

Trabalho 1

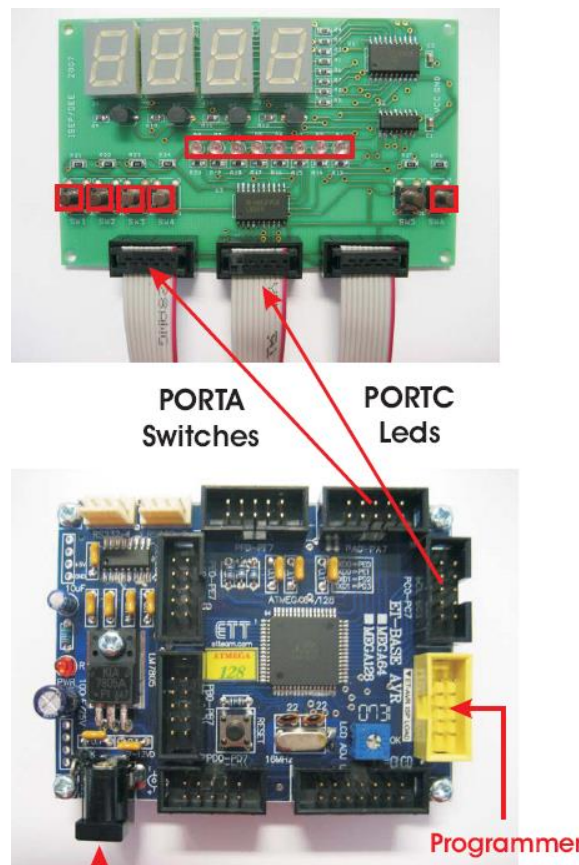
Objectivo: implementação de operações simples de entrada/saída de dados.

Necessário: conhecimentos de arquitectura do μC , arquitectura de memória e portos de I/O

Funcionamento 1: pretende-se controlar o estado dos LEDs D1...D8 utilizando para isso os interruptores SW. O estado dos LEDs deve obedecer à seguinte tabela:

Interruptor activo	LEDs ligados
SW1	D4, D5
SW2	D3, D6
SW3	D2, D7
SW4	D1, D8
SW6	(Leds todos desligados)

Hardware a utilizar:



Porto dos Interruptores

Pino nº	Função	Pino nº	Função
1	SW1 (PA.0)	2	SW2 (PA.1)
3	SW3 (PA.2)	4	SW4 (PA.3)
5	SW5 (PA.4)	6	SW6 (PA.5)
7	MUX.0	8	MUX.1
9	Vcc	10	Ground

Nota: de acordo com o hardware da placa de I/O, o estado de repouso dos interruptores corresponde ao valor lógico 1, o estado de accionado corresponde ao valor lógico 0.

Porto dos LEDs

Pino nº	Função	Pino nº	Função
1	D1 (PC.0)	2	D2 (PC.1)
3	D3 (PC.2)	4	D4 (PC.3)
5	D5 (PC.4)	6	D6 (PC.5)
7	D7 (PC.6)	8	D8 (PC.7)
9	Vcc	10	Ground

Nota: de acordo do o hardware da placa de I/O, para acender um LED deve ser colocado no respectivo pino o valor lógico 0, para apagar um LED deve ser colocado o valor lógico 1.

Implementação do software

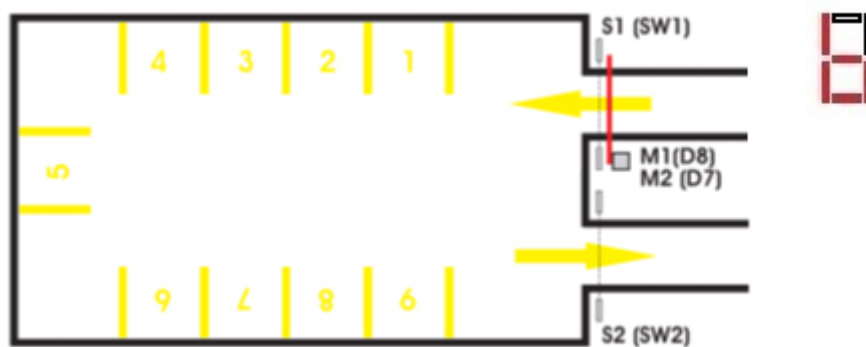
- Utilizando **linguagem Assembly**

Funcionamento 2: partindo da situação em que os 8 leds (**D1 .. D8**) se encontram apagados, pretende-se que, ao acionar **SW1**, os leds sejam ativados sequencialmente, começando pelo led **D1**. Quando todos os leds estiverem ativados, a sequência deve recomeçar com os leds todos desativados. A sequência dos diferentes passos deve ser feita de, aproximadamente, **250 em 250 ms**. A activação de **SW6** permite parar o sequenciamento.

