

Sistemas Embebidos I
Semestre de Inverno de 2014/2015
2º teste - 18 de Fevereiro de 2015

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 ficheiro `f.s` com o seguinte troço de código *assembly* da arquitetura ARM. Qual o resultado da execução do comando `arm-none-eabi-as -o f.o f.s`?

- A – É gerado o ficheiro `f.o`.
- B – Error: invalid constant (7F0) after fixup
- C – Error: invalid constant (FF0) after fixup

```
.text
.global _start
_start:
    mov r0, #0x7F0
    mov r1, #0xFF0
    sub r0, r1, r0
```

2. [1] Considere que é utilizada a seguinte estrutura de diretórias para gerar o ficheiro executável `test.axf`. Qual o resultado da execução do seguinte comando na diretoria `sources`?

`arm-none-eabi-ld -o test.axf main.o -llpc2106 -L../libs init.o`

- A – A execução do comando sucede sem erros e é criado o ficheiro `test.axf`.
- B – A execução do comando apresenta o erro:
`arm-none-eabi-ld: cannot find -llpc2106`
- C – A execução do comando apresenta o erro:
`arm-none-eabi-ld: cannot find init.o: No such file or directory`

```
sel
+- libs
|   +- liblpc2106.a
\-- sources
    +- init.S
    +- init.o
    +- main.c
    +- main.o
```

3. [1] Considere o ficheiro `const.s` com o seguinte troço de código *assembly* da arquitetura ARM. Qual o resultado da execução do comando?

`arm-none-eabi-as -o const.o const.s`

- A – A execução do comando sucede sem erros e é criado o ficheiro `const.o`.
- B – Error: unknown pseudo-op: ``.rodata'`
- C – Error: internal_relocation (type: OFFSET_IMM) not fixed up

```
.section .rodata
const: .word 0x12

.text
ldr r0, const
ldr r0, [r0]

.end
```

4. [1] Considere o seguinte troço de programa escrito em linguagem C para a arquitetura ARM de acordo com a norma APCS. Qual o comportamento resultante da evocação da função `f`?

- A – Não é possível evocar a função, pois esta não pode retornar `struct res`.
- B – Retorna o resultado no registo `R0`.
- C – Retorna o resultado no par de registos `R1:R0`.

```
struct res {
    short c1, c2;
};

struct res f(char c) {
    struct res r = {1,2};
    return r;
}
```

Grupo II

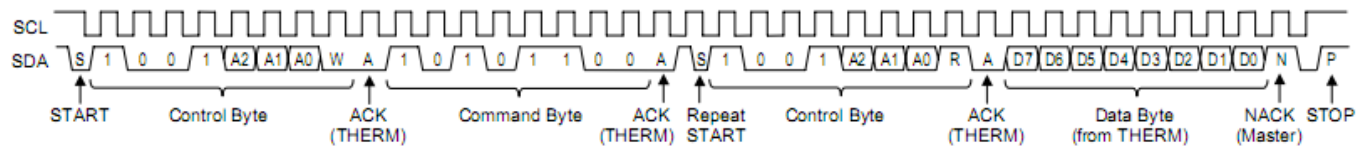
5. [3] Considere a seguinte definição, relativa ao RTC presente no LPC2106.

- a) [1] Justifique a existência dos registos `CTIME0`, `CTIME1` e `CTIME2`.
- b) [2] Apresente um troço de código escrito em linguagem C que, sem usar os registos `CTIME0`, `CTIME1` e `CTIME2`, preencha uma variável do tipo `struct tm` que está definido na biblioteca standard da linguagem C.

```
typedef struct {
    LPC210X_REG ILR, CTC, CCR, CIIR, AMR;
    LPC210X_REG CTIME0, CTIME1, CTIME2;
    LPC210X_REG SEC, MIN, HOUR;
    LPC210X_REG DOM, DOW, DOY, MONTH, YEAR;
    ...
} LPC210X_RTC;

#define LPC2106_RTC \
    (*((LPC210X_RTC *)0xE0024000))
```

6. [3] O diagrama represente a leitura da configuração do termómetro DS1631, onde $A2 = A1 = A0 = 0$.



- a) [1,5] Indique, justificando, o motivo pelo qual são realizadas duas transferências I2C para se obter a configuração do termómetro.
- b) [1,5] Justifique a utilização do comando “Repeat START” apresentado no diagrama.

7. [2] Considere o seguinte troço de código, pertencente ao ficheiro `init.s`, em que os símbolos `__data_start__`, `__data_end__` e `__rom_data_start__` estão definidos no ficheiro `ldscript`.

a) [1] Justifique a utilidade deste troço de código.

b) [1] Indique o valor relativo de `__data_end__` em relação a `__data_start__` para o seguinte programa, escrito em linguagem C.

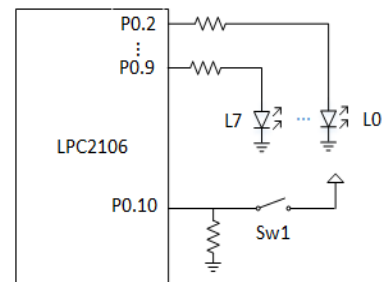
```
void main() {
    int j, i = 10;
    while ( j++ < i );
}
```

```
...
ldr    r1, =__data_start__
ldr    r2, =__rom_data_start__
ldr    r3, =__data_end__
cmp    r1, r2
beq    end_set_loop
set_loop:
cmp    r1, r3
ldrne  r4, [r2], #4
strne  r4, [r1], #4
bne    set_loop
end_set_loop:
bl     main
b      .
```

Grupo III

8. [8] A figura ao lado ilustra um sistema embebido baseado num microcontrolador LPC2106 que implementa o deslizamento de um LED aceso sobre um conjunto de 8 LEDs ($L0 \dots L7$).

Após arranque, o sistema inicia o deslizamento no sentido $L0 \rightarrow L7$ apenas quando o botão de pressão (sw1) é pressionado. Pressionar o botão sw1 (toque curto) durante o deslizamento faz com que o sistema inverta o sentido do deslizamento. Pressionar o botão sw1 durante mais de 500ms (toque longo) para o deslizamento do LED e coloca o sistema na condição inicial. O LED desloca-se de 1s em 1s.



- a) [2] Implemente a função `int Key_Press()` em linguagem C. Esta função retorna `NONE` quando o botão não está a ser pressionado, `SHORT_PRESS` quando ocorreu um toque curto e `LONG_PRESS` se ocorreu um toque longo. Pretende-se que o código seja o mais eficiente possível.
- b) [1] Programe a função `void LedUpdate(unsigned int ledId)` para controlo do estado dos LEDs, em que o parâmetro `ledId` identifica o LED que se quer acender. Esta função deve garantir que, em todos os momentos, apenas um LED esteja aceso.
- c) [1] Considerando que a frequência do sinal de relógio aplicado ao microcontrolador LPC2106 é de 12MHz, programe a função `TIMER0_Init()` que faz a iniciação do `Timer0` para utilização como temporizador da aplicação com uma duração de *tick* de 5ms. Justifique a sua resposta, explicitando toda a programação e definição de valores envolvida.
- 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 troço de código escrito em linguagem C que implementa a solução descrita na alínea d). Considere a existência das funções `TMR0_GetValue(void)` e `TMR0_Elapsed(unsigned int last)` que devolvem o tempo decorrido, em unidades de contagem (`ticks`), até ao instante atual e desde `last` até ao instante atual, respetivamente.