Etec Jacinto Ferreira de Sá - 066 – Ourinhos
Técnico em Automação Industrial

ENTREGA DE GARRAFAS

Murilo Mota dos Santos[1]

Luiz Fernando Pucineli[2]

Nicolas Leonardo Dionizio[3]

Resumo: O projeto se trata de um processo de alimentação de garrafas, onde ocorre a limpeza, o envasamento, a pesagem, a conferência, o rosqueamento e por fim, o posicionamento da garrafa em uma prateleira. Este trabalho se aprofunda no processo de entrega da garrafa para o elevador. A automatização do sistema é feita para melhorar a eficiência da produção e minimizar a possibilidade de falhas. O objetivo do projeto é que o robô seja capaz de identificar as garrafas , coletadas automaticamente e uma garra robótica para transportar ao elevador , juntos fazem a entrega das garrafas para o elevador. O controle para o robô e a garra funcionar foi programado com um Arduino para ajustar a direção conforme os sensores e operar a garra em pontos específicos e foi realizado testes para verificar o processo. O projeto foi desenvolvido com base em projetos similares de robôs seguidores de linha e adaptaremos uma garra mecânica em cima desse robô, que tem uma base feita na impressora 3D , com tudo fizemos a programar o robô para ele seguir a linha corretamente e conseguir levar para o outro robô sem problemas e retornar para pegar a outra garrafa e assim segue um loop , com o objetivo de realizar tarefas automatizadas de transporte e manipulação de objetos em ambientes delimitados por trajetos pré-definidos. Para realizar a tarefa de transporte equipamos o nosso robô com sensores infravermelhos para detectar objetos a sua frente e identificar a garrafa, mas tudo isso junto com a garra mecânica, só que tudo isso em um robô simples e barato. O projeto contribuirá para o aprendizado pratico com automação, com a parte de eletrônica, programação e integração mecânica. Com esse projeto, conseguimos aprender na prática como funciona a automação de um sistema simples, desde a programação no Arduino até o uso de sensores e a montagem de uma garra mecânica. Enfrentamos desafios, mas conseguimos superálos com trabalho em equipe e dedicação. Foi uma experiência muito importante, pois nos mostrou como aplicar os conhecimentos de eletrônica, mecânica e programação em um projeto real. Além disso, nos sentimos mais preparados para o mercado de trabalho, já que entendemos melhor como funcionam os processos automatizados e como eles podem ser usados para facilitar tarefas do dia a dia nas indústria.

Palavras-chave: entrega, garrafas, transporte, tarefa, Arduino

[1] Ensino Fundamental II, EMEF Adelaide Pedroso Racanello Profa, Ourinhos /SP – murilo.santos197@etec.sp.gov.br

[2]Ensino Fundamental II,Fundação Universitaria Vida Cristã, Rio de Janeiro/RJ - luiz.pucineli@etec.sp.gov.br

[3]Ensino Fundamental II, EMEF Adelaide Pedroso Racanello Profa, Ourinhos /SP – nicolas.dionizio@etec.sp.gov.br

Abstract: This project concerns a bottle feeding process, encompassing cleaning, filling, weighing, checking, screwing, and finally, positioning the bottle on a shelf. This work focuses on the process of delivering the bottle to the elevator. The system's automation aims to improve production efficiency and minimize the possibility of errors. The project's objective is for the robot to be able to identify the bottles, collect them automatically, and use a robotic gripper to transport them to the elevator; together, they deliver the bottles to the elevator. The control system for the robot and gripper was programmed with an Arduino to adjust the direction according to the sensors and operate the gripper at specific points. Tests were conducted to verify the process. The project was developed based on similar line-following robot projects, and we adapted a mechanical gripper on top of this robot, which has a base made on a 3D printer. We programmed the robot to follow the line correctly, successfully transport the bottle to another robot without problems, and return to pick up the other bottle, thus creating a loop. The goal is to perform automated tasks of transporting and manipulating objects in environments delimited by predefined paths. To perform the transport task, we equipped our robot with infrared sensors to detect objects in front of it and identify the bottle, all in conjunction with the mechanical gripper, all within a simple and inexpensive robot. This project will contribute to practical learning in automation, covering electronics, programming, and mechanical integration. Through this project, we were able to learn in practice how to automate a simple system, from programming in Arduino to using sensors and assembling a mechanical gripper. We faced challenges, but we overcame them through teamwork and dedication. It was a very important experience because it showed us how to apply our knowledge of electronics, mechanics, and programming to a real project. Furthermore, we feel more prepared for the job market, as we better understand how automated processes work and how they can be used to facilitate daily tasks in industry.

