CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE SANTO ANDRÉ
Tecnologia em Mecatrônica Industrial

Thiago de Freitas Carvalho
Vitor Borges de Melo

ROBÔ AUTÔNOMO PARA MONITORAMENTO DE GASES

Santo André – SP
2022

Thiago de Freitas Carvalho

Vitor Borges de Melo

ROBO AUTÔNOMO PARA MONITORAMENTO DE GASES

Monografia apresentada ao Curso Superior de Tecnologia em Mecatrônica Industrial na FATEC Santo André como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial.

Orientador: Prof. Me. Eliel Wellington Marcelino.

Santo André – SP

2022

FICHA CATALOGRÁFICA

C331r

Carvalho, Thiago de Freitas

Robô autônomo para monitoramento de gases / Thiago de Freitas Carvalho, Vitor Borges de Melo. - Santo André, 2022. – 86f: il.

Trabalho de Conclusão de Curso – FATEC Santo André. Curso de Tecnologia em Mecatrônica Industrial, 2022.

Orientador: Prof. Me. Eliel Wellington Marcelino

1. Mecatrônica. 2. Projeto. 3. Protótipo. 4. Robô autônomo. 5. Tecnologia. 6. Concentração de gases. 7. Monitoramento. 8. Controle. 9. Segurança. I. Melo, Vitor Borges de. II. Robô autônomo para monitoramento de gases.

629.89

Fatec CENTRO PAULA SOUZA GOVERNO DO ESTADO DE SÃO PAULO

Faculdade de Tecnologia de Santo André

LISTA DE PRESENÇA

Santo André, 24 DE JUNHO DE 2022.

LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA: **"ROBÔ PARA MONITORAMENTO DE GASES"** DOS ALUNOS DO 6º SEMESTRE DESTA U.E.

BANCA

PRESIDENTE:
PROF. ELIEL WELLINGTON MARCELINO _____

MEMBROS:
PROF. FERNANDO GARUP DALBO _____

PROF. FRANCISCO JOSÉ DE OLIVEIRA MAIA _____

ALUNO:

THIAGO DE FREITAS CARVALHO _____

VITOR BORGES DE MELO _____

Dedicamos este trabalho a Deus e a todos os nossos familiares que sempre nos apoiaram e a todos que estiveram presentes do desenvolvimento dele.

AGRADECIMENTOS

Primeiramente gostaríamos de agradecer a Deus por nos dar forças e pela oportunidade da realização do trabalho e por nos dar a chance de termos feito parte desta incrível instituição, que é a FATEC Santo André.

Gostaríamos de agradecer a todos os nossos familiares por nos apoiarem, incentivarem e por sempre estarem ao nosso lado em todas as nossas decisões.

Gostaria de agradecer ao meu pai em especial, Juscelino Carvalho por todo o auxílio e suporte durante a construção do protótipo. (Thiago)

Agradecemos ao nosso orientador, Eliel, por nos orientar neste projeto, fornecendo todo o suporte e apoio necessário para a realização dele.

Queremos agradecer em especial ao Professor Fernando Garup Dalbo por todo o apoio, companheirismo e motivação prestados desde o início do curso. E principalmente por todo o conhecimento compartilhado conosco, mostrando que com o conhecimento e a educação voos altos podem ser alcançados.

Gostariamos também de agradecer ao auxiliar docente Mauricio Oliveira pela assistência na produção da estrutura do projeto.

Por Último, agradecemos aos nossos amigos Paulo Gimenez e Mauricio Rosalino, aos professores Edson Kitani, Paulo Tetsuo, Luiz Vasco Puglia e funcionários da FATEC Santo André que fizeram parte dessa etapa de nossas vidas.

“Não devemos ter medo das novas ideias! Elas podem significar a diferença entre o triunfo e o fracasso.”

Napoleon Hill

RESUMO

O projeto “Robô Autônomo Para Monitoramento de Gases” tem como objetivo realizar o monitoramento dos níveis de concentrações gasosas. Os dados são obtidos por sensores específicos da família *MQ* que estão conectados ao *Arduino Mega*, elemento que é responsável também pelo controle de dinâmica do robô. O protótipo apresenta *design* de um robô térreo quatro rodas, tração traseira, orientação por eixo dianteiro e é capaz de realizar rondas de maneira autônoma, quando detectada alguma concentração, independente do sensor ou gás, acima da margem especificada por programação, o protótipo entra em modo de alarme, parando todo o tipo de movimento que estava para ser executado no local onde foi registrada a anomalia e desempenha tanto um sinal visível quanto sonoro. Os alarmes permanecem acionados até que o operador do robô os desligue e destrave o robô, permitindo novamente o seu funcionamento. O projeto foi desenvolvido para proporcionar o aumento da segurança em ambientes industriais, residenciais e em qualquer outro local que haja possíveis riscos de vazamento de gás.

Palavras-Chave: Autônomo. Robô. Segurança. Monitoramento. Gases.

ABSTRACT

The Project “Autonomous Robot to Gas Monitoring” has like goal accomplish the monitoring of the levels of gas concentration. The data are obtained by MQ family sensors, which are connected to the Arduino Mega™, element that is also responsible for controlling the dynamics of the robot. The prototype features design of a for-wheeled ground robot, rear traction, orientation by front axle and it is able to perform rounds autonomously. When detected some concentration, independent of sensor or gas, above the margin specified by programming, the prototype goes into alarm mode, stopping all sorts of movement that was to be executed at the site where the anomaly was recorded. The alarms remain activated until the robot operator turns them off and unlocks the robot, allowing it to work again. The project was developed to provide increased safety in industrial, residential environments and some other locations that there are possible risks of gas leakage.

Keywords: Autonomous. Robot. Safety. Monitoring. Gas.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

µs	Microssegundos
A	Ampére
ºC	Grau celsius
CAD	*Computer aided drawing*
CC	Corrente contínua
cm	Centímetro
CO2	Dióxido de carbono
GLP	Gás liquefeito de petróleo
Hz	Hertz
I/O	*Input/Output*
IDE	*Integrated development environment*
kHz	Kilohertz
Li-ion	Litio-ion
mA	Miliampére
mAh	Miliampére hora
ms	Milissegundos
NTC	*Negative temperature coefficient*
ppm	Partes por milhão
Rx	Canal de recepção
Tx	Canal de transmissão
V	Volt

SUMÁRIO

1. INTRODUÇÃO

Em ambientes onde detecta-se uma alta concentração de gases, é extremamente importante realizar o monitoramento das concentrações gasosas presentes, se forem tóxicos, podem causar danos a todo o ambiente fabril ou ecossistema local.

As principais consequências de um possível vazamento de gás vão desde incêndios a explosões em residências, fábricas, comércios, mas também, gerar ambientes nocivos a vida presente, que podem desempenhar por exemplo asfixia, febre, queimaduras, irritações, deterioração, perda de consciência ou mesmo levando a morte.

Dentre esses locais onde há um alto volume de gases ou vapores, se encontram as indústrias químicas, de bebidas, alimentos, hospitais, aterros sanitários, gasodutos ou em qualquer outro local onde exista um alto risco biológico caso haja um vazamento.

De modo a promover uma interação entre homem e máquina em ambientes diversos, o desenvolvimento de um robô de monitoramento para concentrações gasosas tende a agregar segurança ao ambiente que se está usufruindo, ajudando por fim, a diminuir os riscos que os gases trazem.

A principal vantagem do robô frente a um sensor estacionário é que, devido ao seu funcionamento de forma dinâmica, ele possui liberdade para transitar e fazer suas rondas, sendo assim, ele não fica fixo em apenas um ponto pré-estabelecido.

1.1. Motivação

Dentre o quadro de respostas para este tópico, destacam-se a própria construção deste dispositivo, pois ainda não existe similar nesta aplicação. O que beneficia muito, ambientes de risco por exemplo, é claro que adaptações e melhorias devem ser feitas para que o protótipo se adeque aos ambientes com segurança e eficiência, todavia, estarmos criando um dispositivo que realmente possa ajudar as pessoas contra perigos assim, é algo que nos motiva.

Almejamos ser grandes profissionais no futuro, que possam ter seu lugar no mercado de trabalho e este trabalho é a prova que mesmo de pouco em pouco, estamos mais perto de conquistar nosso objetivo.

1.2. Objetivos gerais

Este trabalho tem como função construir um dispositivo autônomo para colaborar com o ambiente de trabalho, indicando onde há um possível vazamento de gás, proporcionando maior segurança para todos aqueles que tem contato direto com áreas de risco.

1.3. Objetivos específicos

Durante o planejamento do protótipo “Robô Autônomo Para Monitoramento de Gases”, foram estabelecidos objetivos específicos que seriam o núcleo da ideia do projeto, são eles a atuação de modo colaborativo com o trabalhador e o beneficio de segurança ao ambiente de trabalho. A partir destes, conceitos como autônomo e monitoramento de concentrações foram estabelecidos.

1.4. Organização do trabalho

O trabalho está organizado da seguinte maneira, no capítulo 2, foram inseridas todos as ideias que foram necessárias como revisão bibliográfica para a realização do projeto.

No capítulo 3, foi descrita a metodologia que foi utilizada para construção do projeto.

No capítulo 4, estão descritas todas as etapas realizadas durante o desenvolvimento do projeto.

No capítulo 5, estão os resultados desejados e obtidos do trabalho.

No capítulo 6, estão as considerações finais do trabalho, juntamente com propostas futuras para aprimoramento do projeto.

No capítulo 7, encontram-se todas as referências que foram usadas para o desenvolvimento do trabalho. Após a bibliografia estão apêndices do projeto.

2. REVISÃO BIBLIOGRÁFICA

Este capítulo contém alguns elementos teóricos necessários para a construção do trabalho. Para a elaboração deste item foram utilizadas referências, artigos, livros, trabalhos acadêmicos, sites, e outros materiais que contenham conteúdos associados com o projeto.

2.1. Gases

O gás é um dos estados de matéria fluída, sendo este mais fluído do que o líquido. Em seu arranjo, átomos, moléculas ou íons que o compõem possuem um empacotamento quase inexistente, ou seja, estes são quase que totalmente livres uns dos outros, praticamente não há interação entre eles e é por esta razão que são considerados fluídos. Quanto maior for o fator de empacotamento da estrutura, mais rígida a matéria é e menos esta sofre alterações por meio de forças externas (ATKINS; JONES,2006).

Para este trabalho, foram levados em consideração somente alguns dos gases que são utilizados em ambiente fabril e possíveis de detecção pelos sensores, como GLP (Gás Liquefeito de Petróleo), propano, hidrogênio, metano, amônia, benzeno, dióxido de carbono, óxido nítrico e alguns álcoois, como etanol.

2.1.1. Principais gases industriais existentes

Abaixo, estão os principais gases industriais existentes, o método para a escolha deles se deu devido aos sensores que foram usados no projeto. Os sensores e suas aplicações são descritos no tópico seguinte.

- GLP: é um composto retirado através do refino do petróleo. É composto de uma mistura de gases hidrocarbonetos, como o propano, propeno, e o buteno. É um item combustível e gasoso a pressão e temperatura normal, inflamável e inodoro. O GLP em estado gasoso é mais denso do que o ar atmosférico, portanto, em caso de vazamento este ocupa sempre áreas mais baixas (CONSIGAZ, s.d.);
- Propano: é um hidrocarboneto incolor, não tóxico, com odor característico e altamente inflamável. Apresenta propriedades não tóxicas, entretanto tem efeito levemente anestésico e gera uma irritação considerável às membranas mucosas, em complemento a Norma Regulamentadora 15, declara o propano como um asfixiante simples. Em caso de superexposição ao item, este pode ocasionar em asfixia e neste caso, apresenta sintomas como, náuseas, pressão na testa e nos olhos, podendo ainda causar perda de consciência e morte. Em caso de vazamento o propano tende a acumular-se em lugares baixos (GAMA GASES, s.d.);
- Metano: é um gás incolor, inflamável, não tóxico e com odor característico. É considerado como asfixiante simples e em caso de superexposição pode causar náuseas, pressão na testa e nos olhos, podendo ainda causar perda de consciência e morte (GAMA GASES, s.d.);
- Amônia: gás inflamável, tóxico a pressão e temperatura ambiente, corrosivo em ambiente úmido, incolor e de odor muito incômodo. Na presença de umidade, o gás quando em contato com a pele, dependendo da intensidade, poderá causar irritação ou queimaduras. caso inalado normalmente, atua principalmente no sistema respiratório. Caso haja uma superexposição, a pessoa terá um ataque de tosse violento em resultado da ação do gás do sistema respiratório e se o

indivíduo não conseguir sair do local rapidamente, este sofrerá forte irritação nos pulmões, edema pulmonar, queimadura nos olhos e cegueira temporária, em caso de ingestão, corrosão violenta da boca, garganta e estômago ou morte (GAMA GASES, s.d.);

- Benzeno: é um gás altamente tóxico e cancerígeno, incolor, inflamável e de um aroma doce. Em caso de inalação em altas concentrações por curto período, o benzeno pode desencadear sintomas como sonolência, enjoo, aceleração do ritmo cardíaco, cefaleia, tremor, confusão mental e inconsciência. Além do fator cancerígeno que este pode gerar, como a leucemia mieloide, que está ligada à má formação de células vermelhas dentro da medula óssea (ECYCLE, s.d.);
- Dióxido de carbono: é um gás não inflamável a temperatura e pressão ambiente, incolor, inodoro de sabor levemente ácido. É considerado um asfixiante simples, embora apresente alguns efeitos nocivos quando em concentrações elevadas como, forte asfixia e morte, estes sintomas são precedidos de dor de cabeça, tontura, aceleração da respiração e batimento cardíaco, fraqueza muscular e zumbido nos ouvidos (GAMA GASES, s.d.).

2.2. Principais sensores de gás existentes no mercado

O Quadro 1 mostra, quais são os principais modelos de sensores de gases existentes no mercado, exibindo também quais são as suas aplicações.

Quadro 1: Comparação entre os principais sensores de gás do mercado.

Modelo	Aplicação
MQ-2	Gases Inflamáveis e Fumaça
MQ-3	Álcool
MQ-4	Metano
MQ-5	GLP e Gás Natural
MQ-6	GLP, Isobutano e Propano
MQ-7	Monóxido de Carbono
MQ-8	Hidrogênio
MQ-9	Monóxido de Carbono
MQ-135	Gases Tóxicos

Fonte: Autores, 2022.

Para a realização do projeto, foram selecionados os modelos *MQ-2*, *MQ-3* e *MQ-135*.

Nos subtópicos seguintes, haverá um maior detalhamento para cada sensor em específico, com o objetivo de destrinchar cada modelo, tendo como meta, um melhor entendimento de sua função no projeto. A unidade usada é a ppm(partes por milhão).

2.3. Sensor de gás MQ-2

O sensor de gás MQ-2, como mostrado na Figura 1, possui alta sensibilidade para a detecção de gases infamáveis, dos quais se destacam, GLP, Propano e Hidrogênio, podendo ser utilizado também para outros tipos de gases, embora com uma taxa de detecção reduzida. Este sensor é de baixo custo, sendo capaz de realizar detecções de fumaça no ambiente e podendo ser usado em diversas aplicações. A sua sensibilidade é ajustada por meio de um potenciômetro (HANWEI ELECTRONICS, s.d.).

Figura 1: Sensor de Gás MQ-2.

Fonte: https://www.usinainfo.com.br/1016983-thickbox_default/detector-de-gas-sensor-de-gas-mq-2-gas-inflamavel-e-fumaca.jpg

2.4. Sensor de gás MQ-3

O sensor de gás *MQ-3*, como mostra a Figura 2, possui como característica principal, alta sensibilidade para vapores de álcool e etanol. É um sensor de baixo custo, que pode ser usado em diversos projetos. O ajuste da sensibilidade é feito por meio de um *trimpot* (HANWEI ELECTRONICS, s.d.).

Figura 2: Sensor de Gás MQ-3.

Fonte:https://d229kd5ey79jzj.cloudfront.net/340/images/340_1_H.png

2.5. Sensor de gás MQ-135

O sensor de gás *MQ-135*, como mostra a Figura 3 tem como principal função, a alta sensibilidade para gases tóxicos, dos quais destacam-se, amônia, benzeno,

CO2**(**Dióxido de carbono**)**, óxido nítrico etc. O modelo também é capaz de realizar detecções de álcool e fumaça, embora com capacidade de detecção reduzida. Possui sua sensibilidade ajustável por meio de potenciômetro (HANWEI ELECTRONICS, s.d.).

Figura 3: Sensor de Gás MQ-135.

Fonte: https://cdn.multilogica-shop.com/Sensor_de_Qualidade_do_Ar_MQ_135_M.jpg.

2.6. Características técnicas dos sensores de gás

No Quadro 2, estão contidas todas as características técnicas dos sensores de gás *MQ-2*, *MQ-3* e *MQ-135*, respectivamente, que serão usados no projeto.

Quadro 2: Características técnicas dos sensores MQ-2, MQ-3 e MQ-135.

Faixa de detecção(ppm)	Tensão de operação(V)	Pinagem
300 -10.000	3 - 5	VCC, GND, A0 e D0
10 - 10.000	3 - 5	VCC, GND, AOUT e DOUT
10 – 300 (amônia e álcool) 10 – 1000 (benzeno)	1,5 (detecção de CO) 5 (para outros gases)	VCC, GND, A0 e D0

Fonte: Autores, 2022.

O princípio de funcionamento do sensor de gás é baseado na queda de tensão em cima de uma resistência interna e variará seu valor para cada tipo e concentração

de gás presente no sistema. O sinal obtido pelo módulo pode ser tanto do tipo analógico quanto digital.

