

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FEELT – FACULDADE DE ENGENHARIA ELÉTRICA
ENGENHARIA DE COMPUTAÇÃO

LUCAS ALBINO MARTINS
12011ECP022

**ROBÓTICA: TRABALHO DE IMPLEMENTAÇÃO 01 – RELATÓRIO
CONSTRUÇÃO DE UM ROBÔ SEGUIDOR DE TRILHA.**

UBERLÂNDIA
2021

RELATÓRIO CONSTRUÇÃO DE UM ROBÔ SEGUIDOR DE TRILHA.

Relatório elaborado pelo
aluno Lucas Albino Martins,
matriculado no curso de
Engenharia da Computação,
orientado pelo professor da
disciplina de Robótica Keiji
Yamanaka.

UBERLÂNDIA
2021

Sumário

1. Objetivos.....	4
1.1 Objetivo geral.....	4
1.2 Objetivos específicos.....	4
2. Justificativa.....	4
3. Metodologia.....	5
4. Referencial teórico.....	5
5. Material utilizado.....	6
6. Descrição dos componentes.....	7
6.1. Kit chassi 2WD.....	7
6.2 Sensores.....	8
6.2.1. Sensor Óptico	
Reflexivo TCRT5000.....	8
6.2.2. Sensor Ultrassônico	
HC-SR04.....	8
6.3. Ponte H dupla L293D.....	9
6.4. Microcontrolador	
ATmega38 (Arduino NANO)	9
7. Descrição do protótipo.....	10
7.1. Sensores de precisão.....	11
7.2. Controle de velocidade e direção.....	12
7.3. Chassi 2WD.....	13
8. Resultados.....	14
9. Esquemático.....	14
10. Código.....	15
11. Conclusão.....	21
12. Referências.....	22

1. Objetivos.

1.1 Objetivo geral.

Temos como objetivo estender o conhecimento que adquirimos em sala de aula sobre Robótica, através de uma projeção e materialização de um robô seguidor de trilha, com intuito de demonstrarmos a funcionalidade em termos didáticos e práticos, afim de auxiliar no processo de aprendizagem em nossa graduação. Podendo ser utilizado materiais de sucata ou materiais novos como projetos desenhados e impressos em impressora 3D ou artesanais feitos a partir de madeira ou papelão.

1.2 Objetivo específicos.

Dentre o contexto apresentado através do objetivo geral, podemos destacar alguns objetivos específicos, como demonstrar toda a parte de construção e matérias utilizados. Sendo assim, podemos destacar os seguintes objetivos então visaremos:

- Levantamento de informações de o que é um robô seguidor de trilha/faixa.
- Funcionamento de sensores infravermelhos e um sensor de distância.
- A partir do conhecimento passado em sala de aula construir um protótipo.
- Definir então matérias necessários a partir do protótipo e então construir o modelo.

2. Justificativa.

O grande desenvolvimento tecnológico vem reduzindo consideravelmente o custo de componentes eletrônicos, proporcionando

uma maior acessibilidade a determinados produtos para os mais diversos seguimentos. Este processo também ocorre com a robótica e cada vez mais emprega por indústrias de produção em média e larga escala. A área da educação segue caminho, pois com o aparecimento de novas ideias para o uso de tecnologias já existentes, há a possibilidade e de maior acesso à tecnologia robótica, inclusive para o uso pessoal. Com base no contexto nos dados levantados acima, o referido projeto busca demonstrar um produto com as tecnológicas mais atuais sendo sustentado pelas ideias criações tecnológicas, ou seja, aprender na prática, demonstrando desde da montagem até a parte do funcionamento geral.

3. Referencial teórico.

Abordando alguns aspectos teóricos no tocante a ideia de robô seguidor de faixa, em conformidade com o assunto que foi discutido através das aulas de Robótica, juntamente com o acervo absorvido através de pesquisas, os quais foram a base para o desenvolvimento desse projeto, apresentando o modelo de um robô seguidor de faixa de acordo com a teoria robótica de quesito mínimo como concluir um circuito determinado pelo usuário sem haver saída do mesmo, caso haja alguma saída ela tem que ser por algum motivo externo ao circuito como um obstáculo.

