



Centro Estadual de Educação Tecnológica Paula Souza
Escola técnica Júlio de Mesquita
Curso Técnico em Eletrônica

Guilherme Menezes e Silva
Isis Lopes Delboni
Marcos Junior de Oliveira
Miguel Ribeiro Simon
Rafael Oliveira dos Santos

Robô para monitoramento de lugares inóspitos

Relatório técnico apresentado ao Curso Técnico em Eletrônica da Etec Júlio de Mesquita, orientado pelo professor Egmar Accetto e pela professora Adriana Mariko Yonamine Nakatani, como requisito parcial para obtenção do título de Técnico em Eletrônica.

Santo André – SP
2023
Agradecimentos

- Gostaríamos de expressar nosso profundo agradecimento aos professores Egmar Accetto e Adriana Mariko por sua notável dedicação e apoio durante todo o nosso projeto. Sua orientação e carinho foram fundamentais para o sucesso de nossa empreitada. Suas valiosas contribuições, conhecimento e comprometimento nos inspiraram a alcançar resultados que não teríamos conseguido sem a orientação deles. Estamos verdadeiramente gratos pela oportunidade de aprender com dois profissionais tão exemplares. Muito obrigado!
- Agradecemos à escola estadual Júlio de Mesquita, que esteve sempre de portas e mentes abertas para nos ensinar e nos moldar para os nossos próximos passos. É graças ao apoio e à educação de qualidade que recebemos nesta instituição que estamos preparados para enfrentar os desafios que o futuro nos reserva. Estamos profundamente gratos por fazer parte desta comunidade escolar que valoriza o conhecimento e o desenvolvimento de seus alunos.
- Gratulamos calorosamente aos pais dos membros do grupo, cuja colaboração foi essencial para o sucesso deste trabalho. Seu apoio inabalável, incentivo constante e compreensão ao longo deste projeto foram pilares fundamentais para o nosso progresso. Sabemos que sua dedicação e encorajamento foram igualmente valiosos e apreciados. Obrigado por serem uma parte vital deste percurso e por nos inspirarem a alcançar nossos objetivos.

Resumo

Este trabalho de conclusão de curso (TCC) tem como objetivo o desenvolvimento de um robô de segurança projetado para operar em ambientes inóspitos, tais como locais com vazamentos de gás e espaços confinados. O robô, equipado com quatro rodas e controlado remotamente por meio de dois Arduino, com o auxílio de um dispositivo com visor de câmera ESP32, visa garantir a segurança dos funcionários em situações de risco. A principal inovação deste projeto reside na acessibilidade financeira empresarial e facilidade de uso do robô, tornando-o uma solução viável para empresas de diversos portes. A montagem feita a partir de partes do robô, como se fosse um quebra-cabeça, confere uma certa originalidade ao mesmo. As imagens capturadas pela câmera do robô são transmitidas por meio de um sistema baseado em ESP32 com conectividade Wi-Fi e câmera, permitindo uma monitorização em tempo real das condições do ambiente. Este projeto abordará o desenvolvimento do hardware e software do robô, bem como os resultados de testes e sua aplicabilidade prática na segurança ocupacional. A pesquisa visa contribuir significativamente para a segurança dos trabalhadores em ambientes potencialmente perigosos.

Palavras-chave: Espaços confinados, ambientes Inóspitos, visor de câmera ao vivo, Arduino, ESP32, segurança ocupacional.

Sumário

1. Introdução	4
---------------------	---

2. Desenvolvimento	4
2.1. Objetivo	4
2.2. Justificativa.....	4
2.3. Planejamento	5
2.4. Cronograma de Trabalho	5
2.5. Metodologia	7
2.6. Lista de materiais.....	7
2.7. Características dos principais componentes	10
2.7.1. Módulo Esp32 Cam	10
2.7.2. Sensor de temperatura KY-013	10
2.7.3. Sensor de gás e fumaça MQ-2.....	10
2.7.4. Joystick Arduino 3 Eixos KY-023.....	10
2.7.5. Esp 32	11
2.8. Tabela de preços.....	11
2.9. Funcionamento das principais operações	12
2.9.1. Controle do Robô.....	12
2.9.2. Câmera ao vivo.....	12
2.9.3. Sensores	12
2.10. Aplicativo.....	13
2.11. Digrama de Blocos Geral.....	13
2.12. Programação.....	13
2.13. Diagramas esquemáticos elétricos.....	16
2.14. Projeto Estrutural	19
2.14.1. Visão em 3D.....	19
2.14.2. Desenho em AutoCad 2D.....	21
2.15. Imagens do desenvolvimento.....	22
3. Conclusão	28
4. Referências	29
5. Apêndices	32
A. Programação do transmissor.....	32
B. Programação do Receptor.....	37
C. Programação da Câmera	41

1. Introdução

A proposta desse relatório é um projeto de desenvolvimento de um robô para exploração de ambientes inóspitos. O robô transmite imagens via cabo ou internet, permitindo análise em tempo real. Nosso objetivo é oferecer uma solução versátil e resistente para situações desafiadoras. Além da transmissão de imagem, priorizamos a segurança, durabilidade, autonomia e capacidade de navegação do robô. O projeto tem aplicações em inspeções industriais, monitoramento ambiental e atividades de resgate. Junte-se a nós nessa jornada rumo ao avanço tecnológico em ambientes desafiadores.

2. Desenvolvimento

O desenvolvimento do projeto consistiu em etapas de pesquisa, compra dos materiais, divisão das funções, montagem do aparelho, lógica de programação, montagem final com concerto de todas as outras etapas e decoração de algumas partes do monitorador.

2.1. Objetivo

O objetivo deste projeto é desenvolver um robô para explorar ambientes inóspitos e transmitir imagens em tempo real. Buscamos criar uma solução versátil e robusta, com foco na segurança, durabilidade, autonomia e capacidade de navegação do robô. Pretendemos aplicar o robô em inspeções industriais, monitoramento ambiental e atividades de resgate, proporcionando uma ferramenta valiosa para equipes de pesquisa e profissionais de segurança.

2.2. Justificativa

O trabalho vem de uma demanda cada vez mais recorrente em toda sociedade.

Em um mundo onde a necessidade de inspeção e monitoramento de ambientes se torna cada vez maior, a proposta de um robô para monitoramento de ambientes inóspitos torna-se necessária.

Atualmente há vários ambientes em que a entrada de seres humanos é arriscada e em algumas vezes impossível, sabendo desse fato, o aparelho, que adentra nesses meios, passa todas as informações necessárias sobre ele e ao mesmo tempo impede que um ser humano arrisque sua vida é um salvador essencial para qualquer empresa que tenha como demanda o monitoramento de locais.

O projeto consiste em um robô de 30cm controlado a distância por um operador. Nele há a presença de câmera, sensores de ambiente e praticidade para locomoção.

Além disso os custos para existência não são fora do eixo, em razão dele utilizar de componentes e aparelhos básicos, como Arduino, Esp32Cam, motores DC de 3 a 6 volts, protoboard etc.

2.3. Planejamento

A primeira parte consistiu em pesquisas pela busca do funcionamento da locomoção. Tínhamos dúvidas de como faríamos a movimentação, então após buscas no meio digital, decidíamos que 4 motores DC viriam a ser os melhores, pois traziam a velocidade e a praticidade necessária.

Além dessa parte, muitas foram as buscas para encontrar um meio de transmissão de informações. Um dos principais pontos do carrinho era sua capacidade de ser um sensor de ambientes e repassador das características desses mesmos por meio de câmera, com isso, encontrar um transmissor era tarefa essencial. Os escolhidos para tais tarefas foram: Modulo ESP32 cam; Esp 32; Sensor de Temperatura; Sensor de Gás e fumaça; Sensor de Umidade.

Como base, após muitas dúvidas entre cano pvc, Madeira mdf e plástico da impressão 3d, foi decidido que a terceira opção era a mais viável por sua praticidade e custo.

2.4. Cronograma de Trabalho

As funções e datas que compõem o presente trabalho foram divididas da seguinte maneira:

Abril		
13	20	27

Maio			
4	11	18	25

Junho				
1	8	15	22	29

Julho			
6	13	20	27

Green	Grey	Green	Blue	Grey
Green	Green	Green	Blue	Grey
Green	Green	Green	Blue	Grey
Green	Grey	Green	Blue	Grey
Green	Green	Green	Blue	Grey

Orange	Orange	Orange	Orange
Green	Green	Orange	Orange
Orange	Orange	Orange	Orange
Orange	Orange	Orange	Orange
Orange	Orange	Orange	Orange

Agosto				
3	10	17	24	31
Yellow	Green	Grey	Blue	Green
Orange	Orange	Grey	Blue	Green
Yellow	Green	Grey	Blue	Green
Yellow	Green	Grey	Blue	Green
Yellow	Green	Grey	Blue	Green

Setembro			
7	14	21	28
Blue	Grey	Green	Purple
Green	Green	Green	Purple
Green	Green	Green	Purple
Blue	Grey	Green	Purple
Green	Green	Green	Purple

Outubro			
5	12	19	26
Green	Purple	Grey	Cyan
Green	Purple	Grey	Cyan
Green	Purple	Grey	Cyan
Green	Purple	Grey	Cyan
Green	Purple	Grey	Cyan

Novembro				
2	9	16	23	30
Green	Green	Red	Black	
Green	Green	Red	Black	
Green	Green	Red	Black	
Green	Green	Red	Black	
Green	Green	Red	Black	

Legendas e funções:

Data:	
Integrantes	Programação prevista
Guilherme	Programação, Teste e Montagem
Isis	Relatório e pesquisa
Marcos	Slides, Pesquisa, montage,
Miguel	Programação, Teste e Montagem
Rafael	Relatório, Pesquisa e programação
Cor	Legenda
Yellow	Começo de um trabalho

	Entrega de trabalho com nota
	Entrega de trabalho entre o grupo
	Tempo de Realização
	Correção e verificação
	Reunião entre o grupo
	Apresentação de trabalho
	Relação pessoal / Férias
	Finalização

2.5. Metodologia

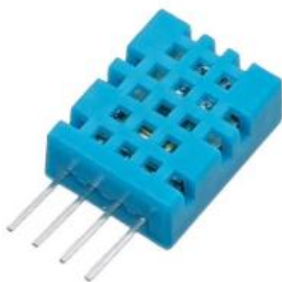
Com a concepção consolidada e todos os componentes devidamente adquiridos, nossa equipe empenhou-se em assegurar que cada etapa se desenrolasse de acordo com o planejado. Nesse intervalo, implementamos uma série de modificações que resultaram na otimização do projeto, tanto no que tange ao desempenho dos circuitos quanto à melhoria da parte externa. Em decorrência disso, foram necessárias novas aquisições de componentes, ao passo que outros tornaram-se obsoletos.

2.6. Lista de materiais

- Módulo Esp32 Cam



- Sensor de temperatura e umidade DHT11



- Sensor de gás e fumaça MQ-2



- 4 motores DC e 4 rodas



- Joystick Arduino 3 Eixos KY-023



- Bateria 9V GP



- Cabo rígido branco 22 AWG (0,30mm) por metro



- Esp 32



2.7. Características dos principais componentes

Foram utilizados uma ampla variedade de componentes, desde os mais elementares até os mais singulares. A seguir, apresentamos aqueles que desempenharam um papel fundamental na estrutura da estufa

2.7.1. Módulo Esp32 Cam

O módulo ESP32-CAM é um dispositivo que combina um microcontrolador ESP32 com uma câmera. Ele se comunica com outros dispositivos por meio de WiFi e pode capturar, processar e transmitir imagens e vídeo. É amplamente utilizado em projetos de IoT, segurança e automação.

2.7.2. Sensor de temperatura KY-013

O sensor de temperatura KY-013 é um termistor NTC que varia sua resistência com a temperatura ambiente. Ele se conecta a um microcontrolador, como o Arduino, e fornece uma leitura analógica da temperatura. Essa leitura pode ser convertida em graus Celsius ou Fahrenheit para monitorar as mudanças de temperatura.

2.7.3. Sensor de gás e fumaça MQ-2

O sensor de gás e fumaça MQ-2 detecta gases inflamáveis, como metano e álcool, além de fumaça no ar. Ele utiliza um elemento de detecção de gás que varia sua resistência com a presença desses gases. Quando a resistência muda, o sensor converte isso em um sinal elétrico, indicando a presença e concentração do gás ou fumaça.

