fronteira de alcance do homem, principalmente em aplicações submarinas

¹*Rovers* são veículos autônomos ou remotamente controlados, projetados para navegar por terrenos tipicamente acidentados (SCHENKER *et al.* (2003)).

e espaciais. Recentemente, o *drone*², um robô teleoperado têm-se popularizado devido a sua acessibilidade e versatilidade. Esses tem sido utilizados nas mais diversas atividades, como: filmagens e fotografias aéreas, inspeções de grandes áreas, entregas de suprimentos ou mesmo por entretenimento. A agropecuária tem-se destacado no uso dessa tecnologia, *drones* e aeromodelos estão sendo empregados na pulverização e monitoramento de rebanhos. A indústria extrativa de mineração, petrolífera e vegetal vem inovando ao usar veículos remotamente controlados para realizar inspeções em zonas de risco, em grandes distâncias e para análises topográficas (SCHMIDT; LANDIS; OLESON (2011)). Por sua vez, a agência americana National Aeronautics and Space Administration - NASA utiliza *rovers* para exploração espacial, como o robô *K-10*, e diversos *AUVs*³ são destinados à exploração marítima.

Tendo conhecimento do potencial e aplicabilidade de dispositivos telerrobóticos, o presente trabalho propõe o desenvolvimento de um robô para o mercado imobiliário e de turismo, o que em conjunto com outra tendência tecnológica - a realidade virtual - apresenta o potencial para melhorar e facilitar o modo através do qual se negociam imóveis e se escolhem destinos turísticos.

O mercado imobiliário é cada vez mais competitivo, fazendo-se necessário inovar para atrair clientes e vencer a ampla concorrência. Quando se fala em inovação no setor imobiliário, logo remete-se às novas técnicas na construção civil. Inovação significa novidade ou renovação, e no segmento imobiliário esse termo está diretamente ligado ao perfil do consumidor. O cliente deve ser sempre fonte de inovação no segmento. Nesse contexto, a exibição de propriedades, por exemplo, se dá por fotografias realizadas por funcionários não especializados, tornando-se um incômodo para os seus clientes, que devem lidar com imagens que não condizem com a realidade. Outra adversidade consiste no procedimento de visita à diversas edificações, o que está relacionado à outros problemas como a indisponibilidade ou incompatibilidade das chaves, deslocamento excessivo e burocracia envolvida. Esta é uma questão identificada por meio de prospecção direta, através de entrevistas com locatários que passaram pelo processo recentemente. Tal metodologia é conhecida como *Customer Discovery* (COWAN (2017)), desenvolvida pelo movimento de *Lean Startup*, muito útil para encontrar as principais angústias enfrentadas por possíveis usuários do produto em desenvolvimento. Nesse contexto, percebeu-se uma tendência dos proprietários destas imobiliárias para incorporar novas ferramentas que proporcionem uma melhor experiência aos compradores, visto que o mercado tem se tornado mais competitivo devido à grande oferta não acompanhada pela demanda. Atualmente, encontra-se disponível no mercado a realidade aumentada, em que o cliente consegue visualizar por

²*Drone*: um tipo de veículo aéreo não tripulado.

³*Autonomous Underwater Vehicles* - são, no nível mais fundamental, sistemas controlados por computador que operam de forma submersa (OCEANIC; ADMINISTRATION (2013)).

meio de imagens esféricas o prédio pronto, seja ele comercial ou residencial. Tem-se ainda o *tour* virtual, onde de casa ou do celular, o futuro comprador consegue olhar todo o imóvel por meio de um passeio virtual como se estivesse dentro do apartamento (MACHADO (2017)). Por isso, a implementação de um robô capaz de capturar diversas fotografias esféricas, de forma mais eficiente e padronizada que seres humanos, é fundamentada e justificada. Logo, o produto proposto apresenta duas propostas de valor como os principais alicerces: a aquisição de imagens e dados de forma remota e automática; e a capacidade de possibilitar ao usuário adentrar em um ambiente e inspecionar sem que seja necessária sua presença física. As duas são relevantes para segmentos de clientes diferentes, de modo que a segunda foca em usuários buscando a imersão proporcionada pela realidade virtual, levando-os à ambientes de interesse de forma fiel ao real. A primeira se aplica a indústrias onde a inspeção de ambientes insalubres, inacessíveis ou perigosos é necessária, como mineradoras, petrolíferas e construtoras de infraestrutura. Por isso, foi definido que o mercado imobiliário será o *Beachhead Market*⁴, o que significa que todo o sistema será projetado para atuar em ambientes comuns a este segmento. Apesar de existirem soluções similares, poucas atuam no mercado nacional e todas as aquisições de imagens esféricas são realizadas manualmente, levando ao principal diferencial competitivo do projeto: sua inovação tecnológica proporcionada pela automação, convergindo aos valores da mecatrônica. A eliminação da necessidade do operador de deslocar a câmera e se ausentar da imagem, reduz consideravelmente o tempo de captação, além de permitir a aquisição de imagens de forma quase contínua. Este último aspecto é o que possibilita a integração com a realidade virtual, proporcionando ao usuário a sensação de caminhar pelo local sem estar presente fisicamente. Esse será tema de futuros projetos para assim, finalizar o produto idealizado como parte do programa *Academic Working Capital*.

1.1 Definição do Problema

Devido ao processo burocrático e ineficiente de apresentação de imóveis por parte das imobiliárias, aliado à estagnação deste mercado, várias empresas do setor estão trabalhando para integração de novas tecnologias ao processo de vendas visando uma melhor competitividade. Já é perceptível a adoção de imagens esféricas nestes empreendimentos, tanto em maquetes eletrônicas, como também em mídias de propriedades reais. Nesse contexto, o protótipo a ser desenvolvido visa a automação do processo de aquisição de imagens do ambiente.

⁴*Beachhead Market*: é o mercado que, ao possuir sua maior parcela de consumidores, a empresa cria força para atuar nos mercados adjacentes com diferentes oportunidades; é o mercado de entrada para escalar uma empresa, no qual os consumidores se encaixam melhor com a proposta de valor (AULET (2013)).

Atualmente, o processo é realizado manualmente por um operador, que instala e posiciona o tripé da câmera no ponto desejado, como por exemplo, o centro do cômodo. Após finalizar o posicionamento da câmera, o operador deve se retirar do ambiente, para então realizar a aquisição da imagem. Esse procedimento é repetido inúmeras vezes durante o mapeamento do ambiente, sendo esse um dos motivos para o baixo número de empresas que desenvolvem o *tour* virtual com uma maior quantidade de imagens por metro quadrado. Normalmente, é exibido apenas uma mídia por cômodo. Complementarmente, para a elaboração de *tour* virtual de ambientes de maior escala, como *shopping centers*, museus e cruzeiros, os problemas processuais se agravam, devido às grandes distâncias envolvidas. Portanto, a utilização da plataforma robótica móvel para deslocar a câmera entre os pontos de aquisição das imagens, não só viabilizaria a aplicação da tecnologia de navegação em ambientes maiores, como também reduziria o tempo necessário para realizar a aquisição de mídias em imóveis, permitindo um maior número de imagens por unidade de área do local.

1.2 Motivação

A motivação que levou à escolha deste tema foi o potencial mercadológico observado em diversas pesquisas em campo e na participação no programa de pré aceleração de *startups Academic Working Capital - AWC*. Durante o processo de *Customer Discovery*⁵ e *Problem-Solution fit*⁶, identificou-se a oportunidade de desenvolvimento de uma plataforma robótica para a aquisição de imagens esféricas voltada para o setor imobiliário e de turismo. As justificativas e aplicabilidade do projeto são validadas pela parceria estabelecida com uma imobiliária de Divinópolis, possibilitando a identificação das características essenciais para a plataforma robótica.

1.3 Objetivos do Trabalho

1.3.1 Objetivo geral

Projetar e construir um robô móvel, controlado remotamente, de fácil operação, capaz de levantar dados que possibilite a construção de um mapa adequado para a realização de um *tour* virtual.

⁵*Customer Discovery*: uma metodologia para desenvolvimento de negócios, cujo objetivo primário é a identificação dos primeiros clientes. Para tanto, são estabelecidas premissas sobre quem são esses clientes, o problema exato sendo resolvido para eles e como irão pagar pela solução, as quais são transformadas em hipóteses e, posteriormente, em certezas, através de testes (principalmente, entrevistas com os consumidores potenciais) BLANK (2013).

⁶*Problem-Solution fit*: o alinhamento entre a solução desenvolvida e o problema dos consumidores a ser resolvido, ou seja, é o encaixe entre a proposta de valor e o segmento de clientes do modelo de negócios COWAN (2017).

1.3.2 Objetivos específicos

- Dimensionar a plataforma mecânica;
- Definir os sensores;
- Definir os atuadores;
- Definir a topologia do robô;
- Projetar e construir os circuitos eletrônicos;
- Dimensionar as células de energia;
- Desenvolver a interface de comunicação com o usuário;
- Definir o método de comunicação sem fio;
- Garantir a integridade do robô e ambiente;
- Desenvolver o sistema operacional do robô;
- Realizar a integração de todos os componentes mecânicos, eletrônicos e *software*;
- Testar e validar o protótipo;

1.4 Estado da arte

Atualmente, a robótica móvel vem se popularizando cada vez mais, com a redução do preço tanto dos componentes como do produto final. Certamente, um dos exemplos mais populares de robôs móveis são os *drones*. Estes iniciaram-se no passado com fins militares, porém essa tecnologia foi disponibilizada para o mercado consumidor comum. Ainda que os *drones* sejam utilizados para entretenimento por algumas pessoas, eles também são empregados em atividades e aplicações profissionais, como: fotografia e filmagem aérea de eventos, análises topográficas de áreas extensas e pulverização de agrotóxicos em plantações (DOCTORDRONE (2016)). Recentemente, esses dispositivos estão sendo estudados e testados pela Amazon no processo de entregas rápidas, o que pode gerar uma inovação disruptiva que mudará a forma como a logística é pensada atualmente (AMAZON (2016)). Entretanto, um problema que foi encontrado é a autonomia da bateria dos *drones*, mas soluções veem sendo estudadas para saná-lo, o que torna esta uma aplicação com grande potencial para o futuro. Esta solução está sendo desenvolvida pela divisão Amazon Prime Air e um de seus drones é apresentado na Figura 1.1.



Figura 1.1: Amazon Prime Air (AMAZON (2016)).

Aplicações inovadoras relacionando à robótica móvel e biônica, quando os robôs são baseados e simulam o comportamento de outros animais, são desenvolvidas por diferentes empresas e centros de pesquisa. A Boston Dynamics, por exemplo, tem desenvolvido diversos protótipos nesse campo, como o Cheetah, BigDog, Atlas, SandFlea, entre outros (BOSTONDYNAMICS (2017)). Estes robôs têm diversas aplicações, alguns para o setor militar e outros para a área acadêmica. Alguns dos robôs da Boston Dynamics podem ser observados na Figura 1.2. Já a empresa de automação Festo desenvolveu o eMotionButterflies, BionicKangaroo e BionicANTs, também no campo da Biônica aplicada à robótica móvel. Estas apresentam diferentes aplicações e inovações, como por exemplo as eMotionButterflies que simulam o voo e as asas de borboletas, além de realizarem voos com comportamento coordenado. Essas têm sido aplicadas para criar sistemas de orientação e monitoramento, que podem ser de grande valia para as fábricas do futuro (FESTO (2015)). As eMotionButterflies são apresentadas pela Figura 1.3.



Figura 1.2: Alguns dos robôs da Boston Dynamics. Da esquerda para direita: ATLAS original, próxima geração do ATLAS, BigDog, WildCat e AlphaDog (IEEE SPECTRUM (2016)).



Figura 1.3: As eMotionButterflies da Festo (FESTO (2015)).

Há muito tempo que a NASA e outras agências espaciais têm investido em veículos motorizados automáticos, os chamados *rovers*, para a exploração espacial. No presente, o *rover* mais atual em funcionamento é o Curiosity, da NASA, que aterrisou em Marte em 6 de agosto de 2012. Os próximos rovers em desenvolvimento são para 2018: o

MAX-C (Mars Astrobiology Explorer-Cacher) da NASA e o ExoMars da Agência Espacial Europeia.

Além disso, mais recentemente, um dos exemplos mais em alta da robótica móvel são os carros autônomos. Esses veículos estão sendo desenvolvidos por diversas empresas de diferentes setores da tecnologia: Uber, Tesla, Google, Ford, Volvo, Nissan, BMW e Mercedes estão todas desenvolvendo automóveis com essa tecnologia. Há exemplos de testes bem sucedidos, com alguns desses carros já transitando normalmente em grandes cidades (TECMUNDO (2016)) (G1 (2017a)), por isso, o estado de Nova York vai começar a autorizar testes com veículos autônomos em suas ruas, se juntando a outros 13 estados dos EUA (G1 (2017b)). Entretanto, ainda são necessários alguns ajustes e testes para se garantir a segurança e ausência de falhas na operação dos mesmos. O carro autônomo da Tesla em funcionamento é apresentado na Figura 1.4



Figura 1.4: O carro autônomo da Tesla (MEIO BIT (2016)).

1.5 Revisão de Literatura

O robô industrial é definido pelo *Robot Institute of America* — RIA como um manipulador multifuncional, reprogramável, projetado para mover materiais, partes, ferramentas ou dispositivos especializados através de movimentos programados variados para a realização de tarefas (SPONG; HUTCHINSON; VIDYASAGAR (2006)).

Nos primórdios, manuseavam-se materiais ou realizavam-se simples trabalhos de aplicações industriais através dos teleoperadores. O primeiro equipamento desta classe eletricamente foi desenvolvido em 1947 (GU (2013)). Previamente, durante a Segunda Guerra Mundial, os teleoperadores, ou dispositivos mestre-escravo, eram utilizados para manipular materiais radioativos. E o robô, como foi definido, surgiu da união destes aparelhos com as fresadoras controladas por comando numérico computacional — CNC, o qual foi criado devido à alta precisão necessária para a fabricação de componentes de alta performance das aeronaves. Assim, os primeiros robôs combinavam os arranjos mecânicos dos teleoperadores à autonomia e programabilidade das máquinas CNC (SPONG; HUTCHINSON; VIDYASAGAR (2006)).

Ainda assim, apesar de todo o sucesso dos manipuladores robóticos, superando 1,6 milhões de unidades no final de 2015, de acordo com a *International Federation Of Robotics* — IFR em seu relatório executivo de 2016 (ROBOTICS (2016)), estes apresentam uma desvantagem fundamental: a falta de mobilidade. Como evidência, tem-se o fato de que um manipulador fixo possui uma variedade limitada de movimentos, sendo dependente do local em que foi instalado. Por sua vez, os robôs móveis são dispositivos equipados com determinado nível de automação, dotados de um sistema de locomoção capaz de navegar através de um determinado ambiente de trabalho (CORKE (2011)). Em contraste aos robôs industriais, um robô móvel seria capaz de deslocar-se por uma fábrica, flexivelmente aplicando seus talentos onde fosse mais efetivo (SIEGWART; NOURBAKHS; SCARAMUZZA (2011)).

Dessa forma, paralelamente ao surgimento dos robôs industriais, iniciaram-se os estudos em robótica móvel e em 1949, Dr. W. Grey Walter construiu Elmer e Elsie, esse último apresentado na Figura 1.5. Eles foram dois robôs autônomos inspirados em tartarugas, formalmente denominados *Machina Speculatrix* (HOLLAND (2003)). Esses dispositivos reagem ao seu ambiente e podiam seguir uma fonte de luz externa sem possuir nenhum plano explícito ou conhecimento de seu posicionamento (CORKE (2011)).

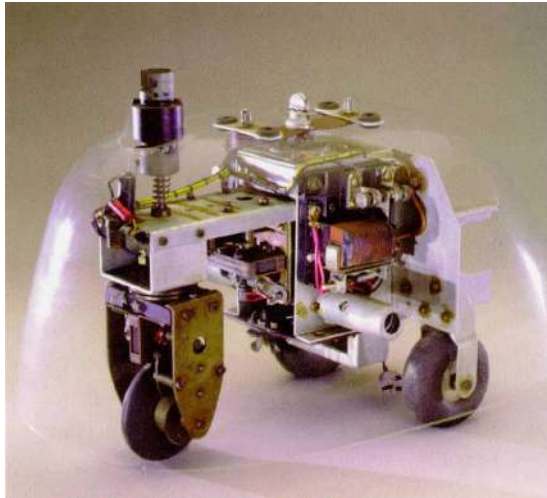


Figura 1.5: Robô Elsie (HOLLAND (2003))

Em 1968, no Instituto de Pesquisa de Stanford, foi desenvolvido o Shakey, o qual era capaz de obter uma percepção 3D e criação de um mapa de seu ambiente, utilizado para planejar uma trajetória até o seu destino (CORKE (2011)). Esse robô tinha agregado em sua estrutura diversos sensores, como uma câmera de vídeo e sensores de toque, sendo capaz de navegar entre as salas do laboratório, enviar sinais de rádio ao seu cérebro, além de um computador DEC PDP-10, que lhe permitia executar algumas tarefas como deslocar caixas e evitar obstáculos (NITZAN (1985)).

Em 1977, a equipe do *Stanford Artificial Intelligence Laboratory*, criou o StanfordCart (BOTELHO (1996)), que operava em um ambiente plano, evitando obstáculos e planejando sua rota depois de parar e ler seus sensores a cada metro percorrido (BRUMITT; COULTER; STENTZ (1993)).

Já em 1983, desenvolveu-se o Hilare no LAAS (*Laboratoire d'Architecture et d'Analyse de Systèmes*), um robô multissensorial de 400Kg, com 4 processadores Intel 80286 e comunicação via rádio serial (BOTELHO (1996)).

A década de 1980 contou com muitos trabalhos no campo de robótica móvel em vários países, destacando-se os sistemas NAVLAB e PSEIKI do *Robot Vision Laboratory* da Universidade de Purdue, focados no reconhecimento de imagens através da visão computacional (DESOUZA; KAK (2002)). Outra importante pesquisa foi o desenvolvimento do robô Khepera, do K-Team, no *Microcomputing Laboratory do Swiss Federal Institute of Technology*, com o apoio de várias instituições europeias como *University of Zürich*, *University of Karlsruhe*, *University of Sussex*, entre outros. Além de suas dimensões reduzidas, de apenas 55 mm de diâmetro e 30 mm de altura, o Khepera tinha a capacidade de desviar de obstáculos e seguir ou evitar fontes luminosas. Também é possível a utilização de algumas extensões, como sistemas de visão e um pequeno manipulador (MONDADA;

FRANZI; GUIGNARD (1999)). Esse robô obteve uma boa aceitação, se tornando objeto de pesquisa inclusive no Brasil (BOTELHO (1996)).

Já no campo de exploração espacial, a NASA utilizou *rovers*, ou veículos terrestres robotizados, para exploração de Marte, sendo a primeira sonda, o Viking 1, enviada em agosto de 1975. Em conjunto ao Viking 2, lançado em setembro do mesmo ano, foram os primeiros a descender à superfície do planeta. Posteriormente, partindo em 4 de dezembro de 1996, a missão Mars Pathfinder, consistindo no *lander* e no famoso robô Sojourner, retornou uma quantidade sem precedentes de dados ao explorar as antigas planícies inundadas do hemisfério norte, conhecidas como *Ares Vallis*. Em janeiro e julho de 2003, foram lançados dois *rovers* sofisticados para explorar a superfície de Marte, o Spirit e o Opportunity com mobilidade superior ao Pathfinder Sojourner. Esses robôs encontram-se equipados com um conjunto sofisticado de instrumentos em busca de evidência de água líquida existente no planeta. Apesar desses robôs apresentarem a mesma estrutura, eles exploraram regiões distintas. O último robô móvel terrestre enviado foi o Curiosity em novembro de 2011, exibido na Figura 1.6. Continuando a partir do grande sucesso obtido com os dois veículos anteriores, tem como missão investigar se Marte alguma vez teve um ambiente capaz de suportar pequenas formas de vida, como micróbios. Esse apresenta duas vezes o comprimento e três vezes o peso do Spirit e Opportunity, ele coleta amostras do solo e rochas para analisá-los em busca de compostos orgânicos e condições ambientais que possam ter suportado vida microbiana atualmente ou no passado (NASA (2015)).

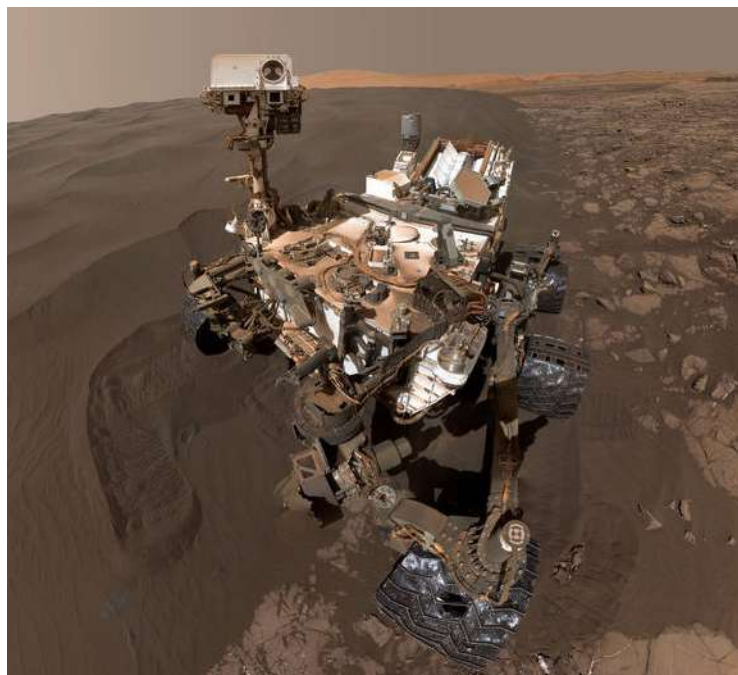


Figura 1.6: Robô Curiosity desenvolvido pela NASA (NASA (2015)).

Paralelamente a esses desenvolvimentos, várias empresas criaram dispositivos e protótipos de pesquisa no campo da robótica móvel. Robôs domésticos usados para limpar a casa como o Roomba e Scooba da iRobot (2009), para cortar grama como o AutoMower da Huskvarna (2009) e o robô Aibo da Sony (2009) com pernas capazes de caminhar como cachorro. Também em 2009, o humanoide Asimo da Honda Figura 1.7 foi apresentado ao público, simultaneamente ao BigDog da Boston Dymanics (2009), completando uma breve lista de destaques produzidos por empresas. Adicionalmente, podem ser citados o Stanley de Stanford (THRUN *et al.* (2006)), que competiu e venceu o Darpa Challenge em 2005, e os Veículos aéreos não tripulados — VANTs brasileiros do Projeto Arara (NERIS (2001)) e AGplane (AGX, 2009).



Figura 1.7: Robô Asimo desenvolvido pela Honda (RIDERS (2014))

1.6 Escopo do trabalho

Este documento se encontra dividido em 5 capítulos. O presente introduz e contextualiza o trabalho, apresenta a definição do problema, a motivação para sua realização, o estado da arte, a revisão de literatura e a organização do texto.

No segundo capítulo é realizada uma revisão dos conceitos e técnicas necessários para o desenvolvimento do projeto. Já no terceiro capítulo é apresentado a metodologia adotada durante o desenvolvimento do projeto do protótipo, como a definição dos materiais utilizados, o projeto mecânico e eletrônico do *hardware* da plataforma proposta. No quarto capítulo são apresentados os resultados alcançados nesse trabalho levando à conclusão e considerações finais abordadas no capítulo 5.

Fundamentação Teórica

Neste capítulo são abordados os principais conceitos teóricos necessários para o projeto e construção da plataforma robótica móvel. Uma breve revisão sobre sistemas de locomoção, topologias e modelos de rodas utilizados na robótica móvel é apresentada. Além disso, esse capítulo aborda sistemas e componentes que constituem o sistema robótico proposto. Por fim, são apresentados tópicos relacionados à programação da página Web para controle do dispositivo.

2.1 Locomoção de robôs móveis

Robôs móveis devem apresentar, em sua estrutura, mecanismos e sistemas de locomoção que possibilitem seu deslocamento pelo ambiente. Encontra-se na literatura diversas topologias para construção dos sistemas de movimentação de robôs. A definição da topologia empregada em um dispositivo robótico é uma importante etapa durante o projeto de sua estrutura mecânica, visto que essa escolha definirá o ambiente pelo qual o dispositivo poderá se deslocar SIEGWART; NOURBAKHS; SCARAMUZZA (2011). Ainda segundo esses autores, a maioria destes mecanismos de locomoção são inspirados em seus homólogos biológicos. Existe, entretanto, uma exceção: a roda acionada ativamente, uma invenção humana que obtém uma alta eficiência em superfícies planas, sendo também empregadas em terrenos irregulares. Mesmo com essa óbvia vantagem, a natureza não desenvolveu uma junta ativamente acionada e totalmente rotacional, o que é justamente a tecnologia necessária para uma locomoção baseada em rodas.

Sistemas biológicos sucedem ao movimentarem através de uma variedade de ambientes hostis, sendo, portanto, interessante copiar seus mecanismos de locomoção. Porém, replicá-los é extremamente difícil por múltiplas razões, conforme pode ser verificado na Figura 2.1. Aponta-se, por exemplo, o fato de que a complexidade mecânica nestes sistemas naturais é obtida através da replicação estrutural proporcionada pela divisão celular combinada à especialização, sendo portanto, capaz de produzir uma milípede com várias

centenas de pernas. Em estruturas projetadas pelo homem, cada parte deve ser fabricada individualmente, não contando com a existência destas economias de escala. Finalmente, os sistemas de armazenamento energético e muscular usado por grandes animais e insetos atingem um torque, tempo de resposta e eficiências de conversão que excedem drasticamente os sistemas similares construídos pelo homem (DUDEK; JENKIN (2010)).

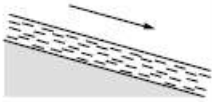
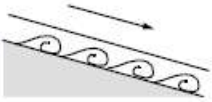

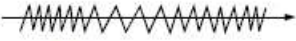



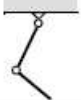



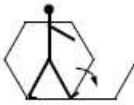
Tipo de movimento	Resistência ao movimento	Cinemática básica do movimento
Fluxo em um canal 	Forças Hidrodinâmicas	Redemoinhos 
Rastejar 	Forças de fricção	Vibração longitudinal 
Escorregar 	Forças de fricção	Vibração transversal 
Correr 	Perdas de energia cinética	Movimento oscilatório de um pêndulo com mais de um link 
Pular 	Perdas de energia cinética	Movimento oscilatório de um pêndulo com mais de um link 
Andar 	Forças gravitacionais	Rotação de um polígono 

Figura 2.1: Mecanismos de locomoção usados em sistemas biológicos (SIEGWART; NOURBAKHS; SCARAMUZZA (2011)).

Devido a dificuldade de reproduzir sistemas mecânicos e energéticos tão eficientes quanto os biológicos, são aplicados em uma vasta gama de plataformas robóticas, sistemas de locomoção baseados em rodas (SIEGWART; NOURBAKHS; SCARAMUZZA (2011)). Esses dispositivos possibilitam que o robô se desloque por diversos ambientes, exigindo também, menos potência energética que outros arranjos, como, pernas articuladas (KELLY (2013)). Ademais, pode ocorrer da fusão de sistemas locomoção como a combinação de articulações e rodas, formando um sistema mais dinâmico e versátil, como é o caso do robô Athlete da NASA, apresentado na Figura 2.2. O conceito deste robô é baseado em seis membros, cada um com seis graus de liberdade, aos quais são acopladas rodas com um grau de liberdade. O Athlete utiliza estas rodas para deslocar-se por terre-

nos estáveis, mas cada membro pode ser utilizado como uma perna. Neste caso, as rodas podem ser travadas e usadas como pés para andar sobre terrenos excessivamente macios, acidentados ou cheios de obstáculos. Ele foi projetado para ser um veículo utilitário com o propósito de auxiliar a exploração humana na superfície lunar, principalmente no que se refere ao transporte de cargas pesadas (NASA (2017)).



Figura 2.2: Robô Athlete da NASA (NASA (2017)).

Conforme pode ser verificado no gráfico da Figura 2.3, nestes terrenos, a locomoção por rodas proporciona eficiências de uma a duas ordens de grandeza maiores que por pernas. A linha férrea é um bom exemplo de um projeto em que se emprega o sistema de locomoção por rodas. Nele não há obstáculos que devem ser superados pelas rodas, além da fricção de rolamento minimizada entre as superfícies. No entanto deve-se observar que quando as rodas ou o ambiente tornam-se macios, a locomoção por rodas acumula ineficiências devido ao aumento de contato com a superfície, o que conseqüentemente causa o aumento do atrito, resultando em um consumo energético maior. Por sua vez, sob as mesmas condições, sistemas de locomoção por pernas são menos afetados, visto que a superfície de contato com o ambiente é reduzida (DUDEK; JENKIN (2010)). De fato, a eficiência das rodas depende grandemente das características do ambiente, principalmente em relação à sua dureza e planicidade. Já o sistema por pernas depende basicamente da massa da perna e do corpo, ambos os quais o robô deve suportar em vários pontos destes tipos de configuração (SIEGWART; NOURBAKSH; SCARAMUZZA (2011)).

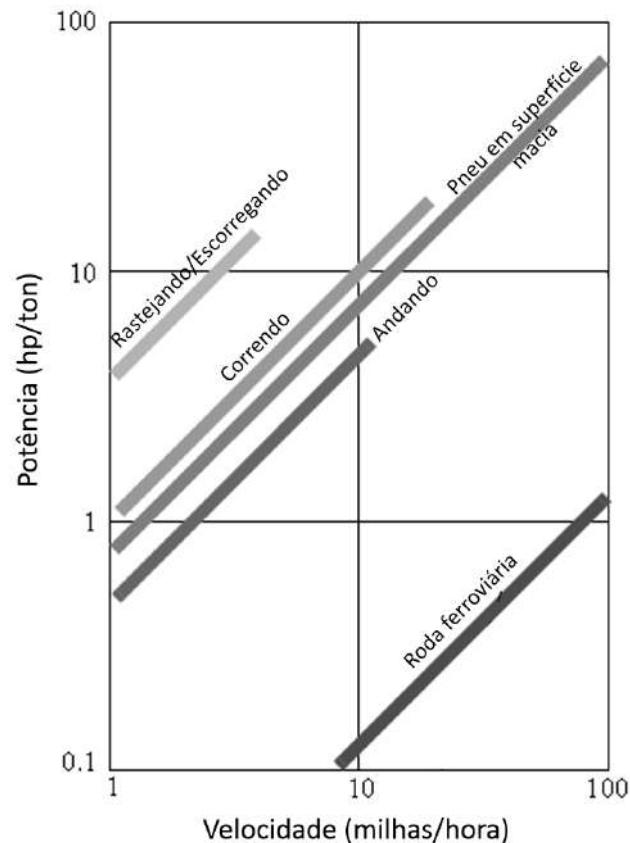


Figura 2.3: Energia *versus* velocidade obtida para vários mecanismos de locomoção. Adaptado. (SIEGWART; NOURBAKHS; SCARAMUZZA (2011)).

2.1.1 Principais aspectos da locomoção

De acordo com SIEGWART; NOURBAKHS; SCARAMUZZA (2011), a locomoção é um complemento da manipulação. Na última, o braço robótico é fixo, mas movimentado objetos no espaço de trabalho ao exercer forças sobre estes elementos. Já no deslocamento, o ambiente é fixo e o robô se move ao exercer força neste meio. Em ambos os casos, estudam-se os atuadores que geram as forças de interação e mecanismos que implementam as propriedades cinemáticas e dinâmicas desejadas. Eles compartilham os mesmos problemas de estabilidade, características de contato e tipo de ambiente:

- Estabilidade:
 - Número e geometria dos pontos de contato;
 - Centro de gravidade;
 - Estabilidade estática e dinâmica;
 - Inclinação do terreno;

- Características de contato:
 - Ponto de contato/tamanho e formato;
 - Ângulo de contato;
 - Atrito;
- Tipo de ambiente:
 - Estrutura (ambiente mapeado ou desconhecido);
 - Meio (água, ar, superfície dura ou macia);

2.2 Morfologia de robôs móveis com rodas

2.2.1 Rodas

A roda é, de fato, o mecanismo de locomoção mais popular no projeto de robôs móveis e de veículos projetados e construídos pelo homem, alcançando desempenho satisfatório. Na Figura 2.3, são apresentadas algumas topologias e modelos de rodas utilizados por robôs com esse tipo de sistema de locomoção (SIEGWART; NOURBAKSH; SCARAMUZZA (2011)).

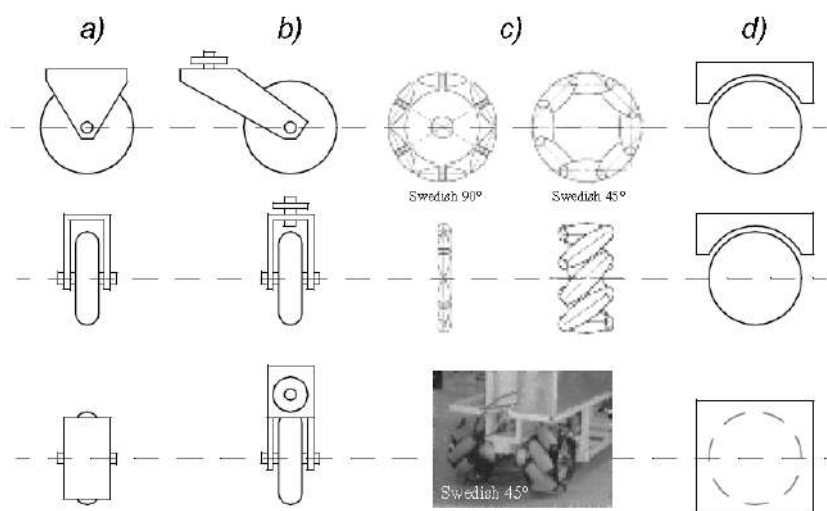


Figura 2.4: Os quatro tipos básicos de rodas. (a) Roda padrão: dois graus de liberdade; rotação em torno do eixo (motorizado) da roda e o ponto de contato. (b) Rodízio: dois graus de liberdade; rotação em torno de uma junta direcional deslocada. (c) Roda sueca: três graus de liberdade; rotação em torno do eixo (motorizado) da roda e o ponto de contato. (d) Roda esférica: difícil implementação técnica. (SIEGWART; NOURBAKSH; SCARAMUZZA (2011)).

