



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SANTA CATARINA ENGENHARIA DE TELECOMUNICAÇÕES  
CAMPUS SÃO JOSÉ

# Sistema de Cadastro de Produtos Microcontrolador 8051

Membros da equipe:  
Jessica da Silva Hahn  
Letícia Aparecida Coelho

São José, 06 de Julho de 2015.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SANTA CATARINA ENGENHARIA DE TELECOMUNICAÇÕES –  
CAMPUS SÃO JOSÉ

# Sistema de Cadastro de Produtos Microcontrolador 8051

Projeto apresentado à disciplina de Microprocessadores do curso  
de Engenharia de Telecomunicações  
do IFSC – Campus São José  
utilizado como outro método avaliativo da disciplina.

Prof. Eraldo Silveira Silva

Membros da equipe:  
Jessica da Silva Hahn  
Letícia Aparecida Coelho

São José, 06 de Julho de 2015.

# 1 Introdução

O 8051 é um microcontrolador lançado pela Intel nos meados da década de 80, apesar de ser um processador antigo, possui a estrutura básica dos microcontroladores atuais, utilizando como linguagem de programação o assembly, podendo também ser programado em linguagem C. O projeto proposto foi um sistema de controle de estoque, responsável pelo cadastramento de até 10 peças com seus respectivos IDs e suas quantidade. O sistema propõe um menu inicial que disponibiliza funções de cadastro de novos produtos e alteração da quantidade do produto. O usuário entraria com essas informações no sistema através do teclado e estas serão mostrados no Display LCD.

## 2 Fundamentação Teórica

### 2.1 Interrupção

O microcontrolador utilizado para a implementação desse sistema foi o 8051, esse microcontrolador oferece três tipos de interrupções:

1. Interrupção Externa (INT0 e INT1): pino físico da porta P3 do 8051 que quando acionado externamente (setado para 0) para a execução do processo que estava sendo executado;
2. Interrupção por Temporizador (TIME/COUNTER): interrupção interna ao 8051 causado pelo estouro da flag TF0;
3. Interrupção pela Serial: interrupção também interna ao microcontrolador causada pelo periférico serial que é ligado a um determinado pino da porta P3.

No caso desse projeto foi necessário utilizar a interrupção INT0 causada pelo teclado, ou seja, cada vez que uma tecla for digitada pelo usuário causará a interrupção do sistema, cujo tratamento será feito pelo endereço 0003h, é neste endereço que há as instruções que serão executadas pelo 8051 cada vez que a interrupção for acionada.

### 2.2 Display LCD

Um display LCD é um tipo de hardware que possui internamente uma memória, são nesses endereços da memória que cada carácter é mostrado na tela do display, essas posições podem ser reprogramadas gerando caracteres especiais, como será visto posteriormente quando será abordado as animações. Um ponto importante que deve ser destacado é que o display mostra apenas caracteres que estejam no formato ASCII, portanto todo número que será digitado pelo teclado terá que ser convertido de acordo com essa tabela. Antes de qualquer carácter ser mostrado no display ele deve ser ligado e pré-programado

de acordo com os comandos a seguir, comandos este que já vem implementados no próprio display.

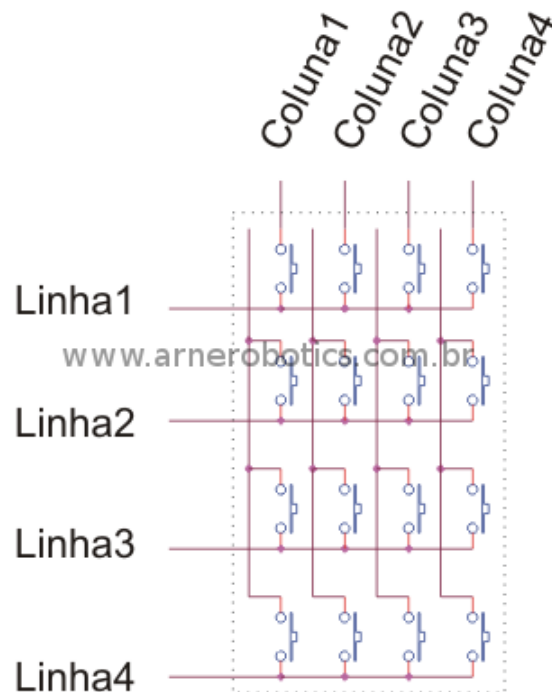
Alguns dos principais comandos em hexadecimal:

- 0CH - liga display
- 01H - limpa display com cursor na posição inicial
- 14H - desloca cursor para a direita
- 0DH - cursor piscante

—

## 2.3 Teclados em Sistemas com Microcontrolador

O teclado utilizado neste projeto é um modelo 4x3, um teclado armazena nível alto quando o botão não está clicado e nível baixo quando o botão está clicado, isto é visualmente fácil de entender quando analisamos a figura abaixo, onde podemos verificar que o teclado é um conjunto de chaves, as quais tem controle mecânico e são acionadas ao teclar no botão.



Vale salientar que o teclado não tem controle interno, portanto, seu controle

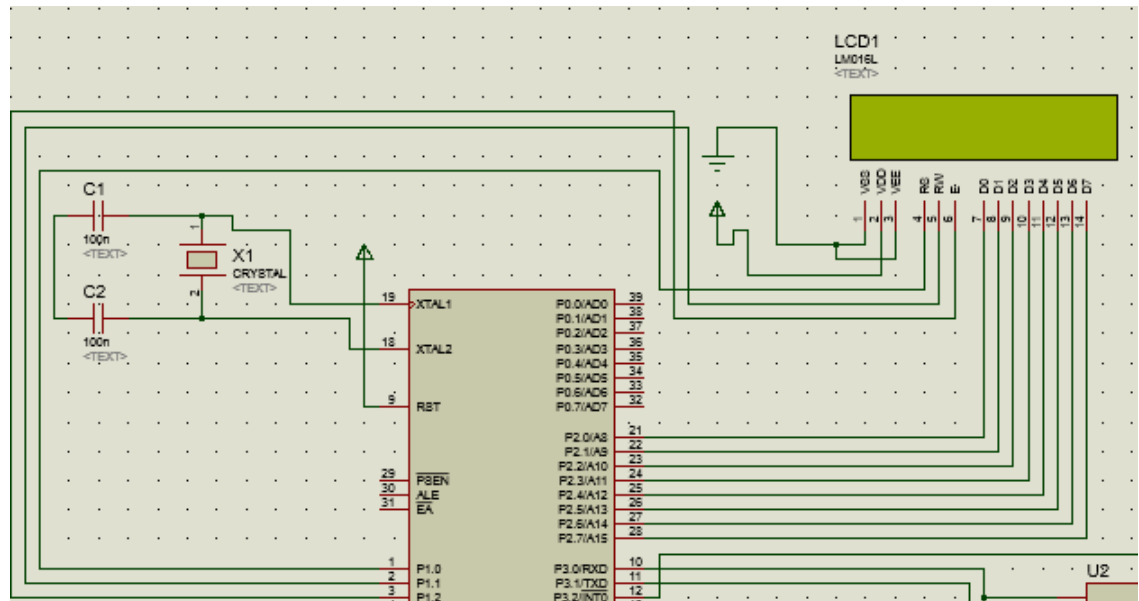
deve ser feito nos pinos onde cada saída do teclado estará conectada nas portas do microprocessador.

### 3 Sistema

#### 3.1 O Esquema de Hardware

#### 3.2 Display

Os pinos de controle como o RS (ativar comandos de controle), Rw (ativar comandos de escrita e leitura) e o E (enable) foram ligados aos pinos da porta P1, P1.0, P1.1 e P1.2 respectivamente. E os setes pinos responsáveis pela escrita da informação em si nos pinos da porta P2 (P2.0 á P2.7). O tipo de display LCD utilizado foi o 16x2, onde se tem 16 posições de memória em cada linhas, neste caso duas linhas.