4. Metodologia.

Em vista do aprendizado obtidos em sala de aula, em conjunto com projetos disponibilizados na internet quanto a esse mesmo tema, este projeto visa a contribuir na formação dos discentes matriculados na disciplina de Robótica, com o desenvolvimento específico de um robô seguidor de faixa, essa por sua vez será capaz de seguir um determinado circuito, com isso auxiliando-nos no conhecimento prático sobre robótica e robôs seguidores e faixa.

O funcionamento do robô seguidor de linha é simples, basicamente ele deve seguir andando por cima de uma linha de cor preta ou branca. Seus circuitos podem alternar entre cores do campo e da faixa, no nosso

caso iremos assumir que a pista é branca e faixa e preta. Usando então dois sensores infravermelho que detectam a presença ou não desta faixa. De acordo com a combinação dos sensores o carrinho irá para frente ou virar para um dos lados, demonstrando de um modo mais técnico o módulo do sensor de obstáculo infravermelho IR é bem simples, ele emite uma luz infravermelha por um LED negro e capta o reflexo com um LED receptor, como sabemos, a luz reflete em superfícies claras e é absorvida em superfícies negras, como a fita isolante, sendo assim o LED receptor irá detectar a luz infravermelha no branco e não detectar no preto, para uma melhor eficácia do sensor, a superfície em contraste com a faixa preta deve ser branca, para ajustar a sensibilidade ao contraste. Quanto a sua programação os sensores estão ligados ao modulo controlador do Arduino NANO, o mesmo também está ligado um circuito integrado para controle da velocidade dos motores a partir da resposta dos sensores, esse circuito integrado e conhecido como PONTE-H, todo o circuito sendo alimentado por quatro baterias de 1,5v e uma chave de liga/desliga, tudo isso integrado a uma estrutura com três rodas, duas controladas pelos motores DC e uma terceira apenas para manobras funcionando como auxiliar de estabilidade, um complemento ligado a estrutura desse robô foi um sensor para obstáculos, sua lógica de funcionamento é feita de um emissor e um receptor ultrassônico, onde o sensor emite (emissor) sinais ultrassônicos que serão refletidos no obstáculo / objeto retornando ao sensor (receptor). Com base no tempo que o sinal emitido levou para retornar ao sensor, o mesmo efetua o cálculo da distância.

5. Materiais utilizados.

- Kit chassi 2WD (2 rodas)
- Base para quatro pilhas AA.
- 4 Pilhas 1.5v AA.
- Dois motores DC (3~6v).
- Roda boba(universal).
- Chave liga/desliga.

- Parafusos.
- Fios 0,3mm.
- Led bicolor 5mm.
- Led 10mm.
- Resistor de 220 ohms.
- Arduino Nano (Microcontrolador ATmega38).
- Mini Protoboard (400 Pontos).
- Circuito Integrado L293D (Ponte H dupla L293 / L293D).
- Sensor infravermelho linha trilha (TCRT5000).
- Sensor Ultrassônico HC-SR04.

6. Descrição dos componentes.

6.1. Kit chassi 2WD.

O kit do chassi 2WD é composto pelo chassi em acrílico, rodas e motores DC com caixa de redução, discos encoders, espaçadores, parafusos, porcas e um suporte para 4 pilhas AA. O kit chassi 2WD permite desenvolver projetos bem interessantes desde robôs seguidores de linha até um carrinho que possa ser controlado de forma remota.

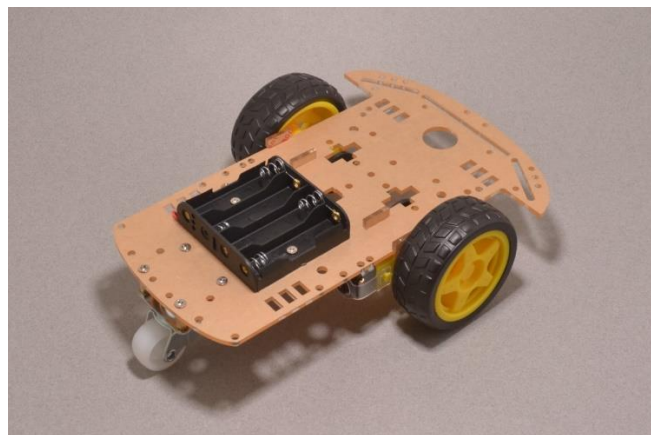


Figura 1. – Kit Chassi 2WD.

