

Aula 2 – Desenvolvendo a Lógica

Curso: Introdução à Robótica Móvel

PatoBots - UTFPR



CRONOGRAMA

- Lógica e Código do Seguidor de Linha;
- Lógica e Código do Sumô;



LÓGICA DO SEGUIDOR DE LINHA



```
1 // Configuração dos motores
2 const int motor1PWM = 9; // Pino PWM para o motor 1
3 const int motor1DirA = 8; // Pino de controle de direção A do motor 1
4 const int motor1DirB = 7; // Pino de controle de direção B do motor 1
5
6 const int motor2PWM = 10; // Pino PWM para o motor 2
7 const int motor2DirA = 12; // Pino de controle de direção A do motor 2
8 const int motor2DirB = 11; // Pino de controle de direção B do motor 2
9
10 // Definindo os pinos das entradas analógicas
11
12 const int pinA0 = A0;
13 const int pinA1 = A1;
14 const int pinA2 = A2;
15
16 long int faixa = 0;
17 long int ref = 1000;
18 long int erro = 0;
19 long int valorA0, valorA1, valorA2;
20 long int num, den;
21
22 long int delta;
23
24 int velMotor_dir = 100;
25 int velMotor_esq = 100;
26 float Kp = 0.1;
27
```



```
27
28 void setup() {
29
30     Serial.begin(9600);
31
32     pinMode(motor1PWM, OUTPUT);
33     pinMode(motor1DirA, OUTPUT);
34     pinMode(motor1DirB, OUTPUT);
35
36     pinMode(motor2PWM, OUTPUT);
37     pinMode(motor2DirA, OUTPUT);
38     pinMode(motor2DirB, OUTPUT);
39
40 }
41
```



```
41
42 void loop() {
43
44     // Lê os valores das entradas analógicas
45     valorA0 = analogRead(pinA0);
46     valorA1 = analogRead(pinA1);
47     valorA2 = analogRead(pinA2);
48
49     // Calculo para ver a direção que o carrinho deve ajustar
50     num = 0*valorA0 + 1000*valorA1 + 2000*valorA2;
51     den = valorA0 + valorA1 + valorA2;
52
53     faixa = num/den;
54
55     erro = ref - faixa;
56
```



```
56
57 if(erro > 0){
58     delta = Kp*(float)erro;
59     // A Velocidade não pode ser maior que 255
60     if(delta > 255) delta = 255;
61
62     // Movimenta o motor da direita re
63     digitalWrite(motor1DirA, LOW);
64     digitalWrite(motor1DirB, HIGH);
65     analogWrite(motor1PWM, delta);
66
67     // Movimenta o motor da esquerda para a frente
68     digitalWrite(motor2DirA, HIGH);
69     digitalWrite(motor2DirB, LOW);
70     analogWrite(motor2PWM, delta);
71
72 }
```



```
72     }
73     else{
74         delta = -Kp*(float)erro;
75         if (delta > 255) delta = 255;
76
77         // Movimenta o motor da direita frente
78         digitalWrite(motor1DirA, HIGH);
79         digitalWrite(motor1DirB, LOW);
80         analogWrite(motor1PWM, delta);
81
82         // Movimenta o motor da esquerda re
83         digitalWrite(motor2DirA, LOW);
84         digitalWrite(motor2DirB, HIGH);
85         analogWrite(motor2PWM, delta);
86
87     }
```



