EC45C - Sistemas Microcontrolados
Turmas: C51A/C51B

Prática L3 – Interface com LCD alfanumérico.

Objetivos:

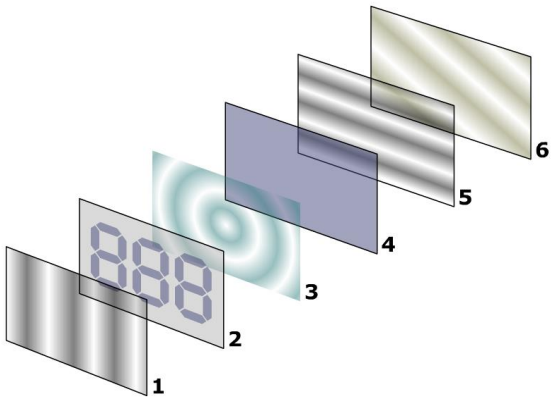
- Criar um *firmware* com foco em:
  - leitura de teclado matricial através de varredura;
  - apresentação da informação em LCD alfanumérico.
- Simular o *firmware* no Proteus.
- Testar o firmware no kit didático XM118.
- Agrupar os arquivos gerados para envio pelo Moodle.

LCD Alfanumérico

Display de cristal líquido (do inglês Liquid Crystal Display - LCD), é um dispositivo eletrônico-óptico usado para exibir informações por via eletrônica na forma de texto e imagens. No caso do LCD alfanumérico, sua forma de apresentação se restringe a letras, números e algumas poucas figuras denominadas caracteres especiais.

Por ser uma forma econômica de apresentar informações complexas ao operador, é comum encontrar LCD alfanuméricos em diversos dispositivos como: relógios, calculadoras, leitores de cartão magnético, balanças comerciais, painéis de instrumentos (interface homem-máquina) e em outros mais. Além disso, seu baixo consumo de energia elétrica lhe permite ser utilizado em equipamentos portáteis alimentados por bateria.

Um LCD consiste de um elemento polarizador de luz, eletricamente controlado, que se encontra comprimido dentro de células entre duas lâminas transparentes polarizadoras. Os eixos polarizadores das duas lâminas estão alinhados perpendicularmente entre si. Cada célula é provida de contatos eléctricos que permitem que um campo elétrico possa ser aplicado ao cristal líquido.



1- Polarizador vertical
2- Eletrodos frontais
3- Filme de cristal líquido
4- Eletrodos traseiros
5- Polarizador horizontal
6- Espelho

Figura 1: Formação física do LCD.

Este dispositivo eletrônico-óptico modulado é composto por um determinado número de pixels, preenchidos com cristais líquidos e dispostos em frente a uma fonte de luz, que, ao combinar quais pixels estão ligados e quais estão desligados é possível produzir letras, números e alguns caracteres especiais programados na memória do controlador do LCD.

Os modelos mais modernos de LCD alfanuméricos possuem um controlador interno, normalmente baseado no microcontrolador Hitachi HD44780 ou no Samsung KS0066U. O controlador é

montado no módulo de LCD, permitindo a comunicação com um o microcontrolador através de um barramento de dados (4 ou 8 bits) e das linhas de controle RS, R/W e EN.

EN é a linha de controle que habilita o funcionamento do LCD.

RS é a linha de controle que seleciona o tipo de informação disponível no barramento de dados.
RS = 0 → Comandos: instruções de controle para o LCD;
RS = 1 → Dados: caracteres a serem escritos.

R/W é a linha de controle que define se a informação disponível no barramento de dados é do tipo escrita (para ser interpretada pelo LCD) ou do tipo leitura (para ser interpretada pelo µC).
R/W= 0 → seleção de escrita no LCD;
R/W= 1 → leitura do LCD.
Opcionalmente R/W pode ser conectada ao GND quando não se deseja ler dados do LCD.

No Kit XM118 o LCD está conectado ao µC na seguinte configuração:

Pino	Símbolo	Função	Ligação com o PIC
1	Vss	GND	-
2	Vdd	+5V	-
3	Vo	Controle de contraste	-
4	RS	Seleção de modo	PORTE, 0
5	R/W	Leitura/Escrita	PORTE, 2
6	EN	Habilitação	PORTE, 1
7	D0		PORTD, 0
8	D1		PORTD, 1
9	D2		PORTD, 2
10	D3	Via de	PORTD, 3
11	D4	Dados	PORTD, 4
12	D5		PORTD, 5
13	D6		PORTD, 6
14	D7		PORTD, 7

O IDE MPLAB fornece uma biblioteca para acesso ao LCD chamada External LCD (ou apenas XLCD). Esta biblioteca foi modificada de forma a possui as seguintes funções:

