Sistemas Embebidos I
Semestre de Inverno de 2016/2017
2º teste - 11 de fevereiro de 2017

Grupo I

Na sua folha de teste, indique **a** opção que considera correta como resposta a cada uma das seguintes questões de escolha múltipla. Note que a indicação de uma opção incorreta **desconta 50%** da cotação atribuída à questão em causa.

1. [1] Considere o `Timer0` do microcontrolador LPC2106. Para que o registo `TC` (*Timer Counter*) evolua ao ritmo de `PCLK` o valor do registo `PR` (*Prescaler Register*) deve ser igual a:

A – `0`
B – `1`
C – `PCLK`

2. [1] Qual é o valor guardado no registo `r0` após a execução do seguinte troço de código, escrito em linguagem *assembly* da arquitetura ARM?

A – `0x0000800C`
B – `0xFFFF8008`
C – `0xFFFF800C`

```
0x00008000         ldrsh r0, label
0x00008004         b .
0x00008008 label:  .word 0x0000800C
0x0000800C         .word 0x00008008
```

3. [1] A função `FUNCT` está codificada em *Thumb.* Qual a sequencia de instruções *assembly* da arquitetura ARM deve ser utilizada para realizar a chamada à função:

A – `mov r14, pc; b FUNCT`
B – `ldr r12, =FUNCT; b r12`
C – `ldr r12, =FUNCT; bx r12`

4. [1] Tendo localizado a secção `.stack` a partir do endereço `0x40001000`. Qual deve ser a iniciação do registo `SP` por forma a maximizar a dimensão de *stack* utilizável.

A – `ldr r0, =__tos__; add sp, r0, #0`
B – `ldr r0, =__tos__; add sp, r0, #1`
C – `ldr r0, =__tos__; add sp, r0, #4`

```
.section .stack
      .space 1024
__tos__:
```

Grupo II

5. [2] Comente a seguinte afirmação: "O modulo `init.S` apresentado nas aulas é sempre necessário para a produção de um programa para o microcontrolador LPC2106 e tem que ser sempre escrito em *assembly* da arquitetura ARM". Justifique a sua resposta.

6. [2] Considere a seguinte definição, relativa ao RTC (*Real Time Clock*) presente no microcontrolador LPC2106 e da estrutura `TimeDsc`.
Escreva a função `TimeDsc RTC_GetT(void)` que retorna, **de forma coerente**, o valor atual das horas, minutos e segundos do RTC.

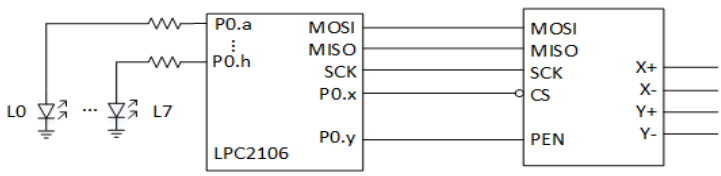
```
typedef struct {
 ...
 volatile unsigned int CTIME0, CTIME1, CTIME2;
 volatile unsigned int SEC, MIN, HOUR;
 ...
} LPC210X_RTC;
#define LPC2106_RTC (*((LPC210X_RTC *)0xE0024000))

typedef struct {
 unsigned int h, m, s;
} TimeDsc;
```

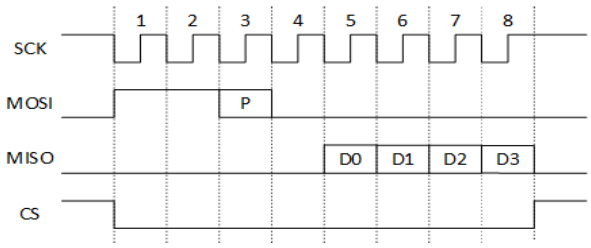
7. [2] Apresente de forma sucinta a motivação para, no microcontrolador LPC2106, o divisor de relógio do *Real Time Clock* (RTC) ter que ser definido por dois registos, `PREINT` e `PREFRAC`. Justifique a sua resposta.

8. [2] Apresente, de forma sucinta, as condições que levam o *bootloader* do microcontrolador LPC2106 a ser executado.

Grupo III

9. [8] A figura em baixo ilustra um sistema embebido baseado no microcontrolador LPC2106, um conjunto de LEDs, L0 a L7, e um controlador de *touchscreen* com interface para barramentos série síncrono.





Os pinos X+, X-, Y+ e Y-, são ligados ao *touchscreen*. A pressão detetada num determinado ponto do *touchscreen* é indicada pelo controlador com a ativação do sinal `PEN`, convertendo a pressão nesse ponto em dois valores (entre `0x1` e `0xF`) que correspondem à coordenada X e Y do ponto.
As leituras dos valor X e Y fazem-se consultando o conversor `0` (eixo dos Xs) e o conversor `1` (eixo dos Ys). O valor de `P` indica qual o conversor a ser consultado (`0` – conversor `0`; `1` – conversor `1`).
O controlador respeita o diagrama temporal apresentado na figura a cima.
Os LEDs devem ser atualizados, com as novas coordenadas do ponto, com período de 5seg, ficando acesos durante 3seg.

a) [1] Indique se a seleção do pino `P0.x` pode ser o pino `P0.7`. No caso de não ser possível, proponha um novo pino do microcontrolador para esta função. Justifique a sua resposta.

b) [1] Realize a função `TOUCH_ChipSelect(int active)` que manipula o pino selecionado para `P0.x`. Se o parâmetro `active` for zero o controlador de *touchscreen* não está selecionado; se o parâmetro for diferente de zero o controlador está selecionado.

c) [2] Programe a função `int TOUCH_Read(int *x, int *y)` responsável pela leitura do controlador. A função devolve `0` se não existir pressão no *touchscreen*. Caso contrario devolve `1` e os parâmetros `x` e `y` recebem o valor da coordenada X e Y do ponto, respetivamente. Admita a existência da função `int SPI_XTransfer(int bitsData, int CPOL, int CPHA, unsigned char *data, int len)`, que realiza a transferência do *buffer* `data` de dimensão `len` de acordo com a os parâmetros de comunicação `bitsData`, `CPOL` e `CPHA`.

d) [2] Desenhe um diagrama de estados representativo do funcionamento do sistema, explicitando os objetos intervenientes e os eventos que o fazem evoluir.

e) [2] Apresente o programa que implementa a solução descrita na alínea anterior, admita a existência das funções:

   i. `unsigned int TIMER0_Elapse(void)` – esta função retorna o valor corrente, em *ticks*, do contador do `Timer0` que está programado com um período de *tick* de 500ms.
   ii. `void Display(int x, int y)` – esta função fixa nos LEDs o valor de `x` e de `y`. Se `x` e `y` forem igual a zero os LEDs ficam apagados.

Pedro Sampaio