

LUÍS FERNANDO PERETTI

**MAPEADOR DE OBSTÁCULOS TRIDIMENSIONAIS POR SENSOR
ULTRASSÔNICO**

TOLEDO

2019

LUÍS FERNANDO PERETTI

MAPEADOR DE OBSTÁCULOS TRIDIMENSIONAIS POR SENSOR ULTRASSÔNICO

Projeto apresentado à disciplina de Trabalho de Conclusão de Curso 2 do Curso de Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná - UTFPR Campus Toledo, como requisito parcial para a obtenção do título de Bacharel em Engenharia Eletrônica.

Orientador: Marcos Roberto Bombacini.

TOLEDO

2019



TERMO DE APROVAÇÃO

Título do Trabalho de Conclusão de Curso Nº 116

MAPEADOR DE OBSTÁCULOS TRIDIMENSIONAIS POR SENSOR ULTRASSÔNICO

por

Luís Fernando Peretti

Esse Trabalho de Conclusão de Curso foi apresentado às 10h do **dia 6 de dezembro de 2019** como requisito parcial para a obtenção do título de Bacharel em **Engenharia Eletrônica**. Após deliberação da Banca Examinadora, composta pelos professores abaixo assinados, o trabalho foi considerado **APROVADO**.

Djones Aldivo Boni
UTFPR

Mauricio Zardo Oliveira
UTFPR

Marcos Roberto Bombacini
UTFPR

Prof. Dr. Fabio Risental Coutinho
UTFPR

O termo de aprovação assinado encontra-se na coordenação do curso.

Toledo, 6 de dezembro de 2019

RESUMO

Neste trabalho de conclusão de curso foi desenvolvido um protótipo de um sistema para reconhecer obstáculos tridimensionais através de sensor ultrassônico. O sistema deve emitir sinais por um sensor de ultrassom que, ao retornar após a reflexão, são processados pelo microcontrolador e exibidos em uma interface gráfica 3D. Neste documento, são apresentados detalhes envolvidos na construção desse sistema, desde a escolha do microcontrolador até suas respectivas implementações. Dadas as características do sensor utilizado, o sistema aplica-se a ambientes abertos e detecta obstáculos até um raio máximo de 2 metros de distância. A partir das distâncias medidas pelo sensor e de suas respectivas posições angulares, os pontos obtidos são representados em uma nuvem de pontos, permitindo interação prática com o modelo gerado. O desempenho do sistema foi testado com diferentes tipos de obstáculos, incluindo dimensões e materiais variados.

Palavras-chave: ATmega328p. Mapeador tridimensional. Ondas de rádio. Radar. Sensor ultrassônico.

ABSTRACT

In this course completion work a prototype of a system to recognize three-dimensional obstacles through ultrasonic sensor was developed. The system must output signals through an ultrasound sensor that, upon return after reflection, are processed by the microcontroller and displayed in a 3D graphical interface. This document presents details involved in the construction of this system, from the choice of the microcontroller to its respective implementations. Given the characteristics of the sensor used, the system applies to open environments and detects obstacles up to a maximum radius of 2 meters. From the distances measured by the sensor and their respective angular positions, the points obtained are represented in a point cloud, allowing practical interaction with the generated model. System performance has been tested against different types of obstacles, including varying dimensions and materials.

Palavras-chave: ATMega328p. Three-dimensional mapper. Radio waves. Radar. Ultrasonic sensor.

LISTA DE ABREVIATURAS

A/D	Analógico-Digital
CPU	<i>Central Processing Unit</i> (trad. Unidade Central de Processamento)
CA	Corrente alternada
CC	Corrente contínua
LIDAR	<i>Light Detection And Ranging</i> (trad. Detecção de luz variante)
I/O	Entrada e Saída
IDE	<i>Integrated Development Environment</i> (trad. Ambiente Integral de Desenvolvimento)
PCI	Placa de Circuito Impresso
ROM	<i>Read-Only Memory</i> (trad. Memória somente de leitura)
RISC	<i>Reduced Instruction Set Computer</i> (trad. Computador com um conjunto reduzido de instruções)
USB	<i>Universal Serial Bus</i> (trad. Porta Serial Universal)
VCC	Tensão de alimentação
GND	Tensão de referência
TDR	<i>Time-Domain Reflectometer</i> (trad. Refletômetro de domínio de tempo)

LISTA DE FIGURAS

Figura 1 – Diagrama esquemático do projeto	10
Figura 2 – Funcionamento do sensor ultrassônico	15
Figura 3 – Equivalência entre o sistema de coordenadas esféricas e o sistema de coordenadas cartesiano	18
Figura 4 – Módulo ultrassônico HC-SR04	20
Figura 5 – Diagrama de tempo de funcionamento do HC-SR04	20
Figura 6 – Servo motor utilizado no mapeador	21
Figura 7 – Sinal de controle de um servo motor	22
Figura 8 – Pinos do microcontrolador Atmega328p, da Atmel	23
Figura 9 – Módulo conversor USB RS2323 TTL CP2102	25
Figura 10 – Desenho esquemático do dispositivo mapeador	26
Figura 11 – Suporte Pan/Tilt com servo motor	27
Figura 12 – Hardware do mapeador tridimensional	28
Figura 13 – Fluxograma do software de controle	30
Figura 14 – Fluxograma do software de controle	32
Figura 15 – Placa de circuito impresso finalizada: (a) Vista superior, (b) Vista inferior.....	33
Figura 16 – Ambiente de medição sem objetos	35
Figura 17 – Resultado do mapeamento no ambiente de medição sem objetos	36
Figura 18 – Ambiente de medição com um objeto	37
Figura 19 – Resultado do mapeamento no ambiente de medição com um objeto	37
Figura 20 – Ambiente de medição com um objeto composto por líquido em seu interior	38
Figura 21 – Resultado do mapeamento no ambiente de medição com um objeto composto por líquido em seu interior	39
Figura 22 – Ambiente de medição com dois objetos presentes	40

Figura 23 – Resultado do mapeamento no ambiente de medição com dois objetos presentes.....	40
Figura 24 – Canto entre duas paredes perpendiculares	41
Figura 25 – Resultado do mapeamento realizado entre duas paredes perpendiculares	41
Figura 26 – Representação de uma situação recorrente na medição utilizando o sensor ultrassônico	43
Figura 27 – Esquema elétrico do sensor de ultrassom HC-SR04	53
Figura 28 – Diagrama de blocos do microcontrolador ATmega328P	54
Figura 29 – Esquema elétrico da placa de circuito impresso	55
Figura 30 – Vista das trilhas do circuito na superfície inferior da placa	56
Figura 31 – Vista da disposição dos componentes na superfície superior da placa	56

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Problema	10
1.2	Justificativa	11
1.3	Objetivos	12
2	REFERENCIAL TEÓRICO	13
1.1	Características de mapeamento tridimensional.....	13
1.1.1	Produção da onda sonora e recepção dos ecos.....	13
2.1	Sensores ultrassônicos	14
2.2	Influência dos materiais nos ensaios de ultrassom	15
2.3	Transdutores ultrassônicos.....	16
2.4	Nuvem de pontos	17
2.5	Transformação de coordenadas	18
3	MATERIAIS E MÉTODOS	19
3.1	Sensor ultrassônico	19
3.2	Servo motor.....	21
3.3	Microcontrolador	22
3.4	Arduino Uno	23
3.5	Interface gráfica	24
3.6	Módulo conversor	24
3.7	Metodologia	25
4	PROJETO DO MAPEADOR.....	27
4.1	Montagem do hardware	27
4.2	Software de controle	28
4.3	Software de interface	30
4.4	Placa de circuito impresso	33

5	RESULTADOS E DISCUSSÕES	35
5.1	Ambiente delimitado sem obstáculo presente.....	35
5.2	Ambiente delimitado com um obstáculo presente	36
5.3	Ambiente delimitado com um obstáculo composto de dois materiais.....	38
5.4	Ambiente delimitado com dois obstáculos presentes	39
5.5	Canto entre duas paredes	40
5.6	Interpretação dos testes	42
6	CONCLUSÃO	45
6.1	Dificuldades encontradas	45
6.2	Sugestões para trabalhos futuros.....	46
7	REFERÊNCIAS	47
	ANEXO A – Esquemático do sensor de ultrassom	53
	ANEXO B – Diagrama de blocos do microcontrolador.....	54
	APÊNDICE A – Esquemático do sistema	55
	APÊNDICE B – <i>Layout</i> PCI	56
	APÊNDICE C – Valor investido no projeto.....	57
	APÊNDICE D – Algoritmo de controle	58
	APÊNDICE E – Algoritmo de interface.....	61

1 INTRODUÇÃO

O termo ultrassom é empregado para definir ondas acústicas com frequência superior à capacidade audível do ser humano, ou seja, acima 20.000 ciclos por segundo (HELERBROCK, 2019). Também conhecidas como sinais ultrassônicos, as ondas acústicas inaudíveis são captadas por meio de transdutores, os quais, em função da ressonância, convertem energia acústica em elétrica e vice-versa (THOMAZINI, 2005).

Esta seção apresenta os recursos necessários para a realização das atividades práticas, a relevância que a eletrônica e a computação possuem para o desenvolvimento tecnológico e, por fim, a importância de tais recursos no desenvolvimento acadêmico e industrial. Adicionalmente, considera-se as atuais limitações envolvidas no decorrer do trabalho, bem como seus benefícios. Dessa forma, tem-se como objeto de pesquisa um sistema embarcado com foco no tratamento de dados, processo que é composto por um *hardware* e *software* (estrutura física e digital de um sistema computacional, respectivamente). O primeiro possui um sensor e dois atuadores e a sua finalidade é coletar esses dados em um ambiente aberto. O segundo é uma interface gráfica capaz de interpretar e exibir na tela do computador o que foi lido pelo *hardware*, através de um sistema microcontrolado. Neste trabalho, foi utilizado um sensor de ultrassom, dois servos motores, um microcontrolador e plataformas gratuitas de desenvolvimento, sendo elas o Ambiente de Desenvolvimento Integrado (do inglês *Integrated Development Environment - IDE*) do Arduino e a interface gráfica *Processing 3*. Ao final, foi confeccionada uma placa de circuito impresso (PCI), cujos componentes eletrônicos serão descritos nas seções seguintes, projetada na plataforma online *EasyEDA*.

No início do século XX, as nações perceberam o potencial que um sistema de radar poderia possuir no cenário bélico. Essa necessidade, junto com estudos da época, fez com que já na segunda Guerra Mundial houvesse uma evolução muito rápida dessa tecnologia. Um radar é um sistema direcional de ondas de rádio que pode determinar o alcance, direção, altitude e velocidade de objetos e a sua aplicação está diretamente associada para fins de meteorologia, vigilância costeira, controle de tráfego aéreo comercial, indústria aeroespacial e até mesmo em aplicações mais cotidianas, como por exemplo no monitoramento de velocidade de veículos no trânsito (BRAGA, 1995).

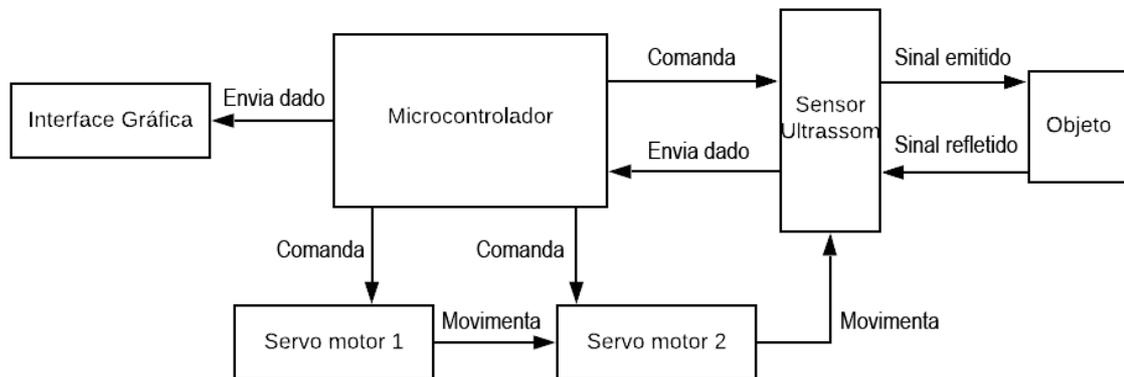
Com o acesso tecnológico cada vez mais disseminado pelo mundo de forma mais democrática e qualitativa, estudos como o que foi citado no parágrafo anterior, que antes eram desenvolvidos apenas em ambientes industriais ou em pesquisas acadêmicas, estão se tornando atividades mais populares, ou seja, mais pessoas têm acesso e interesse por essas atividades. Neste contexto, problemas clássicos de engenharia e computação começam a ser desafiados por um número maior de pessoas e, assim, havendo maior possibilidade para novas soluções (DE FREITAS *et al.* 2013).

Apesar de um vasto tipo de sistemas de radares existentes no mercado, a sua aplicação é definida de forma muito distinta, entre as quais pode-se citar: o mini radar da Google, designado para gestão de vias de trânsito urbano e atualização no computador do usuário; o XSight, um radar de emissão de ondas milimétricas utilizado no tráfego aéreo para identificar restos de objetos estranhos e; o MESA-K-DEV, dispositivo utilizado em diversas aplicações de veículos autônomos. Dessa forma, é possível identificar que o conceito de “radar” abrange mais do que dispositivos que emitem e recebem ondas de rádio. Aqui, pode-se citar como exemplo o sistema *LIDAR* (da sigla inglesa *Light Detection And Ranging*) que faz uso da tecnologia óptica e também o sistema de detecção através de ondas ultrassônicas, este que está presente neste trabalho (BASTOS; ERCOLIN FILHO, 2015).

Para mapear um ambiente de forma eficiente, é preciso coletar dados a respeito desse meio e processá-los de tal forma que gerem uma representação do mundo real no computador. Pensando nisso, o sistema eletrônico que foi proposto possui funcionamento para a leitura de um ambiente análogo ao de uma matriz: o servo motor responsável pelo movimento horizontal faz o sensor ultrassônico coletar dados ao longo de uma abertura; terminada essa varredura, o servo motor responsável pelo movimento vertical move-se um pouco para baixo e a varredura horizontal é realizada novamente. O processo se repete até que o servo motor vertical chegue ao seu limite máximo estabelecido.

Como pode ser visto no diagrama esquemático da Figura 1, o sistema eletrônico é formado por um sensor de ultrassom acoplado a um servo motor (Servo motor 2) responsável pelo movimento vertical. Este servo motor é também acoplado a outro servo (Servo motor 1), responsável pelo movimento horizontal. Ambos são comandados pelo microcontrolador, que monitora o movimento dos servos, bem como envia e recebe pulsos de sinais através do sensor de ultrassônico. Ao final, os dados são processados na interface gráfica e podem ser visualizados tridimensionalmente.

Figura 1 – Diagrama esquemático do projeto.



Fonte: Autoria própria.

Por conta de interferências externas, o mapeador pode ser mais ou menos preciso em sua representação gráfica. Parte dessa incerteza deve-se ao fato de o sensor ultrassônico emitir suas ondas na forma cônica (o que será abordado com mais adiante). De forma complementar, para maior eficiência, o objeto a ser mapeado deveria estar presente em ambiente totalmente aberto, a fim de que fosse evitado as inúmeras refrações causadas pelo ambiente, o que não foi possível realizar. Sobre os servos motores, existe também o fenômeno de trepidação ao se deslocarem, algo que influencia diretamente na leitura do sensor de ultrassom. Todavia, mesmo com distorções, é possível identificar os obstáculos desejados e, uma vez que os dados são exibidos em uma interface gráfica em três dimensões (3D), em trabalhos futuros fica plausível uma melhoria no algoritmo e nos próprios elementos físicos que compõe o dispositivo.

1.1 Problema

Os principais pilares da engenharia baseiam-se no estudo teórico, experimental e, posteriormente, no aperfeiçoamento de uma determinada tecnologia. Nesse cenário, o conhecimento adquirido ao longo dos anos fez com que a era dos mecanismos autônomos chegasse. Não se trata mais de “se” ou “quando”, a discussão agora se concentra em “onde” e “em que medida”, esse desenvolvimento estará presente em tecnologias que já existem, a fim

de otimiza-las, como por exemplo em máquinas industriais, carros autônomos ou tecnologia espacial (DE FREITAS *et al.* 2013).

Mapear um ambiente tridimensional de forma eficiente ao levantar a topologia detalhada do local ou marcar pontos específicos que sirvam de referência para algum sistema de navegação, pode ser útil em uma gama de aplicações como engenharia eletrônica, engenharia mecatrônica, engenharia civil, arquitetura e projetos de plantas industriais. Apesar disso, dependendo do fim dessa atividade, a tarefa pode ser mais simples ou mais complexa (DE FREITAS *et al.* 2013).

