

Robô seguidor de linha

Daniel Barros
up201704271@fe.up.pt

Ricardo Falcão
up201704220@fe.up.pt

2 de Janeiro de 2020

Introdução

No âmbito da cadeira de SBMI, foi nos proposto o desenvolvimento de um robô seguidor de linha, utilizando para isso motores DC e sensores óticos. Neste relatório demonstraremos o nosso processo de desenvolvimento do projeto, assim como as conclusões que dele tiramos.

Arquitetura

O robô projetado é dividido em 3 secções fundamentais: Recolha de informação, processamento dessa informação, e atuação por meio dos motores. Estas serão explicadas ao pormenor de seguida:

Recolha de informação

Na recolha de informação foi utilizada um *array* de 5 sensores IR, o CNY70. O esquema elétrico de cada sensor é o seguinte:

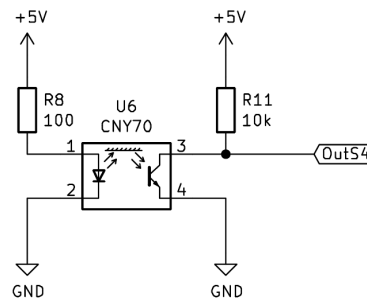


Figura 1: Sensores de linha (CNY70)

Perante esta configuração, era necessário uma corrente de 50 mA no LED emissor, e foi dimensionada então uma resistência limitadora de corrente de 100 Ω ; No coletor do foto-transistor estará uma tensão que varia com a intensidade de luz recebida. Quanto menor for a luz recebida, maior será a tensão V_{out} . Será necessário, na pista, utilizar cores bastante diferentes no chão e na linha, para atingir a máxima resolução deste sensor. Nos nossos testes, foi utilizada uma pista de fundo branco com uma linha preta, mas o contrário era também admissível.

Devido ao elevado número de sinais analógicos a serem processados (que será abordado posteriormente), foi necessário utilizar um multiplexador analógico, nomeadamente o CD4051B. O seu esquema elétrico é o seguinte:

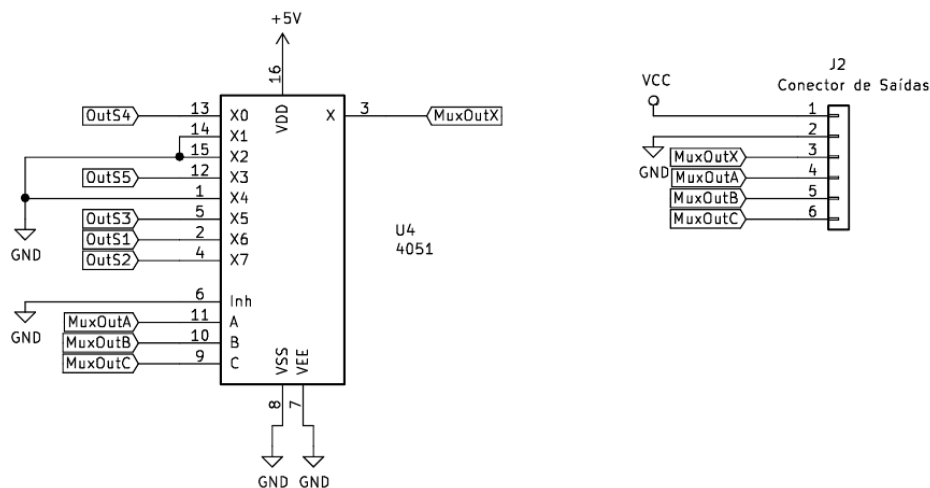


Figura 2: Multiplexador analógico (CD4051B)

Processamento de sinal

Para o processamento de sinal foi utilizado o microcontrolador Atmega328p, numa placa Arduino Uno. O processamento de sinal tem a seguinte sequência de passos:

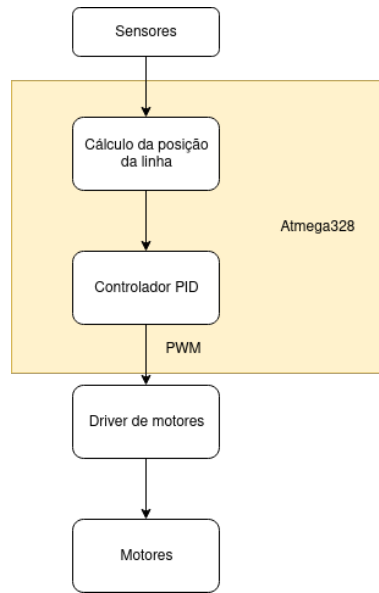


Figura 3: Processamento de sinal (Atmega328)

Para o cálculo da posição da linha é primeiro necessário ler as tensões de saída dos sensores óticos através do conversor ADC interno do microcontrolador. Estes valores devem ser calibrados no código, isto é, armazenado o menor valor lido assim como o valor máximo. Cada leitura posterior do sensor, será então transformada num número decimal, sendo 0 quando corresponde ao valor mínimo, e 1 no valor máximo. Cada leitura é amostrada 4 vezes, sendo feita a média posteriormente, para evitar erros de leitura:

$$P_x = \frac{A_x - A_{min}}{A_{max} - A_{min}}$$

Se para cada sensor atribuirmos o seguinte peso:

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5
1000	2000	3000	4000	5000

então a média pesada destes sensores será a seguinte:

$$L_i = \frac{1000P_1 + 2000P_2 + 3000P_3 + 4000P_4 + 5000P_5}{1000 + 2000 + 3000 + 4000 + 5000}$$