Existem quatro classes de rodas principais, conforme exibido na Figura 2.4. Elas diferem significativamente em relação às suas cinemáticas e graus de liberdade, consequentemente, a escolha do tipo da roda tem um grande efeito na cinemática do robô móvel. A roda padrão e o rodízio são altamente direcionais, possuindo um eixo primário de rotação. Desse modo, para mover-se em uma direção diferente, a roda deve ser rotacionada ao longo de um eixo vertical. A principal diferença entre os dois tipos de rodas é que o modelo padrão pode completar o movimento de mudança de direção sem efeitos colaterais, uma vez que o centro de rotação passa pelo ponto de contato com a superfície. Os rodízios, entretanto, rotacionam-se em torno de um eixo deslocado, gerando uma força a ser sustentada pelo chassi do robô durante a mudança de direção (SIEGWART; NOURBAKHS; SCARAMUZZA (2011)).

A roda sueca funciona como a convencional, mas proporciona menos resistência em outras direções, como perpendicularmente à direção padrão no caso da *Swedish 90*, ou em um ângulo intermediário como na *Swedish 45*. Os pequenos roletes acoplados em torno de sua circunferência são passivos, sendo o eixo primário da roda o único elemento ativo. A principal vantagem deste modelo é a capacidade de movimento com pouco atrito ao longo de diversas trajetórias, não apenas em linhas retas (KACHROO; MELLODGE (2004)).

A roda esférica é realmente omnidirecional, normalmente projetada para ser atuada e girar em qualquer direção. Um mecanismo de implementação deste tipo consiste em roletes posicionados acima desta esfera, e ao serem acionados, geram uma força rotacional para iniciar o movimento (KELLY (2013)).

2.2.2 Topologia de robôs móveis

Segundo KACHROO; MELLODGE (2004), o balanceamento, normalmente, não é um tópico problemático em projetos de robôs com rodas, sendo três rodas suficientes para garantir a estabilidade de balanceamento. Porém, quando mais de três rodas são utilizadas, um sistema de suspensão é necessário para manter contato com o chão quando o robô se encontra em terrenos com desníveis. Um dos procedimentos mais simples para o projeto de suspensões consiste apenas na inclusão de flexibilidade nas próprias rodas, como por exemplo, a utilização de pneus. Esses funcionam como uma suspensão primitiva, e, obviamente, não se comparam com sistemas mais sofisticados aplicados em robôs que se deslocam em terrenos muito acidentados (SIEGWART; NOURBAKHS; SCARAMUZZA (2011)). Desse modo, neste tipo de pesquisa, é uma tendência focar-se em problemas de tração e estabilidade, manobrabilidade e controle.

2.2. Morfologia de robôs móveis com rodas

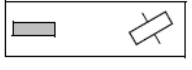
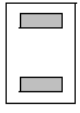
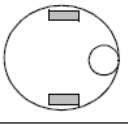
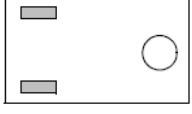
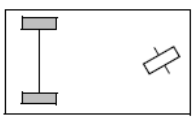
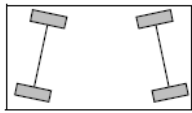
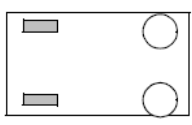
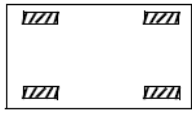
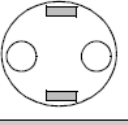







Número de rodas	Configuração	Descrição	Exemplos típicos
2		Uma roda direcional na frente, e uma roda de tração atrás.	Bicicleta, motocicleta;
		Tração diferencial nas duas rodas, com o centro de massa abaixo do eixo.	Robô pessoal Cye;
3		Tração diferencial em duas rodas centrais com um terceiro ponto de contato.	Nomad Scout, smartRob EPFL;
		Duas rodas tracionadas de forma independente atrás ou na frente; e uma roda omnidirecional não acionada no lado oposto.	Muitos robôs de ambientes internos, incluindo os robôs Pygmalion e Alice.
		Duas rodas tracionadas de forma diferencial conectadas na traseira; e roda livre direcional na frente.	Mini caminhões da Piaggio
		Quatro rodas direcionais motorizadas	Tração 4x4, Hyperion (CMU)
		Duas rodas tracionadas de forma diferencial em um lado, e duas rodas omnidirecionais no lado oposto	Charlie (DMT-EPFL)
		Quatro rodas omnidirecionais.	Carnegie Mellon Uranus
		Duas rodas de tração diferencial com dois pontos de contato adicionais.	EPFL Khepera, Hyperbot Chip
Os ícones referentes a cada tipo de roda são como segue:			
	Roda omnidirecional não acionada (esférica, castor, sueca);		
	Roda sueca motorizada;		
	Roda comum não acionada;		
	Roda comum motorizada;		
	Roda castor direcional motorizada;		
	Roda comum direcional;		
	Rodas conectadas.		

Figura 2.5: Configurações das rodas em robôs móveis; Adaptado. (SIEGWART; NOUR-BAKSH; SCARAMUZZA (2011)).

A escolha dos tipos de rodas está diretamente relacionada à suas disposições e geometrias, sendo dever do projetista a consideração das três características fundamentais: manobrabilidade, controlabilidade e estabilidade. Conforme pode ser verificado na Figura 2.5, algumas das disposições não são comumente utilizadas na robótica móvel, mas ainda assim servem para indicar a grande variedade de configurações possíveis (SIEGWART; NOURBAKHS; SCARAMUZZA (2011)).

Na comunidade científica, algumas classes de robôs móveis são populares por desenvolverem alta manobrabilidade (KLANCAR; BLAZIC (2017)). Nestas classes, um certo deslocamento pode exigir um movimento rotacional inicial, mas com um chassi circular e um eixo de rotação no centro do robô, ele pode girar sem deslocar-se linearmente pelo espaço. O robô mais popular deste tipo é o de duas rodas acionadas de forma diferencial, no qual as duas giram em torno do ponto central da plataforma. Um ou dois pontos adicionais de contato podem ser usados para incrementar a estabilidade de acordo com a aplicação para qual está sendo projetado.

2.3 Atuadores

Um motor de corrente contínua — CC é um motor elétrico cuja alimentação é proveniente de uma bateria ou qualquer outra fonte de alimentação CC. Ele realiza a conversão de energia entre rotor e estator através de escovas (escovado) ou sem escovas (*brushless*). Sua velocidade pode ser controlada variando a tensão em seus terminais, diferentemente de um motor elétrico de corrente alternada — CA cuja a velocidade é variada pela frequência. Essa característica justifica a sua utilização em aplicações que focam a velocidade (NIKU (2001)).

Os motores de passo são motores elétricos que possuem polos fixos, não contendo escovas e comutadores. Se destacam pela alta precisão e força obtidas. O número de passos que o motor gera, é exatamente igual ao número de pulsos recebidos e a velocidade do motor é igual a frequência de entrada desses pulsos. Ao mandar um pulso em uma bobina, o eixo se moverá em direção a esta bobina e assim por diante (NIKU (2001)).

Ainda segundo NIKU (2001) os servomotores são parecidos com motores de passo, porém contam com um sistema de controle interno o qual verifica a posição de entrada com a posição de saída e um motor de corrente contínua.

O dimensionamento dos motores CC para um robô móvel poder ser realizado, conforme procedimento proposto por GUIMARÃES (2006), sendo esse:

1. Definir o peso máximo para o robô.;
2. Estimar ou determinar os coeficientes de fricção das rodas com o ambiente;

3. Definir: velocidade máxima, aceleração e o raio da roda.;
4. Calcular a velocidade angular do motor;
5. A partir do peso estimado e da aceleração requerida, determinar o torque mínimo do motor;
6. Escolher um motor cujas especificações excedam o torque e rotação calculados.

Segundo GUIMARÃES (2006), a velocidade linear é descrita em função do raio da roda e velocidade angular para determinado torque. Além disso, essa pode sofrer alguma interferência, caso exista uma caixa redutora entre o eixo do motor e a roda. A distância percorrida por uma roda ao completar uma rotação, ou seja, um giro completo de 360° , é dada por (HIBBELER (2005)):

$$d = 2\pi \cdot r \quad (2.1)$$

em que: d é a distância percorrida e r é o raio da roda. Com a distância por rotação definida, é possível calcular a velocidade linear, v , calculada pela variação da posição em um certo intervalo de tempo, conforme (WALKER *et al.* (2002)):

$$v = n \frac{\partial d}{\partial t} = n \frac{2\pi r}{\Delta t} \quad (2.2)$$

em que: n é o número de rotações e Δt o intervalo de tempo.

A velocidade angular de um corpo rígido em torno do eixo fixo é definida e expressa em termos do vetor posição e velocidade linear deste corpo (BEER; JOHNSTON JR; CORNWELL (2013)), tal que:

$$\omega = \frac{v}{2\pi \cdot r} \quad (2.3)$$

em que ω é a velocidade angular, v a velocidade linear e r ao raio da roda. Assim, a velocidade angular expressa em rotações por minuto — rpm, é dada por

$$\omega = \frac{60v}{2\pi \cdot r} = \frac{30v}{\pi \cdot r}. \quad (2.4)$$

Por definição, o torque é dado pelo produto entre a força aplicada à uma alavanca pela distância entre o ponto de aplicação da força e o eixo de rotação da alavanca (WALKER *et al.* (2002)). De acordo com a definição, para definir o torque do motor é necessário conhecer quais forças estão envolvidas na movimentação do robô. Para tal, assume-se que o robô irá percorrer ambientes com pisos planos, ausentes de obstáculos para as rodas. Uma vez que essas hipóteses são consideradas, apenas duas forças passam a atuar sobre

o movimento, sendo essas: a resultante da inércia quando em aceleração, e aquela devido à força de fricção. Essas forças podem ser expressas por:

$$F = ma + mg\mu \quad (2.5)$$

em que: m é a massa sustentada por cada roda, a a aceleração, g a aceleração gravitacional, e μ o coeficiente de atrito entre as rodas e a superfície.

Uma vez que se conhece a velocidade inicial do deslocamento (normalmente assume-se que o movimento inicia-se do repouso), a aceleração pode ser definida por meio da equação de Torricelli, tal que:

$$a = \frac{v^2 - v_0^2}{2\Delta d} \quad (2.6)$$

em que: v é a velocidade máxima desejada, v_0 é a velocidade inicial e Δd é a distância percorrida durante o movimento.

Retomando a definição do torque, esse pode ser expresso por:

$$T = F \cdot L \quad (2.7)$$

em que F é a força aplicada na alavanca de comprimento L , nesse caso, equivalente ao raio da roda do robô.

Ao substituir (2.5) e (2.6) em (2.7), obtém-se a equação para cálculo do torque em função dos parâmetros conhecidos do robô, conforme:

$$\begin{aligned} T &= (ma + mg\mu)L \\ T &= \left[m \left(\frac{v^2 - v_0^2}{2\Delta d} \right) + m \cdot g \cdot \mu \right] L \\ T &= mL \left(\frac{v^2 - v_0^2}{2\Delta d} + g \cdot \mu \right) \end{aligned} \quad (2.8)$$

considerando que para o robô, $L = r$, então tem-se:

$$T = mr \left(\frac{v^2 - v_0^2}{2\Delta d} + g \cdot \mu \right) \quad (2.9)$$

2.4 Mecanismo *Pan-Tilt*

Conforme descrito por KIKUCHI (2007), uma câmera *pan-tilt* é composta por um dispositivo motorizado acoplado à câmera, o qual possibilita sua rotação nos eixos *pan* e *tilt* 2.6. Com esses graus de liberdade, é possível visualizar qualquer ponto ao seu redor, entretanto, devido às restrições construtivas presentes em aplicações práticas, o movimento é limitado em determinadas orientações. Para implementar esse mecanismo, são utilizados servo motores por permitirem o controle de posição de cada eixo de rotação, garantindo a pose do sistema e portanto, o campo de visão da câmera.

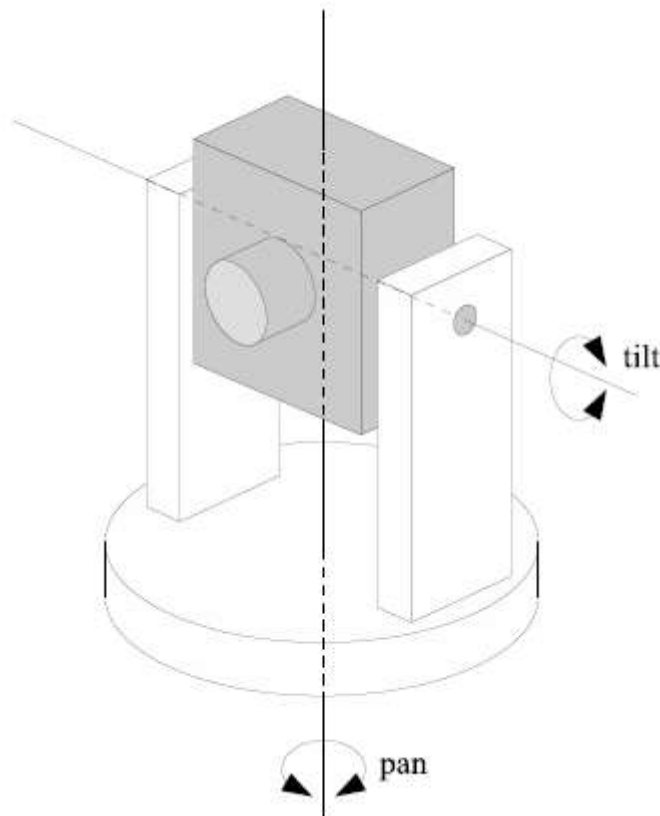


Figura 2.6: Esquema de uma câmera *pan-tilt* e seus eixos de rotação. (KIKUCHI (2007))

2.5 Microcontroladores e Microprocessadores

Segundo CRISP (2004), Microprocessadores são circuitos integrados que, como qualquer sistema, têm entradas e saídas, além de um processo que estabelece uma relação entre elas. Essas entradas e saídas são uma série de tensões que podem ser utilizadas para controlar dispositivos externos. Durante seu processo, que é estipulado segundo programação, as entradas são analisadas e geram saídas referentes a elas. Estes dispositivos têm como principais funções: o processamento de informações e a tomada de decisões segundo instruções previamente determinadas.

Os microprocessadores começaram a ser produzidos no ano de 1971 por duas companhias americanas: a *Intel* e a *Texas Instruments* (CRISP (2004)). Estes dispositivos por si só não conseguem realizar muitas tarefas, sendo necessária a sua integração com outros dispositivos, como memórias e periféricos. Posteriormente, os microprocessadores, associados a esses outros dispositivos, foram montados em apenas um circuito integrado e, dessa integração, nasceram os chamados microcontroladores (NICOLOSI (2006)).

Microcontroladores são sistemas completos de controle baseados em microprocessado-

res (CRISP (2004)). Estes apresentam em um só circuito integrado: um microprocessador, memória RAM, memória ROM, conversores A/D e D/A, temporizadores, contadores, GPIO (General Purpose Input/Output), etc. O seu propósito é executar tarefas específicas gravadas em sua memória de código e em geral se comunica com o mundo real para receber informações deste e assim atuar no mesmo de maneira desejável (NICOLOSI (2006)).

Segundo SCHUNK (2001), microcontroladores devem ser utilizados quando é necessário a um circuito realizar operações e procedimentos que variam conforme estímulos ou condições de um ambiente, comandos e procedimentos, análises e correções. Também existem casos onde podem simplificar sistemas ao minimizar a utilização de outros componentes eletrônicos, como portas lógicas e transistores.

2.6 *System on a Chip* — SoC

System on a Chip, ou em português sistemas em um chip, são circuitos semelhantes aos microcontroladores, pois, são sistemas completos que também se baseiam em microprocessadores. Entretanto, os SoC são circuitos integrados muito mais poderosos, apresentando capacidade de processamento muito superior aos microcontroladores, além disso, apresentam maior quantidade de memória RAM e ROM (BERGER (2002)). Para exemplificar, veja a comparação do microcontrolador *PIC18F2550* com o Raspberry Pi 3, que usa um SoC: enquanto o PIC opera em até 48MHz, cada um dos quatro núcleos do Broadcom BCM2837 ARM Cortex-A53, presente no Raspberry Pi 3, opera com 1.2 GHz, ou seja, cada um de seus núcleos é 25 vezes mais rápido que o PIC. Já em memória RAM o PIC apresenta 2048 *Bytes* de RAM e o Broadcom 1 GB, ou seja, a capacidade de armazenamento do Raspberry Pi 3 é aproximadamente 500 mil vezes maior (MICROCHIP (2009))(RASPBERRY PI FOUNDATION (2016)).

Uma das principais vantagens dos SoCs é que, com tanta capacidade de processamento e memória, esses podem executar sistemas operacionais como *Linux* e *Windows*. Isso tudo torna os SoCs circuitos mais próximos aos computadores e *notebooks* do que aos microcontroladores. Portanto, os SoC são circuitos muito potentes e podem ser utilizados em aplicações mais complexas que os microcontroladores, como aqueles exigidos pelos sistemas robóticos, sistemas de automação residencial, aplicações de IoT, videogames retrô e computadores de baixo custo (BERGER (2002)).

2.7 Modulação por Largura de Pulsos — PWM

A modulação por largura de pulsos (do inglês *pulse width modulation* — PWM) é uma técnica que gera um sinal de onda quadrada com ciclo de pulso fixo, de mesma frequência

(KULKA (2007)). Esse trem de pulsos é apresentado para diversas condições na Figura 2.7.

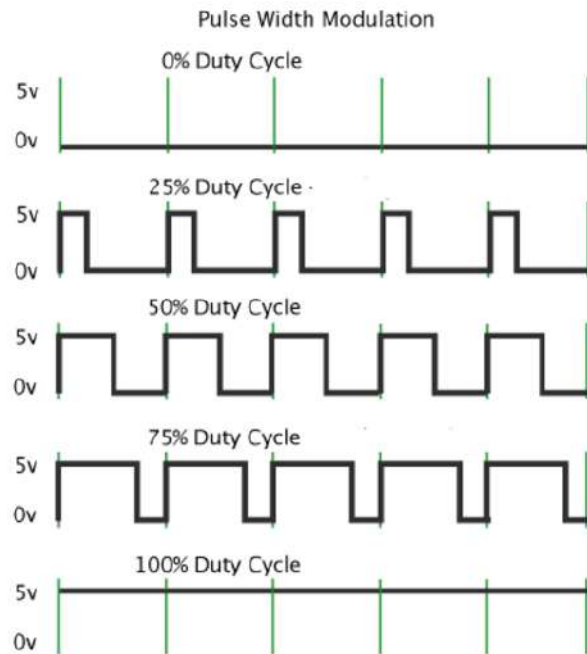


Figura 2.7: Variação do *duty cycle* de um sinal PWM. (ARDUINO (2008))

Conforme pode-se observar na Figura 2.7, apesar do sinal permanecer na mesma frequência, esse tem modificado o tempo em que o sinal é mantido em estado lógico alto. O controle do tempo ligado e desligado do sinal, proporciona uma saída com valor proporcional ao *duty cycle*, que é a porcentagem do período em que o sinal encontra-se ligado, ou seja:

$$Duty\ Cycle = \frac{T_{on}}{T} * 100. \quad (2.10)$$

Em que T é o período da frequência e T_{on} é o tempo que o sinal permanece em nível lógico alto. Portanto, quando se deseja uma tensão de saída menor que o referente ao nível lógico alto, deve-se controlar o tempo em que o sinal permanece ligado, ou seja, o *duty cycle*. O valor médio de tensão deste ciclo é o que é observado pela carga na saída, portanto, a técnica de modulação por largura de pulso possibilita que, através de um sinal digital, gere-se um sinal de saída com um valor intermediário, similar a um sinal analógico (KULKA (2007)). Assim, tem-se:

$$v_{out} = v_{in} Duty\ Cycle = v_{in} \frac{T_{on}}{T} \quad (2.11)$$

2.8 Ponte H

A ponte H é um circuito muito utilizado para se controlar o sentido de rotação e velocidade de motores de corrente contínua. Sabe-se que um microcontrolador não consegue suprir a quantidade necessária de corrente e tensão para o funcionamento de um motor, portanto, a utilização da ponte H se torna necessária. Esse circuito é composto por quatro chaves eletrônicas, normalmente são utilizados transistores TBJ ou MOSFET. O acionamento planejado dessas chaves possibilita aplicar aos terminais do motor a polarização desejada. Os motores de corrente contínua apresentam um funcionamento bem simples, em que, ao se alterar a polarização dos terminais do mesmo, altera-se seu sentido de rotação. O microcontrolador, portanto, aplica sinais lógicos que sempre mantém um par de chaves aberta e outro par fechado (PATSKO (2006)). Desse modo, torna-se possível definir o sentido de rotação do motor, conforme pode ser observado na Figura 2.8.

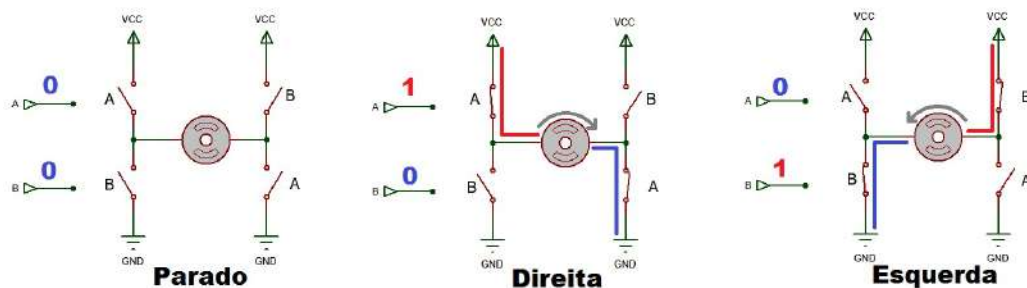


Figura 2.8: Polarização do motor através da Ponte H. (MAGAR (2016))

Conforme pode-se observar na Figura 2.8, no caso *parado* todas as chaves estão abertas e não chega tensão até o motor. No caso *direita*, ao se fechar o par de chaves *A*, um terminal do motor recebe a tensão V_{cc} e o outro fica ligado ao GND, fazendo com que o motor gire no sentido horário. Já no caso *Esquerda*, o par de chaves *B* é fechado, trocando a polarização do motor em relação ao caso anterior, o que faz com que ele gire para o sentido anti-horário.

O controle da velocidade de rotação dos motores será realizado através do valor do sinal que irá chegar até os terminais do motor. O microcontrolador opera com sinais digitais, portanto, emprega-se a técnica de modulação por largura de pulsos às chaves eletrônicas. Desse modo, torna-se possível controlar a direção e velocidade dos motores de corrente contínua (BRAGA (2010)).

2.9 Encoder de Quadratura

O *encoder* é um dispositivo eletromecânico que é utilizado para converter movimentos rotativos ou deslocamentos lineares em pulsos elétricos, gerando uma determinada quantidade de pulsos por volta de um eixo ou por unidade de deslocamento linear. Existem diferentes tipos de *encoders*, sendo os *encoders* ópticos os mais comuns. Um *encoder* óptico rotatório é formado por um disco e de cada um de seus lados um LED e um detector de luz. O disco fica acoplado ao eixo rotativo e tem padrões formados por janelas opacas e transparentes codificadas no disco. Ao girar o eixo, o disco gira, e assim a luz do LED é bloqueada pelas janelas opacas e passa pelas janelas transparentes, assim, quando a luz é identificada pelo detector de luz, sabe-se que esta passou por uma janela transparente. Ao se analisar os padrões de pulsos de onda quadrada detectados pelo detector de luz, pode-se identificar a posição do eixo e sua velocidade (NATIONAL INSTRUMENTS (2013)). Um *encoder* e seus principais componentes são apresentados pela Figura 2.9.

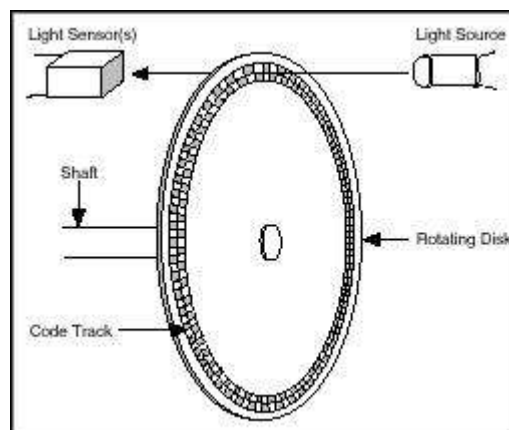


Figura 2.9: Encoder e seus principais componentes. (NATIONAL INSTRUMENTS (2013))

Os *encoders* de quadratura funcionam como os *encoders* incrementais, onde os pulsos são contados para se descobrir qual foi a variação da posição, não sendo cada posição única, como nos *encoders* absolutos. Esses dispositivos têm duas linhas de janelas opacas e transparentes no mesmo disco, onde essas linhas de janelas estão deslocadas uma em relação a outra em uma distância igual à metade de uma janela. Dessa forma, ao se analisar as respostas de ambas as saídas, pode-se identificar o sentido do movimento. A única diferença entre o *encoder* incremental e o *encoder* de quadratura é que a cada pulso que o incremental reconhece, o de quadratura reconhece dois, o que aumenta a sua precisão, por identificar mais pulsos e conseqüentemente mais posições intermediárias (UNO ROBÓTICA (2012)). O disco de um *encoder* de quadratura e as suas duas saídas de pulsos em onda quadrada, podem ser observadas na Figura 2.10.

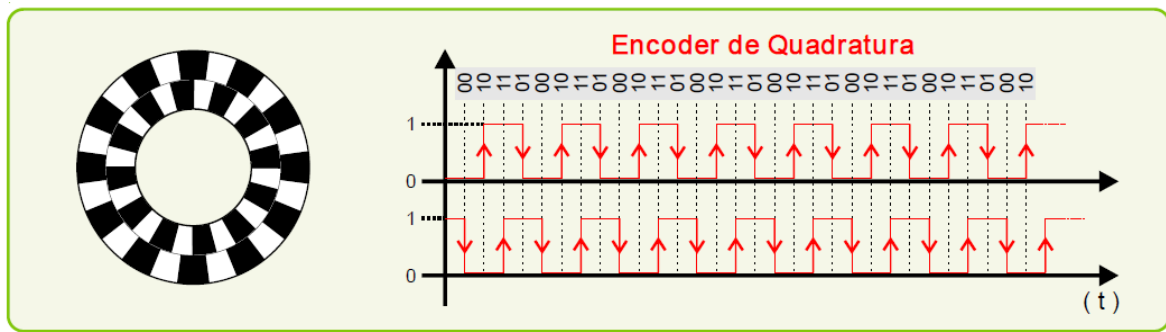


Figura 2.10: Saídas de um encoder de quadratura. (UNO ROBÓTICA (2012))

2.9.1 Conversão da leitura do *encoder* em movimento do robô

Para se converter as leituras de posição obtidas pelo *encoder* deve ser seguido um processo. O *encoder* consegue mensurar a rotação do eixo do motor e da roda, entretanto, para que esta seja convertida na distância em que o robô percorreu é necessário considerar o diâmetro da roda.

Movimento Linear

Sabe-se que quando a roda completa uma volta completa, o deslocamento produzido é igual à sua circunferência, assim, pode ser determinado usando-se a Equação (2.1). Então deve-se obter a distância percorrida a cada pulso do *encoder*, para tanto, deve-se dividir o comprimento da roda pela quantidade de pulsos do *encoder*, tal que (UNO ROBÓTICA (2012)):

$$d_p = \frac{2\pi r}{n} \quad (2.12)$$

Em que d_p é a distância percorrida a cada pulso do *encoder*, $2\pi r$ o tamanho da circunferência da roda e n é o número de pulsos do *encoder* por revolução da roda. Assim, a quantidade de pulsos necessários para percorrer uma determinada distância é dada por:

$$N = \frac{d_d}{d_p} \quad (2.13)$$

em que N é o número de pulsos do *encoder* que correspondem à distância desejada, d_d .

Movimento Circular

Ao se acionar ambas as rodas de um robô diferencial em sentidos opostos, esse irá apresentar um movimento de rotação em torno de um eixo que fica entre suas duas rodas motorizadas, como é apresentado pela Figura 2.11.

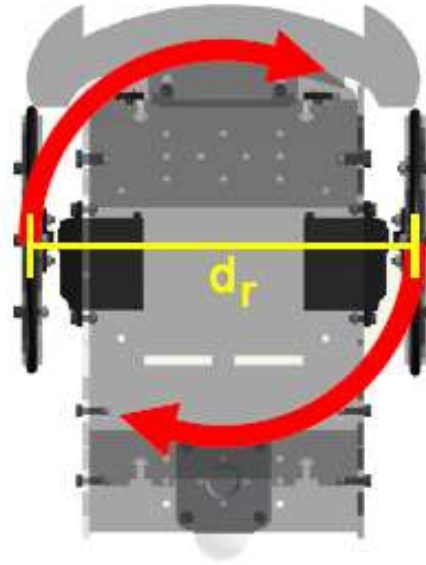


Figura 2.11: Rotação em torno do eixo do robô. (UNO ROBÓTICA (2012))

O tamanho da circunferência promovida pela rotação pode ser determinada, tal que:

$$C = \pi \cdot d_r \quad (2.14)$$

em que C é o tamanho da circunferência resultante do movimento de rotação e d_r é a distância entre as rodas do robô. Para se descobrir quantos pulsos do *encoder* devem ser gerados para que uma volta completa do robô tenha sido concluída tem-se:

$$N = \frac{C}{d_p} \quad (2.15)$$

Em que N é o número de pulsos necessários para uma rotação completa do robô. Portanto, pode-se encontrar o número de graus por pulso através de:

$$N_G = \frac{360^\circ}{N}. \quad (2.16)$$

Em que N_G é o número de graus por pulso do *encoder*. Com essas informações, pode-se então descobrir a quantidade de pulsos necessários para se realizar qualquer ângulo de rotação desejado através da Equação (2.17).

$$N_P = \frac{\theta}{N_G} \quad (2.17)$$

Em que θ é o ângulo de rotação desejado e N_P é o número de pulsos do *encoder* necessários para se rotacionar o ângulo θ .

2.10 Regulador de tensão

Um regulador de tensão é um dispositivo que mantém uma saída de tensão relativamente constante, mesmo a entrada variando dentro de uma faixa considerável. Existem diversos tipos específicos de reguladores baseados em métodos distintos para controlar este parâmetro em um circuito. Em geral, um regulador de tensão compara sua saída à uma referência fixa e minimiza esta diferença utilizando um ramo de realimentação negativa (ROBINSON (2017)).

Ainda segundo ROBINSON (2017), os reguladores passivos constituem um modelo simples que só pode ser usado quando a tensão de entrada é sempre maior que a de saída. Esse contém um resistor responsável por reduzir a tensão de entrada até o nível desejado, dissipando o excesso de tensão como forma de calor.

Já os reguladores ativos, como os conversores *buck*, *boost*, *buck-boost* e *flybacks*, apresentam elementos ativos em sua constituição, mantendo o mesmo propósito exposto acima (SEDRA; SMITH (1998)). Apesar de alguns tipos, como o *boost*, elevarem o valor de tensão em sua saída, tendo por base uma entrada inferior, o conversor *buck* é o modelo utilizado na plataforma robótica com o propósito de reduzir a tensão provida pela bateria.

Conforme exibido na Figura 2.12, o conversor buck consiste no transistor de chaveamento e no circuito volante (do inglês *flywheel*) composto pelo diodo D1, indutor L1, e capacitor C1 conectados à carga. Enquanto o transistor está ativado, a corrente flui através da carga por meio do indutor. Este, como qualquer elemento deste tipo, se opõe a mudanças na corrente e age como um componente armazenador de energia. No caso, a saída do transistor é impedida de aumentar diretamente até seu valor de pico, já que o indutor armazena esta energia, que será liberada ao circuito como uma força eletromotriz quando o transistor é desativado. Anteriormente a essa etapa, na Figura 2.12 fica claro que a corrente na carga aumenta gradualmente ao passo que o capacitor carrega. Durante este período, o diodo está polarizado de forma inversa, transformando-o em um circuito aberto (COATES (2017)).

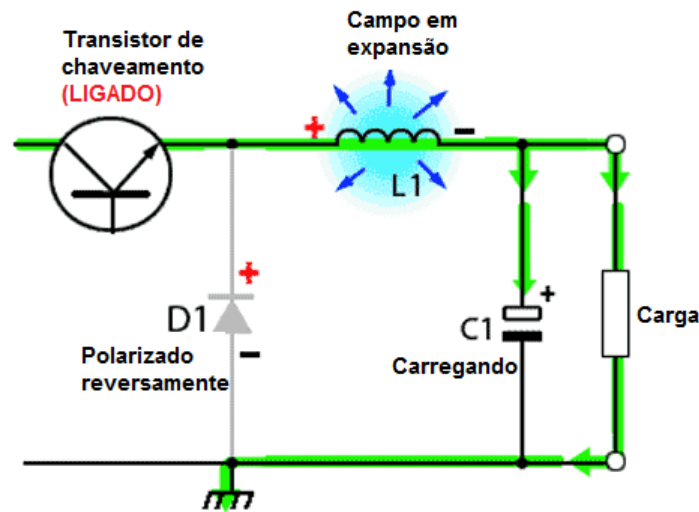


Figura 2.12: Transistor de chaveamento no período ativo. Adaptado. (COATES (2017))

Ainda seguindo o raciocínio explicativo de COATES (2017), no momento em que o transistor de chaveamento é desativado, a energia armazenada no campo eletromagnético em torno de $L1$ é descarregada no circuito, de modo que a tensão sobre este componente é inversa àquela apresentada no período anterior. Este fato garante que a corrente continue fluindo pela carga e, visto que o diodo se encontra polarizado diretamente, a corrente passa por este elemento, fechando a malha. Quando a maior parte da energia armazenada no indutor já foi dissipada, o capacitor passa a fornecer corrente para a carga até que o ciclo ativo recomece.