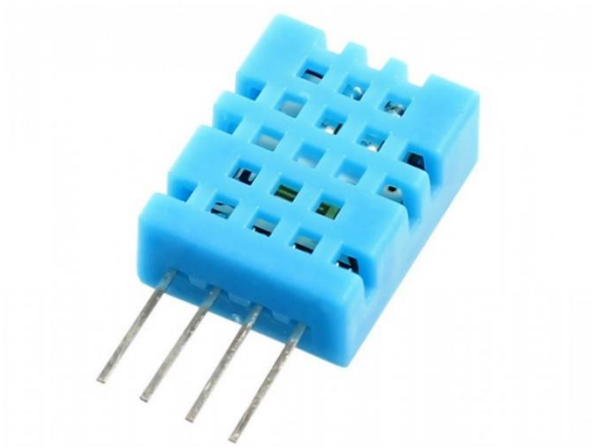
2.7. Sensor de temperatura e umidade DHT11

Além do monitoramento de gases, o projeto conta também com obtenção dos valores de temperatura. Para a aquisição deles. foi escolhido o sensor *DHT11*, como mostra a Figura 4, possuindo em sua estrutura física um *termistor* do tipo *NTC* (*Negative Temperature Coefficient)*, que é mais sensível para variações de temperatura.

No entanto, o *NTC* se comporta de forma não linear, sendo necessário a utilização de um circuito para correção da forma. Por se tratar de um modulo, não há a necessidade da construção de um circuito externo, sendo que este é integrante do dispositivo. Como característica, o sensor realiza medições de temperatura na faixa de 0 a 50 ºC e umidade no intervalo de 20 a 80% (DE SOUZA, s.d.).

Como já tínhamos o *DHT11* em mãos, optamos pela sua utilização, porém para o seu uso, é necessário o uso de uma biblioteca externa. Importante ressaltar também, que este modelo possui um tempo de resposta mais lento, sendo assim, pode-se usar no seu lugar o sensor *LM35*, que dispensa o uso de biblioteca e possui resposta rápida.

Figura 4: Sensor *DHT11*



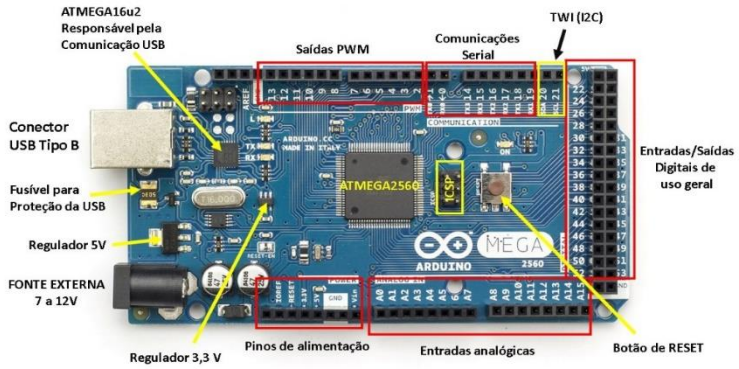
Fonte:https://www.usinainfo.com.br/1020721-thickbox_default/sensor-de-umidade-e-temperatura-dht11.jpg

2.8. Arduino™ Mega

Para o controle da parte robótica e dos sensores do projeto foi escolhido o Arduino™ Mega como mostrado na Figura 5, que possui o microcontrolador *ATMEGA2560*. Como seu principal diferencial, a placa possui 54 pinos digitais que podem ser usados como entradas e saídas e 16 pinos analógicos, permitindo que diversos dispositivos de *I/O* (*Input/Output*) possam ser ligados na placa (SOUZA, 2014).

Como possui uma alta quantidade de pinos em relação ao Arduino™ Uno, a placa é altamente recomendada para projetos onde vários dispositivos serão ligados em um único local para processamento, bem como é o caso deste trabalho.

Figura 5: Representação esquemática do Arduino™ Mega.



Fonte:https://www.embarcados.com.br/wp-content/uploads/2014/04/Arduino-Mega-2560-recursos.jpg

2.9. Sensor ultrassônico HC-SR04

O sensor ultrassônico HC-SR04, como mostra a figura 6, é um componente que faz uso de ondas de som, do espectro localizado como ultrassom. Este intervalo de onda está inserido de 20 KHz (Kilohertz) para cima e é caracterizado por ser um espectro de onda inaudível aos seres humanos que têm sua audição compreendida na faixa de 20 Hz (Hertz) até 20 KHz, também conhecido como espectro audível de onda.

Em seu funcionamento, o pino *trigger* do módulo recebe uma alimentação de 5 V por um período de no mínimo 10 µs (Microssegundos), então o módulo enviará para o ambiente através do pino Tx (Canal de transmissão), 8 pulsos com frequência de 40KHz e o pino Rx (Canal de recepção) aguardará seus retornos. Quando a onda rebatida é captada pelo receptor, o canal *echo* do módulo atinge o nível lógico 1 e é

mantido por um tempo proporcional a distância, em que a onda lançada demora para ser captada.

O dispositivo não informa de maneira direta o valor da distância notada, por isso é necessário implementar um algoritmo no *software*, que use os dados obtidos (tempo) em um cálculo, para que então se tenha o valor da distância.

Figura 6: Sensor Ultrassônico HC-SR04

Fonte: http://microcontrollerelectronics.com/wp-content/uploads/2014/10/HCSR04.jpg

2.10. Motor de corrente contínua

Segundo Chapman (2013), um motor elétrico é um dispositivo que faz a conversão de energia elétrica para energia mecânica e que também estão em diversas aplicações do cotidiano, que vão desde eletrodomésticos para equipamentos industriais.

Para a movimentação do robô, utilizamos dois kits de eixos com motores CC (Corrente continua) 6V, que foram tirados de um carro de controle remoto, um eixo traseiro com caixa de redução acoplada para proporcionar o deslocamento do robô e um eixo dianteiro com um sistema mecânico com caixa de redução e controle de eixo, com retorno de centro de rodas por mola, para efetuar mudanças de orientações durante o movimento do robô.

A figura 7 mostra o kit eixo traseiro com o motor CC visto de cima.

Figura 7: Kit eixo traseiro com motor CC 6V visto de cima

Fonte: Autores, 2022.

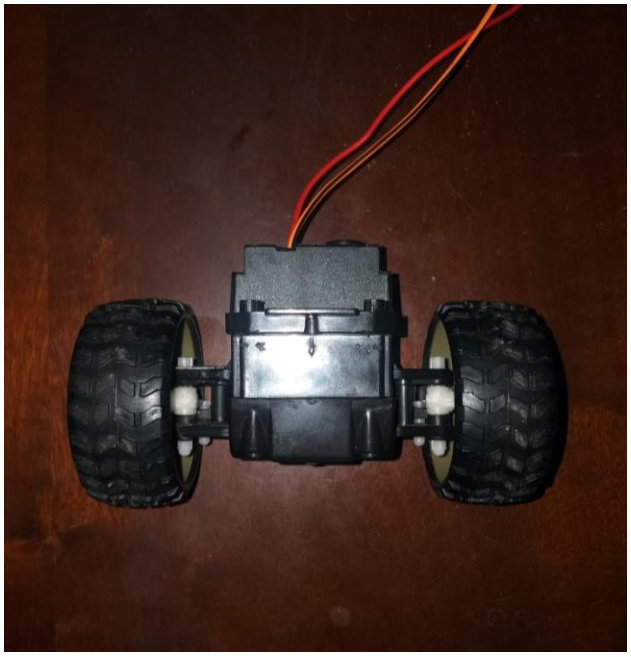
Já a figura 8 mostra o kit eixo traseiro visto na diagonal, onde é possível ver mais detalhes do conjunto mecânico.

Figura 8: Kit eixo traseiro com motor CC 6V visto na diagonal

Fonte: Autores, 2022.

A figura 9 mostra o kit eixo dianteiro visto de frente.

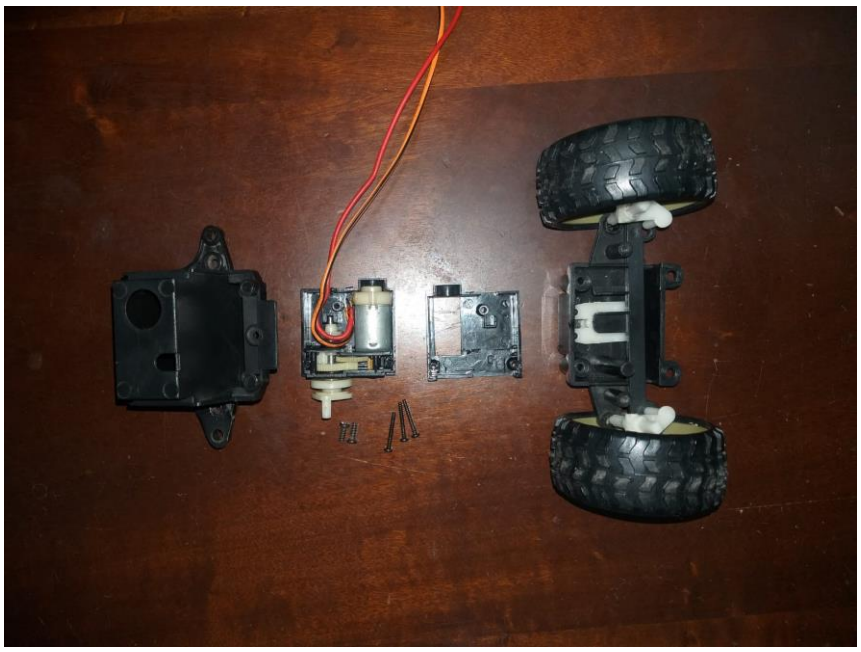
Figura 9: Kit eixo dianteiro com motor CC 6V visto de frente



Fonte: Autores, 2022.

Já a figura 10 mostra o kit eixo dianteiro desmontado, dessa maneira pode-se ver todos os elementos presentes na estrutura.

Figura 10: Kit eixo dianteiro com motor CC 6V desmontado



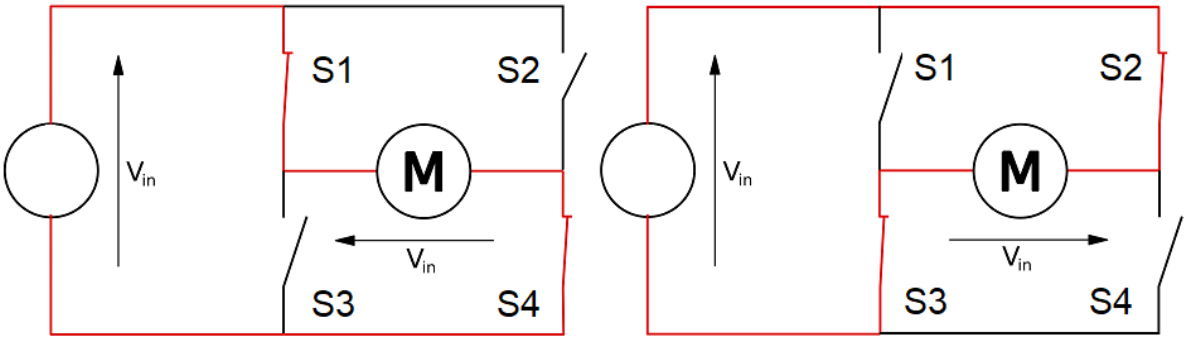
Fonte: Autores, 2022.

2.11. Driver ponte H L298N

O circuito Ponte H, conforme mostrado na figura 11, é amplamente usado em aplicações onde é necessário que haja a inversão da rotação de um motor eletrico, como o caso deste projeto.

Seu funcionamento consiste em um motor que está ligado em quatro chaves, com estas sendo responsaveis pelo sentido do motor. Para que o motor gire em um sentido, as chaves S1 e S4 são fechadas, caso o outro sentido seja desejado, fecham-se as chaves S2 e S3.

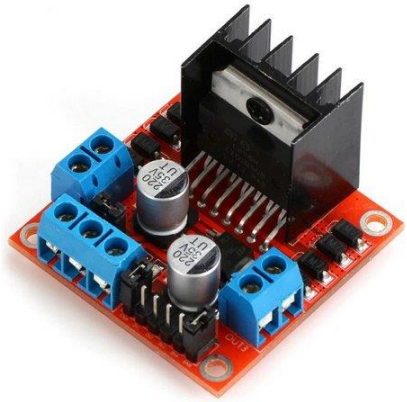
Figura 11: Circuito Ponte H



Fonte: https://wiki.ifsc.edu.br/mediawiki/images/9/99/MIC2PonteH.png

Já o *driver L298N*, como mostra a figura 12. permite que toda lógica de inversão seja feita eletronicamente, por meio do circuito integrado *L298*.

Figura 12: Driver L298N



Fonte:https://cdn.awsli.com.br/600x450/704/704119/produto/28912218/241eaeb4c7.jpg

A configuração dos terminais do driver está organizada da seguinte forma:

- VCC: sua principal função é alimentação dos motores ligados no driver;

- GND: sua função é atuar como o referencial terra, como o projeto utiliza o *Arduino Mega*, é importante que o terra seja o mesmo da placa;
- 4,5V – 7V: sua aplicação é a alimentação lógica para a ponte H que será ligada no *Arduino*, dessa forma é necessário que a tensão fique entre os valores de 4,5 – 7V;
- Regulador 4,5V – 7V: é um *jumper* que aciona o regulador de tensão do terminal acima. Se o jumper estiver acionado, o acionamento do regulador é feito e assim, a tensão lógica da placa passa a ser 5V em relação ao valor do VCC;
- ENA: Liga o motor A ao *Arduino*;
- ENB: Liga o motor B ao *Arduino*;
- IN1, IN2, IN3 e IN4: são entradas de controle do *driver*, sendo que IN1 e IN2 controlam o motor A e IN3 e IN4 controlam o motor B.

2.12. Chassis

Foram criados para este projeto três modelos de chassis para estruturar e proteger os componentes, como os eixos motores e placas eletrônicas e uma estrutura de haste para estruturar os sensores de gás no topo do robô, fora dois para-choques construídos através da carcaça do carro de controle de remoto.

O material utilizado para a construção dos chassis foi o acrílico e as dimensões, bem como o desenhos em CAD(*Computer Aided Drawing*), podem ser vistas no Apêndices A, B, C e D.

Conforme mostra a figura 13, pode-se ver o chassis inferior.

Figura 13: Chassis inferior

Fonte: Autores, 2022

Já a figura 14 mostra o chassis confeccionado para suporte do servo motor na estrutura do robô.

Figura 14: Chassis para fixação do servo motor Futaba

Fonte: Autores, 2022.

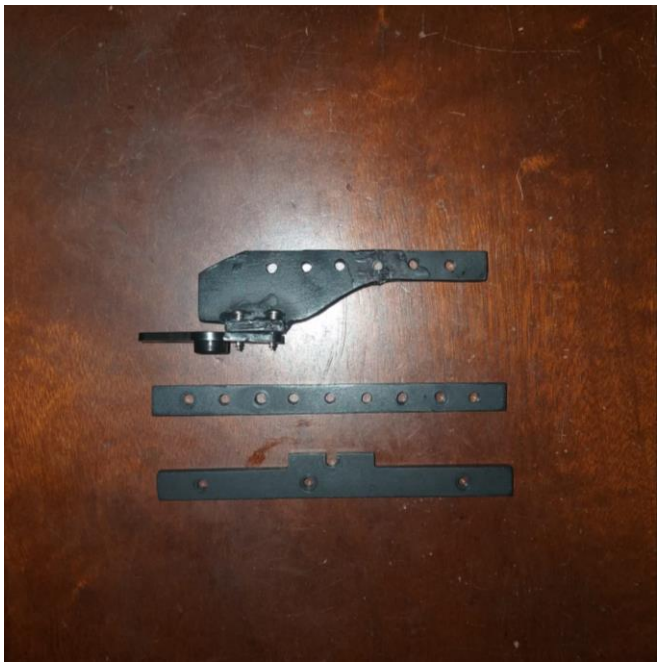
A figura 15 mostra o chassis superior do robô.

Figura 15: Chassis superior

Fonte: Autores, 2022.

Já a figura 16 mostra as hastes para suporte e fixação dos sensores MQ no servo motor *Futaba*.

Figura 16 : Hastes de Fixação



Fonte: Autores, 2022.

A figura 17 mostra os para-choques dianteiro e traseiro instalados no robô.

Figura 17: Para-choques

Fonte: Autores, 2022

2.13. Servo Motor

O servo motor, é um motor elétrico que é acoplado a um sensor de posição, chamado *encoder* e dessa forma é possível fazer o controle de velocidade, aceleração e sua posição angular.

A sua função no projeto é de realizar a movimentação do sensor ultrassónico quando ele determina se há a presença de algum objeto e qual o melhor caminho a ser seguido.

Foram utilizados 2 modelos, o *SG90* da marca *TowerPro*, como mostra a figura 18 e o *S300*3 da fabricante *Futaba* conforme mostra a figura 19. O primeiro modelo teve como função realizar o posicionamento e mover o sensor *HC-SR04*, e o segundo, foi para a movimentação dos sensores.

Figura 18: Micro servo TowerPro SG90

Fonte:https://www.electronicaembajadores.com/datos/fotos/articulos/grandes/mm/mmsv/mmsv002.jpg

Figura 19: Servo motor Futaba S3003

Fonte:https://http2.mlstatic.com/D_NQ_NP_746570-MLB44282252023_122020-O.jpg

2.14. Buzzer

O *Buzzer*, conforme mostra a figura 20, é um componente cuja função é efetuar um disparo sonoro quando recebe um nível alto em seu pino de alimentação. O componente pode ser de 2 configurações diferentes, funcionando de forma ativa, que é quando não se pode variar a sua frequência, ou também pode funcionar de maneira passiva, onde é permitida a mudança da frequência, permitindo a criação de sons diversos.

