

**Sistemas Embebidos I**  
**Semestre de Inverno de 2014/2015**  
1º teste - 2 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 (0FFFFFF0) after fixup

C – Error: invalid constant (0FFFFFF00) after fixup

```
.text
.global _start
_start:
    mov r0, #0x0FFFFFF0
    mov r1, #0x0FFFFFF00
    sub r0, r1, r0
```

2. [1] Admita o seguinte troço de programa, em que a função `GPIO_BitSet(int bit)` está definida no ficheiro `liblpc2106.a`. Indique o resultado da execução do seguinte comando:

`arm-none-eabi-ld -o t.axf -L. -llpc2106 main.o`

A – A execução do comando sucede sem erros e é criado um ficheiro `t.axf`

B – A execução do comando apresenta o erro: undefined reference to `GPIO\_BitSet'

C – A execução do comando apresenta erro, pois a assinatura da função `GPIO_BitSet` não é conhecida.

```
...
int main() {
    ...
    GPIO_BitSet(1);
    ...
}
```

3. [1] Considere o seguinte troço do ficheiro `ldscript`, utilizado para gerar ficheiros executáveis para o microcontrolador LPC2106. Indique a sequência de instruções que permite maximizar a zona de memória `stack`.

A – `ldr r0, =__stack_start__; mov sp, r0`

B – `ldr r0, =__stack_end__; mov sp, r0`

C – `ldr r0, =__stack_end__; add sp, r0, #4`

```
.stack : {
    __stack_start__ = .;
    . += 0x1000;
    __stack_end__ = .;
} > RAM
```

4. [1] Indique a sequência de instruções *assembly* da arquitetura ARM equivalente à instrução `b1 label`?

A – `add lr, pc, #0; b label`

B – `add lr, pc, #4; b label`

C – `add lr, pc, #8; b label`

**Grupo II**

5. [2] Pretende-se carregar a constante `0x54321` para o registo `r1` do microcontrolador LPC2106. Apresente duas formas alternativas para realizar esta operação, enumerando as suas vantagens e desvantagens.

6. [2] Comente a seguinte afirmação: “É impossível interagir com 128 dispositivos diferentes ligados a um barramento I2C quando são utilizados sete bits para codificar os seus endereços.”

7. [2] Considere o seguinte troço de código *assembly* da arquitetura ARM utilizado como *stub* para a execução dos comandos IAP do microcontrolador LPC2106.

a) [1] Justifique a utilização do registo *ip*. Poderia utilizar-se outro registo do processador para realizar a instrução *bx*? Justifique.

b) [1] Admitindo que este *stub* é utilizado pela função *func* durante a execução do comando “Copy RAM to Flash (51)”, explique como se processa o retorno à função *func* após a execução do comando IAP.

```
.text
.global IAP_Entry
IAP_Entry:
    mov ip, 0x7fffffff1
    bx ip
```

8. [2] Considere o seguinte programa escrito em linguagem C, localizado na memória FLASH do microcontrolador LPC2106 após esta ter sido completamente apagada.

a) [1] Indique, justificando, as secções em que serão localizadas todas as variáveis utilizadas no programa.

b) [1] Considerando que o programa é executado imediatamente após o arranque do sistema, indique o valor da variável *r* quando o programa termina. Justifique a sua resposta.

```
#define m 3

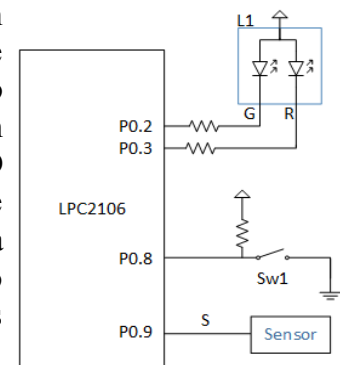
char *x = (char *) 0x1C000;
char r;

void main (void) {
    int i[] = {1, 2, 3, 4};
    int j;

    for(j=m; j; j--)
        r += i[j] * *x;
}
```

### Grupo III

9. [8] A figura ao lado ilustra um sistema embebido baseado num microcontrolador LPC2106 que implementa um semáforo para portão de garagem. Este sistema inclui um botão de pressão (sw1) para dar o comando de abertura do portão, um sensor (s) que indica que o portão está em movimento e um LED bi-cor (L1) para informar sobre o seu estado. O LED L1 apresenta a cor vermelha quando o portão está fechado e a cor verde quando está aberto. Quando o portão está em movimento, L1 pisca à frequência de 1 Hz e apresenta a cor laranja. O portão deve manter-se aberto durante 2min, podendo este tempo ser estendido por iguais períodos através da ativação do botão sw1 durante o seu movimento de fecho.



a) [2] Programe a função `void Set_Led(unsigned int cor)` para controlo do estado do LED L1, em que o parâmetro *cor* indica o estado pretendido, i.e. OFF, GREEN, RED ou ORANGE. Pretende-se que o código seja o mais eficiente possível e garanta que o LED não fica apagado entre mudanças de estado.

Nota: O LED apresenta a cor verde quando o pino G está ativo e a cor vermelha quando o pino R está ativo. A ativação simultânea de ambos os pinos faz o LED acender com a cor laranja.

b) [2] Considerando que a frequência do sinal de relógio aplicado ao microcontrolador LPC2106 é de 5MHz, programe a função `TIMER1_Init()` para iniciação do `Timer1` com o valor adequado à sua utilização como temporizador da aplicação. Justifique a sua resposta.

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

d) [2] Apresente o troço de código C que implementa a solução descrita na alínea anterior. Para tal, considere a existência das seguintes funções:

- `int Key_Press()`, que retorna 1 quando o botão sw1 está pressionado e 0 no caso contrário;
- `int Get_Sensor()`, que retorna 1 quando o portão está em movimento e 0 no caso contrário;
- `unsigned int TMR1_Elapsed(unsigned int lastRead)`, que devolve o tempo decorrido, em unidades de contagem (`ticks`), desde `lastRead` até ao instante atual.