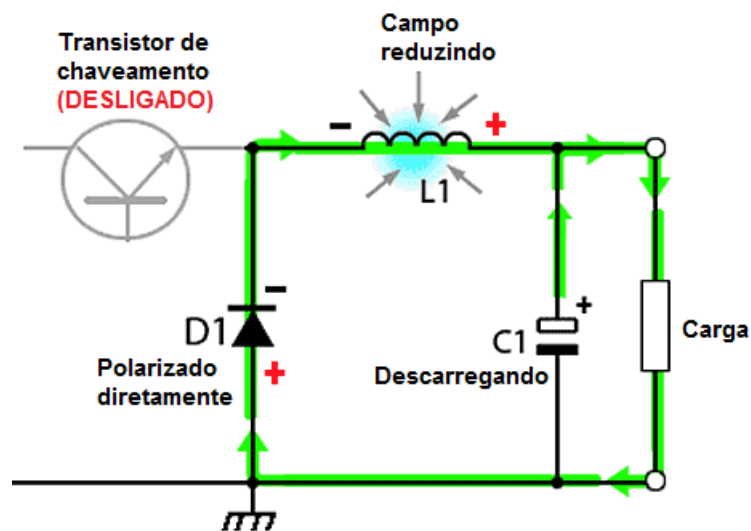


Figura 2.13: Transistor de chaveamento no período desativado. Adaptado. (COATES (2017))

Essa configuração garante que uma tensão com aspecto ondular (*ripple*) de baixa amplitude e alta frequência seja fornecida à carga, e não uma grande onda quadrada derivada diretamente do chaveamento. Essa forma de onda resultante tem um nível CC dado por:

$$V_{out} = V_{in} \frac{t_{on}}{T} \quad (2.18)$$

em que, V_{out} é a tensão de saída, V_{in} a tensão de entrada, t_{on} é o tempo ativo da onda de chaveamento e T é o período dessa curva de chaveamento. Desse modo, torna-se perceptível pela a razão entre t_{on} e T que, se a onda quadrada apresenta um ciclo ativo igual ao ciclo inativo, o conversor buck exibe uma tensão de saída equivalente à metade da tensão de entrada. Portanto, ao variar o ciclo ativo da onda de chaveamento, obtém-se valores entre $0V$ e V_{in} (COATES (2017)).

Um conversor buck pode trabalhar com dois tipos de operações: o modo contínuo e o modo descontínuo. Operar em modo contínuo significa que a corrente no indutor, representada na Figura 2.14 por I_L , nunca chega a zero. A Figura 2.14 demonstra o princípio de operação nesse modo, sendo V_i , V_o e V_L as tensões de entrada e saída respectivamente.

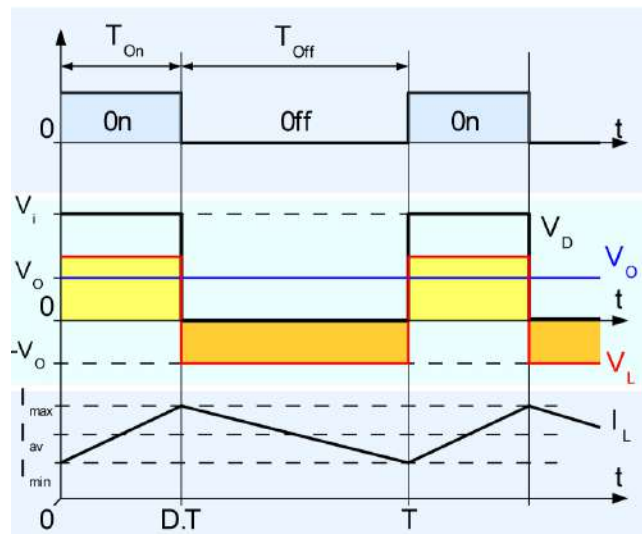


Figura 2.14: Comportamento das tensões e correntes de um buck ideal operando em modo contínuo (WOW (2017)).

Em contrapartida, no modo descontínuo apresentado pela Figura 2.15, o indutor é totalmente descarregado durante o ciclo inativo do transistor. Isso causa a dissipação da energia armazenada no capacitor de saída, resultando em maiores perdas de chaveamento. A modulação por largura de pulso pode ser utilizada para minimizar essas perdas.

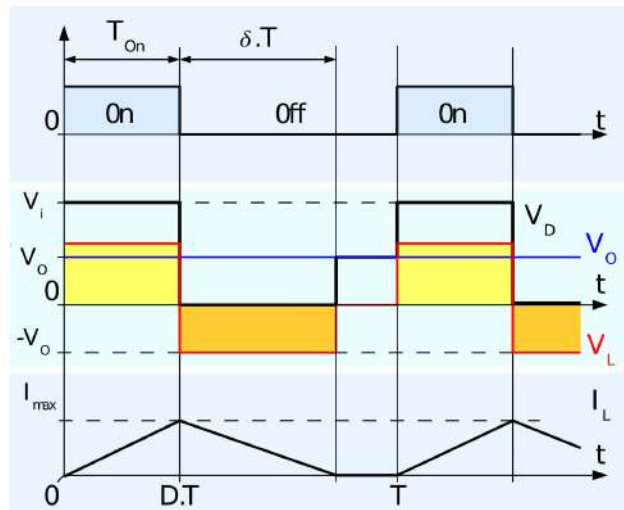


Figura 2.15: Comportamento das tensões e correntes de um buck ideal operando em modo descontínuo (WOW (2017)).

Em ambos os casos, considera-se o conversor operando em modo estável, o que significa que a energia do indutor é a mesma no início e no fim dos ciclos. Essa característica leva à conclusão de que a tensão média no indutor é nula, ou seja, a área em amarelo é igual à área em laranja nas Figuras 2.14 e 2.15.

2.11 Sensor Ultrassônico

O sensor ultrassônico é responsável pela prevenção de colisões com obstáculos não notados pelo operador através das duas câmeras. O módulo HC-SR04, apresentado pela Figura 2.16, comumente utilizado na robótica móvel, é capaz de detectar obstáculos localizados entre 20mm e 4m de distância sem qualquer contato com o objeto. Basicamente, ele é composto por quatro pinos - Vcc, Trig, Echo, GND - sendo o primeiro e o quarto sua alimentação e aterramento respectivamente. Sua tensão, corrente e frequência de operação são de 5V , 15mA e 40Hz respectivamente (FREAKS (2017)).



Figura 2.16: Módulo HC-SR04. Adaptado. (FREAKS (2017)).

De acordo com o *datasheet* do módulo, seu ângulo de medição é de 15°, entretanto, testes práticos demonstram um alcance de aproximadamente 30°, seguindo a distribuição exibida na Figura 2.17 (SAPKOTA (2012)). Essa observação é importante para garantir o posicionamento desses sensores no sistema robótico de modo que não haja interferência entre suas emissões, o que poderia ocasionar medições incertas.

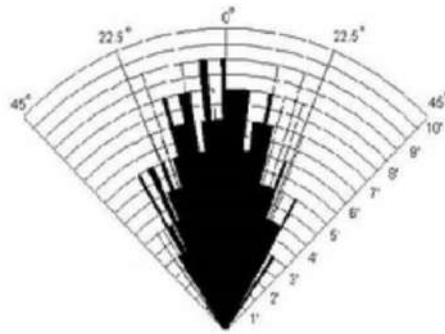


Figura 2.17: Distribuição prática do alcance do módulo HC-SR04. (SAPKOTA (2012)).

Os terminais Trig e Echo compõem o sistema transdutor do sensor, de modo que o primeiro emite as ondas sonoras, enquanto o Echo recebe as ondas refletidas em obstáculos. A partir dessa dinâmica, é possível estimar a distância do objeto em relação ao sistema robótico, tal que:

$$d = \frac{t_{ECHO_{alto}} \cdot V_{som}}{2} \quad (2.19)$$

em que $t_{ECHO_{alto}}$ é o tempo, em segundos, que o pino Echo está em nível alto e V_{som} é a velocidade do som no ar, ou seja, aproximadamente $340m/s$. Então, a expressão é dividida por dois, visto que as ondas sonoras viajam até o obstáculo e retornam, percorrendo o dobro da distância até esse objeto (FREAKS (2017)).

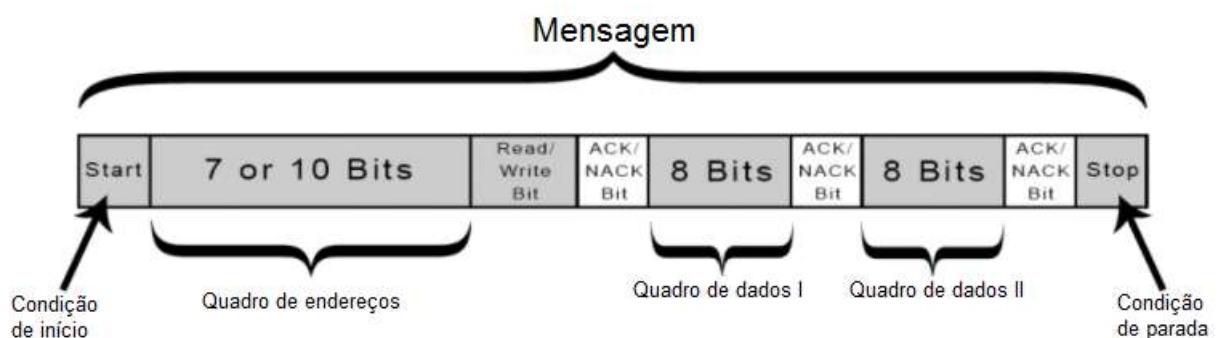
2.12 Comunicação I²C

A comunicação I²C proporciona a capacidade de conectar-se vários dispositivos escravos à um único mestre, ou de múltiplos mestres controlando um ou mais escravos. Esse protocolo utiliza apenas dois barramentos de comunicação: SDA (*Serial Data*) e SCL (*Serial Clock*). Através do primeiro, são transmitidos os dados entre os mestres e escravos, enquanto o segundo carrega o sinal de *clock*. Por ser um protocolo de comunicação serial síncrono, a informação é transferida *bit* por *bit* ao longo do barramento SDA, e a saída desses *bits* é sincronizada à taxa de amostragem do sinal de *clock* compartilhado entre mestres e escravos (LEENS (2009)). A Tabela 2.1 traz as características principais da comunicação I²C.

Tabela 2.1: Principais características da comunicação I²C.

Caraterísticas	Comunicação I ² C
Barramentos	2
Velocidade Máxima	Modo “Standard”= 100 Kbps Modo “Fast”= 400 Kbps Modo “High Speed”= 3.4 Mbps Modo “Ultra Fast”= 5 Mbps
Síncrono ou Assíncrono?	Síncrono
Serial ou Paralelo?	Serial
Número máximo de mestres:	Ilimitado
Número máximo de escravos:	1008

A transmissão de dados é feita através de mensagens, as quais são constituídas de quadros de dados. Cada mensagem possui um quadro de endereçamento que contém o endereço binário do escravo, seguido de quadros de dados e condições de *start/stop*, *read/write* e ACK/NACK entre eles, conforme exibido na Fig. 2.18. Os *bits* ACK e NACK são usados para informar se os *frames* de endereçamento ou dados foram recebidos com sucesso, enviando ACK (*acknowledge*) em caso positivo, e NACK (*no-acknowledge*) para negativo (LEENS (2009)).

Figura 2.18: Estrutura de mensagem da comunicação I²C. Adaptado. (BASICS (2017)).

De forma geral, esse protocolo apresenta vantagens como a necessidade de apenas dois barramentos, o suporte à múltiplos mestres e escravos e os *bits* de confirmação da transmissão bem sucedida de cada quadro de dados. Entretanto, como desvantagens, ressalta-se o limite de 8 *bits* para a palavra de dados e as taxas de transferência mais baixas que a comunicação *Serial Peripheral Interface* —SPI.

2.13 Servidor Web

Um servidor Web (*Web server*), referindo-se ao *hardware*, é o computador que armazena arquivos componentes de *sites*, como documentos *HyperText Markup Language* — HTML ou imagens, e os entrega para o usuário final. Ele é conectado à Internet e pode ser acessado através de seu *Domain Name System* — DNS. Já o servidor Web referindo-se ao *software*, inclui as diversas partes que controlam como os usuários acessam os arquivos armazenados, sendo no mínimo um servidor *HyperText Transfer Protocol* — HTTP. Um servidor desse tipo é um programa capaz de entender endereços da Web e o protocolo usado para visualizar páginas na Web (HTTP). Basicamente, quando um navegador necessita de uma página armazenada em um servidor Web, ele faz a requisição desse arquivo por meio do HTTP. Quando esse pedido chega ao servidor correto (*hardware*), o servidor HTTP (*software*) envia o documento de volta, através do protocolo HTTP, conforme ilustrado na Figura 2.19 (MOZILLA (2017)).

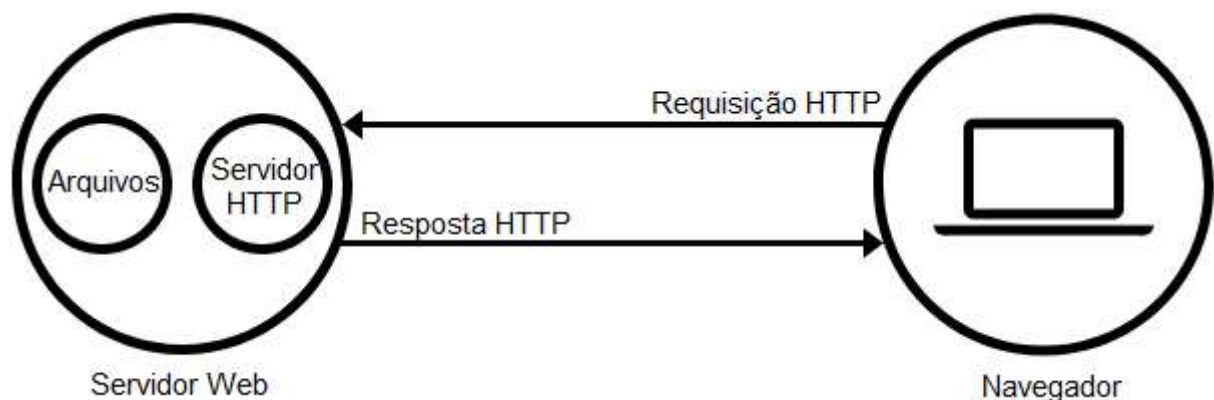


Figura 2.19: Ilustração do acesso de um servidor Web por um navegador. Adaptado. (MOZILLA (2017)).

2.14 XMLHttpRequest

XMLHttpRequest é uma *Application Programming Interface* — API¹ que proporciona ao cliente a capacidade de transferência de dados entre um cliente e um servidor conforme apresentado pela Figura 2.20. Se apresenta como um método simples de recuperar dados de um URL sem que seja necessária a atualização de toda a página Web. Assim, é possível alterar apenas uma parte do conteúdo sem interromper o que o usuário esteja fazendo

¹API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web. Ela é criada quando uma empresa de software tem a intenção de que outros criadores de software desenvolvam produtos associados ao seu serviço. CANALTECH (2017)

(VAN KESTEREN; JACKSON (2007)). O XMLHttpRequest foi criado pela Microsoft e adotado pela Mozilla, Apple e Google, podendo ser usado para recuperar qualquer tipo de dados, e não apenas XML, incluindo protocolos diferentes como HTTP ou mesmo texto puro. Desse modo, ele proporciona outras vantagens aos desenvolvedores, como a possibilidade de requisitar e receber dados do servidor após o carregamento da página e em segundo plano (ADEYEYE; MAKITLA; FOGWILL (2013)).

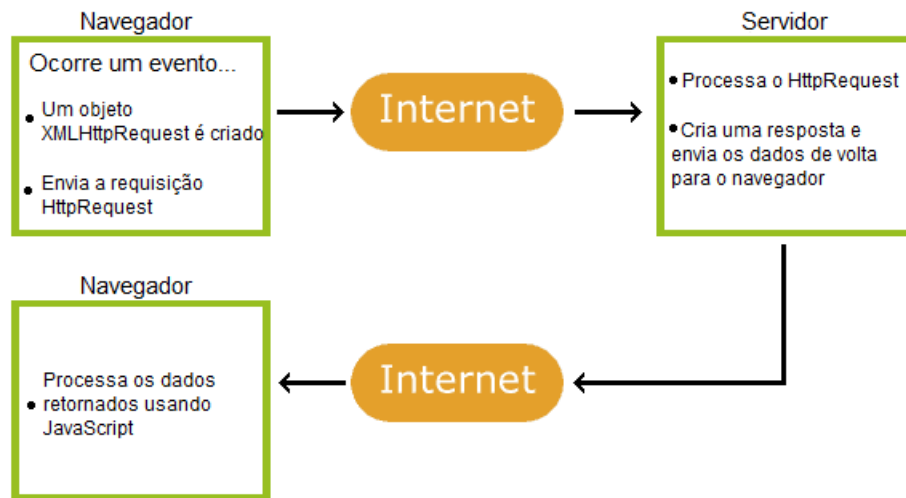


Figura 2.20: Ilustração do funcionamento de um XMLHttpRequest. Adaptado. (OVERFLOW (2012)).

2.15 *Common Gateway Interface*

Common Gateway Interface — CGI é a especificação de uma interface que possibilita aos servidores Web o acesso à funcionalidades oferecidas por programas sendo executados no ambiente da máquina servidora. Por meio dessa interface, é possível conectar uma base de dados à Web ou gerar conteúdos de uma página HTML de forma dinâmica (RICARTE (2002)). A CGI proporciona um meio consistente para os dados passarem de uma requisição do usuário para o programa ou aplicação, e então, retornar ao usuário. Isso garante que a aplicação possa ser usada independentemente do sistema operacional utilizado pelo servidor (ROUSE (2005)).

De acordo com RICARTE (2002), o servidor Web reconhece uma requisição CGI quando o *Uniform Resource Locator* — URL especificado na solicitação identifica um arquivo executável localizado em um diretório dentro dos recursos Web disponibilizados aos clientes. A Figura 2.21 ilustra a função de um *script* CGI, que pode ser utilizado em aplicações como o processamento de mapas, criação de conteúdos personalizados e servir como interface com uma base de dados.

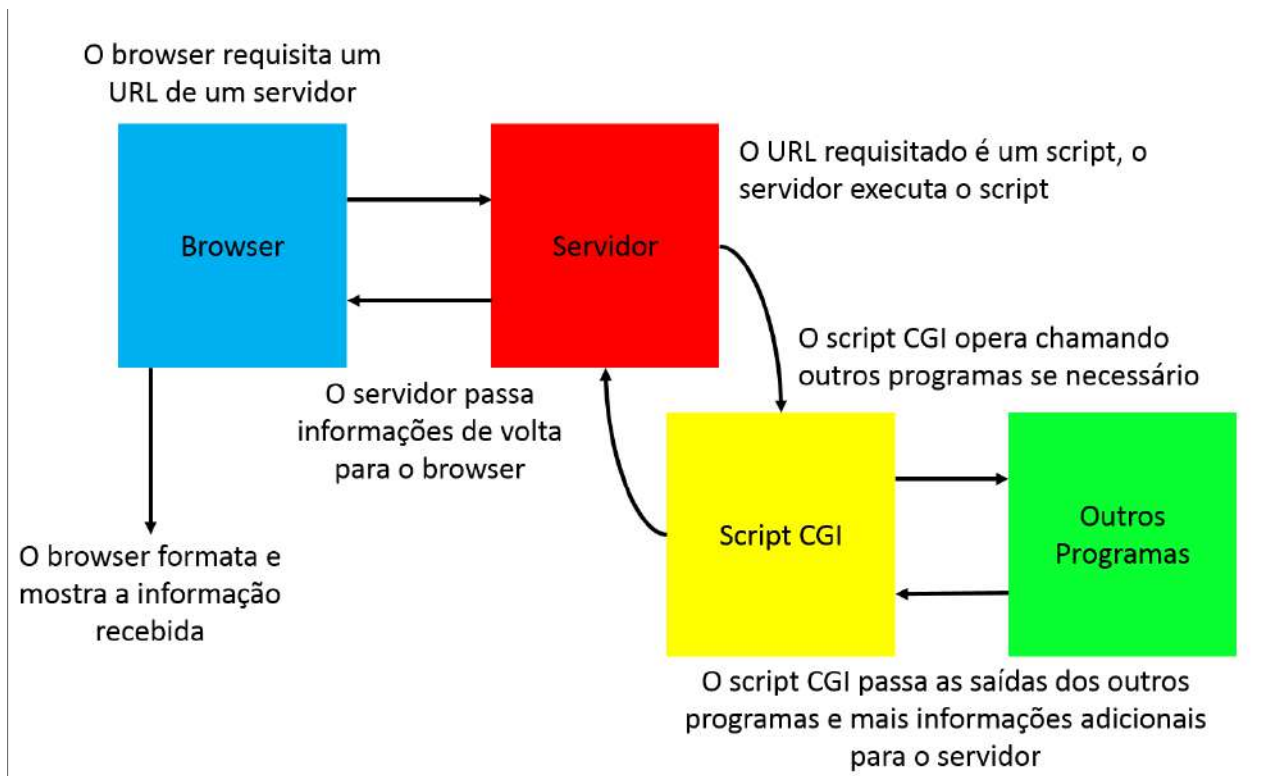


Figura 2.21: Visão geral de um sistema Web utilizando uma CGI. (OLIVEIRA C. NEVES E. (2000)).

Projeto

Neste capítulo são apresentadas as especificações de projeto e os procedimentos utilizados para o desenvolvimento mecânico, eletrônico e da programação do trabalho, aplicando-se os conceitos tratados no Capítulo 2.

3.1 Especificações de Projeto

Conforme descrito no capítulo anterior, o projeto e construção de uma plataforma robótica móvel deve levar em consideração as características do ambiente para o qual será projetada. Desse modo, deve-se atentar às restrições de deslocamento e dimensionamento impostas pelo terreno, ajustando-se os componentes para minimizar os possíveis problemas durante sua utilização. Portanto, visando a aquisição de fotografias esféricas de ambientes internos, foram levantadas as especificações e exigências para o protótipo:

- massa total de $3Kg$;
- velocidade máxima de $0,2m/s$;
- distância de aceleração de $0,05m$;
- comprimento e largura máximos de $0,6m$;
- piso plano, sem irregularidades;
- comunicação Wi-Fi;
- interface de controle do usuário em ambiente WEB;
- autonomia energética mínima para uma hora de funcionamento contínuo;

- O tubo telescópico:
 - deve apresentar comprimento necessário para simular a altura dos olhos do brasileiro médio;
 - possuir acoplamento universal para câmeras, parafuso de 1/4 de polegada;
 - suportar pelo menos 125g, peso referente à câmera Ricoh Theta S;
 - ser resistente e leve.

3.2 Mecânica

3.2.1 Topologia do robô

Conforme fundamentado no capítulo anterior, a definição do tipo de roda e da topologia do robô é uma etapa essencial para a garantia de uma cinemática satisfatória. Baseando-se nas topologias estudadas, foi concluído que a implementação de um sistema de tração diferencial na parte traseira e um rodízio na dianteira do robô, proporcionaria os três graus de liberdade (rotação em torno do próprio eixo e translações em x e y) necessários para movimentar-se nos ambientes propostos. Desse modo, é possível mudar a direção do robô sem causar deslocamentos lineares consideráveis, garantindo maior manobrabilidade. Essa configuração também desenvolve uma estabilidade suficiente sem a necessidade de sistemas de suspensão.

Além disso, para atender os requisitos de projeto estabelecidos, torna-se necessária a utilização de rodas com diâmetro reduzido, mantendo o centro de gravidade próximo da superfície de deslocamento para evitar problemas de estabilidade. Adotando-se rodas de 100mm de diâmetro, é obtida uma distância entre o piso e a plataforma inferior do sistema robótico que o permite superar pequenos obstáculos, mas ainda assim mantém o centro de massa baixo. As rodas especificadas são compostas de poliestireno, com uma pequena superfície de contato com o piso, apresentando um coeficiente de fricção com a cerâmica entre 0,2 e 0,4 TECNICA (2017).



Figura 3.1: Rodas de 100 mm de diâmetro (POLULU (2017)).

Tabela 3.1: Peso dos componentes

Componente	Peso Unitário (Kg)	Peso (Kg)
Raspberry Pi 3 B	0,041	0,041
Camera Pi	0,003	0,003
Cartão de Memória	0,002	0,002
Motor	0,107	0,214
Suporte Motor	0,0085	0,017
Rodas	0,13	0,26
Fixação das rodas	0,0151	0,0302
Driver	0,0033	0,0033
Bateria	0,408	0,816
Sensor Ultrassom	0,0085	0,0255
Câmera	0,136	0,136
Suporte da Câmera	1,22	1,22
	Peso Total:	2,768

3.2.2 Dimensionamento do motor

O dimensionamento do motor foi realizado de modo a atender com segurança as especificações de projeto estabelecidas na Seção 3.1. Para tal, fez-se um levantamento da previsão de massa do robô, conforme apresentado na Tabela 3.1.

Como pode ser verificado na Tabela 3.1, o robô apresenta uma massa prevista de $2,768Kg$. A partir desse levantamento, assumiu que a massa máxima do robô não ultrapassa $9,0Kg$, o que garante ao projeto um coeficiente de segurança de aproximadamente 3,25. Esse valor possibilita a inclusão de novos componentes no robô sem comprometer o desempenho dos motores.

Conforme apresentado na Seção 3.2.1, nesse projeto é desenvolvido um robô diferencial, que apresenta no sistema de tração, rodas de poliestireno com diâmetro de $100mm$. Observando a aplicação e o ambiente de operação do robô, definiu-se para avaliação, que ele deve se movimentar com uma velocidade mínima de $0,05m/s$ e máxima de $0,20m/s$. Além disso, assumiu-se que a distância de aceleração do robô deve ser de $0,05m$, ou seja, após percorrida essa distância o mesmo deve atingir sua velocidade de deslocamento. Por fim, verificou-se que o coeficiente de atrito em piso cerâmico ¹ pode assumir valores entre 0,2 e 0,4. Tomando os parâmetros e condições do ambiente, assim como também, a característica das rodas, a partir das equações (2.4), (2.6) e (2.7), definiu-se as características mínimas dos motores de tração, conforme pode ser verificado na Tabela 3.2.

Conforme pode ser verificado na Tabela 3.2, a situação em que o motor é mais exigido é a condição em que o coeficiente de atrito do piso com a roda é de 0,4, e velocidade de

¹Foi assumido que o piso dos ambientes internos onde o robô se desloca, possui acabamento cerâmico. Essa decisão foi tomada com base no estudo apresentado por INMETRO (2017), o qual demonstra a tendência nacional por pisos cerâmicos.

Tabela 3.2: Parâmetros para avaliação das características dos motores em função da velocidade de deslocamento e condições ambientais.

Coefficiente de atrito	Velocidade máxima (m/s)	RPM	Torque ($N.m$)
0,2	0,05	9,5493	0,2981
	0,1	19,0986	0,3093
	0,15	28,6479	0,3281
	0,2	38,1972	0,3543
0,3	0,05	9,5493	0,4452
	0,1	19,0986	0,4565
	0,15	28,6479	0,4752
	0,2	38,1972	0,5015
0,4	0,05	9,5493	0,5924
	0,1	19,0986	0,6036
	0,15	28,6479	0,6224
	0,2	38,1972	0,6486

deslocamento de $0,2m/s$, quando é necessário um torque mínimo do motor de $0,6486N.m$, e velocidade mínima de $38,2rpm$.

Portanto, a partir das parâmetros apresentados na Tabela 3.2, determinou-se que o motor utilizado deveria ser um motor de corrente contínua de baixa potência, modelo 3268 da Pololu. Esse motor apresenta como principais características: tensão de alimentação de $12V$, caixa de redução de $172 : 1$, velocidade de rotação de $31RPM$, *encoder* de quadratura e torque de $1,27N.m$. Maiores informações sobre as características do motor podem ser encontradas no Apêndice B.

3.2.3 Suporte e Fixação

Para que os motores pudessem ser fixados às rodas e à base do robô, deveriam ser definidos suportes compatíveis e confiáveis para tal aplicação. Para tanto, foram escolhidos suportes próprios para o motor e rodas que já especificados, optando-se pelo mesmo fornecedor do motor. Informações detalhadas das dimensões do sistema de fixação estão expostas no Apêndice B.

Objetivando-se a certificação de que o sistema de fixação suporta as tensões envolvidas na dinâmica de propulsão do sistema robótico, foram calculadas a tensão normal e de cisalhamento máximas. A primeira se refere à força aplicada pelo peso da plataforma robótica sobre a área superior do suporte. Já a tensão de cisalhamento é resultante do torque do motor agindo sobre os parafusos de fixação que impedem o sistema de rotacionar-se junto com o motor.

De acordo com o fornecedor, tanto o suporte, como também os parafusos, são constituídos de alumínio. Segundo a Associação Brasileira do Alumínio, esse material puro

apresenta um limite de escoamento, ou seja, a tensão em que o material começa a deformar-se plasticamente (0,2% do comprimento original), de aproximadamente $12,7MPa$ ou $1,3kg/mm^2$ (ABAL (2017)).

Assim, primeiramente, foi calculada a tensão normal máxima admitida pelo suporte, tal que:

$$\sigma_{normal} = \frac{F}{A} \quad (3.1)$$

em que F é a força aplicada perpendicularmente à superfície de sustentação, cuja área é dada por A . Conforme pode ser visualizado no desenho técnico da peça no Apêndice B, essa face retangular de lados $21,7mm$ e $52mm$ gera uma área de $1,1284.10^{-3}m^2$, a qual deve suportar um terço do peso gerado pela massa total do robô de $9Kg$, ou seja $29,43N$, resultando em:

$$\sigma_{normal} = \frac{29,43N}{1,1284.10^{-3}m^2} = 26081 \frac{N}{m^2} \approx 26KPa$$

Comparando-se esse resultado ao limite de escoamento definido, percebe-se que o suporte será submetido à uma tensão cerca de 487 vezes menor do que seu valor máximo suportado, atendendo satisfatoriamente aos requisitos de desempenho.

Já a tensão de cisalhamento é gerada pelas forças cortantes agindo sobre os parafusos devido ao torque do motor, que tende a fazer o acoplamento girar, calculada por:

$$\tau = \frac{F}{A} \quad (3.2)$$

em que F é a força tangencial agindo na área definida por A . Nesse caso, o torque está sendo aplicado a uma distância de aproximadamente $13,5mm$ existentes entre o eixo do motor e os furos dos parafusos de fixação. Nesse comprimento, o torque de $1,27N.m$ gera uma força de $94,07N$, aplicada na área em que o parafuso tangencia a estrutura de fixação. Assim, apresenta-se uma área retangular de lados $2mm$ (espessura do suporte) e $3mm$ (diâmetro dos parafusos). Ao utilizar a Equação 3.2, obtém-se uma tensão de cisalhamento total de:

$$\tau = 15,679MPa \quad (3.3)$$

Esse valor indica um fato importante, que ao superar o limite de escoamento do alumínio, fica clara a necessidade de múltiplos parafusos para fixar o suporte e evitar deformações plásticas. Logo, devem ser inseridos mais parafusos nos outros furos para diminuir o esforço mecânico nesses componentes. O suporte apresenta sete furos passíveis de utilização, levando à uma situação em que a tensão de cisalhamento em cada parafuso é de apenas $2,239MPa$, consideravelmente menor do que o limite estabelecido.

O modo através do qual a roda acopla-se ao motor é apresentada pela Figura 3.2. Foi realizada uma simulação da montagem contendo: o motor de $25mm$ de diâmetro, a roda

de 100mm de diâmetro, o suporte para o motor e o adaptador para fixá-los. O mecanismo resultante é exposto na Figura 3.3.

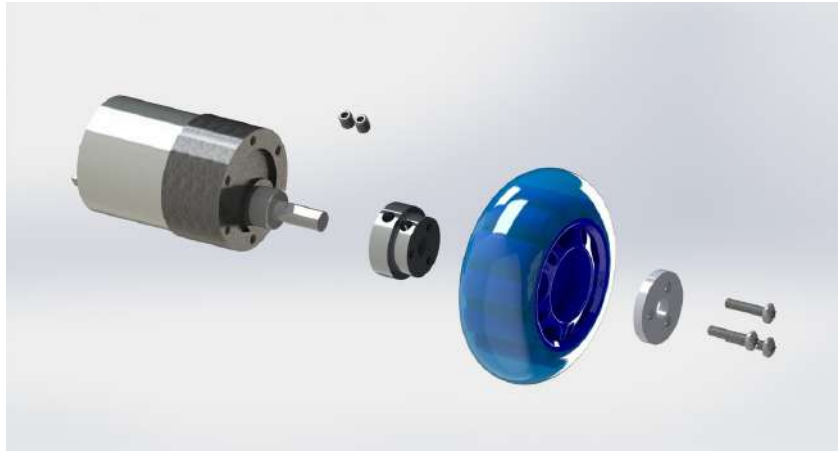


Figura 3.2: Vista explodida da montagem do adaptador de fixação. (POLULU (2017))

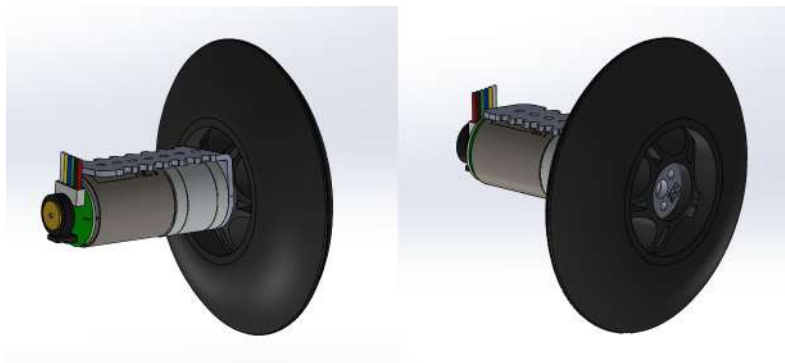


Figura 3.3: Simulação da montagem do motor, roda e suportes.

3.3 Tubo telescópico

O acoplamento da câmera 360° junto ao robô é realizado com auxílio de um tubo telescópico com suporte de fixação da câmera, o qual deve atender as especificações de projeto apresentado na Seção 3.1.

A fim de determinar a altura do bastão de fixação da câmera, deve-se observar a princípio a altura média dos brasileiros. Segundo pesquisa, publicada na revista científica eLife, a altura média do homem brasileiro é de $1,73\text{m}$, e a mulher, $1,60\text{m}$ (G1 (2016)). Já a distribuição da população brasileira, segundo o IBGE, se dá em 48,4% de homens e 51,6% de mulheres (R7 (2015)). Fazendo-se a média ponderada com essas informações encontra-se:

$$H_m = 0,484 * 1,73 + 0,516 * 1,6 = 1,66\text{m} \quad (3.4)$$

Portanto, as lentes da câmera devem estar aproximadamente na altura dos olhos de uma pessoa de $1,66m$. Segundo ROEBUCK; KROEMER; THOMSON (1975), a altura dos olhos de uma pessoa é cerca de 93,6% da altura da mesma. Desse modo, calcula-se que a altura média dos olhos dos brasileiros é de aproximadamente $1,55m$. Ao fotografar nessa altura, gera-se uma experiência do usuário mais imersiva, através de uma reprodução mais fiel da realidade, além de ser generalista para atender diversas pessoas com o mesmo produto. Pequenas variações nessa altura podem ser realizadas sem grandes prejuízos.