Para o projeto, a sua função é de alertar quando algum dos sensores detectar um valor acima dos limites determinados previamente, mantendo-se ligado até que o valor de leitura seja menor que o limite.

Figura 20: Buzzer

Fonte:https://cdn.awsli.com.br/800x800/78/78150/produto/3540429/703ecfd46b.jpg

2.15. LED

O LED (*Lighting Emitting Diode*), como mostra a figura 21, é um componente eletrônico que é bastante usado para sinalização.

Figura 21: Led

Fonte: https://cf.shopee.com.br/file/1de4ce016d01ce6594b5002f70818696

2.16. Fontes de alimentação

Como os elementos elétricos presentes no projeto requerem diferentes valores de tensão, optamos por utilizar diferentes tipos de baterias para alimentar esses grupos, poupando a construção de um único circuito regulador de tensão para ambos os elementos.

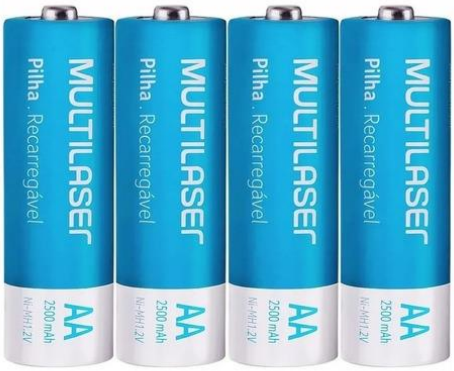
Sendo assim, para o grupo dos motores foram utilizadas as baterias do tipo *Li-íon* (Lítio-Ion), conforme mostra a figura 22. Para o bom funcionamento do protótipo, foi-se necessário utilizar 2 baterias *Li-íon* com tensão de 3.7 V e capacidade de 8000 mAh (Miliampere-hora), que proporcionaram uma autonomia de cerca de uma hora e meia.

Figura 22: Bateria Li-íon

Fonte:https://http2.mlstatic.com/D_NQ_NP_820823-MLB48938085929_012022-O.jpg

Já para a alimentação dos sensores HC e MQ foram utilizadas 4 pilhas AA de NimH recarregáveis com tensão de 1.2 V com capacidade de 2500 mAh, que podem ser vistas na figura 23. Os outros dispositivos, como o sensor DHT11, os LEDs e o *Buzzer* foram alimentados por meio das portas de energia do Arduino, que foi ligado em uma bateria portátil, *powerbank*, que pode ser vista na figura 24.

Figura 23: Pilhas AA NimH recarregáveis



Fonte:https://http2.mlstatic.com/D_NQ_NP_842105-MLA46275796495_062021-O.jpg

Figura 24: Powerbank

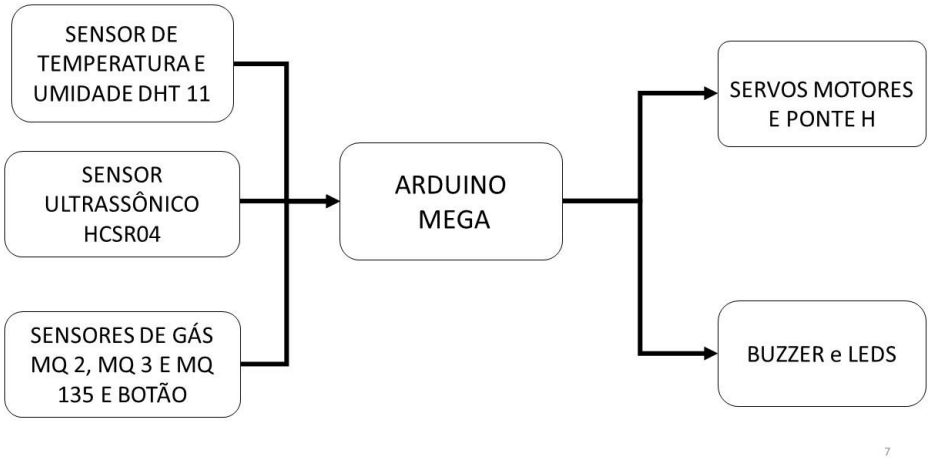
Fonte: https://imgs.extra.com.br/1516609638/1xg.jpg?imwidth=292

3. METODOLOGIA

Para o bom funcionamento do protótipo, estabelecemos o Arduino Mega™ com centro de processamento do sistema, nele são carregados todos os dados provindos dos sensores ultrassônicos, MQs, DHT11 além de variáveis de memória para serem tratados. Após o tratamento por meio da lógica de programação construída, os atuadores como ponte H, *buzzer* e LEDs são acionados devidamente.

A disposição das ligações físicas do robô foi feita por meio de um diagrama de blocos, como mostra a Figura 25.

Figura 25: Disposição das ligações físicas do robô



Fonte: Autores, 2022.

4. DESENVOLVIMENTO

Neste capítulo, estão descritas todas as etapas para o desenvolvimento do projeto.

4.1. Testes com os sensores MQ

Para o início dos testes, foi necessário realizar testes individuais em cada sensor utilizado no projeto, dessa forma, pudemos analisar os comportamentos deles.

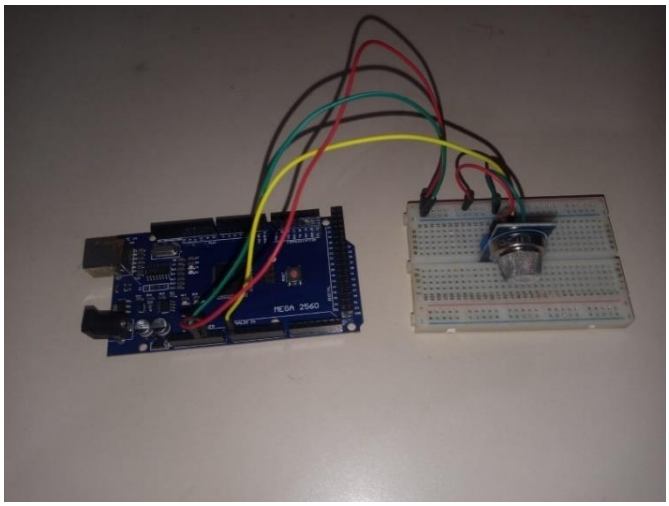
Os testes consistiram em realizar diversas leituras nos sensores por meio de um código que foi escrito na IDE (*Integrated Development Environment*) do *Arduino.* Inicialmente, o objetivo dos testes foram somente para verificar a variação dos valores analógicos do sensor, as conversões para as respectivas unidades de medida dos dispositivos serão feitas posteriormente neste capítulo.

A figura 26, mostra a ligação do sensor MQ2 ao *Arduino Mega*™, e é possível notar que o pino D0 não está conectado na placa, pois a sua função é somente informar se há ou não uma concentração de gás.

Importante ressaltar que antes de ser testado, o sensor foi aquecido por cerca de 24 horas, conforme mencionado em seu *datasheet*. Para o pré-aquecimento, os sensores foram ligados diretamente no pinos de alimentação do *Arduino*.

A saturação do sensor também foi algo importante notado, pois caso isso ocorra, o sensor perde a sua sensibilidade, se descalibra e corta a detecção, dessa forma é necessário fazer uma nova calibração com gases inertes.

Figura 26: Ligação do sensor MQ2 com o Arduino Mega



Fonte: Autores, 2022.

Após as ligações serem realizadas, foi possível testar o sensor por meio do código escrito na Arduino IDE, como mostra a figura 27.

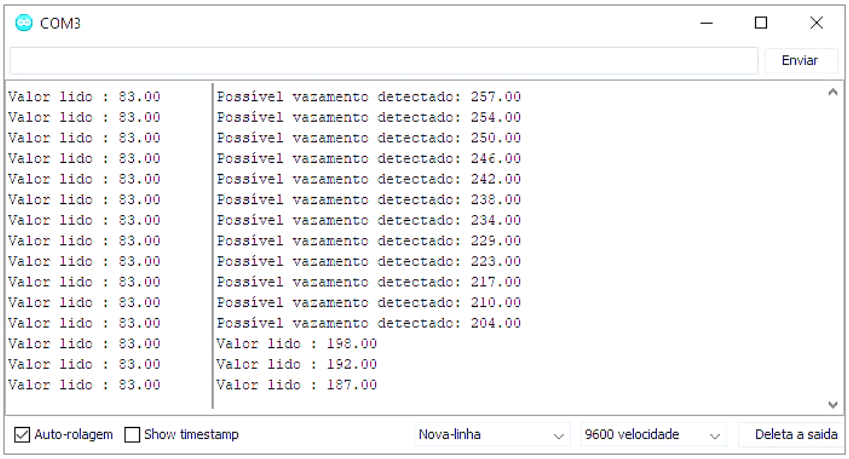
Figura 27: Código teste do sensor MQ2

```
#define MQ2 A0
float leitura;

void setup() {
  Serial.begin(9600);
  pinMode(MQ3, INPUT);
}

void loop()
{
  leitura = analogRead(MQ3);

  if (leitura > 440)
  {
    Serial.print("Possível vazamento detectado: ");
    Serial.println(leitura);
  } else {
    Serial.print("Valor lido : ");
    Serial.println(leitura);
  }
  delay (200);
}
```

Fonte: Autores, 2022.

Para verificarmos a variação, foi utilizado um papel com álcool e ele foi aproximado ao sensor, dessa forma permitindo o início dos testes. Com os testes realizados, foi possível analisar por meio do monitor *serial*, como mostrado na figura 28, as variações nos valores que o sensor estava medindo por meio da entrada analógica A0 do *Arduino Mega*.

Figura 28: Visualização dos valores por meio do monitor serial



Fonte: Autores, 2022.

Nota-se que quando o valor de teste ultrapassa 200, o monitor imprime que um possível vazamento é detectado, dessa forma fazendo com que a tensão do sensor aumente e provoque o aquecimento de sua resistência interna, proporcionando que a medição seja realizada. Como mencionado anteriormente, o intuito desses testes foi apenas verificar a sensibilidade e a variação do sensor e os valores definidos, foram definidos de forma empírica os valores para testes dos sensores.

Como a pinagem e o código é igual para os outros modelos, só mencionamos o MQ2, devido ao fato de que o teste para os outros sensores é igual, porém eles também tiveram êxito nas medições realizadas.

4.2. Mecanismo para acoplamento do HC-SR04

O principal objetivo na adição de um mecanismo para movimentar o sensor ultrassónico, foi ocasionar a redução da quantidade de sensores a serem utilizados, otimizando o processo de detecção e criando uma tecnologia semelhante a um radar.

Sendo assim, como mostra a figura 29, foi construído um suporte para que o *HC-SR04* fosse acoplado ao servo motor e dessa forma permitiu-se a movimentação

do sensor de acordo com o ângulo do servo motor. O desenho técnico do suporte se encontra no apêndice E.

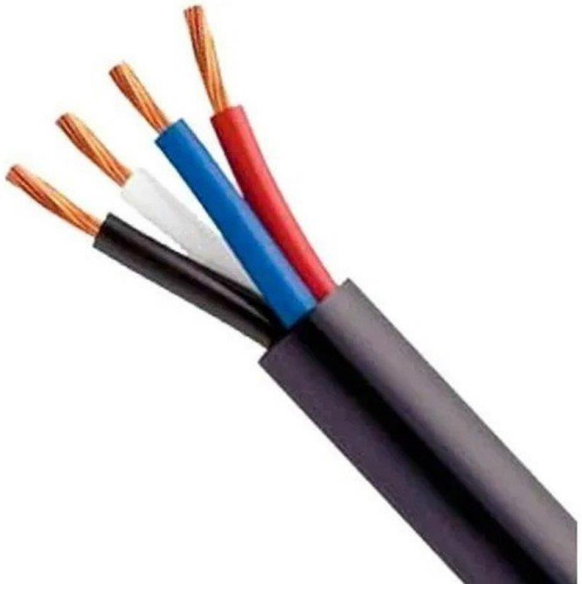
Figura 29: HC-SR04 encaixado no suporte e acoplado no servo motor

Fonte: Autores, 2022.

Importante ressaltar que durante os testes iniciais, foi utilizada uma lógica de captação de distância por grau frequente, entretanto essa ideia provou-se ineficaz devido ao problema do *HC-SR04* em realizar medições dinâmicas, o que ocasionou em grandes erros de medidas e consequentemente gerou uma má movimentação no robô. Devido a isso, uma nova lógica foi implementada para a diminuição do erro de varredura, que diferente da anterior, varre as distâncias frequentemente no ângulo de 90 graus para a identificação de obstáculos, de acordo com a lógica, quando o objeto é detectado, o robô para brevemente e em seguida, o servo motor varia o seu ângulo para realizar medições referentes ao ângulo de 60 e 120 graus para o funcionamento da lógica de curvas e 5 e 175 graus para evitar travamentos ou raspagens em paredes localizadas as laterais do robô.

Outro ponto importante para o pleno funcionamento do radar, foi a utilização do cabo PP, conforme mostra a figura 30, pois devido aos movimentos cíclicos realizados pelo conjunto servo e *HC-SR04*, fios rígidos, mesmo com relativa espessura, são rompidos facilmente.

Figura 30: Cabo PP



Fonte: https://8639028l.ha.azioncdn.net/img/2020/10/produto/221/2551-cabo-pp-preto-4x2-5-mm.jpg?ims=800x800

4.3. Lógica de programação do robô

A estrutura lógica do robô foi modelada de modo com que ele agisse através dos seguintes princípios:

O robô se movimentou de modo autônomo e realizou suas rondas pelo ambiente a ser varrido, desviando de possíveis objetos que venham a entrar em seu caminho e realizar curvas em paredes.

Enquanto faz sua ronda, ele frequentemente deve fazer varreduras das concentrações gasosas detectáveis pelos sensores *MQ2*, *MQ3* e *MQ135*.

Com os dados obtidos das varreduras ele deve julgar se estes estão dentro ou fora de uma margem estabelecida e específica de concentração para cada sensor de gás.

Caso o ambiente em que ele está inserido esteja dentro da margem, o robô continuará em sua ronda, se movimentando pelo ambiente e desempenhando nenhum sinal visível ou sonoro.

Entretanto, caso seja detectada alguma concentração fora da margem e independente do sensor ou gás, o robô terá seu movimento cessado totalmente e

desempenhará sinais visíveis e sonoros até que alguém o perceba no ambiente e o venha reiniciá-lo para que ele possa continuar com sua ronda.

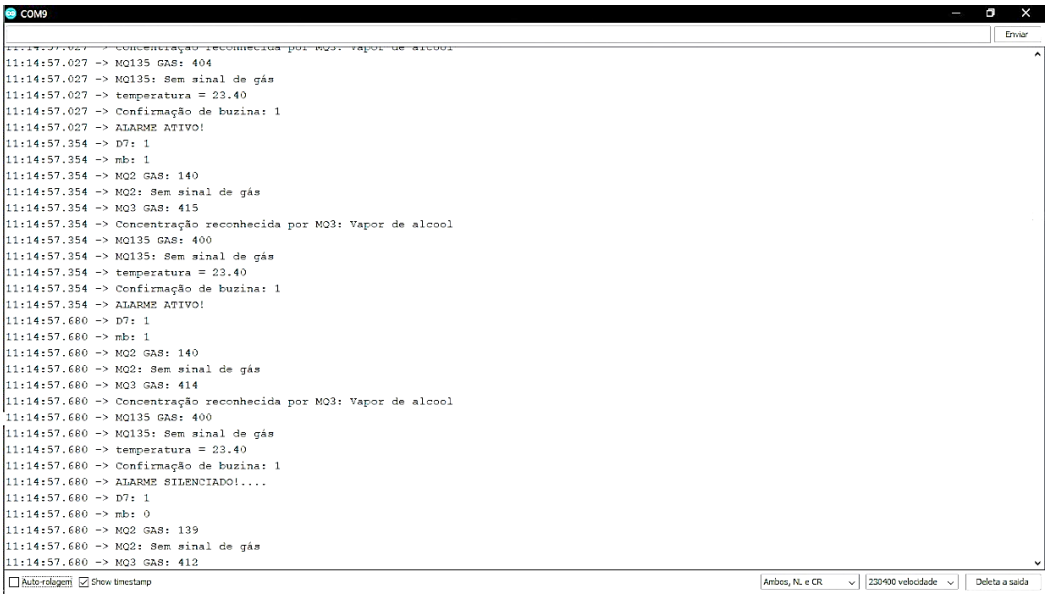
Através destas diretrizes o programa foi modelado e executado. É estruturado basicamente em dois blocos, o sensoriamento, responsável pelas varreduras de gases, temperaturas, controle de ativação do robô e tratamento destes dados obtidos e o controle de movimentação robótica, que agrupa toda a lógica de movimento e padrões direcionais.

4.3.1. Sensoriamento

Para execução do bloco, foi utilizada a função que recebe seu nome, sensoriamento. É a única função do bloco, nela as saídas analógicas provindas dos sensores tanto de gases quanto temperatura são lidas e conferidas se estão dentro das margens estabelecidas através de comparações usando comandos *IF*, menos o sinal de temperatura, que é apenas lido no atual instante.

Nela também foi construído um sistema para leitura do botão de ativação do robô, o efeito de *bounce* foi levado em conta durante o ato de se apertar o botão. O aperto sobre o botão proporciona a inversão dos valores de *memoria_Botao* utilizada para desativar os alarmes e D7, uma das diretrizes de direção, que é caracterizada pelo bloqueio do movimento do robô enquanto verdadeira. Importante ressaltar que caso o robô não tenha a variável *ativa_Buzina* verdadeira, que é obtida quando as concentrações estão acima da margem e a variável memoria_*Botao* falsa, pressionar o botão apenas tornará a variável D7 verdadeira, causando o travamento do robô, mas quando ativa_*Buzina* estiver verdadeira e *memoria_Botao* verdadeira, pressionar o botão irá tornar falso *ativa_Buzina* desativando os alarmes, para reativar o uso de alarmes pelo robô é necessário apertar novamente o botão. Pode-se ver a o funcionamento do bloco na figura 31.