2.7.4. Joystick Arduino 3 Eixos KY-023

O Joystick Arduino 3 Eixos KY-023 é um dispositivo de entrada que fornece informações sobre a posição em três dimensões: X, Y e Z. Ele contém potenciômetros que variam sua resistência conforme o movimento do joystick, gerando sinais analógicos que podem ser lidos por um microcontrolador Arduino. Esses valores são usados para controlar motores, servos ou outros dispositivos com base na posição do joystick.

2.7.5. Esp 32

O Esp 32 um microcontrolador desenvolvido pela empresa chinesa Espressif Systems. Ele é baseado no processador Xtensa Dual-Core de 32 bits e pode ser usado em diversos projetos de IoT (Internet das Coisas), robótica, automação residencial e outros projetos que envolvem conexão com a internet.

2.8. Tabela de preços

Materiais	Quantidade	Preço R\$	Frete	Loja
Transmissor e receptor RF	2	7,15		Eletro Gate
Modulo Esp cam	1	47,30		Eletro Gate
Sensor de temperatura e umidade DHT11	1	11,30		Eletro Gate
LEDs difusos/SMD de alto brilho	1	0,99		Eletro Gate
Sensor de gás e fumaça MQ-2	1	18,90		Eletro Gate
Kit motores DC e 4 rodas	4	71,41		Eletro Gate
Joystick Arduino 3 Eixos KY-023	2	7,00		Eletro Gate
Bateria de 9v Panasonic 6LFF22	1	21,21		Eletro Gate
Bateria de 9v DG 6LFF22	1	20,63		Eletro Gate
Percloroeto de ferro 500mL	1	17,90		Eletro Gate
Placa de fenolite virgem Dupla face 8x12cm	2	7,24		Eletro Gate
Motor de passo + modulo de controle (driver ULN2003)	2	17,96		Eletro Gate
Soquete 28 pinos para Atmega328p-PU	1	1,20		Eletro Gate
Soquete 20 pinos	2	1,50		Eletro Gate
Cabo rígido branco 22 AWG (0,30mm) por metro	4	1,48		Eletro Gate

Esp 32 wifi + bluetooth	2	69,00		Eletro Gate
Total	26	R\$436,46	1	Eletro Gate

2.9. Funcionamento das principais operações

2.9.1. Controle do Robô

O controle do projeto envolve a movimentação do robô de maneira mais coloquial, operando através do Esp 32. Este microcontrolador, que se destaca por sua simplicidade e baixo custo, apresenta funcionalidades como comunicação Wi-Fi, um sistema equipado com processador Dual Core, Bluetooth híbrido e vários sensores integrados.

Para a utilização no controle do projeto, foi empregado dois Esp 32, sendo um deles responsável pela conexão com os motores DC, incumbidos da função de propulsão do veículo. O segundo Esp 32 é dedicado aos controles dos joysticks, desempenhando a tarefa crucial de comandar a movimentação do robô. Todo esse intercâmbio de informações entre os componentes é efetuado por meio de uma comunicação privada, intrínseca ao próprio Esp.

2.9.2. Câmera ao vivo

A câmera, parte fundamental e indispensável do projeto, opera por meio de uma Esp Cam, gravando todo o ambiente, independentemente de estar claro ou escuro. No projeto, posicionada na parte superior do robô, a câmera realiza movimentos rotatórios por meio de um servo motor, proporcionando ao controlador controle total sobre o que ocorre no ambiente. O vídeo resultante da exploração é transmitido em tempo real via Wi-Fi para o aplicativo do projeto em qualidade aceitável, possibilitando a visualização completa do ambiente explorado.

2.9.3. Sensores

Os sensores utilizados compreendem um Sensor de Temperatura e Umidade DHT11, juntamente com um Sensor de Gás e Fumaça MQ-2. Ambos desempenham a função crucial de detectar e analisar as características do ambiente, transmitindo essas informações por meio do aplicativo exclusivo desenvolvido para o projeto. Os dados resultantes são apresentados de forma clara na parte inferior da tela

O Sensor de Gás e Fumaça MQ-2 opera com base em sua sensibilidade a diversos tipos de gases inflamáveis, como GPL, butano, propano, metano, álcool e hidrogênio.

A resistência do sensor varia conforme o tipo de gás, sendo que um potenciômetro integrado possibilita ajustar o limiar de saída digital do sensor (D0). Esse limite determina o valor acima do qual o pino digital emitirá um sinal "HIGH". A voltagem gerada pelo sensor é sensível ao nível de fumaça ou gás na atmosfera. Essa voltagem é proporcional à concentração de fumaça ou gás, indicando que, à medida que a concentração aumenta, a tensão de saída também aumenta, e vice-versa. A saída

pode ser em formato analógico (A0), lida por uma entrada analógica do Arduino, ou em formato digital (D0), lida por uma entrada digital do Arduino.

Sensor de Temperatura e Umidade DHT11 utiliza um termistor para medir a temperatura e um sensor capacitivo para medir a umidade do ambiente

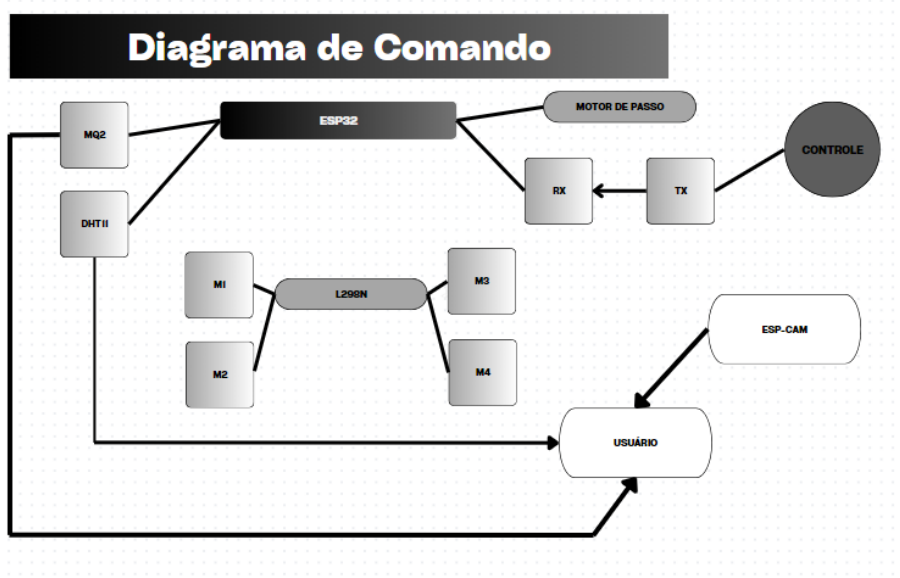
2.10. Aplicativo

O aplicativo atua como um centro de informações, reunindo todos os dados provenientes dos ambientes percorridos pelo robô. Este recebe as informações e as exibe de maneira clara na tela, que possui uma interface organizada e concisa.

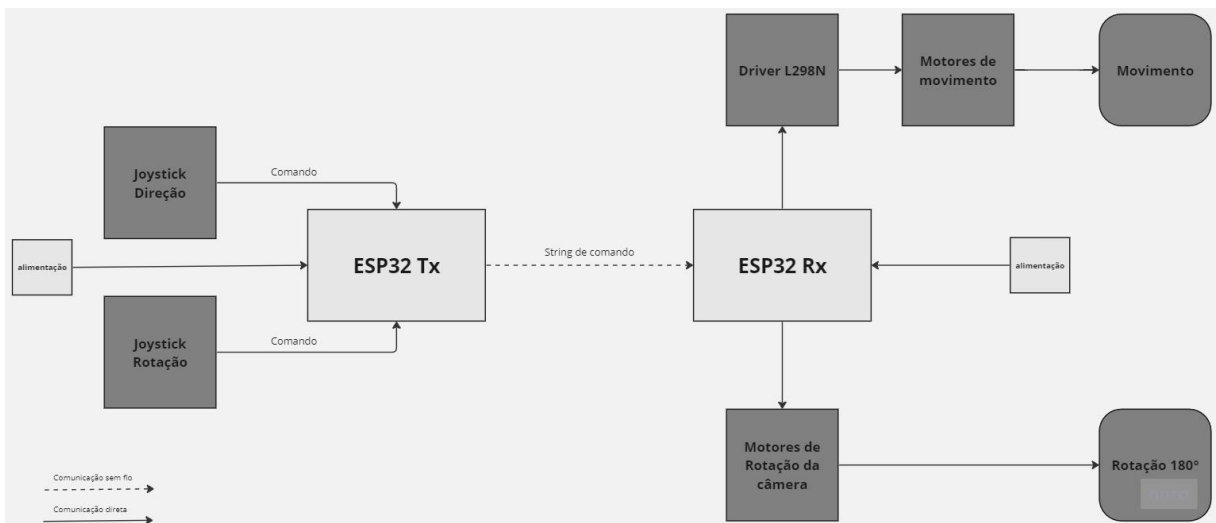
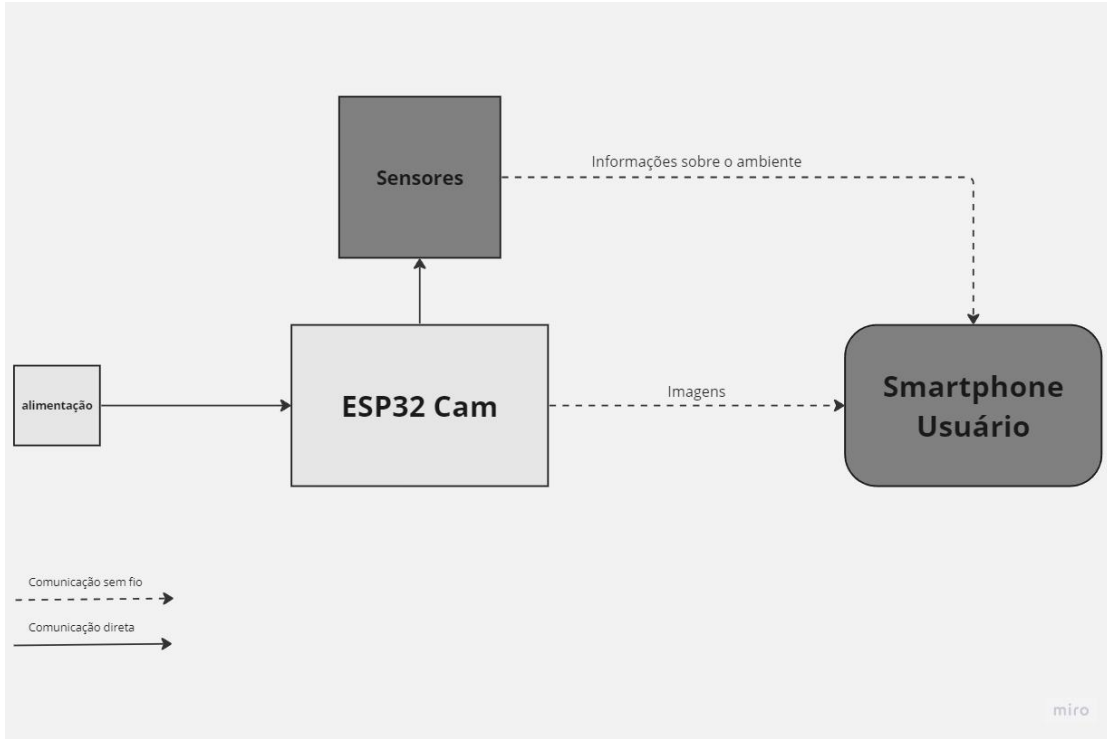
O aplicativo foi feito a partir de um programa (Appgyver) que possibilita a criação de aplicativos diversos para os mais variados tipos de demanda a partir de interações mais simplificadas e sem a utilização de códigos complexos e desgastantes.