Keywords: delivery, bottles, transport, task, Arduino

1 INTRODUÇÃO

Este trabalho tem como objetivo mostrar um projeto de automação voltado para a entrega de garrafas em um processo industrial. A ideia surgiu a partir da necessidade de facilitar e automatizar tarefas que, normalmente, são feitas de forma manual. Na entrega o projeto se concentra em fazer com que um robô, programado com Arduino, leve as garrafas até um elevador, usando sensores e uma garra mecânica. Isso é algo comum nas indústrias, e quisemos simular de forma simples como essa automação pode funcionar. O objetivo geral do projeto é fazer com que o robô consiga identificar a garrafa, pegar com a garra e levá-la até o elevador. Já os objetivos específicos são: montar a parte eletrônica do sistema, programar os sensores para seguir a linha e detectar a garrafa, construir a garra mecânica e integrar tudo isso com o Arduino. Com esse projeto, aprendemos na prática como funciona a integração entre programação, eletrônica e mecânica. Fizemos esse projeto porque acreditamos que a automação é uma área que está crescendo muito e será cada vez mais presente no nosso dia a dia

e no mercado de trabalho. Queremos mostrar que é possível montar algo funcional com materiais simples e de baixo custo. O problema que buscamos resolver foi: como fazer um sistema automático e barato que consiga transportar garrafas com eficiência dentro de um processo industrial. Nossa hipótese é que, mesmo com equipamentos simples como o Arduino e sensores básicos, dá para montar um robô que realiza essa tarefa de forma eficaz. Esperamos, no final, ter um protótipo funcionando bem, mostrando que conseguimos desenvolver um projeto completo de automação, e que ele pode ser aplicado em outros processos parecidos dentro da indústria ou até em trabalhos escolares e técnicos.

2 FUNCIONAMENTO

A parte das peças do robô seguidor de linha é fundamental para garantir seu funcionamento adequado. O projeto utiliza uma base reforçada, que serve para sustentar todos os componentes e proporcionar a estabilidade necessária para o robô. O material dela é resistente, como o acrílico, que oferece o equilíbrio ideal entre resistência e leveza. Ela foi projetada para suportar o peso e a estrutura de outros componentes, como os motores e a garra mecânica. Tal garra é composta por um braço articulado, que permite agarrar objetos com precisão. Ela é movida por servo motores, controlados por um sistema eletrônico integrado ao robô. Ela é projetada para ser leve e eficiente, sendo ajustada para ter uma força adequada para pegar objetos pequenos a médios sem danificá-los, e no tempo correto para não atrasar o processo. A estrutura da garra é feita de filamento da impressora 3d , garantindo durabilidade e desempenho. Outro componente importante na construção do robô é o motor de acionamento, que controla os movimentos. São utilizados motores DC, que permitem ao robô seguir a linha com precisão, realizando curvas e manobras conforme necessário. Os motores são conectados a rodas de borracha para garantir aderência ao solo e facilitar o movimento. Além disso, o robô conta com sensores infravermelhos (IR), responsáveis por identificar a linha no chão. Esses sensores captam a diferença de contraste entre a linha e o fundo, permitindo que o robô faça ajustes em sua trajetória para seguir a linha corretamente. Com essas peças, o robô seguidor de linha funciona de forma precisa e eficiente, com a base reforçada oferecendo estabilidade e a garra mecânica permitindo interação com objetos. Com base nisso, foi utilizado o Arduino que atua como o cérebro do robô , é ele que controla os motores e interpreta os sensores infravermelhos que detectam as linhas , através dessas informações ele

envia comandos para o motor para o mantê-lo na trajetória correta. Utilizando a ponte h (driver de motor) nos permite controlar a velocidade do motor , invertendo a polaridade da tensão aplicada ao motor , assim controlando seu sentido de rotação , que também recebe sinais do Arduino. O L298N possui dois canais , permitindo o controle de até dois motores simultaneamente , cada canal possui dois pinos para controle de direção (in1 , in2 para um motor e in3 e in4 para o outro) e um pino enable (Ena , Enb) para controle PWM , além dos pinos de alimentação e terra. Ele é um componente essencial para controlar a velocidade e direção isolando a parte do circuito de controle (Arduino) da parte de potência (motores) , garantindo segurança e desempenho. Com o sensor de obstáculos possibilita que ele identifique a garra com mais facilidade e eficiência , emitindo um feixe de luz infravermelho que identifica a luz refletida por um objeto próximo. O sensor possui um led emissor e um receptor fotodiodo , quando a luz é refletida de volta ao receptor , indicando a presença do objeto.