De modo geral, é necessário coletar dados a respeito do ambiente e processá-los de tal forma que gerem uma representação real, ou pelo menos o mais fiel possível, para que o sistema a qual esse mapeamento está empregado consiga executar tarefas baseadas nesse reconhecimento.

O primeiro passo, para coletar esses dados, é fixar um sensor em um ponto central e medir as posições geográficas de cada ponto, seja de um objeto ou de um ambiente, em relação a esse ponto fixo. Uma vez que esse sensor possa se mover em relação ao seu próprio eixo, como é o caso deste projeto, cada leitura efetuada possui três coordenadas: ângulo horizontal, ângulo vertical e distância medida. Essa leitura bruta é chamada de *point cloud*, em português “nuvem de pontos”, isso quer dizer, um conjunto de pontos disposto espacialmente em um sistema de coordenadas onde cada ponto coletado refere-se a uma superfície do ambiente real.

1.2 Justificativa

Um dos principais propósitos ao desenvolver um sistema de mapeamento em três dimensões é a sua aplicação em possíveis automações, mais especificamente em robótica. O Laboratório de Sistemas Inteligentes (LASI), localizado na Universidade de São Paulo, campus São Carlos, desenvolve pesquisa em automação e sistemas de controle, filtragem, aprendizado de máquinas e robótica.

Nesse laboratório, é realizado trabalhos voltados para controle de robôs autônomos, entre eles o *Pioneer 3-AT*, uma plataforma robótica móvel de quatro rodas, com sensores, computadores de bordo e software de navegação autônoma. Esse robô pode se utilizar de um mapa predefinido para se localizar durante a navegação, como também pode fazer a leitura de

um ambiente e gerar o próprio mapa, de forma que o seu reconhecimento do território fique cada vez mais eficiente. Portanto, a motivação inicial do trabalho surgiu da possibilidade de futuramente integrar um sistema de mapeamento 3D à projetos similares desenvolvidos na própria no próprio campus da UTFPR-TD.

1.3 Objetivos

O presente trabalho tem como objetivo desenvolver um sistema embarcado de leitura tridimensional, com a finalidade de identificar obstáculos ao seu redor. Como objetivos específicos, tem-se:

- a) Realizar a leitura de um ambiente com alcance de 180 graus na horizontal, 50 graus vertical e alcance de 2 metros; e
- b) Projetar os dados gerados pelo microcontrolador em uma interface gráfica tridimensional para visualização do objeto mapeado.

2 REFERENCIAL TEÓRICO

Nesta seção serão descritos aspectos teóricos a respeito da propagação das ondas sonoras. Complementarmente, será retratado a influência dos diferentes tipos de materiais nos ensaios com ondas de ultrassom, bem como o funcionamento de um transdutor ultrassônico. Enfim, as equações essenciais que modelam o seu funcionamento.

1.1 Características de mapeamento tridimensional

A detecção por ultrassom ocorre através de um pulso curto de energia sonora que é emitido pelo transdutor. O pulso viaja na velocidade do som e, ao detectar um obstáculo, parte do pulso é refletido como um eco que volta para a fonte. Armazenando cada um dos pontos ao longo do tempo, em cada uma das respectivas posições angulares comandadas pelo sistema microcontrolado, é possível obter informações sobre o obstáculo presente (THOMAZINI, 2005).

1.1.1 Produção da onda sonora e recepção dos ecos

Uma tensão oscilando com alta frequência aplicada através do material piezoelétrico, presente no sensor de ultrassom, cria uma onda sonora com a mesma frequência. A medida dessa tensão fornece uma maneira de registrar ondas ultrassônicas. O mesmo material piezoelétrico pode funcionar tanto como fonte quanto como receptor (ULTRASSONOGRAFIA, 2017).

O som é direcionado pelo formato do transdutor. Há um intervalo de alguns milissegundos entre a produção de um pulso de onda ultrassônica. Durante esses intervalos, há captação das ondas que são refletidas. De forma oposta, isto é, ao receber os estímulos mecânicos de volta, os cristais piezoelétrico vibram e geram uma diferença de potencial elétrico, causando impulsos elétricos (ULTRASSONOGRAFIA, 2017). Estes constituem um sinal elétrico, que é lido e interpretado pelo computador, onde o dispositivo mapeador está conectado através da porta USB. No mapeamento tridimensional, a direção do feixe é

rapidamente alterada para cobrir toda região onde o obstáculo está presente. Essa cobertura é realizada por servo motores, comandados pelo microcontrolador.

2.1 Sensores ultrassônicos

O termo ultrassom é empregado para definir ondas acústicas com frequência superior à capacidade audível do ser humano, ou seja, acima 20.000 Hertz (Webster, 1999). Também conhecidas como sinais ultrassônicos, as ondas acústicas inaudíveis são captadas por meio de transdutores, os quais, em função da ressonância, convertem energia acústica em elétrica e vice-versa (THOMAZINI, 2005).

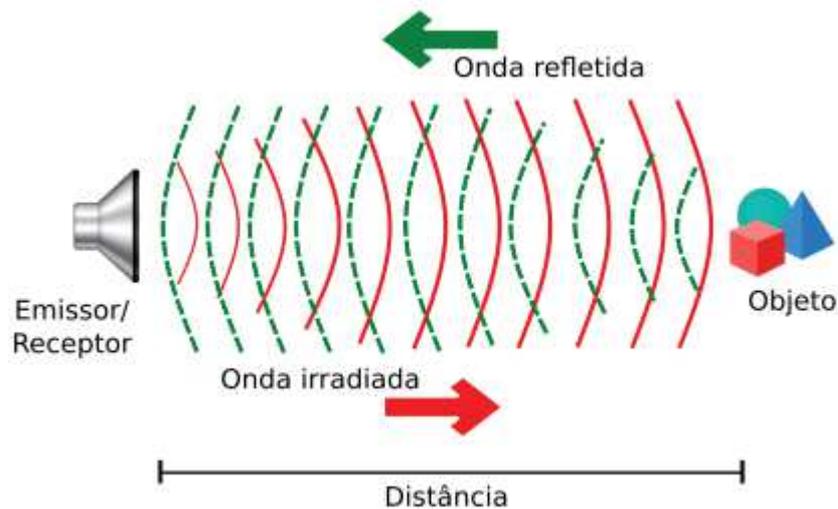
Sensores ultrassônicos são frequentemente utilizados na medição de distâncias e detecção de obstáculos, sejam em estado físico líquido ou sólido. Estes sensores são capazes de detectar objetos, tais como metal, madeira, concreto, plásticos, produto à base de borracha, e objetos transparentes. Diferente dos sensores infravermelho, sensores ultrassônicos não necessitam de informações acerca das propriedades das superfícies dos objetos (THOMAZINI, 2005).

Os sensores ultrassônicos são constituídos a partir de dispositivos eletrônicos chamados transdutores ultrassônicos. Os transdutores são utilizados para emitir e absorver pulsos ultrassônicos em determinada frequência. O funcionamento do sensor ultrassônico é ilustrado na Figura 2. O processo de medição inicia quando o sensor irradia um sinal de pulso sonoro na direção do objeto. Como resposta a este estímulo sonoro, o sensor recebe um sinal referente a reflexão desta onda na superfície do objeto (THOMAZINI, 2005). Quando um sinal com a mesma frequência de pulsos emitida é recebido, é possível calcular a distância até um dado objeto a partir do tempo que os pulsos levaram a ser refletido de volta pelo objeto.

Fatores como temperatura, umidade e presença de ruído no ambiente, interferem na velocidade de propagação dos pulsos ultrassônicos, provocando retardos no sinal de reflexão dos pulsos, causando medições imprecisas (MOHAMMAD, 2009). Outro fator frequentemente observado nestas situações refere-se ao fato de que a resposta ao sensor não é, precisamente, o sinal de reflexão do objeto no qual se observa, já que a presença de ruídos no ambiente pode causar um eco de falso-positivo (CHAKRABARTY; LOVEKAR, 2017).

A distância detectada por um sistema de mapeamento tridimensional é baseada no tempo de propagação de um pulso ultrassônico. Para medir esta distância, o transdutor emite um sinal em uma frequência específica. Então, o módulo de alcance mede o tempo que leva para o pulso sônico refletido retornar ao transmissor. Ainda, o tempo total de medição é estendido pela latência eletrônica (isto é, o tempo necessário para iniciar e interromper a vibração da membrana do componente oscilador) e pelo processamento computacional. Por causa disso, um erro relativamente típico de medição especificado pelos fabricantes de sistemas de medição sonoros é em torno de 1% (ANDREEV; TARASOVA, 2018).

Figura 2 – Funcionamento do sensor ultrassônico.



Fonte: Blog Autocore (Adaptado, 2017).

2.2 Influência dos materiais nos ensaios de ultrassom

A propagação de ondas ultrassônicas está sempre relacionada aos módulos de rigidez (G) e elasticidade (E) dos materiais em que tais ondas se propagam, uma vez que eles determinam a capacidade das partículas resistirem à deformação elástica, causada pelo pulso ultrassônico, respectivamente nas direções transversal e longitudinal. Assim, o coeficiente de *Poisson* (nome dado em homenagem ao a Siméon Denis Poisson, quem o formulou), que determina a deformação transversal, resultante da aplicação de uma carga na direção longitudinal (MARTIN, 2012).

Uma vez que as características físicas do material são compiladas, torna-se mais fácil notar as diferenças em relação às velocidades de propagação de ondas entre polímeros, metais e demais materiais. Ainda, as diferenças podem aumentar ou diminuir dependendo da temperatura de trabalho. A cada 50° C, de diferença, os metais costumam perder cerca de 1% de velocidade de propagação sônica, já nos polímeros essa perda pode subir em até 4% (MARTIN, 2012).

A grande diferença de velocidade de ondas mecânicas, entre polímeros e metais, e diferentes suscetibilidades dessas velocidades com a mudança de temperatura, traz duas barreiras à aplicação de ultrassom em componentes constituídos por camadas desses materiais (MARTIN, 2012).

A primeira barreira está relacionada aos coeficientes de transmissão de ondas mecânicas na interface desses materiais. Quanto maior a diferença entre as velocidades longitudinais dos materiais, menor será a quantidade de intensidade sonora que será transmitido pela interface (MARTIN, 2012).

A segunda barreira é que ao atravessar a interface de materiais com grande diferença de velocidade acústica não somente o feixe sônico perde energia, como também é distorcido o seu ângulo de abertura. Um sistema de ultrassom que lide com camadas sobrepostas de metais e polímeros tem de considerar faixas de frequências onde essas distorções possam ser conhecidas e controladas. No entanto, essa segunda não será abordada no trabalho.

2.3 Transdutores ultrassônicos

Os transdutores ultrassônicos utilizam a deformação dos cristais piezelétricos tanto para gerar energia sônica no elemento examinado, quanto para coletar o restante dessa energia que atravessou o elemento examinado e não foi dissipado por esse.

De forma específica a este trabalho, a técnica de medição utilizada é denominada como pulso-eco. Essa técnica depende do nível de precisão da medição e da possibilidade de acesso ao material inspecionado, podendo ser:

- a) Transdutores exclusivos de emissão, ou seja, esses transdutores somente excitarão o material inspecionado, transferindo energia sônica a este;

- b) Transdutores exclusivos de recepção, ou seja, esses transdutores somente serão excitados por ondas sonoras que atravessam o material examinado ou que foram refletidos por esse. Nesse caso, encontra-se na mesma superfície do transdutor emissor; e
- c) Transdutores de cristal piezelétrico duplo, os quais possuem dois cristais piezelétricos montados em uma carcaça, configurando o mesmo ponto de emissão e recepção (MARTIN, 2012).

Todos os transdutores podem ser excitados em uma frequência determinada pelo equipamento de ultrassom ou simplesmente serem deixados em contato com a peça examinada, à espera de ondas sonoras refratadas ou refletidas pelo material que está em contato. Dessa forma, ainda é possível uma montagem adicional às citadas anteriormente, onde um transdutor de cristal piezelétrico único é excitado pelo aparelho de ultrassom por um determinado tempo e deixado em repouso em um intervalo programado seguinte. Dessa forma o transdutor é capaz de realizar ciclos combinados de emissão e de ondas sonoras e leitura de seu eco na mesma posição (BROOK, 2012). Em particular a este trabalho, essa é a forma de operação definida pelo sensor de ultrassom escolhido, cujo circuito pode ser visto em Apêndice A, na Figura 27.

2.4 Nuvem de pontos

Uma nuvem de pontos é um conjunto de pontos organizados em um mesmo sistema de coordenadas. Em um sistema de três dimensões, geralmente definidos em coordenadas cartesianas, esses pontos representam a superfície externa de um objeto (NUVEM DE PONTOS, 2019).

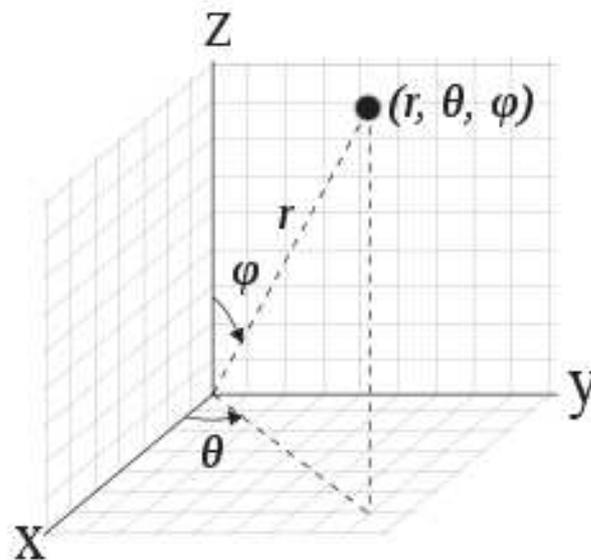
Nuvens de pontos podem ser criadas por *scanner* 3D através da medição de um número significativo de pontos para a representar aquele objeto de interesse (NUVEM DE PONTOS, 2019).

Produtos da digitalização tridimensional, as nuvens de pontos são utilizadas para vários fins, como por exemplo, modelos CAD (desenho assistido por computador, do inglês: *Computer Aided Design*) em 3D, inspeção de qualidade, metrologia e renderizações personalizadas de aplicativos (NUVEM DE PONTOS, 2019).

2.5 Transformação de coordenadas

Analisando os dados gerados pelo sistema, nota-se que os pontos obtidos possuem representação em coordenadas esféricas, isso porque suas posições são representadas por uma distância em relação a origem (r), um ângulo de deslocamento horizontal (θ) e um ângulo de deslocamento vertical (φ), como mostra a Figura 3 (WEISSTEIN, 2019).

Figura 3 – Equivalência entre o sistema de coordenadas esféricas e o sistema de coordenadas cartesiano.



Fonte: Adaptado (WEISSTEIN, 2019).

Levando em conta que a interface gráfica plota a leitura do mapeador em coordenadas cartesianas, antes de armazenar cada um dos pontos lidos na nuvem de dados é necessário convertê-los. Para isso, usa-se as seguintes equações:

$$x = r \cos(\theta) \sin(\varphi) \quad (1)$$

$$y = r \sin(\theta) \sin(\varphi) \quad (2)$$

$$z = r \cos(\varphi) \quad (3)$$

3 MATERIAIS E MÉTODOS

Este capítulo apresenta a metodologia empregada no projeto do mapeador tridimensional descrito no referencial teórico, assim como a descrição dos materiais necessários para a sua realização. Ainda, é descrito de forma detalhada os demais elementos que compõem o sistema na confecção da placa de circuito impresso.

3.1 Sensor ultrassônico

O dispositivo utilizado foi o módulo HC-SR04, exibido na Figura 4. Um pulso ultrassônico é emitido pelo dispositivo e ao ser refletido pelo objeto, retorna para o sonar. Com base no tempo decorrido entre a emissão e a recepção deste pulso, é possível determinar a distância do objeto, com base na Equação 4.

$$d = \frac{vt}{2} \quad (4)$$

Onde,

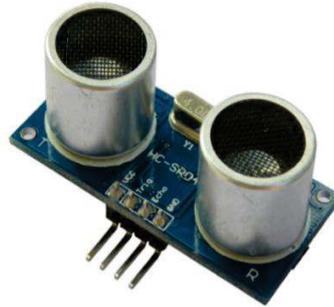
d: distância do sensor ultrassônico até o obstáculo;

v: velocidade do som; e

t: tempo de resposta.

Esse dispositivo é composto por um emissor e um receptor, capaz de medir distâncias de 2 centímetros até 4 metros, com precisão aproximada de 3 milímetros e com ângulo de visão de 15°. Ademais, o sensor HC-SR04 também possui em seu acoplamento um circuito pronto e 4 pinos: Alimentação, *Trigger* (emissor de pulso), *Echo* (receptor de pulso) e Referência.

