Módulo 3 - Displays LCD

Agora que já entendemos bem como os pinos de I/O do microcontrolador funcionam, podemos utilizar uma forma mais profissional de se mostrar dados que vêm de dentro do microcontrolador. Por isso, utilizaremos o display LCD.

Os displays LCD recebem comandos em 8 bits ou 4 bits e os reproduzem na forma de caracteres. Esses comandos podem alterar algumas configurações também, como posição do cursor ou conteúdo da memória.

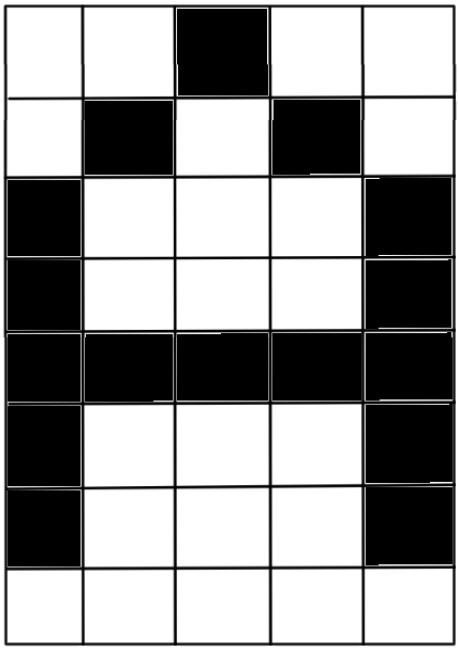
O display LCD

Existem vários tipos de display. O modelo que utilizaremos é um 16x2 com backlight azul, mas também é comum encontrar displays com blacklight amarelo ou com formato 16x4. Todos possuem modo de utilização idênticos.

Pino	Função	Descrição
1 - VSS	Alimentação	GND
2 – VDD	Alimentação	+5V
3 – V0	Contraste	Liga com um resistor variável ao GND para controlar o contraste do display
4 – RS	Modo de comando	1 - Caractere, 0 - Comando. Liga no microcontrolador e define o estado do pino via software.
5 – RW	Leitura ou Escrita	1 - Leitura da memória, 0 - Escrita no display. Geralmente mantido em 0.
6 – E	Clock	Executa o comando enviado ao display. Ligado ao microcontrolador.
D0 → D7	Barramento de Dados	8 bits de dados
15 – A	Anodo	
16 – K	Catodo	

Dizer que o display é 16x2 ou 16x4 está relacionado a quantidade de caracteres que ele pode mostrar de uma só vez. Mas cada um desses caracteres são compostos de uma matriz de pixels 8x5 que criam estes caracteres.

Para mostrar um caractere, é necessário acender cada um dos pixels afim de formar o caracter em sim, como é o caso do caracter A:



A ultima linha seria reservada ao cursor, por isso nosso caractere fica com um tamanho de 7x5. Então, assim como no acendimento dos leds, teríamos que nos preocupar em criar código que gerenciasse eles pixels 1 a 1 para trabalhar com caracteres. Felizmente, isso não é necessário.

Displays LCD possuem um controlador interno com uma memória (ROM) com todas as matrizes de caracteres definidas, além de comandos internos de endereçamento e manipulação de caracteres. Para acessá-los, basta utilizar o barramento de dados do display D<0:7>.

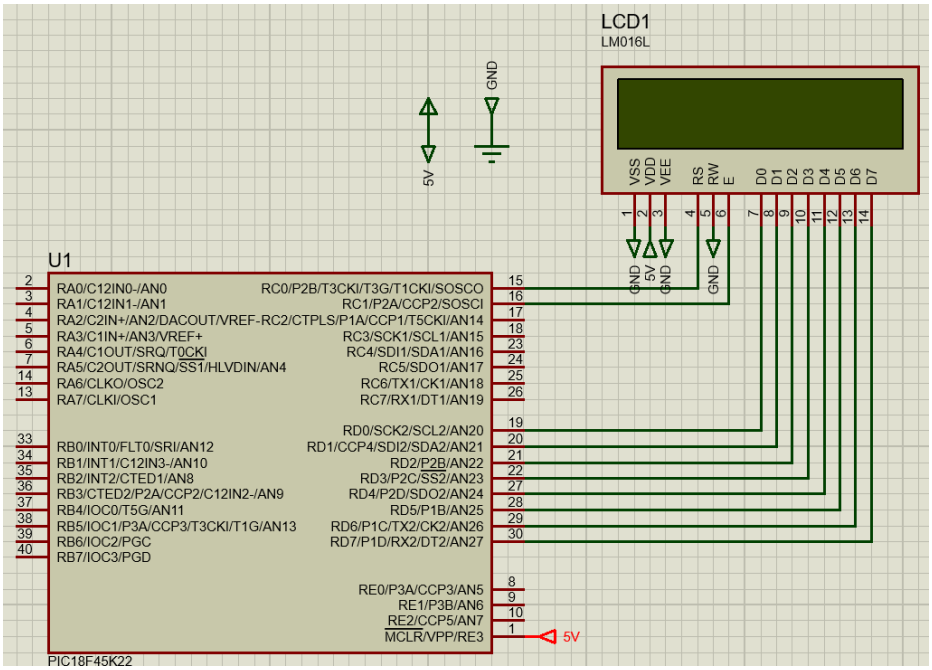
A memória ROM é totalmente endereçada com 8 bits. Cada endereço possui uma matriz de pixels previamente definida. Ao enviar o endereço pelo barramento de dados, você terá um caractere sendo mostrado na tela.