Figura 31: Funcionamento do bloco sensoriamento no monitor serial

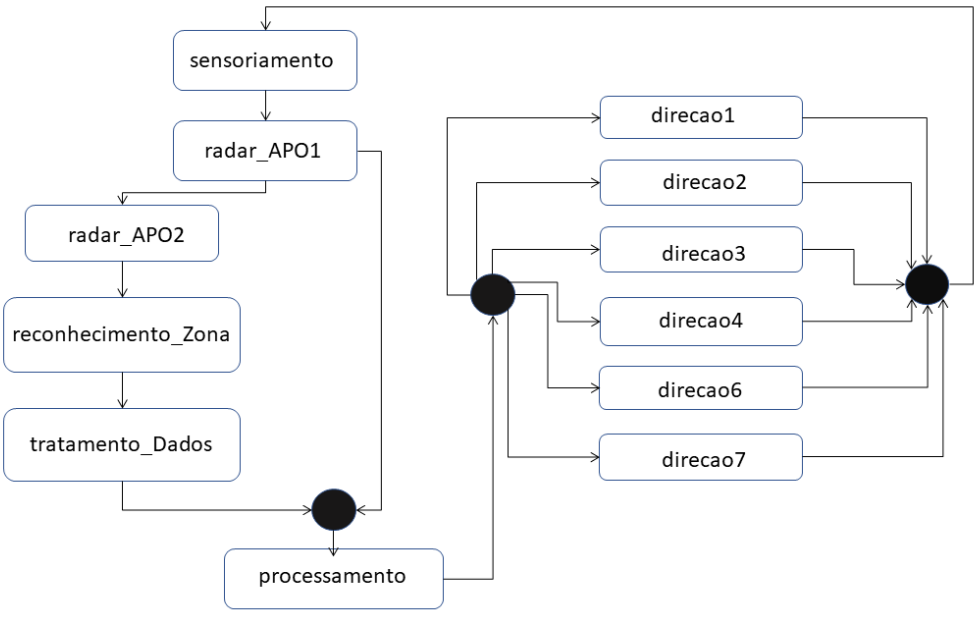


Fonte: Autores, 2022.

4.3.2. Controle de movimento

O bloco, diferente em relação ao de sensoriamento, foi construído utilizando onze funções distintas, divididas pelas seguintes classes mostradas em um fluxograma, conforme mostra a figura 32.

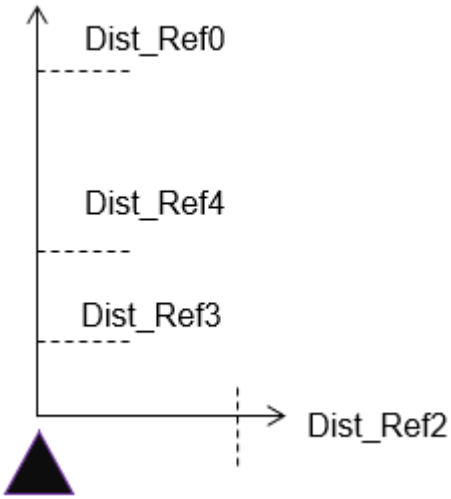
Figura 32: Classes utilizadas na programação.



Fonte: Autores, 2022.

Para estabelecer a movimentação do robô foi preciso utilizar retas referenciais para identificação de obstáculos como mostra a figura 33.

Figura 33: Retas referenciais para identificação de obstáculos

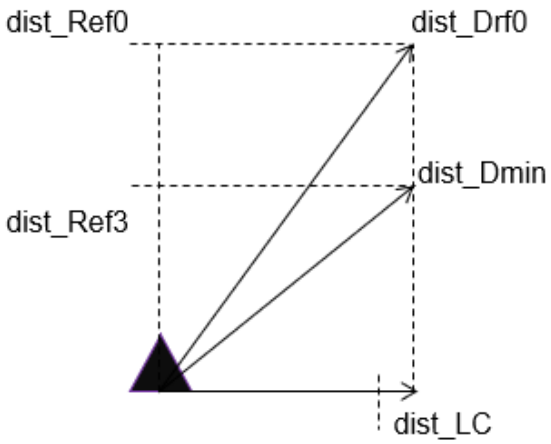


Fonte: Autores, 2022.

- Dist_Ref0: referencial para se considerar se há ou não um obstáculo à frente.
- Dist_Ref2: referencial lateral para o robô compreender se há ou não uma parede na lateral.
- Dist_Ref3: referencial para ativar frenagem ou a ré.
- Dist_Ref4: referencial para excedente de ré.

A figura 34 mostra as diagonais referenciais necessárias segundo a lógica do robô, para o desempenho de curvas.

Figura 34: Diagonais referenciais para execução de curvas

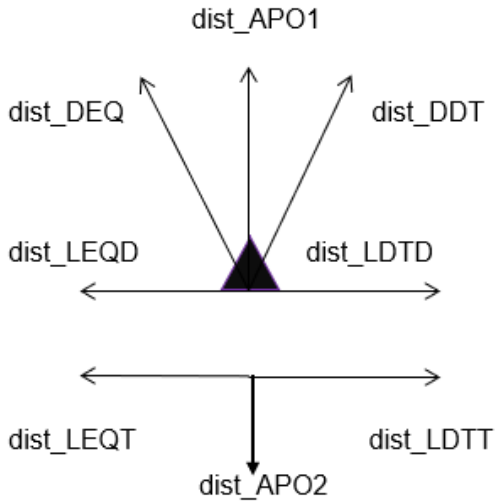


Fonte: Autores, 2022.

- Dist_LC: valor referencial da bissetriz longitudinal do robô até a lateral da sua extremidade mais valor de excedente, define o quão distante das paredes o robô deve estar durante o movimento de curvas.
- Dist_Drf0 : valor da hipotenusa do triangulo referente a distância dist_Ref0 e largura dist_LC, informação usada para que o robô entenda que caso essa distância seja obtida durante a execução de uma curva, o movimento de curva deve ser cessado.
- Dist_Dmin: valor da hipotenusa do triangulo referente a distância dist_Ref3 e largura dist_LC, dado usado pela lógica do robô para que ele considere se é ou não possível efetuar uma curva.

A figura 35 mostra quais foram as medidas necessárias para a varredura dos 2 sensores *HCSR04.*

Figura 35: Medidas usadas para a varredura dos sensores



Fonte: Autores, 2022.

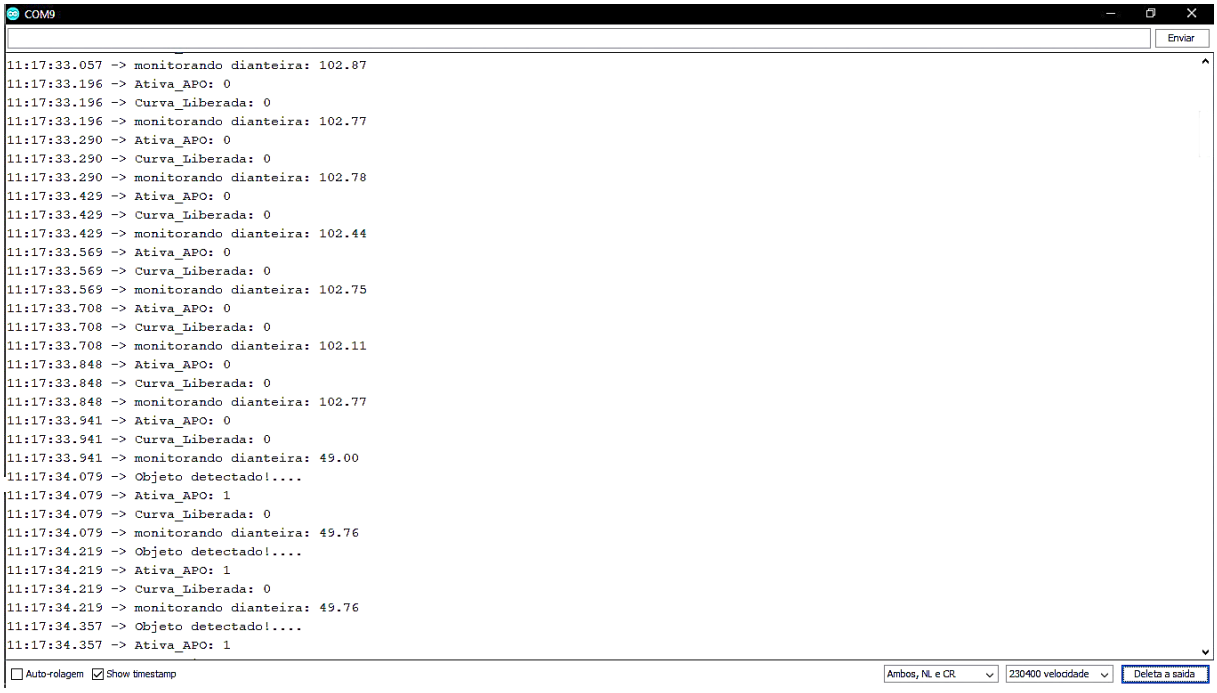
- Dist_APO1: varredura dianteira distância em noventa graus de servo motor.
- Dist_APO2: varredura traseira distância em noventa graus de servo motor.
- Dist_DDT: varredura dianteira distância em sessenta graus de servo motor.
- Dist_DEQ : varredura dianteira distância em cento e vinte graus de servo motor.
- Dist_LEQD: varredura dianteira distância em cinco graus de servo motor.
- Dist_LDTD : varredura dianteira distância em cento e setenta e cinco graus de servo motor.
- Dist_LEQT : varredura traseira distância em cento e setenta e cinco graus de servo motor.
- Dist_LDTT : varredura traseira distância em cinco graus de servo motor.

O radar de ativação por obstáculo ou radar_APO1 é uma função de controle de distância do robô, que frequentemente mede o dado de distância na dianteira fornecido pelo HCSR04 frontal e alinhado de noventa graus no servo. Em sua lógica compara a distância varrida nomeada como dist_APO1 com a distância de referência, dist_Ref0, caso a distância varrida seja menor que a referencial o robô entende que

há um obstáculo à sua frente e torna uma chave de acesso verdadeira para desbloquear as futuras funções a serem feitas, esta chave é a variável ativa_APO. Caso a distância seja maior que a referencial, a variável chave recebe o valor falso e não permite a execução das outras variáveis de reconhecimento de distancias.

O funcionamento do radar_APO1 pode ser visto por meio do monitor serial, conforme mostra a figura 36.

Figura 36: Funcionamento do radar_APO1 no monitor serial



Fonte: Autores, 2022.

O radar de ativação por obstáculo 2 ou radar_APO2 possui basicamente a mesma lógica que o primeiro, todavia, este não funciona frequentemente, só é permitida sua execução caso a variável direção D6 seja verdadeira, que é quando o robô está dando ré e sua lógica para comparação é a distância referencial dist_Ref3, caso dist_APO2 que o valor lido pelo HCSR04 na traseira alinhado de noventa graus seja maior que dist_Ref3, o robô tem seu movimento de ré continuado, já quando for menor a variável recuo_Max torna-se verdadeira, desativando o movimento de ré e como a ré deixará de ser o modelo de direção, monitora_Traseira que ministra o funcionamento da leitura na traseira enquanto verdadeira é tornado falso.

A função reconhecimento_Zona é exclusivamente chamada apenas quando ativa_APO é verdadeira e curva_Liberada é falsa, esta última refere-se ao robô estar

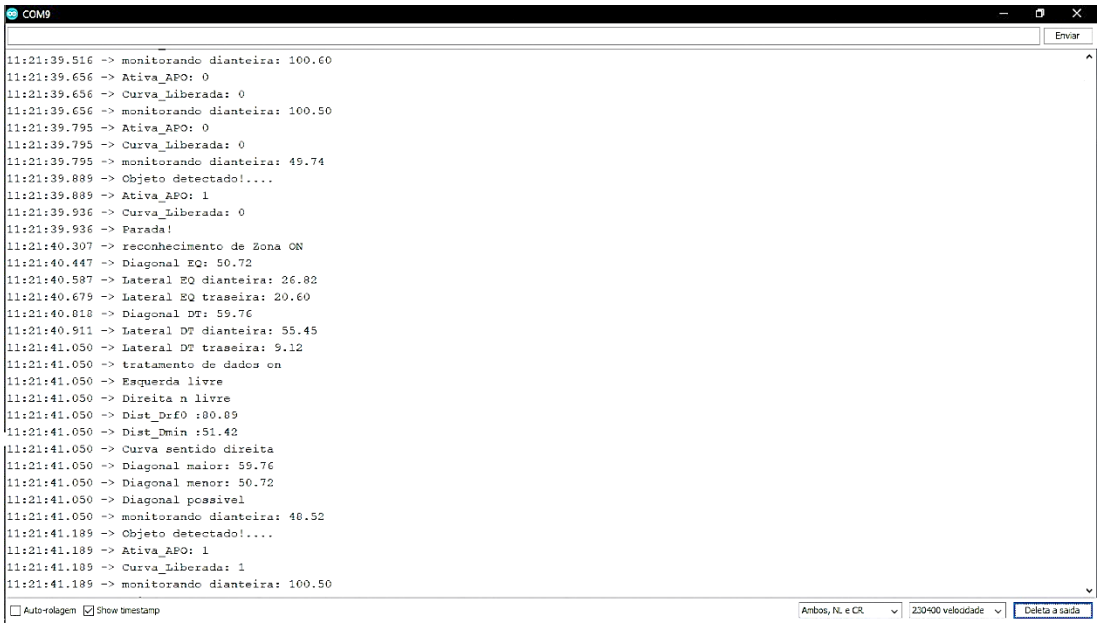
fazendo uma curva ou não, nela são obtidas as distancias nas diagonais frontal do robô, tanto em sessenta graus de alinhamento no servo motor quanto em cento e vinte, além das medidas laterais em relação ao robô, tanto na traseira quanto na dianteira pelos dois HCSR04. A figura 37 mostra as distâncias obtidas no monitor serial.

Figura 37: Funcionamento do bloco reconhecimento_Zona no monitor serial

Fonte: Autores, 2022.

A lógica presente em tratamento_Dados refere-se a tomada de decisão em relação as curvas, tanto se há espaço suficiente nas laterais do robô, a escolha de qual o melhor lado a seguir em relação ao desvio de um objeto e se é possível a realização da curva com os valores registrados. Esta função semelhantemente a reconhecimento_Zona é exclusivamente chamada apenas quando ativa_APO for verdadeira e curva_Liberada falsa. Na figura 38 é possível ver a lógica do bloco.

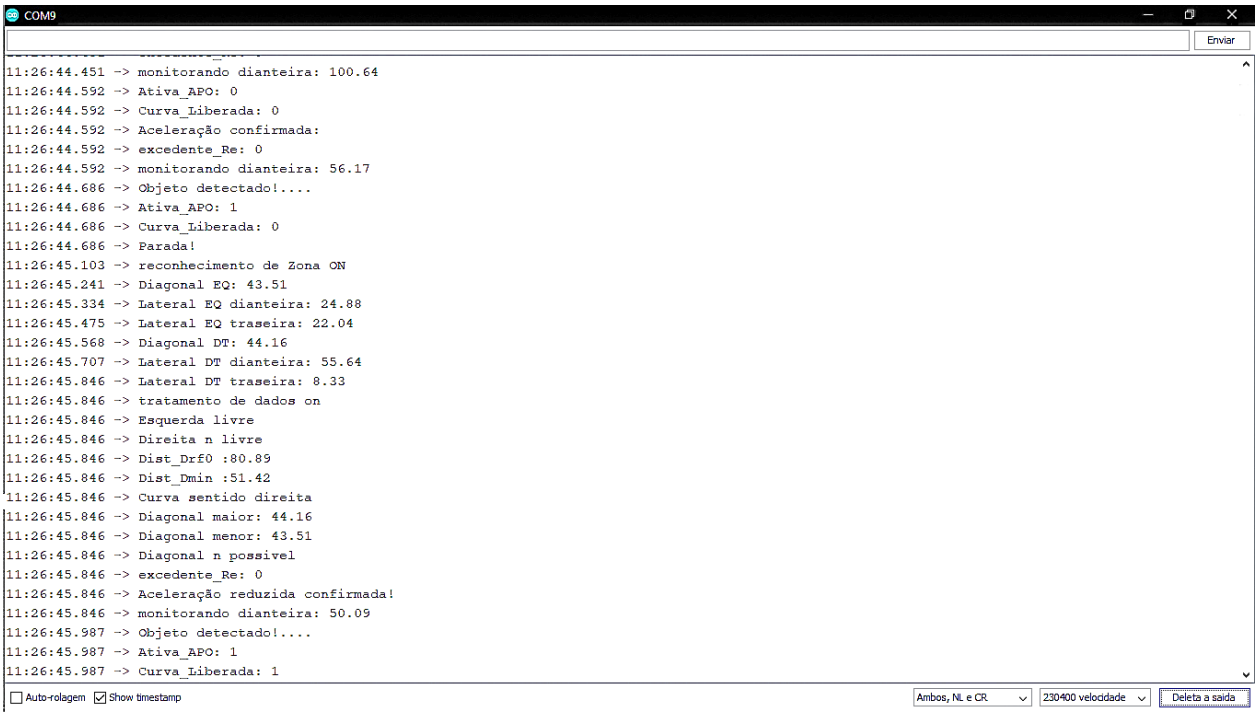
Figura 38: Resultados do bloco tratamento_Dados no monitor serial.



Fonte: Autores, 2022.

