

**Sistemas Embebidos I**  
**Semestre de Inverno de 2015/2016**  
1º teste - 25 de Janeiro de 2016

**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 que é utilizada a seguinte estrutura de diretoria para gerar o ficheiro executável `prog.axf`. Qual o resultado da execução do seguinte comando na diretoria `se1`?
- ```
se1
+-- liblib.a
+-- init.S
+-- init.o
+-- main.c
+-- main.o

$ arm-none-eabi-ld main.o init.o -llib -o prog.axf
```
- A – A execução do comando apresenta o erro: `arm-none-eabi-ld: cannot find -llib`  
B – A execução do comando apresenta o erro: `arm-none-eabi-ld: cannot find main.o: No such file or directory`  
C – A execução do comando sucede sem erros e é criado o ficheiro `prog.axf`.

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?
- ```
mov r0, #-1
adds r2, r0, #1
bmi cont
sub r0, r0, #-1
cont:
ldr r2, [r0,#4]
```
- A – -1  
B – 0  
C – +3

3. [1] Indique as instruções *assembly* da arquitetura ARM que devem ser utilizadas para carregar o conteúdo -873 do endereço `0x4000021A` no registo `r0`.
- A – `ldr r0, =0x40000218; ldrsb r0, [r0, #2]`  
B – `ldr r0, =0x40000218; ldrsh r0, [r0, #2]`  
C – `ldr r0, =0x40000218; ldrsh r0, [r0], #2`

4. [1] Considere o seguinte resultado da execução do programa `arm-none-eabi-objdump` sobre código *assembly* da arquitetura ARM, desenvolvido para ser executado no microcontrolador LPC2106.
- Qual é o valor presente no endereço `0x14`?
- ```
00000000 <_start>:
0: e59ff018 ldr pc, [pc, #24]
4: e51ff00c ldr pc, [pc, #-12]
8: e51ff010 ldr pc, [pc, #-16]
c: e51ff014 ldr pc, [pc, #-20]
10: e51ff018 ldr pc, [pc, #-24]
14:
18: e51ff020 ldr pc, [pc, #-32]
1c: e51ff024 ldr pc, [pc, #-36]
00000020 <reset_addr>:
...
```
- A – `0x445F90A4`  
B – `0xBBA06F5B`  
C – `0xBBA06F5C`

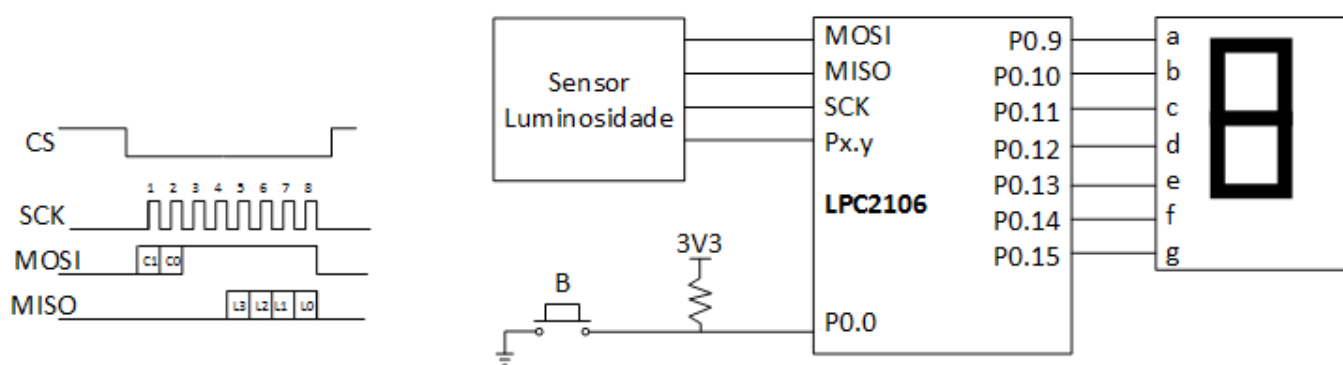
**Grupo II**

5. [2] Relativamente à norma APCS, apresente de forma sucinta como é definida a passagem de parâmetros e o retorno de valores das funções.
6. [2] Comente a seguinte afirmação: “A evocação das funções IAP programadas no microcontrolador LPC2106 pode ser feita utilizando o seguinte troço de código *assembly* ARM desde que este seja codificado em modo Thumb.” Justifique a sua resposta.
- ```
ldr r0, =0x7fffffff
bx r0
```

7. [3] Comente a seguinte afirmação: “Num sistema dedicado baseado no microcontrolador LPC2106, em que este dispositivo é o único *master* SPI, a resolução de endereços dos dispositivos *slaves* não é resolvida pelo *master* SPI.” Justifique a sua resposta.

### Grupo III

8. [9] A figura em baixo ilustra um sistema embebido baseado no microcontrolador LPC2106 que afixa num mostrador de 7 segmentos o valor lido do sensor de luminosidade, cada vez que é pressionado o botão de pressão B (sem repetição). Neste sistema, o sensor de luminosidade tem interface para barramentos serie síncrono e informa valores de luminosidade entre 0 e 9.



- [1] Indique, justificando, qual o pino do microcontrolador a utilizar para  $Px.y$ . Apresente a sua programação.
- [2] Programe a função `void Sensor_Init()`, responsável pela iniciação do módulo do sensor como indicado diagrama temporal. Admita a existência da função `SPI_Init(int bitsData, int CPOL, int CPHA)`, que inicia o controlador SPI para comunicar com uma palavra de dimensão `bitsData` e configura os valores de `CPOL` e `CPHA`.
- [2] Programe a função `unsigned int Sensor_Read()` responsável pela leitura do valor do sensor de luminosidade de acordo com o diagrama temporal indicado. A função devolve um valor entre 0 e 9. Admita a existência da função `unsigned int SPI_Transfer(unsigned int tx)`, que envia o valor passado em `tx` e retorna o recebido.
- [2] Desenhe um diagrama de estados representativo do funcionamento do sistema, explicitando os objetos intervenientes e os eventos que o fazem evoluir.
- [2] Apresente o programa que implementa a solução descrita na alínea anterior, admita a existência das funções:

`int BUTTON_Press(void)` – esta função retorna 0 se não foi detetado botão pressionado ou 1 se pressionado

`void SEG_Show(int val)` – esta função afixa no mostrador de 7 segmentos o valor passado em `val`.

*Pedro Sampaio*