

CENTRO UNIVERSITARIO RUY BARBOSA  
WYDEN

## Projeto microcontroladores

Microcontrollers project

Jardel Souza Alves  
João Moraes da Silva  
José Nilson Oliveira Arcanjo  
Monique Salles  
Vinicius Samuel Nascimento Bomfim

## Projeto Microcontroladores

Trabalho acadêmico para Conclusão do Semestre  
apresentado ao professor Heleno Cardoso como  
requisito necessário para obtenção da nota para  
aprovação na matéria Microcontroladores.

## LISTA DE ILUSTRAÇÕES

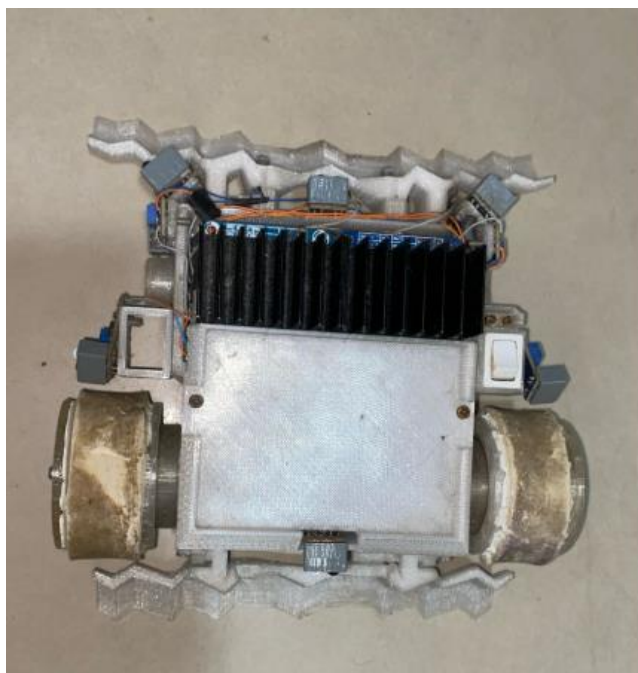
SOBRE.....	5
VISÃO ESQUEMÁTICA.....	11

## SUMÁRIO

SOBRE.....	5
COMPONENTES.....	6
CÓDIGO.....	7

## SOBRE

Robôs de sumo são uma boa prática à robótica, eletrônica, programação e mecânica. Com a utilização de microcontroladores os robôs se tornam totalmente autônomos, com a utilização de sensores, o que significa que não são controlados remotamente durante a luta, mas sim precisam tomar suas próprias decisões baseadas em seus sensores e programação.



## COMPONENTES

- Controlador:
  - Mega2560 R3 + Wifi R3 CH340G USB-TTL
  - ATMEGA2560 ESP8266 32Mb
- 2 motores DC 12V.
- 8 sensores ópticos.
- 2 módulos Driver Ponte H 43a Bts7960.
- 1 interruptor liga/desliga.
- 14 parafusos.
  - Fixação dos sensores, rodas impressas, módulos Drivers, controladores ATmega.
- Fios.
- Bateria 14v de drone.
- Filamento Petg
  - Impressão 3d da carcaça, engrenagens, planetárias de tração das rodas.
- Silicone e amido de milho
  - Rodas emborrachadas para atritar melhor ao solo.

