

Instituto Federal de Ciência, Tecnologia e Educação do estado de São Paulo.

Curso de Graduação em Engenharia Eletrônica

ROBÓTICA APLICADA À COMPETIÇÃO EDUCACIONAL

Relatório da disciplina
introdução à engenharia com
o Profº Ricardo Pires e Profº
Alexandre Brincalepe
Campo.

IGOR GALDEANO RODRIGUES

SP3037223

JOSÉ AUGUSTO SANTOS

SP3037444

LUÍS OTÁVIO LOPES AMORIM

SP3034178

SÃO PAULO

2019

SUMÁRIO

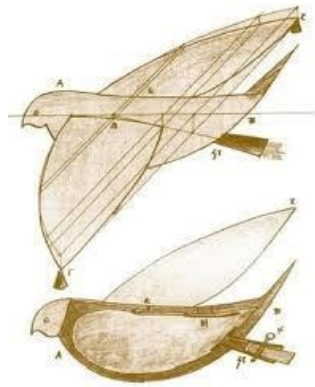
1. INTRODUÇÃO	2
1.1. Motivação	2
1.2. Objetivos	3
1.3. Justificativa	4
2. METODOLOGIA	4
3. CRONOGRAMA	5
4. SENSORES	5
4.1. Sensor de infravermelho	6
4.2. Sensor de Ultrassom	6
4.3. Modo de teste	7
5. ATUADORES	7
5.1. Ponte H	8
5.2. Sinal PWM	9
5.3. Modo de teste	10
6. CONCLUSÃO	10
REFERÊNCIAS	11
APÊNDICE A: CÓDIGO PARA TESTE DOS SENSORES	13
APÊNDICE B: CÓDIGO PARA TESTE DOS MOTORES	17

1. INTRODUÇÃO

1.1. Motivação

A primeira ideia que se conhece de um dispositivo autômato foi batizada de “O pombo” (figura 1) por seu criador, um matemático grego chamado Arquitas de Tarento, e foi esboçada em 350 a.C. A partir de então a robótica começou a ser difundida pelo mundo. Outro exemplo de robô famoso antigo é o “Cavaleiro mecânico” de Leonardo da Vinci (figura 2)(SILVA, 2017).

Figura 1- "O pombo" de Arquitas de Tarento



Fonte:

<https://www.tecmundo.com.br/robotica/66432-tecnologia-conheca-6-robos-imaginados-antiguidade.htm>

Figura 2 - Cavaleiro Mecânico de Leonardo da Vinci



Fonte: https://pt.wikipedia.org/wiki/Cavaleiro_mec%C3%A2nico_de_Leonardo

O intuito da robótica é auxiliar a produção ao automatizar os processos industriais (Equipe Robótica e Eletrônica, 2014). Porém, após a criação dos primeiros computadores modernos, esse ramo tecnológico sofreu vários avanços em uma velocidade muito elevada, até que hoje, menos de 100 anos depois, já existem máquinas dotadas de inteligência artificial permitindo que elas executem as mais variadas tarefas para fins pessoais domésticos, empresariais e industriais. Exemplos de máquinas contemporâneas são os da empresa *Boston Dynamics* como o “Atlas” (figura 3) um robô humanoide com grandes habilidades de equilíbrio e movimentação (Boston Dynamics, 2019).

Figura 3 - Atlas



Fonte: <https://www.bostondynamics.com/atlas>

O projeto do robô locomotor para competição visa aproximar esse conhecimento dos estudantes de ensino médio e técnico, para que estes alunos se interessem pela área e por projetos em geral. Para isso será feita uma parceria entre discentes do curso superior com os de 2º grau em busca de não apenas fomentar o interesse dos últimos nesse ramo tecnológico, mas também incentivar neles uma procura por um curso superior voltado a essa área que é de extrema importância para o desenvolvimento de um país.