A função processamento recebe todas as distancias medidas e é chamada frequentemente pelo sistema e através de comparações entre as variáveis especificas estabelece qual o melhor modelo de movimento a ser utilizado pelo robô com base nos dados obtidos. A figura 39 mostra o funcionamento da função processamento no IDE.

Figura 39: Funcionamento da função processamento na IDE.

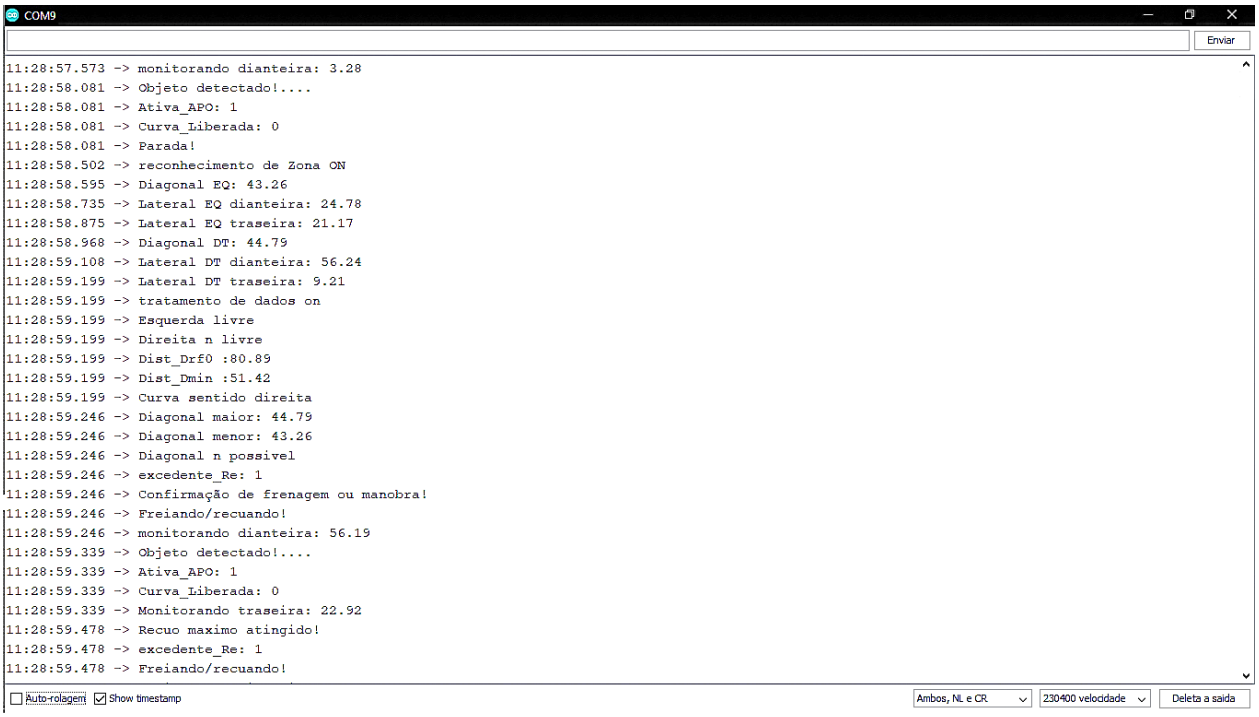


Fonte: Autores, 2022.

As funções direções atuam diretamente sobre os motores traseiro e dianteiro e definem o movimento que o robô há de fazer. A principal, direcao1 refere-se quando não há nenhum obstáculo à frente do robô, logo sua aceleração é aumentada. direcao2 e direcao3 estão relacionadas com o movimento de curva para a esquerda e direita respectivamente com aceleração reduzida. direcao4 produz uma aceleração reduzida em um movimento reto e por fim, direcao7, responsável pela parada do robô.

Na figura 40 é possível ver o funcionamento das lógicas direções no monitor serial da *IDE* do *Arduino*.

Figura 40: Monitor serial com as funções direções



Fonte: Autores, 2022.

5. RESULTADOS

Este capítulo contém todos os resultados esperados e obtidos em relação ao projeto.

5.1. Resultados Esperados

A ideia original do trabalho era que o robô monitorasse os níveis de concentração com a visualização sendo por meio do LED e do *Buzzer* presentes,

servindo como um alarme e em caso de acionamento, o robô ficaria parado até que fosse liberado para rodar novamente.

Em relação a movimentação do robô, foi pensado para que ele andasse de forma autônoma e por meio de sensores ultrassônicos, ele ter independência na hora da escolha de qual caminho seguir, bem como fazer toda a movimentação para desviar de obstáculos.

5.2. Resultados Obtidos

Conseguimos desenvolver todo o algoritmo de movimentação, fazendo com que o carrinho faça a detecção automática de objetos, tomando a decisão na hora de fazer as curvas, ré, aceleração e frenagem.

Conseguimos fazer a leitura da concentração que vem dos sensores, e dessa forma foi construída uma lógica que aciona o *Buzzer* e o LED para indicar quando o valor ultrapassou o limite e parando o robô imediatamente.

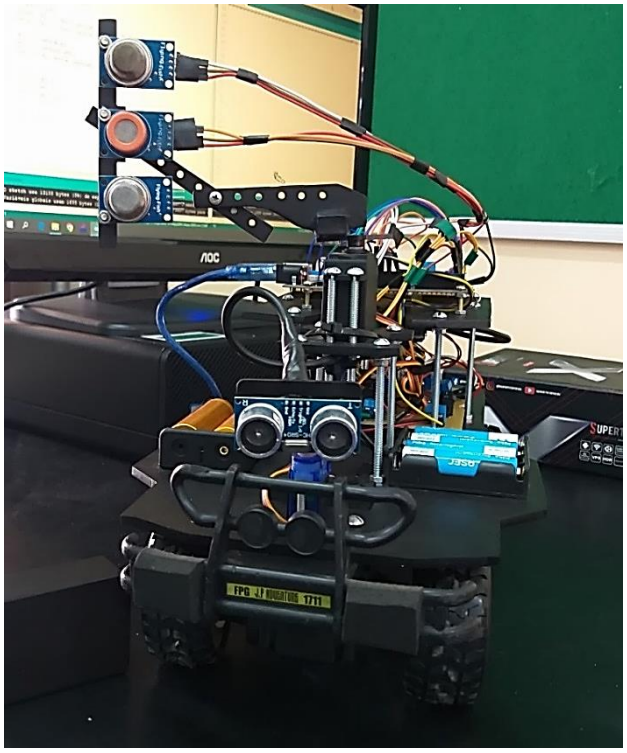
Por último, confeccionamos um chassi que foi feito de acrílico e que foi pintado, utilizando tinta preta, o que possibilitou a montagem de todos os componentes na base.

Em relação ao custo do projeto, por se tratar de um protótipo, tivemos a vantagem de já termos alguns componentes em mãos, e complementando com os materiais adquiridos, o custo ficou na faixa entre 700 e 800 reais para ser construído.

5.3. Vistas do robô

A figura 41 mostra a foto do robô visto de frente, é possível ver todos os sensores e as principais ligações feitas.

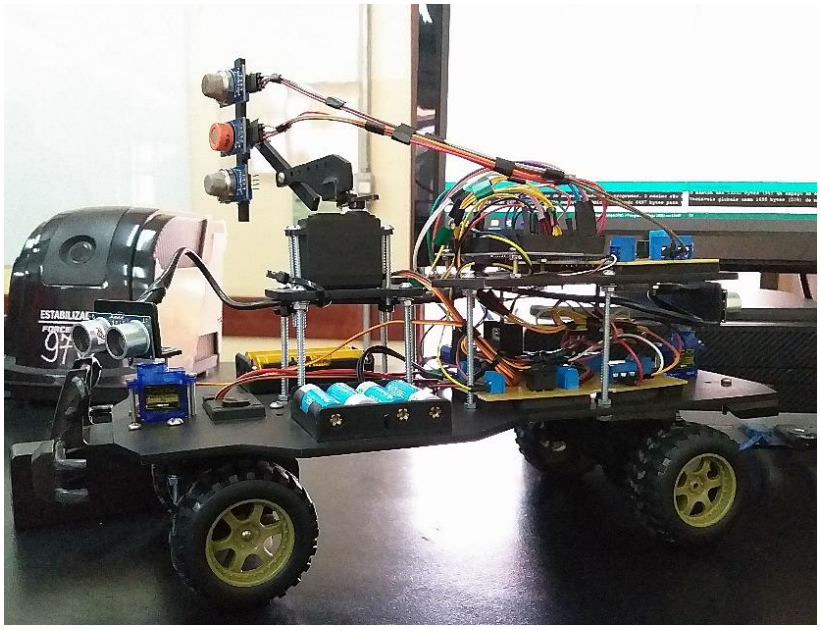
Figura 41: Robô visto frontalmente



Fonte: Autores, 2022.

Já a figura 42 mostra o robô visto de forma lateral.

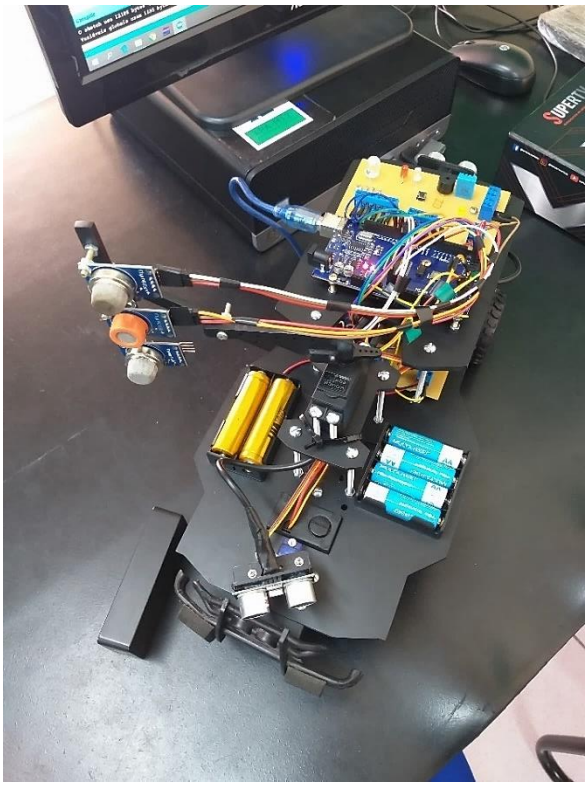
Figura 42: Robô visto lateralmente



Fonte: Autores, 2022.

A figura 43 mostra o robô visto de cima.

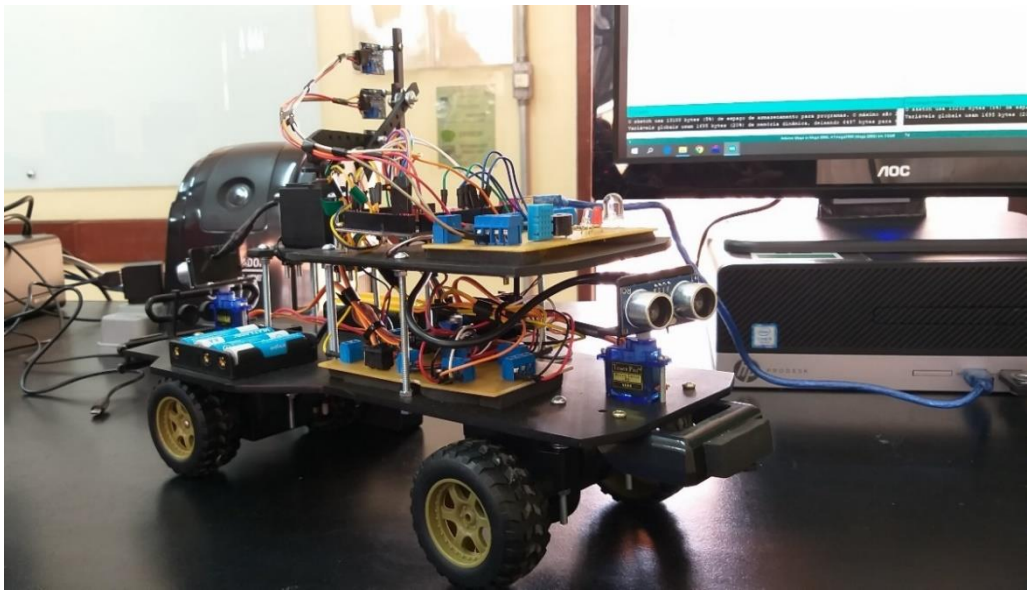
Figura 43: Robô visto de cima



Fonte: Autores, 2022.

E a figura 44 mostra o robô visto de trás.

Figura 44: Robô visto de trás



Fonte: Autores, 2022.

6. CONSIDERAÇÕES FINAIS

Este capítulo apresenta todas as considerações finais do trabalho e propostas futuras para melhorias do robô.

6.1. Conclusão

Este projeto consistia em desenvolver um robô para monitoramento de gases, que pudesse ser inserido em diversas frentes de trabalho, permitindo mais agilidade em medições e que sinalizasse quando tivesse um possível vazamento, tendo o seu funcionamento de maneira autônoma.

A meta foi conquistada, o robô opera de modo independente, é capaz de realizar manobras em situações em que ocorrer detecção de objetos em seu caminho e tem autonomia para operação.

A principal dificuldade encontrada foi solucionar algumas deficiências que necessitam de supervisão para evitar batidas e consequentemente a sua quebra ou travamento, que são os chamados pontos cegos, onde os sensores ultrassónicos não conseguem fazer a varredura e o robô não se movimenta, para isso, é necessário que o robô seja posicionado manualmente em outro ponto.

Destaca-se também que os sensores conseguiram realizar as leituras, e com base em um limite definido no código, é feito o acionamento de um LED e um Buzzer.

Para alertar quando há um valor acima do normal, ocorre um disparo sonoro com o Buzzer e o acionamento de um LED, parando o robô no mesmo instante

Com a finalização do projeto, fomos capazes de desenvolver um dispositivo que pode ser usado em meios industriais, comerciais e podem ser feitas diversas melhorias de modo a aumentar a eficiência e precisão tanto dos sensores e tanto quanto do robô.

6.2. Propostas Futuras

Com base nos resultados obtidos do projeto, as propostas futuras são:

- Desenvolvimento de um sistema de rastreamento para controle de rota

  Utilizar giroscópio, acelerômetro e um módulo de GPS para a criação de um sistema onde seja possível determinar qual a posição exata do robô e que ele se oriente com base no foco da concentração no ambiente em que estiver e se guiando na medida em que o valor aumente, até chegar no ponto com maior incidência.

- Blindagem dos componentes e modelagem da carroceria.

  Como todas as ligações ficaram expostas, é necessário que haja uma blindagem eletrostática, dessa forma protegendo todos os componentes eletroeletrônicos que estão descobertos. Pode-se também realizar a modelagem da estrutura em *software* e a impressão em 3D do chassi, permitindo encaixe exato para cada componente.

- Aprimoramento e otimização do algoritmo de desvio de obstáculos

  O código funcionou, porém, ainda existem melhorias que podem ser feitas. Foram utilizadas muitas linhas de código, então, pode-se otimizar o programa com o objetivo de realizar a mesma lógica no desvio, mas com menos funções, de forma a não sobrecarregar o microcontrolador.

- Uso de um módulo *WIFI* e criação de um *Dashboard*

  A ideia original do projeto consistia na utilização do *ESP* para enviar os dados para a plataforma *Blynk* porém devido a dificuldades na comunicação entre o módulo e o sistema robótico, este objetivo não foi alcançado, sendo assim, as possibilidades são, utilizar o *ESP8266* de forma individual com a criação de um servidor dedicado para visualização dos valores de concentração ou também outras plataformas *IoT*, como *Tago,io*, *Thingspeak* e *Arduino IoT* de modo a efetuar a integração entre o robô e o módulo *WIFI*.

7. REFERENCIAS BIBLIOGRÁFICAS

ATKINS, P. W.; JONES, Loretta**. Princípios de química: questionando a vida moderna e o meio ambiente**. 3.ed. Porto Alegre: Bookman, 2006. 965 p.

Buzzer. Disponível em: <https://www.usinainfo.com.br/buzzer-620#>. Acesso em: 08 jun. 2022.

CHAPMAN, Stephen J. **Fundamentos de máquinas elétricas**, 5ª edição, AMGH Editora Ltda, 2013.

CONSIGAZ. **O que é gás GLP**. Disponível em: <https://www.consigaz.com.br/gas-glp/>. Acesso em: 24 nov. 2021.

CRAVO, Edilson**. O que é um Servo Motor, como funciona e quais as vantagens?** Disponível em: https://blog.kalatec.com.br/o-que-e-servo-motor/. Acesso em: 08 jun. 2022.

DE SOUZA, Gustavo Rodrigues. **Termistores – NTC e PTC.** s.d. Curitiba. Disponível em <http://drbm.org/1mve/Laboratorio/TermistoresNTCePTC.pdf>. Acesso em: 4 nov. 2021.

ECYCLE. **Benzeno: o que é, riscos e como evitar a exposição**. Disponível em: <https://www.ecycle.com.br/benzeno/>. Acesso em: 24 nov. 2021.

GAMA GASES. **Amônia**. Disponível em: <http://www.gamagases.com.br/propriedades-dos-gases-amonia.html>. Acesso em: 23 nov. 2021.

GAMA GASES. **Dióxido de Carbono**. Disponível em: <http://www.gamagases.com.br/propriedades-dos-gases-dioxido-de-carbono.html>. Acesso em: 24 nov. 2021.

GAMA GASES. **Hidrogênio**. Disponível em: <http://www.gamagases.com.br/propriedades-dos-gases-hidrogenio.html>. Acesso em: 24 nov. 2021.

GAMA GASES. **Metano**. Disponível em: <http://www.gamagases.com.br/propriedades-dos-gases-metano.html>. Acesso em: 23 nov. 2021.