6.2 Sensores.

6.2.1 Sensor – Sensor Óptico Reflexivo TCRT5000

Este sensor opera por meio de um sistema de reflexão infravermelho, onde um LED emissor IR e um foto-transistor IR ficam lado a lado. Quando o sensor é aproximado de um objeto, a luz infravermelha do LED emissor é refletida e ativa o foto-transistor que gera uma saída. A tensão da saída varia conforme a cor do objeto. Quando esse sensor é lido como sinal digital, cores mais claras geram sinal baixo e cores escuras, como o preto, geram sinal alto.

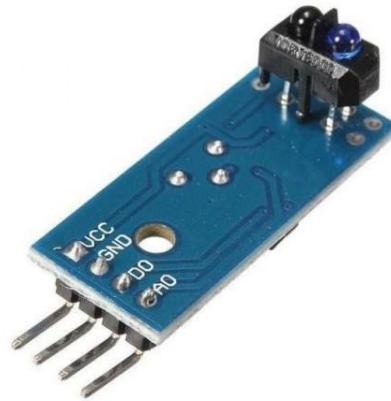


Figura 2 – Sensor infravermelho TCRT5000.

6.2.2. Sensor Ultrassônico HC-SR04.

Para realizar a medição de distância, é necessário deixar o pino Trigger em nível alto por 10 microssegundos. Dessa forma o sensor emite uma onda, que, quando encontra um obstáculo, rebate e volta em direção ao sensor. Entre o tempo que o sinal é emitido e é recebido 7 de volta, o pino Echo fica em nível alto. Com isso é possível calcular a distância que a onda percorreu até a voltar, visto que a mesma possui a velocidade do som e o tempo que ela demorou pra voltar foi registrado.



Figura 3. – Sensor ultrassônico HC-SR04.

6.3. Ponte H dupla L293D.

O L293D é um circuito integrado que contém duas pontes H completas (ou quatro half H-bridges). Com ele, é possível controlar dois motores DC bidirecionalmente, quatro solenoides ou um motor de passo. A versão com a letra “D” vem com diodos internos de proteção. Suas aplicações incluem o controle de cargas indutivas, como solenoides, relés, motores de passo e DC bipolares, assim como outras cargas de alta corrente/tensão. Suporta tensões de 4,5V até 36V, e é alimentado com tensão de 5V.

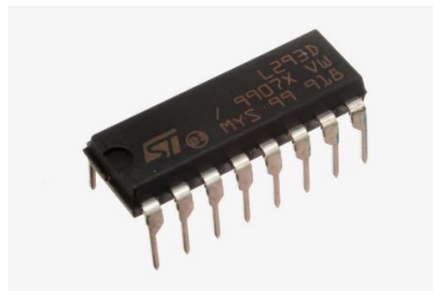


Figura 4. – L293D.

6.4 Microcontrolador ATmega38 (Arduino NANO).

O Arduino Nano é uma versão para ser acoplada a uma protoboard com uma porta USB acoplada. Ele é pequeno e completo. O Nano foi desenvolvido e é produzido pela Gravitech. Eletronicamente ele tem tudo o que o Duemilanove tem com mais pinos de entrada analógica e um jumper acoplado de +5V AREF. A placa Nano V3 é uma pequena versão das placas da tecnologia Arduino, parecida com o Arduino UNO pois também possui um chip ATmega328, porém na versão SMD. Possui um

conector para cabos Mini-USB para gravação (o cabo acompanha a placa, conforme a segunda foto do produto). É pequena, completa e muito amigável. Uma das principais diferenças entre esta placa e a Arduino UNO, é que esta placa possui 2 entradas analógicas a mais, os pinos A6 e A7. Esta placa não possui um conector para fonte externa, mas é possível alimentá-la pelo pino Vin. O Arduino Nano automaticamente seleciona a maior alimentação fornecida.

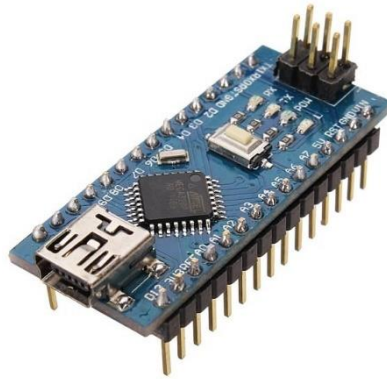


Figura 5. - Microcontrolador ATmega38 (Arduino NANO).

