



Projeto de robô automatizado

Carlos Eduardo de Carvalho Teixeira¹ – carlos-edush5@outlook.com

Jhonatan Sanga de Freitas¹ – jhonatansanga05@outlook.com

Vinício Lopes Teixeira¹ – vinicioteixass144@outlook.com

Lucas Ferreira de Souza¹ – lucasferdesouza1@gmail.com

1 – Instituto Federal do Paraná – IFPR Paranavaí

Resumo: Os equipamentos de locomoção, atualmente, estão em processo de automatização e existem vários modelos autônomos. Grandes empresas de tecnologia, como por exemplo, o Google, estão investindo neste processo. Neste contexto, o objetivo deste projeto é programar e construir um robô que se locomove e desvia de objetos. Para tal, foi montado uma base robótica com um sensor ultrassônico hc-sr04, um Arduino Uno, um Shield Motor L293D Driver Ponte H, um Micro servo Motor, uma Mini protoboard e uma Bateria 9v. Além disso, foi elaborada um código de programação em linguagem C, compilado e transferido para arduino, via cabo usb (porta serial), na interface IDE Arduino. Contudo, o protótipo obteve os resultados esperados, isto é, locomoveu-se identificando e desviando dos objetos.

Palavras-chave: Arduino; servo motor; Ponte H L293D; sensor ultrassônico hc-sr04.

1. Introdução

O presente trabalho é sobre o desenvolvimento de um robô detector de obstáculos, mais precisamente o projeto visa um aprendizado de programação e prática.

O objetivo deste projeto é desenvolver uma programação no programa arduino, que efetue a função de detecção de obstáculos através de um sensor ultrassônico fixado em um chassi de robô. A mesma se encontra organizado em tópicos, em que neles estão explicando como fomos desenvolvendo o projeto.

A metodologia pela qual utilizamos foram livros, pesquisa em diversas bases dados, e o auxílio em todo o projeto do professor.

2. Desenvolvimento do Projeto

Ao iniciar o projeto, foram feitas

diversas pesquisas sobre o assunto, em livros como: ARDUINO BÁSICO, McRoberts Michael, 2010, onde utilizamos, o Capítulo 4, Sonorizadores e sensores simples, e Capítulo 5, Controlando um motor CC.

Realizou-se pesquisa em diferentes base de dados, onde descobriu-se modelos de robôs já construídos, e com isso nos possibilitou ter um panorama do que foi realizado.

O objetivo deste trabalho é desenvolver uma programação no arduino capaz de executar uma função de detecção de obstáculos e ao mesmo tempo estar em movimento, e com esse pensamento fizemos alguns testes com algumas programações. No final chegamos na programação mais adequada, que por sua vez conseguiu realizar essa função.

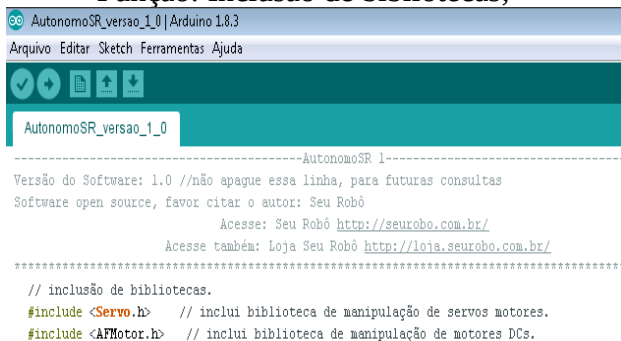
Mesmo com todos os componentes necessários para realizar o projeto, encontramos problemas mecânicos nos motores DC, e após testes vimos que a bateria, não estava adequada ou sua tensão não era o suficiente para alimentar o projeto em si. Os componentes necessários foram: Chassi de robô com dois motores DC de 6v, Arduino Uno, Shield Motor L293D Driver Ponte H, Sensor ultrassônico HC-SR04, Micro servo Motor, Mini protoboard, Bateria 9v.

Cada código de programação em C tem sua respectiva função, que mais tarde, é compilado e executado pela plataforma Arduino. Abaixo é possível conferir as figuras contendo a programação em linguagem C com as respectivas funções, separadas, da figura 1 à 17.

