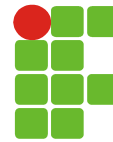




**INSTITUTO FEDERAL DE EDUCAÇÃO,
CIÊNCIA E TECNOLOGIA DO AMAZONAS
CAMPUS MANAUS DISTRITO INDUSTRIAL**



INSTITUTO FEDERAL
AMAZONAS
Campus Manaus-Distrito Industrial

CARLOS DOUGLAS ANDRE E SILVA

**MONITORAMENTO REMOTO POR MEIO DE UM RADAR BASEADO EM
ARDUINO**

**MANAUS/AM
2021**

CARLOS DOUGLAS ANDRE E SILVA

**MONITORAMENTO REMOTO POR MEIO DE UM RADAR BASEADO EM
ARDUINO**

Monografia apresentada à banca examinadora do Campus Manaus Distrito Industrial-CMDI, do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas IFAM, como requisito parcial para obtenção do grau de Tecnólogo em Mecatrônica.

Orientador: DSc. Vitor Bremgartner da Frota

MANAUS/AM

2021

Silva, Carlos Douglas André e
Monitoramento remoto por meio de um radar baseado em Arduino.
Manaus: IFAM/CMDI, 2021.

45 f. Color 10 cm.

Orientador:
Monografia (Trabalho de Conclusão de Curso) - Curso de Tecnologia em
Mecatrônica Industrial. IFAM, Manaus, 2021.

1. Arduino 2. Sensor Ultrassônico 3. Automação Residencial I. Carlos
Douglas André e Silva

CDD-- 669.722

CARLOS DOUGLAS ANDRE E SILVA

**MONITORAMENTO REMOTO POR MEIO DE UM RADAR BASEADO EM
ARDUINO**

Defesa em:

Manaus, 12 de maio de 2021.

BANCA EXAMINADORA

Vitor Bremgartner da Frota

Orientador: Dr. Vitor Bremgartner da Frota

Micila S.M. Pereira

1º Examinador: Micila Sumaria Medeiros Pereira

Priscila Silva Fernandes

2º Examinador: Me. Priscila Silva Fernandes

MANAUS/AM

2021

AGRADECIMENTOS

Agradeço, primeiramente, à Deus, que me deu forças para concluir esse trabalho. Ao nosso professor orientador, DSc. Vítor Bremgartner da Frota, pelo auxílio e dedicação que contribuíram para a realização desse projeto. A todos os professores que me ensinaram e dedicaram seu tempo para me desenvolver como cidadã e profissional. Agradeço, também meus colegas de curso pela parceria ao longo dessa jornada. E a minha família por toda estrutura proporcionada ao longo da minha vida

Dedico este trabalho primeiramente a Deus, por ser essencial em nossas vidas, por nos dar força e coragem durante esta longa caminhada. À nossa família, obrigada pelo carinho, paciência e compreensão.

A persistência é o caminho do êxito.

Charles Chaplin

RESUMO

A automação existe no nosso dia a dia, não só pela sua praticidade ou status, mas também relacionada à redução do consumo de recursos hídricos e energéticos, e à segurança e conforto que os usuários trazem aos usuários por meio da utilização de ambientes interligados. Entre os diversos tipos de automação, temos a automação residencial, que está relacionada ao uso de processos automatizados em casas, apartamentos e escritórios, principalmente para o bem-estar social e conforto de vida, como automação de ar condicionado, iluminação e segurança. O presente trabalho apresenta um radar baseado no Arduino, em que unindo o mesmo com componentes, diversos projetos de segurança doméstica podem ser realizados. Esta pesquisa também demonstra um processo de inserção de proteção de segurança residencial, ou seja, quando um intruso for detectado, mostrará na interface gráfica, em outro projeto os moradores da residência receberão uma mensagem em seu computador alertando sobre problemas em habitações.

Palavras-chave: Arduino, Sensor ultrassônico, Automação Residencial.

ABSTRACT

Automation exists in our daily lives, not only because of its practicality or status, but also related to the reduction in the consumption of water and energy resources, and the safety and comfort that users bring to users through the use of interconnected environments. Among the various types of automation, we have home automation, which is related to the use of automated processes in homes, apartments and offices, mainly for social welfare and comfort of life, such as air conditioning, lighting and security automation. The present work presents an Arduino-based radar, in which by uniting it with components, several home security projects can be carried out. This research also demonstrates a process of inserting residential security protection, that is, when an intruder is detected, it will show on the graphical interface, in another project the residents of the residence will receive a message on their computer alerting them about problems in housing.

Keyword: Arduino, Ultrasonic sensor, Home Automation.

LISTA DE QUADROS

Quadro 1: Quadro.	35
------------------------	----

LISTA DE FIGURAS

Figura 1: Arduino UNO.....	17
Figura 2: IDE Arduino.....	18
Figura 3: principais funções.....	18
Figura 4: Servo motor.....	19
Figura 5: Funcionamento do sensor ultrassônico.....	21
Figura 6: Módulo HC-SR04.	22
Figura 7: Pré montagem do projeto.....	26
Figura 8: Diagrama esquemático do projeto.....	28
Figura 9: Esquema de montagem do Radar.....	29
Figura 10 Exemplo gráfico visual processing.	30
Figura 11: Ambiente de medição com um objeto no ambiente.....	31
Figura 12: Ambiente de medição com dois objetos no ambiente.	32
Figura 13: Ambiente de medição com dois objetos no ambiente com um objeto mais próximo do sensor e o outro mais distante.....	33
Figura 14: Representação gráfica sem objeto presente no ambiente durante a varredura.....	34

LISTA DE ABREVEATURAS

AVR	Automatic Voltage Regulator
cm	Centímetro
Hz	hertz
IDE	Integrated Development Environment
kHz	quiloherzt
°C	Celsius
PCI	placa de circuito impresso
PWM	Pulse Width Modulation
USB	Universal Serial Bus

SUMÁRIO

1. INTRODUÇÃO	14
1.1. Objetivos	15
1.1.1. Objetivo Geral	15
1.1.2. Objetivos Específicos	15
1.2. Justificativa	16
1.3. Estrutura no trabalho	16
2. FUNDAMENTAÇÃO TEÓRICA	17
2.1. Arduino UNO	17
2.2. Software IDE Arduino	17
2.3. Interface gráfica	19
2.4. Servo Motor	19
2.5. Aspectos de mapeamento tridimensional	20
2.6. Sensores ultrassônicos	20
2.6.1. Influência dos materiais nos ensaios de ultrassom	23
2.7. Transdutores ultrassônicos	24
2.8. Microcontrolador Arduino	24
3. MATERIAIS E MÉTODOS	26
4. RESULTADOS E DISCUSSÃO	31
4.1. Ambiente delimitado com um obstáculo presente	31
4.2. Ambiente delimitado com dois obstáculos presentes	32
4.3. Ambiente delimitado com dois obstáculos presentes em posições mais distantes	33
4.4. Interpretação dos testes	34
4.5. Custo do projeto	35
5. CONCLUSÃO	37
6. REFERÊNCIAS	38
APÊNDICE A: Código Processing	41
APÊNDICE B: Radar Arduino	44

1. INTRODUÇÃO

No início do século XX, os países perceberam o potencial que os sistemas de radar podem ter no cenário de guerra. Essa demanda e o estudo do tempo levaram ao rápido desenvolvimento dessa tecnologia na Segunda Guerra Mundial (ÁVILA; MARTINS; CEPIK, 2009). O radar é um sistema de ondas de rádio direcional que pode determinar o alcance, direção, altitude e velocidade dos objetos e sua aplicação está diretamente relacionada à meteorologia, vigilância costeira, controle de tráfego aéreo comercial, aeroespacial e até mesmo a indústria aeroespacial (PÉREZ, 2011).

Dessa forma, a pesquisa em tecnologia de robôs móveis é um tema muito relevante e atual e, nas últimas duas décadas, a pesquisa e o desenvolvimento neste campo deram um grande salto em seu desenvolvimento. Robótica é uma ciência multidisciplinar que reúne vários campos científicos, como mecânica, eletrônica, física, visão artificial de computador, economia, teoria de controle de sistema, etc (WOLF et al., 2009)

Segundo Evan, Noble e Hochenbaum (2013) a automação no Brasil e no mundo está passando por um grande crescimento, onde no passado era um tema que se limitava à classe de menor poder aquisitivo, esse panorama está em mudança. Tófoli (2014) afirma que o número de furtos domiciliares vem aumentando a cada dia, o que torna a segurança residencial um problema comum a todas as classes sociais no Brasil.

Dessa forma, à medida que mais aquisições tecnológicas se espalham pelo globo de forma mais democrática, as pesquisas, que antes eram realizadas apenas em ambientes industriais, acadêmicos e no cotidiano estão se tornando cada vez mais populares, o que significa, mais e mais pessoas podem participar e se interessar. Neste caso, os problemas clássicos de engenharia e computação estão começando a ser desafiados por mais e mais pessoas, então novas soluções são mais prováveis (MARTINS *et al.*, 2015).

Nesse contexto, o termo ultrassom é usado para definir ondas sonoras, cuja frequência é superior à audibilidade humana, ou seja, mais de 20.000 ciclos por segundo. Também conhecido como sinal ultrassônico, as ondas sonoras inaudíveis são capturadas por um transdutor, que converte a energia do som em energia

elétrica devido à ressonância e vice-versa (LEE, 2010). Dessa maneira temos a seguinte problemática: Como desenvolver um radar com monitoramento remoto baseado em Arduino?

Portanto, este trabalho apresenta um sistema embarcado com foco no processamento de dados, por meio de integração hardwares e softwares. O processo consiste em hardware e software (respectivamente a estrutura física e digital do sistema de computador). No qual o hardware consiste em um sensor e um atuador e tem por objetivo coletar esses dados de objetos em um ambiente aberto. E o software compreende uma interface gráfica, que pode interpretar e exibir o conteúdo lido pelo hardware na tela do computador por meio do sistema de microcontrole.