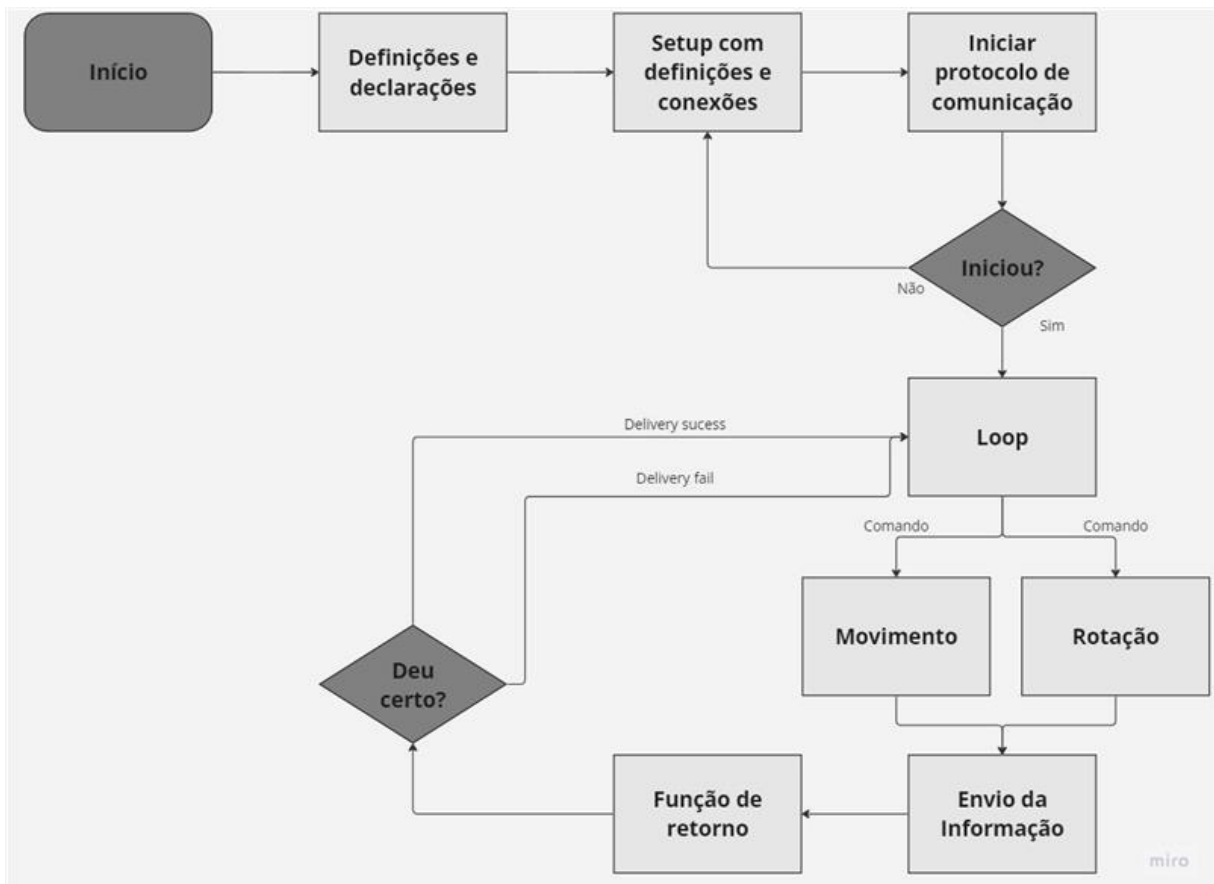
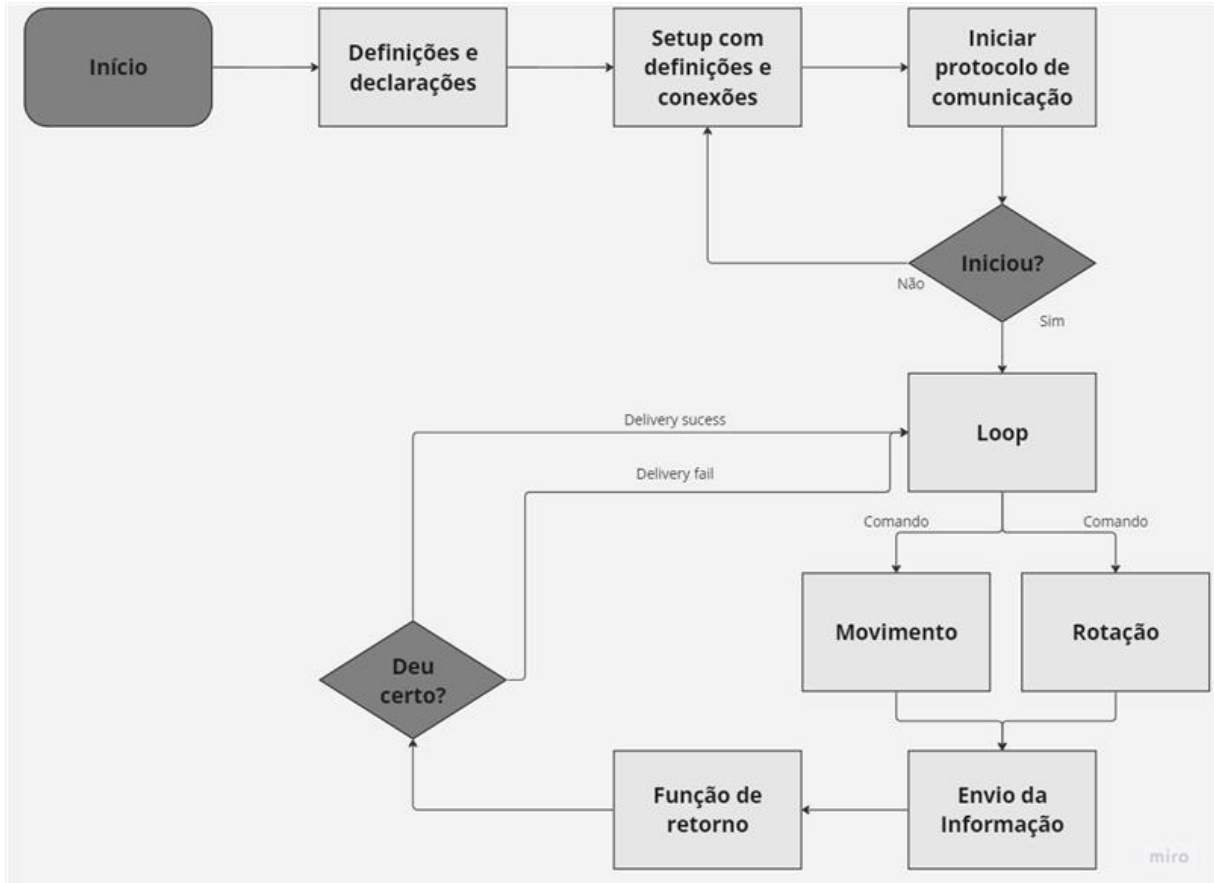
Isso serviu como um facilitador, pois economizou boa parte do nosso tempo e proporcionou uma maior troca de operações

2.11. Diagrama de Blocos Geral

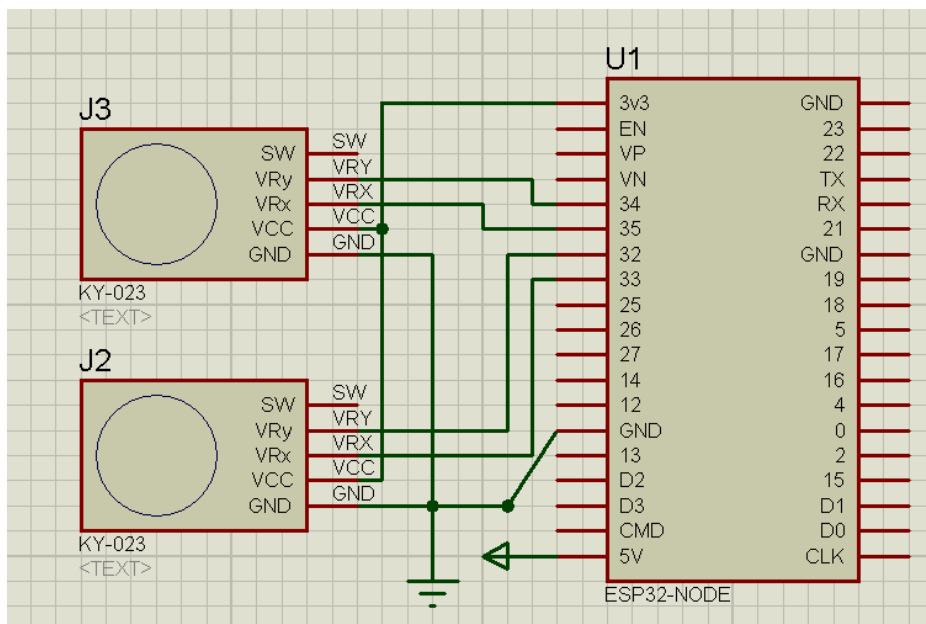
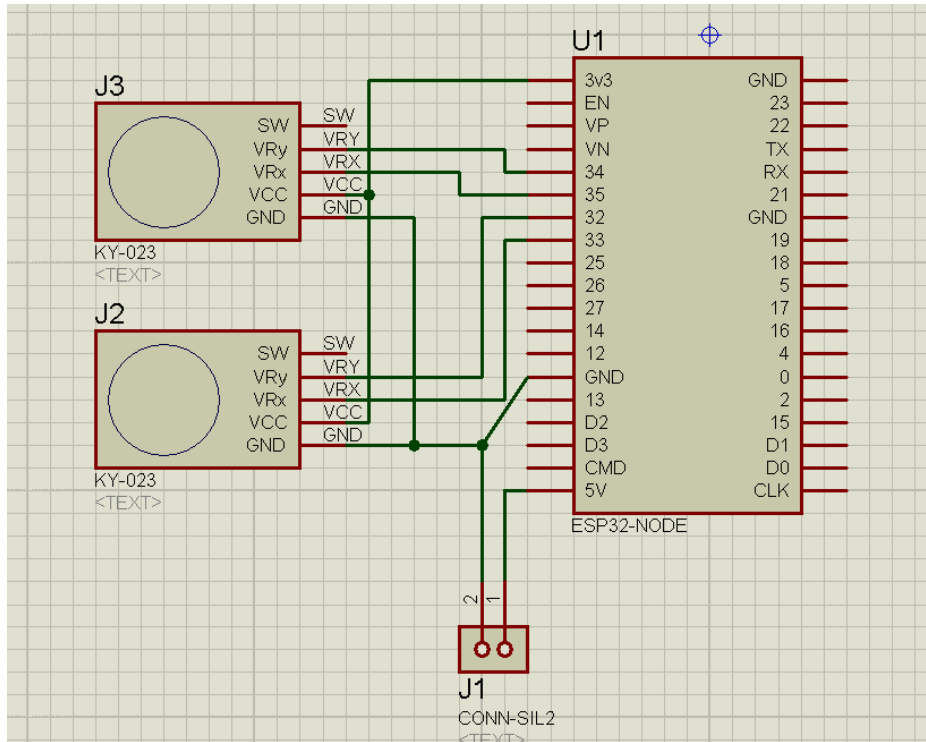


