



UNIVERSIDADE FEDERAL DA PARAÍBA  
CENTRO DE INFORMÁTICA

# MÓDULO LCD I2C USANDO PIC12F675 E 74164

GABRIEL GUTIERREZ P. SOARES

JOÃO PESSOA  
2016

# PROJETO INTEGRADO I

## Módulo LCD I2C com PIC 12F675

Gabriel Gutierrez P. Soares

Microcontroladores 2015.2

Março de 2016

## 1 Descrição da atividade

### 1.1 Objetivo

Atividade integrada para familiarização do conjunto de instruções do PIC, gerenciamento das portas I/O, gerenciamento de tempo com e sem TIMER, gerenciamento de interrupções.

### 1.2 Contexto

Modelar, implementar e construir uma interface para permitir a comunicação de um display LCD (16X2) via protocolo I2C, utilizando o PIC12F675 e um registrador de deslocamento 74164.

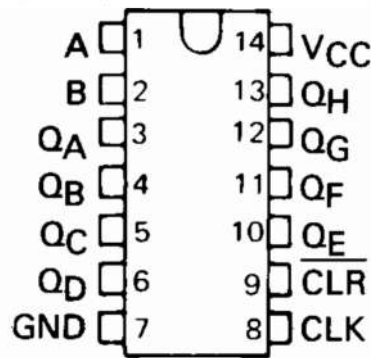
### 1.3 Especificações

1. A interface de ter apenas UM PIC12F675 e UM shift register 74164;
2. Uma placa de circuito impresso para esse fim específico deve ser elaborada e implementada;
3. A placa de circuito impresso deve ter 4 conectores para permitir a utilização dos seguintes modelos de LCD: 1602B, 1602E, 1602G e JHD539;
4. A placa de circuito impresso deve ter um conector para comunicação I2C e um conector para permitir a gravação in circuit (ICSP);
5. A placa de circuito impresso deve ter LEDs indicativos de funcionamento;
6. O endereço da interface será armazenado na posição 0x00 da EEPROM do PIC;
7. O PIC deve ser programado para trabalhar como SLAVE. Quando identificar o seu endereço no barramento I2C, o PIC deve enviar os comandos recebidos ao LCD;
8. Todas as linhas de controle para o registrador de deslocamento e para o LCD serão gerenciadas pelo PIC;
9. Demais características poderão ser acrescentadas/modificadas em função do andamento do projeto.

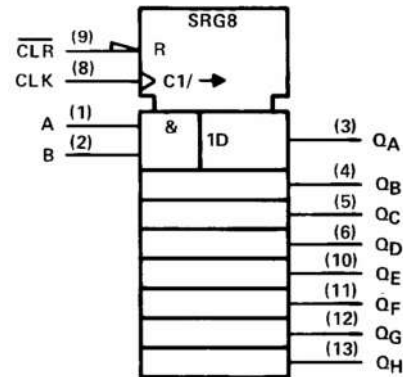
## 2 Execução

### 2.1 Shift register 74164

O 74164 é um registrador de deslocamento de 8 bits, com carregamento serial através de uma operação AND entre as entradas A e B a cada borda de subida na entrada CLK, saída paralela em  $Q_A \dots Q_H$  e reset assíncrono em  $\overline{CLR}$ .



(a) Pinagem



(b) Esquema lógico

Figura 1: Shift register 74164

Como, de acordo com o datasheet, a frequência de clock pode ser de até 25MHz (recomendada), a carga dos dados pode ser feita tranquilamente na frequência de execução de instruções do PIC, 1MHz.

Neste projeto, foram utilizados os pinos GP0 para o clock e GP4 para os dados.

### 2.2 Displays LCD

Além dos displays presentes nas especificações, que são os 1602B/E/G, de 16 colunas x 2 linhas, e o JHD539, de 16 colunas x 4 linhas, foram testados também os modelos 16x2 1602M e 4DMC-162-3.<sup>1,2</sup>

Em todos os modelos compatíveis, foi identificada a mesma quantidade de pinos e função para cada número, conforme a Tabela 1. Porém, o layout dos pinos é diferente, e segue dois padrões: o da Figura 2a, onde os pinos são na parte de cima e a numeração é na sequência 1—16 (caso dos modelos 1602G/M e JHD539); e o da Figura 2b, com os pinos na parte inferior e na sequência 14—1, seguidos dos pinos do backlight, 15 e 16 (caso dos modelos 1602B/E e 1623).