Conforme pode ser observado no diagrama esquemático da Figura 3.4, a base do robô, onde o tubo é fixado, fica a $70,5mm$ do chão: $50mm$ do raio da roda, mais $14,5mm$ do centro do eixo à base do acoplamento do motor e mais $6mm$ de espessura da placa de acrílico. Levando-se em consideração que a distância da base da câmera escolhida até a lente é de aproximadamente $140mm$, sabe-se que o conjunto do tubo mais o acoplamento deste à base do robô deve apresentar:

$$H_t = 1550 - 70,5 - 140 = 1339,5mm \quad (3.5)$$

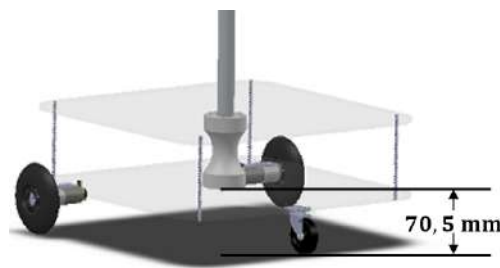


Figura 3.4: Diagrama esquemático da base do robô.

Ao analisar as opções do mercado encontrou-se o tubo telescópio *Manfrotto Compact Extreme*, apresentado pela Figura 3.5.



Figura 3.5: Manfrotto Compact Extreme (MANFROTTO (2016)).

Esse dispositivo possui um acoplamento universal de 1/4 de polegada para câmeras e suporta até 1Kg de carga. Ele é construído em alumínio, o que lhe confere resistência e uma pequena massa de 320g. Quando totalmente aberto ele alcança 51,5 polegadas, o equivalente a 1308,1mm, muito próximo dos 1339,5mm necessários para o projeto. Os 31,4mm restantes são compensados pela peça de acoplamento construída para fixá-lo à base do robô, alcançando a altura especificada. Quando totalmente fechado o tubo fica com 15,8 polegadas o equivalente a 401,32mm, o que proporciona boa facilidade ao ser transportado.

3.4 Esquemático do *Hardware*

Nesta seção apresenta-se como o *hardware* do robô é estruturado para facilitar a compreensão dos subsistemas e como eles comunicam entre si. Os principais circuitos que compõem o esquemático exposto na Figura 3.6 são abordados nas seções seguintes.

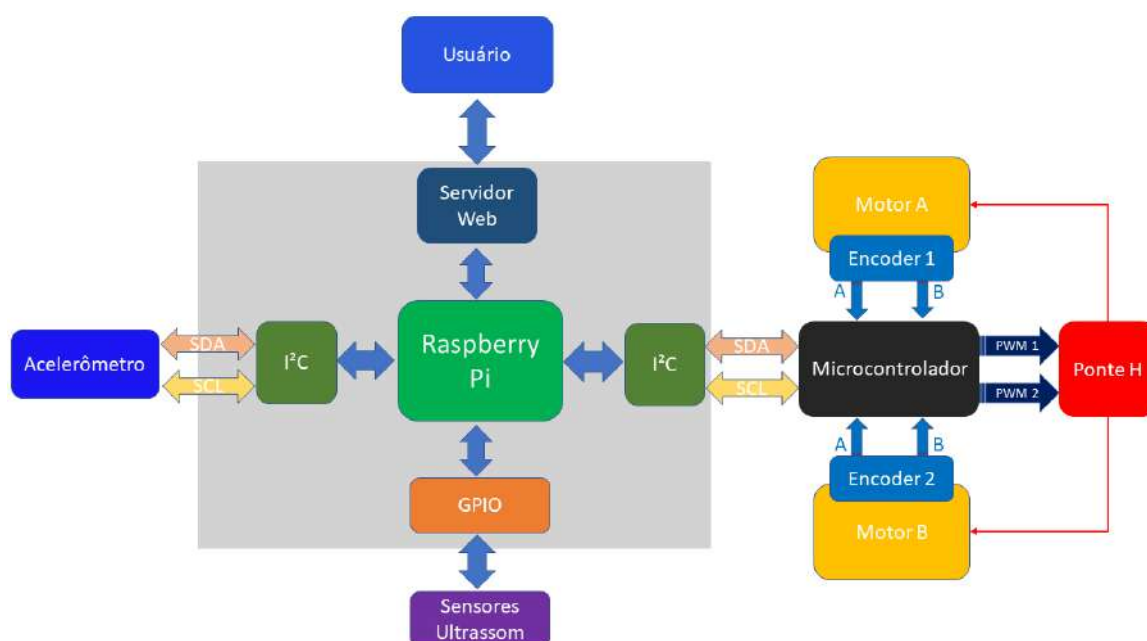


Figura 3.6: Esquemático do *Hardware* do robô.

O usuário comunica-se com o robô e o controla através de um servidor Web com o Raspberry Pi, sendo este o responsável por controlar todas as tarefas do robô. Através da comunicação I²C são obtidas as informações sobre as inclinações do robô em relação ao piso, para assim evitar sua queda. Através do GPIO é possível adicionar sensores de ultrassom para se evitar obstáculos. A parte de controle dos motores é descentralizada através da utilização de um microcontrolador, o qual é escravo do Raspberry Pi. Os motores são então alimentados e controlados através de uma ponte H, que por sua vez,

é controlada através de sinais PWM provenientes do microcontrolador. Os dados dos *encoders* dos motores são lidos também pelo microcontrolador. Essas informações e as ações que deverão ser tomadas pelo microcontrolador são comunicadas com o Raspberry Pi através do protocolo I²C.

3.5 Sensores

3.5.1 Câmera de Aquisição de Imagens 360°

Antes de 2015, era necessário um equipamento profissional composto de várias câmeras separadas e computadores poderosos para criar vídeos e imagens imersivas, como aquelas vistas no *Google Maps* no modo *Street View*. Com o desenvolvimento do *smartphone*, a câmera, processador e seu sensor foram forçados a diminuir tanto que, atualmente, as câmeras 360° se tornaram viáveis e acessíveis (GOLDMAN (2016)).

Existem diferenças técnicas entre imagens 360° e esféricas. A primeira consiste em uma fotografia capturada horizontalmente com um arco de 360°. Se apresentar a liberdade de ver acima e abaixo, denomina-se imagem esférica. Essa é composta por arcos de 360°x180°, e devido ao grande diferencial proporcionado pela possibilidade de enxergar todos os ângulos do campo de visão real, foi selecionada para compor o robô móvel desenvolvido. Ambos os tipos de câmeras são referenciados como sendo 360°, mas é conveniente ressaltar as diferenças nas especificações desses dispositivos.

Portanto, tendo em vista a óbvia importância desse *hardware* para o trabalho, foi realizada uma profunda pesquisa de opções disponíveis no mercado. O parâmetro mais importante para o julgamento destas possibilidades foi a qualidade de imagem, uma vez que será a parcela do projeto em contato com o usuário final. Portanto, lista-se a seguir as opções mais relevantes disponíveis, e suas características principais.

Ricoh Theta S

A Ricoh Theta S, apresentada na Figura 3.7, é uma câmera 360° acessível, que pode capturar vídeos em 1080p a 30fps, bem como imagens esféricas de 14 megapixels. Seu aplicativo de *smartphone* intuitivo permite aos seus usuários a transferência de conteúdo direto da câmera para o aparelho móvel. Ela é considerada a melhor câmera de realidade virtual para uso pessoal em diversos *sites* de tecnologia, como o *Cnet*, *Dailytekk*, *Best-product*, entre outros. Ela é vendida por cerca de 350 dólares, sendo capaz de transmitir vídeos 360° em tempo real. Se destaca dos concorrentes pela a qualidade das imagens esféricas paradas, graças ao seu *software* avançado de redução de ruídos, proporcionando um custo-benefício razoável. Porém, em condições de baixa exposição quando existe pouca luz ambiente, seu sensor não se comporta tão bem.



Figura 3.7: Ricoh Theta S (RICOH (2017)).

Samsung Gear 360

Esta câmera foi lançada pela Samsung junto com seus novos aparelhos celulares Galaxy S7 e S7 Edge. Suas especificações são levemente superiores às da Theta S, entretanto em vários *reviews* de usuários e *rankings* de *sites* ela é avaliada inferiormente. Ela é capaz de capturar imagens de 30 megapixels via duas lentes olho de peixe com sensores de 15MP. Resistente à respingos e poeira, apresenta mais robustez que a sua concorrente da Ricoh. Apesar de possuir vários protocolos de comunicação - *Wi-Fi*, *NFC* e *Bluetooth*, sua conectividade não é fator de destaque por estar limitada à poucos *smartphones* da marca. É possível superar essa limitação com aplicativos de terceiros, entretanto, a melhor qualidade de imagem, portabilidade e usabilidade da Ricoh Theta S sobressai perante a Gear 360.



Figura 3.8: Samsung Gear 360 (SAMSUNG (2016)).

Panono Camera

Dentre os destaques pontuados, a câmera da Panono prevalece em quase todos os aspectos. Ela consiste-se de uma esfera de $110mm$ de diâmetro, pesando aproximadamente $480g$, na qual estão fixadas 36 câmeras de 3MP, resultando em uma imagem esférica de 108 megapixels e resolução 16K. Dispõe de uma capacidade de armazenamento interno de $16Gb$, equivalente a cerca de 600 imagens, conectividade Wi-Fi e controle via aplicativo Android e iOS. Seu sensor é mais avançado que as concorrentes citadas, contando com HDR e diversos outros recursos para produzir excelentes imagens, mesmo em condições de baixa exposição. Sua utilização é muito simples, e como as outras duas, não são necessários costura ou processamento das imagens. Inicialmente dedicada à consumidores entusiastas, a Panono tem focado no uso profissional do seu dispositivo, trabalhando em otimizações para o mercado B2B. O único ponto negativo é o preço de dois mil dólares, quase seis vezes mais que a Ricoh Theta S, prejudicando bastante a relação custo-benefício, uma vez que suas imagens, apesar de melhores, não são justificativas suficientes para a diferença de investimento.



Figura 3.9: Panono Camera (PANONO (2017)).

Insta360 4K

A câmera Insta360 4K é uma das poucas câmeras disponíveis no mercado com a capacidade de resolução 4K por menos de mil dólares. Ela conta com dois sensores CMOS de 8 Megapixels e duas lentes olho de peixe, responsáveis por um ângulo de visão de 230° cada. Suas fotografias apresentam uma resolução combinada de 16,8 Megapixels de 4096x2048. Dentre suas principais características, cita-se o processamento de corte das imagens em tempo real, garantindo a esfericidade das mesmas, conexão via Wi-Fi que permite seu controle por meio do aplicativo para iOS ou Android. Seus sensores Sony garantem imagens mais brilhantes e coloridas, e sua bateria provém uma autonomia de até cem minutos de gravação contínua. Seu ponto fraco é a qualidade do áudio dos vídeos, entretanto, essa deficiência não é relevante para as aplicações desejadas. Comparada à Ricoh Theta S, a qualidade das imagens esféricas paradas é consideravelmente superior, mesmo o investimento sendo similar.



Figura 3.10: Câmera Insta360 4K (SIXTY (2016)).

3.5.2 Sensor Ultrassônico

A prevenção de colisões do robô móvel com os objetos existentes nos ambientes internos de imóveis é essencial para garantir a integridade de seus componentes, principalmente da câmera 360°. Desse modo, foram estudadas configurações de posicionamento dos sensores ultrassônicos de modo a minimizar o número de componentes necessários, mas evitando a presença de áreas não mapeadas que poderiam causar colisões.

Portanto, notou-se que a condição na qual o robô está mais próxima de obstáculos ocorre quando o mesmo atravessa as portas entre cômodos. Considerando, então, a largura mínima de uma porta igual a $0,6m$, e o sistema robótico cruzando a abertura exatamente em seu centro, tem-se uma distância de $0,1m$ em ambos os lados. Logo, posicionando teoricamente, o sensor na extremidade da placa de acrílico, forma-se um triângulo retângulo com ângulo de 30° referente ao alcance prático do sensor e seu cateto adjacente de $0,1m$, conforme pode ser verificado na Figura 3.11.

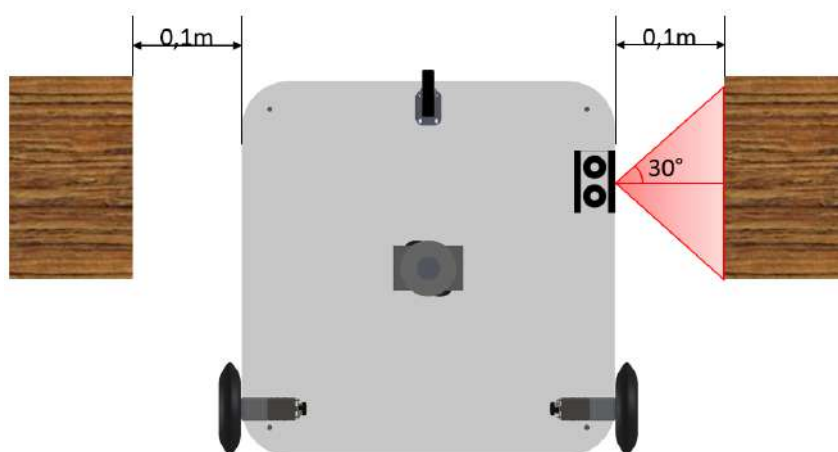


Figura 3.11: Esquemático de posicionamento e alcance do sensor ultrassônico.

Então, nesse cenário, o cateto oposto ao ângulo vale $57,735\text{mm}$ o que implica na ocorrência de um ponto cego na região próxima ao chanfro frontal do robô. Ao executar uma translação do sensor em direção ao centro da placa, aumenta-se o cateto adjacente, e conseqüentemente, a área do robô não captada diminui. Com o posicionamento ajustado, o cateto oposto do triângulo aumenta seu comprimento para $86,603\text{mm}$, resultando em um alcance de $173,206\text{mm}$, conforme representado na Figura 3.12. É importante ressaltar que os sensores laterais são mais utilizados durante os movimentos de rotação, portanto, baseando-se na dinâmica desse movimento, a faixa alcançada pelo sensor ultrassom é suficiente para evitar as colisões laterais.

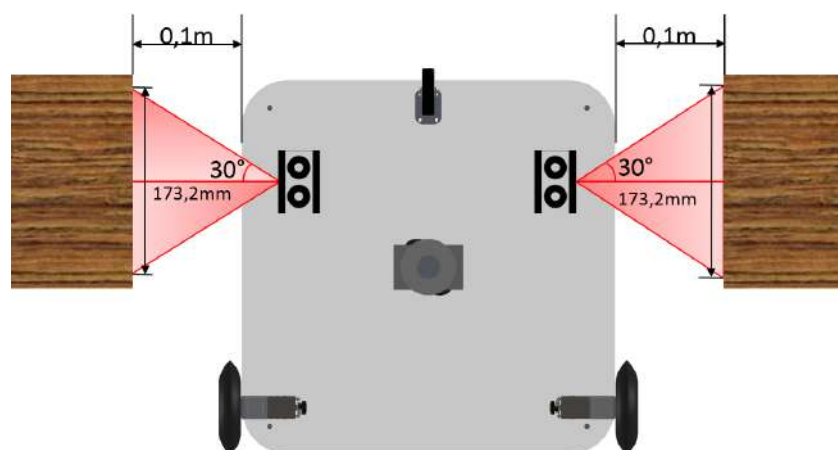


Figura 3.12: Esquemático de posicionamento e alcance dos sensores laterais.

Na área frontal, bem como na traseira do robô, são posicionados dois sensores à 50mm das extremidades diante os emissores e receptores dos componentes. Lateralmente, estão afastados por 10mm das extremidades. Seguindo a mesma análise de alcance dos sensores laterais, porém alterando a distância do robô até o obstáculo para 150mm , tem-se uma

faixa total de detecção de $430,940mm$, conforme pode ser observado na Figura 3.13. Cada sensor detecta $230,94mm$, entretanto, existe uma área onde há sobreposição das ondas, justificando o valor final diferente da soma das áreas de cada componente. Nessa configuração, de fato, existem pontos não detectados pelo conjunto, porém, o sistema robótico apresenta duas câmeras adicionais com transmissão de vídeo em tempo real. Desse modo, o operador conta com diversos meios para compreender o território ao seu redor.

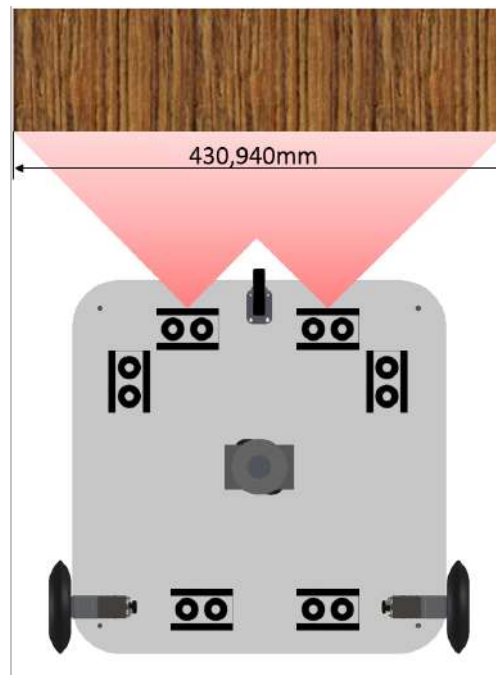


Figura 3.13: Configuração final dos sensores ultrassom evidenciando o alcance dianteiro.

3.6 Hardware

3.6.1 Opções de Mercado

O robô móvel projetado envolve grande processamento de dados, ao trabalhar com a análise de diferentes sensores, captura e envio de imagens, comunicação *wireless* e transmissão de parâmetros para a operação dos atuadores. Por isso, uma central com grande capacidade de processamento de dados é fundamental, o que justifica a escolha de um embarcado baseado em um SoC. As principais opções no mercado atualmente e algumas de suas características são listadas na Figura 3.14:





Nome	Placa	Dimensões	Processador	Clock	GPIO	RAM	Wi-fi	Bluetooth	HDMI	USB	Ethernet
Raspberry Pi 3 Model B		85 x 56 x 17 mm	Broadcom BCM2837 ARMv8 Cortex-A53 Quad-Core	1.2 GHz	40 pinos	1GB	sim	4.1 BLE integrado	sim (1)	4x2.0	sim
BeagleBone Black Rev.C		86,36 x 54,61 mm	AM335x ARM Cortex-A8	1 GHz	92 pinos	512MB	não	não	sim (1)	2x2.0	sim
Intel Galileo Gen 2		100 x 70 mm	Intel Quark SoC X1000 Application	400 MHz	32 pinos	256 MB	não	não	não	2x2.0	sim
Intel Edison		127 x 72 x 12 mm	Intel Atom SoC dual-core	500MHz	40 pinos	1GB	sim	4.1 BLE integrado	não	1x2.0 e 2xMicro	somente com adaptador

Figura 3.14: Comparativo entre embarcados.

Dentre as opções analisadas, pode-se observar que o Raspberry Pi 3 apresenta dimensões reduzidas em comparação aos concorrentes, além de *clock* mais rápido. Ele também dispõe de 1 GB de memória RAM, dividindo o primeiro lugar com o Intel Edison. É também destaque no critério de conectividade, sendo o único que apresenta todos os pontos observados (*Wi-Fi*, *Bluetooth*, HDMI, USB e Ethernet). No caso do USB, é também, entre os analisados, o que apresenta o maior número: quatro portas USB 2.0. Entre os quesitos analisados ele só não é superior no número de pinos GPIO, pois, tem 40, enquanto o BeagleBone Black tem 92. Entretanto, a quantidade de pinos GPIO do Raspberry Pi 3 é suficiente para o projeto proposto.

O Raspberry Pi 3 também tem como vantagem a sua grande popularidade, que tem como consequência o maior número de periféricos compatíveis com a placa e a maior oferta dos mesmos no mercado. Além disso, existe uma maior comunidade de usuários deste embarcado, o que torna mais fácil encontrar informações, suporte e materiais referentes a projetos com o mesmo. Portanto, a análise das opções presentes no mercado mostra o Raspberry Pi 3 como a melhor escolha para o projeto.

3.6.2 Raspberry Pi

O Raspberry Pi 3 apresenta um Broadcom BCM2837, processador *quad core* SoC (*System-on-Chip*) ARM Cortex-A53, de 64 *bits* e *clock* de 1.2GHz. Ele também conta com 1 GB LPDDR2 de memória RAM, além de adaptadores Wifi e Bluetooth 4.1 BLE integrados. Tem uma unidade de processamento gráfico (GPU, do inglês *Graphics Pro-*

cessing Unit) Videocore IV 3D, que permite a visualização de fotografias e vídeos em *Full HD* (1920 × 1080).

Esta placa apresenta vários conectores para facilitar sua integração com outros periféricos, entre os principais pode-se citar: saída HDMI, conector *Ethernet*, 4 portas USB 2.0, conectores para câmera e *display LCD*, entrada para cartão de memória e uma régua GPIO (General Purpose Input/Output).

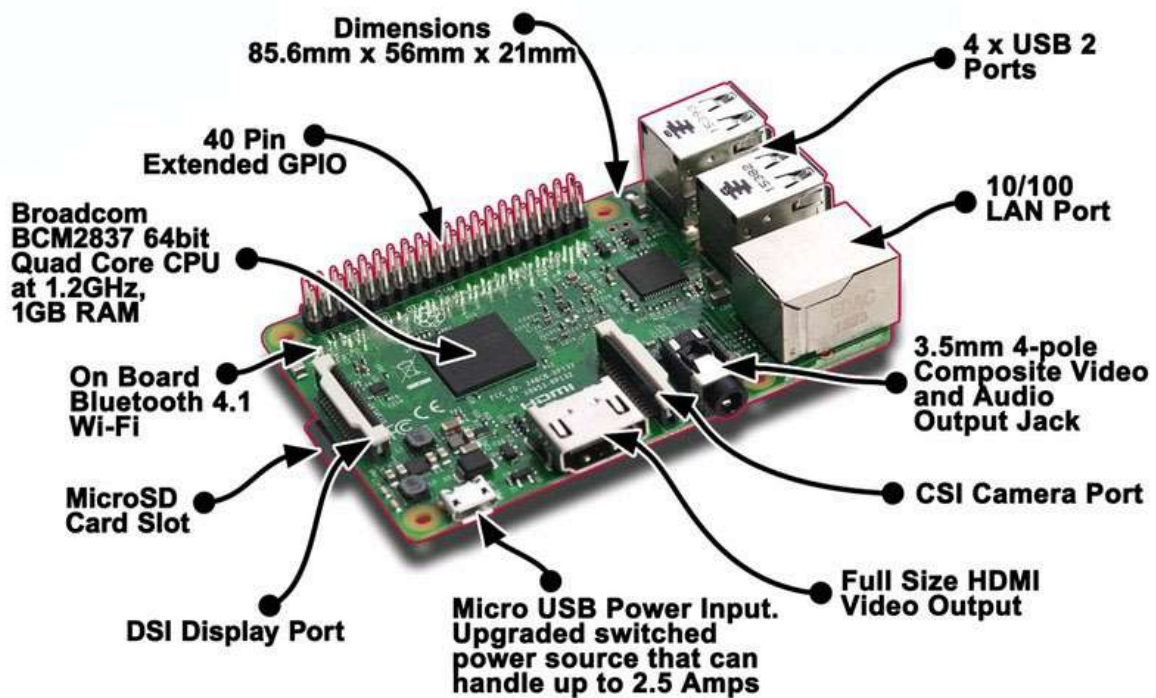


Figura 3.15: Features do Raspberry Pi 3 (ELEMENT14 (2016)).

O *GPIO Header* é um barramento de pinos conectores de entrada e saída de sinais digitais. Segundo Warner (2013), O *GPIO* é extremamente importante para o Raspberry Pi, pois, esses pinos representam a forma como se pode expandir a placa para que esta interaja com *hardware* externo, como microcontroladores, motores, sensores, etc.

O mapa de pinos do *GPIO* do Raspberry Pi 3 é apresentado pela Figura 3.16.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Figura 3.16: Raspberry Pi 3 GPIO Header (ELEMENT14 (2015)).

Uma particularidade do GPIO do Raspberry Pi 3 é que esse funciona com nível alto lógico de 3.3V, diferentemente dos microcontroladores PIC e das placas *Arduino*, que funcionam com 5V de nível alto lógico. Por isso, não se pode aplicar 5V nos pinos do Raspberry Pi, o que irá danificá-los, tornando-se fundamental a utilização de circuitos que realizem as devidas conversões de tensão, quando necessário. Além disso, uma corrente máxima de 50mA pode ser utilizada por todos os pinos de GPIO juntos, portanto, não devem ser conectadas cargas que consomem mais que 16mA por pino, e que a soma não ultrapasse 50mA em todos os pinos. Uma solução possível é a utilização de circuitos de acionamento, com transistores e relés.

A seguir são citadas as funções de cada um dos pinos do GPIO do Raspberry Pi 3:

- Alimentação de 3.3V (a mesma tensão de seus outros pinos): pinos 01 e 17;
- alimentação de 5V (não pode ser aplicada aos seus outros pinos): pinos 02 e 04;
- GND: pinos 06, 09, 14, 20, 25, 30, 34 e 39;
- pinos para se utilizar com o protocolo de comunicação I²C (Inter-Integrated Circuit): pinos 03 e 05;
- portas seriais com protocolo RS-232: pino 08 como TX (transmissor) e pino 10 como RX (receptor);
- pinos para comunicação pelo protocolo *Full Duplex* (comunicação bidirecional): pinos 19, 21, 23, 24 e 26;
- portas de ID EEPROM: pinos 27 e 28;
- pinos GPIO, para entrada e saída de dados digitais: pinos 07, 11, 12, 13, 15, 16, 18, 22, 29, 31, 32, 33, 35, 36, 37, 38 e 40. Além destes, os pinos 03, 05, 08, 10, 19, 21, 23, 24 e 26 também podem ser programados para essa função.

3.6.3 Células de Energia

Outro componente importante para o desenvolvimento da plataforma robótica móvel é o seu sistema de alimentação. Para dimensioná-lo, foram consideradas algumas especificidades do movimento do robô nas aplicações para o qual está sendo projetado. Seu movimento é baseado na aquisição das fotografias, as quais devem ser obtidas com o sistema em repouso. Estimando que as imagens estarão deslocadas um metro entre si, o robô acelera até sua velocidade máxima nos primeiros 5cm (como estabelecido no dimensionamento do motor), enquanto nos últimos 5cm realiza o inverso. Nesses intervalos, os motores drenarão correntes próximas dos seus valores nominais de 1,1A. Na prática, essas correntes serão ligeiramente inferiores, visto que o torque do modelo escolhido é superior ao utilizado nos cálculos de aceleração. Entretanto, na trajetória de um metro, isso equivale apenas à 10% do tempo total. No intervalo restante, os motores estarão atuando com cargas menores, o que significa o consumo de 100mA para cada. Dentre os outros componentes com consumo energético relevante, cita-se a placa eletrônica, o regulador de tensão, o Raspberry Pi 3 e os servomotores. Suas correntes e potências calculadas são exibidas na Tabela 3.3.

Tabela 3.3: Consumo energético dos principais componentes.

Componente	Corrente (A)	Tensão (V)	Potência (W.h)
Motores com carga	2,2	11,1	2,442
Placa Eletrônica	0,25	5	1,25
Regulador de Tensão	0,25	11,1	2,775
Raspberry Pi 3	2	5	10
Total	4,7	-	16,467

A partir de sua análise, percebe-se que a corrente máxima drenada ocorre quando os motores estão acelerando o sistema robótico, chegando à $4,95A$, ocorrendo em cerca 10% do tempo de utilização do robô. Nos outros momentos, a corrente chega a cerca de $2,75A$. A potência máxima dissipada, já considerando-se o perfil de aceleração exposto, é de aproximadamente $20,815W.h$.

Baseando-se nestes parâmetros, foi determinado o modelo de bateria Gens Ace 5200 mAh, 15C, 3S LiPo apresentada na Figura 3.17. Como o próprio nome indica, apresenta $5200mAh$, sendo muito utilizada em quadricópteros, os quais trabalham com motores mais potentes do que os especificados nesse trabalho. Ela disponibiliza $11,1V$ na saída, com uma taxa de *discharge* de 15C, gerando um fornecimento de corrente máxima de $7,8A$ a partir de suas três células e uma potência de $57,7W.h$.



Figura 3.17: Modelo de bateria Gens Ace 5200 mAh, 15C, 3S LiPo (MERCADO LIVRE (2017a)).

3.7 Circuitos Eletrônicos

Esse projeto conta com diversos circuitos com funções específicas, para assim possibilitar a operação correta do robô. Esses circuitos serão apresentados e discutidos a seguir.

3.7.1 Regulador de Tensão

A bateria definida para alimentar os circuitos eletrônicos consiste de uma bateria LiPo de tensão nominal de 11,1V. Essa bateria pode ser utilizada diretamente para o acionamento dos motores, entretanto, para alimentação dos demais componentes eletrônicos e circuitos presente no robô se faz necessária a presença de uma fonte de alimentação de 5V. Para tal, é proposto o uso de um circuito regulador de tensão do tipo *buck*, o qual possibilita a regulação da tensão em 5V. Este circuito pode ser observado na Figura 3.18.

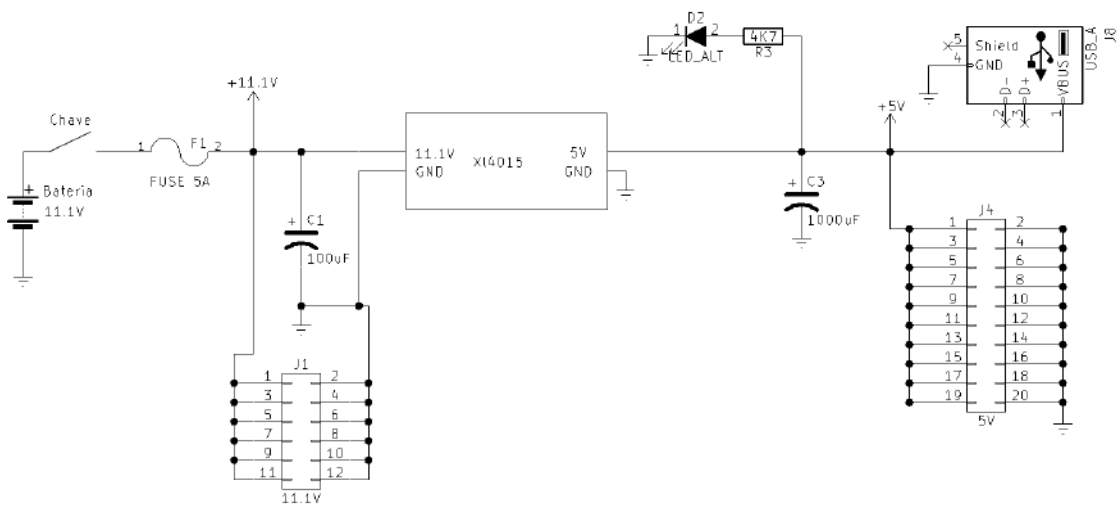


Figura 3.18: Circuito regulador de tensão proposto com o XL4015.

Conforme pode ser verificado no diagrama esquemático da Figura 3.18, o circuito *buck* é composto por componentes ativos e passivos comuns. Esse circuito apresenta como principal característica o circuito integrado XL4015, que possibilita obter uma tensão de saída de 1,25 a 35V e suporta uma corrente contínua máxima de 5A. Porém, ao avaliar o custo para confecção do circuito, verificou-se que esse valor é da ordem de R\$52,00, enquanto que o mesmo circuito, acrescido de voltímetro, pode ser adquirido no mercado por R\$46,00, o que justifica a compra da placa. Na Figura 3.19 é apresentado o circuito regulador de tensão empregado no projeto.

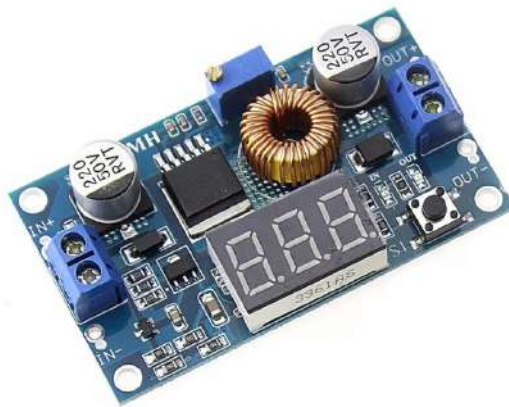


Figura 3.19: Regulador de tensão X14015 (MERCADO LIVRE (2017b)).

O módulo foi adaptado ao projeto, com auxílio do circuito de expansão dos pinos de tensão apresentado na Figura 3.20.

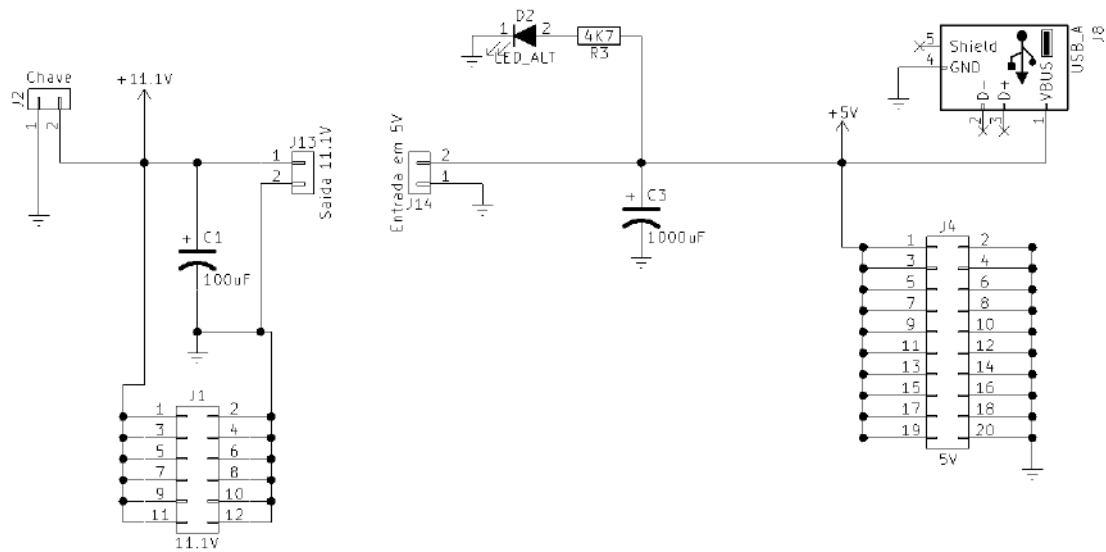


Figura 3.20: Esquemático do circuito com o regulador de tensão.