As figuras 18, 19 e 20 faz referência a

parte física construída no projeto. Nela é possível ver Chassi de robô com dois motores DC de 6v, Arduino Uno, Shield Motor L293D Driver Ponte H, Sensor ultrassônico HC-SR04, Micro servo Motor, Mini protoboard, Bateria 9v. Em seguida, na figura 1, temos o diagrama elétrico do projeto, onde temos as conexões dos diferentes dispositivos do projeto.

Figura 1: Primeira parte da Programação.
Função: Inclusão de bibliotecas;



```
AutonomoSR_versao_1_0 | Arduino 1.8.3
Arquivo Editar Sketch Ferramentas Ajuda

AutonomoSR_versao_1_0

-----AutonomoSR 1-----
Versão do Software: 1.0 //não apague essa linha, para futuras consultas
Software open source, favor citar o autor: Seu Robô
Acesse: Seu Robô http://seurobo.com.br/
Acesse também: Loja Seu Robô http://loja.seurobo.com.br/
*****

// inclusão de bibliotecas.
#include <Servo.h> // inclui biblioteca de manipulação de servos motores.
#include <AFMotor.h> // inclui biblioteca de manipulação de motores DCs.
```

Fonte: do Autor, 2018.

Figura 2: Definindo os pinos;

```
//Definindo os pinos
#define trigPin A0 //Pino TRIG do sensor no pino analógico A0
#define echoPin A1 //Pino ECHO do sensor no pino analógico A1
#define BUZZER A2 // Define o pino do buzzer (Som) no pino ANALÓGICO A0
AF_DCMotor motor1(1); // Define o motor1 ligado ao M1
AF_DCMotor motor2(2); // Define o motor2 ligado ao M2

int TempoGirar = 1; //esse é o tempo para o robô girar em 45° com uma bateria de 9v.
int distanciaObstaculo = 30; //distância para o robô parar e recalculer o melhor caminho
int velocidadeMotores = 80; // velocidade que os motores funcionarão na bateria 9v. Para a bat
Servo servo_ultra_sonico; // nomeando o servo motor
//variáveis para o sensor ultrassonico
long duracao;
long distancia_cm=0;
int minimumRange=5; //tempo de resposta do sensor
int maximumRange=200;
```

Fonte: do Autor, 2018.

Figura 3: Executado na inicialização do
Arduino;

```
// executado na inicialização do Arduino
void setup(){
  Serial.begin(9600); // inicializa a comunicação serial para mostrar dados
  servo_ultra_sonico.attach(10); // Define o mini servo motor ligado no pino digi
  pinMode(trigPin, OUTPUT); //define o pino TRIG como saída
  pinMode(echoPin, INPUT); //define o pino ECHO como entrada
  pinMode(BUZZER,OUTPUT); // Define o pino do buzzer como saída
  motor1.setSpeed(velocidadeMotores); // Define a velocidade para os motores.
  motor2.setSpeed(velocidadeMotores); //Usamos uma bateria de 9v 450mAh, com e
  servo_ultra_sonico.write(90); // O servo do sensor se inicia a 90 graus (meio)
  rotacao_Parado; //inicia com os motores parados
}
```

Fonte: do Autor, 2018.

Figura 4: Função principal do Arduino;

```
// Função principal do Arduino
void loop(){
  pensar(); //inicia a função pensar
}
```

Fonte: do Autor, 2018.

Figura 5: Função para chamar outras funções
e definir o que o robô fará;

```
// Função para chamar outras funções e definir o que o robô fará
void pensar(){
  reposicionaServoSonar(); //Coloca o servo para olhar a frente
  int distancia = lerSonar(); // Ler o sensor de distância
  Serial.print("distancia em cm: ");
  Serial.println(distancia); // Exibe no serial monitor
  if (distancia > distanciaObstaculo) { // Se a distância for maior que 20 cm
    rotacao_Frente(); //robô anda para frente
  }else{
    rotacao_Parado(); //para o robô
    posicionaCarroMelhorCaminho(); //calcula o melhor caminho
    pensar();
  }
}
```

Fonte: do Autor, 2018.