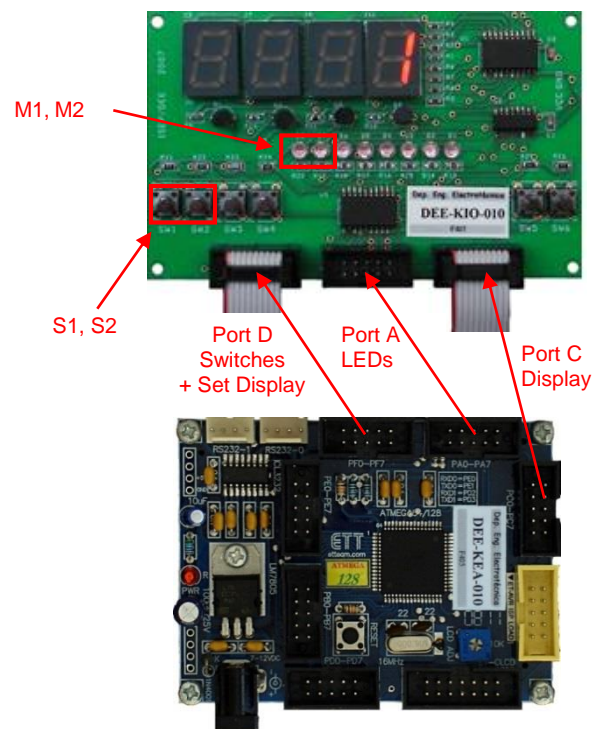
Implementação do software

- Utilizando **linguagem Assembly**

Funcionamento 3: Pretende-se simular o controlo do acesso a uma garagem com uma lotação máxima de 9 viaturas. A entrada e saída de viaturas é detetada, respetivamente, pelos sensores **S1** e **S2**. Sempre que a lotação atingir o máximo, deve ser baixada a cancela de acesso através de **M1 (D8)**. Enquanto houver lugares vagos, a cancela de acesso deve manter-se levantada através de **M2 (D7)**. O display mais à direita deve mostrar o **nº de lugares vagos** na garagem. Quando atingir o valor máximo (**9**) o display deve piscar à frequência de **1 Hz** até que volte a sair uma viatura. Para determinar o valor dos sensores **S1** e **S2** devem ser feitas, para cada sensor, duas leituras com um intervalo de 1 ms.



Hardware a utilizar:



Porto dos Interruptores

Pino nº	Função	Pino nº	Função
1	S1 (PD.0)	2	S2 (PD.1)
3	SW2 (PD.2)	4	SW3 (PD.3)
5	SW4 (PD.4)	6	SW5 (PD.5)
7	MUX.0 (PD.6)	8	MUX.1 (PD.7)
9	Vcc	10	Ground

Nota: PD.0 .. PD.5 devem ser programados como entrada de dados para a leitura dos interruptores. PD.6 e PD.7 devem ser programados como saída de dados e o seu valor deve ser **1** para que o dígito a utilizar seja o dígito da direita.

Porto dos Displays

Pino nº	Função	Pino nº	Função
1	Seg a (PC.0)	2	Seg b (PC.1)
3	Seg c (PC.2)	4	Seg d (PC.3)
5	Seg e (PC.4)	6	Seg f (PC.5)
7	Seg g (PC.6)	8	DP (PC.7)
9	Vcc	10	Ground

Nota: para acender um segmento deve ser colocado no respectivo pino o valor lógico 0, para apagar um segmento deve ser colocado o valor lógico 1.

Tabela dos segmentos

Dígito	DP	Seg g	Seg f	Seg e	Seg d	Seg c	Seg b	Seg a	PORTC
0	1	1	0	0	0	0	0	0	0xC0
1	1	1	1	1	1	0	0	1	0xF9
2	1	0	1	0	0	1	0	0	0xA4
3	1	0	1	1	0	0	0	0	0xB0
4	1	0	0	1	1	0	0	1	0x99
5	1	0	0	1	0	0	1	0	0x92
6	1	0	0	0	0	0	1	0	0x82
7	1	1	1	1	1	0	0	0	0xF8
8	1	0	0	0	0	0	0	0	0x80
9	1	0	0	1	0	0	0	0	0x90

Implementação do software

- Implementar o software utilizando **linguagem Assembly**