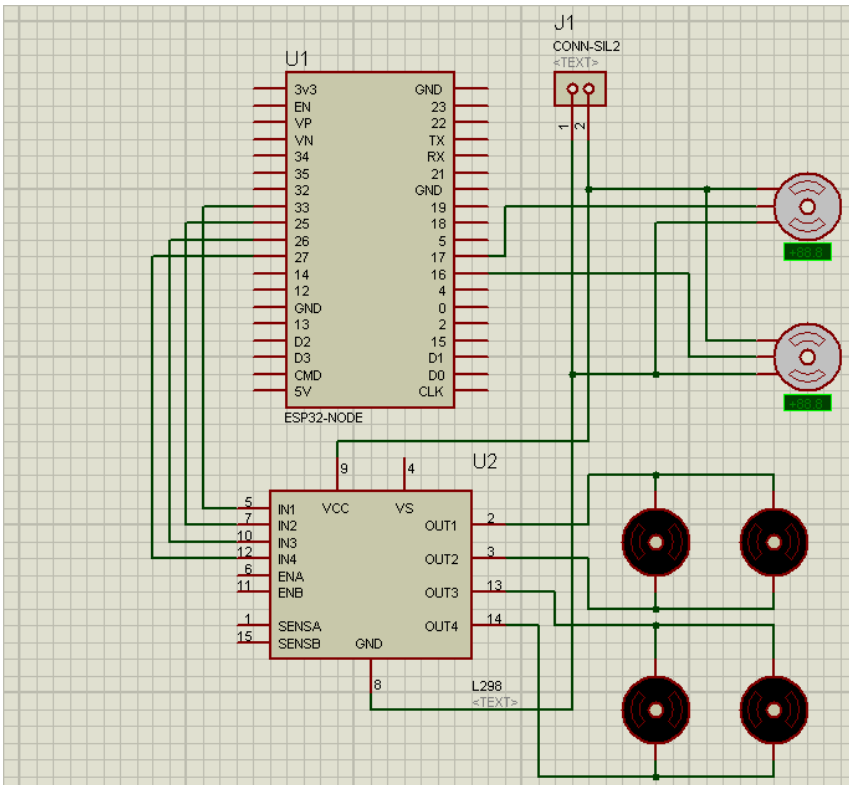
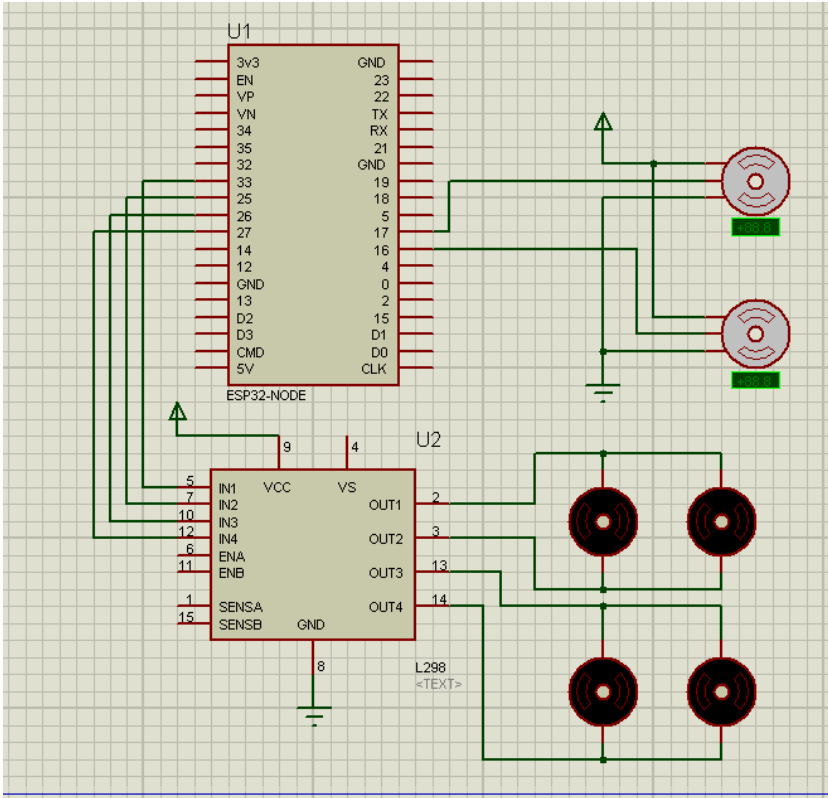
2.12. Programação

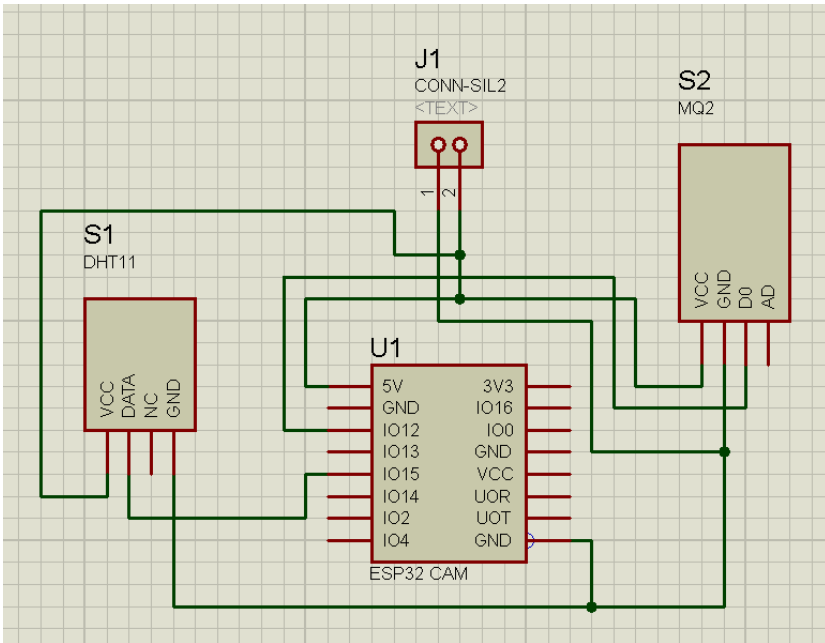
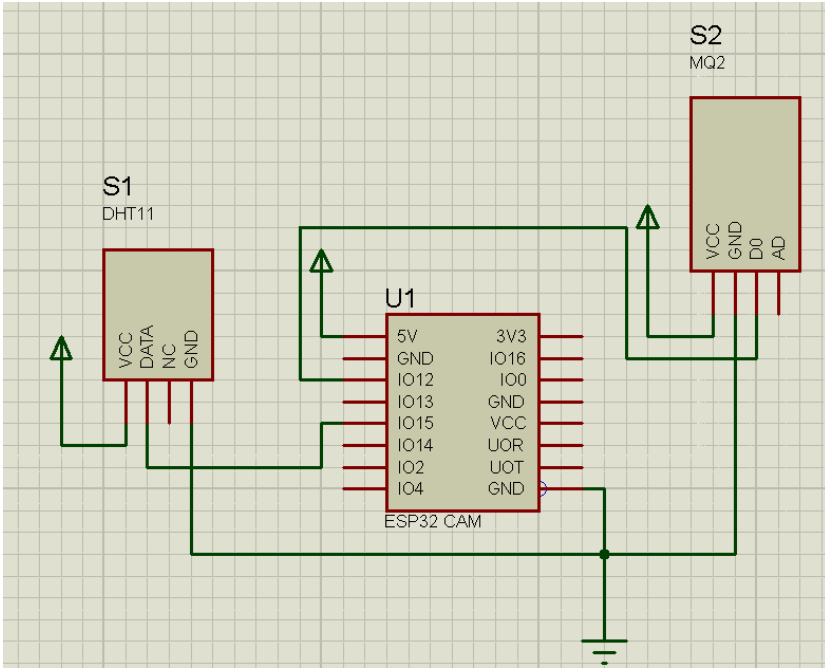




2.13. Diagramas esquemáticos eléctricos







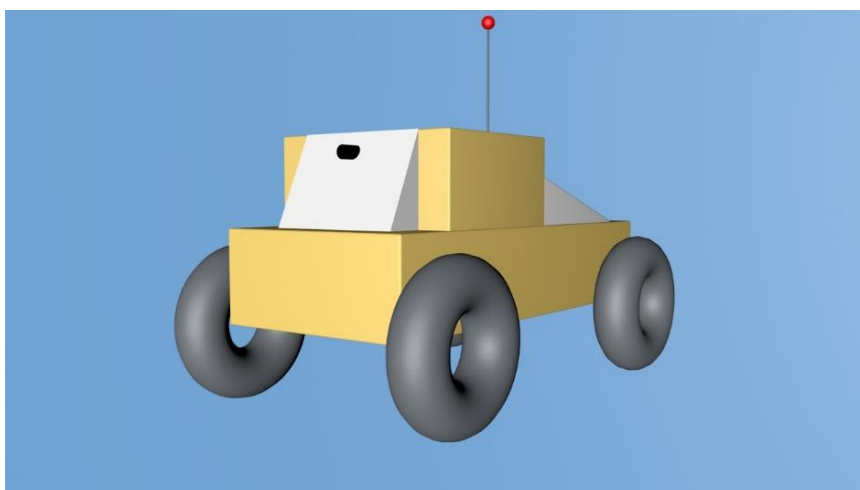
2.14. Projeto Estrutural

A concepção inicial para a estrutura do projeto era a de criar um desenho detalhado utilizando o software AutoCAD. Entretanto, devido à evolução do projeto e a ocorrência de imprevistos não antecipados, não foi viável concluir a estrutura de acordo com o planejado. Como alternativa, optamos por construir uma estrutura provisória utilizando uma placa de papelão, a qual desempenha o papel de um chassi funcional para o projeto.

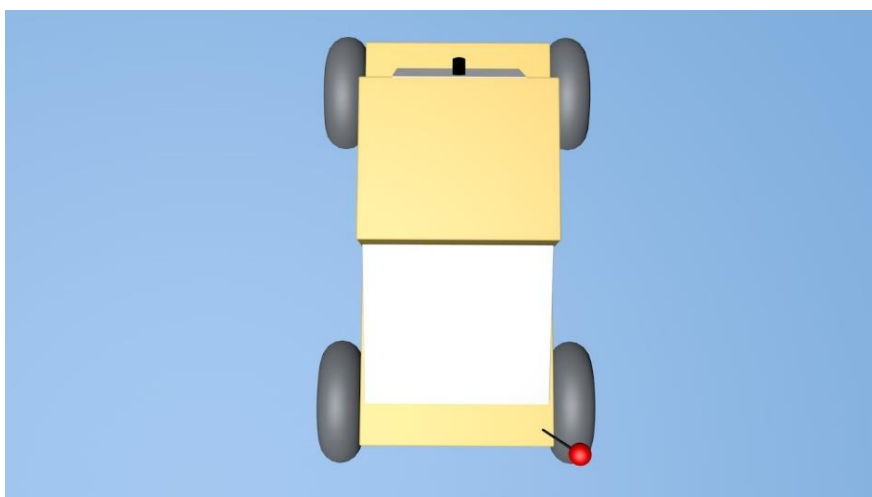
O papelão foi escolhido por conta de toda sua versatilidade e facilidade de manipulação. Além disso é pertinente destacar que ele possibilita um maior reaproveitamento futuro, em razão do material ser renovável e não causar danos ao meio ambiente.