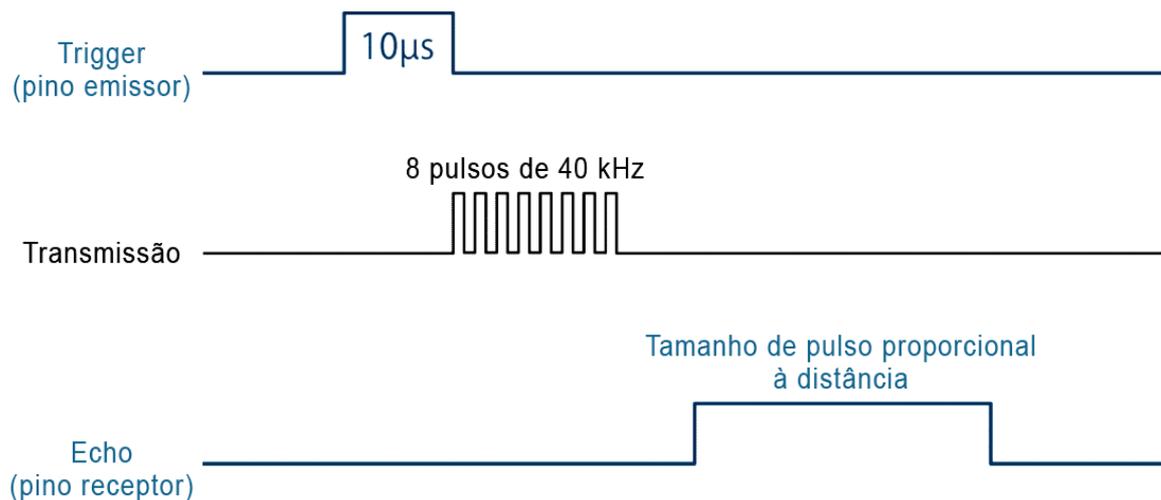
Figura 4 – Módulo HC-SR04.



Fonte: Cytron Technologies (2019).

Tendo como base o diagrama de tempo do funcionamento do sensor de ultrassom, ilustrado na Figura 5, ao receber um sinal de 10 microssegundos em uma tensão de 5 volts no pino emissor, o módulo envia oito pulsos de ultrassom a uma frequência de 40 quilo Hertz. Posteriormente, o sensor capta a reflexão do som que, após ser processada pelo circuito controlador, gera um sinal de resposta no pino receptor. A duração dessa resposta em 5 volts é diretamente proporcional à distância medida.

Figura 5 – Diagrama de tempo de funcionamento do HC-SR04.



Fonte: Adaptado (Cytron Technologies, 2019).

Como foi abordado no referencial teórico, aspectos físicos dos materiais inspecionados podem influenciar na reflexão do som emitido na direção do alvo de interesse. Entretanto, por se tratar de um sistema de inúmeros pontos analisados, algumas medidas discrepantes não alteram de forma significativa o mapeamento. Ademais, pretende-se trabalhar com objetos cuja característica de composição é conhecida.

3.2 Servo motor

O servo motor ilustrado na Figura 6 é um atuador rotativo constituído por um motor acoplado a engrenagens com função de redução, junto a um circuito projetado especificamente para servos motores. Com isso, é possível obter um controle muito preciso de sua posição angular, velocidade e aceleração (DUTT, 2014).

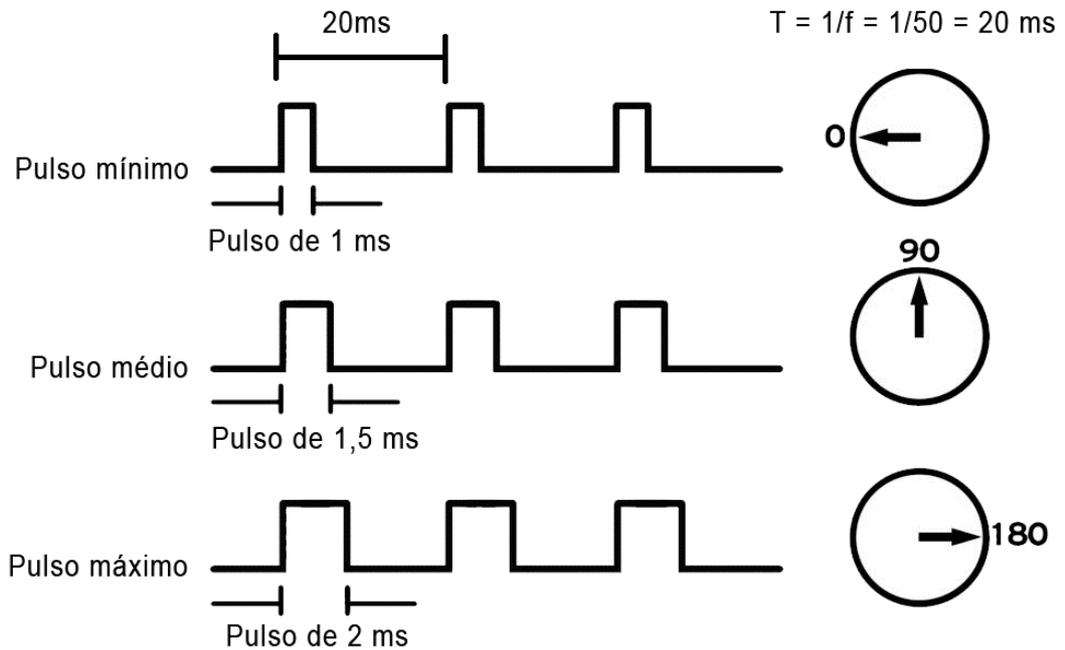
Figura 6 – Servo motor utilizado no mapeador.



Fonte: Tower Pro (2019).

Esse controle é realizado através de pulsos de onda retangular modulada em PWM, onde a largura desses pulsos controla a posição angular do servo motor. Segundo especificações do *datasheet*, o servo motor requer uma frequência de operação de 50 Hertz e a largura de pulso deve variar de 1 milissegundos até 2 milissegundos, fazendo com que o ângulo varie 180°, conforme a Figura 7.

Figura 7 – Sinal de controle de um servo motor.

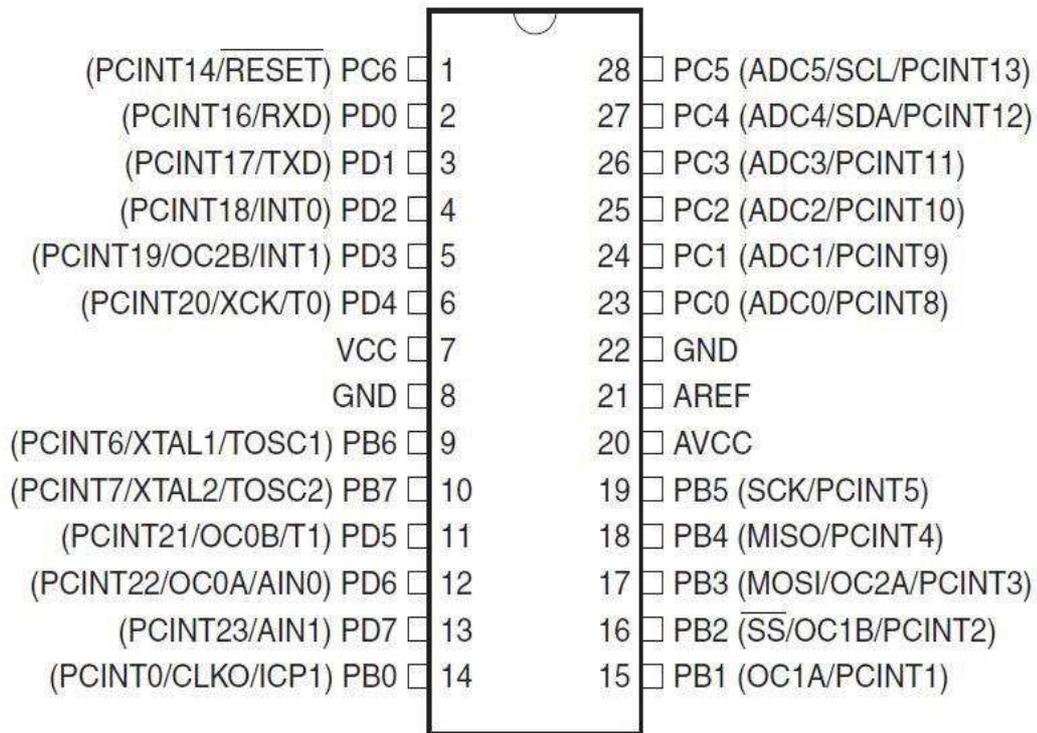


Fonte: Adaptado (Microcontroller Projects, 2019).

3.3 Microcontrolador

Os microcontroladores são produzidos na forma de circuitos integrados e possuem pinos dedicados à alimentação, à gravação de dados em sua memória *ROM* e aos periféricos de entrada e saída. O microcontrolador utilizado no projeto é o ATMega328p (Figura 8), que utiliza arquitetura Harvard de 8 bits baseada em *RISC*, 32 KB de memória flash, 1 KB de EEPROM, 32 registradores de uso geral, 3 temporizadores, uma USART, portas para comunicação SPI, 6 conversores A/D de 10 bits e um *watchdog timer*, entre outras características que podem ser visualizadas em Apêndice B (CAPELLI, 2014). Para a implementar o mapeador foram necessárias 4 portas digitais dedicadas para movimentar os servos motores e os pulsos de emissão e recepção do sensor de ultrassom. Ainda, foi preciso utilizar os pinos respectivos a comunicação serial (RXD e TXD). Levando isso em conta, conclui-se que este dispositivo possui parâmetros suficientes para o controle do mapeador.

Figura 8 – Pinos do microcontrolador Atmega328p, da Atmel.



Fonte: Atmel Corporation (2019).

A função do microcontrolador se deve a sua Unidade de Central Processamento (CPU), ou seja, este é o responsável por conter e executar as instruções necessárias do sistema. Em outras palavras, o microcontrolador executa as informações previamente programadas em sua memória de programa, movimentando os servos motores, que por sua vez contém o sensor ultrassônico acoplado. O pulso recebido de volta pelo sensor é processado pelo microcontrolador e enviado para a interface gráfica através da porta serial.

3.4 Arduino Uno

O Arduino UNO é uma placa de desenvolvimento *open hardware* (plataforma física de livre desenvolvimento, no termo traduzido) baseada no microcontrolador ATmega328p, que possui uma própria interface de desenvolvimento, onde foi elaborado parte do código responsável pelo controle dos periféricos. Seu uso no projeto foi empregado nas fases de testes por conta de sua praticidade e de uma extensa bibliografia disponível. Devido ao seu ressonador

cerâmico interno de 16 Megahertz, pode-se obter respostas satisfatórias no que se diz respeito à aquisição de dados pelo sensor de ultrassom. Para a alimentação basta conectá-lo ao computador com um cabo USB, liga-lo a um adaptador CA-CC ou a uma bateria. Uma vez que o mapeador deve ficar conectado ao computador para envio de dados pela porta serial, a alimentação também foi feita diretamente pela conexão USB.

3.5 Interface gráfica

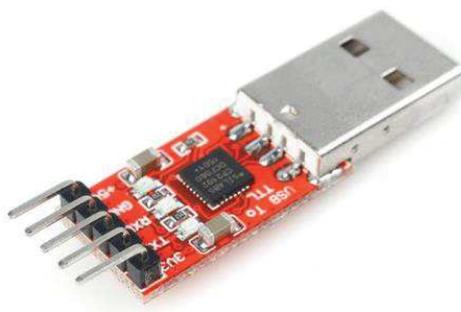
Processing é uma linguagem de programação também *open source*, construído para artes eletrônicas e comunidade de projetos visuais. O projeto foi iniciado em 2011, por Casey Reas e Ben Fry, ambos ex-membros do grupo de computação do *MIT Media Lab*. Seu desenvolvimento foi baseado na ideia de ensinar contextos de programação de forma visual. Por conta disso, o *software* gera imagens através de comandos pré-estabelecidos, podendo ter a função de uma interface gráfica (DUTT, 2014).

A cada leitura realizada pelo microcontrolador, é enviado para a interface gráfica através da porta serial uma *String* (sequência de caracteres) contendo valores respectivos à distância, ângulo do servo motor horizontal e ângulo do servo motor vertical. Esses valores, correspondentes à um ponto em coordenadas esféricas, são convertidos em dados inteiros e depois em coordenadas cartesianas. Os dados são armazenados em vetores até que o mapeamento se encerre. Ao final, a nuvem de pontos gerada é exibida em uma interface gráfica tridimensional.

3.6 Módulo conversor

O módulo conversor é utilizado para comunicação entre o computador e dispositivos como microcontroladores, módulos e outros equipamentos que utilizam comunicação via interface serial. Para a montagem da PCI, foi utilizado o módulo conversor USB RS-232, de 5 pinos (Figura 9). Esse dispositivo permite que a conexão entre o microcontrolador e o computador seja feito de forma simplificada, pelos pinos RXD, TXD e GND.

Figura 9 – Módulo conversor USB RS2323 TTL CP2102.



Fonte: Vida de silício.

Como especificações no *datasheet*, o seu chip de controle trabalha com *baud rate* (taxa de transferência) de 300 quilobit por segundo até 1 megabit por segundo. Uma vez que a taxa de transmissão de dados do mapeador requer 115200 quilobit por segundo, o módulo corresponde às necessidades.

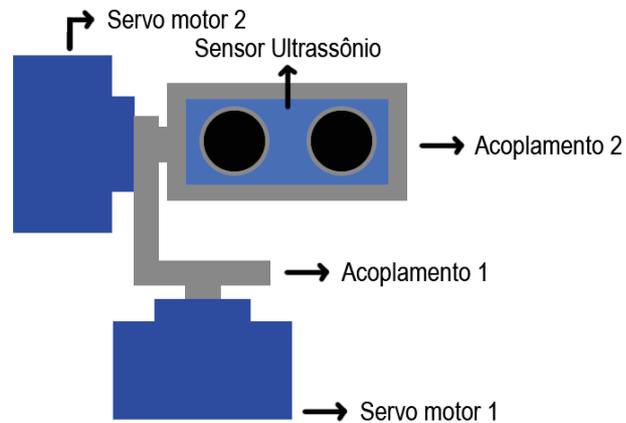
3.7 Metodologia

Para determinar a forma dos obstáculos presentes, propõe-se o uso de deslocamentos angulares para realizar a varredura do sensor de ultrassom. A saída de informações deste sensor é conectada ao microcontrolador, que também controla os servos, cuja duração de pulso formado pelo sensor é convertida em uma distância, medida em centímetros. Junto a isso, o microcontrolador está conectado ao computador através do barramento USB.

O sinal emitido pelo microcontrolador dá funcionamento ao software, que altera a orientação do sensor ultrassônico no espaço com precisão de 2° na horizontal. O computador envia um sinal para o microcontrolador, mudando discretamente o passo do sensor ultrassônico. Entre cada uma das rotações, o sensor mede a distância do ponto obstáculo à sua frente. Dessa forma, é possível determinar a largura do objeto detectado.

Ao final de cada linha obtida, respectiva a largura do objeto, os sensores movimentam o sensor com um passo de 5°, no sentido vertical. De forma análoga ao método do parágrafo anterior, as colunas são formadas com as distâncias correspondentes ao ângulo de rotação do dispositivo (Figura10). Dessa forma, mede-se a altura do obstáculo estudado.

Figura 10 – Desenho esquemático do dispositivo mapeador.



Fonte: Autoria própria.

Para cada uma de todas as possíveis posições são gravadas três informações: a posição angular de cada servo motor, bem como a distância medida em relação ao obstáculo. Associando esses três dados a cada ponto no espaço, pode-se mapear o ambiente através de coordenadas esféricas.

O movimento do sistema e a aquisição dos dados respectivos a cada ponto é feita de forma paralela. Cada um dos pontos obtidos pela porta serial é convertido em coordenadas cartesianas e armazenado em um vetor de três dimensões. Terminada a leitura, o conjunto de pontos é plotado na tela do computador.

4 PROJETO DO MAPEADOR

Esta seção apresenta o projeto do mapeador, segundo a fundamentação teórica, materiais e métodos descritos nos capítulos 2 e 3. As tarefas aqui envolvidas foram divididas em montagem do hardware, software de controle, software de interface e, por fim, na confecção da placa de circuito impresso.