#### 3.3 Teclado

O teclado utilizado foi o modelo para telefone com a limitação de dígitos numéricos de 0 á 9 e com caracteres especiais como o: asterisco(\*) e o cerquilha(#)é do tipo 4x3, ou seja, possui quatro linhas e três colunas, é através do teclado que será gerado a interrupção do sistema (ver tópico a seguir). As portas utilizadas para ligar as linhas do teclado foram as portas P1.3 á P1.6, as colunas foram ligadas pela P1.7, P3.0 e P3.1. O hardware do teclado esta demonstrado na

figura 1 abaixo.

### 3.3.1 Interrupção

Acionada pela porta lógica AND3 onde cada entrada da porta corresponde a uma coluna do teclado, quando alguma tecla for acionada provocará na coluna respectiva o valor zero, levando a saída dessa porta a zero ativando assim fisicamente a interrupção pelo pino INT0. O hardware esta demonstrado na figura 1.

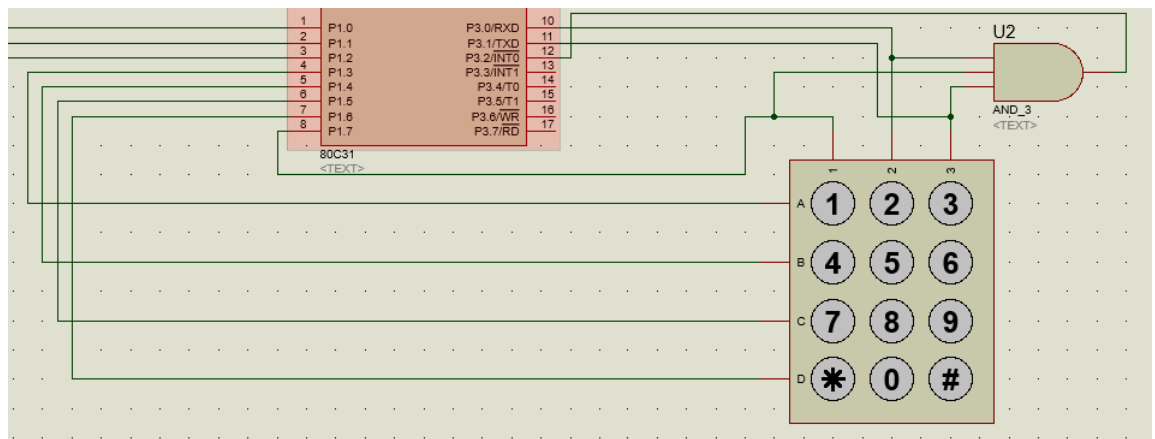
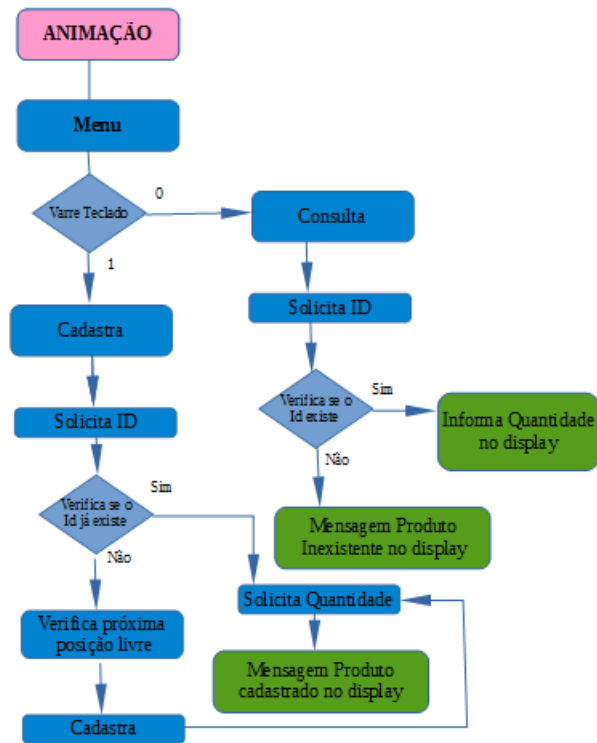