Neste trabalho, são utilizados um sensor ultrassônico, um servo motor, um microcontrolador e uma plataforma de desenvolvimento livre, ou seja, Arduino a *Integrated Development Environment* (IDE) e a interface gráfica Processing 3.

Portanto, o sistema arduino quando bem utilizados, empregando materiais de fácil manuseio, instalação e apresenta uma viabilidade econômica para a pessoa que deseja desenvolver o projeto. Para Oliveira (2013) o microcontrolador que atende a todos esses requisitos é o Arduino e seus componentes. Além disso, tutoriais e manuais estão disponíveis em vários idiomas, até mesmo em português, para que possam ser usados em diferentes projetos.

1.1. Objetivos

1.1.1. Objetivo Geral

O presente trabalho tem como objetivo monitorar um ambiente por meio de um radar feito com processamento de um Arduino.

1.1.2. Objetivos Específicos

- Realizar o monitoramento de um ambiente por um baixo custo;
- Contribuir na segurança com a utilização do radar;

- Permitir o monitoramento de ambientes de forma facilmente acessível para o usuário.

1.2. Justificativa

Com o objetivo de atrair atenção para o tema, com a finalidade do desenvolvimento do Arduino é ser fácil de acessar, flexível e fácil de usar, e seu valor é inferior ao de uma plataforma de prototipagem mais poderosa. O arduino emprega em suas linguagens de programação baseadas em C e C ++, permitindo ler e enviar dados através das portas digitais ou analógicas integradas.

1.3. Estrutura no trabalho

Após este capítulo introdutório, segue o Capítulo 2, onde apresenta os aspectos fundamentais sobre o Arduino UNO, Software IDE Arduino e sensores ultrassônicos. No Capítulo 3 (três), são expostos materiais e métodos utilizados para o desenvolvimento do presente projeto. No Capítulo 4 (quatro), são evidenciados os resultados e discussões sobre o projeto, finalizando com a conclusão acerca do trabalho.

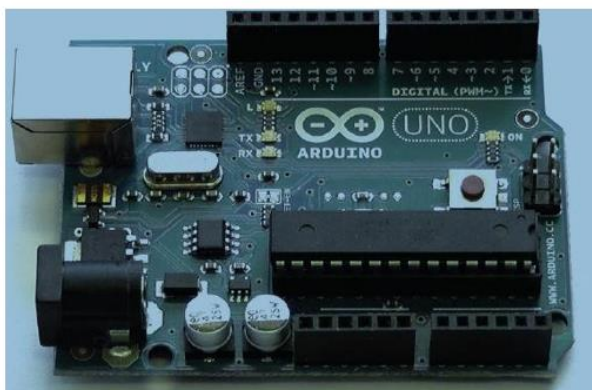
2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, os elementos utilizados neste trabalho e os aspectos teóricos da propagação de ondas sonoras serão descritos. Além disso, serão descritos os efeitos de diferentes tipos de materiais em testes ultrassônicos e a operação de transdutores ultrassônicos

2.1. Arduino UNO

O Arduino UNO é uma placa de desenvolvimento de hardware aberto (traduzido como plataforma física de desenvolvimento livre) baseada no microcontrolador. Dessa forma o Arduino UNO é o modelo mais vendido no mundo e grande parte de seu sucesso se deve ao seu baixo custo e recursos excepcionais. Uno significa uma palavra em italiano, é a versão 1.0 do Arduino, ou seja, o sucessor será baseado nesta placa. Ele contém um chip Atmega8U2, a principal vantagem de seus ancestrais é suportar atualizações de firmware USB. Desta forma, o Arduino aparecerá como outro dispositivo no computador utilizado (ARDUÍNO, 2015).

Figura 1: Arduino UNO.

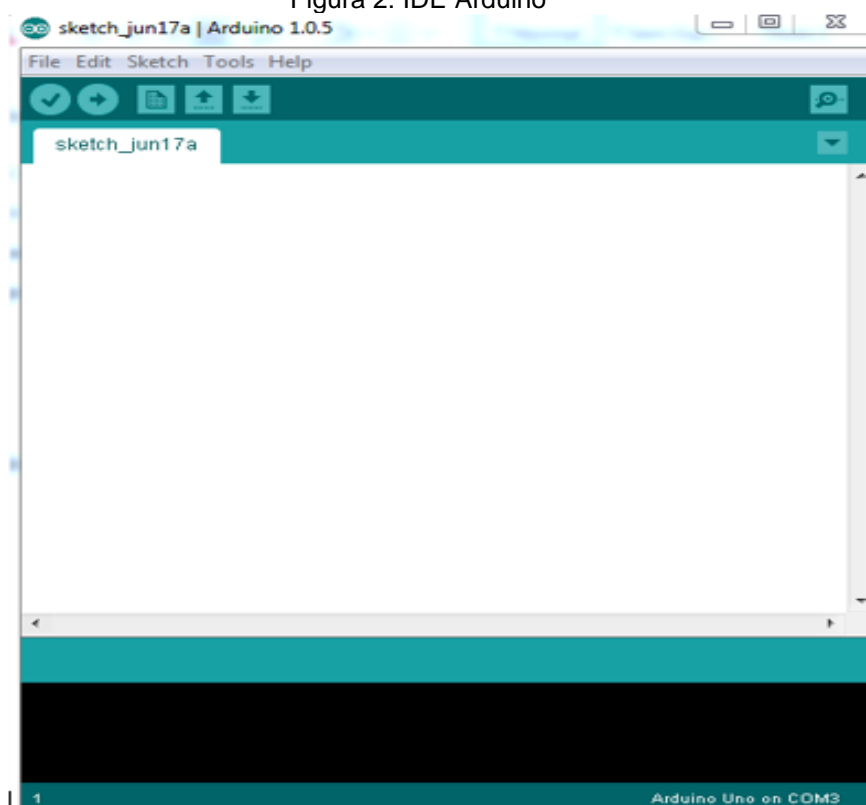


Fonte: Arduíno, 2015.

2.2. Software IDE Arduino

Para que o Arduino carregue programas e execute as operações de acordo, o Arduino IDE é usado. O software é um editor de texto (apêndice B) que pode ser conectado ao hardware do Arduino para verificar se a programação está correta e fazer o upload para a placa correspondente. A Figura 2 mostra a interface do software, a parte branca do programa de desenvolvimento e a parte preta da notificação.

Figura 2: IDE Arduino



Fonte:Print do software.

A Figura 3 mostra os botões principais e suas definições.

Figura 3: principais funções.



Fonte:Print do software.

A sintaxe da linguagem Arduino, derivada do *Wiring* e desenvolvida por Hernando *Barragan*, é basicamente C / C ++ com funções simples e específicas para uso com a porta Arduino. Seu funcionamento requer duas funções básicas: `setup ()` e `loop ()` (ARDUINO, 2016).

2.3. Interface gráfica

Processing também é uma linguagem de programação de código aberto, desenvolvida para as comunidades de arte eletrônica e design visual. O projeto foi iniciado em 2011 por *Casey Reas* e *Bem Fry*, ex-integrantes do Grupo de Computação do MIT Media Lab. Seu desenvolvimento é baseado na ideia de aprender visualmente o contexto da programação. Portanto, o *software* gera imagens por meio de comandos pré-estabelecidos e pode ter função de interface gráfica (TEIXEIRA, 2011).

2.4. Servo Motor

Um servo motor é um atuador rotativo, que é composto por um motor conectado a uma engrenagem de redução e um circuito especialmente projetado para servos motores. Desta forma, sua posição angular, velocidade e aceleração podem ser controladas com muita precisão (KUSRIYANTO; PUTRA, 2016).

Figura 4: Servo motor.



Fonte: Tower Pro (2019).

Controlado por pulsos de onda retangulares modulados em *Pulse Width Modulation* (PWM), a largura desses pulsos controla a posição angular do servo motor. De acordo com as especificações da folha de dados, o servo motor requer uma frequência operacional de 50 Hz, e a largura do pulso deve variar entre 1 milissegundo e 2 milissegundos, resultando em uma mudança de 180 graus no ângulo (ALSHAMSI et al., 2020).

2.5. Aspectos de mapeamento tridimensional

O teste ultrassônico é realizado por pulsos curtos de energia sonora emitida pelo transdutor. O pulso viaja na velocidade do som e, quando detecta um obstáculo, parte do pulso é refletido de volta para a fonte como um eco. Ao armazenar cada ponto ao longo do tempo, as informações sobre os obstáculos atuais podem ser obtidas em várias posições angulares controladas pelo sistema de micro-controle (SOUZA; OLIVEIRA, 2012).

2.6. Sensores ultrassônicos

O termo ultrassom é usado para definir ondas sonoras, cuja frequência é superior à audibilidade humana, ou seja, superior a 20.000 Hz. Também conhecido como sinal ultrassônico, as ondas sonoras inaudíveis são capturadas pelo transdutor, que converte a energia sonora em energia elétrica devido à ressonância e vice-versa (TITTON; ROBOLDI; ROSITO, 2016).

Sensores ultrassônicos são normalmente usados para medir distâncias e detectar obstáculos, sejam no estado líquido ou sólido. Esses sensores podem detectar objetos como metal, madeira, concreto, plástico, produtos de borracha e objetos transparentes. Ao contrário dos sensores infravermelhos, os sensores ultrassônicos não requerem informações sobre as características da superfície do objeto (THOMAZINI, 2005).