4.1 Montagem do hardware

Preliminarmente, foi proposto que o mapeador devesse percorrer dois eixos do plano tridimensional, para que o terceiro plano fosse destinado ao sensor de ultrassom. Baseado no suporte Pan/Tilt ilustrado na Figura 11, onde um servo motor fica acoplado ao outro, a montagem física do dispositivo foi feita de forma similar.

Figura 11 – Suporte Pan/Tilt com servo motor.



Fonte: Felipe Flop.

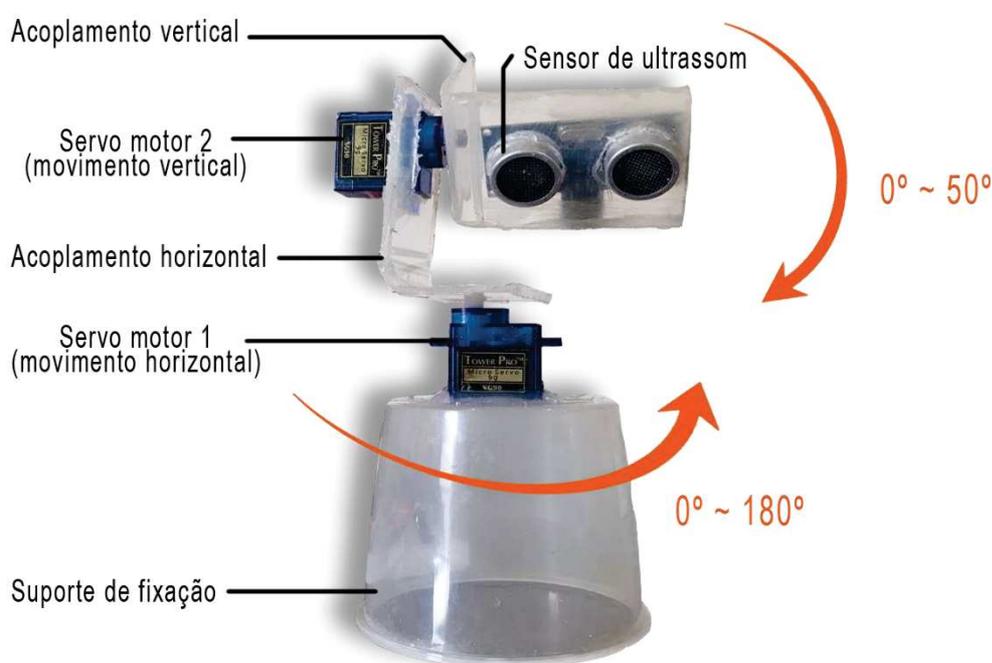
Como pode ser visto na imagem a cima, ao rotacionar, o servo motor inferior movimenta o servo motor superior ao longo do plano horizontal. Da mesma forma, o servo motor superior movimenta o suporte de acoplamento por toda a extensão vertical.

Tendo como base a Figura 12, o comportamento desejado foi obtido através da confecção de duas chapas de plástico rígido em formato de “L”. O acoplamento horizontal foi

fixado ao eixo do servo motor 1 e ao servo motor 2, movimento o segundo servo de 0° a 180° . De forma parecida, o acoplamento vertical foi fixado ao eixo do servo motor 2 e ao sensor de ultrassom, movimentando-o de 0° a 50° . A angulação estipulada foi suficiente para o mapeamento dos objetos desejados.

Devido ao constante movimento dos servos, foi necessário prender o dispositivo em um suporte de fixação com peso adicional, para que o sistema se mantivesse na posição durante todo o percurso. O suporte também foi feito de plástico rígido.

Figura 12 – Hardware do mapeador tridimensional.



Fonte: Autoria própria.

4.2 Software de controle

O Software de controle é responsável por movimentar o dispositivo e enviar, de forma individual, a medição de cada leitura à porta serial. Como pode ser visto na Figura 13, a cada mudança de posição em cada um dos servos motores é chama uma função nomeada como “lerSensor()”, onde seus parâmetros são os respectivos ângulos dos servos em dados inteiros, responsável por escrever os parâmetros medidos na porta serial.

Para controle dos servos motores, foi utilizado a biblioteca “Servo.h”. Essa biblioteca conta com a classe “Servo” que já possui funções prontas, de forma que facilite sua implementação. Das funções disponíveis, foram utilizadas no projeto:

- a) *attach()*: anexa o objeto da classe Servo a um respectivo pino do microcontrolador (ARDUINO, 2019); e
- b) *write()*: grava um valor no servo motor. Em um servo padrão, isso definirá o ângulo do eixo (em graus), movendo-o para essa orientação (ARDUINO, 2019).

Inicialmente, são definidos valores respectivos passos dos servos motores, sendo eles de 2 graus para o servo horizontal e de 5 graus para o servo vertical. Ainda, as posições dos servos são fixadas em zero grau e o limite de distância é estabelecido em 2 metros.

A etapa responsável pelo movimento dos servos funciona em *loop* dentro de uma estrutura de repetição *for*. A cada passo do servo responsável pelo movimento vertical, o servo motor agente do sentido horizontal percorre 180 graus. Ao fim do percurso, ainda com o servo vertical na mesma posição, o servo motor horizontal refaz seu movimento no sentido oposto. Essa repetição foi feita com objetivo de minimizar possíveis valores incertos durante a medição, mas será abordado de forma mais específica na próxima subseção. Como dito, para cada posição horizontal e vertical é a rotina “lerSensor()” é nomeada.

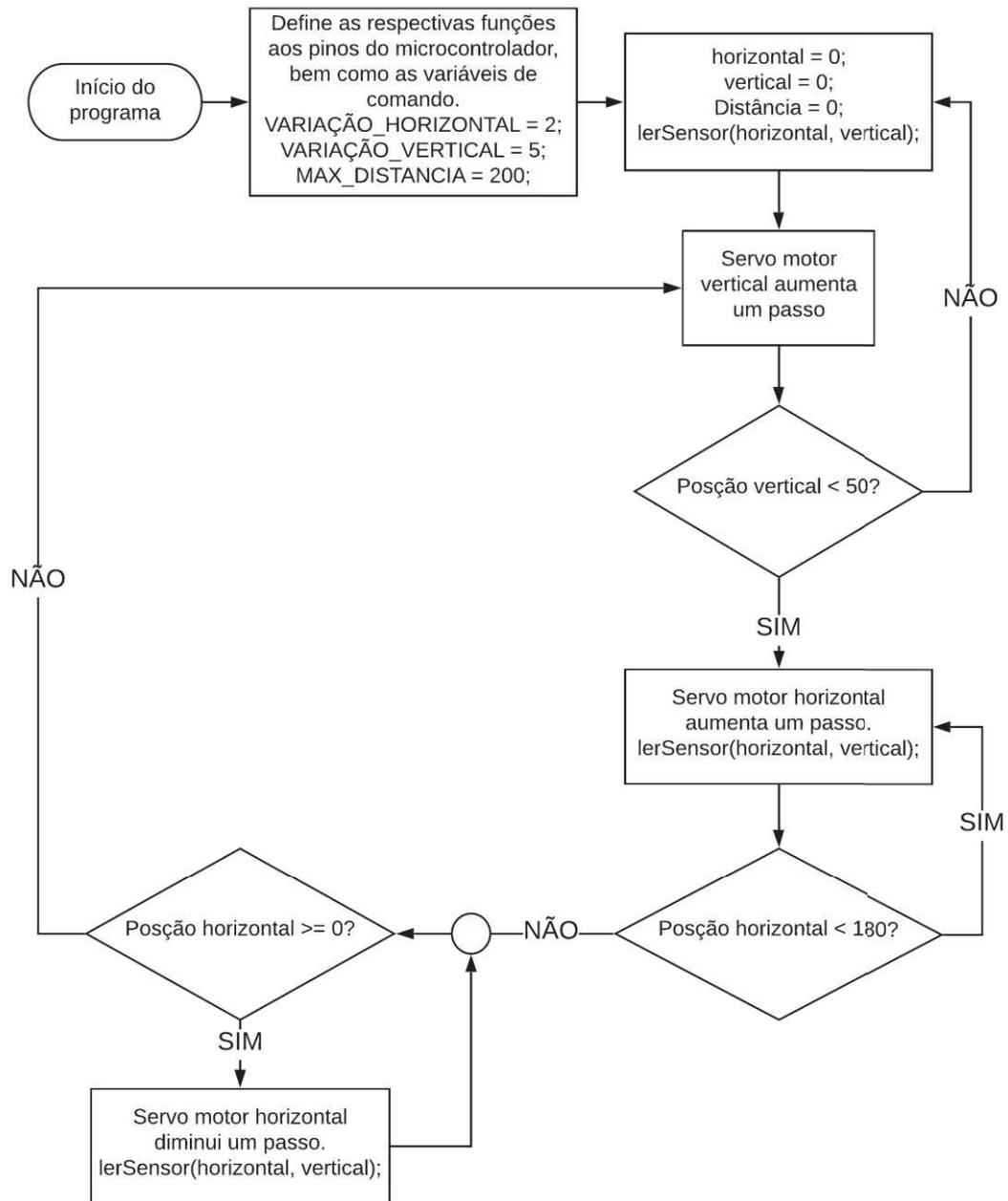
O sensor HC-SR04 possui um emissor de ondas cônicas, mas contém um receptor de ondas lineares. Essa característica faz com que o sensor detecte incorretamente distâncias que não estejam perpendiculares a placa do sensor. Dessa forma, foi empregue a biblioteca “NewPing.h”, que criou métodos para minimizar essas discrepâncias (ARDUINO, 2017).

Na rotina “lerSensor()”, está presente a função “ping()” da biblioteca “NewPing.h” que retorna o tempo de eco do sinal em microssegundos ou zero se nenhum valor estiver dentro do limite de distância definido (ARDUINO, 2017).

Por último, faz-se uma verificação se nenhum dos valores é nulo. Uma vez que isso seja satisfeito, a função “println()” envia os três dados em forma de uma *string* separadas por um caractere auxiliar, na forma: “distância#theta#phi”.

O código da etapa de controle pode ser observado em Anexo 1.

Figura 13 – Fluxograma do software de controle.



Fonte: Autoria própria.

4.3 Software de interface

Conforme pode ser visto em Anexo 2, o software de interface funciona em paralelo com o software de controle, visto na subseção anterior. Nesta etapa, enquanto os servos fazem a

varredura do ambiente, as distâncias e suas respectivas posições angulares são armazenadas em memórias, formando uma nuvem de pontos tridimensional.

O pacote de biblioteca *ToxicLibs* é uma coleção de códigos abertos que foi desenvolvido com propósito de animação, *design* de interface e visualização de dados. Nesse contexto, as bibliotecas “toxi.geom” e “troxi.processing” foram aplicadas para gerar e manipular a nuvem de pontos 3D. Junto a essas, a biblioteca “peasy”, do pacote *PeasyCam*, foi usada para renderização e rotação controlada ao redor dos três eixos do espaço. Para a comunicação serial com o código de controle, foi empregado a biblioteca “processing.serial”, padrão da interface *Processing*.

Antes de iniciar o mapeamento, a função “setup()” gera uma janela para exibir a interface. O objeto “port” foi criado para armazenar os dados recebidos pela porta serial e o objeto “cloud” foi gerado para armazenar os vetores tridimensionais. Também, foi criada uma matriz com todas as possíveis posições dos servos, isto é, uma matriz de 180 linhas e 50 colunas, onde cada valor foi iniciado em zero. Por fim, uma variável booleana “scanOver” para indicar o término do mapeamento.

A rotina de mapeamento mostrada na Figura 14 inicia conferindo se existe um valor válido na porta serial, ou seja, um valor não nulo. Satisfeita essa condição, a próxima etapa é desassociar cada uma das *strings* enviadas. Como visto na subseção anterior, os dados são estruturados na forma “distância#theta#phi”, assim, a cada dois caracter “#” forma-se uma coordenada, sendo elas: distância, ângulo *theta* e ângulo *phi*. Até este ponto, as coordenadas foram separadas, mas ainda são do tipo “string”. Dessa forma, faz-se a conversão para dados flutuantes para realizar a transformação de coordenadas.

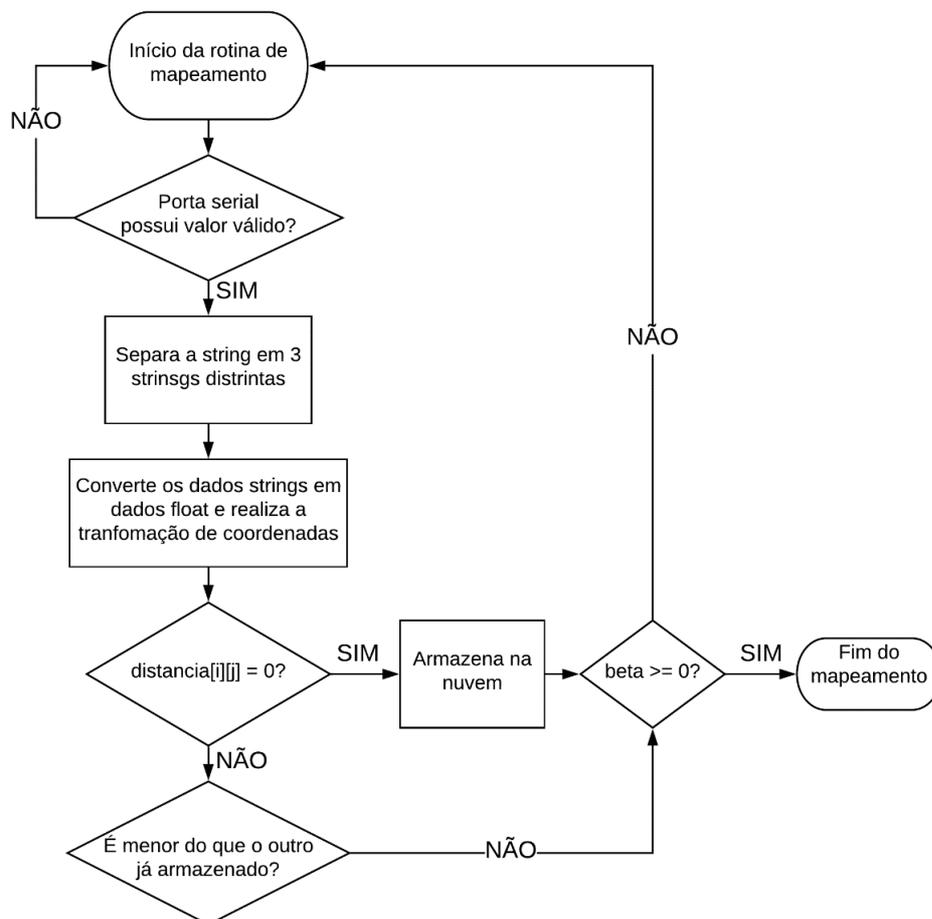
Tendo como base o referencial teórico, através das equações 1, 2 e 3, realiza-se a transformação de coordenadas esféricas em coordenadas cartesianas, para que cada uma das coordenadas possa ser armazenada na nuvem de pontos.

Antes do armazenamento, é feito a comparação dos dados. Relembrando o que foi explicado na metodologia, a cada passo vertical, o servo motor horizontal faz uma leitura de ida e volta. Com isso, é possível fazer duas medições em cada um dos pontos percorridos e escolher a melhor opção, no caso a menor distância, a fim de eliminar possíveis medições discrepantes.

No código, essa comparação é feita da seguinte forma: se a distância naquela coordenada ainda estiver em zero (valor inicializado na rotina “setup()”), significa que ainda é a primeira vez que o sensor passou por aquela posição, portanto a distância é armazenada. Todavia, caso a distância a ser armazenada naquela coordenada for diferente de zero significa que é o sensor já está fazendo o percurso de retorno. Nesse momento, é feita a comparação entre ambos os valores e o maior valor é descartado. O armazenamento é feito através da função “add” e o vetor para aquela posição é gerado com o comando “new”.

Ao longo de cada armazenamento, é feito a verificação do ângulo beta. Ao chegar em um valor superior a 50 graus a leitura se encerra e a variável booleana “scanOver” altera seu valor. Nesse instante, a nuvem de pontos é plotada através da rotina “plot()” e os eixos cartesianos são desenhados pela rotina “drawAxes()”.

Figura 14 – Fluxograma do software de controle.