2.1 Montagem

O processo de montagem do robô seguidor de linha exige atenção, organização e cuidado para garantir que todos os componentes fiquem bem fixados e funcionem de forma correta. A montagem começa pela base reforçada, que é o ponto de partida para todo o projeto. Ela é posicionada em uma superfície limpa e nivelada para facilitar o trabalho, evitando torções ou desalinhamentos. Primeiro, são fixados os motores DC na base, utilizando suportes próprios e parafusos adequados, garantindo firmeza e evitando vibrações excessivas que possam comprometer a leitura dos sensores. As rodas de borracha são acopladas aos eixos dos motores, certificando-se de que estejam alinhadas e girem livremente, sem atrito.Na parte frontal do robô, são posicionados os sensores infravermelhos (IR) voltados para o chão. Eles são fixados a uma pequena estrutura de suporte para manter a distância ideal entre os sensores e a superfície, permitindo a detecção precisa da linha. Essa fixação é feita com cuidado para evitar movimentos ou desalinhamentos, que poderiam gerar leituras incorretas.O Arduino é instalado em uma posição central na base, onde possa receber facilmente os cabos vindos dos motores, sensores e demais componentes. Próximo a ele é fixado o módulo ponte H L298N, que será responsável pelo controle da velocidade e direção dos motores. A fiação é organizada com abraçadeiras plásticas para evitar que os fios fiquem soltos, isso nos garante um visual limpo e que nos previnam de acidentes.A garra mecânica é montada na parte superior da base, utilizando um suporte reforçado

para suportar o peso e a movimentação. Os servomotores responsáveis pelo movimento da garra são conectados ao Arduino, permitindo que os comandos sejam executados com precisão. Para melhorar a funcionalidade, o sensor de obstáculos é posicionado de forma que fique voltado para a frente da garra, detectando objetos antes mesmo de serem agarrados, otimizando o tempo de resposta. Após todos os componentes estarem devidamente fixados, inicia- se a ligação elétrica. Os cabos dos motores são conectados à ponte H, seguindo a polaridade correta, enquanto os sensores são conectados às entradas digitais do Arduino. O cabo de alimentação é ligado de forma segura, garantindo que não haja risco de curto-circuito. Por fim, é feita uma verificação geral, certificando-se de que não há fios soltos, peças mal fixadas ou sensores desalinhados. A organização e o cuidado nessa etapa são essenciais para que o robô funcione de forma estável e confiável. Com todos os componentes no lugar, o robô está fisicamente pronto para ser programado e teste.

2.2 Programação

A programação desenvolvida para o robô seguidor de linha com garra mecânica teve como objetivo principal automatizar o transporte de uma garrafa de água entre dois pontos determinados, utilizando como referência uma linha no chão. Para isso, foi utilizada a plataforma Arduino Uno**,** que atua como a unidade central de processamento, recebendo as informações dos sensores e enviando os comandos adequados para os atuadores do sistema.

O robô conta com sensores infravermelhos posicionados estrategicamente na parte inferior, responsáveis por realizar a leitura contínua do trajeto. Os sensores laterais detectam a presença da linha e permitem que o veículo corrija sua trajetória sempre que houver desvio, garantindo um deslocamento estável e preciso. Já o sensor central desempenha um papel fundamental na identificação de pontos de parada específicos, chamados de marcadores**,** que foram inseridos no percurso para sinalizar as áreas destinadas à coleta e à entrega da carga.

Quando o sensor central identifica o marcador de coleta, o Arduino envia o comando para interromper o movimento dos motores e aciona o servo motor responsável pela garra mecânica. Nesse instante, a garra se fecha com firmeza sobre a garrafa de água, garantindo a segurança do transporte. Após a conclusão dessa etapa, o sistema reinicia o deslocamento, fazendo com que o robô continue a seguir a linha em direção ao próximo marcador.