Na modernidade são utilizados até mesmo recursos de incentivo como olimpíadas de robótica para alunos de ensino médio. Através da robótica é possível realizar experimentos da física, cálculos matemáticos, desenho geométrico, atividades de observação, bem como realizar atividades individuais e, principalmente, em grupos. Esse fator faz com que a aplicação da robótica nas salas de aula seja executada de maneira lúdica em que a aprendizagem ocorra de maneira simplificada e prazerosa.

1.2. Objetivos

O objetivo deste projeto é desenvolver um carrinho de acordo com as normas da Olimpíada Brasileira de Robótica (OBR) visando o menor custo de produção e maior aproveitamento dos componentes utilizados. O robô, através de sensores, deverá se guiar e

seguir o percurso demarcado em preto na pista desviando de obstáculos e buscando concluir a prova sem nenhum erro no menor tempo possível.

Além disso, o projeto procura desenvolver a habilidade de trabalho em equipe no aluno e sua capacidade de solucionar problemas e buscar conhecimento por conta própria, qualidades essenciais em um engenheiro.

1.3. Justificativa

Anualmente, no Brasil, ocorre a OBR para o ensino médio. Esse robô será projetado para participar da competição. Segundo a organização da olimpíada:

“Há dois pontos de vista sob os quais responder a esta pergunta: com foco na ferramenta educacional ou com foco no tipo de recurso humano formado. Em primeiro lugar, a robótica e a automação sobre áreas estratégicas para o país no caminho para o seu desenvolvimento. Apesar de ser uma área em franca expansão no mundo, o Brasil tem se situado de forma marginal nessa área, arriscando-se a perder um imenso potencial para a geração de empregos, técnicas, tecnologias e produtos devido, principalmente, à falta de incentivo para a formação de recursos humanos na área. Além de praticamente não produzir robôs em território nacional, o Brasil também não possui uma cultura que estimule uma maior utilização de tecnologias robóticas no parque tecnológico ou mesmo nas residências.”. (Olimpíada Brasileira de robótica, 2019)

Portanto a robótica é essencial para o desenvolvimento do país, e um contato com ela desde o ensino médio pode garantir uma melhoria na qualidade do aprendizado e um aumento do interesse dos jovens pela área.

2. METODOLOGIA

O projeto ocorrerá principalmente em duas etapas: pesquisa e montagem.