7. Descrição do protótipo.

O protótipo é dividido em três partes, primeira parte e a estrutura um chassi 2WD com uma roda boba anexada na parte de trás e uma base para quatro pilhas, o chassi possui dois motores DC (3~6v), a segunda parte está nos sensores, foi instalado três sensores, nos quais dois sensores infravermelho de linha/trilha (TCRT5000) e um sensor ultrassônico HC-SR04, a terceira parte é o circuito com o circuito integrado L293D (Ponte H) que controla a velocidade dos motores e um microcontrolador ATmega38 (Arduino Nano) preso a uma mini protoboard (400 pinos) e um led bipolar e uma chave liga/desliga.

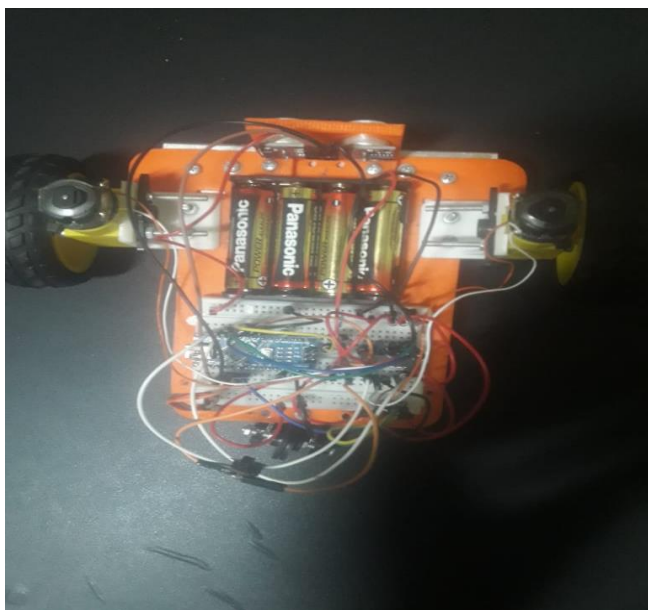


Figura 6. – Corpo do protótipo.

7.1. Sensores de precisão:

A detecção de obstáculos é feita pelo controle e processamento dos sinais emitidos e enviados, respectivamente, pelo sensor HC-SR04. Para tal, se a distância identificada é menor do que 10 cm., é enviado um outro sinal, porém dessa vez para a placa principal, indicando que há um obstáculo próximo, e o protótipo então para, dá ré e procura um novo caminho.

A detecção da linha para que o robô possa seguir a linha, foram instalados dois sensores de refletância. Portanto, a resposta deles sobre a luminosidade é um pico de tensão (leitura digital igual a 1) caso se encontrem na ausência de luz e uma tensão mínima (leitura digital igual a 0) quando há luz sendo refletida de volta. Isto é, quando o sensor está sobre a linha preta, não há luz sendo refletida, portanto a resposta da leitura é 1. Já quando ele está fora dela, a resposta é 0. Dessa forma, quando ambos os sensores da esquerda e da direita estiverem fora da linha, o carrinho deverá andar para frente. Já quando o sensor da esquerda estiver sobre a linha e o da direita, fora, ele deverá virar para esquerda. O contrário ocorre quando o sensor da direita estiver sobre a linha e o da esquerda, fora dela.

