PCO003

Lab 4 Criando um device driver de acordo com o padrão

1. Faça o download, abra o projeto pco003-Base.X no MPLABX e execute o programa.
2. Crie um device driver para o display de LCD.
   1. Crie os arquivos drv\_lcd.c e drv\_lcd.h. Utilize a biblioteca lcd como base.
   2. Para efeito de simplificação o driver pode apenas chamar as funções da biblioteca.
3. Crie um relógio
   1. A contagem de tempo deve ser feita em um processo dedicado. Devem ser criadas variáveis para segundos, minutos e horas.
   2. Um segundo processo deve ser executado a cada 1 segundo para escrever a hora correta. Lembre-se de limpar o LCD ou reescrever sempre na mesma posição
4. Crie também o driver para o teclado (\*Bonus)
   1. Crie um processo para executar a função de debounce e outro para verificar a tecla pressionada (para incrementar/decrementar a contagem de tempo).

Exemplo de uso de uma biblioteca:

void main**(**void**)** **{**

driver**\*** led**;**

led **=** getLedDriver**();**

led**->**initFunc**(**0**);**

led**->**funcoes**[**LED\_SET**](**0xaa**);**

**for(;;);**

**}**

Exemplo de biblioteca

**DD\_Types.h**

#ifndef DD\_TYPES\_H

#define DD\_TYPES\_H

//Device Drivers Types (dd\_types.h)

//ptr. de func. para uma função do driver

**typedef** char**(\***ptrFuncDrv**)(**void **\***parameters**);**

//estrutura do driver

**typedef** struct **{**

char id**;**

ptrFuncDrv **\***funcoes**;**

ptrFuncDrv initFunc**;**

**}** driver**;**

//função de retorno do driver

**typedef** driver**\*** **(\***ptrGetDrv**)(**void**);**

#endif /\* DD\_TYPES\_H \*/

**DRV\_LED.h**

#ifndef drvGeneric\_h

#define drvGeneric\_h

#include "dd\_types.h"

//lista de funções do driver

enum **{**

LED\_SET**,**LED\_FLIP**,** LED\_END

**};**

//função de acesso ao driver

driver**\*** getGenericDriver**(**void**);**

#endif // drvGenerico\_h

**DRV\_LED.c**

#include "kernel.h"

#include "pic18f4520.h"

#include "drvGeneric.h"

static driver myself**;**

static ptrFuncDrv my\_funcs**[**LED\_END**];**

char changePORTD**(**void **\***parameters**)** **{**

PORTD **=** **(**char**)** parameters**;**

**return** SUCCESS**;**

**}**

char inverte**(**void **\*** parameters**){**

PORTD **=** **~**PORTD**;**

**}**

char initGenerico**(**void **\***parameters**)** **{**

TRISD **=** 0x00**;**

myself**.**id **=** **(**char**)** parameters**;**

**return** SUCCESS**;**

**}**

driver**\*** getGenericDriver**(**void**)** **{**

myself**.**initFunc **=** initGenerico**;**

my\_funcs**[**LED\_SET**]** **=** changePORTD**;**

my\_funcs**[**LED\_FLIP**]** **=** inverte**;**

myself**.**funcoes **=** my\_funcs**;**

**return** **&**myself**;**

**}**