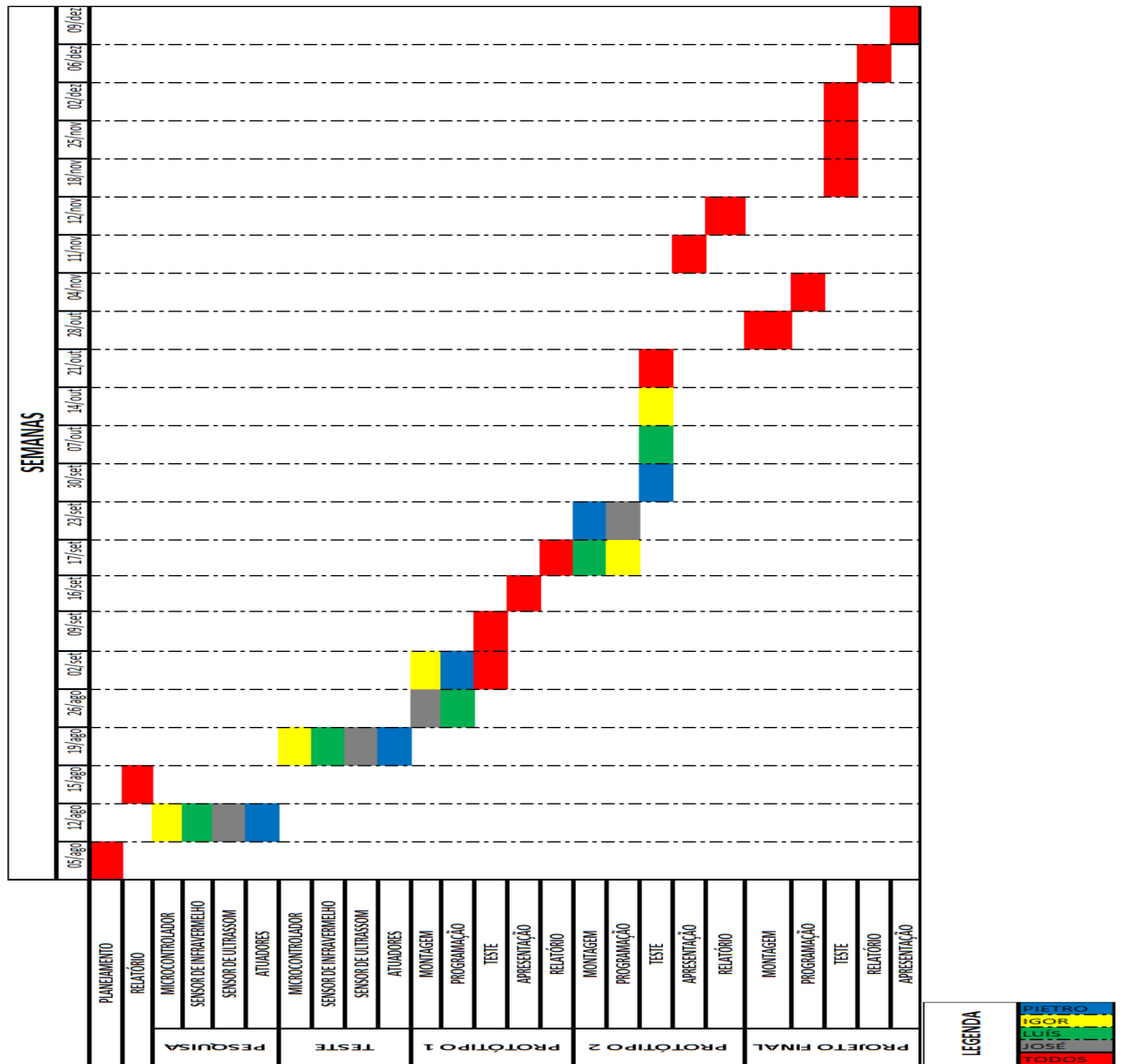
Na parte de pesquisa o conhecimento necessário para a criação do robô será buscado pelos alunos sendo utilizada a ajuda de livros e dos professores. Além disso, será necessário buscar pelos melhores componentes para serem utilizados, tendo sempre em mente o melhor custo-benefício.

Na etapa de montagem serão feitos dois protótipos e o projeto final, já pronto para cumprir os desafios propostos pela OBR. Os protótipos serão feitos para o teste dos sensores, atuadores (motores) e serão remontados, tanto o hardware quanto o software, até que estejam funcionando perfeitamente.

- Protótipo 1: Tem como objetivo a verificação do funcionamento do microcontrolador (arduino) aliado aos sensores de ultrassom e infravermelho.
- Protótipo 2: Os motores serão adicionados ao protótipo 1 e o carrinho será apresentado.
- Projeto final: O carrinho já finalizado realizará o percurso proposto.

3. CRONOGRAMA

Figura 4 - Cronograma do projeto



Fonte: Autores

4. SENSORES

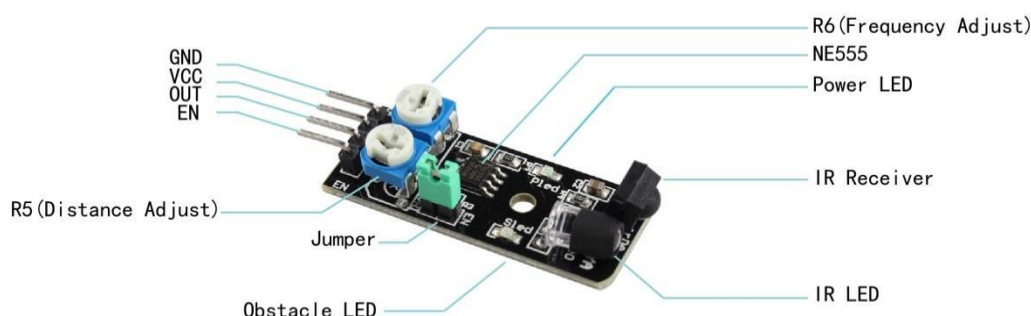
Para a montagem do robô serão utilizados dois tipos de sensores. O primeiro deles, o sensor de infravermelho, deve guiar o autômato pelo trajeto correto a ser seguido, já o segundo, o sensor de ultrassom vai detectar os obstáculos existentes e informar como proceder em caso de proximidade com um deles.