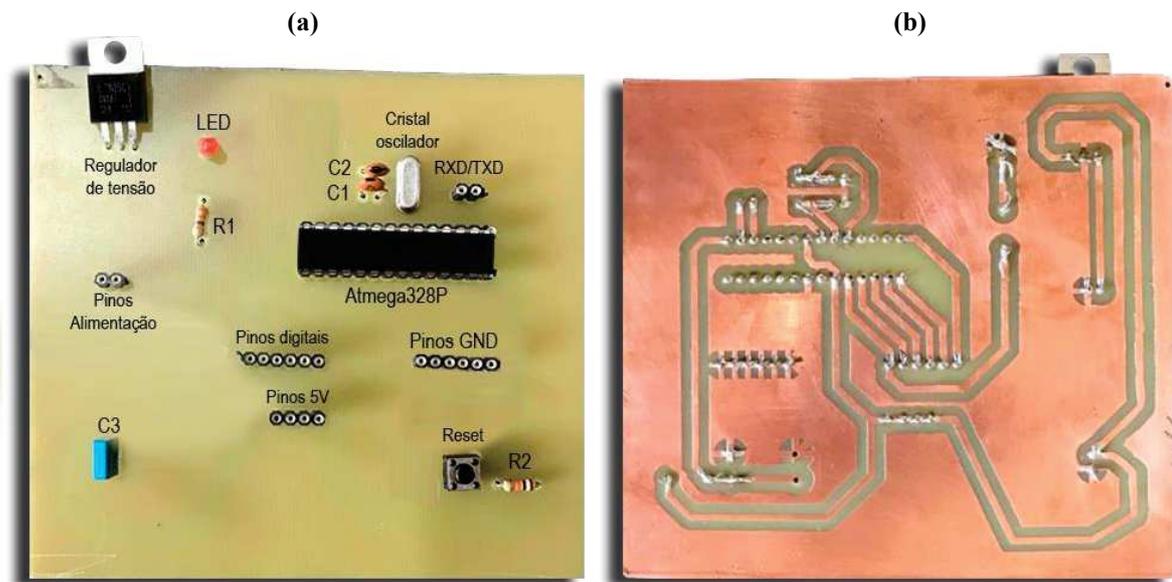


Fonte: Autoria própria.

4.4 Placa de circuito impresso

A Figura 15 apresenta o aspecto final da PCI após a soldagem dos componentes e validação de seu funcionamento.

Figura 15 – Placa de circuito impresso finalizada: (a) Vista superior, (b) Vista inferior.



Fonte: Autoria própria.

O circuito completo, presente na Figura 29 do Apêndice C, reúne os blocos necessários para o correto funcionamento do microcontrolador. Além deste e seus respectivas conexões, estão presentes o bloco de alimentação, do oscilador, de reset e outros blocos que compõem o circuito de forma auxiliar, tais como o LED indicativo de funcionamento e pinos que servem para alimentar os servos motores e o sensor de ultrassom.

O *layout* da PCI é ilustrado nas Figura 30 e 31 do Apêndice D, com dimensões de 100 milímetros quadrados. As trilhas possuem largura de 1,27 milímetros e os bornes de acoplamento possuem 1,8 milímetros de diâmetro. Conforme descrito por BRAGA (2017), baseado nas recomendações norma IPC-2221, foi evitado que trilhas paralelas percorressem um caminho muito longo. Ainda, fundamentado em boas práticas de uma placa de circuito impresso, evitou-se ângulos iguais ou superiores a 45 graus nas trilhas de sinais.

O investimento de produção é detalhado em Apêndice E. A compra dos servos motores e do sensor de ultrassom contabilizou 47,04% do orçamento total. Tendo em vista que alguns materiais necessários para a confecção da placa foram disponibilizados pela universidade, tais como ferro de solda, estanho, furador manual e percloreto de ferro, pôde-se economizar na manufatura.

Em relação à plataforma Arduino UNO, que custa em média 50 reais, a confecção da PCI acarretou em um custo 91,4% maior. No entanto, se a compra dos componentes fosse feita em quantidades maiores, o preço unitário acabaria diminuindo. Outro fator que aumentou o preço de produção em 6,92% foi o custo de transporte dos materiais, que foram comprados em Cascavel, Paraná.

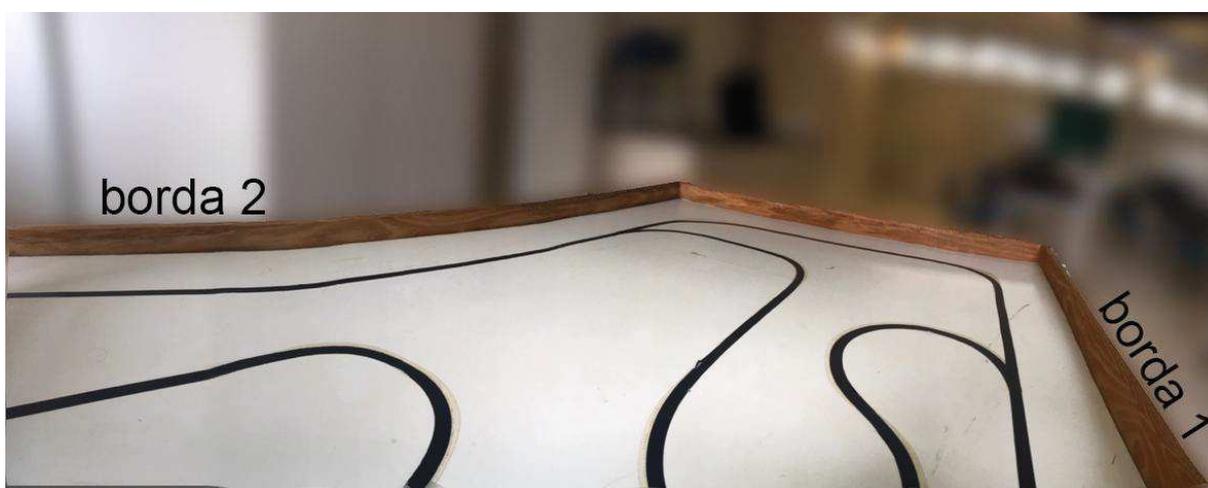
5 RESULTADOS E DISCUSSÕES

O presente capítulo apresenta os resultados experimentais obtidos pelo mapeador operando após a confecção da placa de circuito impresso.

5.1 Ambiente delimitado sem obstáculo presente

Primeiramente, foi feito o mapeamento do ambiente sem a presença dos objetos a serem mapeados. Como pode ser visto na Figura 16, a foto foi registrada na exata posição de fixação do dispositivo. Nessa posição, o mapeador foi colocado um pouco mais próximo da borda 1, do que da borda 2.

Figura 16 – Ambiente de medição sem objetos.

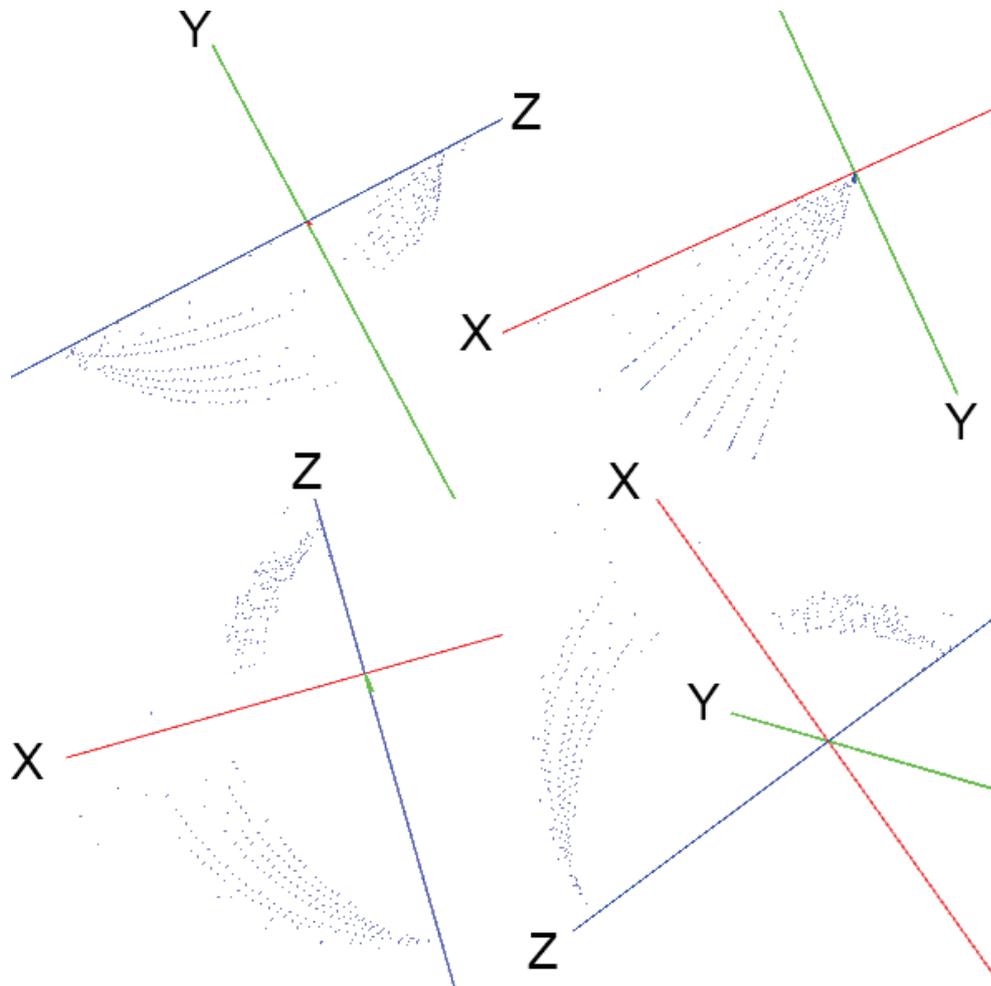


Fonte: Autoria própria.

Realizado o mapeamento na interface tridimensional (Figura 17), pode-se notar a borda 1 mais próxima da origem do que a borda 2, da mesma forma que o mapeador estava posicionado. Apesar de existir uma terceira borda mais distante, o dispositivo obteve dificuldades em detectá-la, devido a sua distância. Por conta disso, o mapeamento realizado demonstrou uma descontinuidade entre as bordas 1 e 2. Ademais, nota-se também uma forma

arredondada ao longo do ambiente mapeado, mas esse comportamento será discutido mais adiante.

Figura 17 – Resultado do mapeamento no ambiente de medição sem objetos.



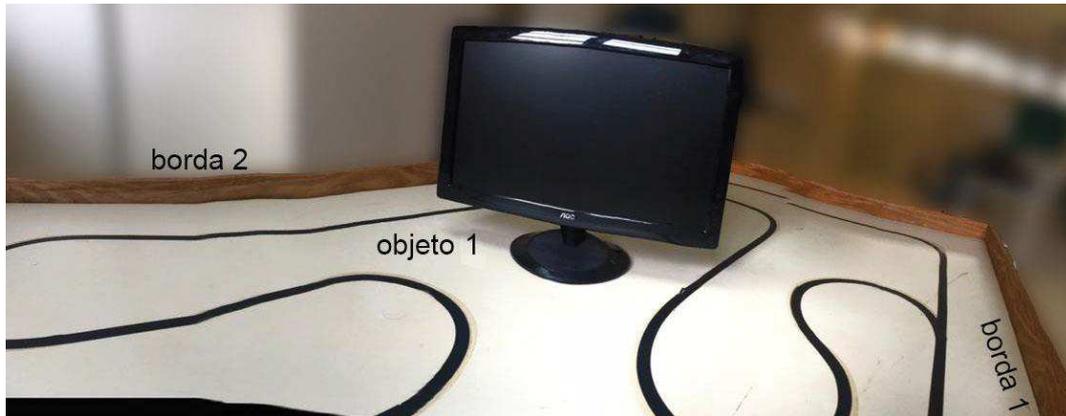
Fonte: Autoria própria.

5.2 Ambiente delimitado com um obstáculo presente

O segundo teste, realizado no mesmo ambiente, foi realizado com um monitor de computador à frente do mapeador (Figura 18). O objeto foi posicionado de forma mais centralizada, para que as distorções nas bordas que ocorreram no teste anterior fossem evitadas ao máximo. Como pode ser visto na Figura 19, apesar de haver dificuldade em identificar

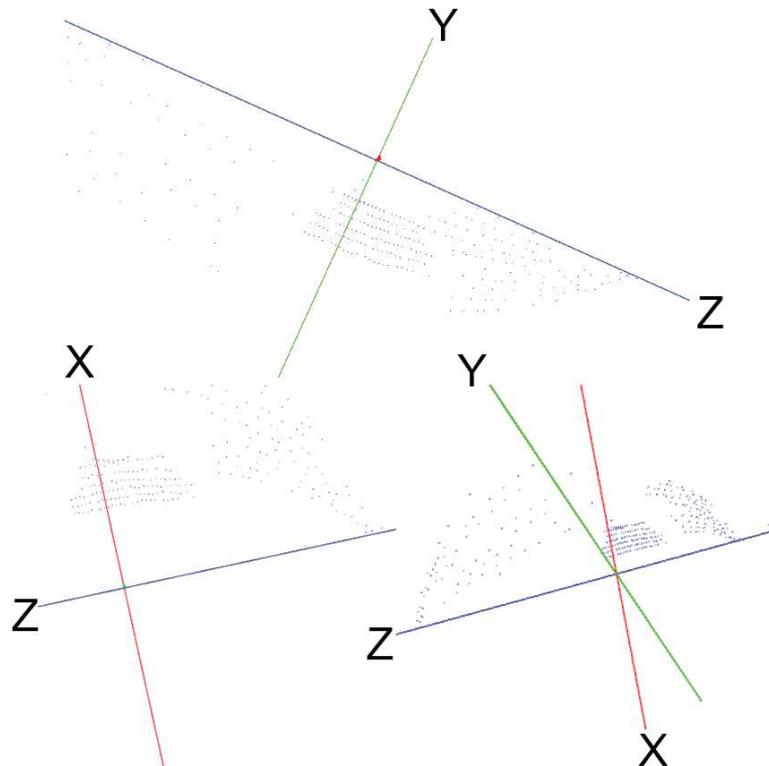
pontos na borda do monitor, o mapeador conseguiu identificar grande parte da dimensão retangular do obstáculo.

Figura 18 – Ambiente de medição com um objeto.



Fonte: Autoria própria.

Figura 19 – Resultado do mapeamento no ambiente de medição com um objeto.



Fonte: Autoria própria.

5.3 Ambiente delimitado com um obstáculo composto de dois materiais

O terceiro teste foi realizado com o propósito de averiguar qual seria o comportamento do mapeamento ao identificar um obstáculo com um líquido em seu interior. Para isso, como pode ser visto na Figura 20, foi feito o experimento com uma garrafa transparente.

Figura 20 – Ambiente de medição com um objeto composto por líquido em seu interior.

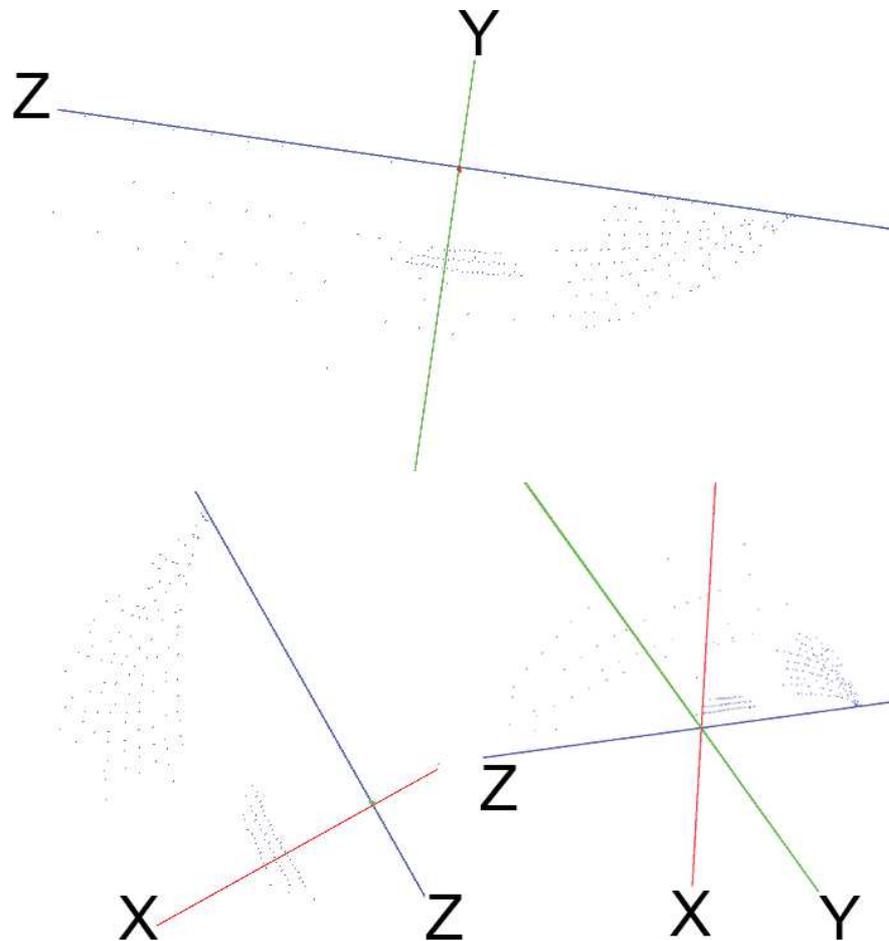


Fonte: Autoria própria.