GAMA GASES. **Propano**. Disponível em: <http://www.gamagases.com.br/propriedades-dos-gases-propano.html>. Acesso em: 24 nov. 2021.

HANWEI ELECTRONICS. **Technical Data MQ-2 Gas Sensor**. s.d. Hanwei Eletronics Co. Ltd. Disponível em: <https://upverter.com/datasheet/05a4d494d8c28d681c71285ffeaa8c509a605037.pdf>. Acesso em: 13 out. 2021.

HANWEI ELECTRONICS. **Technical Data MQ-3 Gas Sensor**. s.d. Hanwei Eletronics Co. Ltd. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf>. Acesso em: 13 out. 2021.

HANWEI ELECTRONICS**. Technical Data MQ-135 Gas Sensor**. s.d. Hanwei Electronics Co. Ltd. Disponível em: <https://www.filipeflop.com/img/files/download/Sensor_de_gas_MQ-135.pdf>. Acesso em: 13 out. 2021.

NERY, Gustavo. **Guia definitivo de uso da Ponte H L298N**. Disponível em: <https://blog.eletrogate.com/guia-definitivo-de-uso-da-ponte-h-l298n/>. Acesso em: 23 nov. 2021.

SENSOR de Gás MQ-2 Inflamável e Fumaça. Disponível em: <https://www.filipeflop.com/produto/sensor-de-gas-mq-2-inflamavel-e-fumaca>. Acesso em: 14 out. 2021.

SENSOR de Gás MQ-3 Álcool. Disponível em: <https://www.filipeflop.com/produto/sensor-de-gas-mq-3-alcool>. Acesso em: 14 out. 2021.

SENSOR de Gás MQ-135 para Gases Tóxicos. Disponível em: <https://www.filipeflop.com/produto/sensor-de-gas-mq-135-para-gases-toxicos>. Acesso em: 14 out. 2021.

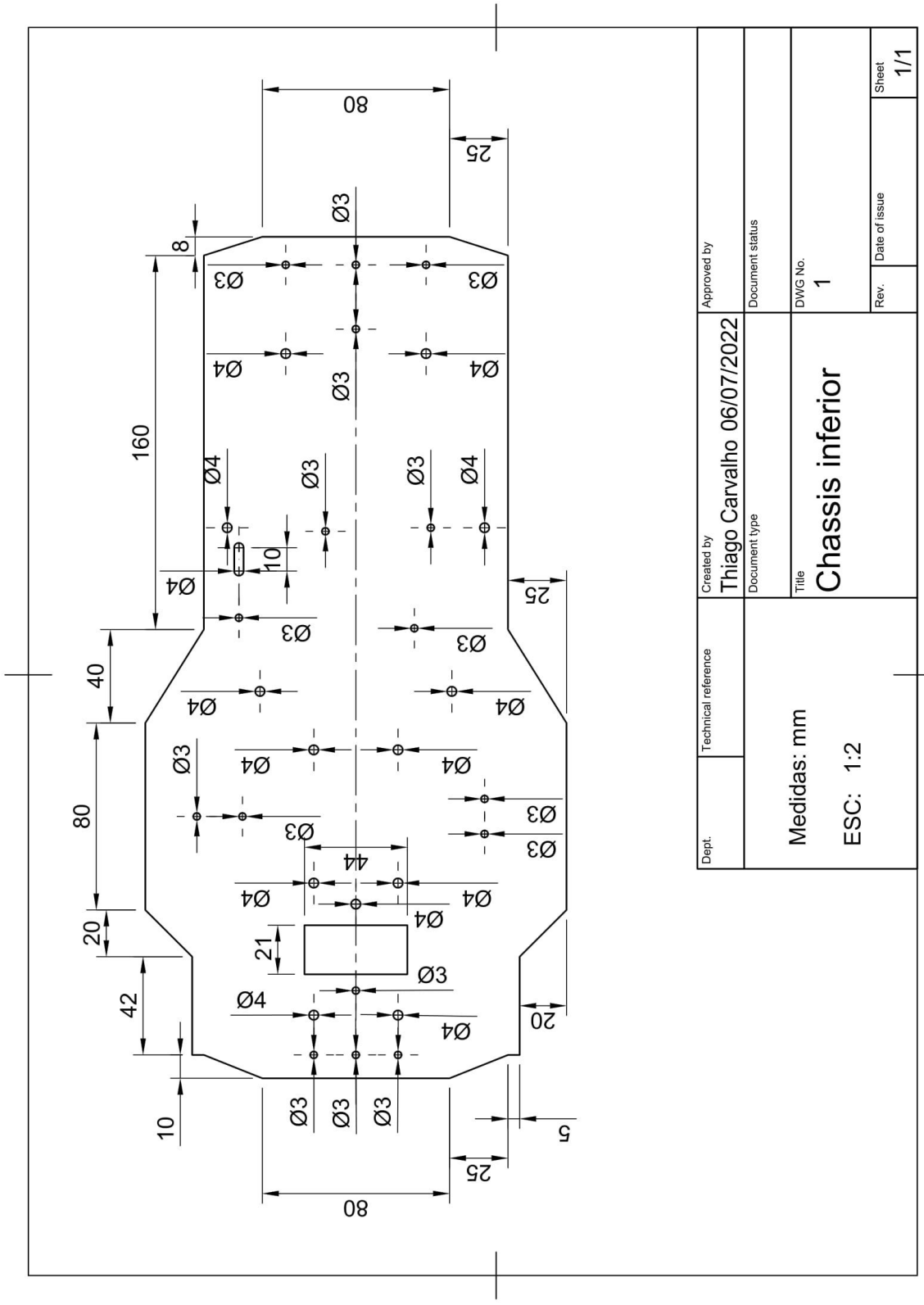
SENSOR DHT11 de Temperatura e Umidade. Disponível em: <https://www.robocore.net/sensor-ambiente/sensor-de-temperatura-dht11>. Acesso em: 4 nov. 2021.

SOUZA, Fábio. **Arduino MEGA 2560.** Disponível em: <https://www.embarcados.com.br/arduino-mega-2560/>. Acesso em: 07 out. 2021

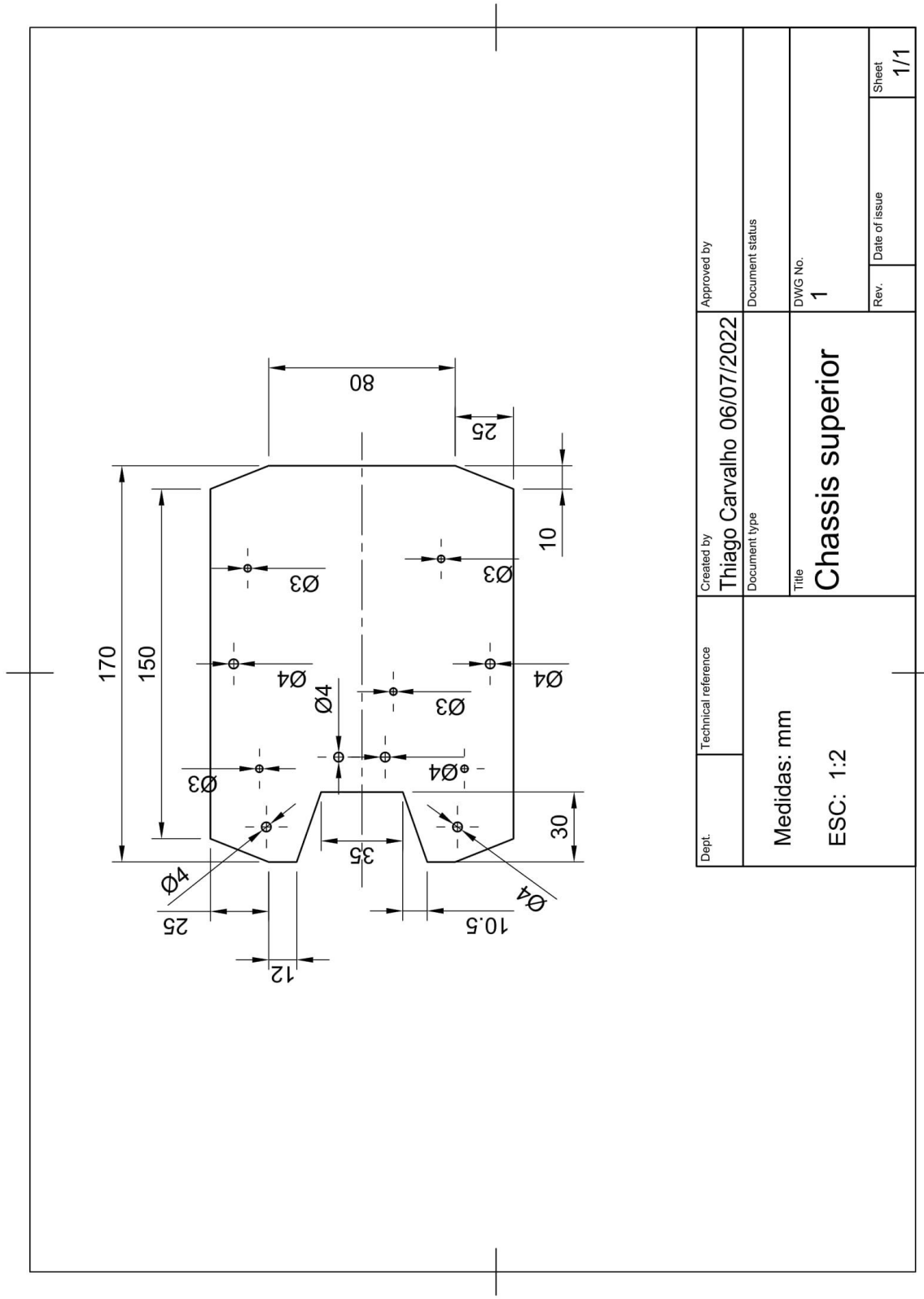
STEFAN, Igor Alexandre; FERREIRA, Paulo Ixtânio Leite; SOUSA, Aldeni Sudário de. **Protótipo de medidor de gases poluentes usando tecnologia de baixo custo**. Revista Princípia: Divulgação Científica e Tecnológica do IFPB, João Pessoa, v. 1, n. 49, p. 31-42, 25 mar. 2020. Disponível em: <https://periodicos.ifpb.edu.br/index.php/principia/article/viewFile/3872/1268>. Acesso em: 24 nov. 2021.

Vantagens e Limitações das Baterias de Lítio-Íon. Disponível em: <https://www.sta-eletronica.com.br/artigos/baterias-recarregaveis/baterias-de-litio/vantagens-e-limitacoes-das-baterias-de-litio-ion>. Acesso em: 08 jun. 2022.