4.1. Sensor de infravermelho

O sensor de infravermelho (sensor de IR) utilizado (figura 5) possui um regulador e dois LEDs (*light emitting diode*) sendo eles um o emissor de ondas infravermelho e o outro o receptor dessa mesma frequência de ondas. Quando essas ondas atingem superfícies claras (o branco fora da pista) elas são completamente refletidas, portanto o LED receptor capta energia. Já no caso de a superfície atingida pelas ondas ser escura (o preto que delimita o trajeto) as ondas não serão tanto refletidas quanto captadas pelo receptor (Etecnophlies, 2019), assim, o robô saberá se está fora ou dentro do percurso. Por fim, o regulador serve para ajustar a sensibilidade do sensor.

Para que o carrinho saiba quando deve fazer uma curva serão utilizados cinco sensores de IR. Idealmente o sensor do meio deve estar sempre sobre a linha preta, ou seja, recebendo as ondas refletidas. Já os sensores laterais são os indicadores de curva, quando o sensor da direita captar a onda refletida, por exemplo, o robô saberá que o trajeto continua nessa direção e assim fará uma curva.

Figura 5 - Sensor de infravermelho



Fonte: <http://osoyoo.com/2017/07/24/arduino-lesson-obstacle-avoidance-sensor/>

4.2. Sensor de Ultrassom

Assim como o sensor de IR, o sensor de ultrassom (figura 6) funciona com base na reflexão de ondas. Ele também possui um emissor e um receptor de ondas, porém, desta vez, o receptor sempre captará a onda, pois sempre ocorre a reflexão. Porém o intuito deste sensor é encontrar obstáculos e descobrir a distância até eles a partir do tempo entre a emissão e a recepção da onda (Mota, 2019).

A onda sonora no ar possui sempre a velocidade de 0,034cm/ms, assim, é possível calcular a distância até o obstáculo a partir da equação 1.

$$D = \frac{0,034 \times t}{2} \quad (1)$$

Em que D representa a distância até o obstáculo, em centímetros, e t é o tempo, em milissegundos, entre a emissão da onda e a posterior recepção.

Dessa forma poderemos saber quando haverá algum obstáculo no trajeto e instruir, por meio da programação como o robô devera agir ao se aproximar deste.

Figura 6 - Sensor de ultrassom



Fonte:

<http://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-sensor-ultrasonico-hcsr04-e-buzzer-5v>

4.3. Modo de teste

Para os testes, além dos sensores foi utilizada uma tela LCD, para podermos analisar melhor os dados colhidos pelos sensores. Primeiro foi feito o teste separadamente com apenas um sensor IR, depois adicionamos os outros quatro, em seguida foi feito o teste para o sensor de ultrassom e, por fim, com todos os seis sensores juntos.

Para testar o funcionamento de sensores de IR foi utilizado um trecho da pista pré-montada fornecida pelos professores. Assim, passamos o sensor por cima da parte branca e por cima do trecho a ser percorrido para observar o sensor diferenciando o trecho que pertence à pista do trecho não pertencente. Assim, foi observado que os sensores de IR estão funcionando como esperado.

O sensor de ultrassom foi testado ao aproximar diversos objetos como mãos e livros e, a distância desses objetos até o sensor era mostrada no visor LCD. A distância mostrada no visor estava dentro do esperado, portanto, assim como os de IR os sensores de ultrassom estão funcionando adequadamente.