2.14.1. Visão em 3D

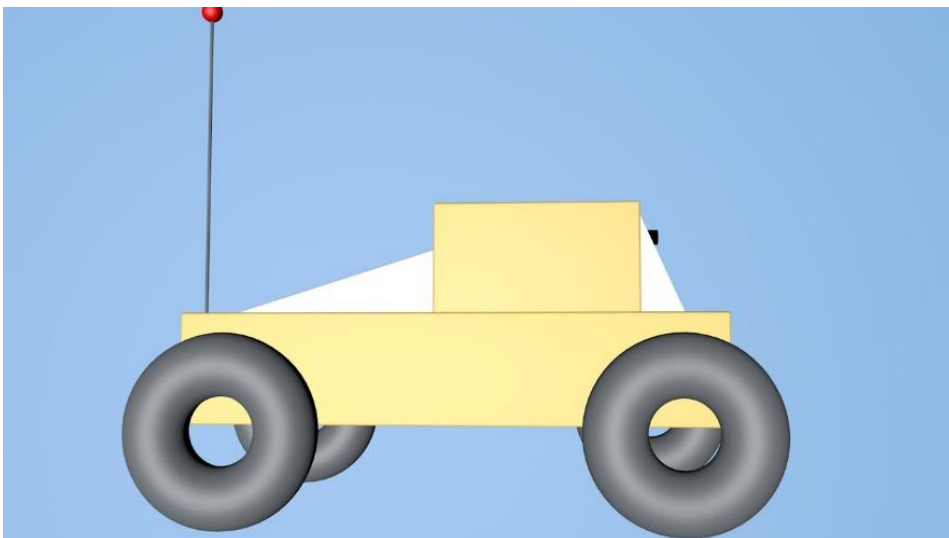
Visão frontal lateral



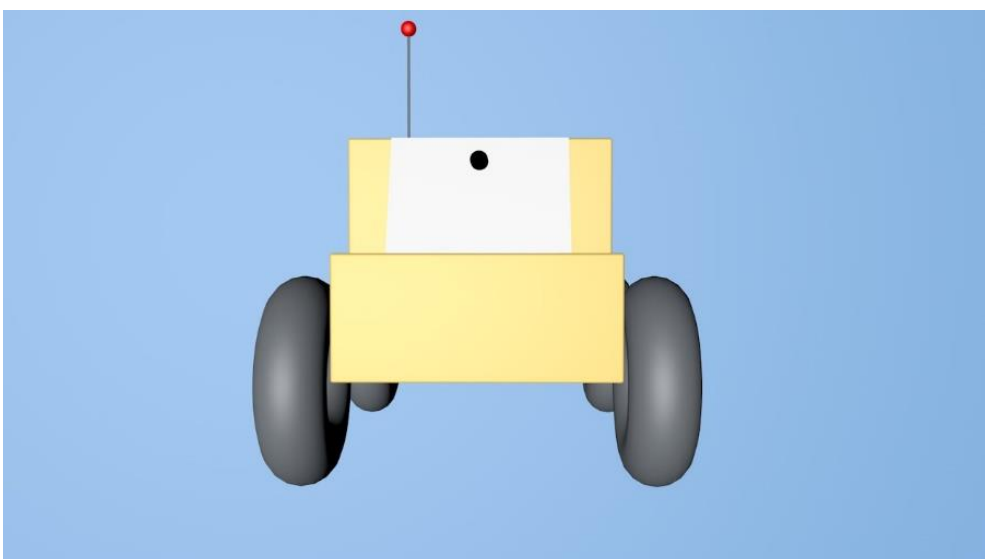
Visão superior



Visão lateral

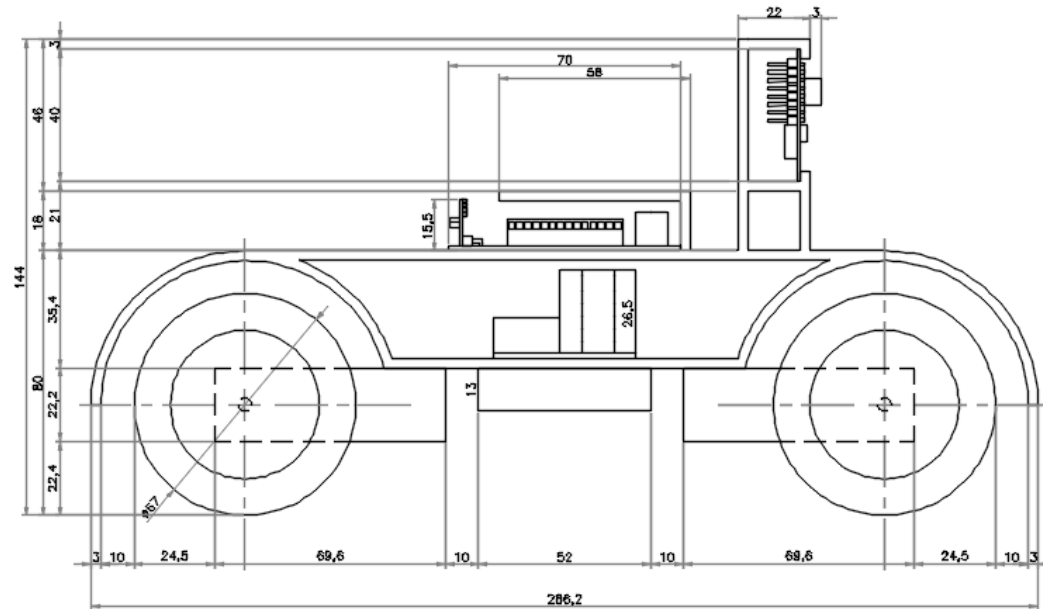


Visão frontal

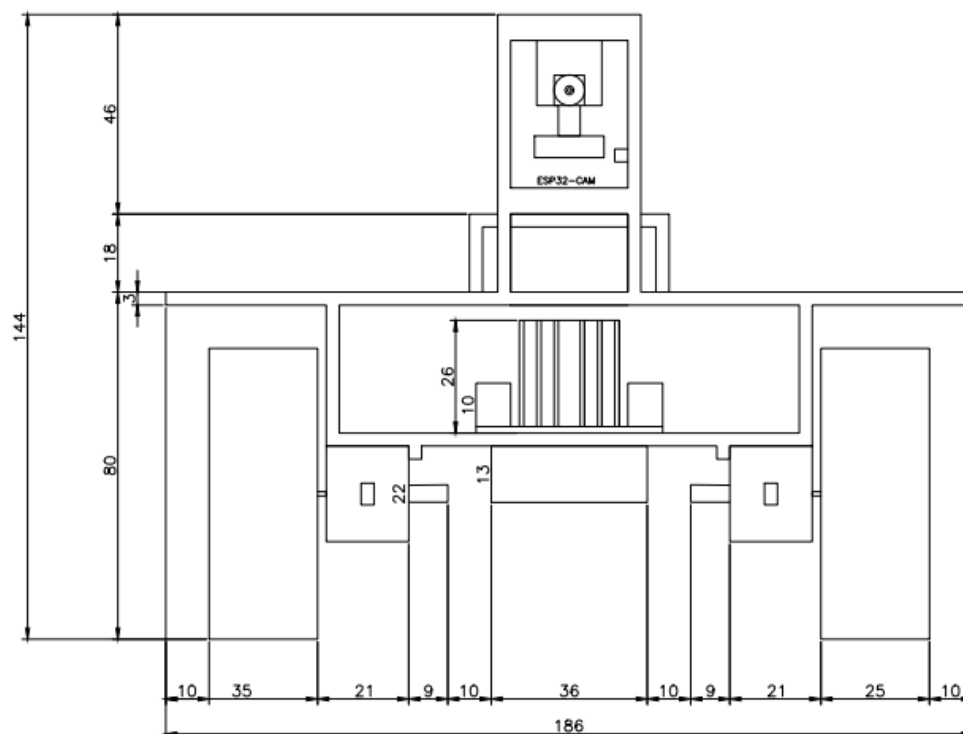


2.14.2. Desenho em AutoCad 2D

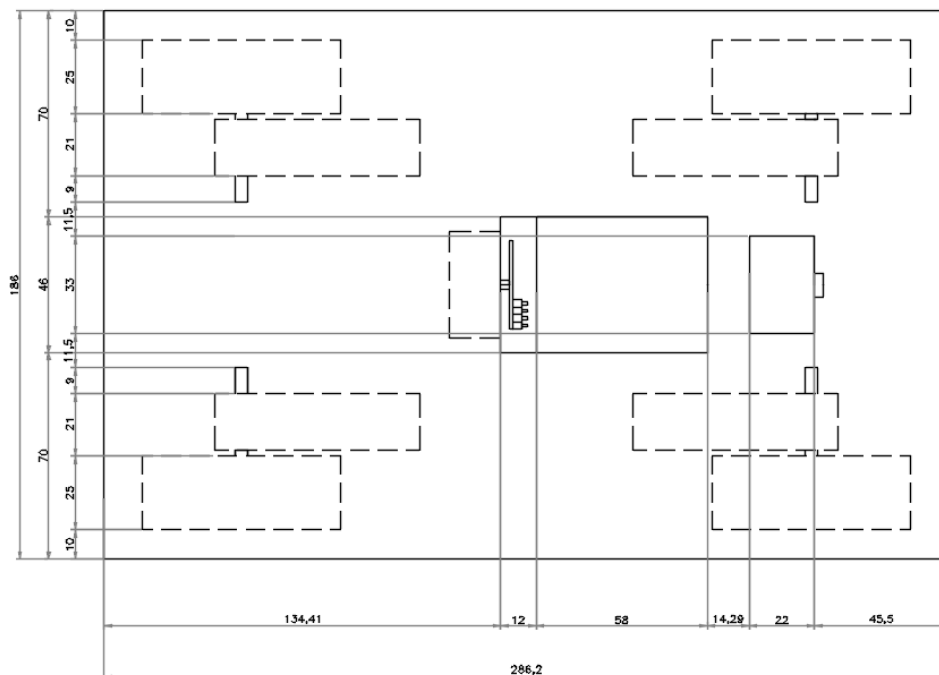
Visão Lateral



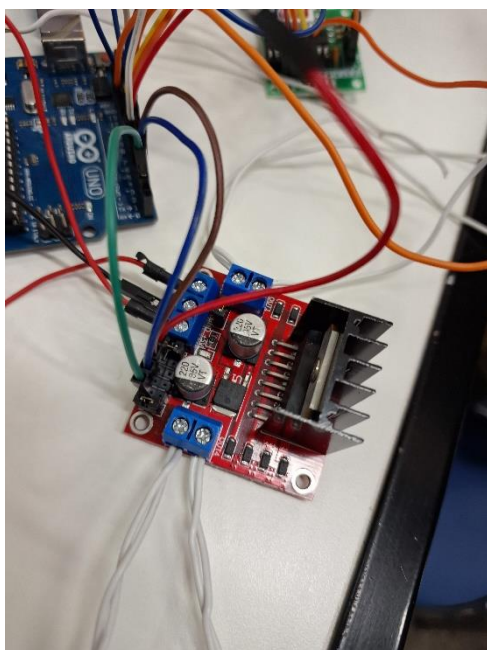
Visão Frontal



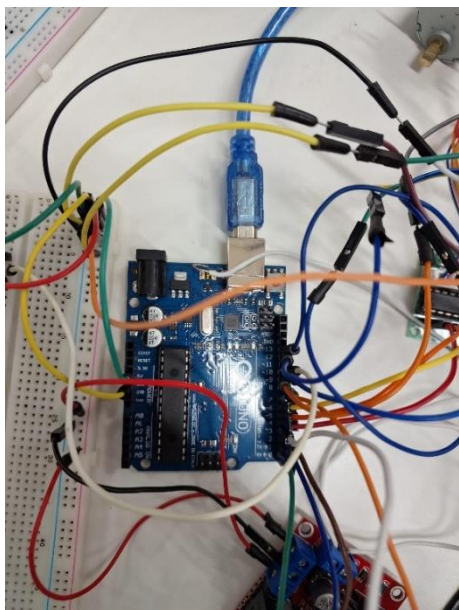
Visão superior



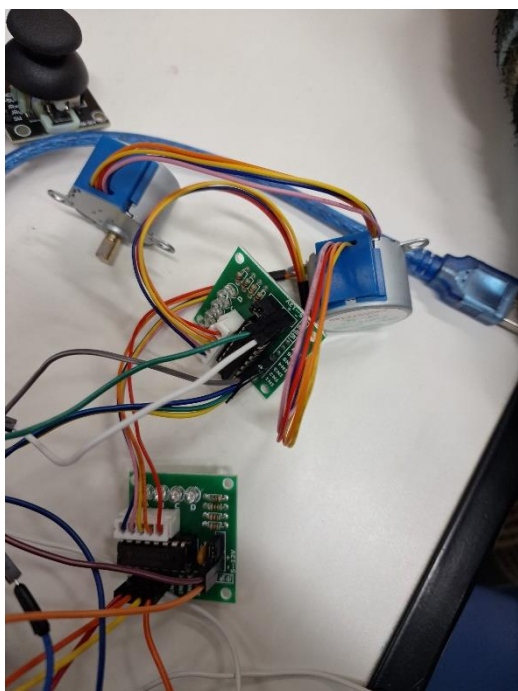
2.15. Imagens do desenvolvimento



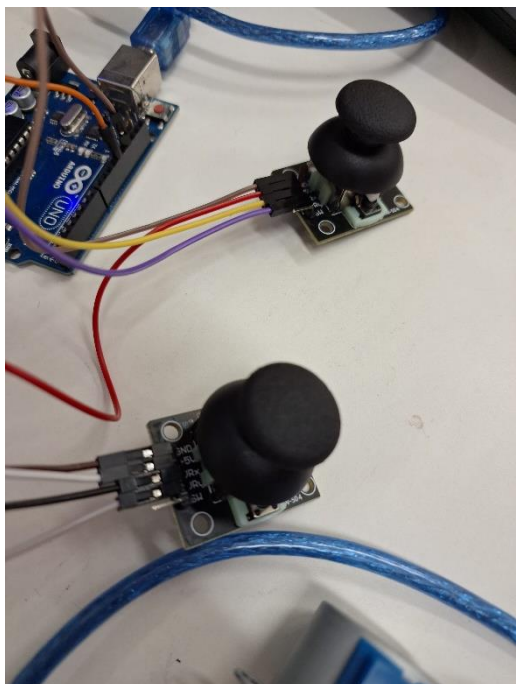
Teste com o RF



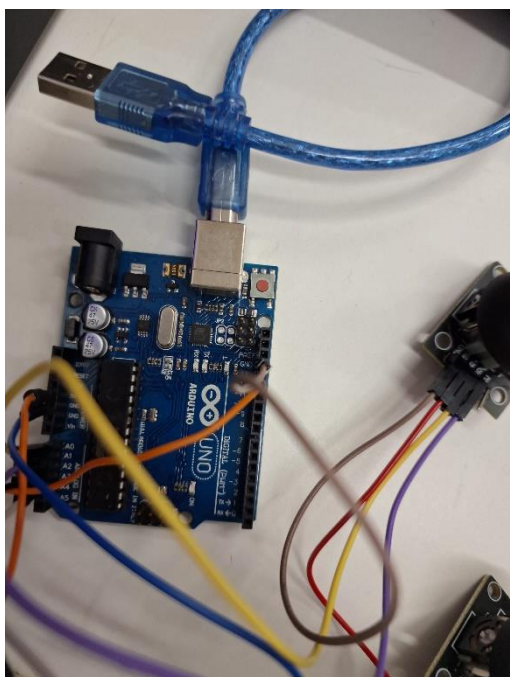
Testes no arduino



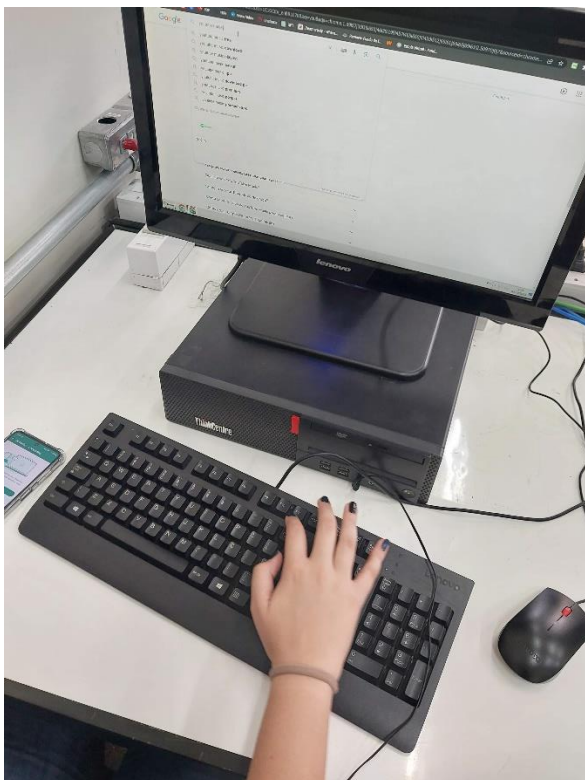
Teste com motores



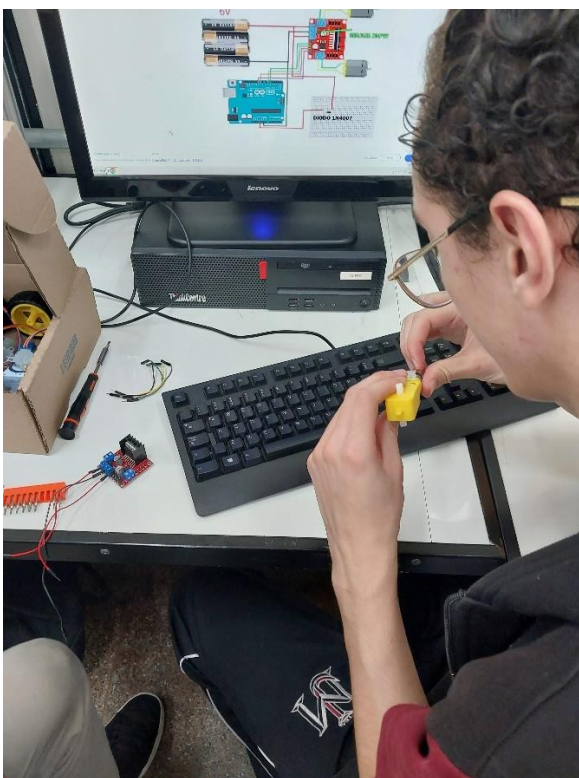
Teste com os joysticks



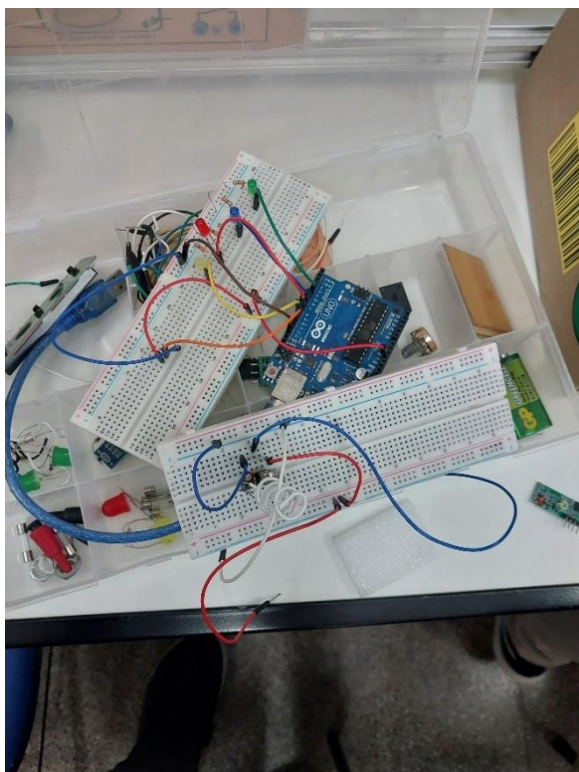
Botando no Arduino os principais componentes



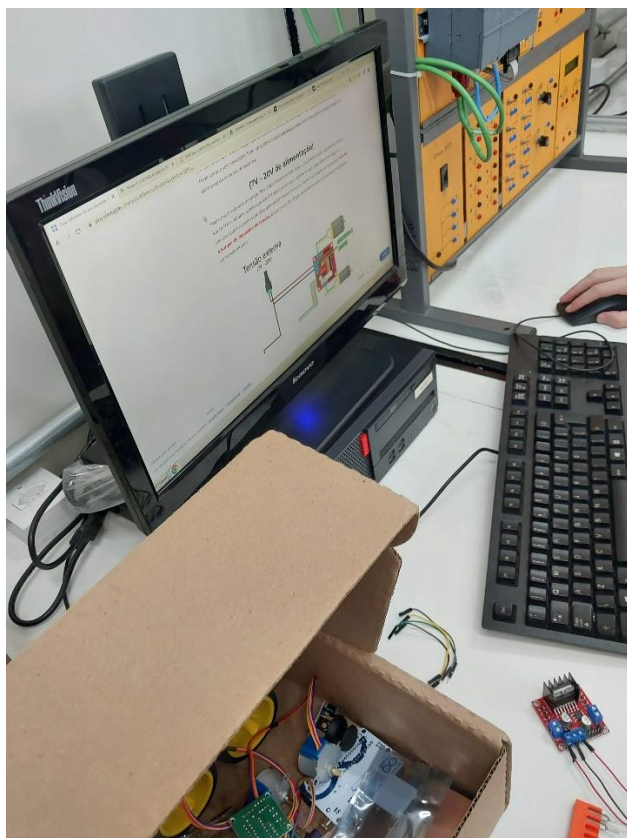
Pesquisa sobre comunicação



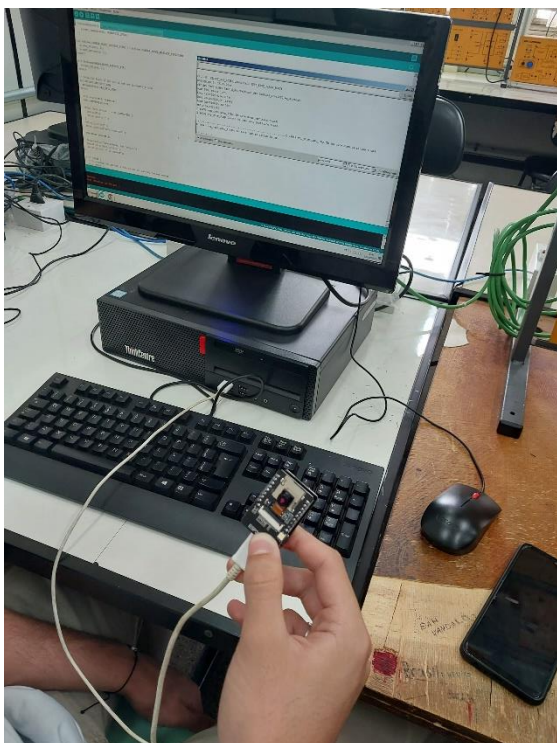
Teste com os motores



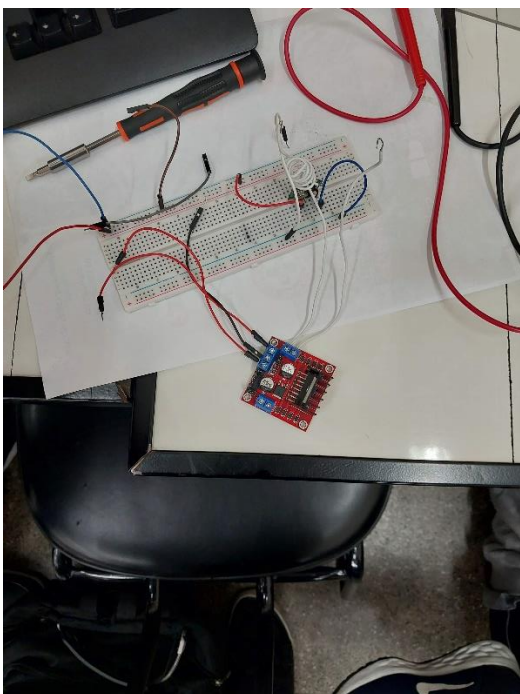
Teste de leds



Pesquisa sobre o Módulo RF



Programação do ESPcam no Arduino



Teste com RF

3. Conclusão

Em suma, o projeto do robô para exploração de ambientes inóspitos e transmissão de imagens em tempo real busca fornecer uma solução versátil e confiável para desafios enfrentados nessas condições. Com seu potencial de melhorar a segurança e eficiência em inspeções industriais, monitoramento ambiental e atividades de resgate, espera-se que esse projeto impulse avanços na robótica. Ao inspirar descobertas e colaboração, buscamos impulsionar o progresso e alcançar resultados significativos nessa área de exploração de ambientes desafiadores.

4. Referências

Imagem espcam: <https://loja.fabricadebolso.com.br/produto/esp32-cam-mb-wifi-modulo-de-camera-bluetooth-placa.html>