Como visto na subseção 2.2 do referencial teórico, a propagação de ondas transversais, de flexão e superficiais praticamente inexistem na presença líquidos. Por conta disso, ao comparar testes feitos com a garrafa totalmente vazia ou totalmente cheia de água, pode-se ter uma diferença aproximadamente inexistente, visto que a técnica de ultrassom busca excitar vários tipos de propagação de onda, de forma que a onda longitudinal e de dilatação pudessem ser detectadas.

Analisando a Figura 21, o resultado do mapeamento ficou bem distante do objeto real. Esse fato se deve pela distância ao qual o objeto estava posicionado, fazendo com que a distância angular do mapeador fosse maior no momento da captura do sinal. Outro fator a ser considerado é que a dimensão vertical da garrafa é inferior a do monitor, do teste anterior. Por conta disso o mapeador iniciou seu funcionamento alguns graus a cima do objeto, capturando menos pontos. No entanto, ainda foi possível identificar o objeto como forma de obstáculo.

Figura 21 – Resultado do mapeamento no ambiente de medição com um objeto composto por líquido em seu interior.



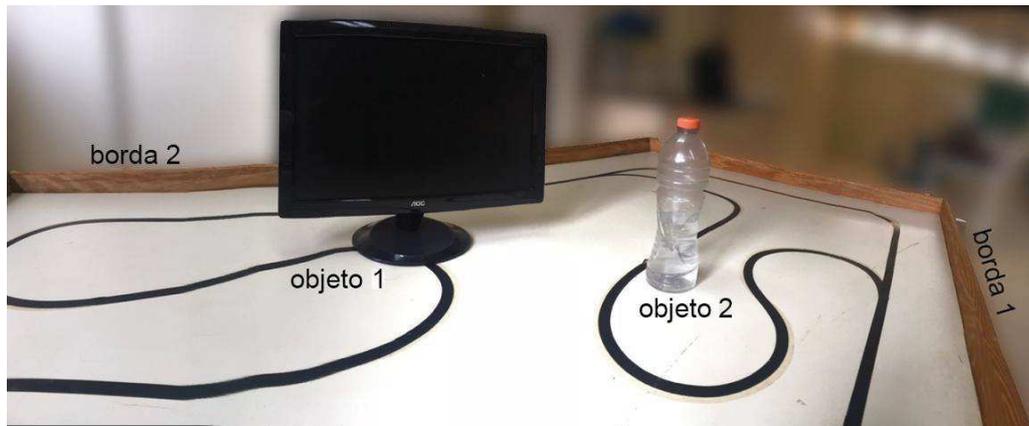
Fonte: Autoria própria.

5.4 Ambiente delimitado com dois obstáculos presentes

Para fins de comparação, o quarto teste foi realizado com a presença simultânea dos dois objetos testados anteriormente (Figura 22). Analisando a Figura 23, o resultado ficou próximo aos obtidos anteriormente. No entanto, o aspecto horizontal dos objetos mapeados dá a entender que ambos possuem mesma dimensão horizontal, o que não é verdade.

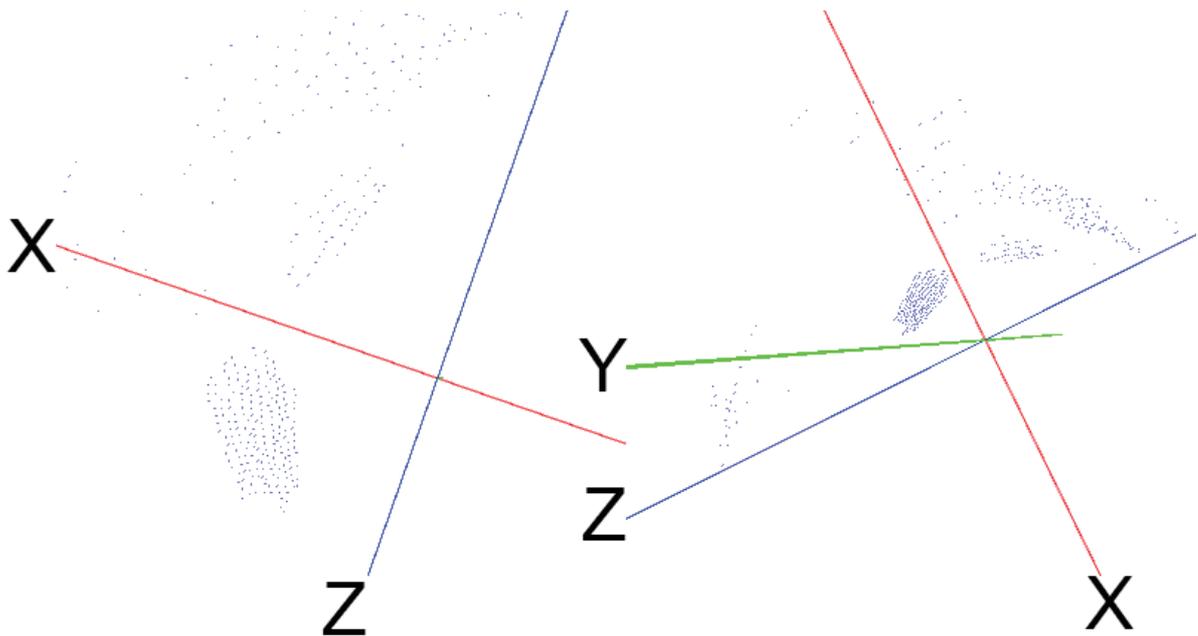
Parte desse comportamento deve-se ao fato de agora o monitor não estar mais centralizado. Dessa forma, estando um pouco mais para o lado, a posição de seus pontos acaba sendo muito mais distante da posição perpendicular, que seria a ideal. Todavia, ainda é possível distinguir na interface a presença de dois corpos distintos.

Figura 22 – Ambiente de medição com dois objetos presentes.



Fonte: Autoria própria.

Figura 23 – Resultado do mapeamento no ambiente de medição com dois objetos presentes.



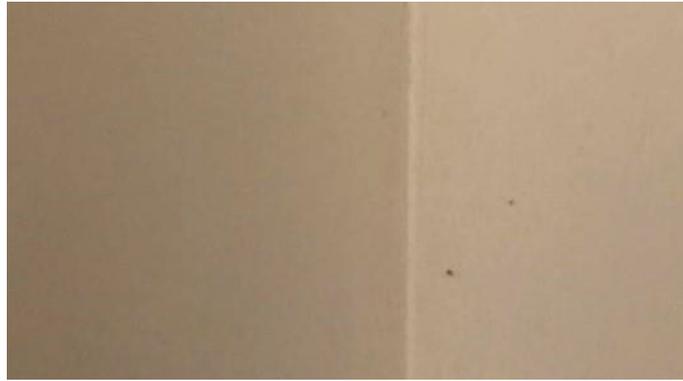
Fonte: Autoria própria.

5.5 Canto entre duas paredes

O quinto e último teste realizado teve como propósito investigar se o mapeador identifica mudanças bruscas de posições, como é o caso entre duas paredes perpendiculares (Figura 24). Neste ensaio, por conta de uma maior proximidade entre o mapeador e o obstáculo

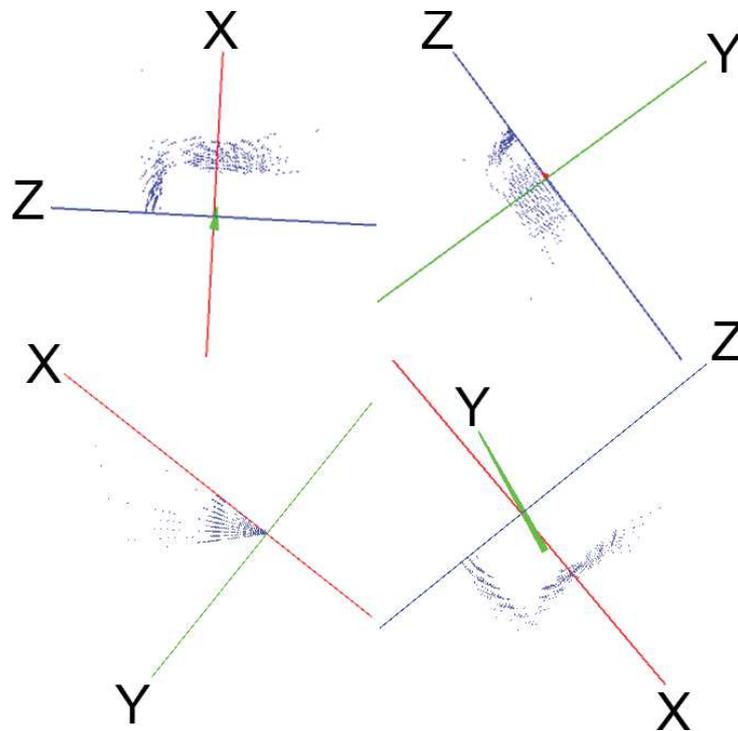
a ser mapeado, o resultado na interface foi bem mais fiel à realidade, como pode ser visto na Figura 25.

Figura 24 – Canto entre duas paredes perpendiculares.



Fonte: Autoria própria.

Figura 25 – Resultado do mapeamento realizado entre duas paredes perpendiculares.



Fonte: Autoria própria.

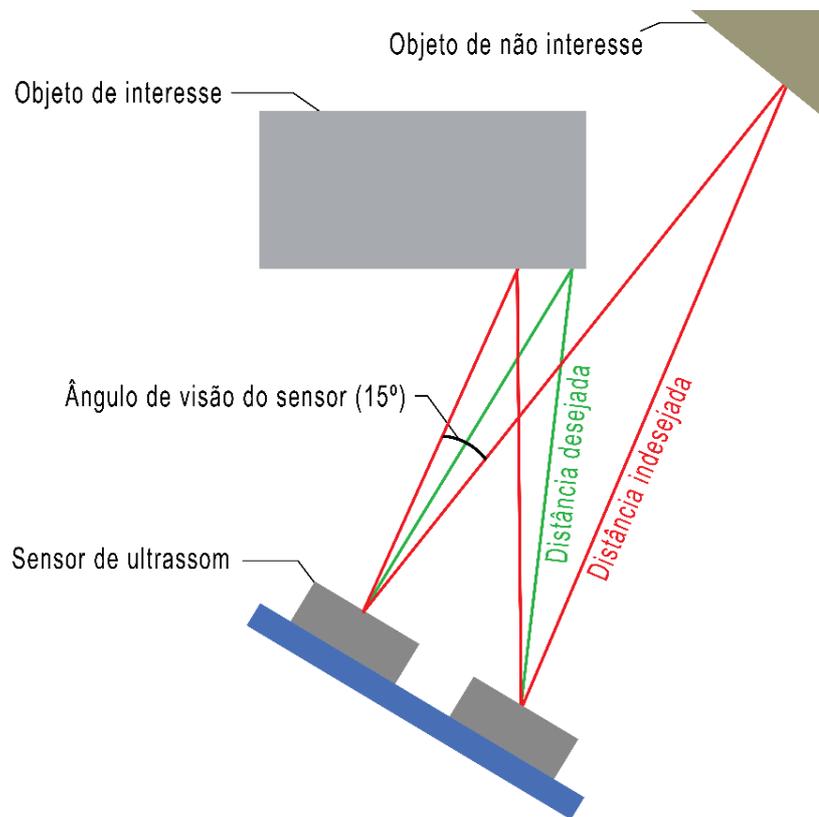
5.6 Interpretação dos testes

A situação ideal para que o dispositivo funcione de acordo deve ser em um ambiente totalmente aberto, de forma que seja evitado reflexões em obstáculos indesejados na medição, tais como paredes, armários ou bancadas, como foi o caso do laboratório da universidade, onde os testes foram realizados.

Dependendo do tipo de ambiente onde o sensor de ultrassom é aplicado, o som emitido pode refletir em vários objetos antes de voltar ao ponto de emissão. Posto que a distância medida pelo mapeador é sempre referente à primeira onda que retorna ao sensor, essa pode não necessariamente ser a onda do ponto analisado naquele momento. Esse fenômeno, ilustrado na Figura 26 para melhor entendimento, é chamado de “reflexão fantasma”, termo relacionado à situação que foi descrita.

Ainda com base na Figura 26, outro problema pode ser observado. A medição ideal do objeto medido pelo dispositivo deve ocorrer de forma perpendicular ao sensor, pois neste caso o som refletido provém exatamente do ponto em que se deseja saber a distância. Contudo, ao movimentar o sensor ao longo de seus eixos horizontais e verticais, o ponto de emissão e reflexão do mapeador fica inclinado em relação ao obstáculo, podendo acarretar em uma distância errada ou fora de alcance em algumas situações.

Figura 26 – Representação de uma situação recorrente na medição utilizando o sensor ultrassônico.



Fonte: Autoria própria.

Tendo dispostos os problemas relacionados ao ambiente e ao método de medição, ainda é possível pontuar possíveis falhas relacionadas ao sensor utilizado.

Idealmente, seria necessário um sensor que medisse exatamente o ponto analisado, independente da disposição que o mesmo esteja em relação ao objeto. Particularmente, o sensor de ultrassom utilizado possui um método de emissão de ondas cônicas, ou seja, uma onda ultrassônica é emitida da fonte e pode atingir objetos dentro de um determinado ângulo de visão. Conforme especificado pelo fabricante, o módulo HC-SR04 utilizado possui ângulo de emissão de aproximadamente 15°.

Como foi descrito na seção de materiais e métodos, foi implementado a biblioteca “NewPing.h” que realiza alguns cálculos de probabilidade para tentar eliminar pontos que sejam muito discrepantes. Ao inspecionar a biblioteca em questão, pode-se notar que o filtro das distâncias medidas funciona para pontos que estejam muito distantes. Como o passo de

movimentação do mapeador é pequeno e essas distâncias são muito próximas, a biblioteca não consegue distinguir quais pontos devem ser selecionados ou descartados. Quanto maior é a distância do obstáculo a ser medido em relação ao objeto, maior é a amplitude angular de emissão e menos pontual é a medição do sensor, gerando deformações maiores na leitura do objeto. O resultado, como aparece nos testes, é que a interface gera formas arredondadas, pois quanto mais distante o objeto está, maior grau de deformação ele obtém.

6 CONCLUSÃO

Ao final, desenvolveu-se um dispositivo mapeador que contempla diversas áreas de conhecimento da graduação, como microcontroladores, sistemas embarcados, eletrônica analógica, medidas e sensores, processamento digital de sinais, dentre outras.

Apesar de cumprir os objetivos propostos, os testes empíricos realizados com o mapeador sofreram distorções. Acredita-se que parte disso se deve ao fato do custo financeiro envolvido no projeto, principalmente o do sensor ultrassônico, que acabou comprometendo a qualidade final do projeto ainda que a interface tenha gerado uma nuvem de pontos possível de ser utilizada para a finalidade que foi proposta.

Independentemente do resultado geral obtido, o empenho foi válido, pois o trabalho ainda pode ser melhorado de diversas formas em trabalhos futuros, seja no algoritmo de mapeamento, seja nos dispositivos empregados no sistema. Uma vez que essas melhorias sejam feitas para que o sistema se torne mais versátil, o mapeador pode até mesmo ser integrado a propostas de robóticas existentes na universidade. Portanto, o sistema tem pleno potencial de ser utilizado em sistemas para mapeamento e navegação em ambientes onde a principal finalidade seja de identificar obstáculos.

De maneira geral, ao se avaliar os resultados obtidos, o dispositivo atendeu os requisitos previstos, garantindo que os objetivos predefinidos fossem alcançados e, apesar de um preço relativamente alto na confecção da placa de circuito impresso, a experiência de montar o projeto de uma placa justifica o investimento.

6.1 Dificuldades encontradas

O maior desafio desde o início do projeto foi o desenvolvimento do algoritmo de interface tridimensional, pois este não é um assunto abordado no curso de Engenharia Eletrônica e requer conhecimento prévio de como elaborar uma interface visual através de linhas de código. No início, a interface ilustrava uma imagem bidimensional, de forma que a profundidade dos pontos não fosse reconhecida, tornando as medidas iguais em qualquer posição analisada. Mais tarde, conforma foi implementado o terceiro eixo no sistema, era impossível mover a tela da interface para observar diferentes ângulos, pois a imagem dos pontos

se arrastava ao longo da tela. Isso foi solucionado com o estudo do princípio de funcionamento de uma nuvem de pontos, de forma que as coordenadas fossem salvas em um vetor tridimensional. A cada vez que o ângulo de visualização é modificado, a interface limpa a tela e chama novamente a rotina para plotagem dos pontos que já estão armazenados.