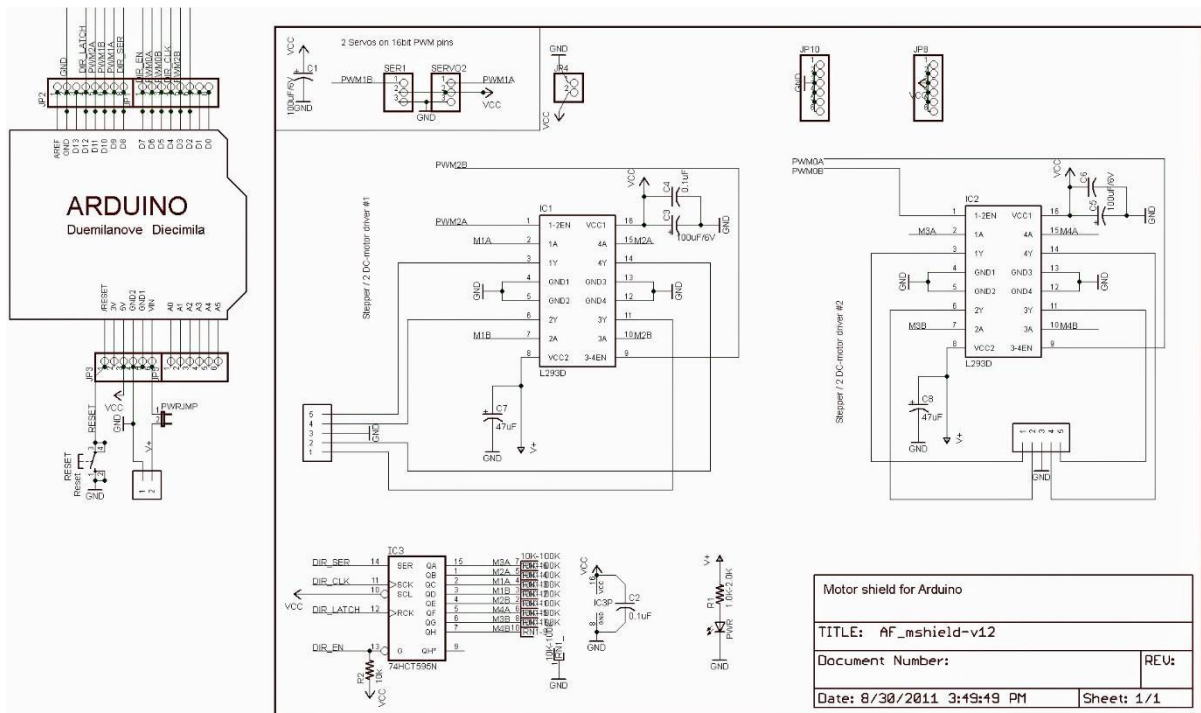
5. ATUADORES

Além dos sensores serão necessários os atuadores (motores) que são responsáveis pelo movimento do robô. Para a montagem do robô serão utilizados dois motores e, como o Arduino não possui a potência necessária para a ativação dos dois, utilizaremos um shield (peça acoplada acima do Arduino) com uma ponte H controlado por um sinal de PWM