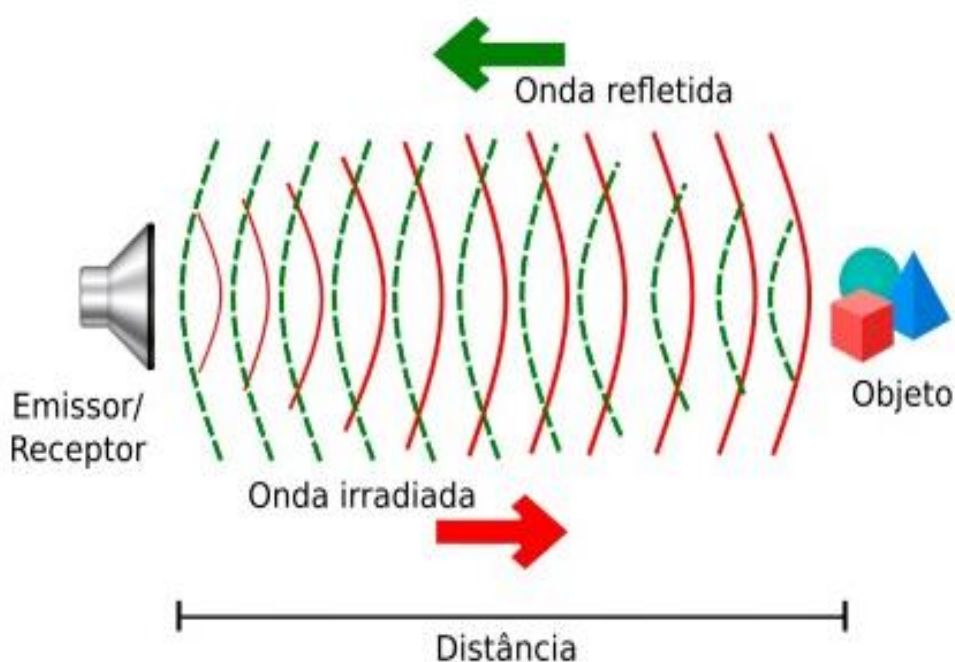
Os sensores ultrassônicos são compostos de dispositivos eletrônicos chamados transdutores ultrassônicos. O transdutor é usado para transmitir e absorver pulsos ultrassônicos em uma frequência específica. Quando o sensor irradia um sinal de pulso audível para o objeto, o processo de medição é iniciado. Em resposta ao estímulo sonoro, o sensor recebe um sinal sobre o reflexo da onda na superfície do objeto. Quando um sinal com a mesma frequência de pulso é recebido, a distância do pulso refletido de um determinado objeto até a parte traseira refletida do objeto pode ser calculada (THOMAZINI, 2005).

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. 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 (LORENZI et al., 2011).

A distância detectada pelo sistema de mapeamento 3D é baseada no tempo de propagação do pulso ultrassônico. Para medir essa distância, o transdutor emite um sinal em uma frequência específica. O módulo de variação mede o tempo necessário para que o pulso de som refletido retorne ao transmissor. Além disso, o tempo total de medição é estendido pelo tempo de espera eletrônico (ou seja, o tempo necessário para iniciar e parar a vibração da membrana do componente do oscilador) e processamento de cálculo. Portanto, o erro de medição típico relativo especificado pelo fabricante do sistema de medição de som é de cerca de 1% (FERREIRA et al., 2014).

Figura 5: Funcionamento do sensor ultrassônico.



Fonte: Autocore Robótica, 2020.

De acordo com o tempo decorrido entre a transmissão e recepção do pulso, a distância até o objeto pode ser determinada, com base na fórmula 1.

$$d = \frac{v \cdot t}{2} \quad (1)$$

Onde,

d: distância do sensor ultrassônico até o obstáculo; v: velocidade do som; e

t: tempo de resposta.

O dispositivo consiste em um transmissor e um receptor, capaz de medir uma distância de 2 cm a 4 metros, com uma precisão de cerca de 3 mm e um ângulo de visão de 15 graus. Além disso, o sensor HC-SR04 também possui um circuito pronto e 4 pinos: fonte de alimentação, gatilho (transmissor de pulso), eco (receptor de pulso) e referência.

Figura 6: Módulo HC-SR04.



Fonte: Cytron Technologies (2019).

De acordo com o diagrama de tempo operacional do sensor ultrassônico, quando um sinal de 10 microssegundos é recebido no pino de transmissão com uma voltagem de 5 volts, o módulo enviará oito pulsos ultrassônicos a uma frequência de 40 kHz. Posteriormente, o sensor capta o reflexo do som, que, após ser processado pelo circuito controlador, gera um sinal de resposta no pino receptor.

A duração desta resposta em 5 volts é proporcional à distância medida. Conforme discutido no referencial teórico, o aspecto físico do material inspecionado afeta a reflexão do som emitido para o alvo de interesse.

No entanto, por ser um sistema composto por muitos pontos de análise, certas medidas de variância não alteram significativamente o mapeamento. Além

disso, a presente invenção pretende trabalhar com objetos cujas características de composição são conhecidas.

2.6.1. Influência dos materiais nos ensaios de ultrassom

A propagação das ondas ultrassônicas está sempre relacionada aos módulos de rigidez (G) e elasticidade (E) dos materiais que propagam essas ondas ultrassônicas, pois determinam, respectivamente, a capacidade das partículas de resistir à deformação elástica causada pelos pulsos ultrassônicos. Portanto, o coeficiente de Poisson (em homenagem a Siméon Denis Poisson e formulado por ele) determina a deformação lateral, que é causada pela carga aplicada longitudinalmente (SARRO, 2017).

Depois de compilar as propriedades físicas do material, será mais fácil perceber a diferença na velocidade de propagação das ondas entre polímeros, metais e outros materiais. Além disso, dependendo da temperatura de operação, a diferença pode aumentar ou diminuir. Para cada diferença de 50°C, os metais tendem a perder cerca de 1% da velocidade de propagação da onda sonora, enquanto nos polímeros essa perda pode aumentar em até 4% (ALVES; CARASEK; CASCUDO, 2010).

A enorme diferença nas velocidades de ondas mecânicas entre polímeros e metais, e as diferentes sensibilidades dessas velocidades às mudanças de temperatura, apresentam dois obstáculos para a aplicação do ultrassom em componentes compostos por essas camadas de materiais (GODINHO et al., 2018).

O primeiro obstáculo está relacionado ao coeficiente de transmissão da onda mecânica na interface desses materiais. Quanto maior a diferença entre as velocidades longitudinais dos materiais, menor será a intensidade do som que será transmitido pela interface (GODINHO et al., 2018).

O segundo obstáculo é que ao passar pela interface do material com grande diferença na velocidade do som, o feixe sonoro não só perde energia, mas seu ângulo de abertura também fica distorcido. Os sistemas de ultrassom que lidam com camadas sobrepostas de metal e polímero devem considerar as bandas de frequência onde essas distorções podem ser conhecidas e controladas.

2.7. Transdutores ultrassônicos

De acordo com Alvarenga, Machado e Pereira (2001) os transdutores ultrassônicos usam a deformação de cristais piezoelétricos não apenas para gerar energia acústica no elemento inspecionado, mas também coletar a energia restante que passa pelo elemento inspecionado, mas não é consumida por ele. Especificamente para este trabalho, a técnica de medição utilizada é denominada pulso-eco. A técnica depende do nível de precisão da medição e da possibilidade de acesso ao material a ser inspecionado:

a) Transdutores de emissão dedicados, ou seja, esses transdutores apenas excitarão o material inspecionado e transmitirão energia sonora para ele;

b) Transdutores receptores dedicados, ou seja, esses transdutores são excitados apenas pelas ondas sonoras que passam ou são refletidas pelo material a ser testado. Nesse caso, ele está localizado na mesma superfície do transdutor de transmissão.

c) Transdutor de cristal piezoelétrico duplo, dois cristais piezoelétricos são instalados na caixa para formar os mesmos pontos de transmissão e recepção

Todos os transdutores podem ser excitados em uma frequência determinada pelo equipamento de ultrassom, ou simplesmente fazer contato com a parte inspecionada e aguardar as ondas sonoras refratadas ou refletidas pelo material de contagem. Desta forma, além dos já citados, ainda é possível a montagem, onde um único transdutor de cristal piezoelétrico é excitado pelo aparelho ultrassônico por um determinado tempo e permanece estacionário pelo próximo intervalo programado. Desta forma, o transdutor pode realizar um ciclo combinado de ondas de emissão e ondas sonoras, e ler seu eco no mesmo local (RODRIGUES, 2020).

2.8. Microcontrolador Arduino

Os microcontroladores existem em todos os dispositivos que envolvem dispositivos eletrônicos devido ao seu tamanho pequeno, excelente desempenho, baixo consumo de energia e grande relação custo-benefício. São placas com microprocessadores que servem para analisar instruções pré-definidas e realizar tarefas, memória utilizada para armazenar dados, pinos de entrada e saída, além de temporizadores, comunicações seriais, resistores, estes O objetivo dos pinos é

permitir que o microcontrolador se comunique com o sensor ou atuador (MCROBERTS, 2018). Segundo Martins (2005) existem vários tipos de microcontroladores. O que os diferencia são:

- Velocidade do processamento;
- Quantidade de memória interna disponível para armazenar dados;
- Quantidade de pinos de entrada e saída;
- Fonte de alimentação;
- Tipos e as quantidades de periféricos;
- Arquitetura do circuito e
- Conjunto de instruções disponibilizado nos circuitos internos.

Arduino é um dos tipos de microcontroladores mais comumente usados e pesquisados. Por ser uma plataforma muito fácil de usar em projetos e por conter uma grande comunidade de usuários. Nesse tipo de caso é utilizado o microcontrolador AVR. Segundo a Atmel, é fácil de usar, tem baixo consumo de energia e alta integração. Os microcontroladores Atmel®AVR® 8 e 32 bits são baseados em microcontroladores O complemento do processador e microprocessador ARM®SMART da Atmel, que combina perfeitamente desempenho, eficiência energética e flexibilidade de design (ATMEL, 2016).

