

# PMR2415 – Microprocessadores em Automação e Robótica

1ª Prova – 28/04/2014

A prova é com consulta permitida aos *data sheets* dos componentes e à apostila da matéria. Pode ser consultada documentação em meio eletrônico em computador ou tablet, sem conexão com a rede.

Não é permitida a consulta às anotações de aula, anotações na documentação permitida, relatórios e Internet.

Para todas as questões, caso não seja mencionado, considere que o microcontrolador é o PIC16F886, com frequência de oscilador de 10 MHz, e que o compilador C é o XC8.

1. (2,5) Em relação a Linguagem C, pergunta-se:

a) Quais os valores de  $x$  e  $y$  após a execução de cada *sprintf()* no trecho de programa abaixo?

```
char y[10];
int x;
x = 'x';
sprintf(y, "%04d", ++x);
sprintf(y, "%3d", x++);
```

b) Qual é a saída produzida pelo programa abaixo?

```
int main(void) {
    int i=3, *j, k=5; j = &i;
    printf("%d\n", i**j*i+*j-k);
    return 0;
}
```

c) Quais os valores de  $v$ , em hexadecimal e ASCII, e de  $w$ , em decimal, após a execução do trecho de programa abaixo?

```
int v, w;
v = 'O' & 0xff4;
w = v >> 3 << 9;
```

2. (2,5) Reprojete a parte do sistema de pedido de interrupção do PIC, mostrada na Figura 1, utilizando portas coletor-aberto (*open-collector*) onde cabível e acertando a lógica de maneira que os sinais permaneçam no mesmo nível original.

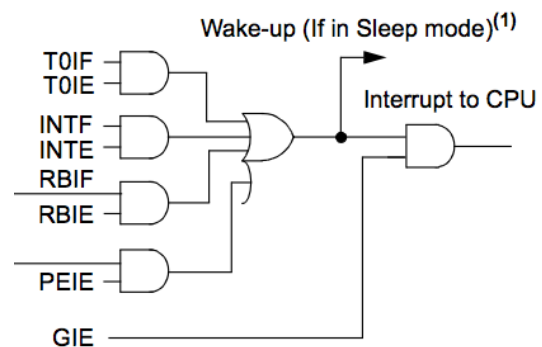


Figura 1: Parte do sistema de pedido de interrupção do PIC (extraído do *data sheet* do PIC 16F886, p. 225, Figura 14-7)

3. (2,5) O sistema de memória de um microcontrolador genérico é organizado de acordo com a Figura 2. Em relação a esse sistema, pergunta-se:

- Quantos bytes possui cada banco de memória?
- Quais bits de endereço controlam os bancos de memória?
- Considerando que os bits de endereço que controlam os bancos de memória são os mais significativos, qual seria a quantidade total de memória que esse microcontrolador conseguiria endereçar?

Todas as respostas devem ser acompanhadas das devidas justificativas.

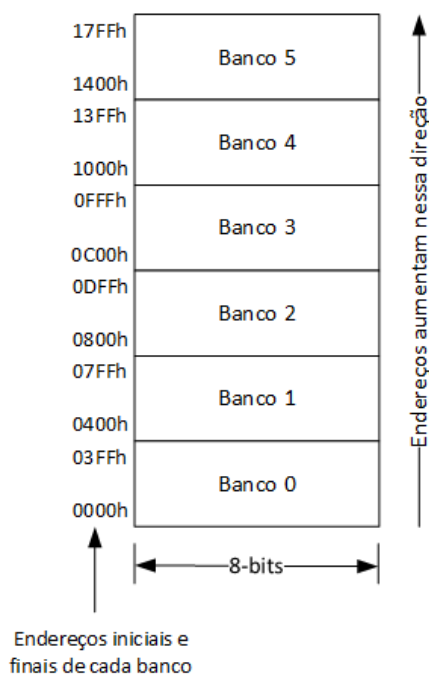


Figura 2: Organização da memória de um microcontrolador genérico

4. (2,5) Mostre os cálculos e escreva a rotina de interrupção e o trecho de código no *main()* para configurar e operar o Timer 1 de um PIC para gerar uma interrupção periódica a cada 10 ms usando o clock interno. A rotina de interrupção e o trecho

de código do *main()* devem ser escritos em Linguagem C.

5. (2,5) Escreva duas funções em C que atendam as seguintes especificações:

*(void) adc\_init(int channel)* – inicializa o canal de conversão A/D cujo número é passado como parâmetro e pode assumir valores entre 0 e 3

*(int) adc\_read(int channel)* – retorna o valor da conversão A/D do canal especificado como parâmetro, justificado à direita. Essa função dispara a conversão e aguarda seu término, retornando somente quando o resultado da conversão estiver pronto.

Lembre-se que no compilador XC8 o tipo de variável *int* é de 16-bits e considere que os sinais analógicos nas entradas dos canais A/D podem variar entre  $V_{SS}$  e  $V_{DD}$ . Indique o valor de  $T_{AD}$  obtido.

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
64	40	100	�#64;	�	96	60	140	�#96;	`
65	41	101	�#65;	A	97	61	141	�#97;	a
66	42	102	�#66;	B	98	62	142	�#98;	b
67	43	103	�#67;	C	99	63	143	�#99;	c
68	44	104	�#68;	D	100	64	144	�#100;	d
69	45	105	�#69;	E	101	65	145	�#101;	e
70	46	106	�#70;	F	102	66	146	�#102;	f
71	47	107	�#71;	G	103	67	147	�#103;	g
72	48	110	�#72;	H	104	68	150	�#104;	h
73	49	111	�#73;	I	105	69	151	�#105;	i
74	4A	112	�#74;	J	106	6A	152	�#106;	j
75	4B	113	�#75;	K	107	6B	153	�#107;	k
76	4C	114	�#76;	L	108	6C	154	�#108;	l
77	4D	115	�#77;	M	109	6D	155	�#109;	m
78	4E	116	�#78;	N	110	6E	156	�#110;	n
79	4F	117	�#79;	O	111	6F	157	�#111;	o
80	50	120	�#80;	P	112	70	160	�#112;	p
81	51	121	�#81;	Q	113	71	161	�#113;	q
82	52	122	�#82;	R	114	72	162	�#114;	r
83	53	123	�#83;	S	115	73	163	�#115;	s
84	54	124	�#84;	T	116	74	164	�#116;	t
85	55	125	�#85;	U	117	75	165	�#117;	u
86	56	126	�#86;	V	118	76	166	�#118;	v
87	57	127	�#87;	W	119	77	167	�#119;	w
88	58	130	�#88;	X	120	78	170	�#120;	x
89	59	131	�#89;	Y	121	79	171	�#121;	y
90	5A	132	�#90;	Z	122	7A	172	�#122;	z
91	5B	133	�#91;	[	123	7B	173	�#123;	{
92	5C	134	�#92;	\	124	7C	174	�#124;	
93	5D	135	�#93;	]	125	7D	175	�#125;	}
94	5E	136	�#94;	^	126	7E	176	�#126;	~
95	5F	137	�#95;	_	127	7F	177	�#127;	DEL

Figura 3: Tabela ASCII