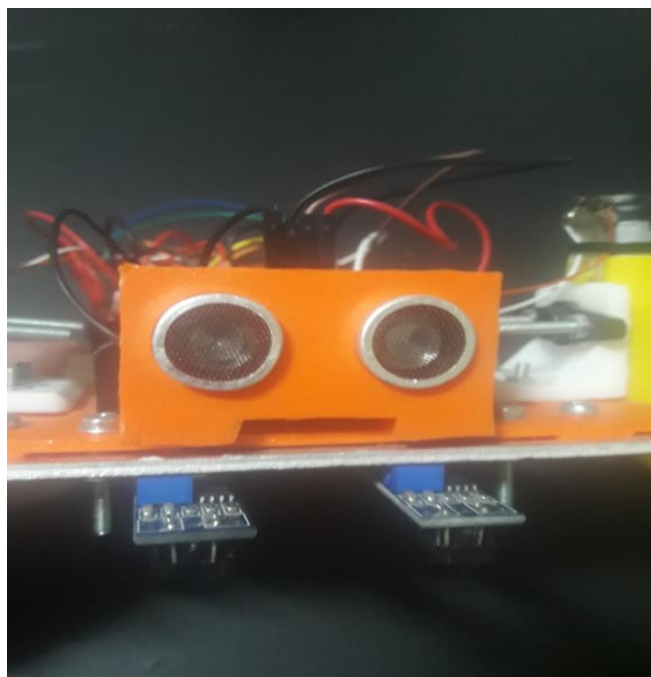


Figura 7. – Sensores do protótipo.

7.2. Controle de velocidade e direção.

Para possibilitar a movimentação do robô, foram utilizados dois motores (para duas rodas). Entretanto, o controle de velocidade e de direção é realizado um a um, um para o da direita e outro para o da esquerda. Neste caso, o lado direito é designado M1 e o esquerdo, M2. Nos pinos de controle de direção, as saídas dependem da tensão gerada em duas das portas de cada motor. As combinações de tensão (LOW, LOW) e (HIGH, HIGH) levam ao travamento das rodas. Já a (HIGH, LOW) faz com que se movimentem para frente e (LOW, HIGH), para trás. O controle de velocidade, por sua vez, é realizado com base em um sinal PWM (Pulse with Modulation). Essa técnica permite simular uma saída de tensão analógica através de pulsos de tensão digitais (isto é, que possuem apenas um valor mínimo e máximo). Assim, mantendo-se a mesma frequência de pulsos e variando apenas a duração de cada um deles, é possível modular a tensão média, possui como parâmetros a porta de saída e um valor que varia de 0 a 255, fazendo com que essa tensão varie de 0 V a 5 V. Esse valor de tensão chega, então aos motores, que se movimentam mais ou menos rapidamente de acordo com ele, esse controle é feito pelo circuito integrado L293D.

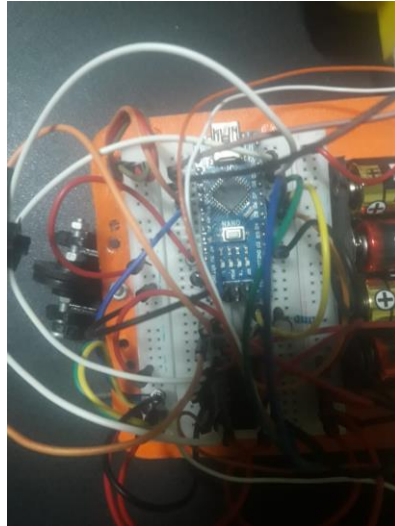


Figura 8. – Circuito controlador (L293D e Arduino NANO).

7.3 Chassi 2WD.

Chassi feito a base de acrílico, possui um par de motores DC (3~6v) com caixa de redução, em uma das extremidades possui uma roda boba (universal) para acompanhar os movimentos. Os motores presos ao chassi possuem a mobilidade limitada. Possui um suporte para quatro pilhas AA.



Figura 9. – Rodas do Chassi 2WD.

8. Resultados.

Durante o desenvolvimento do projeto, foi necessário a realização de testes e cálculos para se determinar certos parâmetros. Como, por exemplo, o cálculo da velocidade ideal, o que em certas horas foi um problema a ser enfrentado. O principal problema a ser resolvido era encontrar uma velocidade que fosse capaz de vencer o atrito, ou seja, que os motores tivessem força o suficiente para que o carrinho andasse. Mas que também fosse uma velocidade ideal para a detecção da linha preta e, conseqüentemente, a realização de curvas e ajuste do percurso. Outro problema mais leve foi a distância entre os sensores infravermelhos, para conseguir completar melhor o percurso a distância foi alterada até uma melhor estabilidade ao andar.