Ao atingir o ponto de entrega, novamente o sensor central detecta o marcador e o robô realiza a parada programada. O servo motor é então acionado para abrir a garra, liberando a garrafa de forma controlada no local desejado. Esse procedimento de coleta, transporte e entrega se assemelha a um processo automatizado de logística industrial**,** no qual máquinas executam tarefas repetitivas com precisão e eficiência.

Após a liberação da carga, o robô é instruído a retornar pelo mesmo trajeto até o ponto de origem, fechando assim o ciclo operacional. O programa foi estruturado para que esse ciclo funcione em loop contínuo, permitindo que o robô esteja sempre disponível para realizar novas operações de coleta e entrega, sem a necessidade de intervenção humana.

Essa integração entre sensores, motores e servo motor da garra representa uma aplicação prática dos conceitos de automação e robótica, evidenciando como sistemas embarcados podem simular, em escala reduzida, processos amplamente utilizados em linhas de produção industriais e sistemas de transporte automatizados.

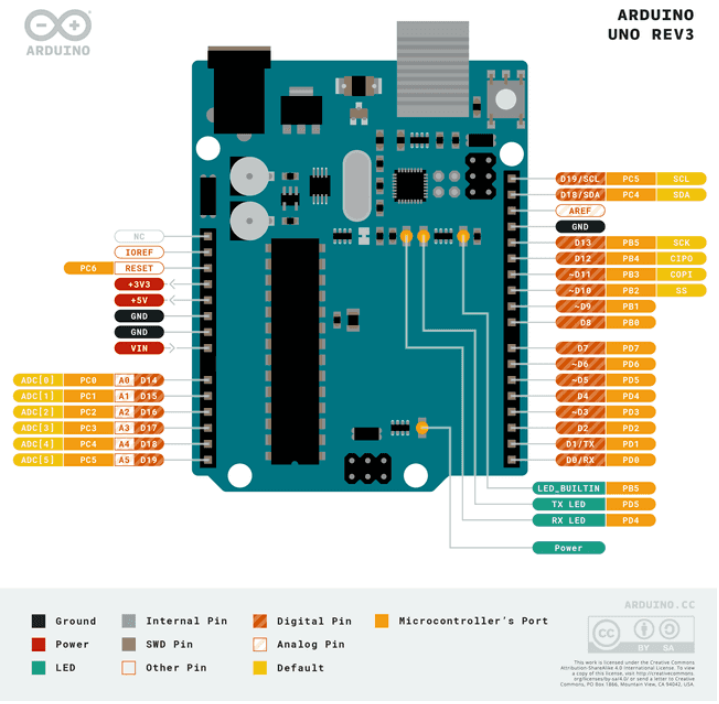

Figura 1 - Declaração de bibliotecas e definição dos pinos do robô

Primeira imagem: Mostra a declaração das bibliotecas utilizadas, como a servo.h, além da definição dos pinos conectados aos sensores infravermelhos, motores e servo da garra. Essa parte do código é essencial porque faz o mapeamento das entradas e saídas que o Arduino irá controlar durante todo o processo.

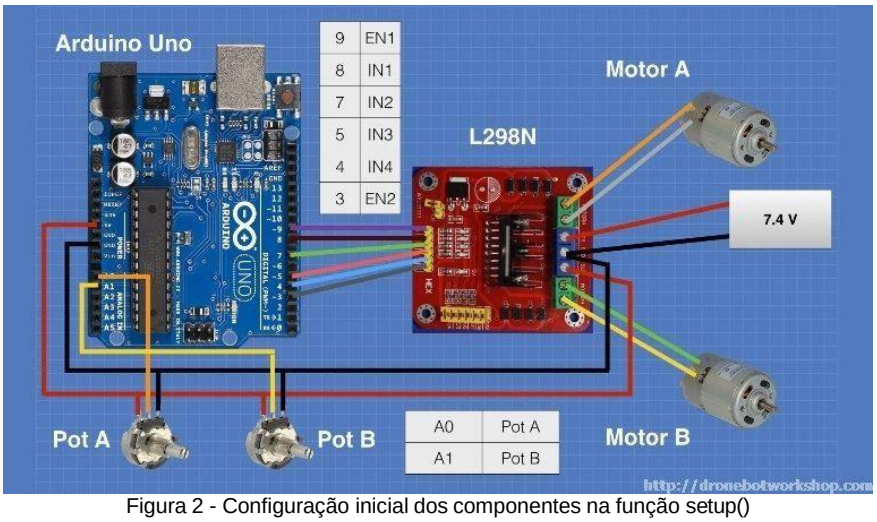

Figura 2 - Configuração inicial dos componentes na função setup()

