

Sistemas Embebidos I
Semestre de Inverno de 2018/2019
2º teste - 29 de janeiro de 2019

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] O processador Cortex-M3 contém o periférico:

- A – SysTick.
- B – GPIO.
- C – TIMER0.

2. [1] A instrução do processador Cortex-M3 `ITTT EQ` significa que, sendo a *flag* `Z = 1`:

- A – as três instruções seguintes não se executam.
- B – as três instruções seguintes executam-se.
- C – as duas instruções seguintes executam-se.

3. [1] Para colocar no microcontrolador LPC1769 o pino `P2.3` com o valor lógico um, utilizando as definições do CMSIS, deve ser feito:

- A – `LPC_GPIO2->FIOSET |= ~0x08;`
- B – `LPC_GPIO2->FIOSET = 0x03;`
- C – `LPC_GPIO2->FIOSET = 0x08;`

4. [1] Admita o microcontrolador LPC1769 a funcionar a 50MHz e com `PCLK` igual a `CCLK`. Para que o contador, `TC`, do periférico `TIMER0` evolua com um período de 1s, qual o valor do registo `PR`:

- A – 1
- B – 49999999
- C – 50000000

Grupo II

5. [2] Comente, justificando, a seguinte afirmação: “O periférico `GPIO` do microcontrolador LPC1769 não permite, de forma atómica, inverter o estado dos pinos. Por exemplo, se no porto 0 estiver o valor `0x00000055`, não é possível de forma atómica, colocar neste porto o valor `0xFFFFFFFFAA`.”

6. [3] Apresente, de forma sucinta, as razões para o periférico `RTC`, existente no microcontrolador LPC1769, disponibilizar os registos `CTIME0`, `CTIME1` e `CTIME2`.

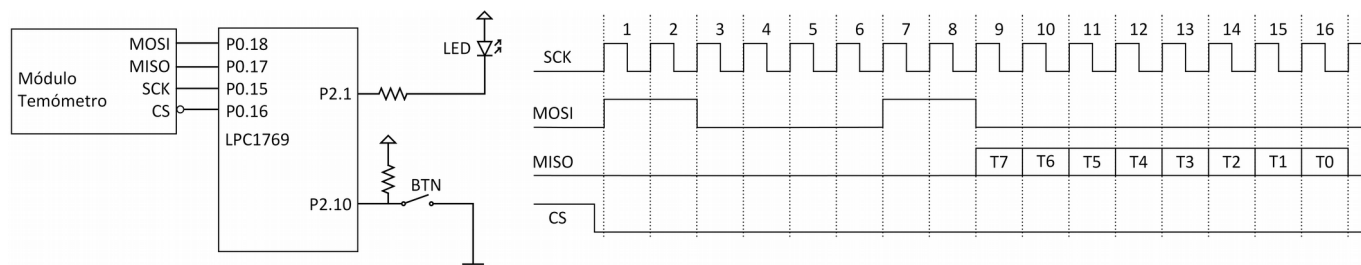
7. [2] Relativamente ao barramento `SPI`, justifique, a seguinte afirmação: “Sendo o `SPI` um barramento série síncrono é possível realizar transferências *full-duplex*.”

Grupo III

8. [9] A figura em baixo ilustra um sistema embebido baseado no microcontrolador LPC1769 para controlo de temperatura. O sistema é composto por um módulo termómetro, com interface para barramento série síncrono, que informa a temperatura, um LED e um botão de pressão. O sistema monitoriza a temperatura. Quando o valor da temperatura for superior a 28 o LED é ligado.

O LED permanece ligado até que, ou a temperatura é inferior a 25, ou é pressionado (detetada transição descendente) o botão.

O módulo termómetro recebe o comando (0xC3) e devolve a temperatura em unidades de graus centígrado.



Considere a existência das funções:

`SPI_Init(int bitsData, int CPOL, int CPHA)` – esta função inicia o controlador SPI para comunicar com uma palavra de dimensão `bitsData` e configura os valores de `CPOL` e `CPHA`.

`void SPI_Transfer(unsigned short *tx, unsigned short *rx, int size)` – esta função realiza a transferência SPI, onde `tx` é o apontador para o *buffer* a transmitir, `rx` é o apontador para o *buffer* de receção e o parâmetro `size` indica a dimensão dos *buffers*.

`void LED_Value(int state)` – esta função posiciona o LED do sistema valor lógico 0 ou 1 de acordo com o parâmetro `state`.

Admita a existência das definições CMSIS para o microcontrolador LPC1769.

- [3] Programe a função `void SYSTEM_Init()`, responsável pela iniciação total do sistema.
- [2] Programe a função `bool BUTTON_Read()`, responsável pela leitura do estado corrente do botão de pressão. Esta função não é bloqueante e retorna `true` se botão pressionado.
- [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.

Pedro Sampaio