Conforme pode ser observado no diagrama da Figura 3.20, a bateria é conectada a placa via o borne de dois pinos, esse distribui a tensão de 11,1V em um barramento de pinos que distribuirá 11,1V entre o módulo regulador de tensão e a ponte H. Por sua vez, a saída do módulo regulador de tensão, 5V, retorna para a placa via um borne de duas vias, o qual disponibiliza em um barramento a tensão de alimentação de 5V, para os demais circuitos. No borne *Chave* são ligadas a bateria e uma chave gangorra. Em série com ambos deverá ser ligado um fusível de 5A para garantir a segurança de todos os circuitos que estarão sendo alimentados pela bateria.

3.7.2 Acelerômetro

O robô montado será alto e irá apresentar uma câmera acoplada no topo, o que pode representar um problema caso o robô se incline muito ao passar por pisos com inclinações acentuadas. Portanto, o ângulo do piso deve ser monitorado para que o sistema possa impedir que o robô se aproxime de situações inseguras, pois, caso o robô caia, a câmera pode ser danificada.

Uma solução para se mensurar a inclinação do piso é o acelerômetro, um dispositivo que serve para medir a aceleração de um corpo/objeto em relação à gravidade nos eixos X, Y e Z. De posse dos dados de cada um dos eixos, pode-se convertê-los no valor dos ângulos de inclinação do robô. As duas rotações que interessam para este projeto são os ângulos *Roll*, em relação a X, e *Pitch*, em relação a Y, que representam as inclinações do robô em relação ao piso. O sistema de ângulos de rotação *Roll-Pitch-Yaw* é apresentado na Figura 3.21

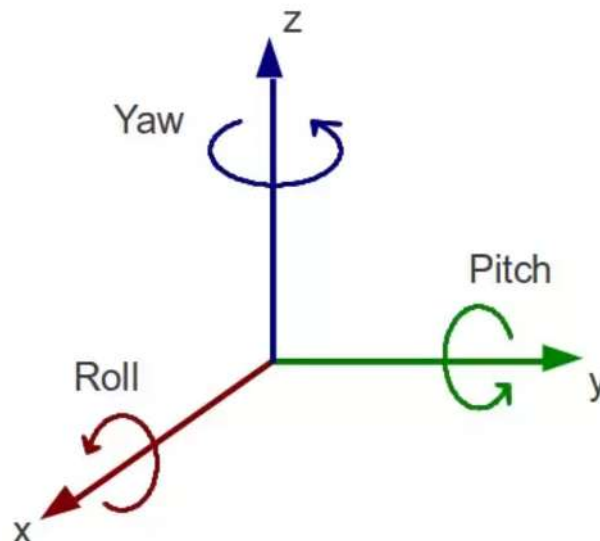


Figura 3.21: O sistema de ângulos de rotação *Roll-Pitch-Yaw* (QUORA (2015)).

Os ângulos *Roll* e *Pitch* são dados conforme as relações:

$$Roll = \text{Atan2}(Y, Z) * \frac{180^\circ}{\pi} \quad (3.6)$$

$$Pitch = \text{Atan2}(X, \sqrt{Y^2 + Z^2}) * \frac{180^\circ}{\pi} \quad (3.7)$$

Em que X, Y e Z são os valores do acelerômetro para cada um desses eixos. O acelerômetro escolhido foi o MMA7361, que tem uma grande sensibilidade e baixo consumo de energia. Ele é um acelerômetro capacitivo com saída analógica que, segundo sua desenvolvedora

NXP (2008), funciona de modo que a aceleração no dispositivo desloca uma placa móvel de um capacitor em relação a placas fixas no dispositivo. Assim, altera-se a capacitância de cada capacitor e são gerados valores de tensão que representam a aceleração sobre cada eixo. Um modelo físico simplificado do acelerômetro capacitivo é apresentado na Figura 3.22.

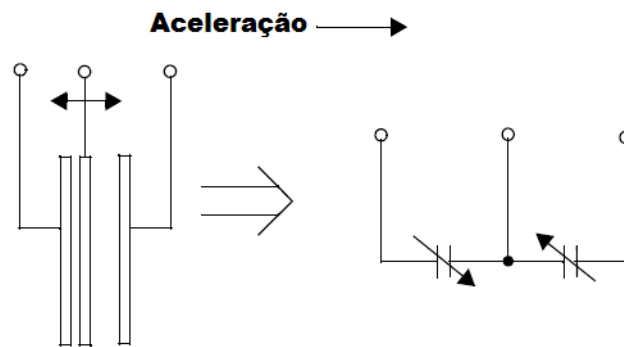


Figura 3.22: Modelo físico simplificado do acelerômetro capacitivo (NXP (2008)).

O acelerômetro escolhido conta com um micro capacitor responsável pela medição do sinal, contando com filtro para diminuição dos ruídos, além de possuir um circuito próprio para compensar a temperatura, gerando assim uma leitura confiável. O acelerômetro MMA7361 é exposto na Figura 3.23



Figura 3.23: Acelerômetro MMA7361 (ELETROGATE (2011)).

O Raspberry Pi é uma placa muito poderosa, mas apresenta um ponto fraco: ele só possui pinos digitais, não tendo nenhum conversor A/D, o que é necessário para se ler os dados do acelerômetro escolhido. Entretanto, esta deficiência pode ser sanada com diferentes soluções, em que três destas foram analisadas. A primeira é a utilização de um circuito com capacitores, nessa solução o sensor é utilizado como um resistor para

se carregar um capacitor, e assim, se mede o tempo gasto até que o capacitor esteja totalmente carregado, assim pode-se mensurar qual o valor de tensão está chegando até o capacitor. Esse método consegue aproximar de forma não muito precisa o valor obtido, e depende da contagem do Raspberry, não sendo a melhor forma de realizar essa tarefa. Além disso, esse método só funciona para sensores que funcionam como resistores, o que não é o caso do acelerômetro capacitivo (ADAFRUIT (2012)). O segundo método possível é através de um outro microcontrolador, usando o conversor A/D do mesmo e enviando os valores convertidos para o Raspberry Pi. O terceiro método é a montagem de um circuito conversor A/D externo, que apresenta leitura e conversão precisas. Existem circuitos integrados que têm essa função, como o MCP3008, escolhido para este projeto.

O MCP3008 envia dados através do protocolo de comunicação SPI (*Serial Peripheral Interface*), então esta será a forma como os dados analógicos do acelerômetro, convertidos para digital, serão transmitidos para o Raspberry Pi. O esquemático do circuito para a aplicação deste CI é apresentada na Figura 3.24

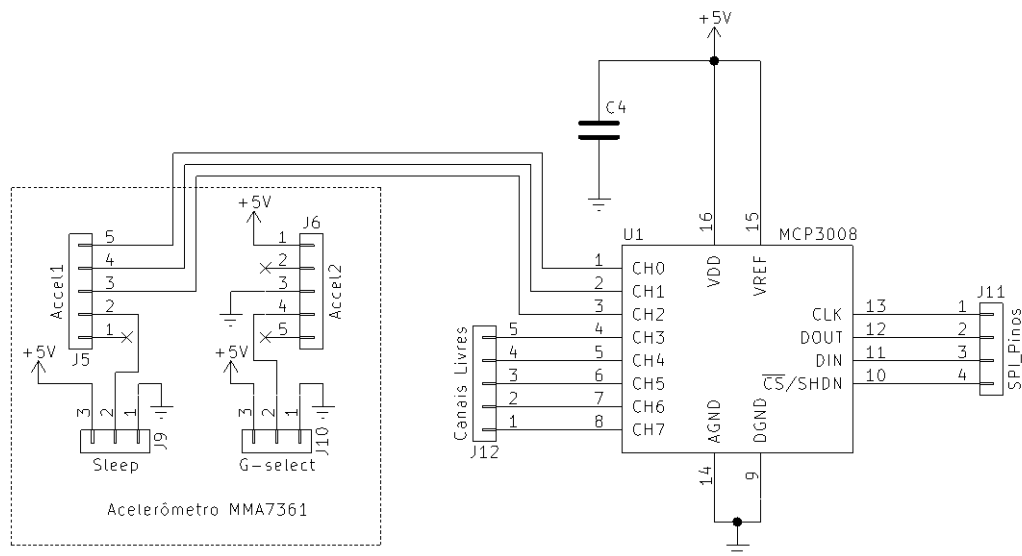


Figura 3.24: Esquemático do circuito com o conversor A/D MCP3008.

3.7.3 Comunicação I²C

Alguns circuitos presente no *hardware* do robô utilizam o protocolo de comunicação, I²C. Porém, deve-se observar que circuitos como o microcontrolador PIC16F876A utilizam o protocolo I²C com nível lógico TTL (do inglês *transistor transistor logic* – TTL), o que resulta em nível lógico 0 - 5V, o qual não pode ser conectado diretamente ao Raspberry Pi, uma vez que esse trabalha com nível lógico 0 - 3,3V. Como a comunicação I²C é

bidirecional, se faz necessária a conversão de tensão para ambos os valores de tensão utilizados. Assim, é proposto o circuito para conversão bidirecional de tensão na comunicação I^2C , apresentado pelo diagrama esquemático da Figura 3.25.

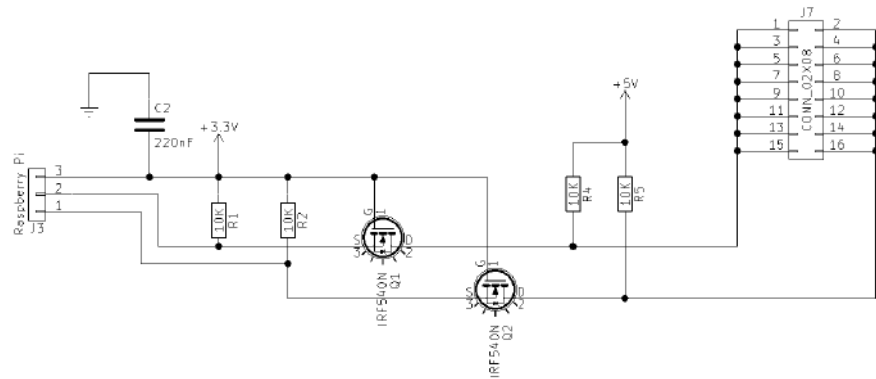


Figura 3.25: Circuito para a comunicação I^2C entre o Raspberry Pi e o PIC.

O método de funcionamento do circuito apresentado na Figura 3.25 é sugerido pela NXP (2007), desenvolvedora do protocolo I^2C . O I^2C apresenta dois barramentos, um serial de dados e um serial de clock. Cada um deles será ligado a um transistor MOSFET de canal N, que para esse projeto foram escolhidos os IRF540N, facilmente encontrados e que suportam até 100V e 33A. Os transistores devem ser ligados com o terminal *drain* no lado de maior tensão, no caso 5V, e o *source* no lado de menor tensão, no caso 3,3V, além dos dois MOSFETs terem seus *gates* ligados na menor tensão, de 3,3V.

Esse circuito apresenta resistores de *pull-up* em ambos os lados, garantindo suas tensões em 3,3V e 5V. Quando nenhum dispositivo em nenhum dos lados estiver abaixando para 0 o estado lógico do barramento, ambos os lados devem continuar em suas respectivas tensões. Como a tensão no *gate* é de 3,3V e o resistor de *pull-up* mantém a tensão no *source* também em 3,3V, a tensão entre *gate* e *source* V_{GS} fica abaixo da tensão de *threshold*, portanto o transistor não irá conduzir, dessa forma, ambos os lados se mantêm em nível lógico alto nessa condição.

Outra possibilidade é quando um dispositivo em um dos lados abaixar o nível lógico do barramento para 0. Caso o dispositivo esteja no lado de 3,3V, o valor no *source* do transistor será 0, e como a tensão no *gate* é de 3,3V, o valor de V_{GS} fica acima da tensão de *threshold* e o transistor começa a conduzir. Assim, o lado de 5V será levado para nível lógico baixo pelo dispositivo presente no lado de 3,3V através da condução do transistor MOSFET. Caso o dispositivo que esteja abaixando o nível lógico do barramento para 0 esteja do lado de 5V, o valor no *drain* do transistor será igual a zero. Como o transistor tem um diodo entre o *drain* e o *substrate* (que é ligado internamente com o *source*), ao se

3.7.5 Sensores Ultrassom

Para a detecção de obstáculos são utilizados seis sensores ultrassom HC-SR04, os quais são conectados diretamente ao Raspberry Pi. Entretanto, eles devem ser alimentados com 5V e dão uma resposta no terminal *echo* também em 5V, enquanto o Raspberry Pi trabalha com 3,3V. Contudo, todos os sensores podem utilizar o mesmo pino do Raspberry Pi para enviá-los o sinal de *trig*, já que não são todos utilizados simultaneamente. Para solucionar ambos os problemas é proposto o circuito apresentado pela Figura 3.27.

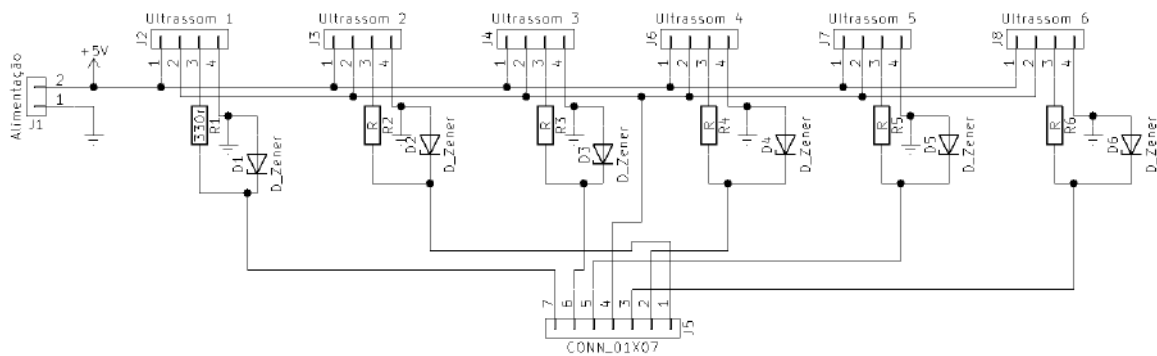


Figura 3.27: Circuito eletrônico para os sensores ultrassom.

Nesse circuito é colocado um diodo zener de 3,3V entre o GND e o terminal de *echo* de cada ultrassom. Assim, garante-se que o valor de resposta *echo* dos sensores se mantenha limitado ao valor de 3,3V, o que previne que o Raspberry Pi seja danificado. Adicionalmente, os sensores recebem a alimentação de 5V que necessitam para o seu funcionamento. O PCB projetado para esse circuito é apresentado pela Figura 3.28.

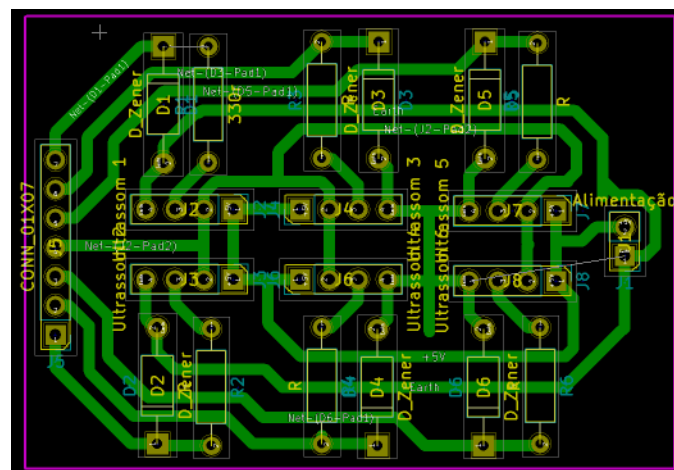


Figura 3.28: PCB do circuito eletrônico para os sensores ultrassom.

3.7.6 Placa para os Motores

Para simplificar os acoplamentos da alimentação aos motores do robô e o acesso para a leitura do sinal de seus *encoders*, foi projetada uma placa eletrônica com barras de pinos conforme apresentado no esquemático da Figura 3.29. Nessa placa entram os sinais de alimentação de cada motor provenientes da ponte H e a alimentação de 5V para os *encoders*. Também saem os sinais dos *encoders* dos dois motores, que são monitorados pelo Arduino.

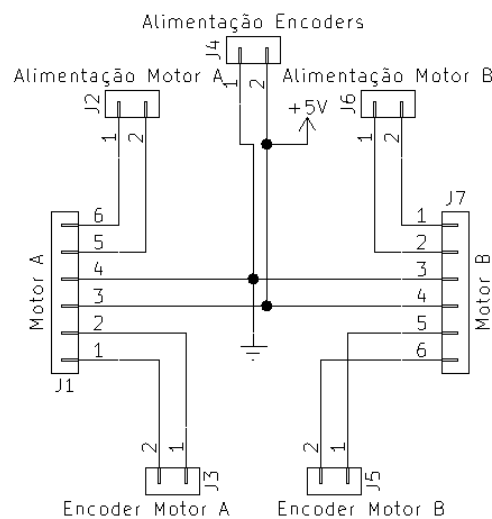


Figura 3.29: Circuito eletrônico para acoplamento dos cabos dos motores.

O PCB projetado para esse circuito é apresentado pela Figura 3.30.

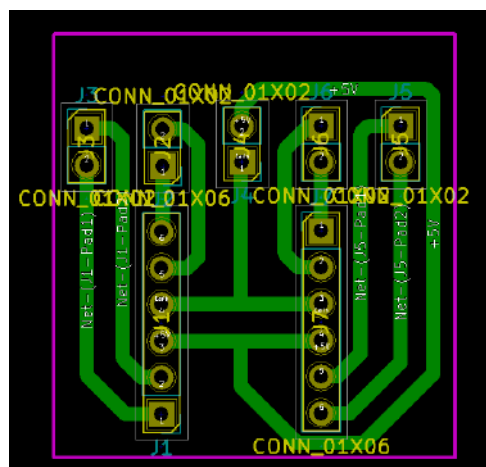


Figura 3.30: PCB do circuito eletrônico para acoplamento dos cabos dos motores.

3.8 Programação

Para a parte de programação do projeto foram utilizadas seis linguagens de: HTML, CSS e JavaScript para a página Web; Shell Script para os *scripts* CGI; Python para os programas do Raspberry Pi e C para o Arduino. Cada uma das partes da programação implementada para o robô são apresentadas nessa seção.

3.8.1 Servidor Web

Para a operação do robô, foi implementada uma página web disponibilizada através de um servidor web. Essa estratégia apresenta como vantagem a eliminação da necessidade de um controle físico. Assim, os recursos que seriam exigidos para comprá-lo ou montá-lo são economizados, além da dificuldade de se implementar novas opções e funcionalidades posteriormente. Já um aplicativo nativo para dispositivos móveis ou computadores apresenta como desvantagens a necessidade de se instalar esse programa na plataforma desejada. Essa prática consome memória, além da falta de flexibilidade entre diferentes dispositivos para utilizá-lo, sendo exigido o desenvolvimento de um programa para cada sistema operacional. O servidor web, em contrapartida, não apresenta tais problemas, graças a sua total flexibilidade para ser alterado, incluindo novas funcionalidades além da redução de custos relacionados à *hardware*. Além disso, ele é acessado através de qualquer navegador de internet ou sistema operacional que o piloto utilize na plataforma de sua escolha, seja um *notebook*, *tablet* ou *smartphone*.

A utilização da plataforma web funciona da seguinte forma: independente de qual plataforma o operador escolha, ele deve criar uma rede Wi-Fi através de seu dispositivo, onde o robô irá se conectar através do Wi-Fi do Raspberry Pi. Portanto, depois de conectado pela primeira vez e sendo salva a senha da rede Wi-Fi, toda vez que essa rede for habilitada, a conexão será realizada automaticamente. Posteriormente, o usuário deve abrir o navegador de internet de sua escolha e digitar o endereço IP do robô, assim, o servidor web instalado no Raspberry Pi disponibiliza a página web de controle do robô.

O servidor web escolhido foi o Lighttpd, um servidor grátis muito leve e simples, tendo o desempenho como sua maior vantagem. Essa escolha em detrimento à de concorrentes mais completos é que, além de gastar menos memória e processamento, módulos mais complexos não são necessários para os objetivos desse projeto, assim, o Lighttpd é uma opção satisfatória.

3.8.2 Página Web

A página web disponibilizada para o operador através do servidor web foi programada por meio do *software* multiplataforma de edição de texto Sublime Text. A escolha desse

foi feita por permitir se programar em diversas linguagens de programação.

Para o desenvolvimento da página web foram utilizadas três linguagens de programação: HTML, CSS e JavaScript. O HTML é usado para dividir a organização do conteúdo da página, assim, pode-se estruturá-la em diferentes blocos com funções distintas. Portanto, a HTML divide toda a página web, mas isso só gera uma página cheia de retângulos. Portanto, torna-se necessária uma linguagem de programação que permite a implementação de um *design* para as divisões criadas. Então, o CSS foi definido para essa função, por ser uma linguagem muito popular em tais aplicações. Com o CSS, é possível tornar a página mais visual, amigável e intuitiva. Com essas duas linguagens obtém-se uma página visualmente satisfatória e organizada, mas estática. Ou seja, não é possível controlar ou alterar o comportamento dos componentes dessa interface. Portanto, utiliza-se o JavaScript para manipular o comportamento da página web que foi desenvolvida, permitindo por exemplo, a definição de diferentes funções para cada um dos botões presentes na página.

Como a operação do robô pode ser realizada através de diferentes dispositivos, foram desenvolvidas duas páginas web: uma para dispositivos *touchscreen*, como *tablets* e *smartphones*, e outra para os dispositivos que utilizam um cursor por *touchpad* ou *mouse*, como os *notebooks*. Essas duas páginas podem ser alternadas através do canto inferior esquerdo de cada uma delas. Elas foram criadas visando a otimização da utilização em cada um dos dispositivos, apresentando funções próprias para *touchscreen* e *mouse*, além de se desenvolver um *design* mais específico para cada um dos tipos de dispositivos.

Na parte superior da página web é possível observar uma caixa em que é exposta a imagem proveniente da câmera do Raspberry Pi em tempo real, para que o operador possa identificar o ambiente onde o robô está se locomovendo. A orientação desta câmera poderá ser variada através dos botões presentes na caixa *Controle da Câmera*, que atuam sobre os servo motores do mecanismo de *pan-tilt*. Os controles de movimento do robô e controle de velocidade do mesmo, são realizados através de ícones colocados na página, com o intuito de gerar um ambiente de operação amigável e intuitivo, focando na experiência do usuário.

3.8.3 CGI e XMLHttpRequest

O funcionamento desses botões é viabilizado através da utilização da API XMLHttpRequest e arquivos CGI, que realizam a interface entre a página web e os outros programas presentes no Raspberry Pi. Quando um botão é pressionado na página web, uma parte de código escrito em JavaScript irá vincular essa operação com o acionamento de um arquivo CGI referente a aquele botão. Um exemplo de código que faz essa função é apresentado pela Figura 3.31.

```
var xmlhttp;
xmlhttp=new XMLHttpRequest();

function forward()
{
    xmlhttp.open("GET","cgi-bin/forward.cgi",true);
    xmlhttp.send();
}
```

Figura 3.31: Função que pode ser vinculada a um botão para abrir um arquivo CGI.

Os arquivos CGI podem ser implementados em diversas linguagens diferentes, mas para esse projeto foi escolhida a linguagem de Shell Script, a mesma utilizada no terminal Linux. O controle dos servo motores, por exemplo, é realizado diretamente pelo Arduino, portanto, o CGI envia, através do I²C do Raspberry Pi, um número hexadecimal para o Arduino e, esse número representa uma determinada operação que é interpretada pelo código presente no microcontrolador. Assim, cada botão de controle da orientação da câmera envia um número hexadecimal diferente para o Arduino, que irá movimentar os servo motores de acordo com a programação desenvolvida. A escolha da velocidade do robô também é feita diretamente no Arduino, sendo enviados números hexadecimais também interpretados pelo Arduino. Desse modo, o valor da variáveis que especifica a velocidade dos motores é alterada entre as três opções predefinidas (mínima, média e máxima) dentro desse microcontrolador.

Já a movimentação do robô não é feita diretamente pelo Arduino, sendo necessária a tomada de decisões dependentes da leitura dos sensores do robô. Essa tarefa é realizada pelo Raspberry Pi, no qual os sensores ultrassom e o acelerômetro estão ligados. Portanto, a função dos arquivos CGI, referentes à movimentação do robô, consiste na abertura de um programa Python para cada direção. Quando um botão de direção do robô é pressionado, sua função correspondente em JavaScript abre o arquivo CGI vinculado à mesma que por sua vez executa o código Python referente a aquele movimento. Entretanto, quando o botão é solto, a função *stop* é aplicada, na qual seu arquivo CGI apresenta duas finalidades: primeiramente ele coloca o pino GPIO14 do Raspberry Pi em estado alto, o que finaliza o *loop* de repetição *while* presente nos códigos de movimento, além de enviar o número hexadecimal 0X00, que representa o desligamento de ambos os motores, para o Arduino através do I²C. Assim, é garantido que nenhum código de movimento estará enviando algum número I²C para ligar os motores e, então, é enviada a ordem para o Arduino desligar os motores com segurança e sem erros.

3.8.4 Movimentação do Robô

Código Python dos Sensores

Para dar início a determinado movimento do robô, é necessário primeiramente avaliar os sensores presentes no mesmo. Caso o robô esteja próximo de uma inclinação crítica, a operação do mesmo não será permitida e o movimento não poderá ser realizado. Além disso, deverá ser avaliada a possibilidade de uma colisão do robô, sendo analisada a distância em relação a obstáculos através dos sensores ultrassom: dois para frente, dois para trás e um para cada lado. Após a avaliação dos sensores, caso um movimento possa ser realizado, o mesmo será iniciado. Entretanto, durante determinado movimento, o código ainda deverá continuar a avaliar os sensores continuamente, para garantir a segurança da operação do robô: impedindo colisões com obstáculos ou sua queda em caso de uma inclinação muito grande do piso.

Para todos os quatro movimentos possíveis do robô deverão ser utilizados os sensores ultrassom e o acelerômetro, portanto, a implementação de métodos, em todos os códigos, para o tratamento do sinal proveniente desses sensores não é a melhor opção. Logo, foi criado um código intitulado *sensores*, que apresenta os métodos referentes aos sensores presentes no robô. Dessa forma, esse código será importado pelos programas de movimentação do robô e, quando necessário, os métodos poderão ser utilizados de acordo com a necessidade de cada movimento do robô. O código dos sensores encontra-se disponível no Apêndice A.1.

Códigos Python dos Movimentos

Foram desenvolvidos quatro códigos diferentes para se realizar a movimentação do robô: locomover para frente, rotacionar no sentido anti-horário, sentido horário e movimentar para trás. Estes são muito parecidos, alterando-se os valores I²C enviados por cada um, referentes ao comando de sentido de movimento, além de se diferir os valores utilizados como parâmetros para as funções referentes aos sensores. Por exemplo, cada tipo de movimento irá utilizar diferentes sensores ultrassom presentes no robô: o movimento para frente utilizará os sensores 1 e 2, presentes na parte frontal do robô, enquanto o movimento no sentido anti-horário utilizará o sensor 3, presente na lateral esquerda do robô.

Para a movimentação do robô deveria ser considerado o seguinte problema: os sensores devem ser avaliados antes de iniciar o movimento do robô, caso não exista nenhum perigo de colisão, o mesmo poderá iniciar o movimento. Entretanto, mesmo após o início da movimentação, os sensores ainda deveriam ser avaliados constantemente, pois, o robô poderia ser colocado novamente em uma situação de colisão ou, no caso da perda de comunicação, o mesmo poderia continuar o movimento e colidir com algum obstáculo,

sem a possibilidade do operador evitar. Assim, é criado um *loop while*, onde os sensores serão constantemente avaliados e, caso exista uma condição insegura, o código envia para o microcontrolador a ordem de parar o robô e finaliza o *script* Python. Entretanto, caso o robô não encontre uma situação que gere essa operação, o *loop while* não irá terminar e esse código Python não será fechado. A solução encontrada para esse problema foi a adoção de dois pinos do Raspberry Pi para indicar ao robô quando o comando de parada ou algum outro código de movimento tiver sido iniciado. Assim, quando um programa começar, ele levará o pino GPIO14 para nível alto, que poderá ser lido pelo pino GPIO15, que é configurado como entrada, e o Raspberry terá acesso a essa informação através de qualquer programa. Dessa forma, foi colocada a condição para o laço *while* funcionar apenas enquanto o pino GPIO15 estiver lendo o valor 0, saindo do laço *while* ao ser lido o valor 1 e, conseqüentemente, do código Python, o que acontecerá ao se iniciar um novo *script*. O código para o movimento para a frente encontra-se disponível no Apêndice A.2.

3.8.5 Código do Microcontrolador

O microcontrolador apresenta três funções dentro do robô: manipular os servo motores do mecanismo pan-tilt da câmera, movimentar o robô através da atuação dos motores cc e realizar a leitura dos *encoders*. Para realizar essas tarefas ele recebe tais comandos do Raspberry Pi através do I²C. Os comandos recebidos estarão na forma de números hexadecimais que o microcontrolador interpretará através de um *switch case*. O código implementado no Arduino encontra-se disponível no Apêndice A.3.

Atuação dos Motores

Para a atuação dos motores são empregados os números: 0x00, 0x01, 0x02, 0x03 e 0x04, que representam respectivamente: parar os motores, movimentar o robô para frente, rotacioná-lo no sentido anti-horário, sentido horário e movimentá-lo para trás. Quando o comando de parar os motores é dado ao microcontrolador, a velocidade de ambos os motores é alterada para zero. Já no caso dos comandos de movimentação serem empregados, os motores são atuados nas direções desejadas do movimento e com a velocidade vigente. O valor da variável velocidade, que é aplicado aos motores, é variado entre as três opções de velocidade que são pré-estabelecidas na programação. Esse valor é iniciado com o valor de velocidade média, mas é alterado de acordo com o comando dado pelo usuário na página web de controle do robô. Os valores dos números I²C enviados pelo Raspberry Pi para essa operação são: 0x10, 0x11 e 0x12, que representam respectivamente as velocidades: mínima, média e máxima.

Leitura dos Encoders

A leitura dos *encoders* é realizada através da interrupção gerada pelo canal A de cada um dos motores. Para tanto, são utilizadas as rotinas de interrupção *HandleLeftMotorInterruptA* e *HandleRightMotorInterruptA*, referentes a cada um dos motores. Dentro dessas rotinas, o canal B do motor é também analisado, para assim se descobrir em qual das direções os motores estão girando. A interrupção é gerada em sinal alto do canal A do motor, portanto, caso o canal B também esteja em nível alto, sabe-se que o motor está no sentido anti-horário. Já no caso do canal B estar em nível baixo, o motor está girando no sentido horário. Assim, as variáveis *_LeftEncoderTicks* e *_RightEncoderTicks*, referentes aos *encoders* dos dois motores, são incrementados ou decrementados de acordo com seu sentido de rotação e, conseqüentemente, a direção em que as rodas do robô estão se movimentando.

Manipulação do Mecanismo de Pan-Tilt

Os números 0x20, 0x21, 0x22, 0x23, 0x24 e 0x25 são referentes à manipulação da posição dos servo motores do mecanismo de pan-tilt, sendo os três primeiros referentes ao movimento horizontal e os outros em relação ao movimento vertical. Ambos os servos têm o código que representa a sua posição central e outros dois que incrementam sua posição em cada um dos sentidos. Em *software*, através do comando condicional *if*, a posição dos servo motores é limitada a certos valores que são os extremos conseqüentes da própria construção física do mecanismo, de forma a não forçá-los.

Resultados

Este capítulo relata os resultados obtidos durante o desenvolvimento do trabalho, retomando os objetivos específicos estabelecidos anteriormente. Em cada seção, são explanadas as principais conclusões sobre o respectivo tema, levando às conclusões gerais sobre o trabalho no próximo capítulo.

4.1 Topologia do robô

Conforme determinado nos objetivos específicos do trabalho e na fundamentação teórica acerca desse tema, a definição da topologia do robô é uma das etapas iniciais no projeto de um robô. Vários componentes são definidos a partir da topologia estabelecida, a qual é intimamente relacionada aos requisitos de projetos levantados. Para garantir que o robô se desloque em ambiente com obstáculos, por exemplo mobílias, foi adotada a topologia descrita no Capítulo 3, a qual apresenta duas rodas tracionadas de forma diferencial na parte traseira e um único rodízio na dianteira. Os graus de liberdade proporcionados pela rotação em torno do eixo entre as duas rodas traseiras, e as translações em x e y , atendem no quesito estabilidade e manobrabilidade para atuação nos ambientes propostos.

4.2 Plataforma Mecânica

Para se ter uma melhor visualização preliminar do robô foi desenvolvida em um *software* de modelagem 3D a montagem apresentada na Figura 4.1.

O raio das rodas é de $50mm$ e pelo acoplamento do motor, tem-se $14,5mm$ de distância entre o centro do eixo do motor até a parte inferior da placa de acrílico. Cada uma das placas apresentam $6mm$ de espessura e estão a $100mm$ de distância uma da outra, sendo ligadas por quatro barras roscadas de $1/4$ polegadas. Têm-se, portanto, $176,5mm$ da superfície até a face superior da placa de acrílico. A distância entre o terreno e a lente

da câmera é de 1550mm , definido como sendo a altura média dos olhos dos brasileiros conforme apresentado e discutido na Seção 3.3.

Esse se comporta como o principal requisito de projeto do trabalho, haja vista que o robô apresenta uma aplicação específica: a captura de imagens para o desenvolvimento de visitas virtuais com foco na imersão do usuário e simulação da realidade. Objetivando a sensação de o indivíduo estar fisicamente no local, é imprescindível que esse valor seja seguido, portanto, o desenvolvimento do robô deve ser realizado com o intuito de viabilizá-lo.