9. Esquemático.

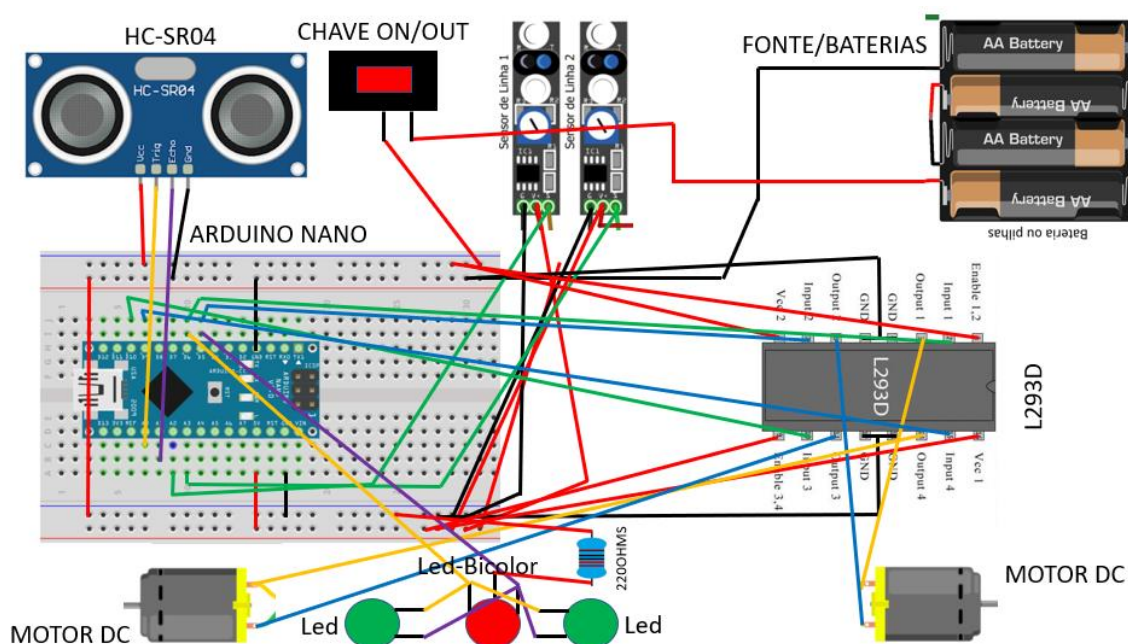


Figura 10. – Circuito esquemático.

10. Código.

```
#include <Servo.h>
#include <NewPing.h>
#define LEFT_SENSOR_PIN A2
#define RIGHT_SENSOR_PIN A3

//L293D
//Motor A
#define LEFT_MOTOR_PIN_1 5
#define LEFT_MOTOR_PIN_2 6
//Motor B
#define RIGHT_MOTOR_PIN_1 10
#define RIGHT_MOTOR_PIN_2 9

#define SENSOR_THRESHOLD 390

void LEFT (void);
void RIGHT(void);
void STOP (void);

short left_sensor;
short right_sensor;

#define BLACK 0
#define WHITE 1

#define TRIG_PIN A0 //analog input 1
#define ECHO_PIN A1 //analog input 2
#define MAX_DIST 200
int distance = 100;

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DIST);

void setup()
```

```

{

    pinMode(A2,INPUT_PULLUP);
    pinMode(A3,INPUT_PULLUP);

    pinMode(LEFT_MOTOR_PIN_1,OUTPUT);
    pinMode(LEFT_MOTOR_PIN_1,OUTPUT);

    pinMode(LEFT_MOTOR_PIN_1,OUTPUT);
    pinMode(RIGHT_MOTOR_PIN_1,OUTPUT);
    pinMode(LEFT_MOTOR_PIN_2,OUTPUT);
    pinMode(RIGHT_MOTOR_PIN_2,OUTPUT);

    pinMode(LEFT_SENSOR_PIN,INPUT);
    pinMode(RIGHT_SENSOR_PIN,INPUT);

    Serial.begin(9600);
    while (!Serial);
    BREAK();

    Serial.println("START!");

}

int readPing(){
    delay(7);
    int cm = sonar.ping_cm();
    if (cm==0){
        cm=250;
    }

    return cm;
}

void readSensors() {
    left_sensor = analogRead(LEFT_SENSOR_PIN);

```



```

right_sensor = analogRead(RIGHT_SENSOR_PIN);

Serial.print(left_sensor);
Serial.print("x");
Serial.print(right_sensor);
Serial.print("*");

if (left_sensor < SENSOR_THRESHOLD) {
    left_sensor = BLACK;
} else {
    left_sensor = WHITE;
}

if (right_sensor < SENSOR_THRESHOLD) {
    right_sensor = BLACK;
} else {
    right_sensor = WHITE;
}

    distance = readPing();

// Serial.print(left_sensor);
// Serial.print("x");
// Serial.print(right_sensor);
// Serial.print("*");
// Serial.print(distance);
// Serial.println(" cm");

}

void DoObstacle(){
    if (distance < 10){
        BREAK(); //Parar
        BACK(); //Ré
        delay(400);
    }
}