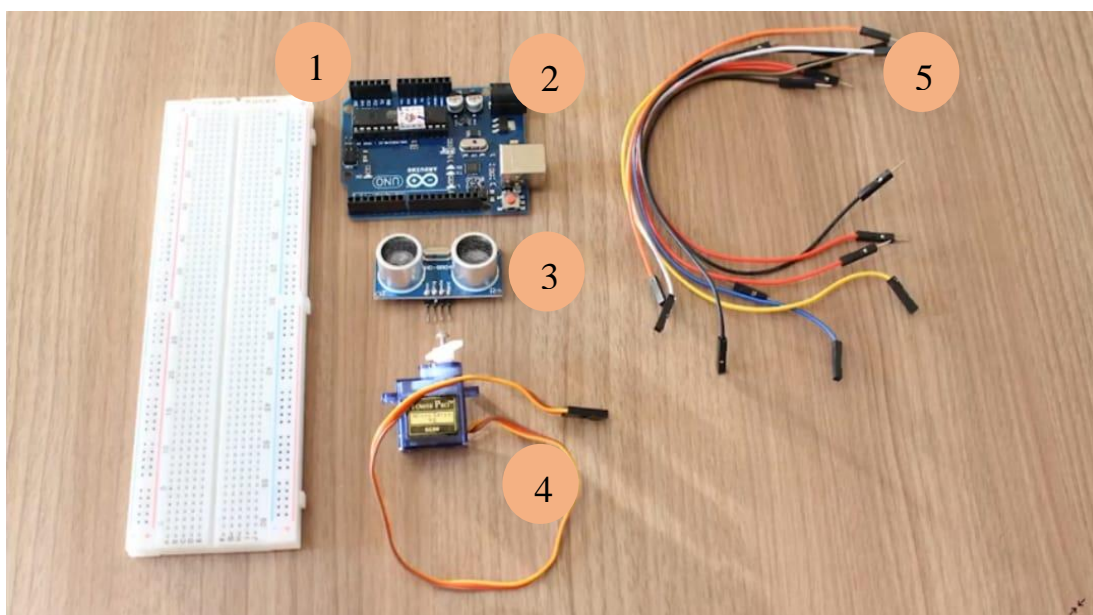
Projeto foi iniciado por dois professores de informática, cuja ideia era imaginar um microcontrolador de fácil acesso e baixo custo, que foi idealizado no Ivrea Interactive Design Institute em 2005. O Prix Ars Electronica recebeu uma menção honrosa na categoria comunidade digital em 2006, que marcou o seu sucesso. Além disso, a marca vendeu mais de 50.000 cartazes até outubro de 2008 (EVANS; BOLE; HOCHENBAUM, 2013).

Uma das características salientes da placa é o seu "*open source*", o que significa que software e hardware são conceitos livres. Existe um grande fórum de debates na Internet, onde muitas pessoas comentam, ajudam e distribuem o código de programação e design de circuitos para que possam ser melhorados a qualquer momento. Em parte, isso se deve ao grande número de comunidades em torno da plataforma, mas também porque o Arduino foi projetado desde o início para ter uma interface simples para pessoas sem habilidades de programação (BANZI; SHILOH, 2011).

3. MATERIAIS E MÉTODOS

Este capítulo discute o modelo de protótipo proposto e suas características. A primeira parte foi pesquisar e obter a melhor placa micro controladora, escudo e sensor para efetivamente atingir o objetivo. A segunda parte são os componentes do circuito usados para conectar a placa Arduino UNO, a blindagem Ethernet e o sensor. A última é programar por meio do software Arduino IDE, de acordo com a figura 7.

Figura 7: Pré montagem do projeto.



Fonte: Arduino Portual, 2020.

As descrições dos itens estão apresentadas abaixo:

1 – Protoboard: uma matriz de contato, é uma placa com furos de conexões condutoras que serviu como base para a montagem dos circuitos elétricos.

2 – Arduino uno: uma plataforma de prototipagem eletrônica de hardware livre e de placa única, onde foram feitas as conexões do sensor ultrassônico, servo motor e protoboard através dos jumpers, usando uma linguagem de programação padrão.

3 – Sensor ultrassônico: O funcionamento do HC-SR04 se baseia no envio de sinais ultrassônicos pelo sensor, que aguarda o retorno do sinal, e com base no tempo entre envio e retorno, calcula a distância entre o sensor e o objeto detectado.

4 –Servo motor: uma máquina, eletromecânica, que apresenta movimento proporcional a um comando, ou seja: recebe um sinal de controle; que verifica a posição atual para controlar o seu movimento indo para a posição desejada definida pelo programa. No caso o servo motor irá girar de 15 a 165 graus como definida na programação (Apêndice B).

5 – Jumpers: pequeno condutor utilizado para conectar dois pontos de um circuito eletrônico, que no caso serviu para realizar as conexões do servo motor, Sensor ultrassônico, Arduino Uno e protoboard.

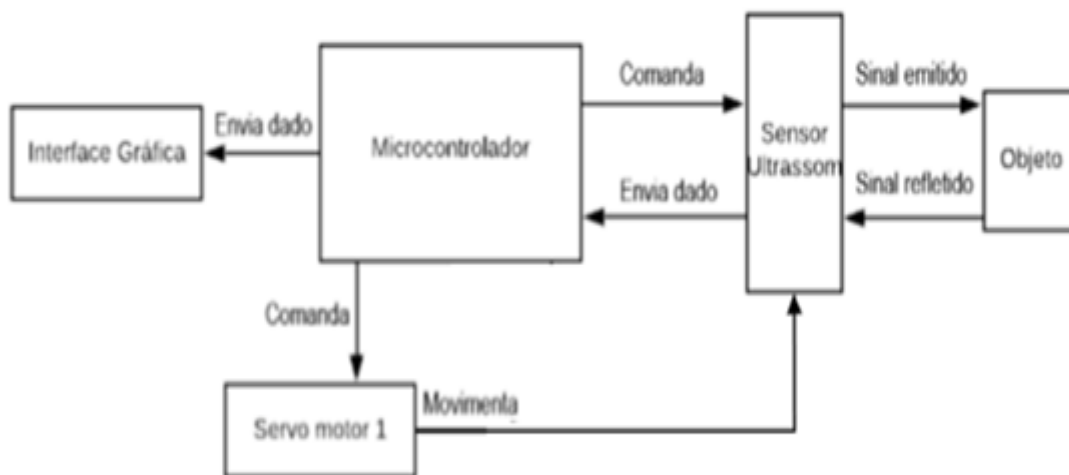
Para determinar a forma dos obstáculos existentes, é recomendável usar o deslocamento angular para fazer a varredura do sensor ultrassônico. A saída de informações do sensor é conectada a um microcontrolador, que também controla um servo, que converte a duração do pulso formado pelo sensor em uma distância em centímetros. Além disso, o microcontrolador é conectado ao computador por meio de um barramento USB (PERRETI, 2019).

O sinal do microcontrolador executa o software, que muda a direção do sensor ultrassônico no ambiente. O computador envia um sinal ao microcontrolador para alterar cuidadosamente o tom do sensor ultrassônico. Entre cada rotação, o sensor mede a distância até o obstáculo à sua frente (KLANN, 2017).

No diagrama esquemático do projeto apresentado na Figura 8, o microcontrolador comanda o servo motor para fazer a movimentação e como o sensor ultrassônico está em cima do servo motor ele se movimenta junto.

O microcontrolador comanda também o sensor ultrassônico a emitir um sinal, afim de localizar um objeto, e quando esse objeto é localizado o sinal é refletido para o sensor que envia esses dados para o microcontrolador que conseqüentemente envia essas informações para a interface gráfica do projeto.

Figura 8: Diagrama esquemático do projeto.



Fonte: Elaborado pelo acadêmico, 2020

Para todas as posições possíveis, duas informações são registradas para cada posição: a posição angular do servo motor e a distância medida em relação ao obstáculo. Ao associar esses dados a cada ponto no espaço, o ambiente pode ser mapeado por meio de coordenadas.

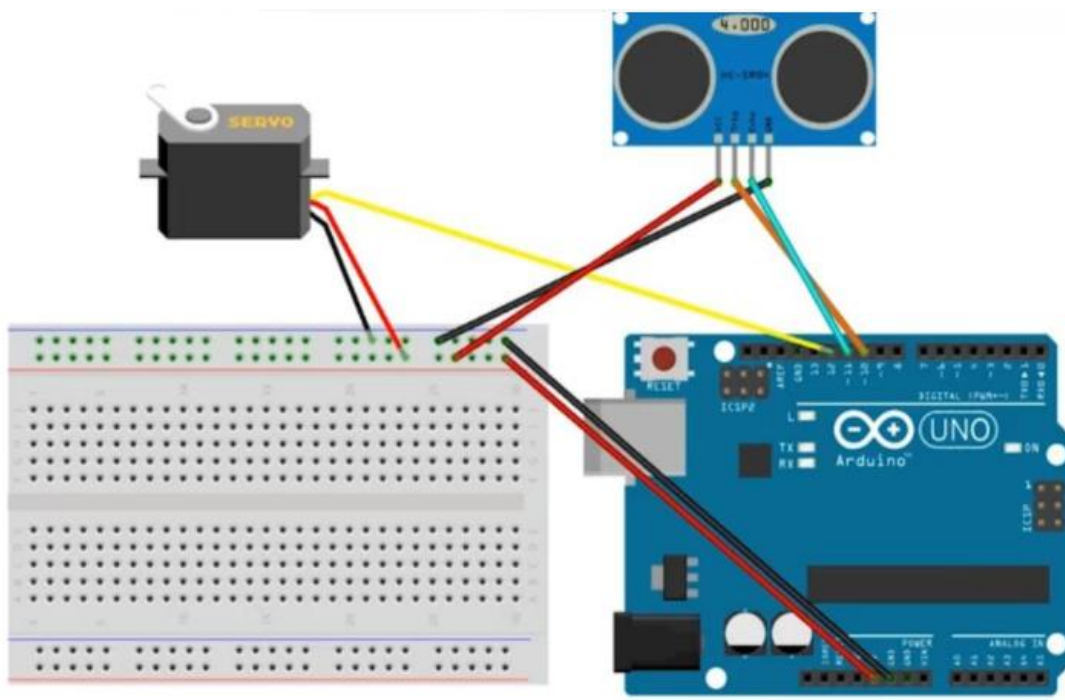
A movimentação do sistema e a aquisição dos dados correspondentes de cada ponto são realizadas em paralelo. Cada ponto obtido pela porta serial é convertido em coordenadas e armazenada. Após a leitura, o conjunto de pontos será desenhado na tela do computador.