5.1. Ponte H

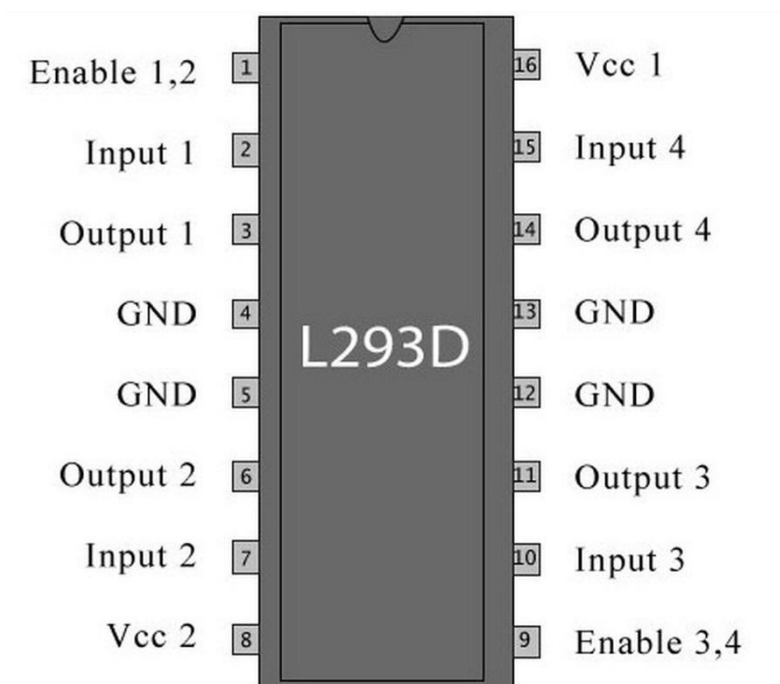
O microcontrolador Atmega 328P, principal componente do Arduino, pode receber e enviar correntes de até 40 mA, valor muito baixo para o funcionamento de um motor, por isso é preciso de um driver exterior para que estes possam ser controlados. Este driver exterior é o circuito integrado L293D que foi utilizado através de um shield chamado *adafruit motor shield v1*. O datasheet do shield pode ser visto na figura 7 e o da ponte H na imagem 8.(RAMBO, Wagner, 2016)

Figura 7 - Datasheet do adafruit motor shield v1



Fonte: <<https://www.filipeflop.com/blog/motor-shield-l293d-motor-dc-arduino/>>

6.

Figura 8 - *Datasheet* do circuito integrado L293D

Fonte: < <https://www.buyapi.ca/product/l293d-dual-h-bridge-ic/> >

A ponte H é basicamente um circuito com 4 chaves cujos estados serão controlados pelo Arduino. Quando duas chaves estão fechadas e as outras duas abertas o motor girará em um sentido, caso as chaves fechadas e abertas troquem de estado, o sentido da rotação do motor também é alterado (ALEXANDRE, Saulo, 2017).

5.2 Sinal PWM

A ponte H é alimentada pelo Arduino e passa essa energia para os motores. Essa alimentação inicial se dá por meio de um sinal PWM.

O sinal PWM é uma forma de diminuir a tensão total no circuito sem alterar suas características. Seu funcionamento se dá a partir da abertura e do fechamento de chaves, no caso as chaves da ponte H já abordadas previamente, quando a chave está fechada o circuito também estará fechado, logo a tensão máxima do gerador será aplicada, porém, quando a chave está aberta, o circuito, por consequência, também está aberto, assim a tensão fornecida pelo gerador é nula.

Dessa forma, variando o tempo em que a chave fica aberta e fechada, é possível que a tensão resultante aplicada seja uma tensão intermediária entre a máxima e a zero. A tensão média pode ser calculada de acordo com a equação 2.

$$V_{\text{média}} = V_{\text{máxima}} \times \frac{\text{Tempo}_{\text{ligado}}}{\text{Período}} \times 100$$

(2)

Em que $V_{\text{média}}$ é a tensão intermediária que foi aplicada, $V_{\text{máxima}}$ é a tensão máxima (aquela aplicada pelo microcontrolador), $\text{Tempo}_{\text{ligado}}$ é o tempo em que a tensão foi aplicada e período é a soma do $\text{Tempo}_{\text{ligado}}$ com o tempo em que a tensão não foi aplicada.

Desta forma, variando a tensão aplicada aos motores, a potência de giro deles é variada, portanto desta forma podemos controlar a velocidade de cada motor a partir do sinal PWM. (Meca web, 2019)

5.3 Modo de teste

Para testar o funcionamento dos motores montamos o protótipo e tentamos variar a velocidade de cada motor a partir da programação. No teste observamos os motores girando em cada sentido com velocidades diferentes.

Percebemos com o teste que um dos motores possui uma velocidade um pouco maior que a do outro se colocados com os mesmos parâmetros na programação. Por isso, para o projeto final isso será corrigido fazendo com que a velocidade do que é, por padrão, mais rápido seja um pouco menor.