O funcionamento do display, implementado neste trabalho, é regido de acordo com os procedimentos, tempos e conjunto de instruções definidos no datasheet do controlador HD44780, da Hitachi, reproduzidos por todos os displays testados.

A comunicação com o display é feita através do registrador de deslocamento, seguindo o mapeamento da Tabela 2. O sinal de *Enable* é enviado por um pino separado do PIC. O envio de um byte ao LCD ( $B < 7 \dots 0 >$ ) é feito em duas etapas, uma para cada nibble, como mostra a Figura 3), onde SRDAT e SRCLK são as linhas do 74164. O bit LED, quando ativo, liga o backlight do display.

<sup>1</sup>Modelos testados: GMD1602B, MGD1602B, FDCC1602E, FECC1602G, JHD539M8, WM-C1602M

<sup>2</sup>A referência '4DMC-1623STLDN', escrita na placa de um deles, não retornou nenhum resultado na pesquisa online. O modelo encontrado foi o '4DMC-162-3'.

Nº pino	1	2	3	4	5	6	7	...	14	15	16
Função	GND	V <sub>CC</sub>	V <sub>O</sub>	RS	R/W	E	D <sub>0</sub>	...	D <sub>7</sub>	LED+	LED-

Tabela 1: Funções dos pinos dos displays LCD

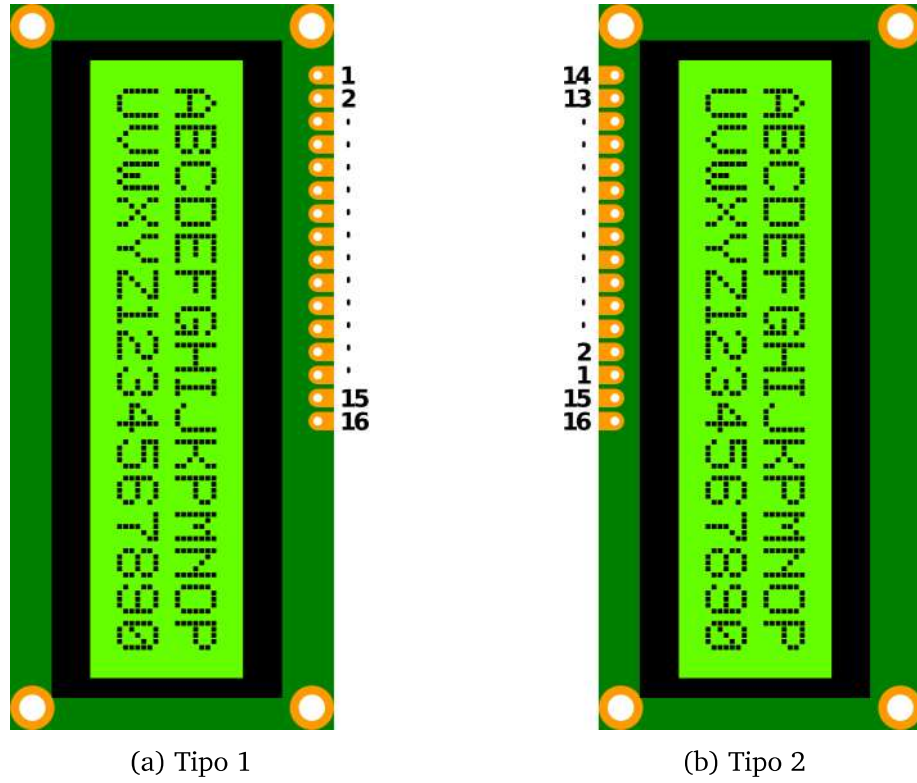


Figura 2: Layouts de pinos do LCD

74164	Q <sub>H</sub>	Q <sub>G</sub>	Q <sub>F</sub>	Q <sub>E</sub>	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
LCD	D7	D6	D5	D4	RS	LED	-nc-	-nc-