Dessa forma o projeto visa calcular a distância e a posição do objeto colocado a alguma distância do sensor. O sensor ultrassônico envia a onda ultrassônica em diferentes direções girando com a ajuda do servo motor. Esta onda viaja no ar e obtém refletido de volta após atingir algum objeto.

Esta onda é novamente detectada pelo sensor e suas características são analisadas e a saída é exibida em tela mostrando parâmetros como distância e posição do objeto. Arduino IDE é usado para escrever código e fazer upload de codificação no Arduino e nos ajuda a detectar a posição do servo motor e postá-lo na serial porta junto com a distância do objeto mais próximo em seu caminho. A

saída do sensor é exibida com a ajuda do software de processamento para dar saída na tela de exibição.

Figura 9: Esquema de montagem do Radar.

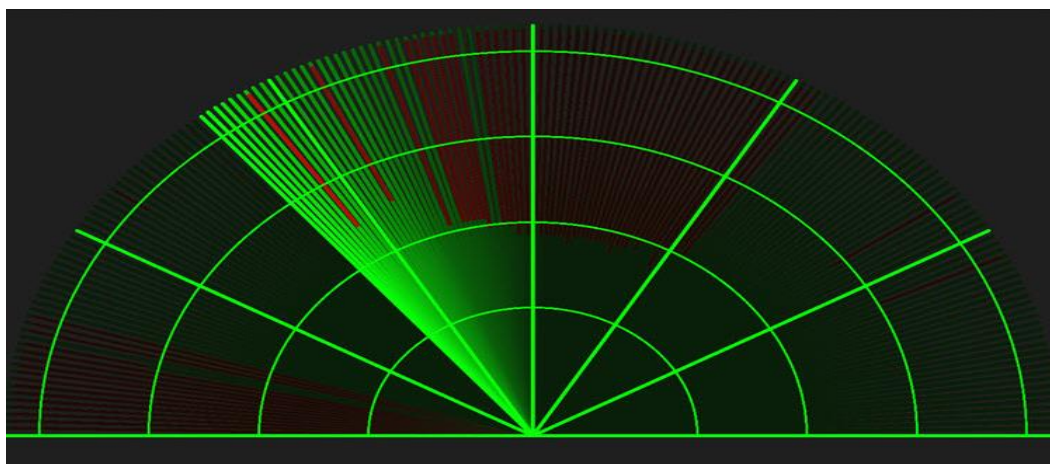


Fonte: Arduino Portual, 2020.

De acordo com a Figura 9, os componentes foram conectados ao Arduino, o qual administra o movimento do servo, realizando assim a leitura do sensor e envia o resultado para o Processing. O processing é um software que tem uma linguagem aberta de programação.

A cada leitura realizada pelo microcontrolador, uma *string* de caracteres (Apêndice A) é enviada para a interface gráfica através da porta serial, que contém os valores correspondentes de distância, ângulo horizontal do servo motor. Esses valores correspondentes serão convertidos para o gráfico mostrado na Figura 10. Os dados são armazenados no vetor até que o mapeamento seja concluído. Finalmente, a nuvem de pontos gerada é exibida na interface gráfica 3D (TEIXEIRA, 2011).

Figura 10 Exemplo gráfico visual processing.



Fonte: Software processing, 2020.

4. RESULTADOS E DISCUSSÃO

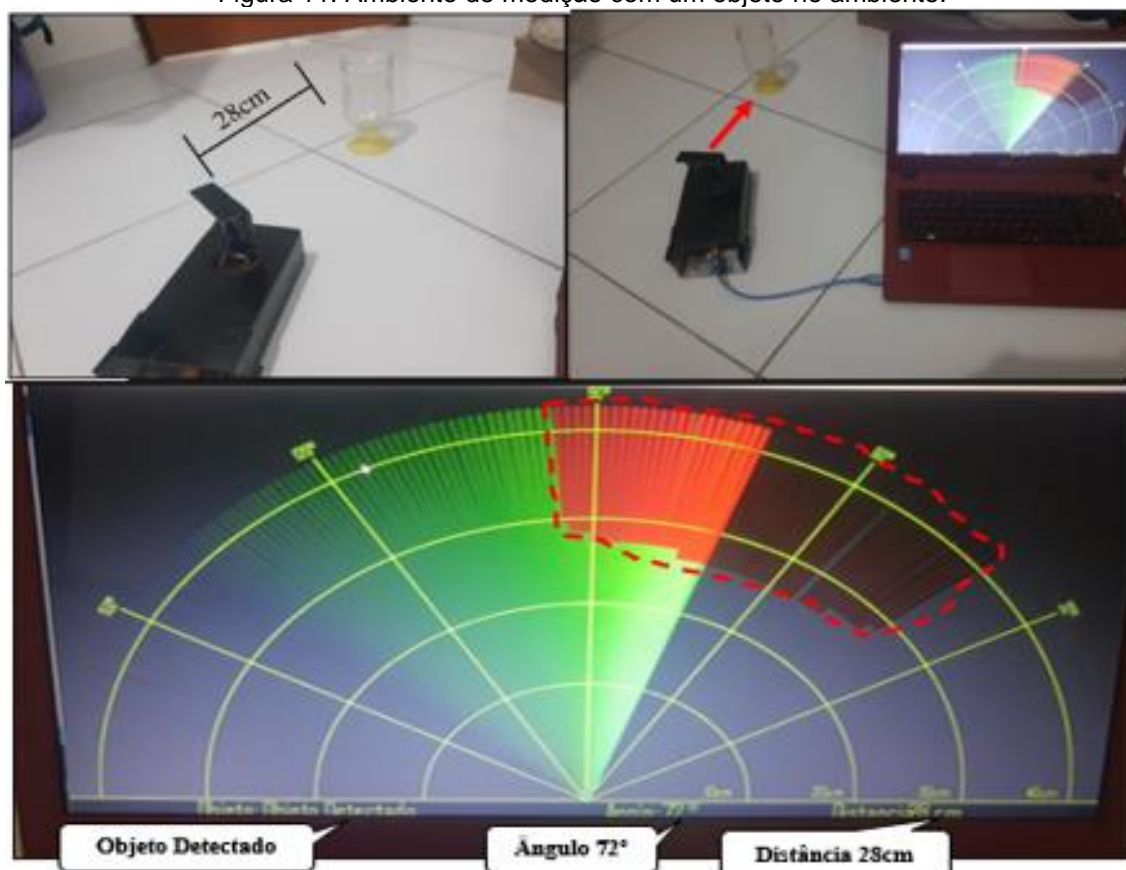
O presente capítulo apresenta os resultados experimentais obtidos pelo mapeador operando após a confecção da placa de circuito impresso.

4.1. Ambiente delimitado com um obstáculo presente

O teste realizado no ambiente, foi realizado com um copo de vidro à frente do mapeador (Figura 11) O objeto foi posicionado de forma vertical, para que a interface gráfica mostrasse em vermelho a posição do copo referente ao sensor.

A parte de vermelho demonstrada no gráfico indica que tem um objeto (copo de vidro) no radar, mostrando que há um objeto detectado em um ângulo de 72° em uma distância de 28cm em relação ao sensor, houve uma medição manual da distância do sensor ao copo que no gráfico resultou no mesmo valor medido manualmente, o restante do gráfico está verde, significando que o sensor não está localizando nenhum objeto durante seu mapeamento.

Figura 11: Ambiente de medição com um objeto no ambiente.

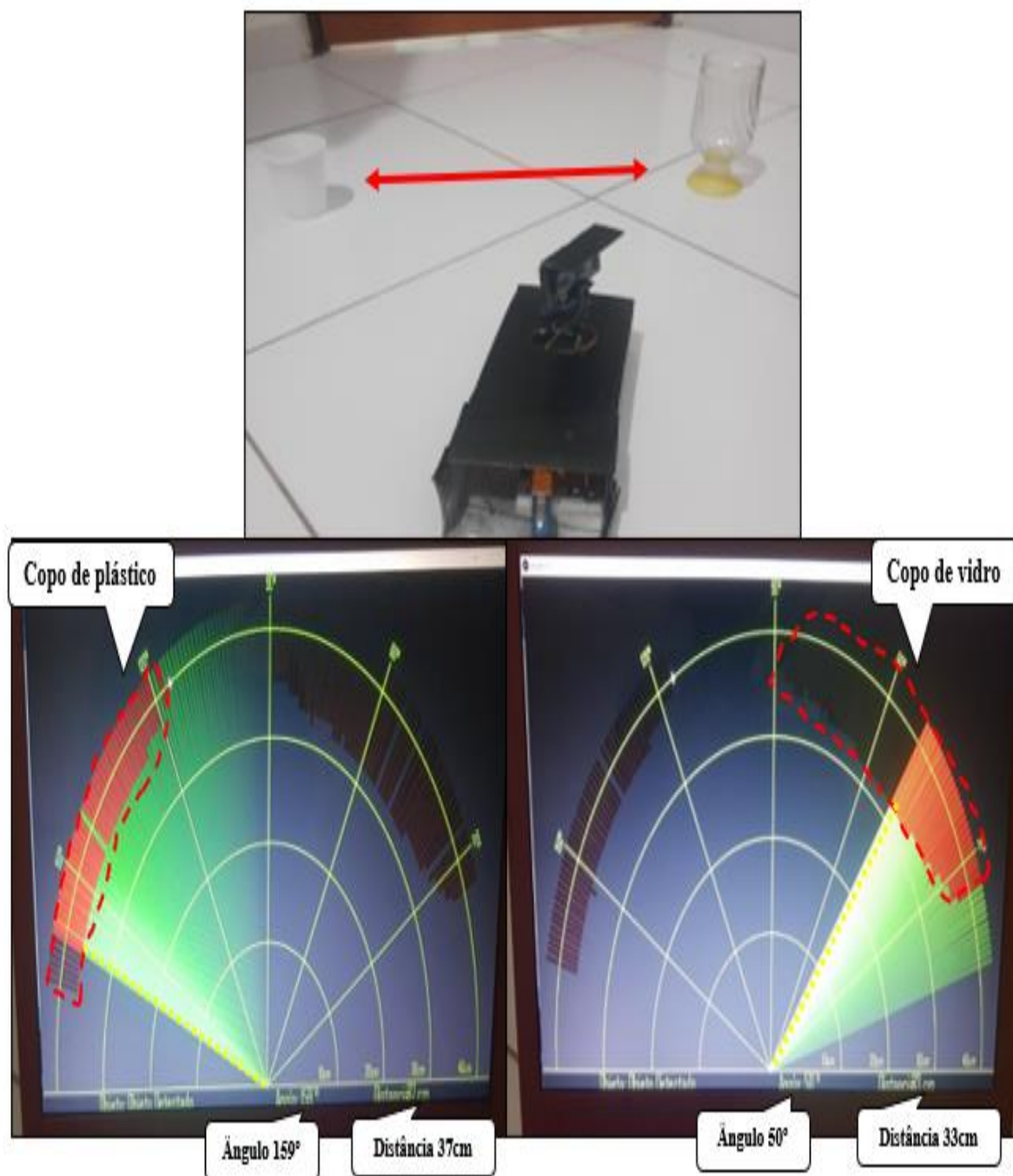


Fonte: Elaborado pelo autor, 2021.

