

```

/*-----*
-----
PROJETO PRIMEIRO SEMESTRE CIENCIAS DA COMPUTAÇÃO
-----*/
//PORTAS ENTRADA E SAIDA DE DADOS

//DEFININDO AS PORTAS DOS MOTORES LADO ESQUERDO
int PWM_M1 = 9 ;
int IN1_M1 = 31; //motor dianteiro
int IN2_M1 = 33;

int PWM_M2 = 11 ;
int IN1_M2 = 30 ; //motor traseiro
int IN2_M2 = 32;

//DEFININDO AS PORTAS DOS MOTORES LADO DIREITO
int PWM_M3 = 5;
int IN1_M3 = 35 ; //motor dianteiro
int IN2_M3 = 36;

int PWM_M4 = 3 ;
int IN1_M4 = 37 ; //motor traseiro
int IN2_M4 = 38;

//DEFININDO POTENCIOMETROS
int POT1 = A0;
int POT2 = A1;

//DEFININDO LEDS
int LEDSESQUERDA = 22;
int LEDSDIREITA = 24;
int LEDFAROL = 26;
;int LEDPARE = 28;

//DEFININDO SENSORES ULTRASSONICOS

//SENSOR FRONTAL-----
int TRIGF = 11;
int ECHOF = 10;
//SENSOR ESQUERDA-----
int TRIGE = 11;
int ECHOE = 10;
//SENSOR DIREITA-----
int TRIGD = 11;
int ECHOD = 10;

int INTERRUPTOR = 40; //BOTAO MODO AUTOMATICO E CONTROLE

```

```
//DEFININDO VARIÁVEIS
```

```
int estPot1, estPot2; //variáveis do potenciômetro  
int SWITCH;
```

```
//sensores ultrassônicos
```

```
unsigned long tempo1;
```

```
double distancia1;
```

```
unsigned long tempo2;
```

```
double distancia2;
```

```
unsigned long tempo3;
```

```
double distancia3;
```

```
void setup() {
```

```
    pinMode(PWM_M1, OUTPUT); //pwm motor dianteiro esquerdo
```

```
    pinMode(PWM_M2, OUTPUT); //pwm motor traseiro esquerdo
```

```
    pinMode(IN1_M1, OUTPUT); //motor dianteiro esquerdo
```

```
    pinMode(IN2_M1, OUTPUT); //motor dianteiro esquerdo
```

```
    pinMode(IN1_M2, OUTPUT); //motor traseiro esquerdo
```

```
    pinMode(IN2_M2, OUTPUT); //motor traseiro esquerdo
```

```
    pinMode(PWM_M3, OUTPUT); //pwm motor dianteiro direito
```

```
    pinMode(PWM_M4, OUTPUT); //pwm motor dianteiro direito
```

```
    pinMode(IN1_M3, OUTPUT); //motor dianteiro direito
```

```
    pinMode(IN2_M3, OUTPUT); //motor dianteiro direito
```

```
    pinMode(IN1_M4, OUTPUT); //motor esquerdo direito
```

```
    pinMode(IN2_M4, OUTPUT); //motor esquerdo direito
```

```
    pinMode (TRIGF, OUTPUT);
```

```
    pinMode (ECHO F, INPUT);
```

```
    pinMode (TRIGE, OUTPUT);
```

```
    pinMode (ECHO E, INPUT);
```

```
    pinMode (TRIGD, OUTPUT);
```

```
    pinMode (ECHO D, INPUT);
```

```
    pinMode(LEDSEQUERDA, OUTPUT);
```

```
    pinMode(LEDSDIREITA, OUTPUT);
```

```
    pinMode(LED FAROL, OUTPUT);
```

```
    pinMode(LED PARE, OUTPUT);
```

```
    pinMode(INTERRUPTOR, INPUT);
```

```
    pinMode(POT1, INPUT);
```

```
    pinMode(POT2, INPUT);
```

```
    digitalWrite(TRIGF, LOW);
```

```
    digitalWrite(TRIGE, LOW);
```

```
    digitalWrite(TRIGD, LOW);
```

```
}
```

```
void motorA(int mode, int percent) //controlador do motor DIANTEIRO ESQUERDO
```

```
{
```

```
    //muda a percentagem de 0 até 100 no PWM
```

```

int duty = map(percent, 0, 100, 0, 255);

switch (mode)
{
    case 0: //desativa o motor
        digitalWrite(PWM_M1, LOW); //define o enable a LOW para desativar o A
        break;

    case 1: //define o sentido horario
        digitalWrite(IN1_M1, HIGH); //define IN1 a HIGH
        digitalWrite(IN2_M1, LOW); //define IN2 a LOW
        analogWrite(PWM_M1, duty); //usa o PWM para controlar a velocidade do motor através
do pin do enable

        break;

    case 2: //define o sentido anti horario
        digitalWrite(IN1_M1, LOW); //define IN1 a LOW
        digitalWrite(IN2_M1, HIGH); //define IN2 a HIGH
        analogWrite(PWM_M1, duty); //usa o PWM para controlar a velocidade do motor através
do pin do enable
        break;

    case 3: //Travar o motor
        digitalWrite(IN1_M1, LOW); //define IN1 a LOW
        digitalWrite(IN2_M1, LOW); //define IN2 a LOW
        analogWrite(PWM_M1, duty); //usa o PWM para travar o motor através do pin do enable

        break;
}
}
//MOTOR TRASEIRO ESQUERDO
void motorB(int mode, int percent) //controlador do motor a
{
    //muda a percentagem de 0 até 100 no PWM

    int duty = map(percent, 0, 100, 0, 255);

    switch (mode)
    {
        case 0: //desativa o motor
            digitalWrite(PWM_M2, LOW); //define o enable a LOW para desativar o A
            break;

        case 1: //define o sentido horario
            digitalWrite(IN1_M2, HIGH); //define IN1 a HIGH
            digitalWrite(IN2_M2, LOW); //define IN2 a LOW