Figura 6: Função para ler e calcular a
distância do sensor ultrassônico;

```
// Função para ler e calcular a distância do sensor ultrassônico
int lerSonar(){
  digitalWrite(trigPin, LOW); //não envia som
  delayMicroseconds(2);
  digitalWrite(trigPin,HIGH); //envia som
  delayMicroseconds(10);
  digitalWrite(trigPin,LOW); //não envia o som e espera o retorno do som enviado
  duracao = pulseIn(echoPin,HIGH); //Captura a duração em tempo do retorno do som.
  distancia_cm = duracao/56; //Calcula a distância
  delay(30);
  return distancia_cm; // Retorna a distância
}
```

Fonte: do Autor, 2018.

Figura 7: Função para calcular a distância do
centro;

```
// Função para calcular a distância do centro
int calcularDistanciaCentro(){
  servo_ultra_sonico.write(90);
  delay(20);
  int leituraDoSonar = lerSonar(); // Ler sensor de distância
  delay(500);
  leituraDoSonar = lerSonar();
  delay(500);
  Serial.print("Distancia do Centro: "); // Exibe no serial
  Serial.println(leituraDoSonar);
  return leituraDoSonar; // Retorna a distância
}
```

Fonte: do Autor, 2018.

Figura 8: Função para calcular a distância da direita;

```
// Função para calcular a distância da direita
int calcularDistanciaDireita(){
    servo_ultra_sonico.write(0);
    delay(200);
    int leituraDoSonar = lerSonar();
    delay(500);
    leituraDoSonar = lerSonar();
    delay(500);
    Serial.print("Distancia da Direita: ");
    Serial.println(leituraDoSonar);
    return leituraDoSonar;
}
```

Fonte: do Autor, 2018.

Figura 9: Função para calcular a distância da esquerda;

```
// Função para calcular a distância da esquerda
int calcularDistanciaEsquerda(){
    servo_ultra_sonico.write(180);
    delay(200);
    int leituraDoSonar = lerSonar();
    delay(500);
    leituraDoSonar = lerSonar();
    delay(500);
    Serial.print("Distancia Esquerda: ");
    Serial.println(leituraDoSonar);
    return leituraDoSonar;
}
```

Fonte: do Autor, 2018.

Figura 10: Função para captar as distâncias lidas e calcular a melhor distância;

```
// Função para captar as distâncias lidas e calcular a melhor distância. Acesso: Set
char calculaMelhorDistancia(){
    int esquerda = calcularDistanciaEsquerda();
    int centro = calcularDistanciaCentro();
    int direita = calcularDistanciaDireita();
    reposicionaServoSonar();
    int maiorDistancia = 0;
    char melhorDistancia = '0';

    if (centro > direita && centro > esquerda){
        melhorDistancia = 'c';
        maiorDistancia = centro;
    }else
    if (direita > centro && direita > esquerda){
        melhorDistancia = 'd';
        maiorDistancia = direita;
    }else
    if (esquerda > centro && esquerda > direita){
        melhorDistancia = 'e';
        maiorDistancia = esquerda;
    }
    if (maiorDistancia <= distanciaObstaculo) { //distância limite para parar o robô
        rotacao_Re();
        posicionaCarroMelhorCaminho();
    }

    reposicionaServoSonar();
    return melhorDistancia;
}
```

Fonte: do Autor, 2018.

Figura 11: Função para colocar o carrinho na melhor distância, isto é, girá-lo para a melhor distância;

```
// Função para colocar o carrinho na melhor distância, isto é, girá-lo para a melhor distância
void posicionaCarroMelhorCaminho(){
    char melhorDist = calculaMelhorDistancia();
    Serial.print("melhor Distancia em cm: ");
    Serial.println(melhorDist);
    if (melhorDist == 'c'){
        pensar();
    }else if (melhorDist == 'd'){
        rotacao_Direita();
    }else if (melhorDist == 'e'){
        rotacao_Esquerda();
    }else{
        rotacao_Re();
    }
    reposicionaServoSonar();
}
```

Fonte: do Autor, 2018.

Figura 12: Função para deixar o sensor "olho" do robô no centro;

```
// Função para deixar o sensor "olho" do robô no centro
void reposicionaServoSonar(){
    servo_ultra_sonico.write(90);
    delay(200);
}
```

Fonte: do Autor, 2018.