6. CONCLUSAO

Após a montagem do robô foram feitos os testes na pista, em geral o percurso foi concluído de forma esperada, porém, encontramos algumas dificuldades para fazer com que ele desvie dos obstáculos de forma eficiente, isso aconteceu pois ao retornar para a pista o carrinho precisava fazer uma curva muito acentuada e os motores não foram capazes de proporcionar isso.

Em busca de contornar estes problemas foram feitas mudanças na programação, para que o robô andasse uma distancia menor fora da pista e para diminuir o delay entre a leitura dos sensores e a resposta dos motores, porém ainda assim o erro persistiu.

Portanto, a montagem final do robô foi essa em que ele não pode ultrapassar o obstáculo, porém acreditamos que em outros percursos, nos quais o obstáculo não esteja tão próximo a uma curva, o carrinho poderá desviar facilmente deste, além de passar pelas outras dificuldades: as curvas, os redutores de velocidades, limitadores de tamanho, gaps e a rampa.

7.

REFERÊNCIAS

ALEXANDRE, Saulo. Aprenda a utilizar a ponte H L293D com Arduino. Disponível em: <<https://autocorerobotica.blog.br/aprenda-utilizar-ponte-h-l293d-com-arduino/>>. Acesso em: 24 de out. 2019.

BOSTON DYNAMICS. Atlas, the World's Most Dynamic Humanoid. Disponível em: <<https://www.bostondynamics.com/atlas.>> Acesso em: 17 de ago. 2019.

Equipe Robótica e Eletrônica. 16 de abr. 2014. O que é robótica? Disponível em <<http://www.joseferreira.com.br/blogs/robotica/arobotica/oqueerobotica/>>. Acesso em: 10 de ago. 2019.

ETECNOPHILES. Beginners guide to IR sensor and how to use it with Arduino. Disponível em: <<https://etechnophiles.com/beginners-guide-to-ir-sensor-and-how-to-use-it-with-arduino/>>. Acesso em: 27 de set. 2019.

Silva, Carlos L. A. da. 4 de ago. 2017. 10 Robôs do passado. Disponível em:<<https://www.codigofonte.com.br/artigos/10-robos-do-passado>>. Acesso em: 9 de ago.2019.

MECA WEB. PWM: modulação por largura de pulso. Disponível em: <http://www.mecaweb.com.br/eletronica/content/e_pwm>. Acesso em: 25 out. 2019.

MOTA, Allan. HC-SR04 – Sensor Ultrassônico de distância com Arduino. Disponível em: <<https://portal.vidadesilicio.com.br/hc-sr04-sensor-ultrassonico/>>. Acesso em: 27 de set. 2019.

OLIMPIÁDA BRASILEIRA DE ROBÓTICA. Por que uma Olimpíada de Robótica? Disponível em: <<http://www.obr.org.br/por-que-uma-olimpiada-robotica/>>. Acesso em: 27 de set. 2019.

RAMBO, Wagner. 12 de maio 2016. Vídeo: Tutorial Motor Shield L293D e Arduino. Disponível em: <<https://www.filipeflop.com/blog/motor-shield-l293d-motor-dc-arduino/>>. Acesso em 24 de out. 2019.