4.2. Ambiente delimitado com dois obstáculos presentes

O segundo teste realizado no ambiente, foi realizado com um copo de vidro e um copo de plástico à frente do mapeador com distancias parecidas (Figura 12) O objeto foi posicionado de forma vertical, para que a interface gráfica mostrasse em vermelho a posição do copo referente ao sensor.

Figura 12: Ambiente de medição com dois objetos no ambiente.



Fonte: Elaborado pelo autor, 2021.

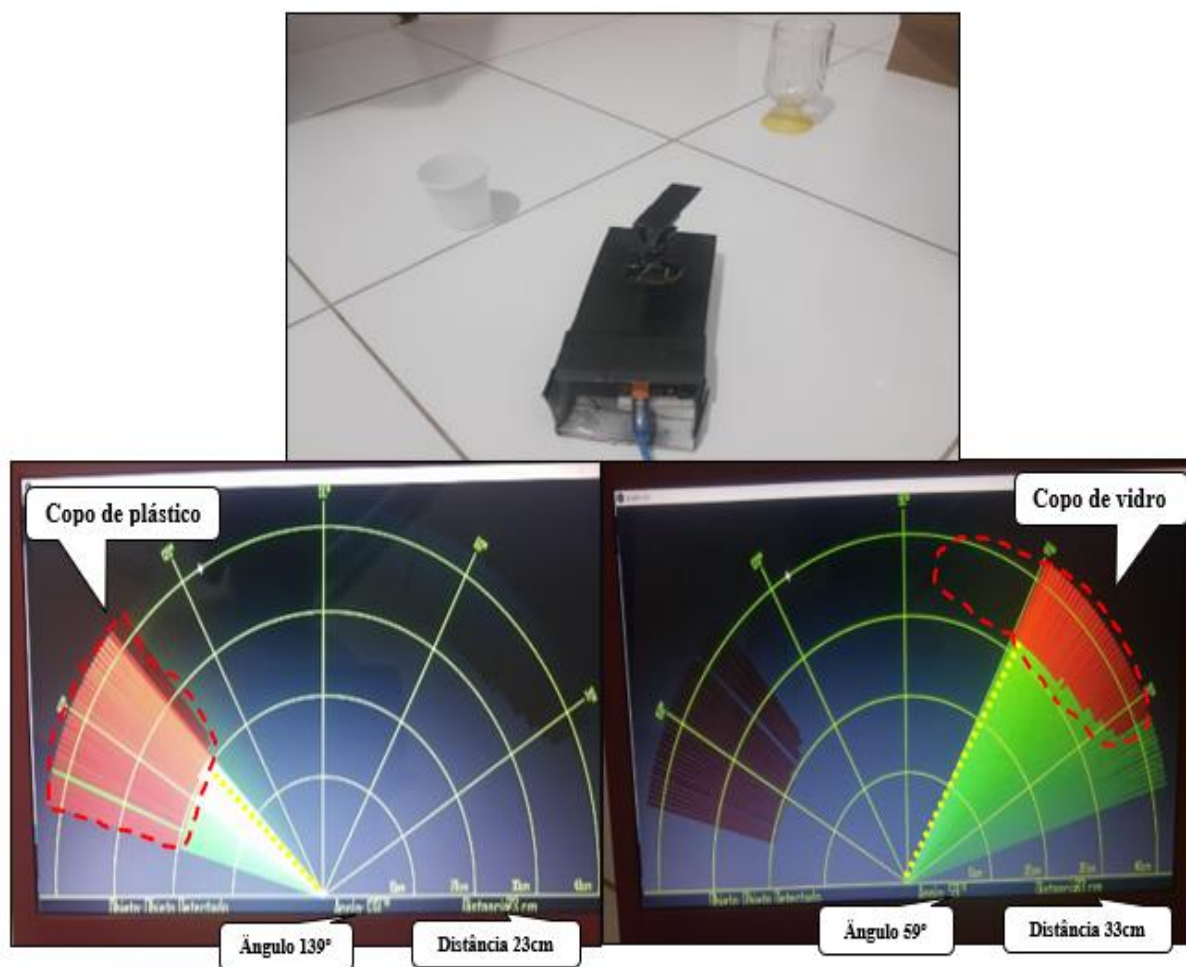
Existe duas partes vermelhas mostrado no gráfico, indicando que ao lado esquerdo há um objeto (copo de plástico) e ao lado direito há outro objeto (copo de vidro) no radar. Quando o mapeador atua pelo lado esquerdo, ele detecta um objeto que está a um ângulo de 159° e uma distância de 37cm em relação ao sensor, e quando o mapeador atua pelo lado direito ele detecta um objeto que está a um ângulo de 50° e uma distância de 33cm em relação ao sensor, o restante do gráfico está verde, significando que o sensor não está localizando nenhum objeto durante seu mapeamento.

4.3. Ambiente delimitado com dois obstáculos presentes em posições mais distantes

O terceiro teste realizado no ambiente, foi realizado com um copo de vidro e um copo de plástico à frente do mapeador, mas cada objeto com distancias diferentes (Figura 13). O objeto foi posicionado de forma horizontal, para que a interface gráfica mostrasse em vermelho a posição do copo referente ao sensor.

Existe duas partes vermelhas mostrado no gráfico, indicando que ao lado esquerdo há um objeto (copo de plástico) e ao lado direito há outro objeto (copo de vidro) no radar. A diferença em relação ao segundo teste, foi que, o copo de plástico foi posicionado mais próximo ao sensor e o copo de vidro mais distante do sensor, no gráfico percebe-se que pôr o copo de plástico está mais próximo ao sensor, ele abrange uma área maior em vermelho e uma distância mais curta.

Quando o mapeador atua pelo lado esquerdo, ele detecta um objeto que está a um ângulo de 139° e uma distância de 23cm em relação ao sensor, e quando o mapeador atua pelo lado direito ele detecta um objeto que está a um ângulo de 59° e uma distância de 33cm em relação ao sensor, o restante do gráfico está verde, significando que o sensor não está localizando nenhum objeto durante seu mapeamento.



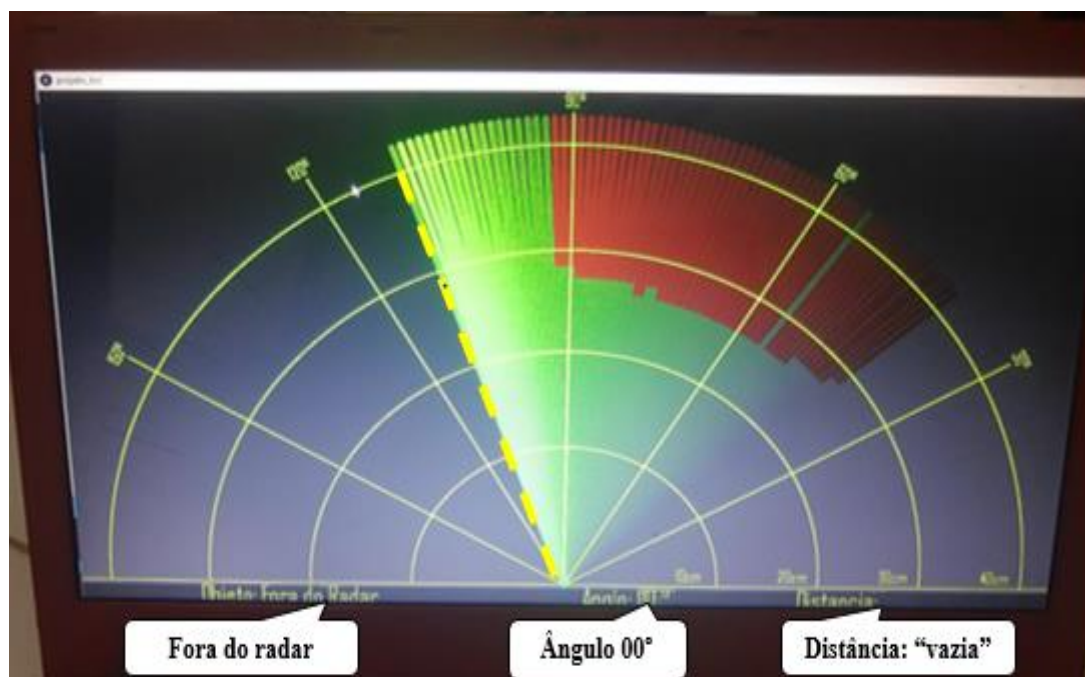
Fonte: Elaborado pelo autor, 2020.

4.4. Interpretação dos testes

A situação ideal para que o dispositivo funcione de acordo como deve ser, é em um ambiente totalmente aberto, de forma que seja evitado reflexões em obstáculos indesejados na medição, tais como paredes, armários ou bancadas.