Segunda imagem: Apresenta a configuração inicial dentro da função setup(). Nela, o código define quais pinos são de entrada e saída, inicializa a comunicação com o monitor serial e posiciona a garra em sua posição inicial através do servo motor. Essa etapa garante que, ao ligar o robô, todos os componentes estejam prontos para funcionar corretamente.

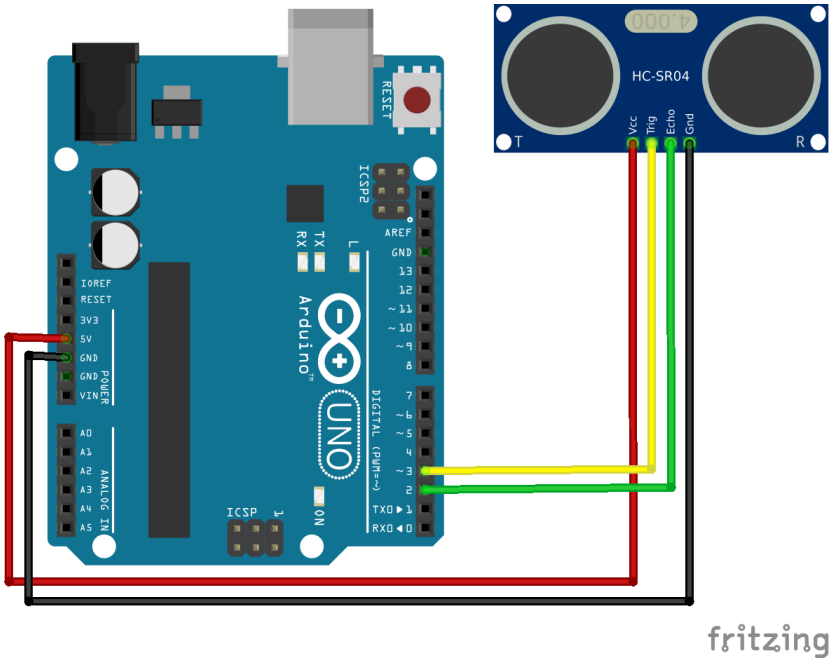

Figura 3 - Lógica principal do robô dentro da função loop()

Terceira imagem: Mostra a lógica do robô durante a função loop(), responsável por repetir o ciclo de funcionamento indefinidamente. Nesse trecho, o Arduino lê constantemente os sensores infravermelhos para corrigir a movimentação do robô na linha. Também é possível observar os trechos que fazem o robô parar em pontos específicos e acionar a garra para pegar ou soltar a garrafa.

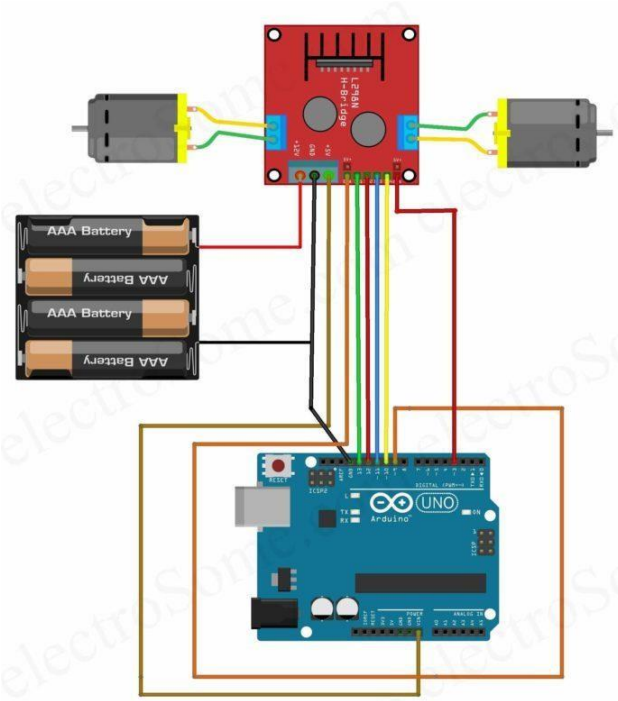

Figura 4 - Funções auxiliares para movimentação e controle da garra

Quarta imagem: Exibe as funções auxiliares que organizam o código, como “andar para frente”, “virar à esquerda”, “virar à direita”, “parar” e também os comandos para abrir ou fechar a garra. Essa estrutura deixa a programação mais clara e modular, facilitando futuras alterações e a compreensão do fluxo de trabalho do robô.