Figura 13: Função para fazer o carro parar;

```
// Função para fazer o carro parar
void rotacao_Parado()
{
    Serial.println(" Motor: Parar ");
    motor1.run(RELEASE); // Motor para
    motor2.run(RELEASE);
}
```

Fonte: do Autor, 2018.

Figura 14: Função para fazer o robô andar para frente;

```
// Função para fazer o robô andar para frente
void rotacao_Frente()
{
    Serial.println("Motor: Frente ");
    motor1.run(FORWARD); // Roda vai para frente
    motor2.run(FORWARD);
    delay(50);
}
```

Fonte: do Autor, 2018.

Figura 15: Função que faz o robô andar para trás e emite som quando ele dá ré;

```
// Função que faz o robô andar para trás e emite som quando ele dá ré
void rotacao_Re()
{
  Serial.println("Motor: ré ");
  for (int i=0; i <= 3; i++){
    digitalWrite(BUZZER, HIGH); // Liga o som
    delay(100);
    motor1.run(BACKWARD); // Roda vai para trás
    motor2.run(BACKWARD); // Roda vai para trás
    delay(100);
    digitalWrite(BUZZER, LOW); // Desliga o som
    delay(100);
  }
  rotacao_Parado();
}
```

Fonte: do Autor, 2018.

Figura 16: Função que faz o robô virar à direita;

```
// Função que faz o robô virar à direita
void rotacao_Direita()
{
  digitalWrite(BUZZER, HIGH); // Liga o s
  delay(100);
  motor1.run(BACKWARD); //o robô dá um
  motor2.run(BACKWARD);
  delay(50);
  digitalWrite(BUZZER, LOW); // Desliga o
  delay(100);
  Serial.println(" Para a direita ");
  motor1.run(FORWARD); // Roda vai para f
  motor2.run(BACKWARD); // Roda vai para
  delay(TempoGirar);
} .
```

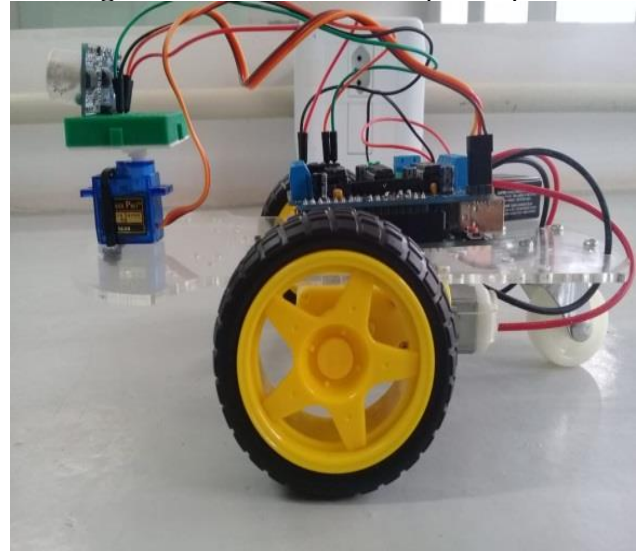
Fonte: do Autor, 2018.

Figura 17: Função que faz o robô virar à esquerda;

```
// Função que faz o robô virar à esquerda
void rotacao_Esquerda()
{
  digitalWrite(BUZZER, HIGH); // Liga o som
  delay(100);
  motor1.run(BACKWARD); // // O robô dá uma ré para não colidir ao girar
  motor2.run(BACKWARD);
  delay(50);
  digitalWrite(BUZZER, LOW); // Desliga o som
  delay(100);
  Serial.println(" Para a esquerda ");
  motor1.run(BACKWARD); // Roda vai para trás
  motor2.run(FORWARD); // Roda vai para frente
  delay(TempoGirar);
}
```