Imagem sensor de umidade e temperatura DHT11:
https://www.google.com/url?url=https://www.baudaeletronica.com.br/produto/sensor-de-umidade-e-temperatura-dht11.html%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKewjl-prlwsyCAxUQBrkGHTfDBtMQ1SkIggcoAA&usq=AOvVaw3htZ5dklxxCDCrHJSDk9AZ

Imagem Led difusos/SMD de alto brilho:
https://www.google.com/url?url=https://shopee.com.br/product/322044496/4870333318&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwib5PjgwsyCAxX1D7kGHZyDCp4Q1SkInggoAA&usq=AOvVaw3pww0poYqj9_kKwMoyzx0h

Imagem Sensor de gás MQ-2 Gás inflamável:
https://www.google.com/url?url=https://loja.fabricadebolso.com.br/produto/sensor-de-gas-mq-2-gas-inflamavel-e-fumaca%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwirm7WDw8yCAxWbH7kGHVwwD6IQ2SkIwAc&usq=AOvVaw37v6bXPS5nO7HCi68ERuBD

Imagem 4 motores DC e 4 rodas:
https://www.google.com/url?url=https://produto.mercadolivre.com.br/MLB-1352382973-kit-4-rodas-4-motores-dc-3-a-6v-com-reduco-robotica-JM%3Fmatt_tool%3D18956390%26utm_source%3Dgoogle_shopping%26utm_medium%3Dorganic&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwisxd3fw8yCAxWsOrkGHcKpBqQQ1SkIkwcoAA&usq=AOvVaw033w67Hu6rXyFTci2wil0W

Imagem Joystick Arduino 3 Eixos KY-023:
https://www.google.com/url?url=https://www.baudaeletronica.com.br/produto/joystick-arduino-3-eixos-ky-023.html%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwiSgpnw8yCAxVaBbkGHbYcDJwQ1SkI5AYoAA&usq=AOvVaw3iw76JJF55E-lzpObL0Bx7

Imagem Bateria de 9V Panasonic 6LFF22:
https://www.google.com/shopping/product/2094176356659606703?q=%E2%80%A2+Bateria+de+9V+Panasonic+6LFF22&sca_esv=583532019&sxsrf=AM9HkKkTrNzbKFLjta4-OFr9RPsUo7cCPg:1700275394555&biw=1049&bih=873&prds=eto:13506330415874146578_0,pid:4481200072870322102,rsk:PC_9900708419050345588&sa=X&ved=0ahUKEwirv_39w8yCAxXOO7kGHVh-BKMQ8gll4QcoAA