Todas as fotos foram gerados do software oficial de programação dos microcontroladores Arduino

https://www.arduino.cc/en/software

3 AJUSTES

O robô montado, com todas as programações, peças e componentes em seus devidos lugares, o teste foi realizado em um ambiente controlado, utilizando uma pista preparada

com uma linha de cor preta em cima de uma base branca, garantindo que os sensores infravermelhos façam a leitura com precisão.O robô é colocado no início da pista e inicia seu deslocamento. Assim, o sistema é ligado e o Arduino começa a interpretar os sinais enviados pelos sensores IR. Logo nos primeiros movimentos, observa-se a resposta do robô às curvas e retas da pista. Caso um dos motores apresente velocidade diferente, a ponte H (L298N) permite fazer ajustes por meio do controle PWM, garantindo que ambos os lados avancem de forma equilibrada. Esse ajuste foi fundamental para evitar que o robô se desvie da linha. A garra mecânica também foi testada. Utilizando o sensor de obstáculos, o robô identifica a aproximação de uma demarcação posicionada estrategicamente na pista. Ao detectá-la, o sistema aciona o servo motor da garra, realizando a abertura e fechamento no tempo certo, garantindo que o objeto seja agarrado de forma firme, mas sem causar danos. Durante o teste, são observados aspectos como estabilidade, tempo de resposta dos sensores, suavidade nas curvas e eficiência na detecção de obstáculos. Por fim, após rodar toda a pista com sucesso, o robô foi testado em diferentes condições , como, mudanças de iluminação, para verificar sua adaptação. Essa etapa garantiu que o robô funcione mesmo a iluminação estando clara ou escura.

4 METODOLOGIA

A metodologia deste trabalho foi elaborada com o objetivo de estruturar o desenvolvimento de um robô seguidor de linha, baseado em sensores infravermelhos e controlado por microcontrolador Arduino. A pesquisa é de natureza qualitativa, de caráter descritivo e aplicado, utilizou como procedimento técnico a pesquisa experimental, escolha e montagem dos componentes eletrônicos, programação do sistema e realização de testes práticos para análise de desempenho em diferentes trajetos. A montagem do robô foi feita com cola quente , colando os 4 motores com as rodas ,que foram feitas na impressora 3D, na parte superior da mesma base foram colocados os componentes,foi colado na ponta da estrutura os dois sensores infravermelhos que fazem a leitura da linha preta para o robô seguir o trajeto, após isso foi colocado o suporte para a bateria, são duas pilhas que garantem autonomia para o robô, a ponte H e o Arduino na parte superior, a segunda base esta com pequenos pilares de separação, para que a mesma nao encoste nos componentes, na parte de cima, colocamos a garra,

tambem feita na impressora 3D, controlada por um servo motor. Nesse trabalho, tanto as rodas , componentes e bases , serviram para dar a altura necessária para quando a garra pegar a garrafa, para que a mesma não seja arrastada no chão. O robô vai se orientar por uma linha demarcada por uma fita, por onde deve passar, tal fita está localizada na base do protótipo. Esse trajeto é um circuito fechado, onde o robô coleta a garrafa no final da esteira e a leva até o elevador. Na etapa da programação, estruturou- se a própria com base em alguns robôs de entrega, máquina de estados, definindo cada etapa do funcionamento do robô.

Em seguida, utilizou-se o software Tinkercad, onde o funcionamento lógico do robô foi visualizado, utilizando sensores e atuadores do projeto físico ou semelhantes. Nessa etapa, foram ajustados detalhes da programação em C++

5 CONSIDERAÇÕES FINAIS

O desenvolvimento do robô seguidor de linha demonstrou, de forma prática, a aplicação de conceitos fundamentais de robótica, eletrônica e, principalmente, de programação. Por meio dos sensores, motores e da lógica de controle, isto é, do código desenvolvido foi possível construir um robô autônomo, capaz de identificar trajetórias e reconhecer obstáculos, como garrafas. Além de colocar em prática o conhecimento teórico e prático adquirido ao longo do curso, o projeto evidenciou a importância da programação no funcionamento do sistema, mostrando o quanto pequenas alterações no código podem impactar significativamente o desempenho do robô.