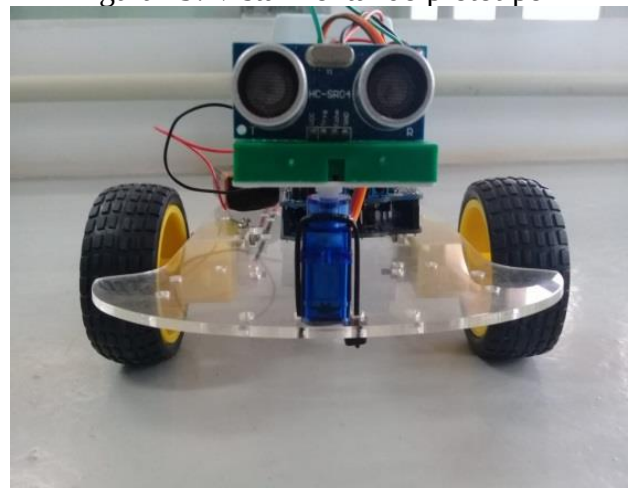
Fonte: do Autor, 2018.

Figura 18: Vista Lateral do protótipo



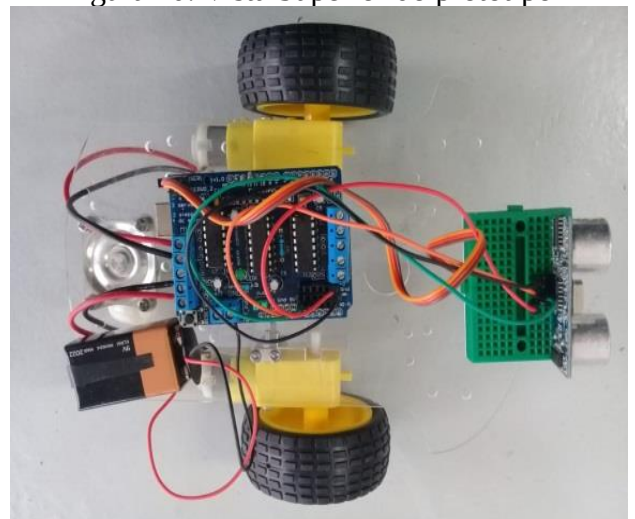
Fonte: do Autor, 2018.

Figura 19: Vista Frontal do protótipo



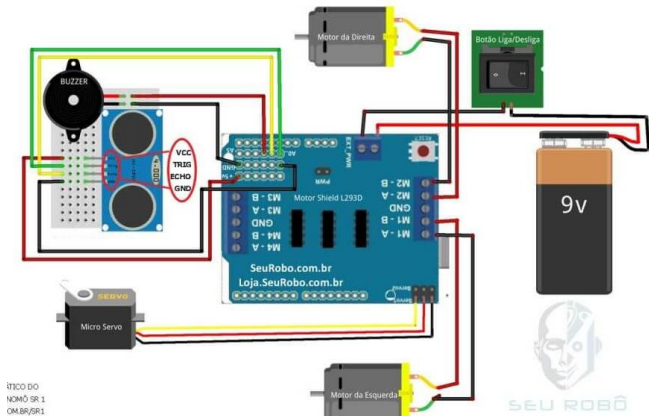
Fonte: do Autor, 2018.

Figura 20: Vista Superior do protótipo



Fonte: do Autor, 2018.

Figura 21: Diagrama do sistema de controle do protótipo;



Fonte: Retirado de LEONARDO CAMPBELL, 2018.

complexo, permitindo compreender melhor a parte de programação e execução de um projeto.

REFERÊNCIAS

MCROBERTS, Michael. **Arduino Básico**. 1. ed. São Paulo-SP, 2011.

LEONARDO CAMPBELL. **Robô Autônomo SR 1: o robô autônomo Simples com arduino**. Disponível em <<https://seurobo.com.br/robo-autonomosr-1-o-robo-autonomo-simples-com-arduino/>>. Acesso em 27 abr. 2018.

3. Conclusão

Neste projeto abordamos a realização de um robô detector de obstáculos por meio de um sensor ultrassônico e concluímos que, foi possível apesar das dificuldades encontradas, desenvolver o protótipo, e com isso, aprender um pouco mais sobre programação.

Todos os objetivos que tínhamos propostos foram cumpridos, o que nos ajuda a progredir, principalmente no que se refere ao que foi agregado ao nosso conhecimento, pois nos aprofundamos em algo um pouco mais