Os caracteres do display seguem o mesmo código da tabela ASCII, o que torna possível enviar os caracteres diretamente pelo PORT do microcontrolador. Isso significa que, no seu código, você poderá utilizar os caracteres na forma como você está acostumado a fazer, facilitando seu trabalho.

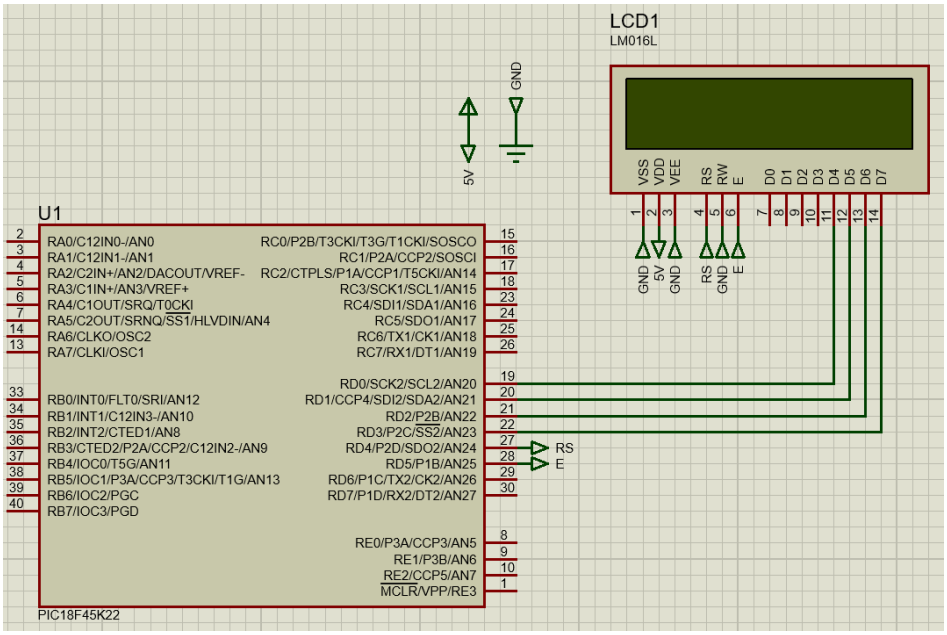
Para entender o funcionamento de uma forma didática, podemos utilizar um software simulador de LCD, disponível em: http://paginapessoal.utfpr.edu.br/rosangela/lcd-e-adc

Ligação do display no microcontrolador

O display possui suporte para ligação tanto em 8 bits quanto em 4 bits. Também existe a possibilidade de utilizar um conversor serial I2C. Abaixo é a ligação em 8 bits, veja que ela consome muitas vias de comunicação:



Mas a forma mais recomendada é a conexão em 4 bits. O circuito se torna mais simples e eficiente. Desta forma é necessária algumas modificações em software.



Características do display

Algums termos que você deve conhecer:

- A **CGRAM** é uma porção de memória interna para desenhar no display LCD;
- A **ROM** é a memória principal do display;
- O pino **E** funciona como clock do display. Ele é controlado pelo próprio microcontrolador;
- O pino **R/W** define as funcionalidades de Read e Write. Quando em 0, o display pode receber dados;
- O display pode receber comandos (inicialização, seta cursor, limpa display) ou caracteres (a, b, c) todos pelo barramento **D<0:7>**;
- O pino **RS** define se o display estará recebendo comandos ou caracteres. Já que o barramento de comunicação é somente 1, é preciso definir isto por outro pino.

Comandos principais do display

Abaixo você verá os comandos mais comuns no modo binário. As letras indicam as modificações que o comando representam

Modo de operação do display - 001ABC00

- A = (1) Modo 8 bits – (0) Modo 4 bits
- B = (1) Modo 2 linas – (0) Modo 1 linha
- C = (1) Matriz 10x5 – (0) Matriz 7x5

Controle do Display – 00001ABC

- A = (1) Liga display – (0) Desliga display
- B = (1) Liga cursor – (0) Desliga cursor
- C = (1) Cursor piscante – (0) Cursor normal

Controle automático do cursor – 000001AB

- A = (1) Desloca para a direita – (0) Desloca para a esquerda
- B = (1) Desloca automático - (0) Não desloca

Controle manual do cursor - 0001AB00

- A = (1) Desloca mensagem – (0) Desloca o cursor
- B = (1) Para a direita – (0) para a esquerda

Limpa e retorna cursor - 00000010

Reinicializa display – 00000001

CGRAM

A CGRAM é uma porção de memória que permite que você desenhe qualquer caractere para depois utilizá-lo na sua DDRAM. É muito útil quando você precisa mostrar algum tipo de dado na tela, e este dado não existe na ROM do display.

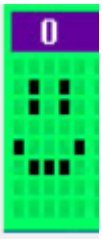
O comando para acessar a CGRAM inicia com o pino DB6 em nível lógico alto e os outros servem como endereçamento. Então, envie o comando 0x40 para acessar o primeiro endereço da CGRAM.

```
lcdComando(0, 0x40);
```

Agora, seu cursor está posicionado na CGRAM, você pode enviar no barramento cada pixel que formará o caractere desejado:

```
lcdComando(1, 0);
lcdComando(1, 0x0A);
lcdComando(1, 0x0A);
lcdComando(1, 0);
lcdComando(1, 0x11);
lcdComando(1, 0x0E);
lcdComando(1, 0);
lcdComando(1, 0);
```

O resultando destes comandos são:



Agora, posicione novamente o cursor na DDRAM visível do display e mande um caractere, mas desta vez referenciando aquele endereço de CGRAM que você desenhou:

```
lcdSetCursor(1, 7);
lcdComando(1, 0x00);
```