LÓGICA DO SUMÔ



```
1 // Definindo os pinos do motor na Ponte H
2 const int pinoMotorA1 = 7;    // Pino de controle 1 para motor A
3 const int pinoMotorA2 = 8;    // Pino de controle 2 para motor A
4 const int pinoHabilitarA = 9; // Pino de controle de velocidade para motor A
5 const int pinoMotorB1 = 11;   // Pino de controle 1 para motor B
6 const int pinoMotorB2 = 12;   // Pino de controle 2 para motor B
7 const int pinoHabilitarB = 10; // Pino de controle de velocidade para motor B
8
9 // Definindo os pinos dos sensores reflexivos
10 const int pinoSensorEsquerda = 6; // Pino do sensor reflexivo esquerdo
11 const int pinoSensorCentro = 5;   // Pino do sensor reflexivo central
12 const int pinoSensorDireita = 4;  // Pino do sensor reflexivo direito
13
14 void setup()
15 {
16     // Configurando os pinos como saídas e entradas
17
18     // Configurando os pinos da Ponte H
19     pinMode(pinoMotorA1, OUTPUT);
20     pinMode(pinoMotorA2, OUTPUT);
21     pinMode(pinoHabilitarA, OUTPUT);
22
23     pinMode(pinoMotorB1, OUTPUT);
24     pinMode(pinoMotorB2, OUTPUT);
25     pinMode(pinoHabilitarB, OUTPUT);
26
27     // Configurando os pinos do sensor Óptico Reflexivo
28     pinMode(pinoSensorEsquerda, INPUT);
29     pinMode(pinoSensorCentro, INPUT);
30     pinMode(pinoSensorDireita, INPUT);
31 }
```



```
32 void moverParaFrente()  
33 {  
34     // Define a direção do motor para frente  
35     digitalWrite(pinoMotorA1, LOW);  
36     digitalWrite(pinoMotorA2, HIGH);  
37     digitalWrite(pinoMotorB1, LOW);  
38     digitalWrite(pinoMotorB2, HIGH);  
39  
40     // Ajusta a velocidade do motor  
41     analogWrite(pinoHabilitarA, 50); // 255 é a velocidade máxima  
42     analogWrite(pinoHabilitarB, 66); // 255 é a velocidade máxima  
43 }  
44  
45 void ParardeMover()  
46 {  
47     // Define a direção do motor para frente  
48     digitalWrite(pinoMotorA1, LOW);  
49     digitalWrite(pinoMotorA2, LOW);  
50     digitalWrite(pinoMotorB1, LOW);  
51     digitalWrite(pinoMotorB2, LOW);  
52 }  
53  
54 void moverParaTras()  
55 {  
56     // Define a direção do motor para trás  
57     digitalWrite(pinoMotorA1, HIGH);  
58     digitalWrite(pinoMotorA2, LOW);  
59     digitalWrite(pinoMotorB1, HIGH);  
60     digitalWrite(pinoMotorB2, LOW);  
61  
62     // Ajusta a velocidade do motor  
63     analogWrite(pinoHabilitarA, 50);  
64     analogWrite(pinoHabilitarB, 66);  
65 }
```

```
67 void virarParaDireita()
68 {
69     // Gira para a direita
70     digitalWrite(pinoMotorA1, LOW);
71     digitalWrite(pinoMotorA2, HIGH);
72     analogWrite(pinoHabilitarA, 75); // Reduz a velocidade para facilitar a curva
73     digitalWrite(pinoMotorB1, LOW);
74     digitalWrite(pinoMotorB2, LOW);
75     delay(1000);
76 }
77
78 void virarParaEsquerda()
79 {
80     // Gira para a direita
81     digitalWrite(pinoMotorB1, LOW);
82     digitalWrite(pinoMotorB2, HIGH);
83     analogWrite(pinoHabilitarB, 75); // Reduz a velocidade para facilitar a curva
84     digitalWrite(pinoMotorA1, LOW);
85     digitalWrite(pinoMotorA2, LOW);
86
87     delay(1000);
88 }
```

```
93  if(digitalRead(pinoSensorCentro) == LOW){
94  ParardeMover();
95  delay(500);
96  moverParaTras();
97  delay(1000);
98
99  virarParaDireita();
100 }
101 else if(digitalRead(pinoSensorCentro) == HIGH)
102 {
103     // Nenhuma condição especial, vá em frente
104     moverParaFrente();
105 }
106 }
```