Figura 4.1: Protótipo virtual do robô proposto

A altura do robô é um dos desafios do projeto, o que poderia causar problemas em relação à estabilidade do mesmo. Portanto, foram consideradas algumas alternativas para diminuir a sensibilidade do sistema robótico à inclinações. Dentre elas, destacam-se o aumento das dimensões das placas estruturais e a alteração do material utilizado em sua construção. A primeira opção foi descartada devido à necessidade de um robô portátil para ser transportado em motocicletas até os locais de mapeamento dos imóveis. Assim, as placas deveriam ser mantidas com 400mm de lado. Já o segundo método se mostrou hipoteticamente promissor, o que levou a um estudo do comportamento do centro de massa do sistema ao utilizar-se liga de alumínio 1060 ou o acrílico de alto impacto como materiais constituintes das plataformas.

Utilizando um *software* de simulação mecânica, essas propriedades foram alteradas e, através de um recurso de análise do centro de massa, foram obtidas as informações exibidas nas Figuras 4.2 e 4.3.

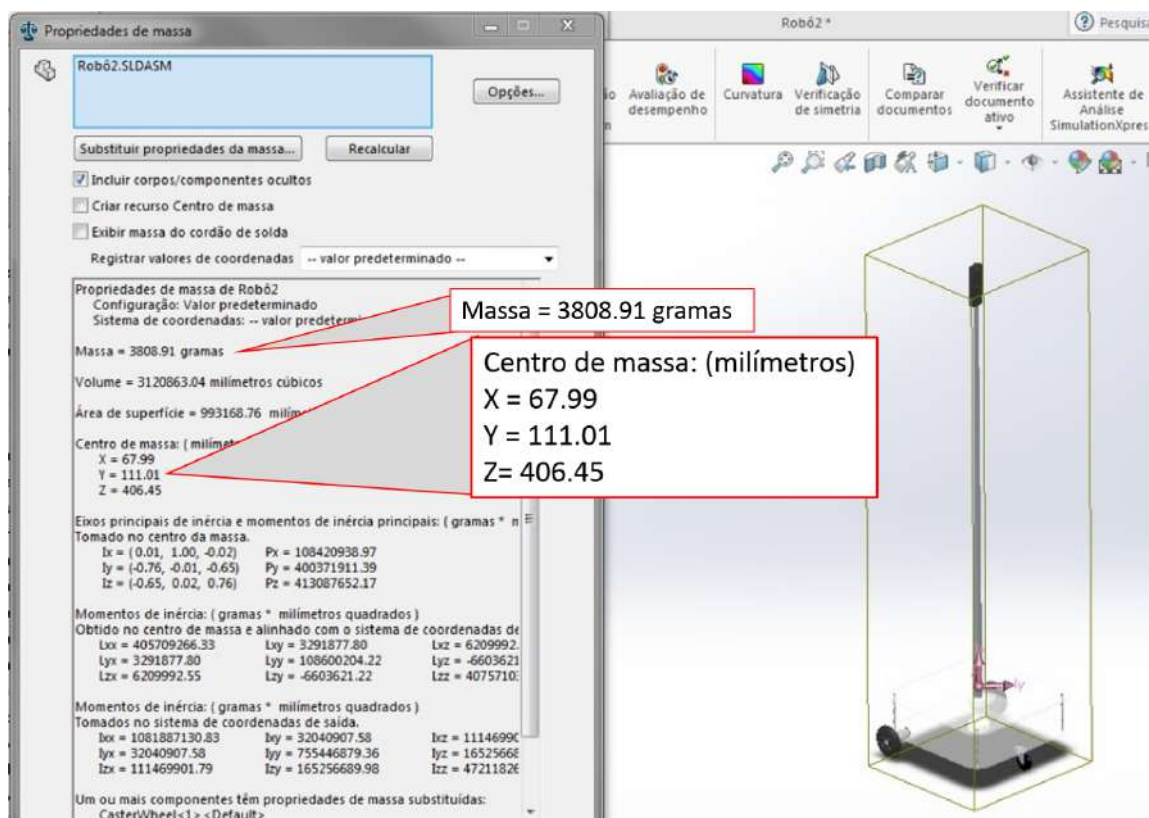


Figura 4.2: Análise do centro de massa do sistema robótico com placas de acrílico de alto impacto.

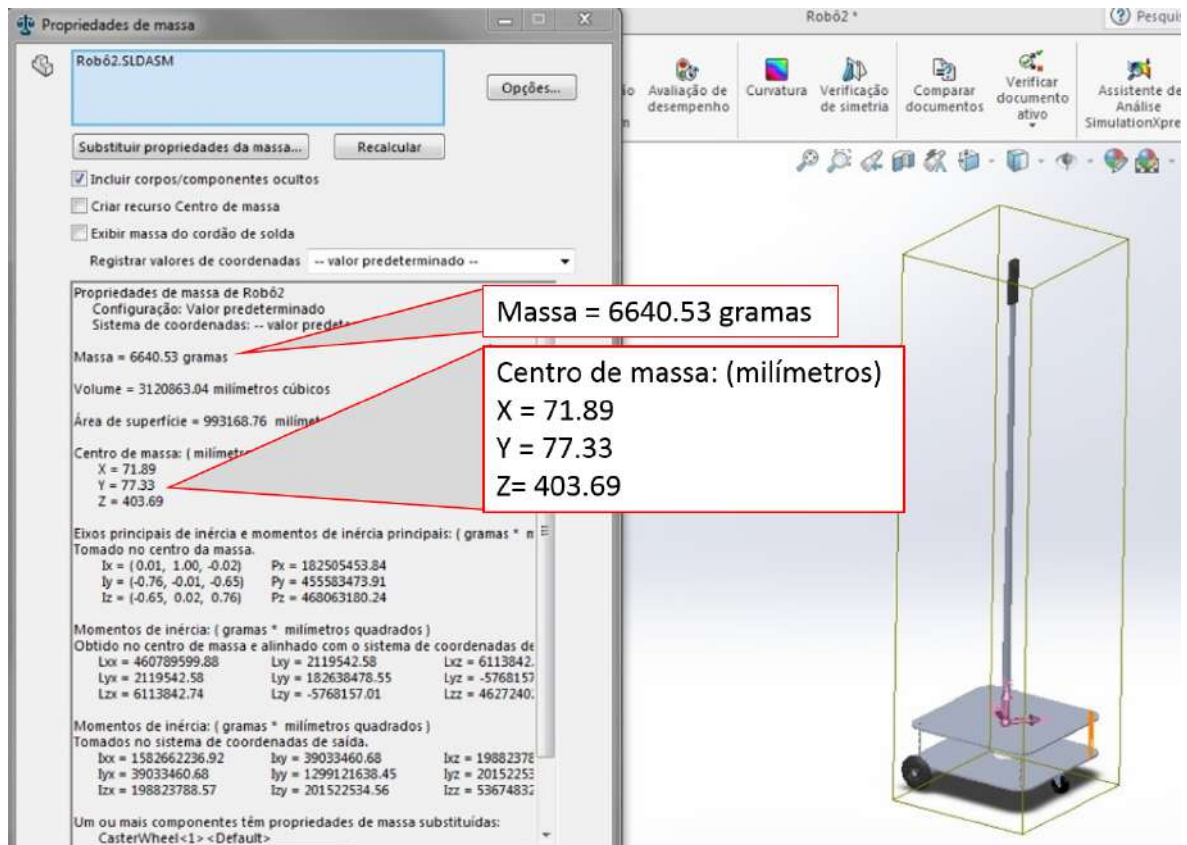


Figura 4.3: Análise do centro de massa do sistema robótico com placas de alumínio de alto impacto.

É perceptível a alteração das coordenadas do centro de massa ao longo do eixo Y para os dois casos. Ele indica a direção vertical, logo, houve uma aproximação de 33.68mm do centro de massa em relação à superfície de deslocamento do robô. Essa mudança é positiva, visto que um centro de massa mais baixo atende aos requisitos de melhor estabilidade desejados. Em contrapartida, ocorreu um aumento de $2,832\text{Kg}$ na massa do robô, o que opõe a necessidade de portabilidade já mencionada. Portanto, foi realizada uma análise crítica, ponderando as duas alternativas, e foi concluído que um aumento de massa tão grande não justifica a pequena redução de altura do centro de massa. A perda de portabilidade supera o ganho de estabilidade proporcionado pela utilização do alumínio, levando à confecção das placas em acrílico de alto impacto. O aspecto final desses componentes é exibido na Figura 4.4.

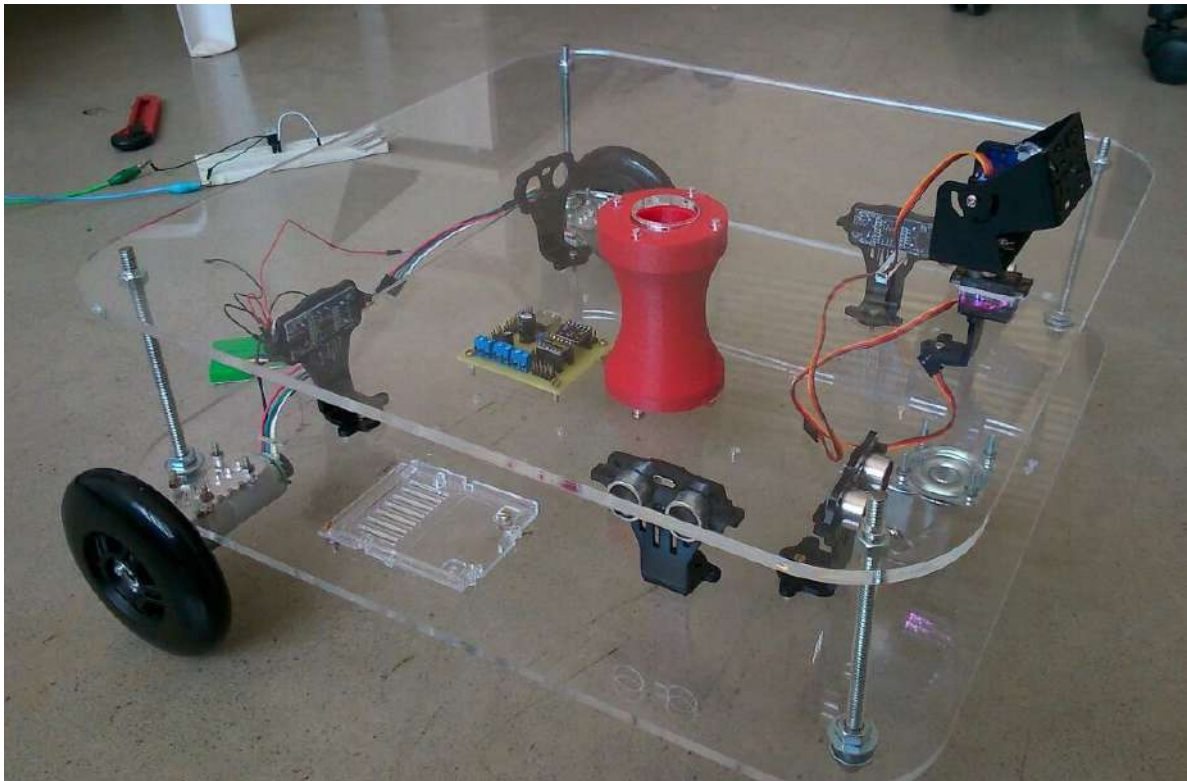


Figura 4.4: Placas de acrílico cortadas e fixadas no sistema robótico.

4.3 Sensores

4.3.1 Câmera 360°

Um dos principais pontos observados durante as entrevistas de *Customer Discovery* e os testes de produto mínimo viável, foi a importância da qualidade das imagens para garantir a aceitação do público-alvo em relação ao *tour* virtual. Consultando um dos desenvolvedores da *startup* de realidade virtual, VR Monkey, foi descoberto que o parâmetro de qualidade de imagem muda muito ao trabalhar-se com imagens esféricas. Segundo ele, a resolução deve ser consideravelmente maior do que o convencional, sendo uma imagem com resolução em torno de 25K o ideal para tais aplicações. Entretanto, câmeras com essa resolução não são facilmente encontradas no mercado. Conforme descrito na Seção 3.5.1, a Panono 360° é uma câmera que captura imagens em 16K, entretanto, apresenta um custo financeiro elevado. Outra opção é a Insta360 Pro, que captura imagens profissionais em 8K pelo valor de U\$ 3499. Ambas as câmeras apresentam qualidades ideais, porém, o valor se mantém inacessível. Logo, visando um equilíbrio entre a resolução das fotografias e acessibilidade, concluiu-se que a Insta360 4K é a melhor escolha no que se refere a custo benefício. Suas especificações demonstram que esta câmera é superior à Ricoh Theta S, a outra alternativa viável que, entretanto, não proporcionaria imagens tão boas quanto as

da anterior.

Todavia, durante o processo de aquisição dos componentes para montagem, a Insta360 4K foi descontinuada pela sua fabricante. Apesar de ainda existirem exemplares disponíveis naquele momento, o fato de não ser mais produzida tornou sua compra menos atrativa. Além dos fatores óbvios como a redução da oferta de peças repositórias e assistência técnica, os autores chegaram à conclusão de que não seria viável adquirir um equipamento que não poderia ser replicado em um momento futuro, ao escalar o negócio baseado no sistema robótico desenvolvido. Ou seja, o produto final não contaria com a mesma câmera 360° do protótipo, significando que a integração do hardware deveria ser refeita. Portanto, a Ricoh Theta S se apresentou como a opção mais atrativa, que mesmo apresentando uma qualidade de imagem inferior, é suficiente para a maioria dos segmentos de clientes prospectados.

Uma vez que a câmera foi adquirida, fatores positivos relacionados à usabilidade foram notados. O aplicativo Android de controle à distância do dispositivo é muito simples de se usar, e a plataforma Ricoh permite edições e transmissões das fotografias de forma muito eficaz. As fotos são armazenadas na memória interna de 8Gb da própria câmera, podendo também serem salvas no *smartphone* utilizado para controlá-la. A velocidade de processamento das imagens capturadas é muito alta, cerca de 5 segundos para configurações comuns e de aproximadamente 10 segundos para imagens em HDR.

4.3.2 *Encoders*

Para a leitura do movimento das rodas e utilização dessa para a odometria do robô, serão utilizados *encoders* de quadratura. O acoplamento dos *encoders* no eixo é uma tarefa complicada, pois, o acoplamento deve ser bem feito, sem folgas ou possibilidade de escorregar, o que poderia gerar inconsistências na leitura dos mesmos. A solução encontrada foi a escolha de motores que apresentam *encoders* de quadratura acoplados internamente no eixo do motor. Dessa forma se obtém uma leitura mais confiável, além de uma maior precisão, já que estes são ligados antes da caixa de redução e sua leitura da saída é, portanto, multiplicada pelo fator de redução. O motor escolhido na Seção 3.2.2 apresenta uma caixa de redução de 172:1, e como o encoder de quadratura acoplado ao eixo do motor tem 48 contagens por revolução, são obtidas em conjunto com a caixa de redução 8256 contagens do encoder por revolução de cada roda do motor.

Os *encoders* serão lidos pelo microcontrolador e estes valores serão transmitidos através de I²C para o Raspberry Pi. Através da leitura desses sensores pode-se realizar a odometria do robô, mensurando-se a sua posição dentro do ambiente.

4.3.3 Ultrassom

Conforme descrito na etapa de projeto, são necessários seis sensores ultrassom para proporcionar um sistema de prevenção de colisões eficaz. Assim, ele será capaz de auxiliar o operador durante sua navegação agindo em conjunto com as duas câmeras, garantindo que a parte inferior do robô se mantenha protegida. Portanto, para garantir o posicionamento coerente com o especificado, foram utilizados suportes especificamente projetados para os sensores adquiridos. Eles possuem uma base para serem fixados, a qual foi usada para prendê-los à plataforma inferior do robô. A montagem final desse conjunto e a disposição dos sensores conforme projetado são exibidas nas Figuras 4.5 e 4.6.



Figura 4.5: Sensor ultrassom montado em seu suporte fixado à base do robô.



Figura 4.6: Vista superior da base do robô evidenciando a configuração espacial dos sensores ultrassom.

4.4 Motores

O dimensionamento do motor descrito na Seção 3.2.2 traz os principais parâmetros na definição do modelo deste componente. A opção de adotar-se uma redução maior proporciona benefícios para o prosseguimento do projeto, como uma maior insensibilidade às diferenças inerentes à construção dos motores que podem gerar rotações diferentes quando submetidas a um mesmo sinal, resultando em uma cinemática indesejada. No caso da topologia adotada, tais diferenças impediriam o robô de seguir uma trajetória retilínea. Também cita-se o aumento da leitura do *encoder* por rotação, o que proporciona uma maior precisão para manusear-se o sistema robótico. A disponibilidade comercial também foi um fator influente, outros modelos similares não encontravam-se disponíveis no fornecedor definido no processo de compra junto ao AWC.

Após a aquisição dos motores, bem como de seus suportes e fixação, foi possível montar o mecanismo de acoplamento das rodas, conforme ilustrado previamente na Figura 3.2. Foram utilizadas as entradas laterais do suporte para fixar o conjunto à placa de acrílico, garantindo o alinhamento horizontal do eixo do motor. Então, as rodas foram acopladas através dos adaptadores e o mecanismo resultante é exibido nas Figuras 4.7, 4.8 e 4.9.

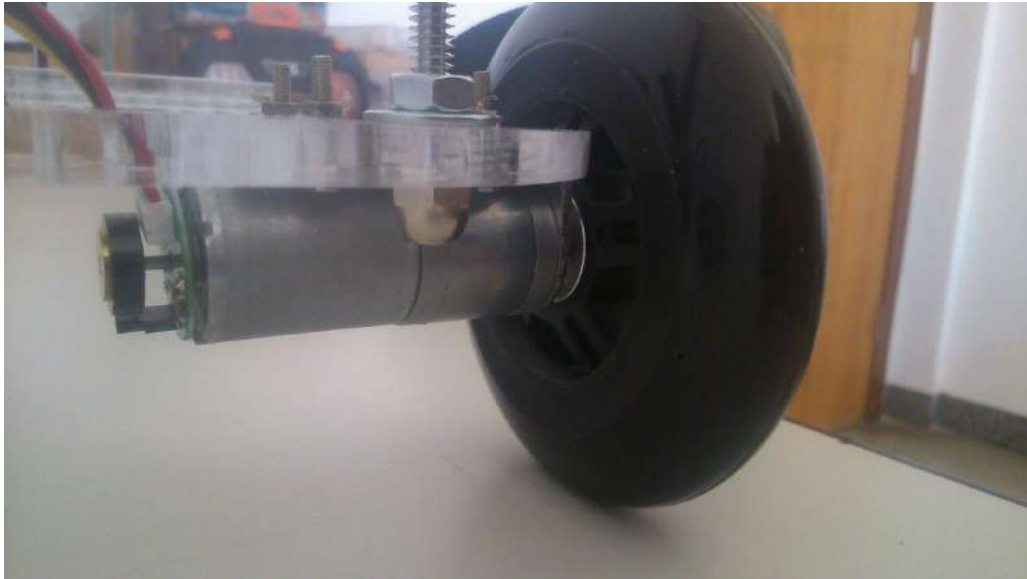


Figura 4.7: Montagem final do mecanismo de acoplamento do motor à roda.



Figura 4.8: Fixação do motor e seu suporte à plataforma inferior do robô.

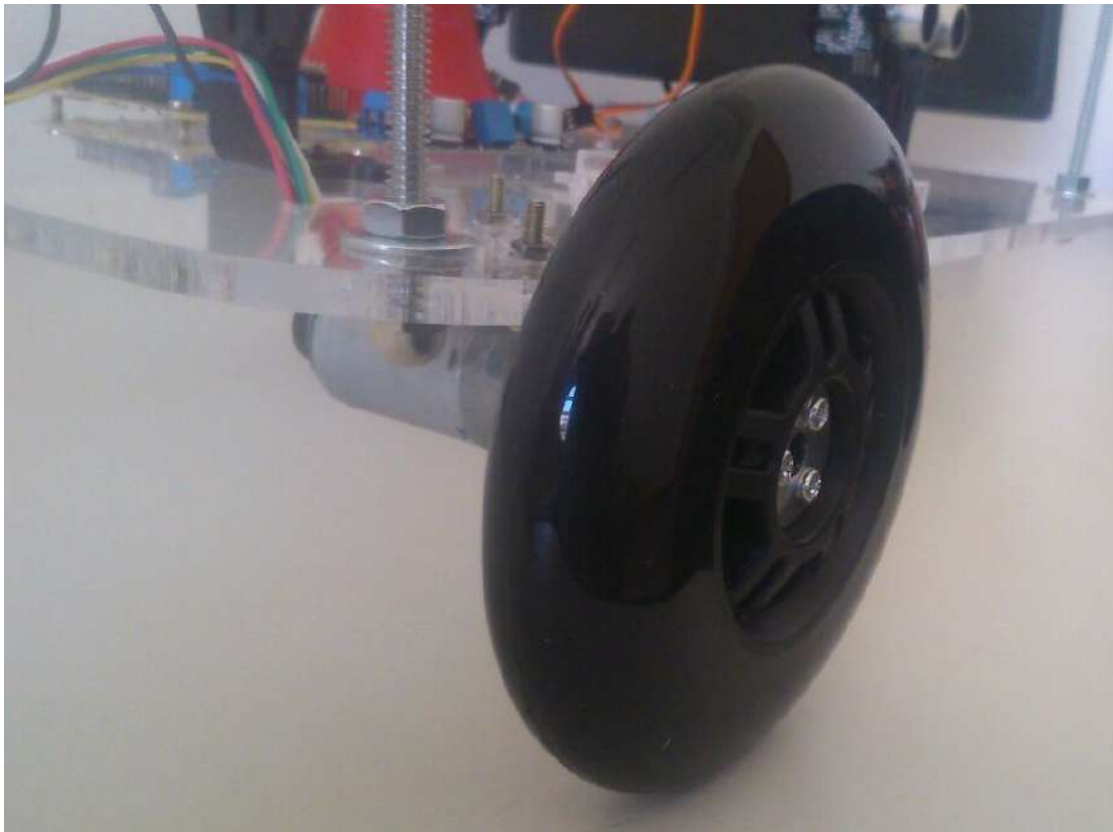


Figura 4.9: Montagem final da roda acoplada ao motor através de seus adaptadores.

4.4.1 Controle dos Motores

A aplicação de controle nos motores tem como objetivo a eliminação de erros na trajetória desejada para o sistema robótico. Caso os motores desempenhem de forma distinta, as diferenças de velocidade pode resultar em uma trajetória curvilínea, quando na verdade, são desejados movimentos retos. Entretanto, uma vez que o robô foi montado e os motores passaram a trabalhar sob carga, as pequenas diferenças existentes à vazio foram minimizadas. Desse modo, foram observadas trajetórias condizentes com o desejado, fato comprovado pelas leituras similares dos *encoders* exibidas na Figura 4.10. O ensaio foi realizado com três movimentos independentes, nos quais cada leitura parte de zero, não referindo-se ao valor anterior. Portanto, quanto mais próxima a leitura dos *encoders* da direita e esquerda em cada caso, menor o desvio da trajetória desejada. Além disso, o fato de o robô ser remotamente controlado permite que o próprio operador conserte os desvios quando o mesmo perceber que prejudicarão a aquisição das imagens.

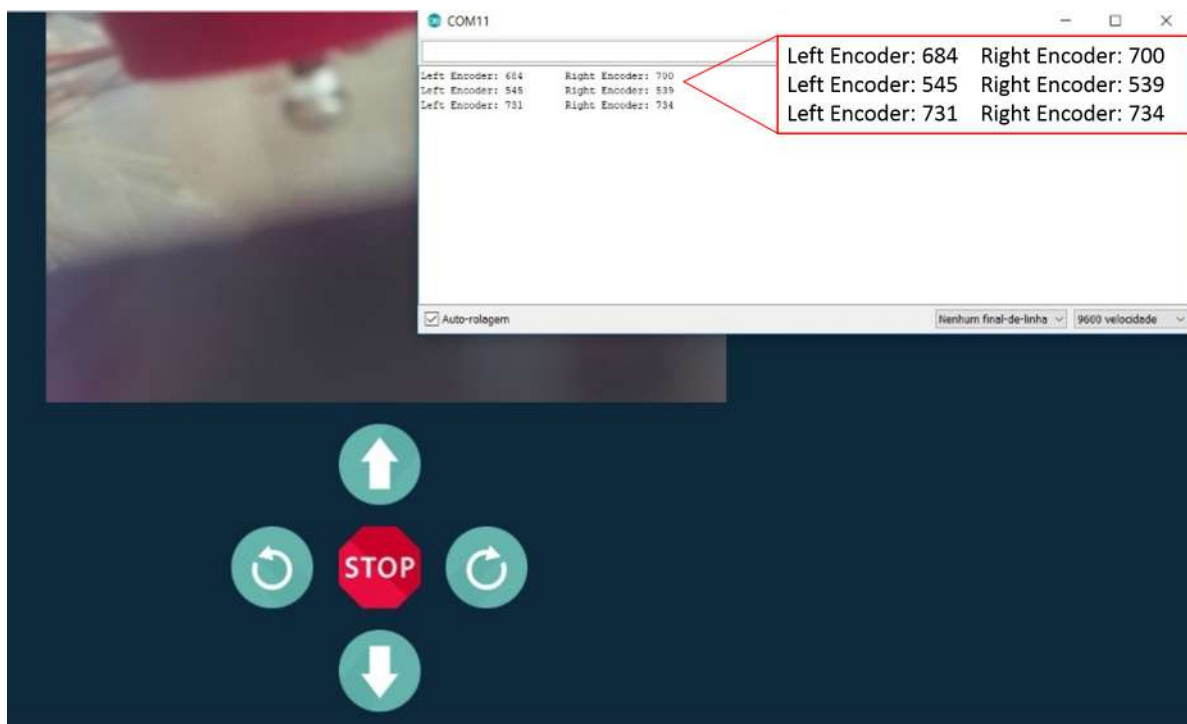


Figura 4.10: Leitura dos *encoders* avaliadas durante teste de deslocamento.

Com o resultado apresentado na Figura 4.10, verificou-se que os atuadores presentes no sistema, apresentam desempenho aceitável para o controle em malha aberta. É importante observar que o robô proposto não é um dispositivo autônomo, mas sim um dispositivo tele operado. Desse modo, apesar de não ser necessário a aquisição e inclusão de mais componentes no sistema, optou-se por não projetar um sistema de controle em malha fechada, deixando a responsabilidade de controle e suas respectivas correções para o operador do sistema.

4.5 Células de Energia

Após os cálculos de consumo energético dos principais componentes do robô, foi definido o modelo de bateria da Gens Ace que fornece valores superiores para todas as especificações, garantindo uma margem de segurança para alterações no sistema e incertezas não consideradas nos cálculos, mas inerentes à implementação prática de projetos como esse. Sua potência de $57,7W.h$ é consideravelmente maior que os $31,8W.h$ necessários, a taxa de descarga garante um fornecimento de corrente máxima de $7,8A$, também superior aos $6,2A$ drenados pelo sistema. Entretanto, durante o processo de aquisição da bateria, o modelo especificado ficou fora de estoque nos fornecedores cadastrados no procedimento determinado pelo AWC. Porém, foi possível comprar um modelo similar, de $5000mAh$, porém com o dobro da capacidade de carga e descarga, ou seja, $30C$, que

pode ser observado na Figura 4.11.



Figura 4.11: Modelo de bateria de LiPo 3S 5000mAh 30C utilizada (FABMODELISMO (2017)).

4.6 Circuitos Eletrônicos

4.6.1 Simplificações para a parte eletrônica

Para simplificar os circuitos, quantidade de componentes, protocolos de comunicação e programação, foram projetadas algumas mudanças possibilitando o aprimoramento do projeto. Essas alterações são apresentadas a seguir.

Acelerômetro

O acelerômetro analógico MMA7361, escolhido anteriormente apresentava a limitação de não poder ser lido diretamente pelo Raspberry Pi. Visando a eliminação desse problema, foi projetado um circuito eletrônico com a presença de um CI MCP3008. Esse *chip* converteria os valores analógicos do acelerômetro e enviaria os dados para o Raspberry Pi através do protocolo de comunicação SPI. Dessa forma, seria necessária a utilização de mais um protocolo de comunicação ao circuito. Outro agravante seria o alto custo do CI, em torno de R\$19,00, somado aos R\$36,00 do MMA7361 e mais alguns componentes eletrônicos menores, elevando o valor desse circuito perante outras opções.

A solução encontrada foi a utilização de um módulo acelerômetro MMA8452Q, que envia todos os seus dados digitalmente por meio da comunicação I²C, a qual já é utilizada no projeto. Outra vantagem é o seu funcionamento com 3,3V, possibilitando sua conexão direta ao Raspberry Pi. Além disso, esse sensor é encontrado por R\$29,00 e não necessita de nenhum componente externo para sua implementação, diminuindo consideravelmente o custo em relação à opção anterior. O acelerômetro MMA8452Q é apresentado na Figura 4.12, valendo ressaltar que já é calibrado de fábrica, de modo que os valores de calibração são armazenados em memória não volátil no circuito. Assim, dispensa-se uma calibração adicional pelo usuário para aplicações comuns (FREESCALE (2013), p.12).

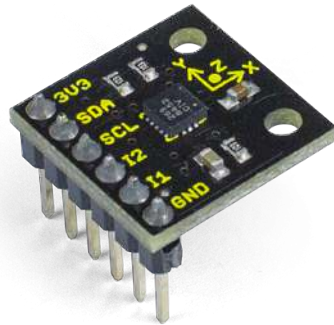


Figura 4.12: Acelerômetro de 3 Eixos MMA8452Q (ROBOCORE (2016)).

Comunicação I²C

O circuito proposto para conversão bidirecional de nível lógico realiza sua função de forma satisfatória, porém cada MOSFET IRF540N, presente nesse circuitos, é vendido por volta de R\$3,00. Adicionalmente, a soldagem dos mesmos deve ser cuidadosa, visto que podem ser facilmente danificados pelo calor do ferro de solda ou estática das mãos. Assim, durante a aquisição dos componentes é recomendada a inclusão de MOSFETs sobressalentes, em caso de danos durante a soldagem, aumentando o preço do circuito. Entretanto, foram encontrados no mercado, módulos conversores de nível lógico bidirecionais por R\$8,00, capazes de exercer a função necessária para o projeto além de serem desenvolvidos em placas profissionais, garantindo mais confiabilidade para o projeto. Portanto, essa opção foi considerada mais atraente e foi escolhida para o projeto. O módulo conversor de lógica bidirecional adquirido é exposto pela Figura 4.13.

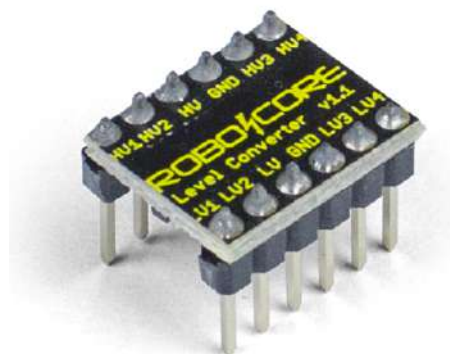


Figura 4.13: Conversor de lógica bidirecional (ROBOCORE (2015)).

Microcontrolador

O circuito necessário para a utilização do microcontrolador PIC16F876A apresenta muitos componentes, o que demanda tempo para sua confecção e ainda conta com a possibilidade de falhas de soldagem serem encontradas por não ser uma placa profissional. Complementarmente, esse microcontrolador é encontrado no mercado a um custo de R\$18,00, também sendo necessários componentes adicionais, como um soquete de 28 pinos, cristal oscilador, botão, LEDs e resistores. Entretanto, uma placa Arduino Uno com um microcontrolador ATmega328, apresentado na Figura 4.14, pode ser adquirida por R\$20,00, sendo essa uma placa profissional. A facilidade de programação nessa placa, aliada à familiaridade dos pesquisadores com a mesma e a eliminação da confecção da placa eletrônica, tornam sua escolha ainda mais justificável para o projeto. Assim são garantidos uma maior eficiência e resultados mais confiáveis.



Figura 4.14: Placa Arduino Uno (ARDUINO (2015)).

Circuito Eletrônico Simplificado

O circuito completo após as simplificações anteriormente citadas foi montado no programa KiCad e seu esquemático é apresentado pela Figura 4.15. Ainda através desse programa, foi desenvolvido o PCB para a montagem da placa eletrônica, o qual é apresentado pela Figura 4.16.

4.6.2 Placas eletrônicas

Como apresentado na Seção 3.7, foram propostos três circuitos eletrônicos para a implementação do projeto, que são apresentados pelas Figuras 4.15, 3.27 e 3.29. Esses circuitos permitirão que o robô possa ser operado através do Raspberry Pi sem que hajam possíveis problemas. Além desses circuitos garantirem a alimentação de todos os dispositivos presentes no robô, eles são desenvolvidos para a utilização plena do Raspberry Pi com as outras partes do robô: permitindo a comunicação segura entre ele e outros dispositivos; a utilização de um microcontrolador para se realizar a leitura dos *encoders* presentes nos dois motores, não sobrecarregando o Raspberry Pi com as interrupções geradas pelos mesmos; além de melhorar a organização do robô o facilitar a sua montagem. Os PCBs para a implementação dos circuitos propostos anteriormente, foram apresentados nas Figuras 4.16, 3.28 e 3.30. No primeiro desses circuitos foram colocados barramentos de pinos fêmea para que os módulos do acelerômetro MMA8452Q e do conversor de lógica bidirecional pudessem ser encaixados diretamente na placa. Essa placa eletrônica já confeccionada é apresentada pela Figura 4.17. Já a segunda placa, implementada para a utilização dos sensores ultrassom, é exposta pela Figura 4.18. A placa eletrônica implementada para simplificar o acoplamento dos fios dos motores é apresentada pela Figura 4.19.