```

```
    analogWrite(PWM_M2, duty); //usa o PWM para controlar a velocidade do motor através  
do pin do enable
```

```
    break;
```

```
case 2: //define o sentido anti horario  
    digitalWrite(IN1_M2, LOW); //define IN1 a LOW  
    digitalWrite(IN2_M2, HIGH); //define IN2 a HIGH  
    analogWrite(PWM_M2, duty); //usa o PWM para controlar a velocidade do motor através  
do pin do enable
```

```
    break;
```

```
case 3: //Travar o motor  
    digitalWrite(IN1_M2, LOW); //define IN1 a LOW  
    digitalWrite(IN2_M2, LOW); //define IN2 a LOW  
    analogWrite(PWM_M2, duty); //usa o PWM para travar o motor através do pin do enable
```

```
    break;
```

```
    }  
}
```

```
//0 = desativa a ponte H
```

```
//1 = o motor gira no sentido horário
```

```
//2 = o motor gira no sentido anti horário
```

```
//3 = trava o motor
```

```
//Motor dianteiro Direito
```

```
void motorC(int mode, int percent) //controlador do motor a
```

```
{
```

```
    //muda a percentagem de 0 até 100 no PWM
```

```
    int duty = map(percent, 0, 100, 0, 255);
```

```
    switch (mode)
```

```
    {
```

```
        case 0: //desativa o motor
```

```
            digitalWrite(PWM_M3, LOW); //define o enable a LOW para desativar o A
```

```
            break;
```

```
        case 1: //define o sentido horario
```

```
            digitalWrite(IN1_M3, HIGH); //define IN1 a HIGH
```

```
            digitalWrite(IN2_M3, LOW); //define IN2 a LOW
```

```
            analogWrite(PWM_M3, duty); //usa o PWM para controlar a velocidade do motor através  
do pin do enable
```

```
            break;
```

```
        case 2: //define o sentido anti horario
```

```
digitalWrite(IN1_M3, LOW); //define IN1 a LOW
digitalWrite(IN2_M3, HIGH); //define IN2 a HIGH
analogWrite(PWM_M3, duty); //usa o PWM para controlar a velocidade do motor através
do pin do enable
```

```
break;
```

```
case 3: //Travar o motor
```

```
digitalWrite(IN1_M3, LOW); //define IN1 a LOW
```

```
digitalWrite(IN2_M3, LOW); //define IN2 a LOW
```

```
analogWrite(PWM_M3, duty); //usa o PWM para travar o motor através do pin do enable
```

```
break;
```

```
}
```

```
}
```

```
//MOTOR TRASEIRO DIREITO
```

```
void motorD(int mode, int percent) //controlador do motor a
```

```
{
```

```
//muda a percentagem de 0 até 100 no PWM
```

```
int duty = map(percent, 0, 100, 0, 255);
```

```
switch (mode)
```

```
{
```

```
case 0: //desativa o motor
```

```
digitalWrite(PWM_M4, LOW); //define o enable a LOW para desativar o A
```

```
break;
```

```
case 1: //define o sentido horario
```

```
digitalWrite(IN1_M4, HIGH); //define IN1 a HIGH
```

```
digitalWrite(IN2_M4, LOW); //define IN2 a LOW
```

```
analogWrite(PWM_M4, duty); //usa o PWM para controlar a velocidade do motor através
```

```
do pin do enable
```

```
break;
```

```
case 2: //define o sentido anti horario
```

```
digitalWrite(IN1_M4, LOW); //define IN1 a LOW
```

```
digitalWrite(IN2_M4, HIGH); //define IN2 a HIGH
```

```
analogWrite(PWM_M4, duty); //usa o PWM para controlar a velocidade do motor através
```

```
do pin do enable
```

```
break;
```

```
case 3: //Travar o motor
```

```
digitalWrite(IN1_M4, LOW); //define IN1 a LOW
```

```
digitalWrite(IN2_M4, LOW); //define IN2 a LOW
```

```
analogWrite(PWM_M4, duty); //usa o PWM para travar o motor através do pin do enable
```

```

        break;
    }
}

//girar carrinho 90 graus para esquerda
void virarEsquerda() {
    int i;
    for (i = 0; i < 10; i++) {
        motorA(2, 70);
        motorB(2, 70);
        motorC(1, 70);
        motorD(1, 70);
    }
}

//girar carrinho 90 graus para direita
void virarDireita() {
    int u;
    for (u = 0; u < 10; u++) {
        motorA(1, 70);
        motorB(1, 70);
        motorC(2, 70);
        motorD(2, 70);
    }
}

//girar carrinho 180 graus no mesmo eixo
void virar180() {
    int y;
    for (y = 0; y < 20; y++) {
        motorA(2, 90);
        motorB(2, 90);
        motorC(1, 90);
        motorD(1, 90);
    }
}

//segue em frente
void reto() {
    int t;
    for (t = 0; t < 10; t++) {
        motorA(1, 90);
        motorB(1, 90);
        motorC(1, 90);
        motorD(1, 90);
    }
}

//segue em para
void parar() {
    int o;
    for (o = 0; o < 10; o++) {

