

Trabalho 1

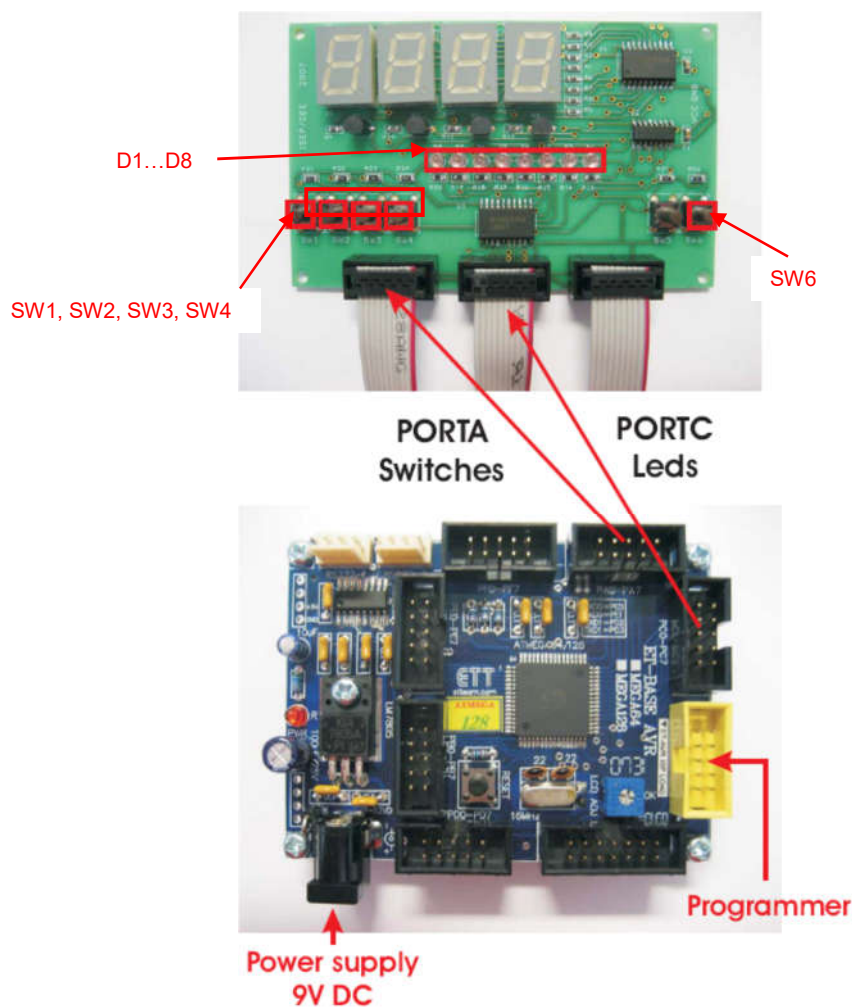
Objectivo: implementação de operações simples de entrada/saída de dados.

Necessário: conhecimentos de arquitetura do μC , arquitetura 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 ativo	LEDs ligados
SW1	(Leds todos ligados)
SW2	D2,D3,D4,D5,D6,D7
SW3	D3,D4,D5,D6
SW4	D4, D5
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 acionado 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 com o hardware da placa de I/O, para acender um LED deve ser colocado no respetivo pino o valor lógico 0, para apagar um LED deve ser colocado o valor lógico 1.

Implementação do software

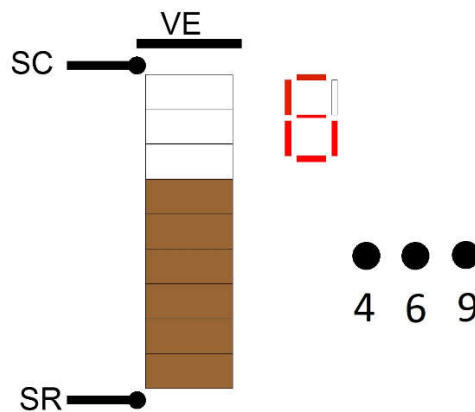
- Utilizando **linguagem Assembly**

Funcionamento 2: partindo da situação em que os 8 leds (**D1 ... D8**) se encontram apagados, pretende-se que, ao acionar **SW1** ou **SW2**, os leds sejam ativados sequencialmente, começando pelo led **D1**. Quando todos os leds estiverem ativados, a sequência deve recomeçar com os leds todos desativados. O tempo entre sequências é controlado por **SW1** e **SW2**. Acionando **SW1** define um tempo de **500 ms**, caso seja acionando **SW2** é definido um tempo de **1 s**. No início considera-se um tempo de **500 ms**. O acionamento de **SW6** permite parar o sequenciamento.

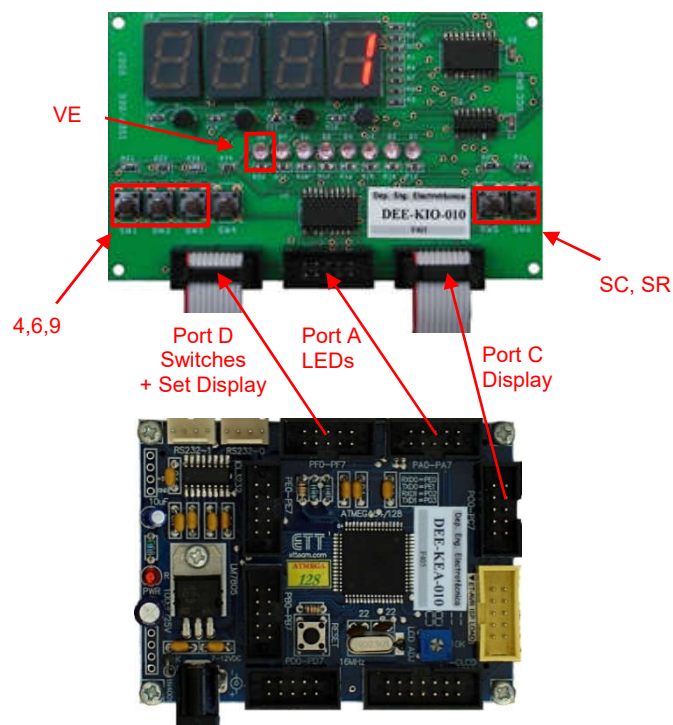
Implementação do software

- Utilizando **linguagem Assembly**

Funcionamento 3: Pretende-se simular o controlo de um dispensador de cápsulas de café com capacidade física máxima de 9 cápsulas. As ações de colocar e retirar cápsulas são detetadas respetivamente pelos sensores SC (SW5) e SR (SW6). Quando é atingida a capacidade máxima pretendida, é atuada a válvula VE (D8) que impede a entrada de mais cápsulas, e que só voltará a abrir quando a capacidade do depósito for inferior a 30% do valor máximo pretendido. A capacidade máxima poderá ser de 4, 6 ou 9 cápsulas. A seleção do número máximo de cápsulas é feita ativando o interruptor respetivo (4,6,9). Existe ainda um mostrador (*display* mais à direita) que indica o **nº cápsulas existentes no depósito**. Para determinar o valor dos interruptores 4, 6 e 9 devem ser feitas, para cada interruptor, duas leituras com um intervalo de 1,5 ms e ambas devem ter o mesmo valor lógico para que seja uma leitura válida. (Inicialmente considera-se que a capacidade máxima pretendida é 9)



Hardware a utilizar:



Porto dos Interruptores

Pino nº	Função	Pino nº	Função
1	4 (PD.0)	2	6 (PD.1)
3	9 (PD.2)	4	SW4 (PD.3)
5	SC (PD.4)	6	SR (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**