Durante o desenvolvimento, foi possível compreender de forma mais profunda o funcionamento dos motores e sensores, bem como a maneira como eles interagem para detectar e seguir a linha da pista. Essa experiência contribuiu para o aprimoramento das habilidades técnicas e consolidou o aprendizado em áreas essenciais da mecatrônica e da automação.

ANEXOS:

Fluxograma:

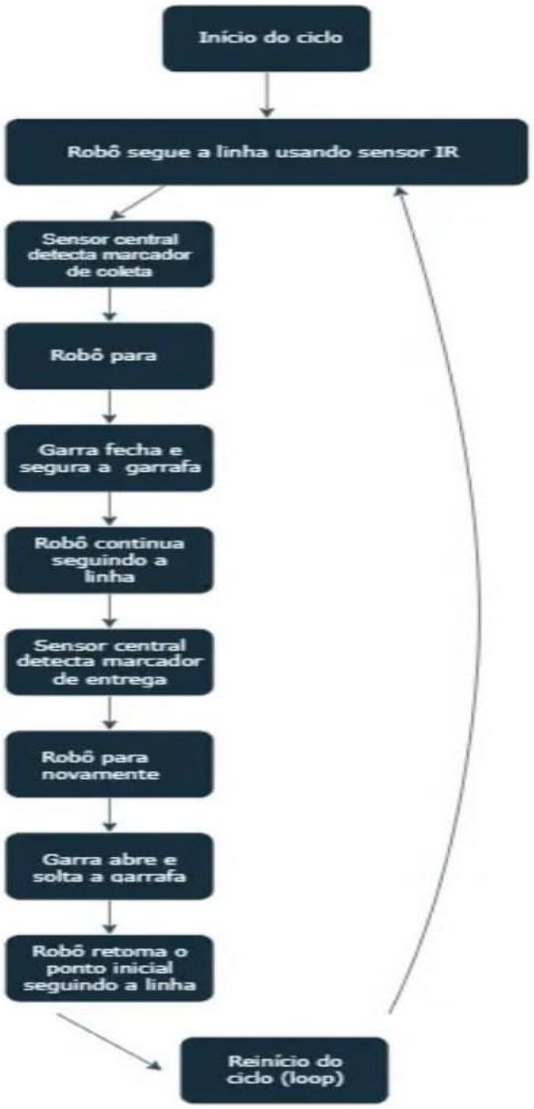

Figura 1 - Fluxograma

PROGRAMAÇÃO - Código da programação do robô:

```
/* Robô seguidor de linha com garra (Arduino UNO)
- 2 sensores IR para seguir linha (A0,A1)
- 1 sensor IR para marcar ponto (A2)
- Ultrassom HC-SR04 (TRIG 7, ECHO 6) para detectar garrafa
- L298N com ENA/ENB (PWM) + IN1..IN4 para direção
- Servo na pin 9 (garra)
*/

#include <Servo.h>

// --- PINAGEM ---
const int IR_LEFT = A0;
const int IR_RIGHT = A1;
const int IR_POINT = A2;

const int TRIG_PIN = 7;
const int ECHO_PIN = 6;

const int SERVO_PIN = 9;
```

```
// L298N pins (ENA/ENB = PWM)
const int ENA = 10;   // PWM motor esquerdo
const int IN1 = 5;
const int IN2 = 4;

const int ENB = 11;   // PWM motor direito
const int IN3 = 3;
const int IN4 = 2;

// --- PARÂMETROS DE MOVIMENTO ---
const int BASE_SPEED = 200;   // 0..255
const int TURN_SPEED = 150;
const int SLOW_SPEED = 120;

const int DIST_STOP_CM = 10;  // distância para considerar que há garrafa (cm)

// Estados do robô
enum Estado { WAIT_LOAD=0, PICKUP=1, GO_TO_B=2, DROP=3, RETURN_TO_A=4 };
Estado estado = WAIT_LOAD;

Servo garra;

unsigned long lastPointDetectTime = 0;
const unsigned long pointDebounce = 250; // ms

// ---------- SETUP ----------
void setup() {
Serial.begin(9600);
// Motores
pinMode(ENA, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);

pinMode(ENB, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);

// Sensores
pinMode(IR_LEFT, INPUT);
pinMode(IR_RIGHT, INPUT);
pinMode(IR_POINT, INPUT);

pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

// Servo
garra.attach(SERVO_PIN);
abrirGarra();

parar();
delay(200);
}
// ---------- LOOP ----------
void loop() {
switch (estado) {
case WAIT_LOAD:
parar();
abrirGarra();
if (garrafaDetectada()) {
Serial.println("Garrafa detectada -> pegar");
delay(200); // debounce
estado = PICKUP;
}
break;

case PICKUP:
// fechar garra e afastar um pouco
fecharGarra();
delay(600);
leavePointForward(); // desloca um pouco para sair do ponto
estado = GO_TO_B;
break;

case GO_TO_B:
seguirLinha();
```

```
if (pontoDetectadoDebounced()) {
Serial.println("Chegou no ponto B");
parar();
delay(200);
estado = DROP;
}
break;

case DROP:
abrirGarra(); // solta a garrafa
delay(600);
// avance um pouco para não re-detectar o mesmo ponto
leavePointForward();
estado = RETURN_TO_A;
break;

case RETURN_TO_A:
seguirLinha();
if (pontoDetectadoDebounced()) {