Função	Protótipos	Descrição
BusyXLCD	`unsigned char BusyXLCD(void);`	Retorna 1 se o LCD estiver realizando alguma operação e 0 se estiver livre para receber novos comandos
OpenXLCD	`void OpenXLCD(unsigned char lcdtype);`	Inicializa o LCD
putcXLCD	`void OpenXLCD(unsigned char lcdtype);`	Escreve um caractere no LCD
putsXLCD	`void putsXLCD(char *buffer);`	Escreve uma string (da memória RAM) no LCD.
putrsXLCD	`void putrsXLCD(const rom char *buffer);`	Escreve uma string (da memória FLASH) no LCD
ReadAddrXLCD	`unsigned char ReadAddrXLCD(void);`	Le o byte de endereço da memória do controlador de LCD
ReadDataXLCD	`char ReadDataXLCD(void);`	Le um byte do controlador de LCD
SetCGRamAddr	`Void SetCGRamAddr(unsigned char addr);`	Aponta o endereço do gerador de caracteres
SetDDRamAddr	`void SetDDRamAddr(unsigned char addr);`	Aponta para um endereço de dados do LCD
WriteCmdXLCD	`void WriteCmdXLCD(unsigned char cmd);`	Escreve um comando no LCD
WriteDataXLCD	`void WriteDataXLCD(char data);`	Escreve um byte no LCD

Os comandos que o LCD interpreta através da função **WriteCmdXLCD** são os seguintes:

Descrição do Comando	Modo	RS	R/W	Código (Hexa)
Controle do display	Ativo (sem cusor)	0	0	0C
	Inativo	0	0	0A, 08
Limpeza do Display com retorno do cursor	0	0	0	01
Retorno do cursor à 1ª linha e da mensagem à sua posição inicial	0	0	0	02
Controle do Cursor	Ativo (ligado, fixo)	0	0	0E
	Inativo	0	0	0C
	Alternado	0	0	0F
	Desloc. à esquerda	0	0	10
	Desloc. à direita	0	0	14
	Retorno	0	0	02
	Piscante	0	0	0D
Sentido de deslocamento do cursor na entrada de um novo caractere	Para esquerda	0	0	04
	Para direita	0	0	06
Deslocamento da mensagem com a entrada de um novo caractere	Para esquerda	0	0	07
	Para direita	0	0	05
Deslocamento da mensagem sem entrada de novos caracteres	Para esquerda	0	0	18
	Para direita	0	0	1C
Endereço da primeira posição (à esquerda)	1ª Linha	0	0	80
	2ª Linha	0	0	C0

Utilização do LCD com a biblioteca mod_xlcd.c

Para inicializar o LCD utilize o seguinte código:

```
//Inicialização do LCD
OpenXLCD(FOUR_BIT & LINES_5X7); // Modo 4 bits de dados e caracteres 5x7
WriteCmdXLCD(0x01);             // Limpa o LCD com retorno do cursor
Delay1KTCYx(10);                // Atraso de ~2ms para aguardar a execução do comando
```

Para escrever no LCD, selecione a posição de memória do primeiro caractere que se deseja escrever e em seguida envie o caractere ou a *string* para o LCD. Ex:

```
char palavra[] = "Amigos";      // declara a string palavra

WriteCmdXLCD(0x80);             // Seleciona a posição Coluna 1 e Linha 1
putrsXLCD ("Adios");            // Escreve Adios
WriteCmdXLCD(0xC0);             // Seleciona a posição Coluna 1 e Linha 2
putsXLCD (palavra);             // Escreve Amigos
putcXLCD ('!');                 // Escreve o caractere !
```

No LCD alfanumérico 16x2, as posições de memória para escrita dos caracteres obedecem a seguinte ordem em valores hexadecimal:

Coluna → / Linha ↓																
	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

ATIVIDADES

1- Circuito de simulação.

Acesse a plataforma Moodle e faça o *download* da biblioteca NXLCD.
Salve os arquivos no seguinte diretório de trabalho:
C:\{Disciplina}_{Turma}_{Bancada_{Número do roteiro}
ex: C:\EC45C_C51A_B1_L3