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.

Figura 14: Representação gráfica sem objeto presente no ambiente durante a varredura.



Fonte: Elaborado pelo autor, 2021.

O Gráfico sempre mostra a distância e o ângulo do objeto em relação ao sensor quando é detectado no mapeamento, mas durante a varredura se o sensor não identificar nenhum objeto, o gráfico apresenta outras informações (figura 14).

4.5. Custo do projeto

O material utilizado no trabalho são materiais de baixo custo, a fim de ser um meio mais barato, mas que atende todas as necessidades de um radar e seu funcionamento. No trabalho poderia utilizar *shield gsm ou bluetooth*, para fazer uma ligação sem fio, mas isso iria aumentar o custo do projeto.

Quadro 1: Custo do projeto.

Descrição	Valor
Arduino Uno + Cabo 2.0 a/b,	R\$ 62,42
Protoboard	R\$ 23,00
Micro servo 9g SG90 TowerPro	R\$ 15,90
Kit 40 Jumpers Macho\Fêmea	R\$ 20,00
Sensor ultrassônico	R\$ 12,50
TOTAL	R\$ 133,82

Fonte: Elaborado pelo autor, 2021.

Segundo Ferreira (2019), a partir dos resultados obtidos, a análise do sistema é registrada, pois pode exibir valores em tempo real e armazenar todos os parâmetros solicitados, que podem ser implementados para fornecer suporte para manutenção preditiva. De acordo com Dilly e Mendes (2015) o Arduino provou ser uma solução acessível e, ao se comunicar com o monitor em tempo real, uma solução flexível pode ser construída e permite que os gerentes acessem facilmente os dados sem depender de software proprietário, e o custo não é alto. No entendimento Marchesan (2012) O custo do projeto pode variar, dependendo do número e tipo de sensores usados, e do uso de fios para interconectar os vários sensores ao Arduino. No entanto essa plataforma de Arduino utilizada no sistema é totalmente *open source* em termos de hardware e software, podendo ser desenvolvidos algoritmos, garantindo assim o baixo custo de aquisição do equipamento.

5. CONCLUSÃO

O presente trabalho possibilitou a compreensão de informações sobre o desenvolvimento de equipamentos de mapeamento incluindo diversas áreas do conhecimento de graduação, como microcontroladores, sistemas embarcados, eletrônica analógica, medição e sensores, processamento digital de sinais. Para este projeto, um sensor ultrassônico foi usado no servo motor. Eles são responsáveis por examinar os obstáculos no ambiente. Esses dois componentes foram conectados ao Arduino, que coordena o movimento do servo, realiza leituras do sensor e envia os resultados para o *Processing*.

Dentre as principais contribuições, é necessário ressaltar que este trabalho pode ser utilizado como modelo para a concretização de outras tecnologias de automação (como residencial, comercial e assistiva) apenas alterando a programação e adaptando de acordo com ambiente. Enfim, de maneira geral, ao se avaliar os resultados obtidos, o dispositivo atendeu os requisitos previstos para funcionamento do Arduino, garantindo que os objetivos predefinidos fossem alcançados.

O primeiro desafio do projeto foi o desenvolvimento do algoritmo de interface, pois este não é um assunto abordado no curso de Tecnologia em Mecatrônica Industrial e requer conhecimento prévio de como elaborar uma interface visual através de linhas de código.

6. REFERÊNCIAS

ÁVILA, Fabrício Schiavo; MARTINS, José Miguel; CEPIK, Marco. Armas estratégicas e poder no sistema internacional: o advento das armas de energia direta e seu impacto potencial sobre a guerra e a distribuição multipolar de capacidades. **Contexto internacional**, v. 31, n. 1, p. 49-83, 2009.

ALSHAMSI, Humaid; KĚPUSKA, Veton; ALSHAMSI, Hazza. Real time vehicle tracking using arduino mega. **International Journal of Science and Technology**, v. 5, n. 12, p. 624, 2016.

ALVARENGA, André Victor; MACHADO, João Carlos; PEREIRA, W. C. Implementação do protocolo para a obtenção de parâmetros do feixe acústico de transdutores ultra-sônicos usados em medicina. **Revista Brasileira de Engenharia Biomédica**, p. 151-163, 2001.

ALVES, Amália; CARASEK, Helena; CASCUDO, Oswaldo. Influência da umidade do revestimento na resistência superficial avaliada por diferentes métodos de ensaio. In: **3ro Congresso Portugues de Argamassas de Construção**, Lisboa, Portugal. 2010.

ARDUINO, **Como criar um Radar com o Arduino?**. 2020. Disponível em: <https://arduinoportugal-pt.tumblr.com/post/172597859213/como-criar-um-radar-com-o-arduino-video>. Acesso em: 26 dez 2020.

ARDUINO, Store Arduino. **Arduino**. Arduino LLC, 2015.

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: 12 set. 2020.

AUTOCORE ROBÓTICA. **Aprenda a utilizar o sensor de distância ultrassônico HC_SR04 com Arduino**. 2020. Disponível em: 11 nov. 2020.

BANZI, Massimo; SHILOH, Michael. **Primeiros passos com o Arduino**. São Paulo: Novatec, p. p1, 2011.

CARVALHO, Gustavo Dambiski Gomes et al. **Radar da inovação como ferramenta para o alcance de vantagem competitiva para micro e pequenas empresas**. RAI-Revista de Administração e Inovação, v. 12, n. 4, p. 162-186, 2015.

CYTRON **TECHNOLOGIES**, HCSR04.
<https://www.datasheetspdf.com/pdf/1291829/Cytron/HC-SR04/1>. Acesso em 11 nov. 2020

DILLY, Rosiene Oliveira; MENDES, Luiz Felipe Carvalho. Aplicação em tempo real de monitoramento de umidade e temperatura utilizando arduino. **Caderno de Estudos em Sistemas de Informação**, v. 2, n. 1, 2015.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em ação**. Novatec Editora, 2013.

FERREIRA, Carlos Andre. **Sistema de monitoramento em motores de indução trifásico utilizando a plataforma arduino**. 2019. Trabalho de Conclusão de Curso (Bacharel em Tecnólogo em manutenção industrial) Universidade Tecnológica Federal do Paraná, 2019.

FERREIRA, Guilherme Torres et al. **Sistema de mapeamento tridimensional de ambientes**. Monografia (Graduação em Engenharia Elétrica) –Universidade de São Paulo, São Paulo, 2014.

GODINHO, Jayson Pereira et al. Influência do formato dos corpos de prova e da idade de cura nas leituras de velocidade de ultrassom, resistividade elétrica e resistência à compressão axial. **3º Simpósio paranaense de patologia das construções**, v. 3, p. 116-129, 2018.

KLANN, Jean Carlos. **Software Simulador Do Microcontrolador M+++**. 2017. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) UNIVERSIDADE REGIONAL DE BLUMENAU, 2017.

KUSRIYANTO, Medilla; PUTRA, Bambang Dwi. Smart home using local area network (LAN) based arduino mega 2560. In: 2016 2nd **International Conference on Wireless and Telematics (ICWT)**. IEEE, 2016. p. 127-131.

LEE, Jen John. **Formação e processamento de imagens de ultrassom**. 2010. Dissertação (Mestrado em ormação e processamento de imagens de ultrassom. 2010. Tese de Doutorado. Universidade de São Paulo). Universidade de São Paulo, 2010.

LORENZI, A.; SILVA FILHO, L. C. P.; CAMPAGNOLO, J. L. Desenvolvimento de redes neurais artificiais para interpretação de ensaios de velocidade de propagação de pulso ultrassônico no concreto. **Revista IBRACON de Estruturas e Materiais**, v. 4, n. 5, p. 829-844, 2011.

MARCHESAN, Marcelo. Sistema de monitoramento residencial utilizando a plataforma arduino. **Santa Maria**, p. 17-19, 2012.

MARTIN, C. C. **Ultrassom**. Editora Associação Brasileira de Ensaios Não Destrutivos e Inspeção, 1. ed., São Paulo, 2012.

MARTINS, Ronan Marcelo et al. Popularização da Ciência e Tecnologia por meio da Robótica. In: **VI WORKSHOP DE ROBÓTICA EDUCACIONAL**. 2015. p. 118.

MCROBERTS, Michael. **Arduino básico**. Novatec Editora, 2018.

MESQUITA, Arduino. **Radar Ultrassônico com Arduino e Processing**. 2020. Disponível em: <https://www.filipeflop.com/blog/radar-ultrassonico-com-arduino-e-processing/>. Acesso em: 26 dez. 2020.

OLIVEIRA, Sérgio. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. Novatec Editora, 2017.

PERETTI, Luís Fernando. **Mapeador De Obstáculos Tridimensionais Por Sensor Ultrassônico**. Trabalho de Conclusão de Curso (Bacharel em Engenharia Eletrônica) Universidade Tecnológica Federal do Paraná - UTFPR, 2019.

PÉREZ, M. Quesada. El radar y la Segunda Guerra Mundial. **Revista Española de Física**, v. 18, n. 2, 2011.

PERRETI, Luís Fernando. **Mapeador De Obstáculos Tridimensionais Por Sensor Ultrassônico**. Trabalho de Conclusão de Curso (Bacharel em Engenharia Eletrônica) Universidade Tecnológica Federal do Paraná, 2019.

RODRIGUES, Estevão Patricio. **Desenvolvimento de um transdutor ultrassônico matricial bidimensional para aplicações submarinas**. Dissertação (Mestrado em Ciências) Universidade de São Paulo, 2020.

SARRO, Wélida de Sousa. **Constantes elásticas de solo compactado a partir do ensaio de ultrassom**. 2017. Dissertação (Mestrado em Ciências Materiais) - Universidade Estadual de Campinas, Faculdade de Tecnologia, Limeira, SP, 2017.