```

```

    BREAK(); //Parar

    LEFT(); //Iniciar giro
    delay(500);
    //Girar
    while(true) {

        readSensors();

        if (right_sensor != BLACK) {
            break;
        }
    }
}

//0 = expectedRight
//1 = expectedLeft
//2 = nothing
int expected = 0;

void loop()
{
    readSensors();
    DoObstacle();
    //DoServo(); //Opcional

    if (left_sensor == BLACK && right_sensor == WHITE){
        Serial.println("left_sensor");
        expected = 1;
        RIGHT2();
    } else if (left_sensor == WHITE && right_sensor == BLACK) {
        Serial.println("right_sensor");
        expected = 0;
        LEFT2();
    } else {

```

```

        Serial.println("FORWARD");
        if (expected == 1 && right_sensor == BLACK || expected == 0 && left_sensor
== BLACK ) {
            expected = 2;
        }
        if (expected == 1) {
            RIGHT();
        } else if (expected == 0) {
            LEFT();
        } else if (left_sensor == WHITE && right_sensor == WHITE) {
            FORWARD();
        }
    }
}

```

```

void FORWARD() {
    //Serial.println("FORWARD");
    digitalWrite(LEFT_MOTOR_PIN_1, HIGH);
    digitalWrite(LEFT_MOTOR_PIN_2, LOW);
    digitalWrite(RIGHT_MOTOR_PIN_1, LOW);
    digitalWrite(RIGHT_MOTOR_PIN_2, HIGH);
}

```

```

void BACK() {
    digitalWrite(LEFT_MOTOR_PIN_1, LOW);
    digitalWrite(LEFT_MOTOR_PIN_2, HIGH);
    digitalWrite(RIGHT_MOTOR_PIN_1, HIGH);
    digitalWrite(RIGHT_MOTOR_PIN_2, LOW);
}

```

```

void RIGHT2 (void)
{
    digitalWrite(LEFT_MOTOR_PIN_1, HIGH); //AQUI
    digitalWrite(LEFT_MOTOR_PIN_2, LOW); //AQUI
    digitalWrite(RIGHT_MOTOR_PIN_1, LOW);
}

```

```

    digitalWrite(RIGHT_MOTOR_PIN_2, LOW);
}

void RIGHT (void)
{
    digitalWrite(LEFT_MOTOR_PIN_1, HIGH); //AQUI
    digitalWrite(LEFT_MOTOR_PIN_2, LOW); //AQUI
    digitalWrite(RIGHT_MOTOR_PIN_1, HIGH);
    digitalWrite(RIGHT_MOTOR_PIN_2, LOW);
}

void LEFT2 (void)
{
    digitalWrite(LEFT_MOTOR_PIN_1, LOW);
    digitalWrite(LEFT_MOTOR_PIN_2, LOW);
    digitalWrite(RIGHT_MOTOR_PIN_1, LOW); //AQUI
    digitalWrite(RIGHT_MOTOR_PIN_2, HIGH); //AQUI
}

void LEFT (void)
{
    digitalWrite(LEFT_MOTOR_PIN_1, LOW);
    digitalWrite(LEFT_MOTOR_PIN_2, HIGH);
    digitalWrite(RIGHT_MOTOR_PIN_1, LOW); //AQUI
    digitalWrite(RIGHT_MOTOR_PIN_2, HIGH); //AQUI
}

void STOP (void)
{
    digitalWrite(LEFT_MOTOR_PIN_1, LOW);
    digitalWrite(LEFT_MOTOR_PIN_2, LOW);
    digitalWrite(RIGHT_MOTOR_PIN_1, LOW);
    digitalWrite(RIGHT_MOTOR_PIN_2, LOW);
}

```

```
void BREAK (void)
{
    digitalWrite(LEFT_MOTOR_PIN_1, HIGH);
    digitalWrite(LEFT_MOTOR_PIN_2, HIGH);
    digitalWrite(RIGHT_MOTOR_PIN_1, HIGH);
    digitalWrite(RIGHT_MOTOR_PIN_2, HIGH);
}
```