6.2 Sugestões para trabalhos futuros

Para dar continuidade ao projeto, tem-se como sugestões:

- a) Escolha sistema de mapeamento mais eficiente, como por exemplo o *LIDAR*;
- b) Utilizar mais de um sensor, com objetivo de aumentar o número de leituras por unidade de tempo;
- c) Implementar o algoritmo de controle de forma que a interpretação dos pontos medidos pelo sensor seja mais coesa; e
- d) Integrar o dispositivo em um sistema robótico, tornando a sua movimentação autônoma.

7 REFERÊNCIAS

ALYAS, Mohammed; SHAMA, Shreyash; ERRARHEB, Mohammed; REDA, Youssef; AGIEZ, Youssef; ZIONG, Zingguo. **Low-cost Arduino-Based Ultrasonic Radar System.** University of Bridgeport. Bridgeport. Disponível em <https://scholarworks.bridgeport.edu/xmlui/bitstream/handle/123456789/2226/1612-Ultrasonic_Radar_poster.ppt_%281%29.pdf?sequence=1&isAllowed=y>. Acesso em 24 de abril. 2019.

ANDREEV, Victor; TARASOVA, Victoria. **Identification of the obstacle shape using the ultrasonic module of modular mobile robot.** International Symposium On Intelligent Manufacturing And Automation. 2018.

ATMEL. **Datasheet - Microchip Technology**, Novembro 2016. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf>. Acesso em: 26 de novembro de 2019.

Arduino. **NewPing Library for Arduino.** 2017. Disponível em: <<https://playground.arduino.cc/Code/NewPing/>>. Acesso em 19 de dezembro de 2019.

BASTOS, Izabella Carneiro. **Proposta para melhoria da resolução em Radares FMCW utilizados na medição de nível.** Universidade Federal de Itajubá. Itajubá. 2010.

Blog Autocore. **Aprenda a utilizar o sensor de distância ultrassônico HC_SR04 com Arduino.** 2017. Disponível em: <<https://autocorerobotica.blog.br/aprenda-utilizar-o-sensor-de-distancia-ultrassonico-hc-sr04-com-arduino/>>. Acesso em 26 de novembro de 2019.

BOLLER, C. *et al.* **Enciclopedia of Structural Health Monitoring**. Ed. John Wiley and Sons, 1ª edição, Nova York, 2009.

BRAGA, Newton. **Como funciona o radar**. Instituto Newton C. Braga. 1995. Disponível em: <<https://www.newtonbraga.com.br/index.php/como-funciona/10739-como-funciona-o-radar-art154>>. Acesso em 03 de novembro de 2019.

BRAGA, Newton. **Constante dielétrica de alguns materiais**. Instituto Newton C. Braga. 2011. Disponível em: <<http://www.newtonbraga.com.br/index.php/almanaque/406-constante-dieletrica-de-alguns-materiais.html>>. Acesso em 02 de junho de 2019.

BRAGA, Newton. **Regulador de Tensão 7800**. Instituto Newton C. Braga. 2011. Disponível em: <<https://www.newtonbraga.com.br/index.php/como-funciona/1076-art156>>. Acesso em 02 de junho de 2019.

BROOK, M. V. **Ultrasonic Inspection Technology Development and Search Unit Design**. Editora Wiley, ed.1, Nova York, 2012.

BUENO, Michael Bueno. **Tipos e aplicações de sensores na indústria**. Engerey. 2017. Disponível em: <<http://www.engerey.com.br/blog/tipos-e-aplicacoes-de-sensores-na-industria>>. Acesso em 02 de junho de 2019.

Buzzer Ativo 5V Bip Contínuo – PCI 12mm. Usina info. Disponível em: <<https://www.usinainfo.com.br/diversos/buzzer-ativo-5v-bip-continuo-pci-12mm-2988.html>>. Acesso em 03 de junho de 2019.

CAPELLI, Beatriz Midena. **Desenvolvimento de uma estufa controlada e monitorada remotamente**. Escola de Engenharia de São Carlos. São Carlos. 2014.

CHAKRABARTY, Prithvijit; LOVEKAR, Kartik. **An ultrasonic 3D scanner**. 2017. Disponível em <http://prithvijc.pythonanywhere.com/static/3D_SCANNER.pdf>. Acesso em 26 de novembro 2019.

Cytron Technologies, **HCSR04**. <<https://www.datasheetspdf.com/pdf/1291829/Cytron/HCSR04/1>>. Acesso em 28 de maio. 2019.

DE FREITAS, Diogo; GUERREIRO, Luís Fernando; KUME, Leonardo Watanabe; LEE, Diego Gabriel. **Mapeador tridimensional de ambiente baseado em ultrassom**. Universidade Tecnológica Federal do Paraná. Curitiba. 2013.

DHANSHRI, Ahire; PRIYANKA, Ahire. **Radar System Using Arduino**. India. 2017. Disponível em: <<https://www.academia.edu/RegisterToDownload#RelatedPapers>>. Acesso em 28 de maio. 2019.

DUTT, Anuj. **Arduino based RADAR System**. München, Germany: GRIN Verlag, 2014. Disponível em: <https://scholar.google.com/scholar?cluster=7263849830335695688&hl=pt-BR&as_sdt=0,5&scioldt=0,5>. Acesso em 28 de maio. 2019.

HALLIDAY, David. RESNICK, Robert. WALKER, Jearl. **Fundamentos de Física 3, 4 ed.** Editora LTC, 2006.

HATEM, Haraa Raheem. **Design and implementation of ultrasonic radar system for distance measurements using Arduino**. International Journal of Engineering & Technology. 5 pag. 2018. Disponível em: <https://www.researchgate.net/publication/329019259_Design_and_implementation_of_ultra_sonic_radar_system_for_distance_measurements_using_Arduino>. Acesso em 06 de junho. 2019.

HEADRICK J; THOMASON, J. **Applications of high-frequency radar**. Radio Science, vol 33, 1998. Disponível em: <<https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/98RS01013>>. Acesso em 28 de maio. 2019

HELERBROCK, Rafael. **Infrassom e ultrassom; Brasil Escola**. Disponível em: <https://brasilecola.uol.com.br/fisica/o-infrasom-ultrasom.htm>. Acesso em 16 de dezembro de 2019.

IPC-2221B. **Generic Standard on printed board design**. Novembro. 2010. Disponível em <<http://www.ipc.org/TOC/IPC-2221B.pdf>>. Acesso em 26 de novembro de 2019.

KOSE, Fatih; KAPLAN, Kaplan; ERTUNC, H. Metin. **Designing of a Shooting System Using Ultrasonic Radar Sensor**. Faculty of Engineering. Turquia. 2017.

LUZ, A. M. R. & ÁLVARES, B. A. **Física: volume 2**. Editora Scipione S.A, São Paulo, 2005.

MACHADO, Matheus de Melo. **Hardware integrado de baixo custo para automação veicular**. Universidade Federal de Mato Grosso do Sul. Pantanal, 2015.

MAJCHRZAK, JAROSŁAW; MICHALSKI, MATEUSZ; WICZYNSKI, GRZEGORZ. **Distance Estimation With a Long-Range Ultrasonic Sensor System**. IEEE Sensors Journal. [s. L.], p. 767-773. jul. 2010.

MARTIN, C. C. **Ultrassom**. Editora Associação Brasileira de Ensaio Não Destrutivos e Inspeção, 1ª edição, São Paulo, 2012.

MicroServo 9g SG90 TowerPro. Baú da Eletrônica. Disponível em: <<http://www.baudaeletronica.com.br/micro-servo-9g-sg90-towerpro.html>>. Acesso em 03 de julho. 2019.

NUVEM DE PONTOS. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2019. Disponível em: <https://pt.wikipedia.org/w/index.php?title=Nuvem_de_pontos&oldid=55092367>. Acesso em: 26 de novembro de 2019.

Processing. Disponível em: <<https://www.processing.org/>>. Acesso em 02 de junho. 2019.

RIDENOUR, N. Louis. **Radar System Engineerring.** McGraw Hill, 1947.

SILVEIRA, Cristiano Bertulucci. **Sensor: Você sabe o que e quais os tipos?.** Citisystems. Disponível em: <<https://www.citisystems.com.br/sensor-voce-sabe-que-quais-tipos/>>. Acesso em 02 de junho. 2019.

SKOLNIK, M. **Radar Handbook.** Mc Graw Hill, Nova York, 3 ed. 2008.

SMIDT, André Crepaldi Geiger. **Implementação de uma plataforma robótica controlada remotamente utilizando o Arduíno.** Escola de Engenharia de São Carlos, da Universidade de São Paulo. 2013. Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/97/970010/tce-19112013-093717/?&lang=br>>. Acesso em 02 de junho. 2019.

SMYERS, KEN. **StatTalk: Warren S. Jhonson patented the thermostat in 1883.** Control Trends. 2013. Disponível em <<https://controltrends.org/controltrends-news/news-and-information/06/stattalk-warren-s-johnson-patented-the-thermostat-in-1883-thanks-asme-milwaukee-for-the-history-heritage/>>. Acesso em 02 de junho. 2019.

TARCHI, Dario; VESPE, Michele; GIOIA, Ciro; SERMI, Francesco; KYOVTOROV, Vladimir; GUGLUIERI, Giorgio. **Low Cost Mini Radar: Design Prototyping and Tests.**

Journal of Sensor. 15 pag. 2017. Disponível em:
<<https://www.hindawi.com/journals/js/2017/8029364/>>. Acesso em 28 de maio. 2019.

THOMAZINI, Daniel; **Sensores Industriais: Fundamentos e Aplicações**. 8ª Ed. Editora Érica. 2018.

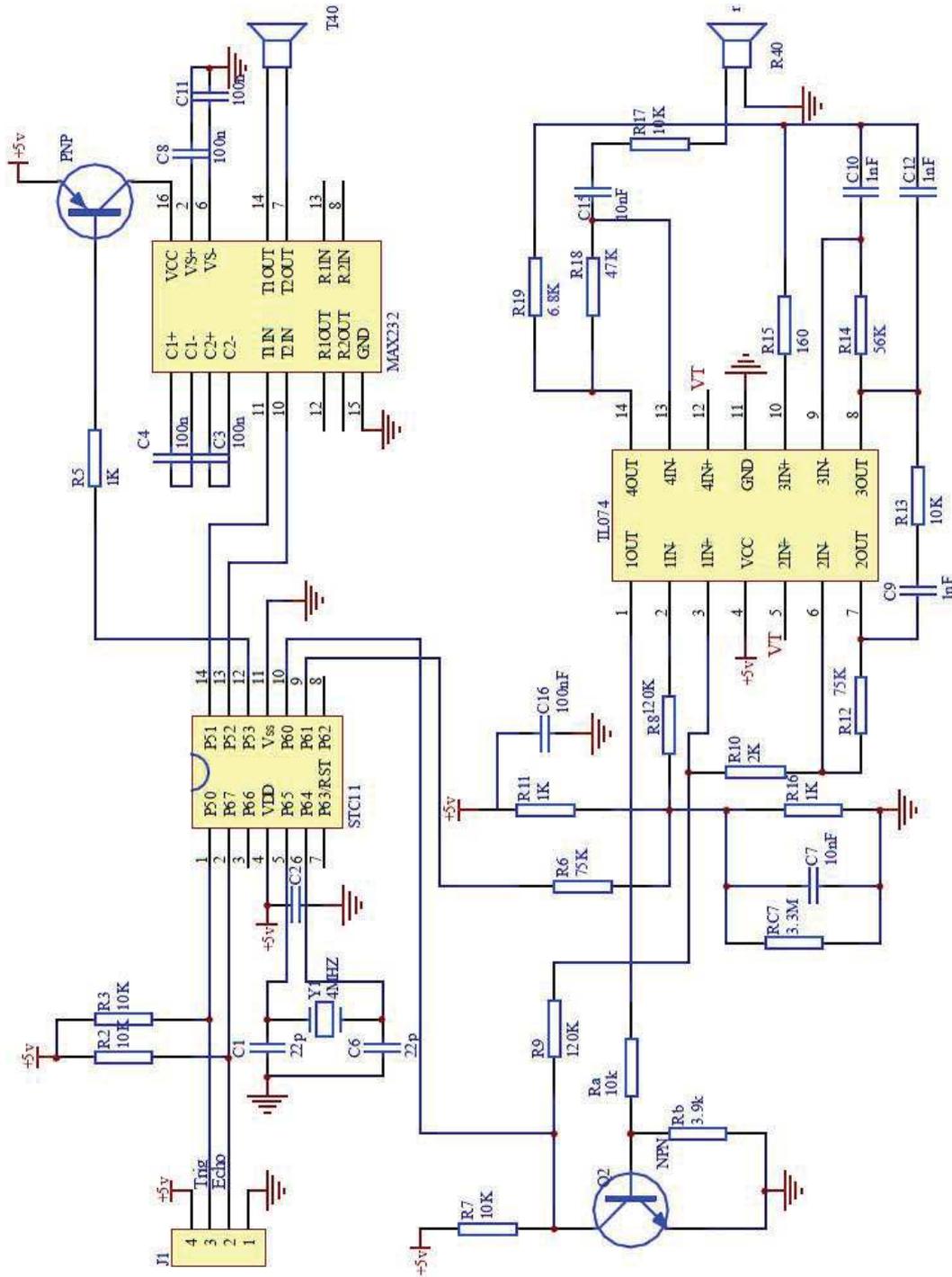
ULTRASSONOGRRAFIA. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2017. Disponível em:
<<https://pt.wikipedia.org/w/index.php?title=Ultrassonografia&oldid=50752827>>. Acesso em: 26 de novembro de 2019.

UTFPR. Institucional. **Site da Universidade Tecnológica Federal Do Paraná**, 19 outubro 2017. Disponível em: <<http://portal.utfpr.edu.br/institucional>>. Acesso em: 26 de novembro de 2019.

WEISSTEIN, Eric W. **Spherical Coordinates. From MathWorld – A Wolfram Web Resource**. 2019. Disponível em: <<http://mathworld.wolfram.com/SphericalCoordinates.html>>. Acesso em: 19 de novembro de 2019.

ANEXO A – Esquemático do sensor de ultrassom

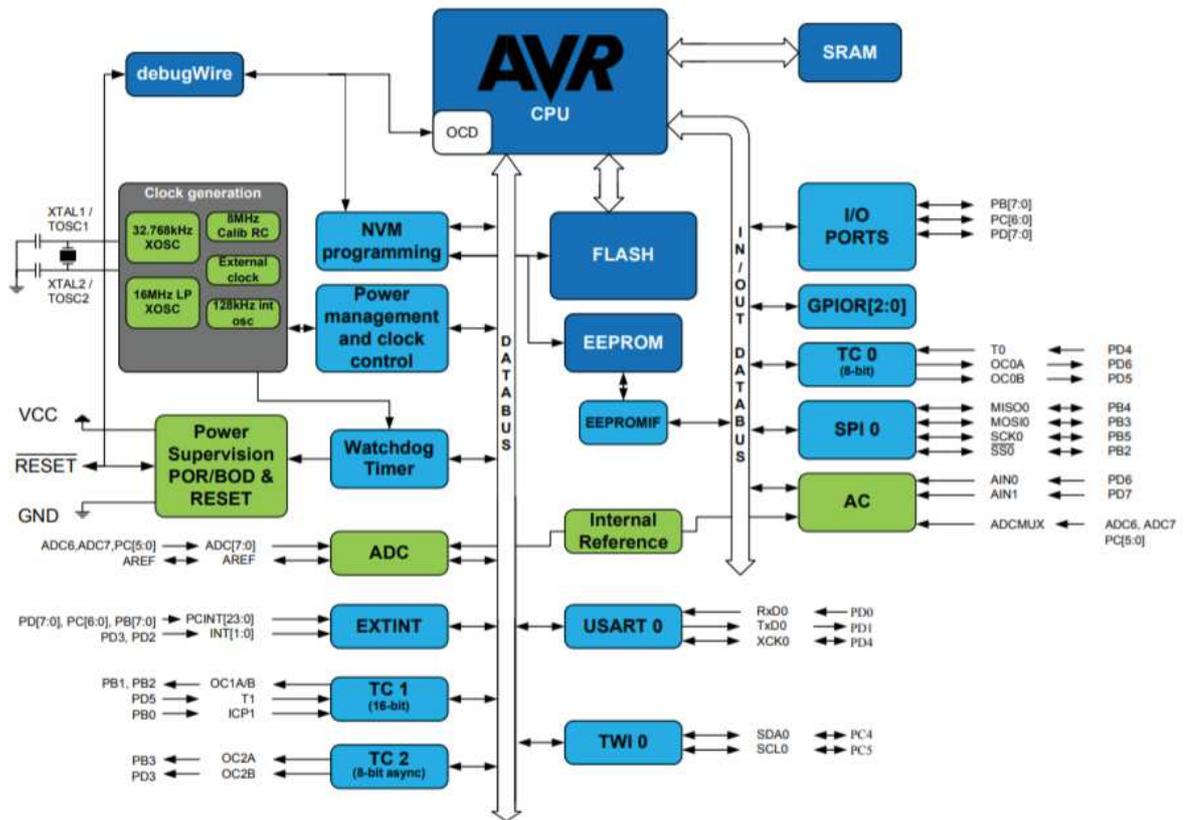
Figura 27 – Esquema elétrico do sensor de ultrassom HC-SR04.