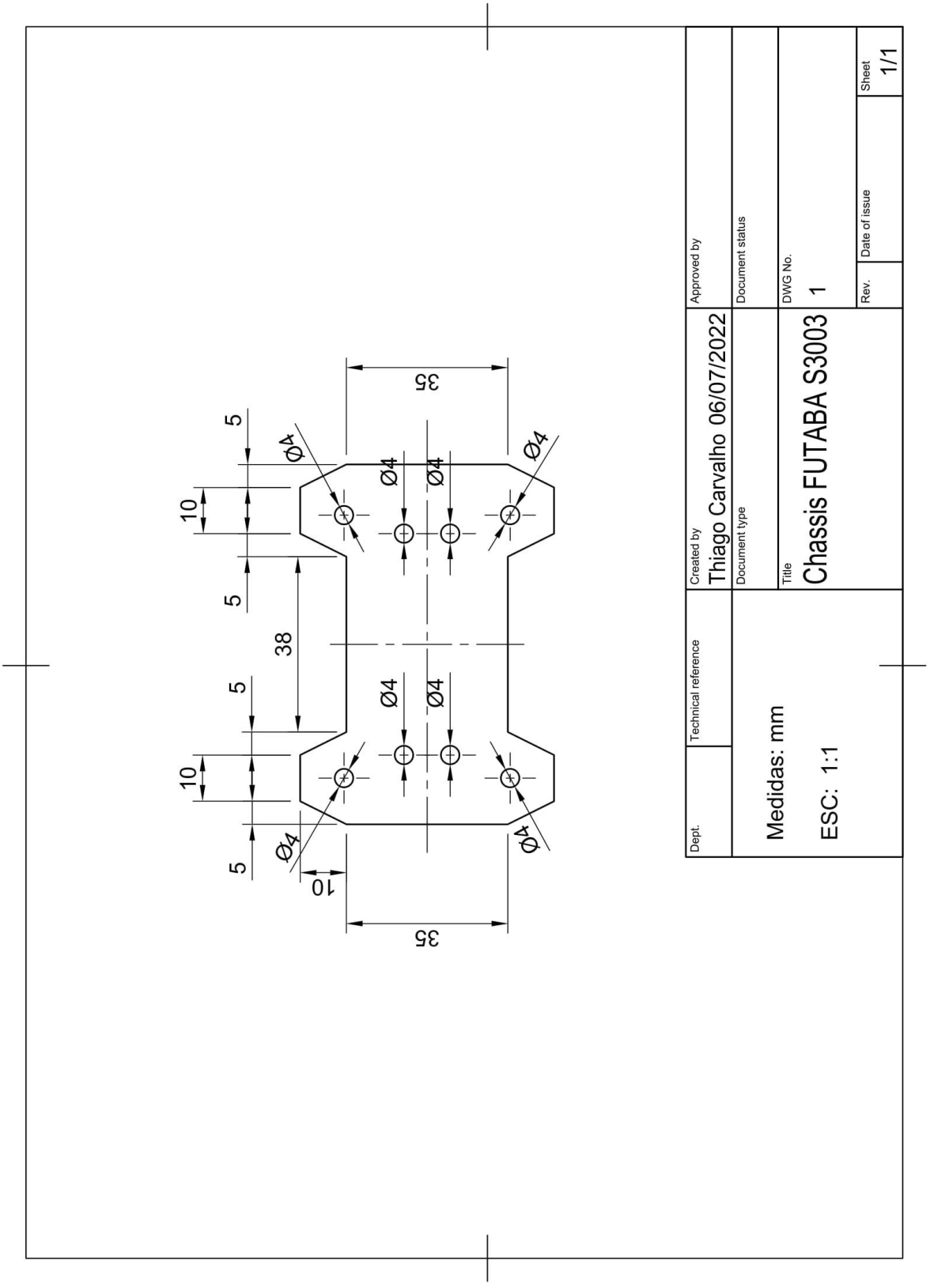
APÊNCIDE A – DESENHO CHASSIS INFERIOR



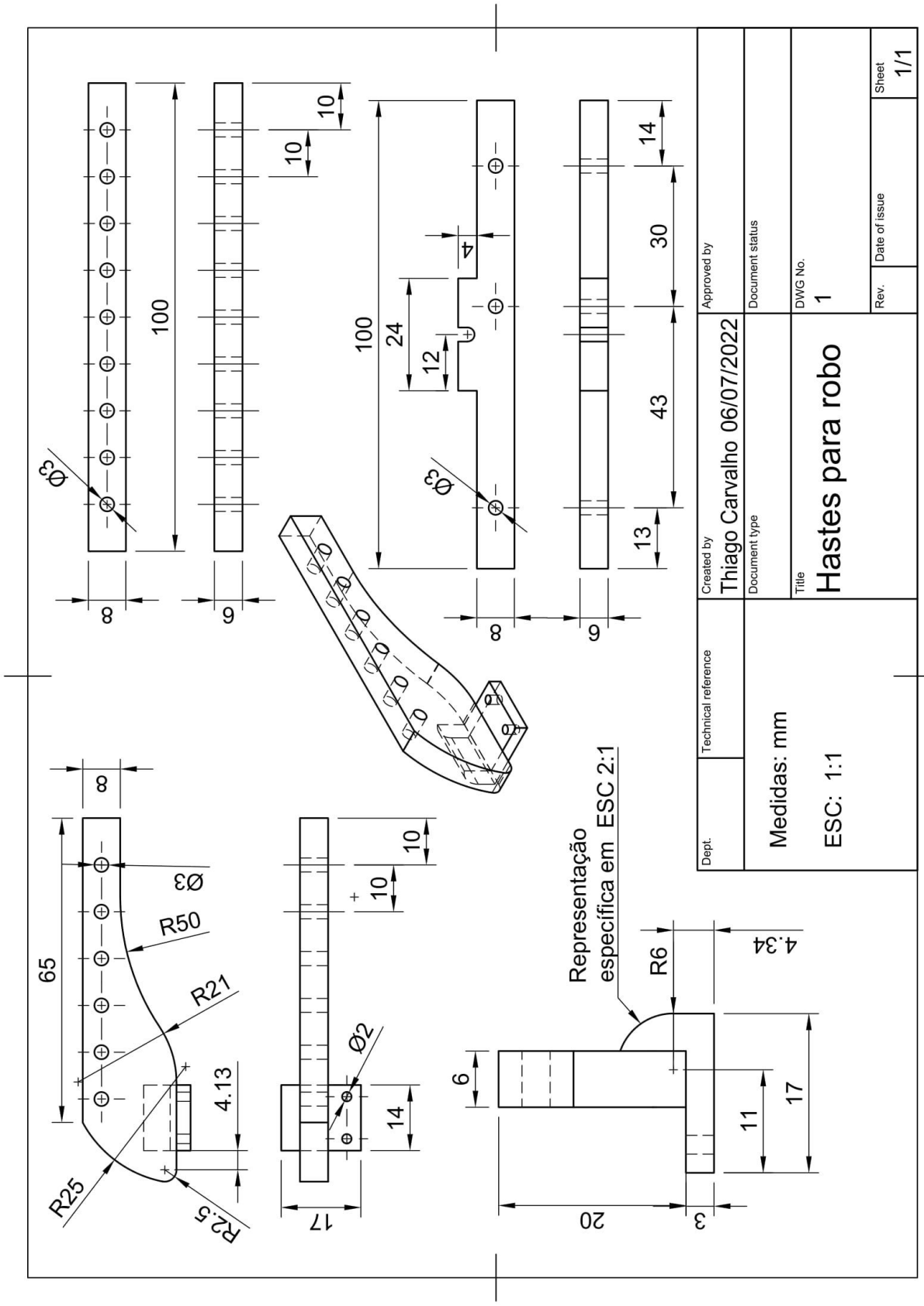
APÊNDICE B – DESENHO CHASSIS SUPERIOR



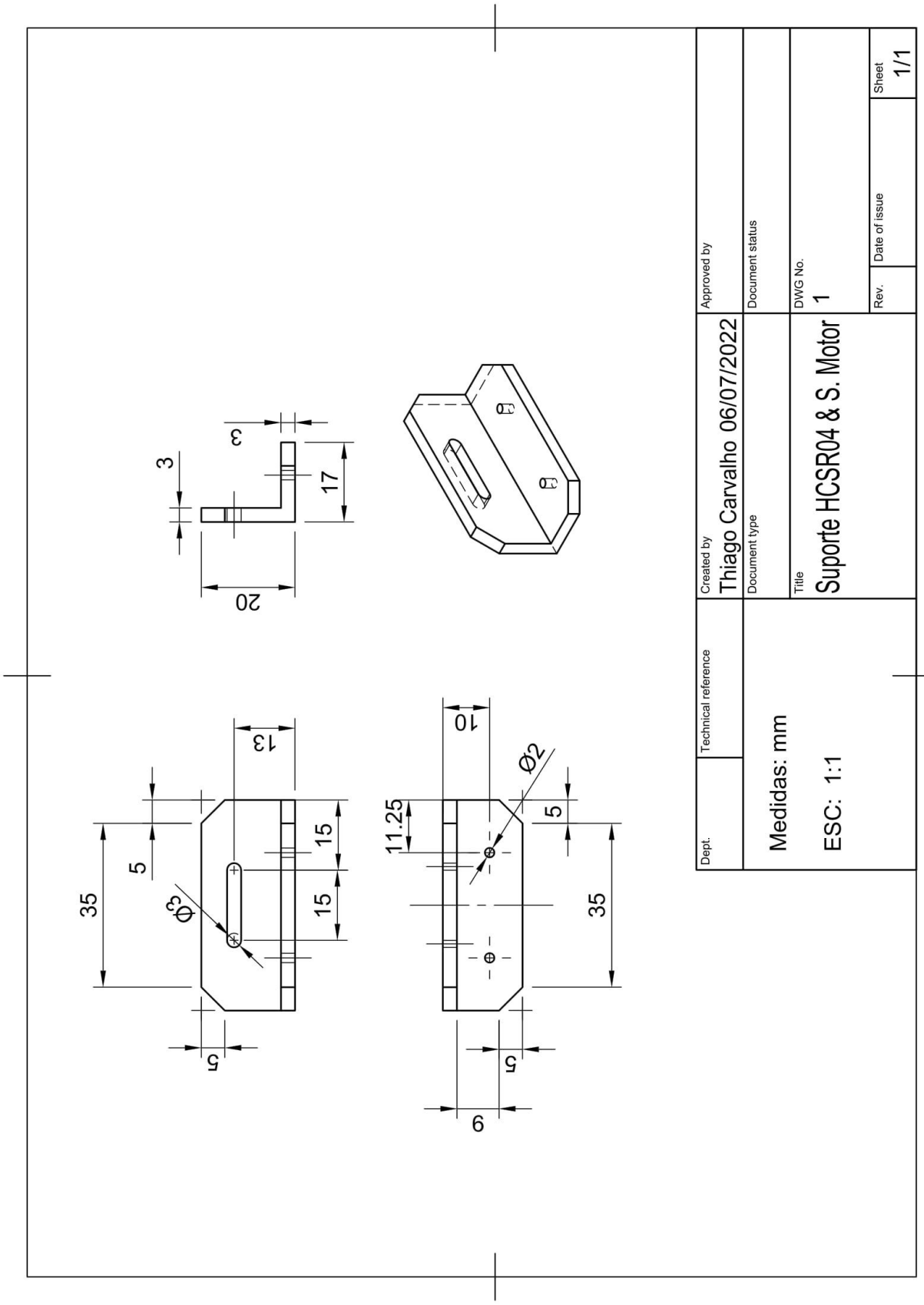
APÊNCIDE C – DESENHO CHASSIS BASE SERVO MOTOR



APÊNCIDE D – DESENHO HASTES SENSORES MQ



APÊNCIDE E – DESENHO SUPORTE HC SERVO MOTOR



APÊNDICE F – LISTA DE MATERIAIS

MATERIAIS UTILIZADOS COMPLETO

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Arruela de pressão M3	MAA	19	Barra pino 20 mm	EAA	4
Arruela de pressão M4	MAB	24	Bateria 18650 Li-íon 8000 mAh 3.7 V	EAB	2
Arruela reta M3	MAC	20	Bateria AA NimH 2500 mAh 1.2V	EAC	4
Arruela reta M4	MAD	24	Bateria power bank autores 5V	EAD	1
Chassis base S. M. Futaba	MAE	1	Botão NA convencional	EAE	1
Chassis inferior	MAF	1	Buzzer passivo	EAF	1
Chassis superior	MAG	1	Cabo PP 4 fios Ø0.5 mm 450mm	EAG	2
Eixo dianteiro RC	MAH	1	Chave liga e desliga 125 V 6 A	EAH	2
Eixo traseiro RC	MAI	1	Conector de parafuso 2 pinos 300 V 16 A	EAI	10
Engrenagem Futaba S3003	MAJ	1	Conector de parafuso 3 pinos 300 V 16 A	EAJ	5
Engrenagem S. M. SG90	MAK	2	DHT 11	EAK	1
Fita Hellerman (Braçadeira) 150 mm comprimento	MAL	4	Espuma para apoio da placa	EAL	1
Haste A	MAM	1	Estanho para solda Cobix	EAM	1
Haste B	MAN	1	Ferro de solda 45 W 127 V	EAN	1
Haste C	MAO	1	Fita isolante	EAO	1
Para-choque dianteiro	MAP	2	HCSR04	EAP	2
Para-choque traseiro	MAQ	3	Led alto brilho 5 mm	EAQ	1
Parafuso máquina c/ cabeça chata M3 22 mm	MAR	4	Led amarelo 5 mm	EAR	1
Parafuso máquina c/ cabeça cogumelo M4 40 mm	MAS	4	Led roxo 10 mm	EAS	1
Parafuso máquina c/ cabeça cogumelo M4 50 mm	MAT	4	Led vermelho 5 mm	EAT	1
Parafuso máquina c/ cabeça cogumelo M4 55 mm	MAU	4	Placa Arduino Mega	EAU	1
Parafuso máquina c/ cabeça cogumelo M4 75 mm	MAV	8	Placa circuito de energia	EAV	1
Parafuso máquina c/ cabeça panela M3 22 mm	MAW	14	Placa circuito de sensoriamento	EAW	1
Parafuso máquina c/ cabeça panela M3 30 mm	MAX	4	Placa de circuito impresso 110 mm * 60 mm	EAX	1
Parafuso S. M. Futaba	MAY	2	Placa de circuito impresso 120 mm * 60 mm	EAY	1
Parafuso S.M. SG90	MAZ	4	Ponte H L298N	EAZ	1
Porca simples M3	MBA	32	Resistor 560 Ω 5%	EBA	4
Porca simples M4	MBB	38	S. M. Futaba S3003	EBB	1
Suporte baterias 18650 2 pilhas	MBC	1	S. M. SG90	EBC	2
Suporte baterias AA 4 pilhas	MBD	1	Sens. MQ135	EBD	1
Suporte para bateria power bank autores	MBE	2	Sens. MQ2	EBE	1
			Sens. MQ3	EBF	1

MATERIAS UTILIZADOS PARA CONSTRUÇÃO DA HASTE DE APOIO PARA OS SENSORES MQ'S

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Arruela de pressão M3	MAA	2	Sens. Mq135	EBD	1
Arruela reta M3	MAC	3	Sens. MQ2	EBE	1
Engrenagem Futaba S3003	MAJ	1	Sens. MQ3	EBF	1
Haste A	MAM	1			
Haste B	MAN	1			
Haste C	MAO	1			
Parafuso máquina c/ cabeça panela M3 22 mm	MAW	5			
Parafuso S. M. Futaba	MAY	2			
Porca simples M3	MBA	13			

MATERIAS UTILIZADOS PARA CONSTRUÇÃO DOS SUPORTES PARA BATERIAS

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Arruela de pressão M3	MAA	4	Bateria 18650 Li-íon 8000 mAh 3.7 V	EAB	2
Arruela reta M3	MAC	4	Bateria AA NimH 2500 mAh 1.2V	EAC	4
Parafuso máquina c/ cabeça chata M3 22 mm	MAR	4	Bateria power bank autores 5V	EAD	1
Porca simples M3	MBA	4			
Suporte baterias 18650 2 pilhas	MBC	1			
Suporte baterias AA 4 pilhas	MBD	1			
Suporte para bateria power bank autores	MBE	2			

MATERIAS UTILIZADOS PARA A CONEXÃO DAS PLACAS NO CHASSIS INFERIOR

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Arruela de pressão M3	MAA	4	Ponte H L298N	EAZ	1
Arruela reta M3	MAC	4	Placa circuito de energia	EAV	1
Parafuso máquina c/ cabeça panela M3 22 mm	MAW	4			
Porca simples M3	MBA	4			

MATERIAS UTILIZADOS PARA A ESTRUTURA DO CHASSI SUPERIOR

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Arruela de pressão M4	MAB	8	VAZIO		
Arruela reta M4	MAD	8			
Chassis superior	MAG	1			
Fita Hellerman (Braçadeira) 150 mm comprimento	MAL	1			
Parafuso máquina c/ cabeça cogumelo M4 75 mm	MAV	4			
Porca simples M4	MBB	12			

MATERIAS UTILIZADOS PARA OS CONJUNTOS HCSR04'S E SERVO MOTORES

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Engrenagem S. M. SG90	MAK	2	Barra pino 20 mm	EAA	4
Parafuso máquina c/ cabeça panela M3 30 mm	MAX	4	Cabo PP 4 fios Ø0.5 mm 450mm	EAG	2
Parafuso S.M. SG90	MAZ	4	Fita isolante	EAO	
Porca simples M3	MBA		HCSR04	EAP	2
		4	S. M. SG90	EBC	2

MATERIAS UTILIZADOS PARA CONEXÃO DAS PLACAS NO CHASSI SUPERIOR

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Arruela de pressão M3	MAA	5	Placa Arduino Mega	EAU	1
Arruela reta M3	MAC	5	Placa circuito de sensoriamento	EAW	1
Parafuso máquina c/ cabeça panela M3 22 mm	MAW	5			
Porca simples M3	MBA	11			

MATERIAS UTILIZADOS NA ESTRUTURA DO CHASSIS PARA A HASTE DOS SENSORES

Mecânicos			Eletrônicos		
Nome	Identificação	Qtd.	Nome	Identificação	Qtd.
Arruela de pressão M4	MAB	8	S. M. Futaba S3003	EBB	1
Arruela reta M4	MAD	8			
Chassis base S. M. Futaba	MAE	1			
Fita Hellerman (Braçadeira) 150 mm comprimento	MAL	3			
Parafuso máquina c/ cabeça cogumelo M4 50 mm	MAT	4			
Parafuso máquina c/ cabeça cogumelo M4 75 mm	MAV	4			
Porca simples M4	MBB	16			

MATERIAS UTILIZADOS PARA A ESTRUTURA DO CHASSIS INFERIOR

Mecânicos

Nome	Identificação	Qtd.
Arruela de pressão M3	MAA	4
Arruela de pressão M4	MAB	8
Arruela reta M3	MAC	4
Arruela reta M4	MAD	8
Chassis inferior	MAF	1
Eixo dianteiro RC	MAH	1
Eixo traseiro RC	MAI	1
Para-choque dianteiro	MAP	2
Para-choque traseiro	MAQ	3
Parafuso máquina c/ cabeça cogumelo M4 40 mm	MAS	4
Parafuso máquina c/ cabeça cogumelo M4 55 mm	MAU	4
Parafuso máquina c/ cabeça panela M3 22 mm	MAW	
Porca simples M3	MBA	
Porca simples M4	MBB	10

Eletrônicos

Nome	Identificação	Qtd.

VAZIO

MATERIAS UTILIZADOS PARA CONSTRUÇÃO DA PLACA CIRCUITO DE ENERGIA

Mecânicos

Nome	Identificação	Qtd.

VAZIO

Eletrônicos

Nome	Identificação	Qtd.
Chave liga e desliga 125 V 6 A	EAH	2
Conector de parafuso 2 pinos 300 V 16 A	EAI	5
Conector de parafuso 3 pinos 300 V 16 A	EAJ	3
Espuma para apoio da placa	EAL	
Estanho para solda Cobix	EAM	
Ferro de solda 45 W 127 V	EAN	
Placa de circuito impresso 120 mm * 60 mm	EAY	1

MATERIAS UTILIZADOS PARA CONSTRUÇÃO DA PLACA CIRCUITO DE SENSORIAMENTO

Mecânicos

Nome	Identificação	Qtd.

VAZIO

Eletrônicos

Nome	Identificação	Qtd.
Botão NA convencional	EAE	1
Buzzer passivo	EAF	1
Conector de parafuso 2 pinos 300 V 16 A	EAI	5
Conector de parafuso 3 pinos 300 V 16 A	EAJ	2
DHT 11	EAK	1
Espuma para apoio da placa	EAL	
Estanho para solda Cobix	EAM	
Ferro de solda 45 W 127 V	EAN	
Led alto brilho 5 mm	EAQ	1
Led amarelo 5 mm	EAR	1
Led roxo 10 mm	EAS	1
Led vermelho 5 mm	EAT	1
Placa de circuito impresso 110 mm * 60 mm	EAX	1
Resistor 560 Ω 5%	EBA	4