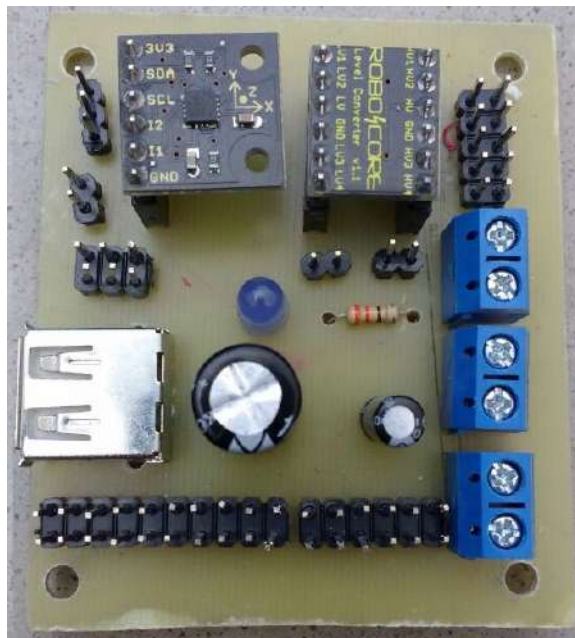


Figura 4.17:

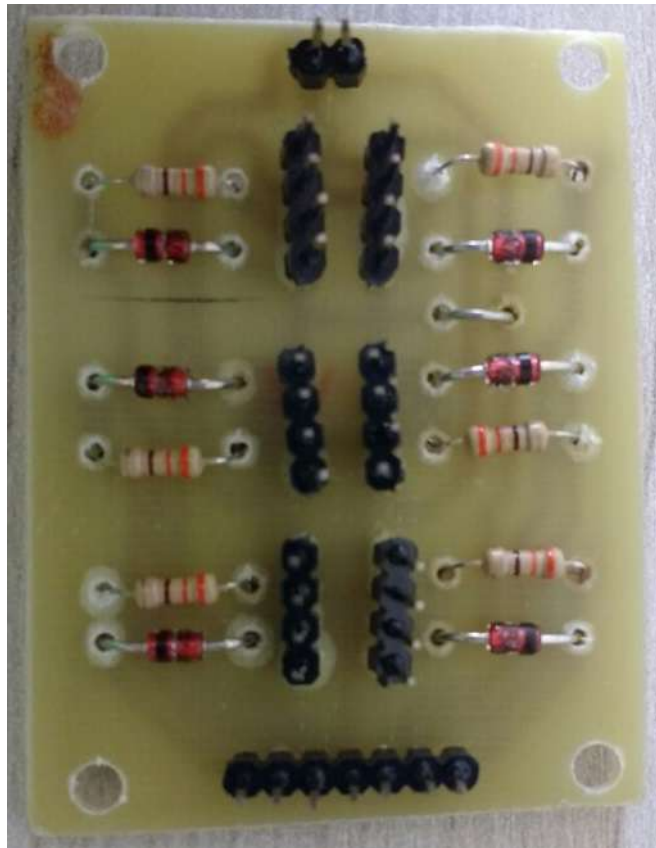


Figura 4.18: Placa eletrônica para os sensores ultrassom

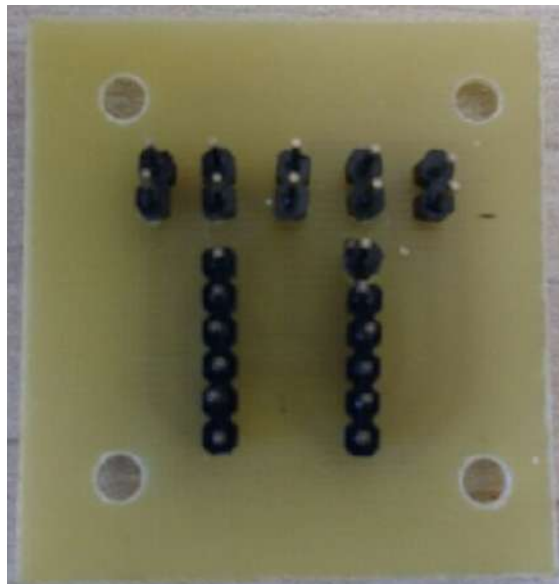


Figura 4.19: Placa eletrônica para acoplamento dos cabos dos motores.

4.7 Programação

Na parte de programação foram implementadas duas páginas web para se realizar o controle dos movimentos do robô. A primeira página foi implementada para a utilização em *notebooks*, portanto, apresenta um *layout* em formato de paisagem, para facilitar a visualização na tela. Além disso, a página é desenvolvida com botões que utilizam funções próprias para avaliar o clique do *mouse*. A página web para *notebooks* pode ser observada pela Figura 4.20.

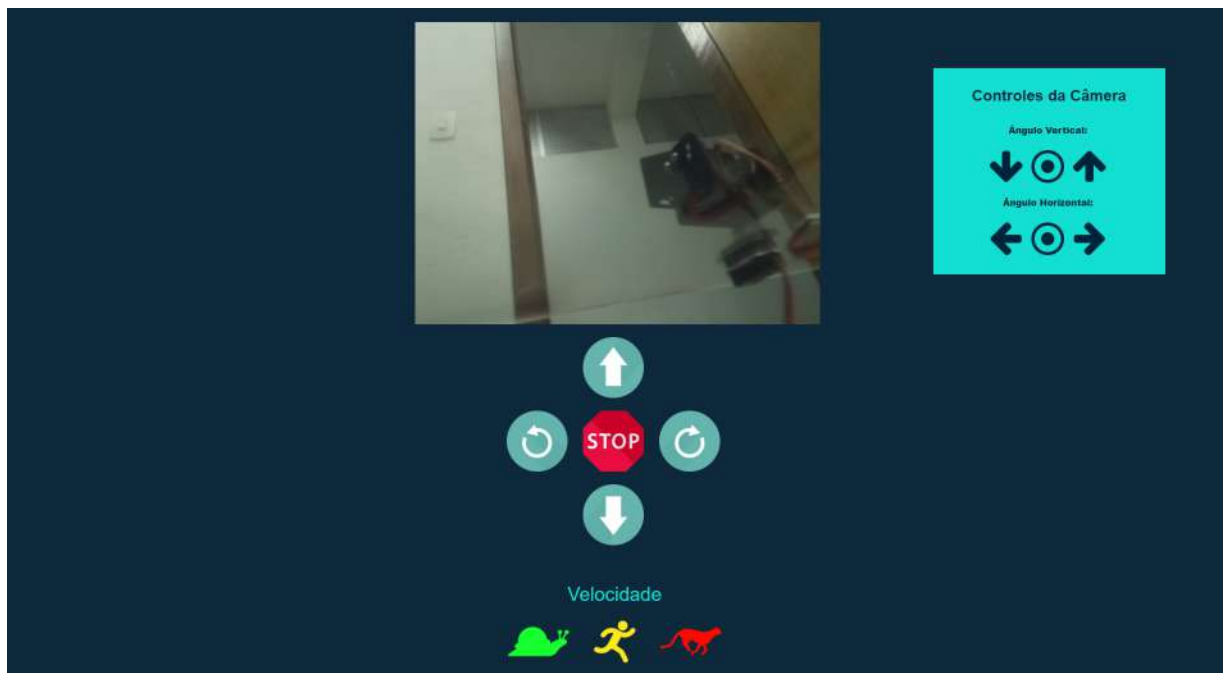


Figura 4.20: Página web para *notebooks*.

A outra página é focada em dispositivos móveis, como *tablets* e *smartphones*. Dessa forma, a orientação da página pode ser feita no formato de retrato e foram utilizados nos botões funções que avaliam o toque na tela do aparelho. Essa página deve ser utilizada preferivelmente em *tablets*, pois, se torna mais fácil a visualização da imagem da câmera por terem telas maiores, entretanto, podem também serem usados os *smartphones* para se realizar a operação do robô. A página web para dispositivos móveis é apresentada pela Figura 4.21.

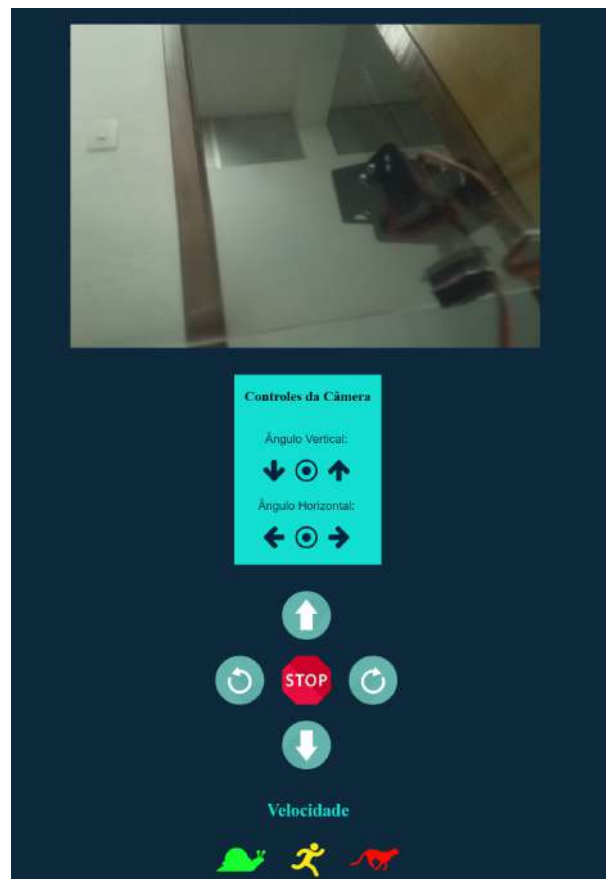


Figura 4.21: Página web para dispositivos móveis.

No topo da página web é apresentada a imagem da câmera do Raspberry Pi, permitindo assim a identificação do ambiente onde o robô está presente para que seja realizada a sua operação. A variação da orientação da câmera, para se alterar seu ângulo de visão, pode ser realizada através dos comandos presentes na parte *Controles da Câmera*. Pode-se mudar a orientação da câmera verticalmente ou horizontalmente e para isso são implementados controles intuitivos, que visam facilitar o entendimento do usuário, tanto com a linguagem verbal quanto a não verbal.

Os controles de movimentação do robô aparecem abaixo, através de cinco botões, nos quais são usados ícones que também buscam facilitar o entendimento de suas operações. Por fim, são implementados controles de velocidade para o robô, que também buscam ser tão intuitivos quanto possível: utilizam cores que remetem ao aumento de velocidade e perigo, sendo utilizada a cor verde, para a velocidade mais lenta e a cor vermelha para a velocidade máxima. São também utilizados animais conhecidos para representar as opções: um caracol para a velocidade lenta, um humano para a média e um guepardo para a velocidade máxima. Dessa forma, busca-se em toda a implementação visual do controle que este seja amigável e intuitivo, para facilitar ao máximo o entendimento de suas opções e a operação do robô.

4.8 Integração dos Subsistemas

Em virtude do término de todas as partes específicas de cada subsistema, foi iniciada a montagem e integração das partes do robô. O funcionamento simultâneo dos elementos do robô foi satisfatório e atendeu aos requisitos propostos durante o desenvolvimento do projeto. A plataforma robótica funcionou de maneira eficiente e segura durante sua operação, além de garantir facilidade de controle por diferentes usuários. O robô montado é apresentado na Figura 4.22.



Figura 4.22: Montagem final do robô.

4.9 Plataforma de Visitas Virtuais

Todo o desenvolvimento do robô móvel tem como objetivo o aumento da eficiência do processo de aquisição das imagens 360° para a criação de uma plataforma de visitas virtuais. Durante o período burocrático da compra e entrega dos componentes do *hardware* do robô, foi possível criar a plataforma de visualização dos imóveis e suas visitas virtuais.

Visando uma melhoria integral do processo de pesquisa e avaliação desses imóveis, foi construído um sistema de procura geográfica dos locais, cuja interface é exibida na Figura 4.23. Utilizando a API do Google Maps, é possível visualizar a localização de todos os imóveis cadastrados, ou mesmo definir uma região de interesse através da ferramenta “Desenhar”. Assim, o usuário cria retas a cada clique, que quando unidas, formam um polígono dentro do qual serão exibidos os imóveis disponíveis exclusivamente em seu interior, conforme pode ser verificado na Figura 4.24.

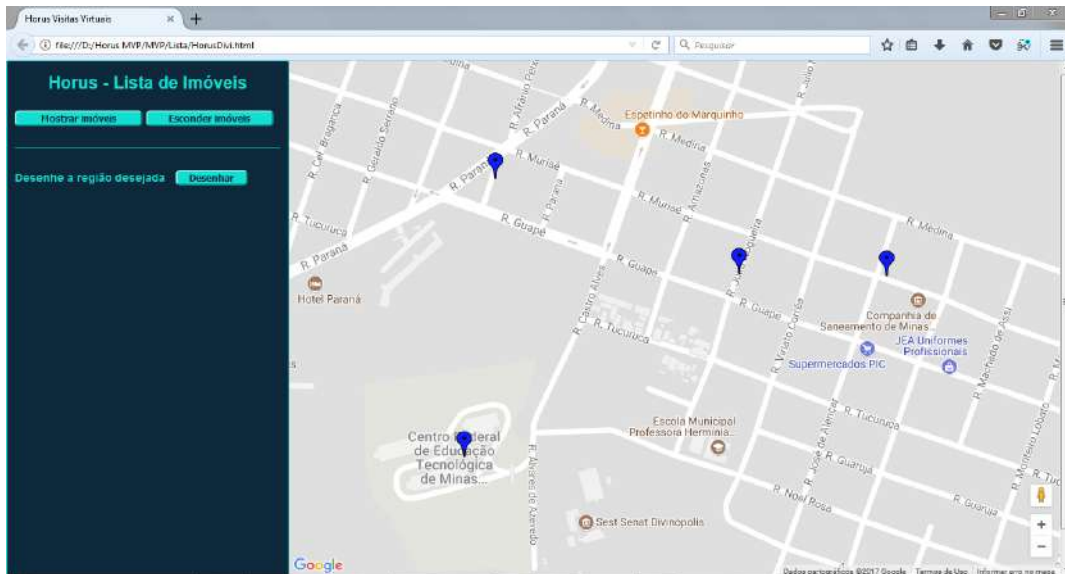


Figura 4.23: Exibição da tela inicial da plataforma de visitas virtuais evidenciando imóveis cadastrados.

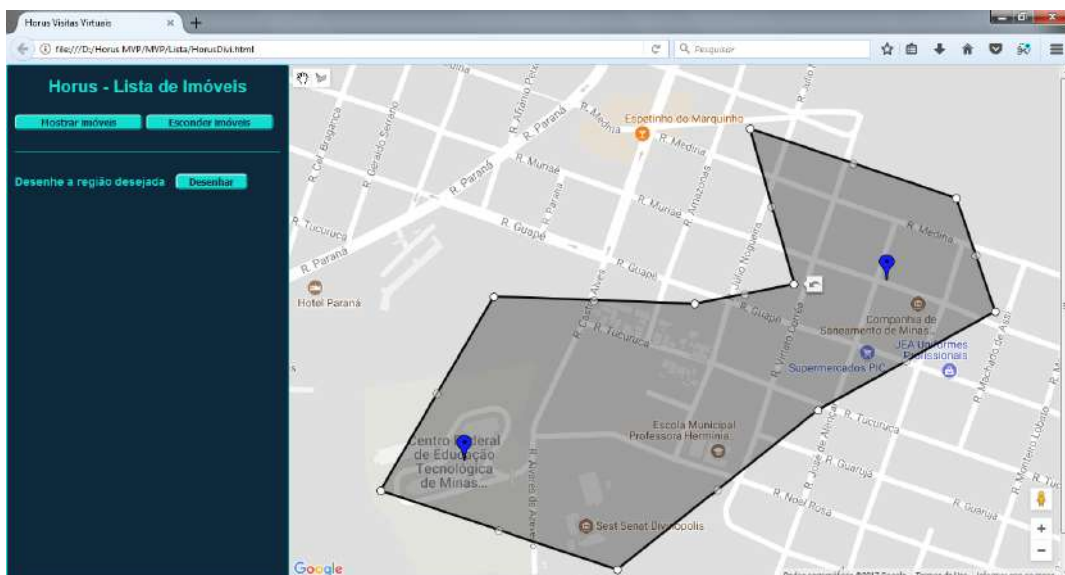


Figura 4.24: Ferramenta “Desenhar” da plataforma de visitas virtuais.

Ao clicar sobre o ícone de uma propriedade específica, uma janela se abre com a visualização do Google Street View daquele ponto, permitindo ao usuário a visualização da vizinhança e navegação pelas ruas próximas, conforme apresentado pela Figura 4.25. É possível optar pela visualização em tela inteira, útil para uma navegação mais detalhada do exterior do imóvel e sua região. Caso a área em que uma propriedade cadastrada se situa não tenha sido mapeada pelo Street View, uma imagem padrão ou 360° da fachada pode ser inserida, aproximando o usuário da experiência da Google, conforme apresentado pela Figura 4.26.

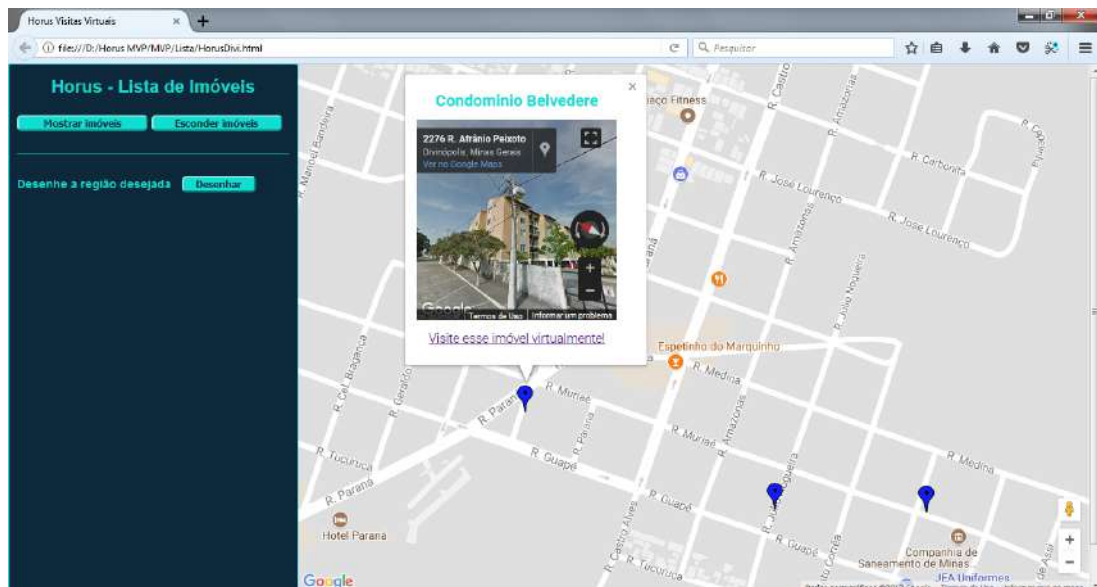


Figura 4.25: *Pop-up* do Google Street View para visualizar a vizinhança.

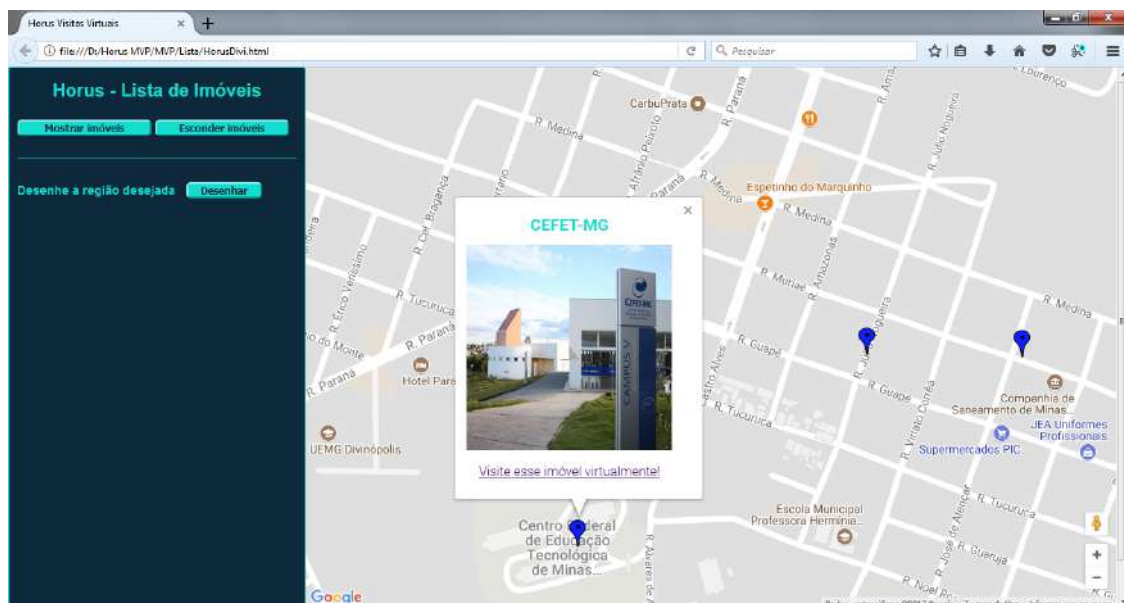


Figura 4.26: Exemplo de fotografia padrão para imóveis sem suporte do Google Street View.

Na parte inferior desse *pop-up*, existe um link para visita virtual, caso o usuário deseje visualizar o interior do imóvel. Nesse estágio, ele pode navegar de forma similar ao Google Street View, alternando entre as imagens 360° capturadas pela Ricoh Theta S acoplada ao robô, conforme pode ser verificado na Figura 4.27. Desse modo, o locatário ou comprador imobiliário é capaz de visualizar o interior da propriedade com um nível de detalhes bem superior às imagens padrões, além do melhor entendimento dimensional e da configuração do imóvel.

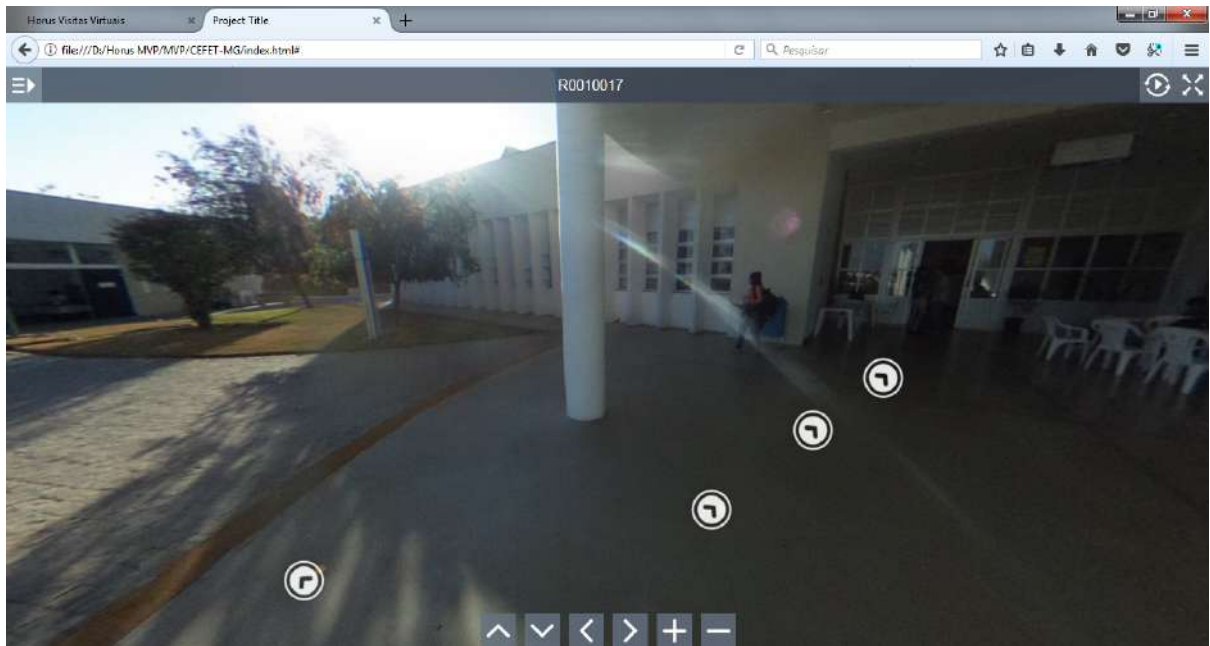


Figura 4.27: Interface de navegação dos ambientes internos mapeados.

Conclusões

Neste capítulo, são tratados os conceitos gerais obtidos a partir da realização do trabalho. Inicialmente, o cronograma e orçamento são brevemente abordados. Na seção 5.1 retorna-se aos resultados alcançados objetivando-se a apresentação das conclusões geradas por eles. Na seção 5.2 cita-se pontos de continuidade para o aprimoramento do trabalho desenvolvido.

Cronograma

Quanto ao seguimento do cronograma inicialmente proposto, houveram alguns contratempos no segundo semestre. Devido ao processo de compra altamente burocrático exigido judicialmente para financiamentos como o obtido junto ao AWC, os prazos para realização das aquisições dos componentes se mostraram mais longos do que o planejado. Problemas alfandegários durante a importação de componentes americanos como a câmera e motores alteraram o prazo máximo proposto de cerca de um mês para quatro meses no caso da Ricoh Theta S. Foi necessário que o dispositivo retornasse ao país de origem duas vezes, o que inviabilizou o comprimento de algumas tarefas dentro do cronograma. Apesar dos contratempos, esse intervalo possibilitou o desenvolvimento da plataforma de visitas virtuais, componente de alta importância por concluir todo o desenvolvimento tecnológico do contexto do problema e oportunidade mercadológica identificados. Os atrasos foram superados nas últimas etapas, de modo que o prazo final pôde ser cumprido e os resultados apresentados condizentes com o objetivo proposto.

Orçamento

Os custos totais do projeto foram de aproximadamente R\$ 7000. Conforme já mencionado, o projeto foi financiado pelo AWC, o que viabilizou sua execução, apesar do custo elevado. Essa quantia inclui também a participação dos autores na maior feira do mercado imobiliário da América Latina, onde foi possível conhecer empresas que trabalham com

soluções similares e o sistema de *hardware* considerado a referência na criação das visitas virtuais atualmente. Na ocasião, foi possível entender os pontos negativos e positivos do equipamento, além de validar a hipótese dos problemas processuais do mapeamento ao conversar com os operadores que o fazem de forma manual. Grande parcela do orçamento foi dispendida em impostos de importação, visto que certos componentes não estavam disponíveis no mercado nacional. Outra justificativa para o alto custo foi a necessidade de compra de fornecedores específicos que atendiam as demandas do programa para a realização do pagamento. Em sua maioria, os produtos se apresentavam com um preço maior do que o praticado pela concorrência. Logo, estima-se que a plataforma robótica possa ser construída dentro de um orçamento de R\$ 4000, incluindo a Ricoh Theta S de aproximadamente R\$ 2200.

5.1 Considerações Finais

De forma geral, tanto o objetivo geral, quanto os objetivos específicos foram atendidos conforme proposto. O robô móvel se mostrou capaz de operar com o objetivo para o qual foi projetado. Sua operação é simples, sendo capaz de percorrer ambientes mobiliados sem colidir e com boa manobrabilidade. A inclinação máxima admitida comprova a estabilidade do sistema, aspecto questionado durante o projeto. O principal contribuinte para tal resultado é o tubo telescópico consideravelmente leve, aliado à base pesada e de grandes dimensões proporcionadas pelas placas de acrílico. Assim, seu centro de massa se manteve baixo.

O desempenho dos motores certamente atenderam aos requisitos estabelecidos, além de atuarem de forma similar, o que eliminou a necessidade de aplicação de controle no sistema. Assim, a mecânica em geral se comportou conforme o estimado, apresentando uma montagem simples e sem grandes contratempos, graças a um projeto elaborado satisfatoriamente. A estabilidade dinâmica do robô superou drasticamente o esperado, sendo capaz de deslocar-se em todas superfícies inclinadas permitidas pela NBR9050, que limita-se a inclinações de 12,5% para rampas. O robô foi submetido à vários testes dentro do próprio *campus*, e desempenhou conforme o esperado tanto em morros, como também em superfícies levemente acidentadas.

O projeto de eletrônica do robô funcionou de forma satisfatória, realizando a alimentação de todas as partes que compõem o robô com os valores de tensão ideais para cada uma, além de viabilizar a transmissão de dados entre os sensores, o Raspberry Pi, o Arduino e os motores. Dessa forma, a eletrônica foi parte importante para viabilizar todo o funcionamento e integração entre os elementos do robô.

Todos os programas implementados, utilizando-se diferentes linguagens de programa-

ção, puderam ser integrados de forma satisfatória para uma operação eficiente e segura do robô. A plataforma pôde ser controlada facilmente por diferentes usuários, provando-se que a página de controle é intuitiva. Além disso, toda a parte de código por trás das operações funcionou corretamente e sem problemas. O robô pôde ser operado com segurança, com os programas que utilizam os sensores presentes na plataforma funcionando a fim de evitar colisões ou o tombamento do robô. Os comandos enviados ao microcontrolador também foram cumpridos com sucesso. Todos os programas presentes no sistema funcionaram em perfeita harmonia ao fim da implementação do robô.

O *feedback* dos primeiros usuários é positivo, validando as hipóteses levantadas no início da execução do trabalho e justificando a construção de um sistema robótico para tal aplicação. A utilização do robô também permite a gravação de vídeo 360°, os quais podem ser usados para introduzir e apresentar o imóvel de uma forma mais rápida para aquele usuário no início do processo de compra. Nessa etapa, ele normalmente está interessado em conhecer um grande número de imóveis. Assim, o vídeo pode preceder a visita virtual, sendo essa reservada para as propriedades consideradas mais atrativas por parte do consumidor.

Finalmente, de forma geral, o trabalho serviu com o propósito de aplicar-se os conhecimentos acumulados durante a graduação em Engenharia Mecatrônica. O fato de sua execução ter sido realizada em paralelo a um programa de educação empreendedora e desenvolvimento de negócios, resultou em um nível mais abrangente de conhecimentos sendo abordados. Assim, não só se obteve um protótipo funcional, mas sim uma ferramenta aplicável em situações reais de mapeamento de imóveis, viabilizando a criação de um negócio que se mostra promissor. Desse modo, as decisões de projeto foram tomadas levando em consideração pontos mais importantes que apenas sua finalização como projeto de conclusão de curso. Essa abordagem diferenciada contribuiu para a formação de engenheiros capazes de basear seu julgamento em conceitos técnicos profundos, mas também considerando os aspectos econômicos e financeiros, os quais se mostram igualmente importantes na atuação como profissional da área. A atenção à opinião de usuários e possíveis clientes também influenciou na obtenção de resultados satisfatórios para um público mais geral. Como já mencionado, a elaboração da plataforma de visitas virtuais completa o ciclo de desenvolvimento tecnológico necessário para viabilizar o negócio envolvido, suprimindo uma necessidade de mercado através da utilização de um dispositivo mecatrônico.

5.2 Propostas de Trabalhos Futuros

Existem alguns pontos que podem ser aprimorados no trabalho, especificamente no robô móvel. Primeiramente, o sistema de prevenção de colisões pode ser incrementado com a inserção de mais sensores. Ainda nessa linha, apesar da parte inferior do robô estar satisfatoriamente instrumentada, o conjunto tubo telescópico e câmera não conta com um sistema passivo para evitar colisões com obstáculos elevados e distantes do solo.

Outro aspecto passível de melhorias é a comunicação Wi-Fi da câmera e o *smartphone*. Apesar de o robô poder se distanciar do operador, o curto alcance da rede da câmera obriga que o operador esteja razoavelmente próximo. Normalmente, nos ambientes testados, essa distância foi de cerca de 5 a 7 metros. Um sistema de amplificação ou intermediação para essa conexão entre os dois dispositivos pode ser implementado. Isso leva à outra melhoria possível: a inclusão da opção de captura das fotografias na própria plataforma de controle do robô. Assim, elimina-se a necessidade de alternância entre aplicativos e telas em um *gadget*, ou a utilização de mais de um dispositivo.

Caso os motores passem a trabalhar de forma distinta entre eles, os desvios da trajetória retilínea podem ser tornar relevante. Portanto, para evitar que a aquisição das imagens seja prejudicada, é possível melhorar o desempenho dos motores, e consequentemente, a leitura dos *encoders*, implementando controladores que garantem a mesma velocidade para ambos os motores.

Em um nível mais avançado, também seria promissor o aprimoramento do robô para torná-lo autônomo, sendo capaz de reconhecer o ambiente a ser mapeado e decidir sobre o melhor posicionamento para as fotografias. Assim, elimina-se a necessidade de um operador capaz de pilotá-lo, bastando um funcionário para transportá-lo até os imóveis, visto que o processo de aquisição tornar-se-ia automático.