Tabela 2: Mapeamento dos pinos do 74164 para o LCD

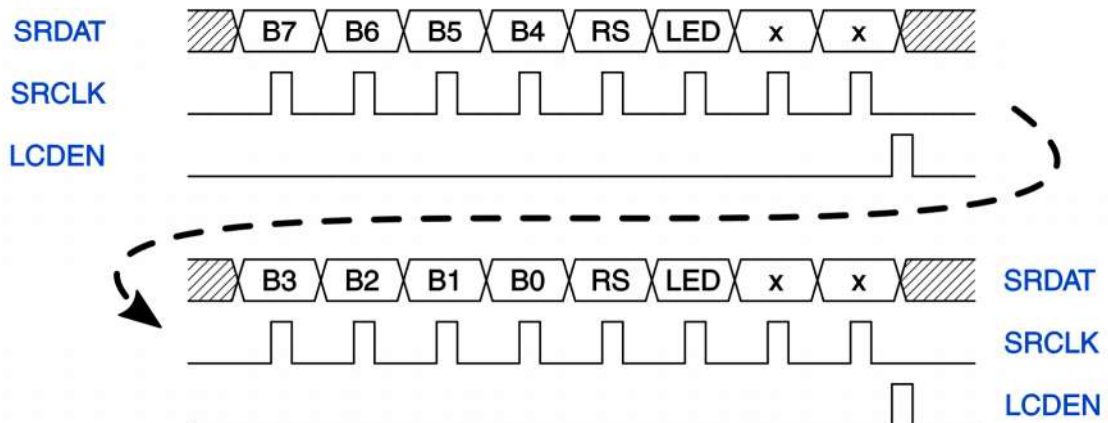


Figura 3: Diagrama de tempo do envio dos dados ao LCD

## 2.3 I2C

O I<sup>2</sup>C é um protocolo de comunicação serial síncrono criado pela Philips (agora NXP). Utiliza apenas duas linhas bidirecionais em coletor aberto, SDA (serial data) e SCL (serial clock), com resistores de pull-up. Os dispositivos conectados ao barramento são do tipo *master* (quem controla os dispositivos) ou *slave* (periféricos).

O estado natural do barramento é com SDA e SCL = 1. Uma transação é iniciada pelo *master* quando SDA desce enquanto SCL = 1. Em seguida, transmite-se o endereço de 7 bits do *slave* alvo (MSB first), seguido de um bit R/W que indica uma operação de leitura ou escrita. O *slave* que reconhecer seu endereço deve enviar um ACK (acknowledge), mantendo a linha SDA em 0 no 9º pulso de clock (Figura 4). Se for uma operação de escrita, o *master* começa então a enviar os dados, e para cada byte enviado o *slave* deve emitir um ACK. A transação é encerrada quando acontece uma borda de subida em SDA enquanto SCL está alto (Figura 5).

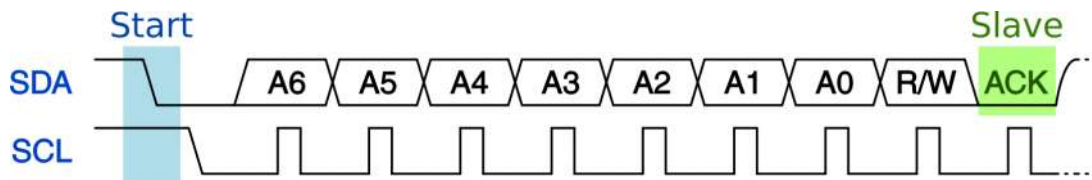


Figura 4: Endereçamento do I2C

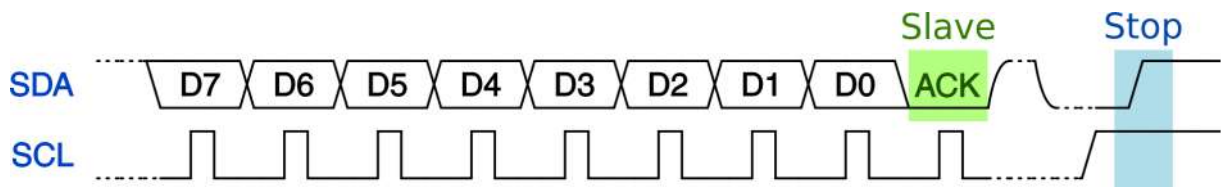


Figura 5: Envio de dados no I2C

Como o PIC 12F675 não implementa o protocolo I2C em hardware, a atividade requer uma implementação de um *slave* em software. Entretanto, após análise do código e testes práticos exaustivos, concluiu-se que não é possível implementar um *slave* compatível com a frequência padrão de clock do protocolo, 100kHz, visto que o PIC 12F675 executa suas instruções a 1MHz, insuficiente para executar as operações necessárias. A implementação foi validada com sucesso numa frequência menor, de 33kHz.