Figura 1: Teclado com interrupção T0

## 4 O Software

### 4.1 Sistema Geral

O fluxograma a seguir demonstra como seria a estrutura do programa de cadastramento de produtos com a integração do hardware do teclado e do display LCD.



## 4.2 Display

Basicamente ligamos o display e colocamos o cursor para a posição inicial através do comando 0EH e com a subrotina chamada controleLCD que joga o valor zero aos pinos RW e RS e ativa pela borda negativa o pino E. Após essa programação o display ficará em estado de espera até que alguma tecla seja digitada (função VARRETECLADO), após esse acontecimento o número digitado é transformado em hexadecimal de acordo com seu valor correspondente da tabela ASCII, caso nada for digitado o display ficará em estado de espera. A figura 2 representa essa implementação.

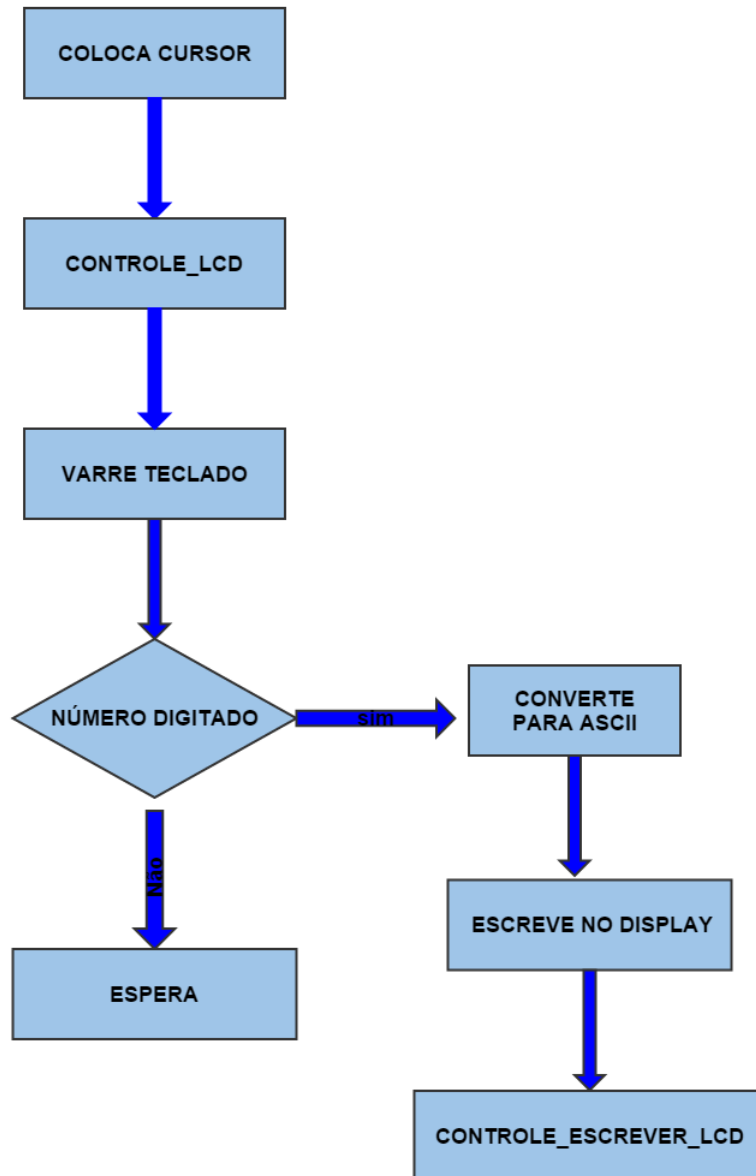


Figura 2: Fuxograma Display

#### 4.2.1 Animação

A CGRAM do display 16x2 possui 8 registradores reservados para caracteres especiais, sendo estes 40,48,50,58,60 e 68. Para utilizar estes caracteres especi-