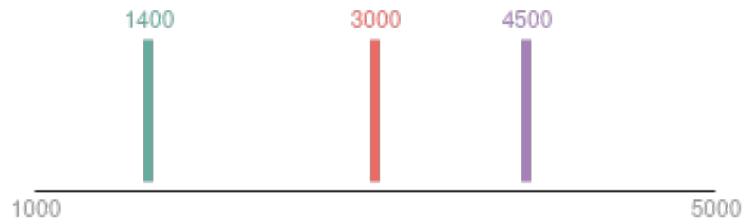


Figura 4: Algumas posições da linha

Posteriormente, este valor correspondente à posição da linha [1000;5000] é transformado numa gama de [-1;1] com a seguinte equação:

$$L = \frac{L_i - 3000}{2000}$$

Atuação dos motores

Utilizamos dois motores de 500 RPM, assim como uma esfera metálica como terceira roda. Para o controlo destes motores era necessário utilizar um driver. Escolhemos o L298N pois permite controlar dois motores ao mesmo tempo, através de um sinal PWM, gerado no nosso microcontrolador. O esquema elétrico do driver é o seguinte:

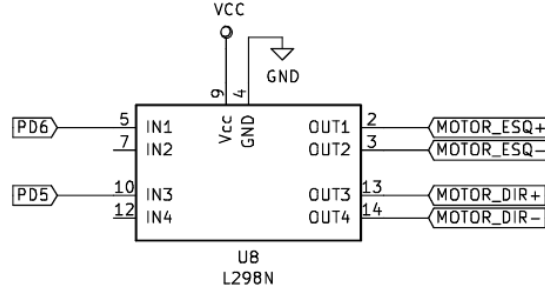


Figura 5: Driver de motores (L298N)

Para controlar os motores, utilizamos dois sinais PWM, sendo que quanto maior for o *duty cycle*, maior será a velocidade dos motores.

Controlo do movimento

Com estas três componentes criamos então um sistema realimentado, como na figura seguinte:

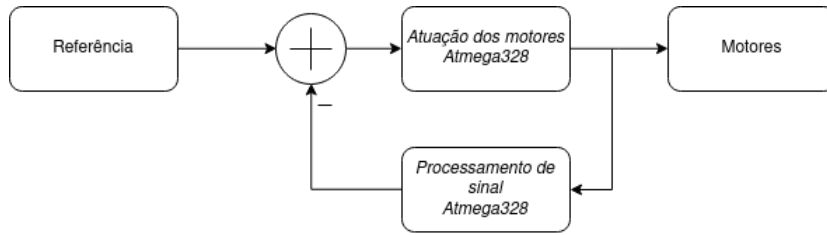


Figura 6: Sistema realimentado simples

De forma a conseguirmos um melhor controlo sobre o movimento do carro, decidimos utilizar um controlador PID. Assim, podemos não só seguir linhas retas, como também diferentes tipos de linhas curvas. Para além das três componentes do PID, utilizamos uma quarta componente, chamada constante de redução, K_R , que utilizamos para reduzir a velocidade do carro quando deteta uma curva.

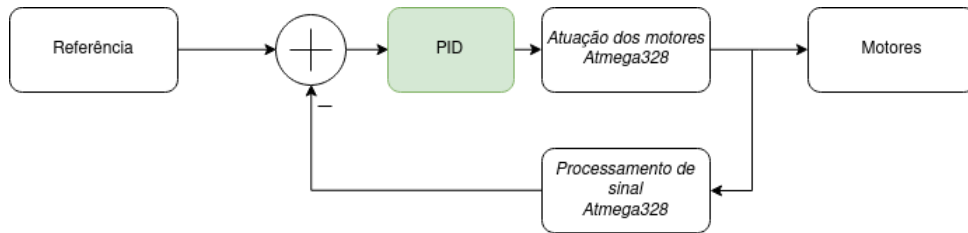


Figura 7: Sistema realimentado com PID

Sendo L a posição atual da linha $[-1;1]$, e L_o a posição da linha anterior, o algoritmo de controlo é o seguinte:

1. $C = K_P L + K_I(L + L_o) + K_D(L_o - L)$
2. $R = K_R(|L| + |L_o|)$
3. $M_{esq} = v_{base} + C - R$
4. $M_{dir} = v_{base} - C - R$

Componentes secundárias

Para além da funcionalidade principal do robô seguidor de linha, implementamos umas funcionalidades extra neste projeto:

Visualizador

Utilizamos um display OLED de 0,96 polegadas para mostrar informações sobre o estado atual do robô:

- Percentagem de bateria restante
- Posição atual da linha (lida dos sensores)
- Estado do robô (Em espera, Automático)

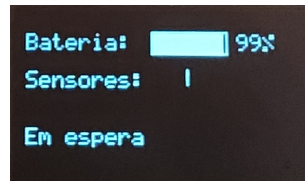


Figura 8: Exemplo de um layout do visualizador

Comando remoto

Para facilitar o desenvolvimento do robô, mas também para adicionar funcionalidade extra, utilizamos um comando de infravermelhos, assim como um recetor ligado ao microcontrolador.



Figura 9: Comando IR

Para pausar/continuar o movimento do robô é premido o botão OK.

As setas são utilizadas para controlar o movimento do robô quando este se encontra no modo manual. Para iniciar o modo manual deve se premir o botão 0 quando este se encontra no estado *Em espera*.

Os restantes números são utilizados para funções internas (calibração, velocidades).