Fonte: INFOOTEC (2018).

ANEXO B – Diagrama de blocos do microcontrolador

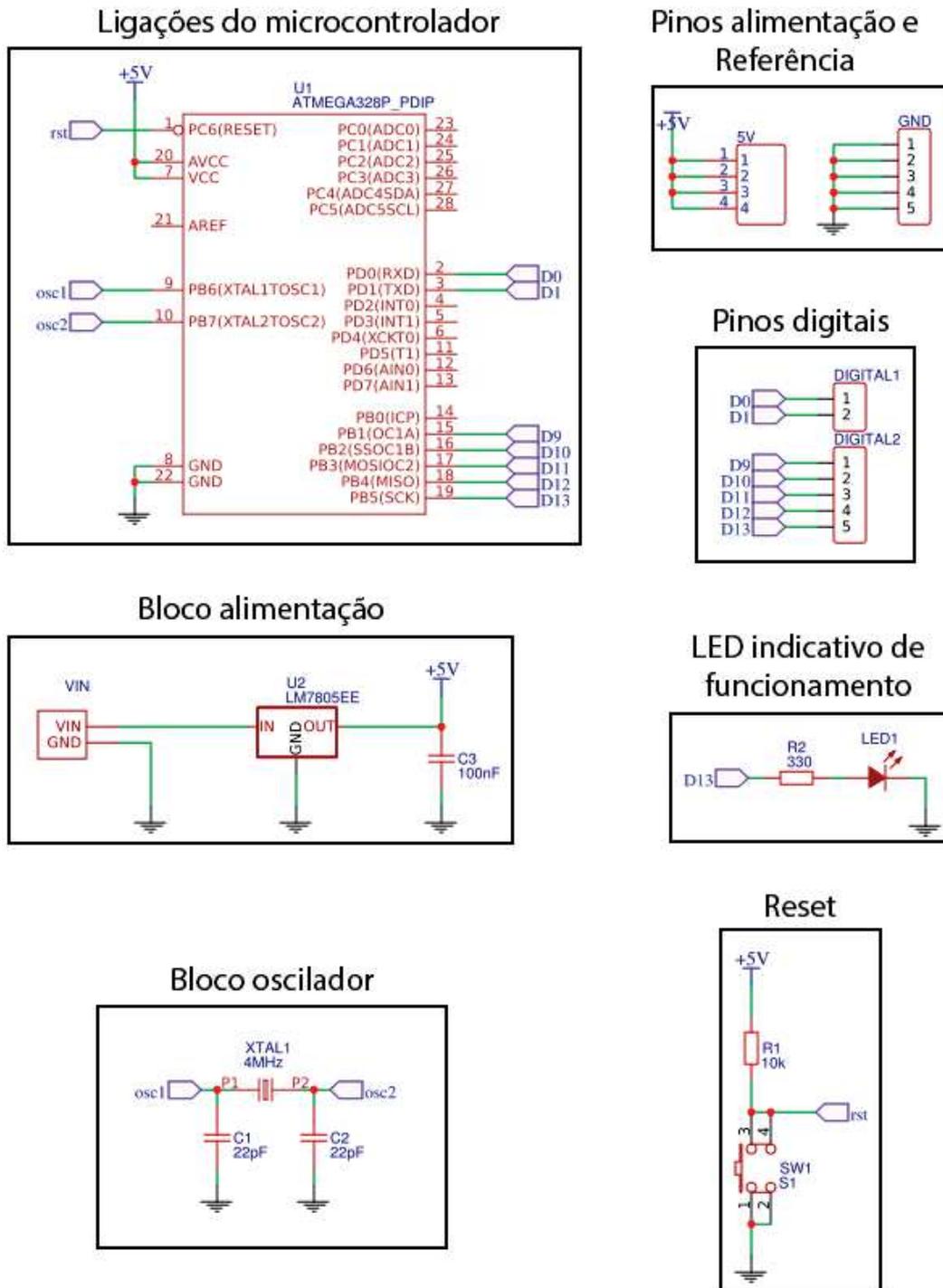
Figura 28 – Diagrama de blocos do microcontrolador ATmega328P.



Fonte: Atmel (2016).

APÊNDICE A – Esquemático do sistema

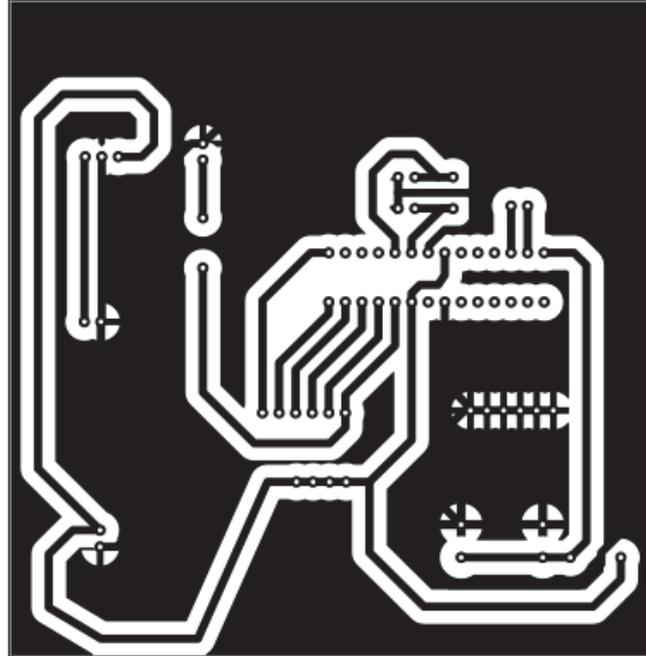
Figura 29 – Esquema elétrico da placa de circuito impresso



Fonte: Autoria própria.

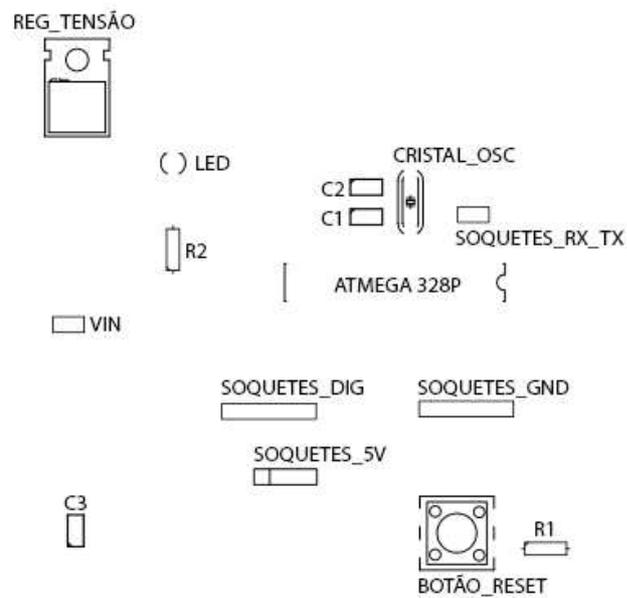
APÊNDICE B – *Layout* PCI

Figura 30 – Vista das trilhas do circuito na superfície inferior da placa.



Fonte: Autoria própria.

Figura 31 – Vista da disposição dos componentes na superfície superior da placa.



Fonte: Autoria própria.

APÊNDICE C – Valor investido no projeto

Tabela 1 – Preço dos componentes frete.

Item	Quantidade	Preço unitário (R\$)	Subtotal (R\$)
Capacitor cerâmico 22 pF / 50 V	2	0,10	0,20
Capacitor poliéster 100 nF / 63 V	1	2,10	2,10
Chave tátil push-button	1	0,50	0,50
Conector PCI tira 40 pinos macho-fêmea	2	2,00	4,00
Cristal oscilador 16 MHz	1	2,50	2,50
Jumper macho-fêmea	10	0,50	5,00
Jumper macho-macho	10	0,50	5,00
LED difuso 5 mm vermelho	1	0,20	0,20
Microcontrolador ATMEGA 328P Atmel	1	27,00	27,00
Módulo conversor USB RS-232	1	23,00	23,00
Placa de fibra de vidro 10x10 cm	1	10,00	10,00
Regulador de tensão 7805 metálico	1	3,50	3,50
Resistor 330 Ω 1/4 W	1	0,10	0,10
Resistor 10 k Ω 1/4 W	1	0,10	0,10
Subtotal itens PCB			83,20
Micros Servos 9g SG90 TowerPro	2	25,00	50,00
Sensor ultrassônico HC-SR04	1	35,00	35,00
Subtotal Componentes mapeador			85,00
Frete			12,50
Total geral			180,70

Fonte: Autoria própria.

APÊNDICE D – Algoritmo de controle

```
#include <Servo.h> // biblioteca para movimentar os servos motores

#include <NewPing.h> // biblioteca para calcular a distância do sensor ultrassônico

#define TRIGGER_PIN 6 //pino utilizado para o trigger do sensor de ultrassom

#define ECHO_PIN 7 // pino utilizado para o echo do sensor de ultrassom

#define MAX_DISTANCE 200 // máxima distância medida, em centímetros

#define VARIACAO_HORIZONTAL 2 //determina a variação de ângulo horizontal

#define VARIACAO_VERTICAL 5 //determina a variação de ângulo vertical

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // setup dos pinos e da
máxima distância

Servo servoHorizontal, servoVertical; // define dois servos motores

int horizontal = 0, vertical = 0; // valores de ângulos iniciais

int distancia = 0; // valor de distância inicial

void setup()

{

servoHorizontal.attach(9); // associa o pino 9 ao objeto do servo horizontal

servoVertical.attach(10); // associa o pino 10 ao objeto do servo vertical

Serial.begin(115200); // porta serial opera em 115200 bits por segundo

servoVertical.write(vertical); // define a posição inicial do servo motor vertical

servoHorizontal.write(horizontal); // define a posição inicial do servo motor horizontal

}
```

```

void loop()
{
  for(int vertical = 1; vertical <= 51; vertical+=VARIACAO_VERTICAL)
  { // varia o angulo theta
    for(int horizontal = 1; horizontal <= 180; horizontal+=VARIACAO_HORIZONTAL)
    { // varia o angulo phi
      servoHorizontal.write(horizontal); // incrementa o servo motor horizontal

      lerSensor(horizontal, vertical); // envia a posição
    }

    for(int horizontal = 179; horizontal >= 1; horizontal-=VARIACAO_HORIZONTAL)
    { // varia o angulo theta
      servoHorizontal.write(horizontal); // decrementa o servo motor horizontal

      lerSensor(horizontal, vertical); // envia a posição
    }

    servoVertical.write(vertical); // incrementa o servo motor vertical

    lerSensor(horizontal, vertical); // envia a posição
  }
}

void lerSensor(int theta, int phi)
{
  delay(30); // espera 30ms, aproximadamente 20 pulsos/segundo, antes de modificar a posição
  unsigned int uS = sonar.ping(); // recebe o tempo necessário para a distância medida, em
  microssegundos
}

```

```
distancia = uS / US_ROUNDTRIP_CM; // converte o tempo obtido em distância, em centímetros
```

```
if ((distancia != 0) && (theta != 0) && (phi != 0)) // verifica se houve medição válida
```

```
{
```

```
    //envia para o computador os valores da distância medida e ângulos dos servos por comunicação serial
```

```
    Serial.println(String(distancia) + "#" + String(theta) + "#" + String(phi));
```

```
}
```

```
}
```

APÊNDICE E – Algoritmo de interface

```
import processing.serial.*;

import toxi.geom.*;

import toxi.processing.*;

import peasy.*;

Serial port;

ToxiclibsSupport gfx;

PeasyCam cam;

int servo_num = 1, val = 0, alpha = 0, beta = 0, dist = 0, dist_aux = 0;

float x_aux, x, y_aux, y, z_aux, z;

float distances[][];

ArrayList<Vec3D> cloud;

boolean scanOver;

void setup()

{

    size(700, 500, P3D); // tamanho da janela de mapeamento

    frameRate(100); // taxa de quadros

    // dados seriais

    println(Serial.list());

    port = new Serial(this, Serial.list()[4], 115200);
```

```
// configura a ferramenta de plotagem 3D

gfx = new ToxiclibsSupport(this);

cam = new PeasyCam(this, 0, 0, 0, 100);

cloud = new ArrayList<Vec3D>();

scanOver = false;

// define valores iniciais para as posições

distances = new float[181][52];

for(int i = 0; i < 181; i++)

    for(int j = 0; j < 52; j++)

        distances[i][j] = -1;

}

void draw(){

    background(255); // fundo branco

    if(scanOver) // aguarda o término do mapeamento

    {

        plot(); // plota a nuvem de pontos 3D

        eixos(); // plota os eixos cartesianos

    }

}

void serialEvent(Serial port) // analisa os dados enviados pela porta serial
```

```

{
if (port.available () > 0) // verifica se o dado é válido
{
String posicao = port.readStringUntil('\n'); // lê a String recebida
if (posicao != null) // verifica se a String recebida não é nula
{
String coordenada[] = new String[3]; // separa a string Distância#theta#phi em 3
coordenada = posicao.split("#"); // através do separados "#"

int coordenadasize = coordenada.length; // recebe a quantidade de strings

if (coordenadasize == 3) // a string foi separa em 3?
{
System.out.println(coordenada[0]); // string1: distância
System.out.println(coordenada[1]); // string1: angulo theta
System.out.println(coordenada[2]); // string1: angulo phi

float Dist = Float.parseFloat(coordenada[0]); // converte a string distância em flutuante
float Alpha = Float.parseFloat(coordenada[1]); // converte a string do ang theta em
flutuante
float Beta = Float.parseFloat(coordenada[2]); // converte a string do ang phi em
flutuante

// transformação de coordenadas esféricas em cartesianas
x = (Dist * cos(radians(Beta)) * sin(radians(Alpha))); // transforma eixo x

```

```
y = (Dist * sin(radians(Beta)) * sin(radians(Alpha))); // transforma eixo y
```

```
z = (Dist * cos(radians(Alpha))); // transforma eixo z
```

```
int dist = (int) Dist; // converte o dado flutuante em inteiro
```

```
int alpha = (int) Alpha; // converte o dado flutuante em inteiro
```

```
int beta = (int) Beta; // converte o dado flutuante em inteiro
```

```
if(distances[alpha][beta] == -1) // a posição atual ainda está em seu valor inicial?
```

```
{ // sim
```

```
    x_aux = x; // salva eixo x em uma variável auxiliar
```

```
    y_aux = y; // salva eixo y em uma variável auxiliar
```

```
    z_aux = z; // salva eixo z em uma variável auxiliar
```

```
    dist_aux = dist; // salva a distância em uma variável auxiliar
```

```
    distances[alpha][beta] = dist; // a posição recebe o valor da distância
```

```
}
```

```
else
```

```
{ // não
```

```
    if (dist_aux < dist) // a nova distância medida é maior do que a anterior?
```

```
        // sim
```

```
        cloud.add(new Vec3D(x_aux,y_aux,z_aux)); // descarta a nova distância medida
```

```
    else
```

```
        //sim
```

```
        cloud.add(new Vec3D(x,y,z)); // salva a nova distância medida
```

```
}
```

```
// condição de parada do mapeamento

if (Beta >= 50)

    scanOver = true;

    }

    }

}

}

void plot()

{

    for(int i = 0; i < cloud.size(); i++)

        gfx.point(cloud.get(i)); // Adiciona pontos da nuvem no espaço 3D

}

void eixos()

{

    // eixo X, vermelho

    stroke(255, 0, 0);

    line(-500, 0, 0, 500, 0, 0);

    // eixo Y, verde

    stroke(0, 255, 0);
```

```
line(0, -500, 0, 0, 500, 0);
```

```
// eixo Z, azul
```

```
stroke(0, 0, 255);
```

```
line(0, 0, -500, 0, 0, 500);
```

```
}
```