ais é necessário reprogramar esta posição linha por linha, fazendo uma analogia com uma matriz 8x5 onde os bits que estiverem em 1 escrevem na tela e os bits que estiverem em 0 não aparecem na tela, conforme a figura abaixo que ao ser implementada mostraria um 'robo'.

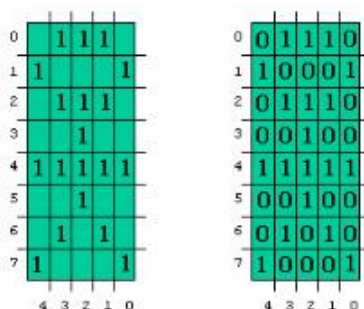


Figura 3: Robo

O escrita dos caracteres especiais é feita com o numero de acesso a memória escrito na tela, por exemplo, para escrever o caracter especial armazenado no endereço 40 é necessário escrever em modo dado 1 no display. Além disso o acesso as outros caracteres especiais pode ser feito segundo a tabela:

<b>40</b>	<b>1</b>
<b>48</b>	<b>2</b>
<b>50</b>	<b>3</b>
<b>58</b>	<b>4</b>
<b>60</b>	<b>5</b>
<b>68</b>	<b>6</b>

Neste projeto utilizamos os endereços 40, 48 e 50 da CGRAM reprogramando-os e formando 4 diferentes desenhos, conforme as figuras abaixo, além disto movimentamos os desenhos ao longo do display utilizando os endereços do display.

0					0h					0h	1	1	1	1	1Bh
1					0h					0h	1	1	1	1	1Fh
2					0h	1	1	1	1	1Bh		1	1	1	0Eh
3	1	1		1	1	1Bh	1	1	1	1	1	1	1	1	4h
4	1	1	1	1	1	1Fh	1	1	1	0Eh					0h
5		1	1	1	0Eh		1			4h					0h
6			1		4h					0h					0h
7					0h					0h					0h
4 3 2 1 0					4 3 2 1 0					4 3 2 1 0					
Coração_1					Coração_2					Coração_3					

Figura 4: Corações

A movimentação ao longo do display é efetuada utilizando o endereço de cada posição do display, incrementando até chegar ao final da primeira linha.

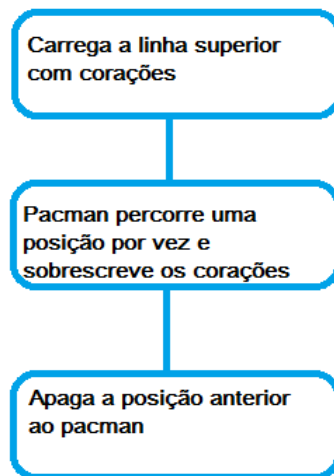
0							0h		1	1	1		0Eh
1		1	1	1			0Eh	1	1		1	1	1Bh
2	1	1		1	1		1Bh	1	1	1	1	1	1Fh
3	1	1	1	1	1		1Fh	1	1	1			1Ch
4	1	1					18h	1	1				18h
5	1	1	1	1	1		1Fh	1	1	1	1		1E
6		1	1	1			0Eh		1	1	1	1	0F
7							0h						0h
1	4	3	2	1	0			4	3	2	1	0	
Pac 1							Pac 2						

Figura 5: Pacman

Este endereçamento pode ser visualizado na figura a seguir :

<b>LCD 16x2</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>	<b>16</b>
linha 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
linha 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

