

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA  
LEETC, LEIC, MEET, MEIC  
**Sistemas Embebidos I**  
Semestre de Inverno de 2010/2011

## Quarta actividade prática

### Teclado

Programe as funções `void keyboard_init()`, `int keyboard_pressed()` e `int keyboard_read()` para acesso ao teclado de matriz usado na segunda actividade prática.

A função `keyboard_init` fará as iniciações necessárias.

A função `keyboard_pressed` verifica se há tecla premida.

A função `keyboard_read` devolve o código da tecla premida ou -1 se não existir.

A interpretação do teclado deve tratar o *bounce* produzido pelos contactos mecânicos das teclas.

As funções `keyboard_read` ou `keyboard_pressed` só devem assinalar novo carácter apenas quando o teclado passa do estado de "nenhuma tecla premida" para "tecla premida". Se estas funções forem chamadas uma segunda vez ainda com a mesma tecla premida não devem assinalar nova tecla.

Use o seguinte programa de teste, que faz piscar o LED o número de vezes correspondente à tecla premida.

```
#include "lpc2106.h"
#include "chrono.h"
#include "led.h"
#include "keyboard.h"

int main() {
    lpc2106_init();
    chrono_init();
    led_init();
    keyboard_init();
    while (1) {
        int tecla;
        while ( ! keyboard_pressed())
            ;
        tecla = keyboard_read();
        if (tecla != -1) {
            for (i = 0; i < tecla - '0'; ++i) {
                led_write(1);
                chrono_delay(500);
                led_write(0);
                chrono_delay(500);
            }
        }
    }
}
```

Ficheiros intervenientes:

```
sel
|
+-- code
|   +-- app
|       |   +-- keyboard
|       |       +-- main.c
|       |       +-- makefile
|       |
|       +-- mylib
|           +-- cstart.s
|           +-- ldscript
|           +-- keyboard.c
|           +-- keyboard.h
|           +-- chrono.c
|           +-- chrono.h
|           +-- led.c
|           +-- led.h
|           +-- lpc2106.c
|           +-- lpc2106.h
```

## Programas para ROM - parte 2

O desenvolvimento deste ponte é baseado no documento "Programas para ROM - parte 2".

O resultado final deste exercício é a criação de uma base (*framework*) que permita o desenvolvimento de programas para execução em RAM - situação de teste ou em ROM - situação de produto final.

Para concretizar este objectivo é necessário intervir nos ficheiros **cstart.s**, **ldscript** e **makefile**.

No módulo **cstart.s** é necessário acrescentar a cópia da secção **.data** da ROM para a RAM.

Sugere-se a criação do ficheiro **ldscript\_ram** para a geração de programas para RAM e a criação do ficheiro **ldscript\_rom** para a geração de programas para ROM.

O **makefile** deve receber um parâmetro que permita escolher entre a geração para RAM ou para ROM.

Use como ponto de partida o exercício anterior.

```
sel
|
+-- code
|   +-- app
|       |   +-- keyboard
|       |       +-- main.c
|       |       +-- makefile *
|       |
|       +-- mylib
```

```
+-- cstart.s *
+-- ldscript_ram *
+-- ldscript_rom *
+-- keyboard.c
+-- keyboard.h
+-- chrono.c
+-- chrono.h
+-- led.c
+-- led.h
+-- lpc2106.c
+-- lpc2106.h
```

Depois de concluída a actividade prática deve entregar ao docente a listagem dos programas realizados. Entregue por email e segundo a estrutura aconselhada em "Ambiente de trabalho".

29/9/2010  
Ezequiel Conde, Pedro Sampaio