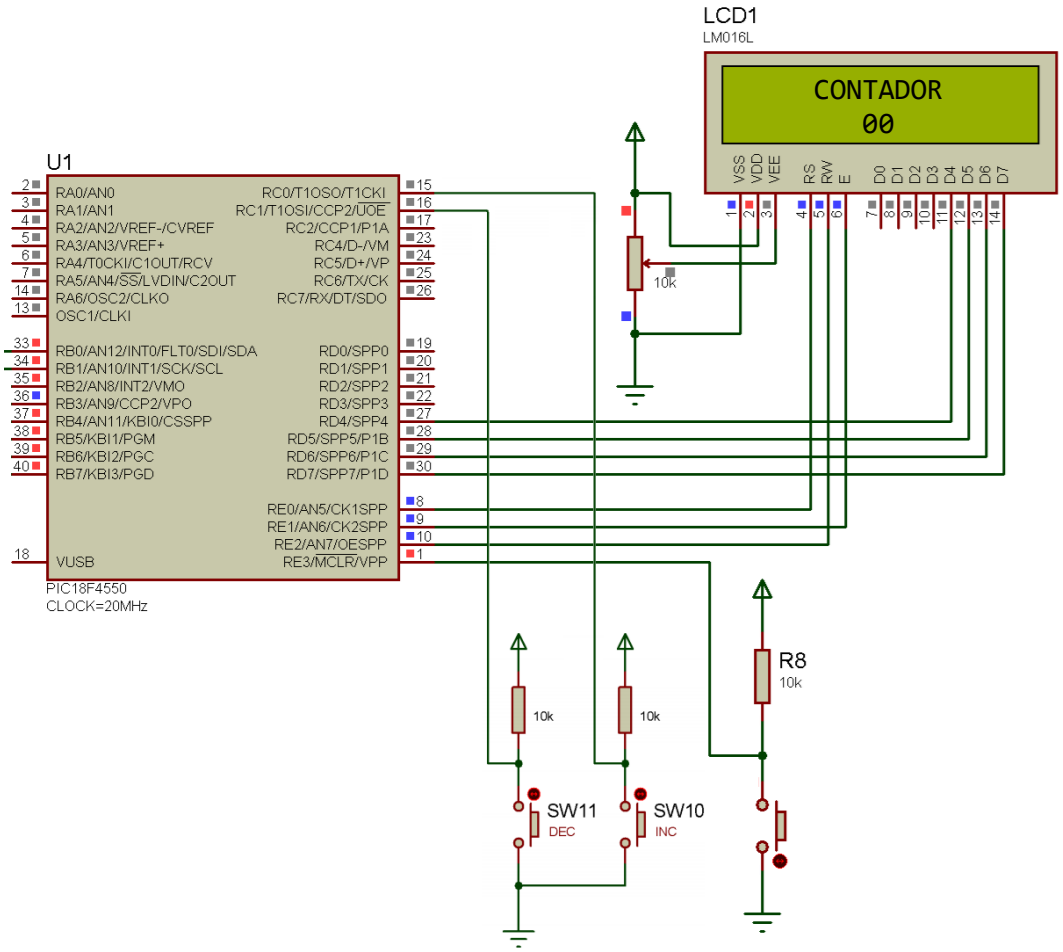


Figura 3: Circuito da atividade L3.

2- Desenvolvimento do *firmware*.

Crie um novo projeto no MPLAB X denominado L3 e salve-o no diretório de trabalho criado na atividade anterior.

Faça o *download* da biblioteca nxlcd e salve os arquivos no diretório do projeto.

Desenvolva um *firmware* capaz de fazer a leitura dos botões SW10, SW11 e apresentar o valor de um contador no centro da segunda linha do LCD. Na primeira linha deve aparecer a palavra “CONTADOR”, centralizada. Na segunda linha deve aparecer o valor do contador com 2 dígitos centralizados. O botão SW10 deve incrementar o contador e o botão SW11 deve decrementar o conrador.

Configure o display LCD para o modo 4 bits.

Observações sobre o kit didático XM118:
Possui um cristal de 20 MHz. Para o uso do teclado, após a gravação do código de máquina, posicione as chaves 1234 do *dipswitch* ICSP na posição OFF. Lembre-se de retornar chaves 1234 do *dipswitch* ICSP na posição ON para fazer nova gravação quando necessário.

Utilize o seguinte código para iniciar seu projeto:

```c
#include <xc.h>
#include "nxlcd.h"

// Configurações
#pragma config PLLDIV = 5          // PLL para 20MHz
#pragma config CPUDIV = OSC1_PLL2  // PLL desligado
#pragma config FOSC = HS           // Fosc = 20MHz; Tcy = 200ns
#pragma config WDT = OFF           // Watchdog timer desativado
#pragma config PBADEN = OFF        // Pinos do PORTB começam como digitais
#pragma config LVP = OFF           // Desabilita gravação em baixa tensão
#pragma config DEBUG = ON          // Habilita debug
#pragma config MCLRE = ON          // Habilita MCLR e desabilita RE3 como I/O

#define _XTAL_FREQ 20000000
```

OBS: Incluir o arquivo “nxlcd.c” na pasta *Source Files* do projeto.

3- Simulação do *firmware* gerado no Proteus.

Simule o *firware* no Proteus e apresente o funcionamento ao professor.

4- Gravação e execução do código de máquina no microcontrolador

Grave o código de máquina no microcontrolador e apresente o funcionamento para o professor.

5- Envio dos resultados para plataforma Moodle.

Compacte o diretório de trabalho com o projeto do *firmware* L3 em um arquivo .zip.

Nomeie o arquivo obedecendo o seguinte formato:
{Disciplina}_{Turma}_{Bancada}_{Número do roteiro}.zip
ex: EC45C_C51A_G1_L3.zip

Envie o arquivo compactado acessando a atividade “L3”.