## CÓDIGO

```
#define Sensor_Frente 52 #define Sensor_Direita_Frente 46
#define Sensor_Esquerda_Frente 47 #define Sensor_Direita 30
#define Sensor_Esquerda 31 #define Sensor_Tras 24
#define Sensor_Seguidor_Direita 40 #define Sensor_Seguidor_Esquerda 41
#define Motor_Direita_Frente_PWM 6 #define Motor_Direita_Tras_PWM 5
#define Motor_Esquerda_Frente_PWM 8 #define Motor_Esquerda_Tras_PWM 9

int Motores_Ligados(10);

void setup() {
  ////////////sensores de entrada////////////////////
  pinMode(Sensor_Frente, INPUT); pinMode(Sensor_Direita_Frente, INPUT);
  pinMode(Sensor_Esquerda_Frente, INPUT); pinMode(Sensor_Direita, INPUT);
  pinMode(Sensor_Esquerda, INPUT); pinMode(Sensor_Tras, INPUT);
  pinMode(Sensor_Seguidor_Direita, INPUT); pinMode(Sensor_Seguidor_Esquerda, INPUT);

  ////////////motores de saida////////////////////
  pinMode(Motor_Direita_Frente_PWM, OUTPUT); pinMode(Motor_Direita_Tras_PWM,
  OUTPUT); pinMode(Motor_Esquerda_Frente_PWM, OUTPUT);
  pinMode(Motor_Esquerda_Tras_PWM, OUTPUT);

  ////////////Set motores On////////////////////
  pinMode(Motores_Ligados, OUTPUT);

  /* ////////////Setando saidas PWM para 0////////////////////
  analogWrite(Motor_Direita_Frente_PWM, 0); analogWrite(Motor_Direita_Tras_PWM, 0);
  analogWrite(Motor_Esquerda_Frente_PWM, 0); analogWrite(Motor_Esquerda_Tras_PWM,
  0);

  */ ////////////tempo antes de iniciar o combate/////
  delay(5000); digitalWrite(Motores_Ligados, true);
}
```

```

void loop() {

//////////Lendo valores dos sensores//////////
bool estado_sensor_frente = digitalRead(Sensor_Frente); bool
    estado_sensor_direita_frente = digitalRead(Sensor_Direita_Frente); bool
estado_sensor_esquerda_frente = digitalRead(Sensor_Esquerda_Frente); bool
estado_sensor_direita = digitalRead(Sensor_Frente); bool estado_sensor_esquerda =
digitalRead(Sensor_Esquerda); bool estado_sensor_tras = digitalRead(Sensor_Tras); bool
estado_sensor_seguidor_direita = digitalRead(Sensor_Seguidor_Direita); bool
estado_sensor_seguidor_esquerda = digitalRead(Sensor_Seguidor_Esquerda);

//////////Se sensores seguidores de linha estiverem OK, ande para frente//////////
if(!estado_sensor_seguidor_direita & !estado_sensor_seguidor_esquerda){
//////////ande para frente////////////////////////////////////////
    analogWrite(Motor_Esquerda_Frente_PWM, 200);
    analogWrite(Motor_Direita_Frente_PWM, 250);
    delay (10);
}

//////////Se andando para frente o da direita encostar no limite, dê ré e vire para
esquerda//////////
if(estado_sensor_seguidor_direita & !estado_sensor_seguidor_esquerda){
//////////pare ambos os motores antes de startar algo//////////
    analogWrite(Motor_Direita_Frente_PWM, 0);
    analogWrite(Motor_Esquerda_Frente_PWM, 0);
    delay(50);
//////////ando meio segundo para tras////////////////////////////////////////
    analogWrite(Motor_Direita_Tras_PWM, 250);
    analogWrite(Motor_Esquerda_Tras_PWM, 200);
    delay(200);

//////////pare ambos os motores antes de startar algo//////////
    analogWrite(Motor_Direita_Tras_PWM, 0);
    analogWrite(Motor_Esquerda_Tras_PWM, 0);
    delay(50);

//////////faça curva 45° graus para esquerda////////////////////////////////////////
    analogWrite(Motor_Esquerda_Tras_PWM, 200);

```



```

analogWrite(Motor_Direita_Frente_PWM, 250);
delay(850);

//////////pare ambos os motores antes de startar algo//////////
analogWrite(Motor_Direita_Frente_PWM, 0);
analogWrite(Motor_Esquerda_Tras_PWM, 0);
delay(50);

}

//////////Se andando para frente o da esquerda encostar no limite, dê ré e vire para
direita//////////
if(!estado_sensor_seguidor_direita & estado_sensor_seguidor_esquerda){

//////////pare ambos os motores antes de startar algo//////////
analogWrite(Motor_Direita_Frente_PWM, 0);
analogWrite(Motor_Esquerda_Frente_PWM, 0);
delay(50);

//////////ando 200 m-segundo para tras//////////
analogWrite(Motor_Esquerda_Tras_PWM, 200);
analogWrite(Motor_Direita_Tras_PWM, 250);
delay(200);

//////////pare ambos os motores antes de startar algo//////////
analogWrite(Motor_Direita_Tras_PWM, 0);
analogWrite(Motor_Esquerda_Tras_PWM, 0);
delay(50);

//////////faça curva 45° graus para esquerda//////////
analogWrite(Motor_Direita_Tras_PWM, 250);
analogWrite(Motor_Esquerda_Frente_PWM, 200);
delay(850);

//////////pare ambos os motores antes de startar algo//////////
analogWrite(Motor_Esquerda_Frente_PWM, 0);
analogWrite(Motor_Direita_Tras_PWM, 0);
delay(50);

}

```

```

/////Se encontrar algo no sensor de trás, empurre para trás, até que algum sensor de linha seja
atuado/////
if(!estado_sensor_tras & !estado_sensor_seguidor_direita &
!estado_sensor_seguidor_esquerda){
//////////pare ambos os motores//////////
    analogWrite(Motor_Esquerda_Frente_PWM, 0);
    analogWrite(Motor_Direita_Frente_PWM, 0);
    delay(50);

    bool memoria = 1;
    switch(memoria){

        case 1 :
            ////////////ande para tras//////////
                analogWrite(Motor_Direita_Tras_PWM, 250);
                analogWrite(Motor_Esquerda_Tras_PWM, 200);
                return 1;
            }
        }
    }
}

```

## VISÃO ESQUEMATICA