Imagem Bateria 9V GP

https://www.google.com/shopping/product/6853165559633504989?q=%E2%80%A2+Bateria+9V+GP&bih=873&biw=1049&hl=pt-BR&prds=eto:761701264761898456_0,pid:1295547382334475660,rsk:PC_11754581359994916181&sa=X&ved=0ahUKEwiNs_HxMyCAxWvLbkGHWtLDJlQ8gllrQgoAA

Imagem Percloroeto de Ferro 500mL:

https://www.google.com/url?url=https://www.baudaeletronica.com.br/produto/percloro-to-de-ferro-500ml.html%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEWjdyZ2UxMyCAxWSLbkGHdu-BmkQ1SkIraYoAA&usg=AOvVaw3l6k-08XXi5VUOgOsKmMzS

Imagem Placa de fenolite virgem Dupla face 8x12cm:

https://www.google.com/url?url=https://www.baudaeletronica.com.br/produto/placa-de-fenolite-virgem-dupla-face-8x12-cm.html%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEWj-k7GpxMyCAxVKlbkGHbigAQwQ1SkIxAyoAA&usg=AOvVaw0oDZzuMa6sKHPmoyy5tqiu

Imagem Soquete 28 pinos para Atmega328p-PU:

https://www.google.com/url?url=https://www.eletrorate.com/soquete-28-pinos-para-atmega328p-pu%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEWjCkLu0xMyCAxVjA7kGHfpjBwlQ1SkIqqYoAA&usg=AOvVaw1boaMyVIXO_xtFCIA4JSS5

imagem Cabo rígido branco 22 AWG (0,30mm) por metro:

<https://www.google.com/url?url=https://www.saravati.com.br/cabo-flexivel-22-awg-0-30mm-branco-vendido-p-metro.html&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwiXpu-xMyCAxWhLLkGHfC3AKIQ1SkIIQYoAA&usg=AOvVaw09o82ZpqczwB3JGTaqnMEw>

imagem soquete 20 pinos :

<https://www.google.com/url?url=https://www.saravati.com.br/cabo-flexivel-22-awg-0-30mm-branco-vendido-p-metro.html&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwiXpu-xMyCAxWhLLkGHfC3AKIQ1SkIIQYoAA&usg=AOvVaw09o82ZpqczwB3JGTaqnMEw>

iamgem Esp 32 : https://encrypted-tbn2.gstatic.com/shopping?q=tbn:ANd9GcR2UvIXN4B8-yCUSas_VvYr6X68JU6EQF1iU6xcnFiCPJTtjO38KY--Jz--Q2rcwveGOYFjqlJQlliEeSyHEVr7G60z5kd19MzC_nJq2peQvamFmhwrQ7oA&usqp=CAE

imagem Motor de passo + modulo de controle (driver ULN2003)

<https://www.google.com/url?url=https://www.eletrorate.com/motor-de-passo-modulo-de-controle-driver->

[uln2003%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwiyxlz9xMyCAxUspJUCHZEYDgQQ1SkIQYoAA&usq=AOvVaw1NnDQfPqwBBTdeZgeHNIiz](https://www.google.com/search?q=uln2003%3Futm_source%3DSite%26utm_medium%3DGoogleMerchant%26utm_campaign%3DGoogleMerchant&rct=j&q=&esrc=s&opi=95576897&sa=U&ved=0ahUKEwiyxlz9xMyCAxUspJUCHZEYDgQQ1SkIQYoAA&usq=AOvVaw1NnDQfPqwBBTdeZgeHNIiz)

imagens do diagrama: <https://www.canva.com/>

imagens do Autocad: Programa Autocad

imagens do circuito: Programa Phrotheus

Pesquisa:

<https://chat.openai.com/>

<https://www.mundoeletronica.com.br/>

<https://www.baudaeletronica.com.br/>

<https://www.eletrogate.com/esp32>

<https://www.makehero.com/categoria/wireless-e-iot/esp32/>

<https://curtocircuito.com.br/placa-doit-esp32-esp32-wroom-32d-wifi-bluetooth.html>

<https://www.espressif.com/en/products/socs/esp32>

5. Apêndices

A. Programação do transmissor

/*

PROJETO DE TCC - 3C 2023

PROJETO: ROBÔ PARA MONITORAMENTO DE AMBIENTES INÓSPITOS

COMPONENTE: TRANSMISSOR/CONTROLE REMOTO

AUTORES: GRUPO 5 - GUILHERME E MIGUEL

PROPÓSITO: PROGRAMAÇÃO DO CONTROLE REMOTO DO PROJETO DE TCC
- RMAI

SAIBA MAIS EM:

APROVEITE O CÓDIGO ;)

```
*/
```

```
//Inclusão de bibliotecas
```

```
#include <esp_now.h>
```

```
#include <WiFi.h>
```

```
#include <esp_wifi.h>
```

```
//Definições
```

```
#define xPin1 35
```

```
#define yPin1 34
```

```
#define xPin2 33
```

```
#define yPin2 32
```

```
//Declaração de variáveis
```

```
int eixo_X1;
```

```
int eixo_Y1;
```

```
int eixo_X2;
```

```
int eixo_Y2;
```

```
esp_now_peer_info_t peerInfo;
```

```
//Endereço MAC do receptor
```

```
uint8_t broadcastAddress[] = {0xA0, 0xB7, 0x65, 0x4F, 0x3A, 0x34};
```

```
//Estrutura para o envio das mensagens
typedef struct struct_message
{
    String dir;
    String rot;
} struct_message;

//Variavel de comando
struct_message comando;

//Inicialização do protocolo de comunicação ESP-NOW
void InitESPNow() {
    WiFi.disconnect();
    if (esp_now_init() == ESP_OK) {
        Serial.println("Iniciado com sucesso");
    }
    else {
        Serial.println("Erro de inicialização");
        ESP.restart();
    }
}

//Função de callback para o envio de dados
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
    Serial.print("\r\nLast Packet Send Status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" :
"Delivery Fail");
}

//Setup
void setup(){
    Serial.begin(115200);
```

```
pinMode(xPin1, INPUT);
pinMode(yPin1, INPUT);
pinMode(xPin2, INPUT);
pinMode(yPin2, INPUT);

WiFi.mode(WIFI_STA);

InitESPNow();

memcpy(peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.channel = 0;
peerInfo.encrypt = false;

if (esp_now_add_peer(&peerInfo) != ESP_OK){
  Serial.println("Failed to add peer");
  return;
}

}

//loop
void loop(){
  eixo_X1 = analogRead(xPin1);
  eixo_Y1 = analogRead(yPin1);
  eixo_X2 = analogRead(xPin2);
  eixo_Y2 = analogRead(yPin2);

  comando.dir = "parado";
  comando.rot = "parado";

  if(eixo_X1<1188){
```

```
    comando.dir = "tras";
}

if(eixo_X1>2212){
    comando.dir = "frente";
}

if(eixo_Y1<1188){
    comando.dir = "esquerda";
}

if(eixo_Y1>2212){
    comando.dir = "direita";
}

//Rotina de rotação da Camera

if(eixo_X2<1188){
    comando.rot = "tras";
}

if(eixo_X2>2212){
    comando.rot = "frente";
}

if(eixo_Y2<1188){
    comando.rot = "esquerda";
}

if(eixo_Y2>2212){
    comando.rot = "direita";
}
```

```
//Serial.println(eixo_X1);
//Serial.println(eixo_Y1);
Serial.println(comando.dir);
Serial.print(comando.rot);

esp_now_send(broadcastAddress, (uint8_t *) &comando, sizeof(comando));
}
```

B. Programação do Receptor

```
/*
```

PROJETO DE TCC - 3C 2023

PROJETO: ROBÔ PARA MONITORAMENTO DE AMBIENTES INÓSPITOS

COMPONENTE: TRANSMISSOR/CONTROLE REMOTO

AUTORES: GRUPO 5 - GUILHERME E MIGUEL

PROPÓSITO: PROGRAMAÇÃO DO CONTROLE REMOTO DO PROJETO DE TCC
- RMAI

SAIBA MAIS EM:

APROVEITE O CÓDIGO ;)

```
*/
```

```
//Inclusão de bibliotecas
```

```
#include <esp_now.h>
```

```
#include <WiFi.h>
```

```
#include <esp_wifi.h>
```

```
//Definições
```

```
#define xPin1 35
#define yPin1 34
#define xPin2 33
#define yPin2 32

//Declaração de variáveis
int eixo_X1;
int eixo_Y1;
int eixo_X2;
int eixo_Y2;

esp_now_peer_info_t peerInfo;

//Endereço MAC do receptor
uint8_t broadcastAddress[] = {0xA0, 0xB7, 0x65, 0x4F, 0x3A, 0x34};

//Estrutura para o envio das mensagens
typedef struct struct_message
{
    String dir;
    String rot;
} struct_message;

//Variavel de comando
struct_message comando;

//Inicialização do protocolo de comunicação ESP-NOW
void InitESPNow() {
    WiFi.disconnect();
    if (esp_now_init() == ESP_OK) {
        Serial.println("Iniciado com sucesso");
    }
}
```

```
}  
else {  
    Serial.println("Erro de inicialização");  
    ESP.restart();  
}  
}  
  
//Função de callback para o envio de dados  
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {  
    Serial.print("\r\nLast Packet Send Status:\t");  
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" :  
"Delivery Fail");  
}  
  
//Setup  
void setup(){  
    Serial.begin(115200);  
  
    pinMode(xPin1, INPUT);  
    pinMode(yPin1, INPUT);  
    pinMode(xPin2, INPUT);  
    pinMode(yPin2, INPUT);  
  
    WiFi.mode(WIFI_STA);  
  
    InitESPNow();  
  
    memcpy(peerInfo.peer_addr, broadcastAddress, 6);  
    peerInfo.channel = 0;  
    peerInfo.encrypt = false;  
  
    if (esp_now_add_peer(&peerInfo) != ESP_OK){  
        Serial.println("Failed to add peer");  
    }  
}
```



```
    return;
  }

}

//loop
void loop(){
  eixo_X1 = analogRead(xPin1);
  eixo_Y1 = analogRead(yPin1);
  eixo_X2 = analogRead(xPin2);
  eixo_Y2 = analogRead(yPin2);

  comando.dir = "parado";
  comando.rot = "parado";

  if(eixo_X1<1188){
    comando.dir = "tras";
  }

  if(eixo_X1>2212){
    comando.dir = "frente";
  }

  if(eixo_Y1<1188){
    comando.dir = "esquerda";
  }

  if(eixo_Y1>2212){
    comando.dir = "direita";
  }

  //Rotina de rotação da Camera
```

```
if(eixo_X2<1188){
    comando.rot = "tras";
}

if(eixo_X2>2212){
    comando.rot = "frente";
}

if(eixo_Y2<1188){
    comando.rot = "esquerda";
}

if(eixo_Y2>2212){
    comando.rot = "direita";
}

//Serial.println(eixo_X1);
//Serial.println(eixo_Y1);
Serial.println(comando.dir);
Serial.print(comando.rot);

esp_now_send(broadcastAddress, (uint8_t *) &comando, sizeof(comando));
}
```

C. Programação da Câmera

i. WebServer

```
#include "esp_camera.h"
#include <WiFi.h>
```

```

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//     Ensure ESP32 Wrover Module or other board with PSRAM is selected
//     Partial images will be transmitted if image exceeds buffer size
//
//     You must select partition scheme from the board menu that has at least 3MB
APP space.
//     Face Recognition is DISABLED for ESP32 and ESP32-S2, because it takes
up from 15
//     seconds to process single frame. Face Detection is ENABLED if PSRAM is
enabled as well

// =====
// Select camera model
// =====
#define CAMERA_MODEL_AI_THINKER // Has PSRAM

#include "camera_pins.h"

// =====
// Enter your WiFi credentials
// =====
const char* ssid = "*****";
const char* password = "*****";

void startCameraServer();
void setupLedFlash(int pin);

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
}