APÊNDICES

APÊNDICE A: CÓDIGO PARA TESTE DOS SENSORES

```
// inclui a biblioteca do display

#include <LiquidCrystal.h>

//seleciona os pinos que receberão dados dos sensores IR

int S1 = A0;

int S2 = A1;

int S3 = A2;

int S4 = A3;

int S5 = A4;

//seleciona os pinos que receberão e fornecerão os dados para o sensor de ultrassom

const int trigPin = 9;

const int echoPin = 10;

//cria as variaveis tempo para a recepção da onda e distancia do obstaculo

long tempo;

long distancia;

//seleciona os pinos do monitor LCD

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

    // seleciona o numero de colunas e linhas do LCD

    lcd.begin(16, 2);

    //caracteriza cada pino como input ou output

    pinMode(S1, INPUT);

    pinMode(S2, INPUT);

    pinMode(S3, INPUT);

    pinMode(S4, INPUT);

    pinMode(S5, INPUT);
```

```

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

// Separa o LCD para mostrar o valor recebido de cada sensor

lcd.print("S1 S2 S3 S4 S5 US");

}

void loop() {

  //Sensor 1

  int detect1 = digitalRead(S1);

  if (detect1 == LOW) {

    lcd.setCursor(0, 1);

    lcd.print("1");

  }

  else {

    lcd.setCursor(0, 1);

    lcd.print("0");

  }

  //Sensor 2

  int detect2 = digitalRead(S2);

  if (detect2 == LOW) {

    lcd.setCursor(3, 1);

    lcd.print("1");

  }

  else {

    lcd.setCursor(3, 1);

    lcd.print("0");

  }

  //Sensor 3

```

```
int detect3 = digitalRead(S3);  
if (detect3 == LOW) {  
    lcd.setCursor(6, 1);  
    lcd.print("1");  
}  
else {  
    lcd.setCursor(6, 1);  
    lcd.print("0");  
}  
  
//Sensor 4  
int detect4 = digitalRead(S4);  
if (detect4 == LOW) {  
    lcd.setCursor(9, 1);  
    lcd.print("1");  
}  
else {  
    lcd.setCursor(9, 1);  
    lcd.print("0");  
}  
  
//Sensor 5  
int detect5 = digitalRead(S5);  
if (detect5 == LOW) {  
    lcd.setCursor(12, 1);  
    lcd.print("1");  
}  
else {  
    lcd.setCursor(12, 1);
```



```
    lcd.print("0");  
}  
  
//Ultrassom  
  
digitalWrite(trigPin, HIGH);  
  
delay(10);  
  
digitalWrite(trigPin, LOW);  
  
//Le o echoPin e retorna o tempo da onda  
tempo = pulseIn(echoPin, HIGH);  
  
//calcula a distancia  
distancia = (0.034 * tempo)/2;  
  
//Imprime a distancia no LCD  
lcd.setCursor(14, 1);  
lcd.print(distancia);  
lcd.print(" ");  
delay(250);  
}
```

APÊNDICE B: CÓDIGO PARA TESTE DOS MOTORES

```
//adicionando bibliotecas

#include <AFMotor.h>

//Configuração do shield

AF_DCMotor motor1(1, MOTOR34_64KHZ);
AF_DCMotor motor2(2, MOTOR34_64KHZ);
AF_DCMotor motor3(3, MOTOR34_64KHZ);
AF_DCMotor motor4(4, MOTOR34_64KHZ);

//variaveis

int i;

int tempo = 10;

void setup(){

    teste_motores;

}

void loop(){

}

void teste_motores(){

    Serial.println("Testando Motores");

    i = 30;
```

```
motor1.setSpeed(i);
```

```
motor2.setSpeed(i);
```

```
motor3.setSpeed(i);
```

```
motor4.setSpeed(i);
```

```
motor1.run(FORWARD);
```

```
motor2.run(FORWARD);
```

```
motor3.run(FORWARD);
```

```
motor4.run(FORWARD);
```

```
delay(2000);
```

```
for (i=30; i<255; i++) {
```

```
    motor1.setSpeed(i);
```

```
    motor2.setSpeed(i);
```

```
    motor3.setSpeed(i);
```

```
    motor4.setSpeed(i);
```

```
    Serial.print("Velocidade: ");
```

```
    Serial.println(i-30);
```

```
    delay(tempo);
```

```
}
```

```
for (i=255; i!=0; i--) {
```

```
    motor1.setSpeed(i);
```

```
    motor2.setSpeed(i);
```

```
    motor3.setSpeed(i);
```

```
    motor4.setSpeed(i);
```

```
Serial.print("Velocidade: ");  
Serial.println(i);  
delay(tempo);  
}  
delay(1000);  
}
```