

Ponto de Controle 2 – Grupo 2

Robô Seguidor de Linha utilizando MSP430

Explicando o Código

Milena Andressa da Silva Santos

Brunno Cardoso de Santana

Engenharia Eletrônica

Faculdade Gama – UnB

As entradas para os sensores são definidas como P6_1 e P6_2. Onde os mesmos serão responsáveis pelo sentido de movimento do robô seguidor de linha.

```
#define Direita_Sensor    P6_1    //define a porta de entrada para o sensor IR da direita
#define Esquerda_Sensor  P6_2    //define a porta de entrada para o sensor IR da esquerda
```

Os pinos são colocados em um vetor, para os mesmos realizarem o controle do driver do motor e assim movimentar os motores conforme o valor que o vetor receber.

Os sentidos de movimento são definidos conforme as possibilidades apresentadas pelos sensores de movimento (0 e 1).

Os valores para frente, esquerda, direita e parar, são definidos de acordo com o datasheet do módulo L298N, que analisa os dois valores na porta IN1 e IN2 para um motor e IN3 e IN4 para o segundo motor, para determinar se deverá girar para frente, dar ré ou parar o motor.

Inputs		Function
V _{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V _{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don'tcare

Tabela de Funções (Datasheet), Módulo L298N. [1]

```
int pinos[4] = {Pl_2,Pl_3,Pl_4,Pl_5};    //define um vetor de saida para o Driver do motor

#define frente    0x05
#define esquerda  0x06
#define direita   0x09
#define parar     0x00
```

Nesse bloco, caso o sensor seja configurado em LowMode, o mesmo entrará na primeira condição e o vetor resultado será definido como na figura, caso seja configurado como highMode, entrará na segunda condição(else) e o vetor resultado será definido com o da figura.

```
#ifndef activeLowMode
    int resultado[4] = {frente,esquerda,direita,parar};
#else
    int resultado[4] = {parar,direita,esquerda,frente};
#endif
```

Nesta função, os pinos são colocados como saída. Com a variação de “i”, cada posição é definida como output e as entradas são sempre as mesmas (sensores IR).

```
void setup() {
    for(int i = 0; i < 4; i++)
        pinMode(pinos[i], OUTPUT);
    pinMode(Direta_Sensor, INPUT);
    pinMode(Esquerda_Sensor, INPUT);
}
```

Por fim, tem – se o loop que dará aos pinos o movimento a ser executado. A explicação do código abaixo será feita através de um exemplo para melhor compreensão.

Ex:

Imagina – se que os sensores estão identificando a cor branca em ambos os lados (esquerdo e direito), assim, o movimento a ser executado será o de linha reta, ou seja, o robô seguidor de linha continuará se movimentando em linha reta para frente. Deste modo, será atribuído aos sensores os seguintes números binários:

Sensor direito: 0000 0001

Sensor esquerdo: 0000 0001

Então, ao entrar no loop, o comando da primeira linha irá deslocar o valor do sensor esquerdo um bit para esquerda e realizar uma “OR” com o valor do sensor direito. O valor da soma será armazenado em uma variável chamada “sentido” e

esta variável será responsável por selecionar o movimento no driver do motor. Assim, tem – se o seguinte resultado armazenado na variável sentido:

Sentido = 0000 0011

Ao entrar no loop for, é definida a primeira posição do vetor pinos, que será a posição “i = 0”. A posição do vetor resultado, é definida pelo valor da variável sentido, como no caso a posição 3 corresponde ao movimento de ir para frente, seu valor será 0x05, e este valor será deslocado para direita conforme os valores que “i” assumir, ou seja, como i = 0, não ocorrerá o deslocamento. Logo após o deslocamento, será realizado uma “AND” entre o valor deslocado e o valor correspondente a 0x01, e o resultado será armazenado no vetor pinos na posição 0. E isso acontecerá para os demais valores de “i” enquanto for menor que 4.

No final da interação, o vetor pinos possuirá os seguintes valores:

Pinos[4] = {0, 1, 0, 1}

Como o vetor pinos é saída, o drive utilizado para o controle do motor interpretará que os dois motores devem girar para frente. E isso será feito para cada movimento.

```
void loop() {  
  int sentido = (digitalRead(Esquerda_sensor) << 1) | digitalRead(Direita_sensor); //desloca Esqueda_Sensor uma casa a esquerda  
  for(int i = 0; i < 4; i++)  
    digitalWrite(pinos[i], (resultado[sentido] >> i) & 0x01);  
}
```

Referências Bibliográficas:

[1] DUAL FULL-BRIDGE DRIVER - L298. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/22440/STMICROELECTRONICS/L298N.html>>. Acesso em: 20 de outubro de 2019.