

# Sistemas Embebidos I

Ano Letivo 2014/2015 - Semestre de Inverno

## Lab4 - Teclado e LEDs

### 1 Introdução

Este trabalho de laboratório tem como principal objetivo o estudo e desenvolvimento de gestores de periférico com funcionamento por pesquisa de estado, suportados no GPIO do microcontrolador LPC2106[D1] da NXP.

### 2 Montagem do *target*

Desenhe o esquema de ligações de um circuito composto pelo microcontrolador LPC2106, um teclado com 16 teclas, organizadas em matriz de 4x4 teclas, e um LED. Ambos os periféricos devem ser acessíveis através do GPIO do microcontrolador.

Nesta montagem deve ter em atenção as seguintes restrições:

- os pinos P0.17 a P0.31 do microcontrolador LPC2106 não estão acessíveis;
- os andares de saída dos pinos P0.2 e P0.3 do microcontrolador LPC2106 apresentam saídas *open collector*;
- o pino P0.14 do microcontrolador LPC2106 não deve ser usado.

### 3 Gestor de periférico do teclado

Implemente em linguagem C a seguinte API de acesso ao teclado [D11]:

```
void KBD_Init(unsigned int layout[16]);  
/* Faz a iniciação do sistema para permitir o acesso ao periférico. Para tal, recebe como parâmetro o vetor com os códigos das 16 teclas (layout). */  
int KBD_Hit(void);  
/* Devolve 1 se existe alguma tecla pressionada e 0 no caso contrário. Não é bloqueante. */  
int KBD_Read(void);  
/* Devolve o código da tecla pressionada. É bloqueante. */
```

### 4 Gestor de periférico do LED

Implemente em linguagem C a seguinte API para manipulação do LED:

```
void LED_Init(int pinId, int state);  
/* Faz a iniciação do sistema para permitir a manipulação do estado LED ligado bit pinId do GPIO. Deixa o LED apagado quando state toma o valor 0 ou aceso no caso contrário. */  
int LED_GetState(void);  
/* Devolve 0 se o LED está apagado e um valor diferente de zero no caso contrário. */  
void LED_SetState(int state);  
/* Apaga o LED se state toma o valor 0 ou acende o LED no caso contrário. */
```

### 5 Programa de teste

Escreva um programa em linguagem C que detete o pressionar das teclas e faça o LED piscar o número vezes correspondente ao código da tecla pressionada. O LED deve piscar à frequência de

2,5Hz e apenas de cada vez que uma tecla nova é pressionada.

Considere que o gestor de periférico do teclado é iniciado com os seguintes parâmetros:

```
unsigned int layoutTest[] = {5,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21};
```

Para contar o tempo que o LED deve estar acesso e apagado pode utilizar a função **delay** que implementou no terceiro trabalho de laboratório.

## **6 Referências**

As referências indicadas no documento estão disponíveis na página da unidade curricular SE1 na plataforma Thoth (secção Bibliografia).