APÊNDICE G – CÓDIGO FONTE DO PROJETO

```
// FATEC SANTO ANDRÉ                    --- COM APERFEIÇOAMENTO #2
// THIAGO CARVALHO & VITOR BORGES
// ROBÔ DETECTOR DE GÁS: SCRIPT FINAL #V1

//Bibliotecas:
#include <Servo.h> // Servo
#include <DHT.h>   // DHT11

//Definições:
Servo   S1; // Servo para HC dianteiro da frente.
Servo   S2; // Servo para HC dianteiro de tras.

#define DHTPIN A3
#define DHTTYPE DHT11
DHT     dht(DHTPIN, DHTTYPE);



#define MQ2 A0
#define MQ3 A1
#define MQ135 A2
#define Buzzer 15

#define led 2   // Led alarme gás
#define led1 8  // Led memoria_Botao
#define led2 9  // Led D7

//Variaveis:

  //Motor traseiro
int    in1 = 28;
int    in2 = 26;
byte   pino_Pwm1 = 7;
byte   pwm1;       // Velocidade do carro partida
```

```
byte   pwm11;     // Velocidade da ré

 //Motor dianteiro
int    in3 = 24;
int    in4 = 22;
byte   pino_Pwm2 = 6;
byte   pwm2 = 255;     // Intensidade angulação de roda

 //Conjunto Servo HC
int    trigger1 = 38;
int    echo1 = 40;
int    trigger2 = 39;
int    echo2 = 41;

 //Métricas

int    delay0 = 120;    // tempo de acomodação para o HC não dar erro de medida
int    delay1 = 100;    // tempo de acomodação para o HC não dar erro de medida
int    delay2 = 400;    // apenas um delay

float  cronos_Echo1;
float  cronos_Echo2;

float  dist_APO1;    // medida captada pelo HCSR04 da frente
float  dist_APO2;    // medida captada pelo HCSR04 de tras
float  dist_DEQ;     // medida captada diagonal frontal esquerda
float  dist_LEQD;    // medida captada na lateral esquerda da dianteira
float  dist_LEQT;    // medida captada na lateral esquerda da traseira
float  dist_LC = 12; // medida de referencia do centro do sensor até a lateral esquerda com sobra.
float  dist_DDT;     // medida captada diagonal frontal direita
float  dist_LDTD;    // medida captada na lateral direita da dianteira
float  dist_LDTT;    // medida captada na lateral direita da traseira
float  dist_Drf0;    // diagonal cessa curva
float  dist_Dmin;    // diagonal minima para fazer uma curva
float  dist_Dmaior;  // recebe o valor da diagonal da curva a se fazer
```

```
float  dist_Dmenor;  // recebe o valor da diagonal oposta

float  dist_Ref0 = 80;  // medida de referencia para distancia detecta obstaculo
float  dist_Ref2 = 20;  // medida de referencia para lateral detecta parede
float  dist_Ref3 = 50;  // medida de referencia para a ativar freio/ré
float  dist_Ref4 = dist_Ref3 + 20;  // medida para o excedente de ré

int    ang_0 = 90;      // Servo centrado
int    ang_1 = 135;     // APO esquerdo
int    ang_2 = 45;      // APO direito
int    ang_3 = 175;     // LATERAL esquerda
int    ang_4 = 5;       // LATERAL direito

int    leituraMQ2 = 0;
int    leituraMQ3 = 0;
int    leituraMQ135 = 0;

 //CONTROLE:
bool   monitora_Dianteira = true;
bool   monitora_Traseira = false;
bool   ativa_APO   = false;
bool   curva_Liberada = false;

 //DIREÇÃO:
bool   curva_EQ = false;
bool   eq_Livre = false;
bool   curva_DT = false;
bool   dt_Livre = false;
bool   curva_Possivel = false;
bool   manobra = false;
bool   recuo_Max = false;
bool   excedente_Re = false;

bool   memoria_EQ = false;
bool   memoria_DT = false;
```

```
int    orientacao_IN3 = 1;
int    orientacao_IN4 = 1;

bool   D1 = false;   // Acelera+
bool   D2 = false;   // Faz a curva para esquerda
bool   D3 = false;   // Faz a curva para direita
bool   D4 = false;   // Aceleracao reduzida
bool   D6 = false;   // Freio/acelera-
bool   D7 = true;    // Para

bool   gas_MQ2 = false;
bool   gas_MQ3 = false;
bool   gas_MQ135 = false;

bool   ativa_Buzina = false;

 //ACIONAMENTO:
int    tom;
int    estado_Botao;
int    botao = 4;
int    delay_Bounce = 100;

bool   memoria_Botao = false;
bool   estado_Atual = false;
bool   estado_ANT   = false;
unsigned long  delay_Botao  = 0;

//Funções ADD:
  //Sensoriamento:
void Sensoriamento(){

  leituraMQ2 = analogRead(MQ2); //Leitura sensor MQ2
  Serial.print("MQ2 GAS: ");
  Serial.println(leituraMQ2);
```

```
  if (leituraMQ2 > 350) {
    Serial.println("Concentração reconhecida por MQ2: Gases combustiveis ou fumaça");
    gas_MQ2 = true;
  } else {
    Serial.println("MQ2: Sem sinal de gás");
    gas_MQ2 = false;
  }


  leituraMQ3 = analogRead(MQ3); //Leitura sensor MQ3
  Serial.print("MQ3 GAS: ");
  Serial.println(leituraMQ3);
   if (leituraMQ3 >450) {
    Serial.println("Concentração reconhecida por MQ3: Vapor de alcool");
    gas_MQ3 = true;
  } else {
    Serial.println("MQ3: Sem sinal de gás");
    gas_MQ3 = false;
  }


  leituraMQ135 = analogRead(MQ135); //Leitura sensor MQ135
  Serial.print("MQ135 GAS: ");
  Serial.println(leituraMQ135);
   if (leituraMQ135 > 1200) {
    Serial.println("Concentração reconhecida por MQ135: TOXICIDADE ALTA: Possivel NH3 / NOX /
Alcool / BENZENO");
    gas_MQ3 = true;
  } else {
    Serial.println("MQ135: Sem sinal de gás");
    gas_MQ135 = false;
  }

  float temperatura = dht.readTemperature(); //Função para ler a temperatura no sensor DHT1
  Serial.print("temperatura = ");
```

```
        Serial.println(temperatura);

        if((gas_MQ2 == true || gas_MQ3 == true || gas_MQ135 == true) && D7 == false){ // Bloco para ativação da buzina e ordem de parada do robô, quando gas for detectado
          ativa_Buzina = true;
          D7 = true;
          }

        if(millis() - delay_Botao > delay_Bounce){   // Bloco para ativação do botao com controle de efeito bounce
          estado_Botao = digitalRead(botao);
          if(estado_Botao && (estado_Botao != estado_ANT)){
            memoria_Botao = !memoria_Botao;
            D7 = !memoria_Botao;
            delay_Botao = millis();
            }
          estado_ANT = estado_Botao;
          }

          Serial.print("Confirmação de buzina: ");
          Serial.println(ativa_Buzina);

        if(ativa_Buzina == true && memoria_Botao == true){ // Bloco controle do alarme
          Serial.println("ALARME ATIVO!");
          digitalWrite(led,1);
        digitalWrite(Buzzer,1);
          delay(110);
          digitalWrite(Buzzer,0);
          digitalWrite(Buzzer,1);
          delay(200);
          digitalWrite(Buzzer,0);
          digitalWrite(Buzzer,1);
          delay(20);
          digitalWrite(Buzzer,0);
          }
```

```
    else{
    Serial.println("ALARME SILENCIADO!....");
    ativa_Buzina = false;
    digitalWrite(Buzzer,0);
    digitalWrite(led,0);
     }
   Serial.print("D7: ");
   Serial.println(D7);
   Serial.print("mb: ");
   Serial.println(memoria_Botao);
   digitalWrite(led1,memoria_Botao);
  }

  //Movimentacao robótica:
 void radar_AP01(){ // Radar de ativação por objeto, dianteiro
  if(monitora_Dianteira == true && D7 == false){
    Serial.print("monitorando dianteira: ");
    S1.write(ang_0);
    S2.write(ang_0);
    delay(delay0);
    digitalWrite(trigger1,1);
    delayMicroseconds(12);
    digitalWrite(trigger1,0);
    cronos_Echo1 = pulseIn(echo1,1);
    dist_AP01 = cronos_Echo1/58.2;
    Serial.println(dist_AP01);

    if((dist_AP01 < dist_Ref0) && (dist_AP01 != 0)){ // Verificação se objeto há algum objeto no caminho, no raio de alcance dref0
      Serial.println("Objeto detectado!.... ");
      ativa_AP0 = true;
     }
     else{
       ativa_AP0 = false;
       manobra = false;
```

```
      memoria_EQ = false;
      memoria_DT = false;
     }
   if(ativa_AP0 == false){  // Bloco para finalizar curva
    curva_Liberada = false;
    }

    Serial.print("Ativa_AP0: ");
    Serial.println(ativa_AP0);
    Serial.print("Curva_Liberada: ");
    Serial.println(curva_Liberada);

  }
 }

void radar_AP02(){ // Radar de ativação por objeto, traseiro
 if(monitora_Traseira == true && D7 == false){
  Serial.print("Monitorando traseira: ");
  S2.write(ang_0);
  delay(delay0);
  digitalWrite(trigger2,1);
  delayMicroseconds(12);
  digitalWrite(trigger2,0);
  cronos_Echo2 = pulseIn(echo2,1);
  dist_AP02 = cronos_Echo2/58.2;
  Serial.println(dist_AP02);

  if(dist_AP02 < dist_Ref3){ // Bloco de controle de distancia traseiro, em função do raio dref3
   Serial.println("Recuo maximo atingido!");
   monitora_Traseira == false;
   recuo_Max = true;
   }
  else{
   monitora_Traseira = true;
   recuo_Max = false;
```

```
    }
   }
  }

void reconhecimento_Zona(){
  if(ativa_APO == true && curva_Liberada == false && manobra == false && memoria_EQ == false && memoria_DT == false && D7 == false && D6 == false){

    D7 = true;  // Breve pausa pré curva
    direcao7();
    D7 = false;

    Serial.println("reconhecimento de Zona ON");  //Segue a obtenção das diagonais e retas essenciais para movimentação
    S1.write(ang_1);
    delay(delay0);
    digitalWrite(trigger1,1);
    delayMicroseconds(14);
    digitalWrite(trigger1,0);
    cronos_Echo1 = pulseIn(echo1,1);
    dist_DEQ = (cronos_Echo1/58.2);
    Serial.print("Diagonal EQ: ");
    Serial.println(dist_DEQ);

    S1.write(ang_3);
    S2.write(ang_4);
    delay(delay0);
    digitalWrite(trigger1,1);
    delayMicroseconds(14);
    digitalWrite(trigger1,0);
    cronos_Echo1 = pulseIn(echo1,1);
    dist_LEQD = (cronos_Echo1/58.2);
    Serial.print("Lateral EQ dianteira: ");
    Serial.println(dist_LEQD);
```

```
delay(delay0);
digitalWrite(trigger2,1);
delayMicroseconds(14);
digitalWrite(trigger2,0);
cronos_Echo2 = pulseIn(echo2,1);
dist_LEQT = (cronos_Echo2/58.2);
Serial.print("Lateral EQ traseira: ");
Serial.println(dist_LEQT);

S1.write(ang_2);
delay(delay1);
digitalWrite(trigger1,1);
delayMicroseconds(14);
digitalWrite(trigger1,0);
cronos_Echo1 = pulseIn(echo1,1);
dist_DDT = (cronos_Echo1/58.2);
Serial.print("Diagonal DT: ");
Serial.println(dist_DDT);

S1.write(ang_4);
S2.write(ang_3);
delay(delay0);
digitalWrite(trigger1,1);
delayMicroseconds(14);
digitalWrite(trigger1,0);
cronos_Echo1 = pulseIn(echo1,1);
dist_LDTD = (cronos_Echo1/58.2);
Serial.print("Lateral DT dianteira: ");
Serial.println(dist_LDTD);

delay(delay0);
digitalWrite(trigger2,1);
delayMicroseconds(14);
digitalWrite(trigger2,0);
cronos_Echo2 = pulseIn(echo2,1);
```

```
  dist_LDTT = (cronos_Echo2/58.2);
  Serial.print("Lateral DT traseira: ");
  Serial.println(dist_LDTT);
  }
 }

 void tratamento_Dados(){ //Trata as diagonais e retas obtidas anteriormente
  if(ativa_APO == true && curva_Liberada == false && manobra == false && memoria_EQ == false &&
memoria_DT == false && D7 == false && D6 == false){
  Serial.println("tratamento de dados on ");
  if(dist_LEQD > dist_Ref2 && dist_LEQT > dist_Ref2){
   Serial.println("Esquerda livre");
   eq_Livre = true;
   }
   else{
    Serial.println("Esquerda n livre");
    eq_Livre = false;
    }
  if(dist_LDTD > dist_Ref2 && dist_LDTT > dist_Ref2){
   Serial.println("Direita livre");
   dt_Livre = true;
   }
   else{
    Serial.println("Direita n livre");
    dt_Livre = false;
    }

  dist_Drf0 = sqrt(sq(dist_LC)+sq(dist_Ref0));
  Serial.print("Dist_Drf0 :");
  Serial.println(dist_Drf0);

  dist_Dmin = sqrt(sq(dist_LC)+sq(dist_Ref3));
  Serial.print("Dist_Dmin :");
  Serial.println(dist_Dmin);
```

```
  if(dist_DEQ > dist_DDT){
   Serial.println("Curva sentido esquerda");
   curva_EQ = true;
   curva_DT = false;
   dist_Dmaior = dist_DEQ;
   dist_Dmenor = dist_DDT;
   }
   else{
    Serial.println("Curva sentido direita");
    curva_EQ = false;
    curva_DT = true;
    dist_Dmaior = dist_DDT;
    dist_Dmenor = dist_DEQ;
    }
    Serial.print("Diagonal maior: ");
    Serial.println(dist_Dmaior);
    Serial.print("Diagonal menor: ");
    Serial.println(dist_Dmenor);

  if(dist_Dmaior >= dist_Dmin){
   Serial.println("Diagonal possivel");
   curva_Possivel = true;
   }
   else{
    Serial.println("Diagonal n possivel");
    curva_Possivel = false;
    }
  curva_Liberada = true;
  }
 }

void processamento(){  //Centro de controle de diretrizes de ações e movimentos do robô
 if(dist_AP01 > dist_Ref0 && D7 == false){ // D1
  Serial.println("Aceleração confirmada:");
  pwm1 = 175;
```

```
            D1 = true;
            D2 = false;
            D3 = false;
            D4 = false;
            D6 = false;

            orientacao_IN3 = 1;
            orientacao_IN4 = 1;
          }

        if(ativa_APO == true && curva_Possivel == true && curva_EQ == true && eq_Livre == true &&
curva_Liberada == true && D7 == false && memoria_DT == false){    // D2
            Serial.println("Curva esquerda confirmada:");
            pwm1 = 175;
            D1 = false;
            D2 = true;
            D3 = false;
            D4 = true;
            D6 = false;

            orientacao_IN3 = 0;
            orientacao_IN4 = 1;

            memoria_EQ = true;
            memoria_DT = false;
            }
          if(ativa_APO == true && curva_Liberada == true && curva_EQ == true && dist_Dmenor > dist_Dmin
&& dt_Livre == true && D7 == false && memoria_EQ == false){ // D3#
              Serial.println("Curva direita secundaria confirmada:");
              pwm1 = 175;
              D1 = false;
              D2 = false;
              D3 = true;
              D4 = true;
              D6 = false;
```

```
                orientacao_IN4 = 1;
                orientacao_IN4 = 0;
                memoria_DT = true;
                memoria_EQ = false;
                }

            if(ativa_APO == true && curva_Possivel == true && curva_DT == true && dt_Livre == true && curva_Liberada == true && D7 == false && memoria_EQ == false){     // D3
                Serial.println("Curva direita confirmada:");
                pwm1 = 175;
                D1 = false;
                D2 = false;
                D3 = true;
                D4 = true;
                D6 = false;
                orientacao_IN4 = 1;
                orientacao_IN4 = 0;
                memoria_DT = true;
                memoria_EQ = false;
                }
                if(ativa_APO == true && curva_Liberada == true && curva_DT == true && dist_Dmenor > dist_Dmin && eq_Livre == true && D7 == false && memoria_DT == false){ // D2#
                  Serial.println("Curva esquerda secundaria confirmada:");
                  pwm1 = 175;
                  D1 = false;
                  D2 = true;
                  D3 = false;
                  D4 = true;
                  D6 = false;
                orientacao_IN3 = 0;
                orientacao_IN4 = 1;
                memoria_EQ = true;
                memoria_DT = false;
                 }
```

```
        if(dist_AP01 < dist_Ref3 && dist_AP01 > 0 && excedente_Re == false){ // Logica do excedente de ré
PARTE1
          excedente_Re = true;
          }
        else if(excedente_Re == true && dist_AP01 >= dist_Ref4){ // Logica do excedente de ré PARTE2
          excedente_Re = false;
          }
        Serial.print("excedente_Re: ");
        Serial.println(excedente_Re);

        if(dist_AP01 > dist_Ref3 && dist_AP01 < dist_Ref0 && D7 == false && excedente_Re == false){  // D4
          Serial.println("Aceleração reduzida confirmada!");
          pwm1 = 170;
          D4 = true;
          D6 = false;
          D1 = false;
          manobra = false;
          memoria_EQ = false;
          memoria_DT = false;
          }

        if(recuo_Max == false && D7 == false && excedente_Re == true){ // D6
          Serial.println("Confirmação de frenagem ou manobra!");
          pwm1 = 170;
          D6 = true;
          D1 = false;
          D4 = false;
          manobra == true;
          monitora_Traseira = true;
          curva_Liberada = false;
          orientacao_IN3 = !orientacao_IN3;
          orientacao_IN3 = !orientacao_IN3;
          }
        }
```

```
void direcao1(){ // Aceleração++
  if(D1 == true && D7 == false){
    Serial.println("Acelerando!");
    analogWrite(pino_Pwm1,pwm1);
    digitalWrite(in1,0);
    digitalWrite(in2,1);

    analogWrite(pino_Pwm2,pwm2);
    digitalWrite(in3,orientacao_IN3);
    digitalWrite(in4,orientacao_IN4);

    }
  }

void direcao2(){ // Curva a esquerda com aceleração reduzida
  if(D2 == true && D7 == false){
    Serial.println("Efetuando curva a esquerda");
    analogWrite(pino_Pwm1,pwm1);
    digitalWrite(in3,orientacao_IN3);
    digitalWrite(in4,orientacao_IN4);
    }
  }

void direcao3(){ // Curva a direita com aceleração reduzida
  if(D3 == true && D7 == false){
    Serial.println("Efetuando curva a direita");
    analogWrite(pino_Pwm1,pwm1);
    digitalWrite(in3,orientacao_IN3);
    digitalWrite(in4,orientacao_IN4);
    }
  }

void direcao4(){ // Aceleração reduzida em movimento reto
  if(D4 == true && D7 == false){
```

```
  Serial.println("Acelerando em reduzida!");
  analogWrite(pino_Pwm1,pwm1);
  digitalWrite(in1,0);
  digitalWrite(in2,1);
   }
  }

void direcao6(){ // Comando de frenagem e recuo
  if(D6 == true && D7 == false){
    Serial.println("Freiando/recuando!");
    analogWrite(pino_Pwm1,pwm1);
    digitalWrite(in1,1);
    digitalWrite(in2,0);
    }
  }

void direcao7(){ // Parada total
  if(D7 == true){
    Serial.println("Parada!");
    digitalWrite(led2,1);
    analogWrite(pino_Pwm1,pwm1);
    digitalWrite(in1,1);
    digitalWrite(in2,1);
    delay(delay2);
    }
  digitalWrite(led2,0);
  }

//Funções padrão:
void setup() {
  Serial.begin(230400);

  S1.attach(48);
  S2.attach(52);
```

```
  dht.begin();

  pinMode(trigger1,OUTPUT);
  pinMode(trigger2,OUTPUT);
  pinMode(echo1,INPUT);
  pinMode(echo2,INPUT);

  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  pinMode(pino_Pwm1,OUTPUT);
  pinMode(pino_Pwm2,OUTPUT);

  pinMode(MQ2,INPUT);
  pinMode(MQ3,INPUT);
  pinMode(MQ135,INPUT);
  pinMode(Buzzer,OUTPUT);
  pinMode(led,OUTPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(botao,INPUT_PULLUP);

}

void loop() {
//Sensores
  Sensoriamento();

//Funções para movimentação do robô:
  radar_AP01();                 // Radar de ativação por obstaculo dianteiro ou 'AP0'
  radar_AP02();                 // Radar auxiliar AP0 traseiro
  reconhecimento_Zona();        // Reconhecimento de distancias de retas e diagonais referenciais
  tratamento_Dados();           // Função de tratamento das retas e diagonais referências
  processamento();              // Função para o processamento de todos os dados
```

```
direcao1();              // Classe de movimento: aceleração++
direcao2();              // Clase de movimento: curva a esquerda
direcao3();              // Classe de movimento: curva a direita
direcao4();              // Classe de movimento: aceleração reduzida s/ curva
direcao6();              // Classe de movimento: frenagem e aceleração--
direcao7();              // Classe de movimento: parada total
}
```