Códigos

A.1 Código dos Sensores

```
import smbus
import time
import math
import sys
5 import RPi.GPIO as GPIO

class Accel:
    def initAccel(self):
        bus = smbus.SMBus(1)
10         bus.write_byte_data(0x1D, 0x2A, 0x00)
           bus.write_byte_data(0x1D, 0x2A, 0x01)
           bus.write_byte_data(0x1D, 0x0E, 0x00)
           time.sleep(0.5)

15     def angle(self, pitchMax, rollMax):
        bus = smbus.SMBus(1)
        data = bus.read_i2c_block_data(0x1D, 0x00, 7)

        # Conversão dos dados
20         xAccel = (data[1] * 256 + data[2]) / 16
           if xAccel > 2047 :
               xAccel -= 4096

           yAccel = (data[3] * 256 + data[4]) / 16
25         if yAccel > 2047 :
               yAccel -= 4096

           zAccel = (data[5] * 256 + data[6]) / 16
30         if zAccel > 2047 :
               zAccel -= 4096

           pitch = (math.atan2(xAccel, math.sqrt(math.pow(yAccel, 2) +
               math.pow(zAccel, 2)))) * 180 / math.pi
           roll = (math.atan2(yAccel, zAccel)) * 180 / math.pi

35         if abs(pitch) > pitchMax or abs(roll) > rollMax:
            bus.write_byte(0x07, 0x00)
            sys.exit(0)
```

```

class Ultra:
40     def dist(self, echo, distMax):
        GPIO.setmode(GPIO.BCM)
        GPIO_TRIGGER = 21
        GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
        GPIO.output(GPIO_TRIGGER, True)
45     time.sleep(0.00001)
        GPIO.output(GPIO_TRIGGER, False)

        StartTime = time.time()
        StopTime = time.time()

50     while GPIO.input(echo) == 0:
        StartTime = time.time()

        while GPIO.input(echo) == 1:
55     StopTime = time.time()

        TimeElapsed = StopTime - StartTime
        # multiplica pela velocidade do som (34300 cm/s)
        # e divide por 2, porque o sinal vai e volta
60     distancia = (TimeElapsed * 34300) / 2

        if (distancia > distMax):
            bus.write_byte(0x07, 0x00)
            sys.exit(0)

```

A.2 Código de movimentação retilínea

```

import smbus
import time
import sys
import RPi.GPIO as GPIO
5 from sensores import Accel
from sensores import Ultra

distMax, pitchMax, rollMax = 17.3, 30.0, 30.0

10 bus = smbus.SMBus(1)
GPIO.setmode(GPIO.BCM)

GPIO_ECHO1 = 20
GPIO_ECHO2 = 26
15 GPIO.setup(14, GPIO.OUT, initial=0)
GPIO.setup(15, GPIO.IN)
GPIO.setup(GPIO_ECHO1, GPIO.IN)
GPIO.setup(GPIO_ECHO2, GPIO.IN)

20 bus = smbus.SMBus(1)
bus.write_byte_data(0x1D, 0x2A, 0x00)
bus.write_byte_data(0x1D, 0x2A, 0x01)
bus.write_byte_data(0x1D, 0x0E, 0x00)

25 u = Ultra()
a = Accel()

```

```

u. dist (GPIO_ECHO1, distMax)
u. dist (GPIO_ECHO2, distMax)
30 a. angle (pitchMax, rollMax)

bus. write_byte (0x07, 0x01)

while (True):
35     u. dist (GPIO_ECHO1, distMax)
        u. dist (GPIO_ECHO2, distMax)
        a. angle (pitchMax, rollMax)
        if (GPIO.input (15)):
            bus. write_byte (0x07, 0x00)
40     sys. exit (0)

```

A.3 Código do Arduino

```

#include "Arduino.h"
#include <Wire.h>
#include <Servo.h>

5 // Quadrature encoders
// Left encoder
#define c_LeftEncoderInterrupt 0 //2
#define c_LeftEncoderPinA 2 //0
#define c_LeftEncoderPinB 11
10 #define LeftEncoderIsReversed
volatile bool _LeftEncoderBSet;
volatile long _LeftEncoderTicks = 0;

// Right encoder
15 #define c_RightEncoderInterrupt 1 //3
#define c_RightEncoderPinA 3 //1
#define c_RightEncoderPinB 12
volatile bool _RightEncoderBSet;
volatile long _RightEncoderTicks = 0;

20 #define vel_A 5
#define motor_A 7
#define vel_B 6
#define motor_B 8

25 Servo pan; //pino 9
Servo tilt; //pino 10

byte slave_address = 7;
30 byte stp = 0x00;
byte forward = 0x01;
byte left = 0x02;
byte right = 0x03;
byte reverse = 0x04;

35 bool mA = false;
bool mB = false;
bool ligado = false;
int panAng=90, passoPan=15, tiltAng=90, passoTilt=15;
40 int minima = 75;

```

```

int media = 160;
int maxima = 255;
int velocidade = media;

45 void setup() {
    // Start I2C Bus as Slave
    Wire.begin(slave_address);
    Wire.onReceive(receiveEvent);

50    pan.attach(9);
    pan.write(90);
    tilt.attach(10);
    tilt.write(85);

55    // Quadrature encoders
    // Left encoder
    pinMode(c_LeftEncoderPinA, INPUT); // sets pin A as input
    digitalWrite(c_LeftEncoderPinA, LOW); // turn on pullup resistors
    pinMode(c_LeftEncoderPinB, INPUT); // sets pin B as input
60    digitalWrite(c_LeftEncoderPinB, LOW); // turn on pullup resistors
    attachInterrupt(c_LeftEncoderInterrupt, HandleLeftMotorInterruptA,
        RISING);

    // Right encoder
    pinMode(c_RightEncoderPinA, INPUT); // sets pin A as input
65    digitalWrite(c_RightEncoderPinA, LOW); // turn on pullup resistors
    pinMode(c_RightEncoderPinB, INPUT); // sets pin B as input
    digitalWrite(c_RightEncoderPinB, LOW); // turn on pullup resistors
    attachInterrupt(c_RightEncoderInterrupt, HandleRightMotorInterruptA,
        RISING);

70    pinMode(vel_A, OUTPUT);
    pinMode(motor_A, OUTPUT);
    pinMode(vel_B, OUTPUT);
    pinMode(motor_B, OUTPUT);
    digitalWrite(vel_A, LOW);
75    digitalWrite(motor_A, LOW);
    digitalWrite(vel_B, LOW);
    digitalWrite(motor_B, LOW);
}

80 void loop() {
    digitalWrite(motor_A, mA);
    digitalWrite(motor_B, mB);
    if(ligado){
        analogWrite(vel_A, velocidade);
85        analogWrite(vel_B, velocidade);
    }else{
        analogWrite(vel_A, 0);
        analogWrite(vel_B, 0);
    }
90 }

void receiveEvent(int howMany) {
    byte cmd = Wire.read();
    switch(cmd){
95        case 0x00: //stop
            ligado=false;

```

```

        Serial.print("Left Encoder: ");
        Serial.print(_LeftEncoderTicks);
        Serial.print("\t");
100      Serial.print("Right Encoder: ");
        Serial.print(_RightEncoderTicks);
        Serial.print("\n");
    break;
    case 0x01:          //forward
105      ligado=true;
        mA=true;
        mB=true;
        tipo=1;
    break;
    case 0x02:          //left
110      ligado=true;
        mA=true;
        mB=false;
        tipo=2;
    break;
    case 0x03:          //right
115      ligado=true;
        mA=false;
        mB=true;
        tipo=3;
    break;
    case 0x04:          //reverse
120      ligado=true;
        mA=false;
        mB=false;
        tipo=4;
    break;
    case 0x10:          //minima
125      velocidade=minima;
    break;
    case 0x11:          //media
130      velocidade=media;
    break;
    case 0x12:          //maxima
135      velocidade=maxima;
    break;
    case 0x20:          //esquerda
        panAng+=passoPan;
        if (panAng>180){
140             panAng=180;
        }
        pan.write(panAng);
    break;
    case 0x21:          //centro
145      pan.write(90);
        panAng=90;
    break;
    case 0x22:          //direita
        panAng-=passoPan;
150      if (panAng<0){
        panAng=0;
        }
        pan.write(panAng);
    break;

```

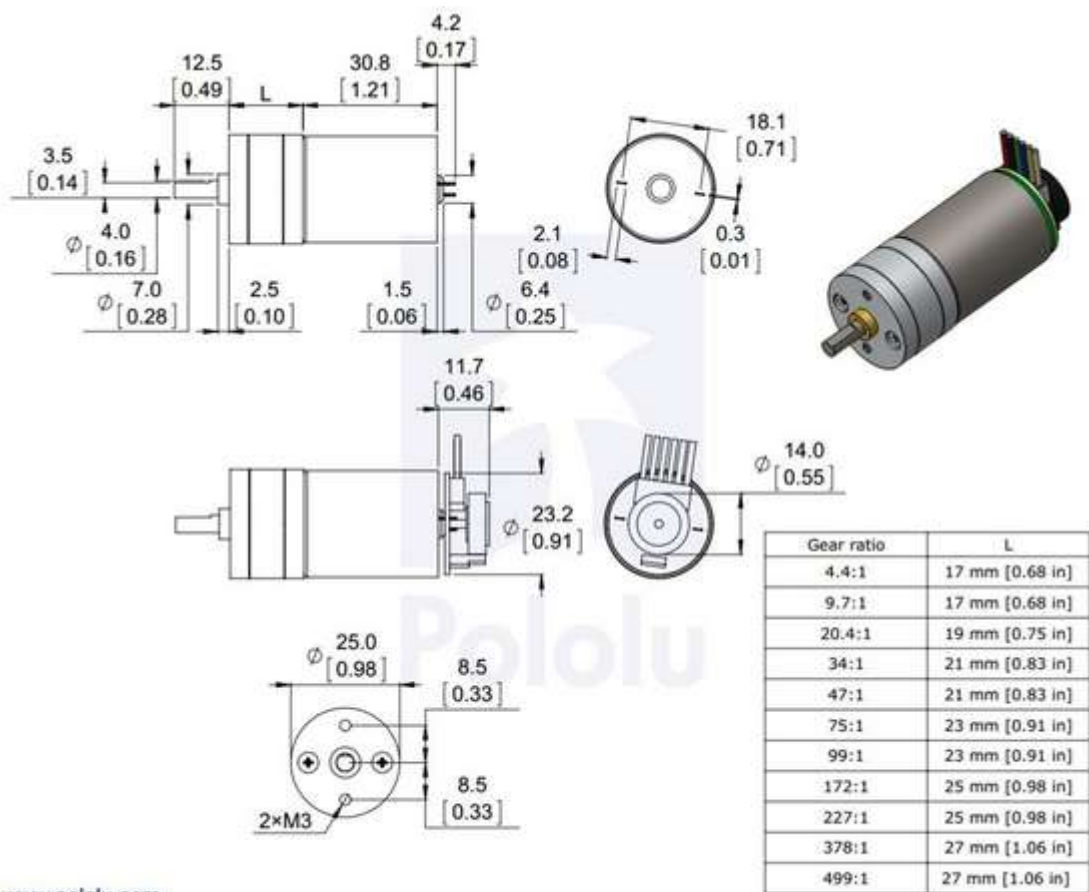
```
155         case 0x23:           //para baixo
                tiltAng+=passoTilt;
                if (tiltAng>180){
                    tiltAng=180;
                }
160         tilt.write(tiltAng);
        break;
        case 0x24:           //centro
                tilt.write(90);
                tiltAng=90;
165        break;
        case 0x25:           //para cima
                tiltAng-=passoTilt;
                if (tiltAng<0){
                    tiltAng=0;
170                }
                tilt.write(tiltAng);
        break;
    }
}
175 void HandleLeftMotorInterruptA () {
    _LeftEncoderBSet = digitalRead(c_LeftEncoderPinB); // read the
        input pin

    #ifdef LeftEncoderIsReversed
180    _LeftEncoderTicks -= _LeftEncoderBSet ? -1 : +1;
    #else
        _LeftEncoderTicks += _LeftEncoderBSet ? -1 : +1;
    #endif
}
185 void HandleRightMotorInterruptA () {
    _RightEncoderBSet = digitalRead(c_RightEncoderPinB); // read the
        input pin

    #ifdef RightEncoderIsReversed
190    _RightEncoderTicks -= _RightEncoderBSet ? -1 : +1;
    #else
        _RightEncoderTicks += _RightEncoderBSet ? -1 : +1;
    #endif
}
```

Apêndice B

Desenhos técnicos



www.pololu.com

Figura B.1: Modelo do motor especificado (POLULU (2014a)).

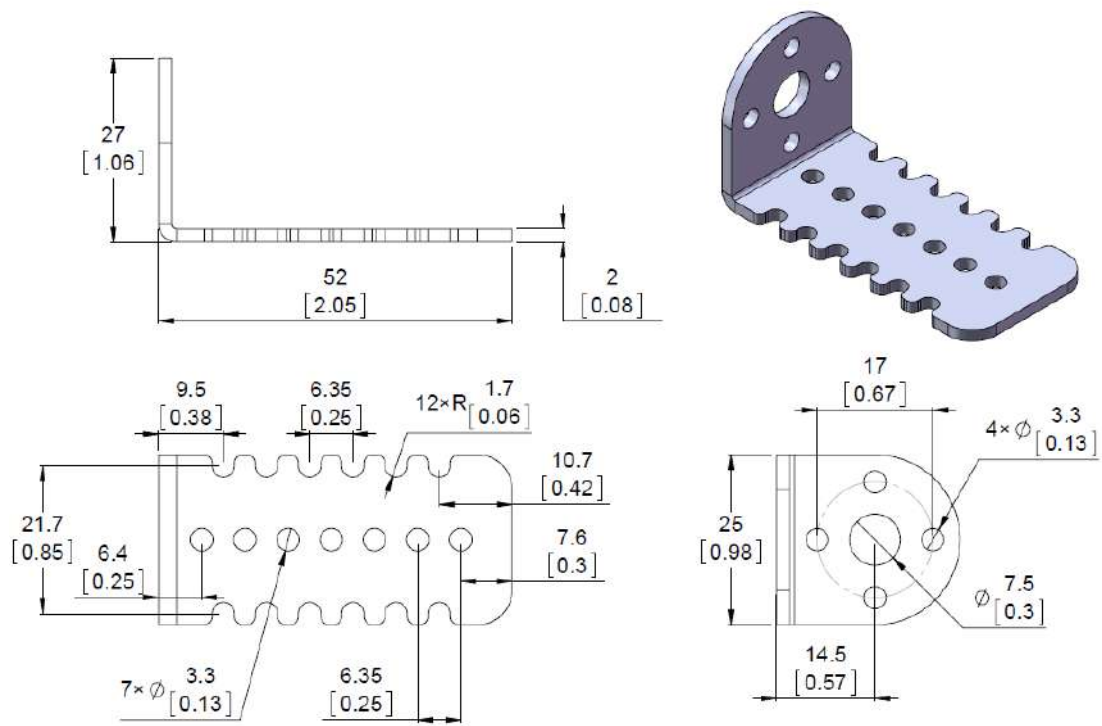


Figura B.2: Suporte para motor (POLULU (2014b)).

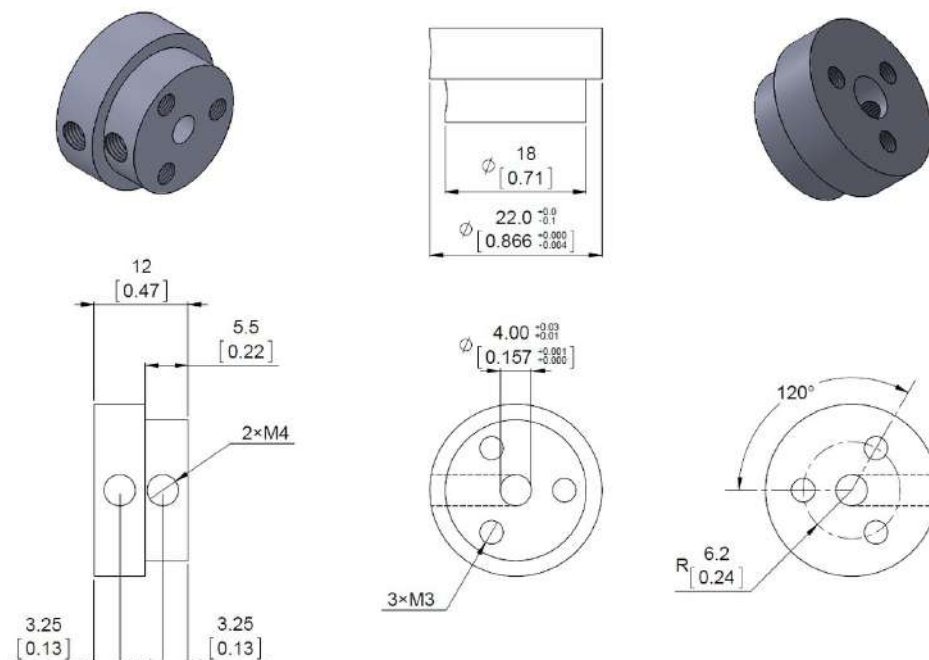


Figura B.3: Adaptador para fixação do motor à roda - parte 1. (POLULU (2014c))

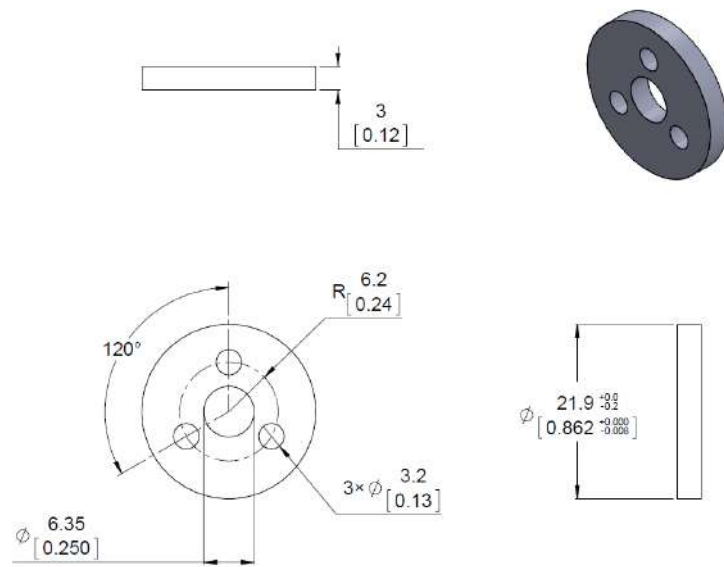


Figura B.4: Adaptador para fixação do motor à roda - parte 2. (POLULU (2014c))

Referências

- ABAL. *Propriedades mecânicas*. Acesso: 15/04/2017, <http://abal.org.br/aluminio/caracteristicas-quimicas-e-fisicas/propriedades-mecanicas/#accordion2>.
- ADAFRUIT. *Basic Resistor Sensor Reading on Raspberry Pi*. Acesso: 12/05/2017, <https://learn.adafruit.com/basic-resistor-sensor-reading-on-raspberry-pi/overview>.
- ADEYEYE, M.; MAKITLA, I.; FOGWILL, T. Determining the signalling overhead of two common WebRTC methods: json via xmlhttprequest and sip over websocket. In: AFRICON, 2013. *Anais...* [S.l.: s.n.], 2013. p.1–5.
- AMAZON. *Amazon Prime Air*. Acesso: 15/05/2017, <https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011>.
- ARDUINO. *Arduino - PWM*. Acesso: 10/05/2017, <https://www.arduino.cc/en/Tutorial/PWM>.
- ARDUINO. *ARDUINO UNO REV3*. Acesso: 25/10/2017, <https://store.arduino.cc/arduino-uno-rev3>.
- AULET, B. *Disciplined entrepreneurship: 24 steps to a successful startup*. [S.l.]: John Wiley & Sons, 2013.
- BASICS, C. *Basics of the I²C Communication Protocol*. Acesso: 13/11/2017, <http://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>.
- BEER, F. P.; JOHNSTON JR, E. R.; CORNWELL, P. J. *Mecânica Vetorial para Engenheiros-Dinâmica*. [S.l.]: AMGH Editora, 2013.
- BERGER, A. S. *Embedded Systems Design: an introduction to processes, tools, and techniques*. [S.l.: s.n.], 2002.

- BLANK, S. *The four steps to the epiphany: successful strategies for products that win*. [S.l.]: BookBaby, 2013.
- BOSTONDYNAMICS. *Boston Dynamics Changing your idea of what robots can do*. Acesso: 15/04/2017, <https://www.bostondynamics.com/robots>.
- BOTELHO, S. S. d. C. Desenvolvimento de sistemas inteligentes para controle de robos moveis. *CPGCC da UFRGS. Porto Alegre*, [S.l.], 1996.
- BRAGA, I. N. C. *Ponte-H com controle PWM (mec009) - Instituto Newton C. Braga*. Acesso: 10/05/2017, <http://www.newtonbraga.com.br/index.php/robotica/1213-ponte-h-com-pwm>.
- BRUMITT, B. L.; COULTER, R. C.; STENTZ, A. Dynamic trajectory planning for a cross-country navigator. In: APPLICATIONS IN OPTICAL SCIENCE AND ENGINEERING. *Anais...* [S.l.: s.n.], 1993. p.564–575.
- CANALTECH. *O que é API?* Acesso: 11/11/2017, <https://canaltech.com.br/software/o-que-e-api/>.
- COATES, E. *Buck converters*. Acesso: 28/04/2017, <http://www.learnabout-electronics.org/PSU/psu31.php>.
- CORKE, P. *Robotics, vision and control: fundamental algorithms in matlab*. [S.l.]: Springer, 2011. v.73.
- COWAN, A. *The Customer Discovery Handbook*. Acesso: 10/03/2017, <http://www.alexandercowan.com/customer-discovery-handbook/>.
- CRISP, J. A. *Introduction to Microprocessors and Microcontrollers*. [S.l.]: Newnes, Elsevier, 2004. v.2.
- DESOUZA, G. N.; KAK, A. C. Vision for mobile robot navigation: a survey. *IEEE transactions on pattern analysis and machine intelligence*, [S.l.], v.24, n.2, p.237–267, 2002.
- DOCTORDRONE. *Qual a utilizacao dos drones?* Acesso: 15/04/2017, <http://doctordrone.com.br/c/utilizacao/>.
- DUDEK, G.; JENKIN, M. *Computational principles of mobile robotics*. [S.l.]: Cambridge university press, 2010.

- ELEMENT14. *Raspberry Pi 3 Model B GPIO 40 Pin Block Pinout*. Acesso: 23/03/2017, <https://www.element14.com/community/docs/DOC-73950/1/raspberry-pi-3-model-b-gpio-40-pin-block-pinout>.
- ELEMENT14. *Raspberry Pi 3 Model B Technical Specifications*. Acesso: 23/03/2017, <https://www.element14.com/community/docs/DOC-80899/1/raspberry-pi-3-model-b-technical-specifications>.
- ELETROGATE. *Módulo Acelerômetro de 3 Eixos - MMA7361*. Acesso: 05/05/2017, <http://www.eletrogate.com/pd-88f58-modulo-acelerometro-de-3-eixos-mma7361.html>.
- FABMODELISMO. *Bateria de LiPo 3S 5000mAh 30C*. Acesso: 29/07/2017, <https://www.fabmodelismo.com.br/produto/pecas-e-acessorios/baterias-em-geral/6476-bateria-de-lipo-3s-5000mah-30c>.
- FAGGELLA, D. *Machine Learning in Robotics: 5 modern applications*. Acesso: 14/03/2017, <https://www.techemergence.com/machine-learning-in-robotics/>.
- FESTO. *eMotionButterflies: ultralight flying objects with collective behaviour*. Acesso: 15/05/2017, <https://www.festo.com/group/en/cms/10216.htm>.
- FREAKS, E. *Ultrasonic Ranging Module HC-SR04*. Acesso: 21/10/2017, <http://www.micropik.com/PDF/HCSR04.pdf>.
- FREESCALE. *MMA8452Q*. [S.l.]: Freescale Semiconductor, 2013. Rev. 8.1.
- G1. *Brasileiro cresce em altura nos últimos cem anos, mas ainda é 'baixinho'; conheça o ranking global*. Acesso: 21/05/2017, <https://glo.bo/2rvYwE1>.
- G1. *Carros autônomos precisam cada vez menos de humanos em testes*. Acesso: 15/05/2017, <http://g1.globo.com/carros/noticia/carros-autonomos-precisam-cada-vez-menos-de-humanos-em-testes.ghtml>.
- G1. *Nova York vai autorizar testes com veículos autônomos nas ruas*. Acesso: 15/05/2017, <http://g1.globo.com/carros/noticia/nova-york-vai-autorizar-testes-com-veiculos-autonomos-nas-ruas.ghtml>.
- GOLDMAN, J. *Street View your life: these are the 360-degree cameras coming for 2016*. Acesso: 11/04/2017, <https://www.cnet.com/news/360-degree-cameras-2016/>.
- GU, E. Y. *A journey from robot to digital human: mathematical principles and applications with matlab programming*. [S.l.]: Springer Science & Business Media, 2013. v.1.

- GUIMARÃES, F. A. *Desenvolvimento de Robô Móvel Utilizado para a Exploração de Ambientes Hostis*. 2006. Tese (Doutorado em Ciência da Computação) — Dissertação de Mestrado em Engenharia de Processos Químicos e Bioquímicos, Instituto Mauá de Tecnologia.
- HIBBELER, R. C. *Estática: mecânica para engenharia*. [S.l.]: Pearson Education do Brasil, 2005.
- HOLLAND, O. Exploration and high adventure: the legacy of grey walter. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, [S.l.], p.2085–2121, 2003.
- IEEE SPECTRUM, . *The Next Generation of Boston Dynamics' ATLAS Robot Is Quiet, Robust, and Tether Free*. Acesso: 15/05/2017, <http://spectrum.ieee.org/automaton/robotics/humanoids/next-generation-of-boston-dynamics-atlas-robot>.
- INMETRO. *Revestimentos Cerâmicos (pisos e azulejos)*. 2017.
- KACHROO, P.; MELLODGE, P. *Mobile robotic car design*. [S.l.]: McGraw-Hill Companies, Inc., 2004.
- KELLY, A. *Mobile Robotics: mathematics, models, and methods*. [S.l.]: Cambridge University Press, 2013.
- KIKUCHI, D. Y. *Sistema de controle servo visual de uma câmera pan-tilt com rastreamento de uma região de referência*. 2007. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.
- KLANCAR, Z.; BLAZIC, S. *Wheeled Mobile Robotics: from fundamentals towards autonomous systems*. [S.l.]: Butterworth-Heinemann, 2017.
- KULKA, Z. Application of pulse modulation techniques for class-D audion power amplifiers. *Archives of Acoustics*, [S.l.], v.32, 2007.
- LEENS, F. An introduction to I²C and SPI protocols. *IEEE Instrumentation & Measurement Magazine*, [S.l.], v.12, n.1, p.8–13, 2009.
- MACHADO, G. *Como buscar a inovação no mercado imobiliário?* Acesso: 14/03/2017, <http://www.guilhermemachado.com/como-buscar-a-inovacao-no-mercado-imobiliario/>.
- MAGAR, P. *Ponte H: controlando motores dc*. Acesso: 10/05/2017, <http://projetomagar.blogspot.com.br/2016/02/ponte-h.html>.

- MANFROTTO. *Compact Xtreme 2-In-1 Photo Monopod and Pole*. Acesso: 22/05/2017, <https://www.manfrotto.us/compact-xtreme-2-in-1-photo-monopod-and-pole>.
- MEIO BIT, . *Vídeo da Tesla mostra a visão de um carro autónomo ao som dos Rolling Stones*. Acesso: 15/05/2017, <http://bit.ly/2sxxgC9e>.
- MERCADO LIVRE, . *Lipo Bateria 5200mah Gens Ace Tattu Drone Aero*. Acesso: 04/06/2017, <http://bit.ly/2s5KHLw>.
- MERCADO LIVRE, . *Regulador De Tensão C/ Voltmetro - Step Down - 75w - XL4015*. Acesso: 05/05/2017, <http://bit.ly/2stDQfv>.
- MICROCHIP. *PIC18F2455/2550/4455/4550 Data Sheet*. [S.l.]: Microchip Technology Inc., 2009. DS39632E.
- MONDADA, F.; FRANZI, E.; GUIGNARD, A. The development of khepera. *Experiments with the Mini-Robot Khepera, Proceedings of the First International Khepera Workshop*, [S.l.], n.LSRO-CONF-2006-060, p.7–14, 1999.
- MOZILLA. *MDN Web docs - What is a Web server?* Acesso: 28/10/2017, https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server.
- NASA. *Mars Exploration Current Missions*. Acesso: 19/04/2017, https://www.nasa.gov/mission_pages/mars/missions/index.html.
- NASA. *The ATHLETE Rover*. Acesso: 05/05/2017, <https://www-robotics.jpl.nasa.gov/systems/system.cfm?System=111>.
- NATIONAL INSTRUMENTS, . *Medições com encoders de quadratura: guia prático*. Acesso: 10/05/2017, <http://www.ni.com/tutorial/7109/pt/>.
- NERIS, L. d. O. *Um piloto automatico para as aeronaves do projeto ARARA*. 2001. Tese (Doutorado em Ciência da Computação) — Universidade de Sao Paulo.
- NICOLOSI, D. E. C. *Microcontrolador 8051 detalhado*. [S.l.]: Érica, 2006. v.7.
- NIKU, S. B. *Introduction to robotics: analysis, systems, applications*. [S.l.]: Prentice Hall New Jersey, 2001. v.7.
- NITZAN, D. Development of intelligent robots: achievements and issues. *IEEE Journal on Robotics and Automation*, [S.l.], v.1, n.1, p.3–13, 1985.

- NXP. *Level shifting techniques in I2C-bus design*. [S.l.]: NXP Semiconductors, 2007. n.AN10441. Rev. 01.
- NXP. *MMA7361L - NXP*. [S.l.]: NXP Semiconductors, 2008.
- OCEANIC, N.; ADMINISTRATION, A. *What is the difference between an AUV and a ROV?* Acesso: 28/04/2017, <http://oceanservice.noaa.gov/facts/auv-rov.html>.
- OLIVEIRA C. NEVES E., F. J. *Competencias basicas de Internet - Formularios*. Acesso: 28/10/2017, <https://web.fe.up.pt/~goii2000/M9/cgi.htm>.
- OVERFLOW, S. *AJAX - Order of commands with XMLHttpRequest*. Acesso: 11/11/2017, <https://i.stack.imgur.com/wHxji.gif>.
- PANONO. *Panono Camera*. Acesso: 18/03/2017, <https://www.panono.com/en/product>.
- PATSKO, L. F. Tutorial Montagem da Ponte H. *Maxwell Bohr — Instrumentação Eletrônica*, [S.l.], 2006.
- POLULU. *Metal Gearmotor 25Dx56L mm LP 12V with 48 CPR Encoder*. Acesso: 04/04/2017, <https://www.pololu.com/product/3268>.
- POLULU. *Pololu 25D mm Metal Gearmotor Bracket Pair*. Acesso: 04/04/2017, <https://www.pololu.com/product/2676>.
- POLULU. *Pololu Aluminum Scooter Wheel Adapter for 4mm Shaft*. Acesso: 04/04/2017, <https://www.pololu.com/product/2672>.
- POLULU. *Scooter Wheel 100x24mm*. Acesso: 04/04/2017, <https://www.pololu.com/product/3278>.
- QUORA. *How do I calculate total acceleration from the x, y, and z g-force values given by an accelerometer?* Acesso: 25/05/2017, <http://bit.ly/2s5vKZS>.
- R7. *Brasil tem 6,3 milhões de mulheres a mais que homens*. Acesso: 21/05/2017, <http://noticias.r7.com/brasil/brasil-tem-63-milhoes-de-mulheres-a-mais-que-homens-13112015>.
- RASPBERRY PI FOUNDATION, . *Raspberry Pi Hardware*. Acesso: 23/03/2017, <https://www.raspberrypi.org/documentation/hardware/raspberrypi/README.md>.
- RICARTE, I. L. M. *Common Gateway Interface*. Acesso: 28/10/2017, <http://www.dca.fee.unicamp.br/cursos/PooJava/network/webcgi.html>.

- RICOH. *Ricoh Theta S*. Acesso: 18/03/2017, <https://theta360.com/en/about/theta/s.html>.
- RIDERS, B. *Honda Asimo*. Acesso: 02/04/2017, <http://www.bestriders.com.br/honda-asimo-o-robo-com-tecnologia-autonoma-da-honda/>.
- ROBINSON, A. *Voltage regulator: theory of operation*. Acesso: 27/04/2017, <http://sciencing.com/voltage-regulator-theory-operation-2451.html>.
- ROBOCORE. *Conversor de Nivel Logico RC*. Acesso: 25/10/2017, <https://www.robocore.net/loja/produtos/conversor-de-nivel-logico.html>.
- ROBOCORE. *Acelerometro de 3 Eixos MMA8452Q*. Acesso: 25/10/2017, <https://www.robocore.net/loja/produtos/acelerometro-de-3-eixos-mma8452q.html>.
- ROBOTICS, I. F. of. *World Robotics Report 2016*. Acesso: 10/05/2017, <https://ifr.org/ifr-press-releases/news/world-robotics-report-2016>.
- ROEBUCK, J. A.; KROEMER, K. H. E.; THOMSON, W. G. *Engineering Anthropometry Methods*. [S.l.]: Wiley-Interscience New York, 1975. v.3.
- ROUSE, M. *Common Gateway Interface (CGI)*. Acesso: 28/10/2017, <http://searchmicroservices.techtarget.com/definition/common-gateway-interface-CGI>.
- SAMSUNG. *Samsung Gear 360*. Acesso: 18/03/2017, <http://www.samsung.com/us/mobile/virtual-reality/gear-360/sm-c200nzwaxar-sm-c200nzwaxar/>.
- SAPKOTA, S. *Simple Ultrasonic Range Finder Using HC-SR04*. Acesso: 22/10/2017, <http://www.buildcircuit.com/simple-ultrasonic-range-finder-using-hc-sr04/>.
- SCHENKER, P. S. *et al.* Planetary rover developments supporting mars exploration, sample return and future human-robotic colonization. *Autonomous Robots*, [S.l.], v.14, n.2, p.103–126, 2003.
- SCHMIDT, G.; LANDIS, G.; OLESON, S. HERRO missions to Mars and venus using telerobotic surface exploration from orbit. In: AIAA SPACE 2011 CONFERENCE & EXPOSITION. *Anais...* [S.l.: s.n.], 2011. p.7343.
- SCHUNK, L. M. *Microcontroladores AVR - Teoria e Aplicações Práticas*. [S.l.]: Érica, 2001.

- SECCHI, H. A. Uma Introdução aos Robôs Móveis. *Instituto de Automática-INAUT-Universidade Nacional de San Juan-UNSJ-Argentina*, [S.l.], 2008.
- SEDRA, A. S.; SMITH, K. C. *Microelectronic circuits*. [S.l.]: New York: Oxford University Press, 1998. v.1.
- SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011.
- SIXTY, T. *Insta360 4K Review*. Acesso: 18/03/2017, <http://www.threesixtycameras.com/insta360-4k-review-livestream-360-camera/>.
- SOUZA PIO, J. L. de; CASTRO, T. H. C. de; CASTRO JÚNIOR, A. N. de. A robótica móvel como instrumento de apoio à aprendizagem de computação. *Simpósio Brasileiro de Informática na Educação-SBIE*, [S.l.], v.1, n.1, p.497-506, 2006.
- SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *Robot modeling and control*. [S.l.]: wiley New York, 2006. v.3.
- TECMUNDO. *Vídeo da Tesla mostrando um carro andando totalmente sozinho é espetacular*. Acesso: 15/05/2017, <https://www.tecmundo.com.br/tesla/110768-tesla-previa-novo-sistema-direcao-autonoma-video-fantastico.htm>.
- TECNICA, P. *Manual do especificador*. Acesso: 03/05/2017, <https://www.portobelloshop.com.br/atendimento/open/ManualdoEspecificador.pdf/4163>.
- THRUN, S. *et al.* Stanley: the robot that won the darpa grand challenge. *Journal of field Robotics*, [S.l.], v.23, n.9, p.661-692, 2006.
- UNO ROBÓTICA, . *Sensor Smart Encoder de Quadratura*. [S.l.]: UNO Robótica, 2012.
- VAN KESTEREN, A.; JACKSON, D. The xmlhttprequest object. *World Wide Web Consortium, Working Draft WD-XMLHttpRequest-20070618*, [S.l.], v.72, 2007.
- WALKER, J. *et al.* Fundamentos de física. *São Paulo: LTC*, [S.l.], 2002.
- WOW. *Buck Converter*. Acesso: 28/04/2017, http://www.wow.com/wiki/Buck_converter.