Os pinos escolhidos para o barramento foram o GP2 para o SCL, e GP1, para o SDA.

O endereço I2C do PIC foi gravado na primeira posição de memória da EEPROM, seguindo as especificações, e é carregado para a memória RAM na inicialização. Após o endereçamento pelo *master*, cada byte de dados é tratados da seguinte forma:

- Se dado < 0x20, é enviado como um byte de comando ao LCD (consultar datasheet do LCD para comandos disponíveis);
- Se dado < 0x80, é enviado como um caractere ASCII ao LCD;
- Se dado >= 0x80, o byte é interpretado como uma mudança de posição do cursor, conforme Figura 6.

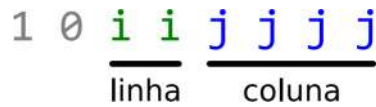


Figura 6: Byte de comando para posicionamento de cursor

## 2.4 Circuito

Visto que foram utilizados apenas displays com dois tipos de pinagens, dois conectores foram disponibilizados, e dispostos de forma a possibilitar a conexão simultânea de um display de cada tipo.

A pinagem do conector ICSP adotada é compatível com os gravadores Microchip PICkit e Kitsrus K150, seguindo a sequência {MCLR, Vdd, GND, PGD, PGC}.

Foram usados os pinos GP0 e GP4 para a comunicação com o 74164 e GP5 para o sinal de *Enable* do LCD. Para ligar o backlight do display, foi utilizado um circuito de chave com transistor. O contraste é ajustado através de um trimpot.

Os pinos GP1 e GP2 foram reservados para a comunicação I2C. Como GP1 também é utilizado pelo circuito ICSP, faz-se necessário desconectá-lo para programar o PIC. Ao pino GP3 foi conectado um botão para reset.

A alimentação pode ser provida pelo gravador, ou por fonte externa, através do conector J2. O jumper J1 serve para isolar a alimentação do gravador, caso seja adotada a outra opção.

Julgou-se desnecessário o uso de LEDs, visto que ambas as comunicações seriais são em alta frequência.

## 3 Considerações finais

A comunicação com o registrador de deslocamento foi feita de forma tranquila e sem problemas. A alta velocidade de trabalho do 74164 isenta-o de ser um dos gargalos do sistema, e o torna bastante atrativo quando se quer expandir a quantidade de saídas do PIC. Porém, uma opção melhor para tal função é o 74595, por ter um buffer interno que permite configurar as saídas antes de alterá-las. A ausência deste recurso se reflete no *backlight* do LCD, que pisca rapidamente toda vez que se envia dados ao LCD. Com o buffer seria possível também incluir o sinal de *Enable* no registrador de deslocamento.

A codificação da rotina de inicialização dos displays foi dificultada devido à informações incorretas encontradas em um dos datasheets, que apresentava uma sequência incorreta de configuração inicial. Superado este problema, as demais funções foram transcritas com fluidez.

A parte mais trabalhosa foi, sem dúvida, a implementação em software do I2C. Após diversas tentativas frustradas pela alta frequência do protocolo, foi-se constatado que, mesmo com um código mínimo, não é possível codificar uma implementação compatível com os 100kHz padrões para a linha de clock. Só a redução para 33kHz viabilizou este recurso.

Por fim, observou-se que a implementação do processamento dos comandos para o LCD foi limitada, visto que a memória de caracteres do LCD (CGRAM) é endereçável através de 8 bits. Utilizando apenas 7 bits (dado < 0x80), não é possível ter acesso à todos os caracteres disponíveis. Uma abordagem melhor seria a definição de um byte de comando NEXT\_IS\_CMD, que indique que o byte seguinte recebido seja enviado ao LCD como comando. Ou então a definição de dois bytes, LISTEN\_CMD e LISTEN\_DATA, que mudariam a maneira de interpretar os próximos bytes até que fosse enviado outro comando desse tipo (LISTEN).