SOUZA, TISSIANA DE ALMEIDA; OLIVEIRA, REGINA CÉLIA. **Avaliação da potencialidade de imagens tridimensionais em meio digital para o mapeamento geomorfológico**. Revista Geonorte, v. 3, n. 5, p. 1348–1355-1348–1355, 2012.

TEIXEIRA, Ana Sofia Barros. **Desenvolvimento de uma interface gráfica para classificadores de imagem**. 2011. Dissertação (Mestrado em Desenvolvimento de Software e Sistemas Interativos), Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco, 2011.

THOMAZINI, Daniel. **Sensores industriais: fundamentos e aplicações**. Saraiva Educação SA, 2005.

TITTON, Mathias Giordani; RIBOLDI, Rodrigo Martini; ROSITO, Fernando Covolan. **Desenvolvimento de um sensor ultrassônico**. In: Mostra de Pesquisa, Ensino e Extensão. 2016.

TÓFOLI, Ricardo José. **Casa inteligente—sistema de automação residencial**. Trabalho de Conclusão de Curso—Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Municipal de Ensino Superior de Assis-IMESA, Assis, SP, 2014.

WOLF, Denis Fernando et al. **Robótica Móvel Inteligente: Da simulação às aplicações no mundo real**. In: Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC. sn, 2009.

APÊNDICE A: Código Processing

```

1  /*  Arduino Radar Project
2  *
3  *  by Dejan Nedelkovski,
4  *  www.HowToMechatronics.com
5  *  Tradução: PontoMakers
6  *
7  */
8
9  import processing.serial.*; // importa biblioteca para comunicação serial
10 import java.awt.event.KeyEvent; // biblioteca de importações para ler os dados da porta serial
11 import java.io.IOException;
12
13 Serial myPort;
14 // variáveis
15 String angle="";
16 String distance="";
17 String data="";
18 String noObject;
19 float pixsDistance;
20 int iAngle, iDistance;
21 int index1=0;
22 int index2=0;
23 PFont orcFont;
24
25 void setup() {
26
27   size (1366, 768); // ***MUDE ISTO PARA SUA RESOLUÇÃO DE TELA***
28   smooth();
29   myPort = new Serial(this,"COM4", 9600); // inicia a comunicação serial
30   myPort.bufferUntil('.'); // lê os dados da porta serial até o caractere '.' Então, na verdade, lê isto: ângulo, distância.
31   orcFont = loadFont("AgencyFB-Reg-30.vlw");
32 }
33
34 void draw() {
35
36   fill(98,245,31);
37   textFont(orcFont);
38   // simulating motion blur and slow fade of the moving line
39   noStroke();
40   fill(0,4);
41   rect(0, 0, width, height-height*0.065);
42
43   fill(98,245,31); // cor verde
44   // chama as funções para desenhar o radar
45   drawRadar();
46   drawLine();
47   drawObject();
48   drawText();
49 }
50
51 void serialEvent (Serial myPort) { // começa a ler dados da porta serial
52 // lê os dados da porta serial até o caractere '.' e coloca na variável String "data".
53 data = myPort.readStringUntil('.');
54 data = data.substring(0,data.length()-1);
55
56 index1 = data.indexOf(","); // encontre o caractere ',' e coloque-o na variável "index1"
57 angle= data.substring(0, index1); // leia os dados da posição "0" para a posição da variável index1 ou seja o valor do ângulo que a placa Arduino enviou para a porta serial

```

```

58     distance= data.substring(index1+1, data.length()); // ler os dados da posição "index1" para o final dos dados pr thats o valor da distância
59
60     // converts the String variables into Integer
61     iAngle = int(angle);
62     iDistance = int(distance);
63 }
64
65 void drawRadar() {
66     pushMatrix();
67     translate(width/2,height-height*0.074); // move as coordenadas iniciais para o novo local
68     noFill();
69     strokeWeight(2);
70     stroke(98,245,31);
71     // desenha as linhas do arco
72     arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);
73     arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI);
74     arc(0,0,(width-width*0.479),(width-width*0.479),PI,TWO_PI);
75     arc(0,0,(width-width*0.687),(width-width*0.687),PI,TWO_PI);
76     // desenha as linhas angulares
77     line(-width/2,0,width/2,0);
78     line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));
79     line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));
80     line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(radians(90)));
81     line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120)));
82     line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(radians(150)));
83     line((-width/2)*cos(radians(30)),0,width/2,0);
84     popMatrix();
85 }
86
87 void drawObject() {
88     pushMatrix();
89     translate(width/2,height-height*0.074); // move as coordenadas iniciais para o novo local
90     strokeWeight(9);
91     stroke(255,10,10); // cor vermelha
92     pixsDistance = iDistance*((height-height*0.1666)*0.025); // cobre a distância do sensor de cm para pixels
93     // limitando o alcance a 40 cms
94     if(iDistance<40){
95         // draws the object according to the angle and the distance
96         line(pixsDistance*cos(radians(iAngle)),-pixsDistance*sin(radians(iAngle)),(width-width*0.505)*cos(radians(iAngle)),-(width-width*0.505)*sin(radians(iAngle)));
97     }
98     popMatrix();
99 }
100
101 void drawLine() {
102     pushMatrix();
103     strokeWeight(9);
104     stroke(30,250,60);
105     translate(width/2,height-height*0.074); // move as coordenadas iniciais para o novo local
106     line(0,0,(height-height*0.12)*cos(radians(iAngle)),-(height-height*0.12)*sin(radians(iAngle))); // desenha a linha de acordo com o ângulo
107     popMatrix();
108 }
109
110 void drawText() { // desenha os textos na tela
111
112     pushMatrix();
113     if(iDistance>40) {
114         noObject = "Fora do Radar";
115     }
116     else {
117         noObject = "Objeto Detectado";
118     }
119 }

```

```

119 fill(0,0,0);
120 noStroke();
121 rect(0, height-height*0.0648, width, height);
122 fill(98,245,31);
123 textSize(25);
124
125 text("10cm",width-width*0.3854,height-height*0.0833);
126 text("20cm",width-width*0.281,height-height*0.0833);
127 text("30cm",width-width*0.177,height-height*0.0833);
128 text("40cm",width-width*0.0729,height-height*0.0833);
129 textSize(40);
130 text("Objeto: " + noObject, width-width*0.875, height-height*0.0277);
131 text("Anglo: " + iAngle + " °", width-width*0.48, height-height*0.0277);
132 text("Distancia: ", width-width*0.26, height-height*0.0277);
133 if(iDistance<40) {
134 text("      " + iDistance + " cm", width-width*0.225, height-height*0.0277);
135 }
136 textSize(25);
137 fill(98,245,60);
138 translate((width-width*0.4994)+width/2*cos(radians(30)),(height-height*0.0907)-width/2*sin(radians(30)));
139 rotate(-radians(-60));
140 text("30°",0,0);
141 resetMatrix();
142 translate((width-width*0.503)+width/2*cos(radians(60)),(height-height*0.0888)-width/2*sin(radians(60)));
143 rotate(-radians(-30));
144 text("60°",0,0);
145 resetMatrix();
146 translate((width-width*0.507)+width/2*cos(radians(90)),(height-height*0.0833)-width/2*sin(radians(90)));
147 rotate(radians(0));
148 text("90°",0,0);
149 resetMatrix();
150 translate((width-width*0.507)+width/2*cos(radians(90)),(height-height*0.0833)-width/2*sin(radians(90)));
151 rotate(radians(0));
152 text("90°",0,0);
153 resetMatrix();
154 translate(width-width*0.513+width/2*cos(radians(120)),(height-height*0.07129)-width/2*sin(radians(120)));
155 rotate(radians(-30));
156 text("120°",0,0);
157 resetMatrix();
158 translate((width-width*0.5104)+width/2*cos(radians(150)),(height-height*0.0574)-width/2*sin(radians(150)));
159 rotate(radians(-60));
160 text("150°",0,0);
161 popMatrix();
162 }

```

APÊNDICE B: Radar Arduino

```

1 // Inclui a biblioteca Servo Motor
2 #include <Servo.h>.
3
4 // Define os pinos Trig e Echo do sensor ultrassônico
5 const int trigPin = 10;
6 const int echoPin = 11;
7 // Variáveis para a duração e a distância
8 long duration;
9 int distance;
10
11 Servo myServo; // Cria um objeto servo para controlar o servo motor
12
13 void setup() {
14   pinMode(trigPin, OUTPUT);
15   pinMode(echoPin, INPUT);
16   Serial.begin(9600);
17   myServo.attach(12); // Define em qual pino o Servo Motor está conectado
18 }
19 void loop() {
20   // gira o Servo Motor de 15 a 165 graus
21   for(int i=15;i<=165;i++){
22     myServo.write(i);
23     delay(30);
24     distance = calculateDistance();// Chama uma função para calcular a distância medida pelo sensor ultrassônico para cada grau
25
26     Serial.print(i); // Envia o grau atual para a porta serial
27     Serial.print(","); // Envia o caractere de adição ao lado do valor anterior necessário posteriormente no IDE de processamento para indexação
28     Serial.print(distance); // Envia o valor da distância para a porta serial
29     Serial.print("."); // Envia o caractere de adição ao lado do valor anterior necessário posteriormente no IDE de processamento para indexação
30   }
31   // Repete as linhas anteriores de 165 a 15 graus
32   for(int i=165;i>15;i--){
33     myServo.write(i);
34     delay(30);
35     distance = calculateDistance();
36     Serial.print(i);
37     Serial.print(",");
38     Serial.print(distance);
39     Serial.print(".");
40   }
41 }
42 // Função para calcular a distância medida pelo sensor ultrassônico
43 int calculateDistance(){
44
45   digitalWrite(trigPin, LOW);
46   delayMicroseconds(2);
47   // Define o trigPin no estado HIGH por 10 micro segundos
48   digitalWrite(trigPin, HIGH);
49   delayMicroseconds(10);
50   digitalWrite(trigPin, LOW);
51   duration = pulseIn(echoPin, HIGH); // Lê o echoPin, retorna o tempo de viagem da onda sonora em microssegundos
52   distance= duration*0.034/2;
53   return distance;
54 }

```