```

```

    motorA(0, 100);
    motorB(0, 100);
    motorC(0, 100);
    motorD(0, 100);
}
}

```

```

void sensorUltrasonico() {
    //sensor frente
    digitalWrite(TRIGF, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGF, LOW);
    tempo1 = pulseIn(ECHOF, HIGH);
    distancia1 = tempo1 / 58;
    //sensor esquerdo
    digitalWrite(TRIGE, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGE, LOW);
    tempo2 = pulseIn(ECHOE, HIGH);
    distancia2 = tempo2 / 58;
    //sensor direito
    digitalWrite(TRIGD, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGD, LOW);
    tempo3 = pulseIn(ECHOD, HIGH);
    distancia3 = tempo3 / 58;

    if (distancia1 <= 20) { //identificou uma parede
        if (distancia2 > distancia3) {
            virarEsquerda();
        }
        if (distancia2 > distancia3) {
            virarDireita();
        }
        if (distancia2 == distancia3) {
            virar180();
        } else {
            reto();
        }
    }
}
}

```

```

void control() {
    estPot1 = analogRead(POT1);
    estPot2 = analogRead(POT2);
    int mapPot1 = map(estPot1, 0, 1023, 0, 100);
    int mapPot2 = map(estPot2, 0, 1023, 0, 100);
}

```

```

//40 a 60 parado, <40 tras ,>60 frente
//40 a 60 frente , < 40 vira a esquerda, > 60 vira a direita);
if (mapPot1 > 40 && mapPot1 < 60) {
    //parar motores
    motorA(0, 100);
    motorB(0, 100);
    motorC(0, 100);
    motorD(0, 100);
}
if (mapPot1 < 40) { //indo para tras
    if (mapPot2 < 40) {
        //motor virar para esquerda indo para tras (valor velocidade motor negativo)
        motorA(2, -70);
        motorB(2, -70);
        motorC(1, -70);
        motorD(1, -70);
    }
    if (mapPot2 > 60) {
        //motor vira para a direita indo para tras (valor velocidade motor negativo)
        motorA(1, -70);
        motorB(1, -70);
        motorC(2, -70);
        motorD(2, -70);
    }
    if (mapPot2 >= 40 && mapPot2 <= 60 ) {
        //motor vai reto para tras ( velocidade do motor negativa)
        motorA(1, -90);
        motorB(1, -90);
        motorC(1, -90);
        motorD(1, -90);
    }
}
}

if (mapPot1 > 60) { //indo para frente
    if (mapPot2 < 40) {
        //motor virar para esquerda indo para frente(valor velocidade motor positivo)
        motorA(2, 70);
        motorB(2, 70);
        motorC(1, 70);
        motorD(1, 70);
    }
    if (mapPot2 > 60) {
        //motor vira para a direita indo para frente(valor velocidade motor positivo)
        motorA(1, 70);
        motorB(1, 70);
        motorC(2, 70);
        motorD(2, 70);
    }
}
}

```



```
if (mapPot2 >= 40 && mapPot2 <= 60 ) {  
    //motor vai reto para frente ( velocidade do motor positivo)  
    motorA(1, 90);  
    motorB(1, 90);  
    motorC(1, 90);  
    motorD(1, 90);  
}  
}  
  
}  
void loop() {  
    SWITCH = digitalRead(INTERRUPTOR);  
    if (SWITCH == 0) {  
        sensorUltrassonico();  
    } else {  
        control();  
    }  
}
```