```

```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.frame_size = FRAMESIZE_UXGA;
config.pixel_format = PIXFORMAT_JPEG; // for streaming
//config.pixel_format = PIXFORMAT_RGB565; // for face detection/recognition
config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
config.fb_location = CAMERA_FB_IN_PSRAM;
config.jpeg_quality = 12;
config.fb_count = 1;

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
//           for larger pre-allocated frame buffer.
if(config.pixel_format == PIXFORMAT_JPEG){
    if(psramFound()){
```

```
    config.jpeg_quality = 10;
    config.fb_count = 2;
    config.grab_mode = CAMERA_GRAB_LATEST;
} else {
    // Limit the frame size when PSRAM is not available
    config.frame_size = FRAMESIZE_SVGA;
    config.fb_location = CAMERA_FB_IN_DRAM;
}
} else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#ifdef CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
}

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
}

// drop down frame size for higher initial frame rate
if(config.pixel_format == PIXFORMAT_JPEG){
```

```
s->set_framesize(s, FRAMESIZE_QVGA);
}

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

#if defined(CAMERA_MODEL_ESP32S3_EYE)
s->set_vflip(s, 1);
#endif

// Setup LED FLash if LED pin is defined in camera_pins.h
#if defined(LED_GPIO_NUM)
setupLedFlash(LED_GPIO_NUM);
#endif

WiFi.begin(ssid, password);
WiFi.setSleep(false);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
```

```
}  
  
void loop() {  
    // Do nothing. Everything is done in another task by the web server  
    delay(10000);  
}
```

ii. Pinos da Câmera

```
#if defined(CAMERA_MODEL_WROVER_KIT)  
#define PWDN_GPIO_NUM  -1  
#define RESET_GPIO_NUM  -1  
#define XCLK_GPIO_NUM  21  
#define SIOD_GPIO_NUM  26  
#define SIOC_GPIO_NUM  27  
  
#define Y9_GPIO_NUM  35  
#define Y8_GPIO_NUM  34  
#define Y7_GPIO_NUM  39  
#define Y6_GPIO_NUM  36  
#define Y5_GPIO_NUM  19  
#define Y4_GPIO_NUM  18  
#define Y3_GPIO_NUM  5  
#define Y2_GPIO_NUM  4  
#define VSYNC_GPIO_NUM  25  
#define HREF_GPIO_NUM  23  
#define PCLK_GPIO_NUM  22  
  
#elif defined(CAMERA_MODEL_ESP_EYE)  
#define PWDN_GPIO_NUM  -1  
#define RESET_GPIO_NUM  -1  
#define XCLK_GPIO_NUM  4
```

```
#define SIOD_GPIO_NUM 18
#define SIOC_GPIO_NUM 23

#define Y9_GPIO_NUM 36
#define Y8_GPIO_NUM 37
#define Y7_GPIO_NUM 38
#define Y6_GPIO_NUM 39
#define Y5_GPIO_NUM 35
#define Y4_GPIO_NUM 14
#define Y3_GPIO_NUM 13
#define Y2_GPIO_NUM 34

#define VSYNC_GPIO_NUM 5
#define HREF_GPIO_NUM 27
#define PCLK_GPIO_NUM 25

#define LED_GPIO_NUM 22

#ifdef CAMERA_MODEL_M5STACK_PSRAM
#define PWDN_GPIO_NUM -1
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM 27
#define SIOD_GPIO_NUM 25
#define SIOC_GPIO_NUM 23

#define Y9_GPIO_NUM 19
#define Y8_GPIO_NUM 36
#define Y7_GPIO_NUM 18
#define Y6_GPIO_NUM 39
#define Y5_GPIO_NUM 5
#define Y4_GPIO_NUM 34
#define Y3_GPIO_NUM 35
#define Y2_GPIO_NUM 32
```



```
#define VSYNC_GPIO_NUM 22
#define HREF_GPIO_NUM 26
#define PCLK_GPIO_NUM 21

#elif defined(CAMERA_MODEL_M5STACK_V2_PSRAM)
#define PWDN_GPIO_NUM -1
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM 27
#define SIOD_GPIO_NUM 22
#define SIOC_GPIO_NUM 23

#define Y9_GPIO_NUM 19
#define Y8_GPIO_NUM 36
#define Y7_GPIO_NUM 18
#define Y6_GPIO_NUM 39
#define Y5_GPIO_NUM 5
#define Y4_GPIO_NUM 34
#define Y3_GPIO_NUM 35
#define Y2_GPIO_NUM 32
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 26
#define PCLK_GPIO_NUM 21

#elif defined(CAMERA_MODEL_M5STACK_WIDE)
#define PWDN_GPIO_NUM -1
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM 27
#define SIOD_GPIO_NUM 22
#define SIOC_GPIO_NUM 23

#define Y9_GPIO_NUM 19
#define Y8_GPIO_NUM 36
```

```
#define Y7_GPIO_NUM    18
#define Y6_GPIO_NUM    39
#define Y5_GPIO_NUM    5
#define Y4_GPIO_NUM    34
#define Y3_GPIO_NUM    35
#define Y2_GPIO_NUM    32
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM  26
#define PCLK_GPIO_NUM  21

#define LED_GPIO_NUM   2

#ifdef CAMERA_MODEL_M5STACK_ESP32CAM
#define PWDN_GPIO_NUM  -1
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM  27
#define SIOD_GPIO_NUM  25
#define SIOC_GPIO_NUM  23

#define Y9_GPIO_NUM    19
#define Y8_GPIO_NUM    36
#define Y7_GPIO_NUM    18
#define Y6_GPIO_NUM    39
#define Y5_GPIO_NUM    5
#define Y4_GPIO_NUM    34
#define Y3_GPIO_NUM    35
#define Y2_GPIO_NUM    17
#define VSYNC_GPIO_NUM 22
#define HREF_GPIO_NUM  26
#define PCLK_GPIO_NUM  21

#ifdef CAMERA_MODEL_M5STACK_UNITCAM
```

```
#define PWDN_GPIO_NUM -1
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM 27
#define SIOD_GPIO_NUM 25
#define SIOC_GPIO_NUM 23

#define Y9_GPIO_NUM 19
#define Y8_GPIO_NUM 36
#define Y7_GPIO_NUM 18
#define Y6_GPIO_NUM 39
#define Y5_GPIO_NUM 5
#define Y4_GPIO_NUM 34
#define Y3_GPIO_NUM 35
#define Y2_GPIO_NUM 32
#define VSYNC_GPIO_NUM 22
#define HREF_GPIO_NUM 26
#define PCLK_GPIO_NUM 21

#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
```

```
#define Y2_GPIO_NUM    5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22

// 4 for flash led or 33 for normal led
#define LED_GPIO_NUM   4

#elif defined(CAMERA_MODEL_TTGO_T_JOURNAL)
#define PWDN_GPIO_NUM  0
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM 27
#define SIOD_GPIO_NUM 25
#define SIOC_GPIO_NUM 23

#define Y9_GPIO_NUM    19
#define Y8_GPIO_NUM    36
#define Y7_GPIO_NUM    18
#define Y6_GPIO_NUM    39
#define Y5_GPIO_NUM    5
#define Y4_GPIO_NUM    34
#define Y3_GPIO_NUM    35
#define Y2_GPIO_NUM    17
#define VSYNC_GPIO_NUM 22
#define HREF_GPIO_NUM 26
#define PCLK_GPIO_NUM 21

#elif defined(CAMERA_MODEL_XIAO_ESP32S3)
#define PWDN_GPIO_NUM  -1
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM  10
#define SIOD_GPIO_NUM  40
```

```
#define SIOC_GPIO_NUM 39

#define Y9_GPIO_NUM 48
#define Y8_GPIO_NUM 11
#define Y7_GPIO_NUM 12
#define Y6_GPIO_NUM 14
#define Y5_GPIO_NUM 16
#define Y4_GPIO_NUM 18
#define Y3_GPIO_NUM 17
#define Y2_GPIO_NUM 15
#define VSYNC_GPIO_NUM 38
#define HREF_GPIO_NUM 47
#define PCLK_GPIO_NUM 13

#elif defined(CAMERA_MODEL_ESP32_CAM_BOARD)
// The 18 pin header on the board has Y5 and Y3 swapped
#define USE_BOARD_HEADER 0
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM 33
#define XCLK_GPIO_NUM 4
#define SIOD_GPIO_NUM 18
#define SIOC_GPIO_NUM 23

#define Y9_GPIO_NUM 36
#define Y8_GPIO_NUM 19
#define Y7_GPIO_NUM 21
#define Y6_GPIO_NUM 39
#if USE_BOARD_HEADER
#define Y5_GPIO_NUM 13
#else
#define Y5_GPIO_NUM 35
#endif
#endif
```

```
#define Y4_GPIO_NUM    14
#if USE_BOARD_HEADER
#define Y3_GPIO_NUM    35
#else
#define Y3_GPIO_NUM    13
#endif
#define Y2_GPIO_NUM    34
#define VSYNC_GPIO_NUM  5
#define HREF_GPIO_NUM  27
#define PCLK_GPIO_NUM   25

#elif defined(CAMERA_MODEL_ESP32S3_CAM_LCD)
#define PWDN_GPIO_NUM   -1
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM   40
#define SIOD_GPIO_NUM   17
#define SIOC_GPIO_NUM   18

#define Y9_GPIO_NUM     39
#define Y8_GPIO_NUM     41
#define Y7_GPIO_NUM     42
#define Y6_GPIO_NUM     12
#define Y5_GPIO_NUM     3
#define Y4_GPIO_NUM     14
#define Y3_GPIO_NUM     47
#define Y2_GPIO_NUM     13
#define VSYNC_GPIO_NUM  21
#define HREF_GPIO_NUM   38
#define PCLK_GPIO_NUM   11

#elif defined(CAMERA_MODEL_ESP32S2_CAM_BOARD)
// The 18 pin header on the board has Y5 and Y3 swapped
```

```
#define USE_BOARD_HEADER 0
#define PWDN_GPIO_NUM 1
#define RESET_GPIO_NUM 2
#define XCLK_GPIO_NUM 42
#define SIOD_GPIO_NUM 41
#define SIOC_GPIO_NUM 18

#define Y9_GPIO_NUM 16
#define Y8_GPIO_NUM 39
#define Y7_GPIO_NUM 40
#define Y6_GPIO_NUM 15
#if USE_BOARD_HEADER
#define Y5_GPIO_NUM 12
#else
#define Y5_GPIO_NUM 13
#endif
#define Y4_GPIO_NUM 5
#if USE_BOARD_HEADER
#define Y3_GPIO_NUM 13
#else
#define Y3_GPIO_NUM 12
#endif
#define Y2_GPIO_NUM 14
#define VSYNC_GPIO_NUM 38
#define HREF_GPIO_NUM 4
#define PCLK_GPIO_NUM 3

#elif defined(CAMERA_MODEL_ESP32S3_EYE)
#define PWDN_GPIO_NUM -1
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 15
#define SIOD_GPIO_NUM 4
```

```
#define SIOC_GPIO_NUM 5
```

```
#define Y2_GPIO_NUM 11
```

```
#define Y3_GPIO_NUM 9
```

```
#define Y4_GPIO_NUM 8
```

```
#define Y5_GPIO_NUM 10
```

```
#define Y6_GPIO_NUM 12
```

```
#define Y7_GPIO_NUM 18
```

```
#define Y8_GPIO_NUM 17
```

```
#define Y9_GPIO_NUM 16
```

```
#define VSYNC_GPIO_NUM 6
```

```
#define HREF_GPIO_NUM 7
```

```
#define PCLK_GPIO_NUM 13
```

```
#elif defined(CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3) ||  
defined(CAMERA_MODEL_DFRobot_Romeo_ESP32S3)
```

```
#define PWDN_GPIO_NUM -1
```

```
#define RESET_GPIO_NUM -1
```

```
#define XCLK_GPIO_NUM 45
```

```
#define SIOD_GPIO_NUM 1
```

```
#define SIOC_GPIO_NUM 2
```

```
#define Y9_GPIO_NUM 48
```

```
#define Y8_GPIO_NUM 46
```

```
#define Y7_GPIO_NUM 8
```

```
#define Y6_GPIO_NUM 7
```

```
#define Y5_GPIO_NUM 4
```

```
#define Y4_GPIO_NUM 41
```

```
#define Y3_GPIO_NUM 40
```

```
#define Y2_GPIO_NUM 39
```

```
#define VSYNC_GPIO_NUM 6
```

```
#define HREF_GPIO_NUM 42
```



```
#define PCLK_GPIO_NUM 5
```

```
#else
```

```
#error "Camera model not selected"
```

```
#endif
```