11. Conclusão

Com a conclusão do projeto, acarretou a possibilidade de experienciar inúmeras situações da parte prática do curso de Engenharia de Computação, o que agregou uma vivência bastante enriquecedora. Com a realização do trabalho, foi possível constatar a importância das boas práticas da eletrônica e da programação, especialmente no que tange a organização do projeto. A grande parte dos erros cometidos não se deu na lógica em si do robô, que é relativamente simples, mas em sua execução. Cabos com mau contato, soldas mal feitas e falta de capricho na hora da montagem dos componentes foram as maiores dificuldades e os maiores causadores de erros e danos ao trabalho. Por outro lado, o aprendizado que pode ser obtido foi imensurável. O fato de ter-se trabalhado com dois tipos de sensores possibilitou um amplo conhecimento acerca das ligações entre eles, deles com as placas e das placas entre si. Os conceitos de input e output, imprescindíveis para as respostas dadas pelo programa, puderam ser melhor consolidados, por exemplo. Por fim, a união desses dois sensores permite uma análise bastante rica dos arredores do “robô” a ser construído, mesmo enquanto ocorre sua movimentação. O fato de ele seguir uma linha permite delimitar um “circuito” a ser percorrido, mesmo com a interrupção dele por obstáculos. Isso poderia resultar em uma aplicação industrial, como duas máquinas que precisem se comunicar através de um trajeto pré-definido (e sujeito a descontinuidades). Ou mesmo demonstrar de maneira primitiva a tecnologia empregada em carros autônomos. Em suma, os conhecimentos adquiridos através do desenvolvimento desse projeto são de extrema utilidade em inúmeros

campos da saúde, indústria e inteligência artificial. Sendo assim, tende a agregar conceitos e experiências de extrema valia

12. Referências

BAUDAELETRONICA. Arduíno NANO R3. 2021. Disponível em: <<https://www.baudaeletronica.com.br/placa-nano-r3.html>>. Acessado em: 20-03-2021.

CHILTON, A. Using Defect Sensors to Detect Signaling Problems and Prevent Train Collisions. 2021. Disponível em <<https://www.azosensors.com/article.aspx?ArticleID=237>>. Acessado em: 20-03-2021.

FILIPEFLOP. Como conectar o sensor HC-SR04 ao Arduino. 2019. Disponível em: <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Acessado em: 20-03-2021.

Motor DC 3-6V com Caixa de Redução e Eixo Duplo. 2021. Disponível em: <<https://www.filipeflop.com/produto/motor-dc-3-6v-com-caixa-de-reducao-e-eixo-duplo/>>. Acessado em: 20-03-2021.

Sensor de Distância Ultrassônico HC-SR04. 2021. Disponível em: <<https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>>. Acessado em: 20-03-2021.

FLESCH, U. The Application Of Infrared-Sensors In Medicine. 2021. Disponível em: <<https://www.spiedigitallibrary.org/conference-proceedings-of-spie/0395/0000/The-Application-Of-Infrared-Sensors-In-Medicine/10.1117/12.935204.short?SSO=1>>. Acessado em: 20-03-2021.

LASER. Laboratório Avançado de Sistemas Embarcados e Robótica. 2016. Disponível em: <<http://laser.dainf.ct.utfpr.edu.br/doku.php>>. Acessado em: 20-03-2021.

MASTERWALKER. Como usar com Arduino – Sensor Óptico Reflexivo TCRT5000. 2019. Disponível em: <<http://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-sensor-reflexivo-tcrt5000/>>. Acessado em: 20-03-2021.

MECAWEB. PWM - Modulação Por Largura de Pulso. 2019. Disponível em: <http://www.mecaweb.com.br/eletronica/content/e_pwm>. Acessado em: 20-03-2021.

NETO, A. de M. Lógica Fuzzy. 2014. Disponível em: <<https://www.ime.usp.br/~adao/LOGICAFUZZY2017F.pdf>>. Acessado em: 20-03-2021.

ROBOCORE. Robô Seguidor de Linha. 2019. Disponível em: <<https://www.robocore.net/tutorials/robo-seguidor-de-linha.html>>. Acessado em: 20-03-2021.