Serial.println("Chegou no ponto A (retorno)");
parar();
delay(200);
estado = WAIT_LOAD;
}
break;
}
}

// ---------- FUNÇÕES DE MOVIMENTO ----------
void frente(int speed) {
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
analogWrite(ENA, speed);

digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(ENB, speed);
}

void parar() {
analogWrite(ENA,  0);
analogWrite(ENB, 0);
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
}

void virarEsquerda() {
// reduzir velocidade esquerda, manter direita
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
analogWrite(ENA, TURN_SPEED/2);

digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(ENB, TURN_SPEED);
}

void virarDireita() {
analogWrite(ENA, TURN_SPEED);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);

analogWrite(ENB, TURN_SPEED/2);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
}

// desloca um pouco pra frente para sair do ponto (quando pegar/soltar)
void leavePointForward() {
frente(SLOW_SPEED);
delay(500);
parar();
}
```

```
// ---------- SEGUIDOR DE LINHA (lógica simples) ----------
/* Nota: sensores IR típicos retornam LOW ao detectar superfície escura (linha preta)
e HIGH no piso claro. Ajuste conforme seus sensores. */
void seguirLinha() {
int esq = digitalRead(IR_LEFT);
int dir = digitalRead(IR_RIGHT);

// ambos HIGH (no branco) -> pode ser perda de linha: seguir reto por segurança
if (esq == HIGH && dir == HIGH) {
frente(BASE_SPEED);
}
// esquerdo detecta preto -> virar levemente esquerda
else if (esq == LOW && dir == HIGH) {
virarEsquerda();
}
// direito detecta preto -> virar levemente direita
else if (esq == HIGH && dir == LOW) {
virarDireita();
}
// ambos LOW -> linha centralizada (cruzamento ou marca grande)
else {
// aqui podemos interpretar como ficar no centro (vai pra frente)
frente(BASE_SPEED);
}
}

// ---------- DETECÇÃO DE PONTO (debounce) ----------
bool pontoDetectado() {
return digitalRead(IR_POINT) == LOW; // marque sua fita/estria como 'preto'
}
bool pontoDetectadoDebounced() {
if (pontoDetectado()) {
if (millis() - lastPointDetectTime > pointDebounce) {
lastPointDetectTime = millis();
return true;
}
}
return false;
}

// ---------- ULTRASSOM ----------
long getDistanceCM() {
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
long dur = pulseIn(ECHO_PIN, HIGH, 30000); // timeout 30ms
if (dur == 0) return 999; // nada detectado
long cm = dur * 0.034 / 2;
return cm;
}

bool garrafaDetectada() {
long d = getDistanceCM();
Serial.print("Dist (cm): "); Serial.println(d);
return d > 0 && d < DIST_STOP_CM;
}

// ---------- GARRA (servo) ----------
void abrirGarra() {
// ajuste o ângulo conforme sua montagem
garra.write(90); // aberto
}
void fecharGarra() {
garra.write(10); // fechado (ajuste)
}
```