A figura 5 mostra os dados utilizados para a reprogramação do endereço 40, sendo que este endereço tem 3 reprogramações consecutivas o que da a impressão de que os corações estão 'pulando'. Ao reprogramar mais de uma vez o mesmo endereço ele mudará os caracteres especiais varrendo todos os que estão reprogramados um atrás do outro. A figura 6 mostra os dados utilizados para a reprogramação do endereço 48, sendo que este endereço tem 2 reprogramações consecutivas o que da a impressão de que o pacman esta abrindo e fechando a boca. Além disto no endereço 50 inserimos um desenho em branco, o qual é responsável por limpar a tela após o pacman percorre-la. Para um entendimento básico do funcionamento da animação podemos analisar a figura abaixo :



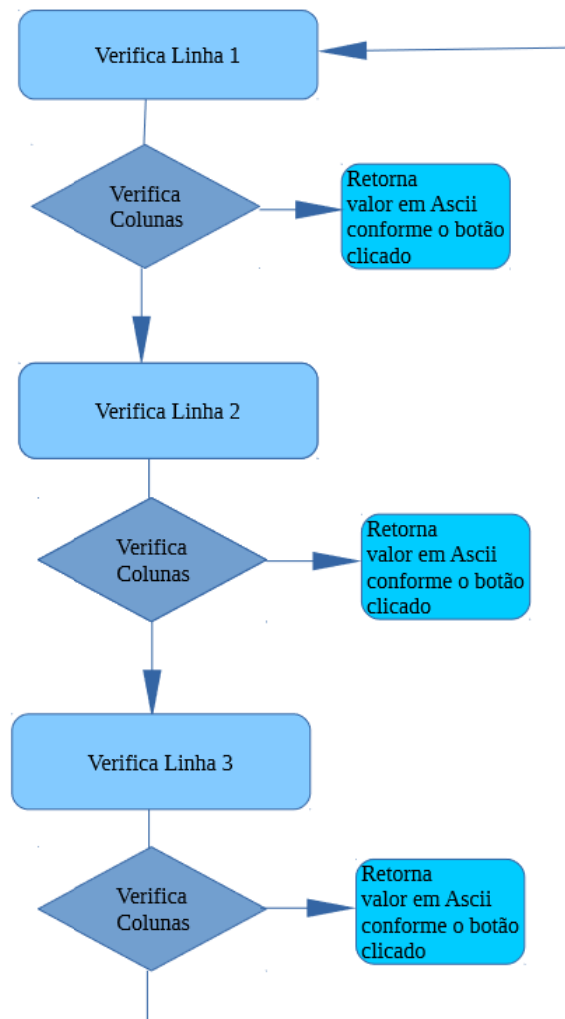
E além disto, podemos visualizar as figuras abaixo que demonstram a animação no display em ação:



### 4.3 Teclado

Podemos analisar de maneira simplificada no organograma abaixo a estrutura de programação do teclado. Basicamente, analisamos linha por linha colocando 0 na linha que esta sendo lida no momento e 1 nas outras linhas, para cada linha analisamos as colunas verificando-as. A coluna que estiver em 0 o que indica que o botão esta clicado. O conjunto entra linha e coluna com nível

lógico baixo (0) lidas direciona o programa para um retorno específico o qual envia o caracter em ascii para a verificação e continuação nos processos.



## 5 Testes Realizados

### 5.1 Display

Primeiramente ligamos o display e colocamos o cursor na primeira posição e mandamos escrever os seguintes caracteres: 'O', 'I' em cada posição diferente da memória e com a função mostrarEstring enviamos essa mensagem para o display, na última posição não mostrada na mensagem é guardado o valor zero é através desse valor que a função tem um controle sob a mensagem, pois mandamos através do endereçamento indireto o registrador R0 apontar para a posição inicial do carácter, após era feito o teste se o valor apontado por este registrador era zero(@R0), caso não fosse o R0 seria incrementado e testado novamente.

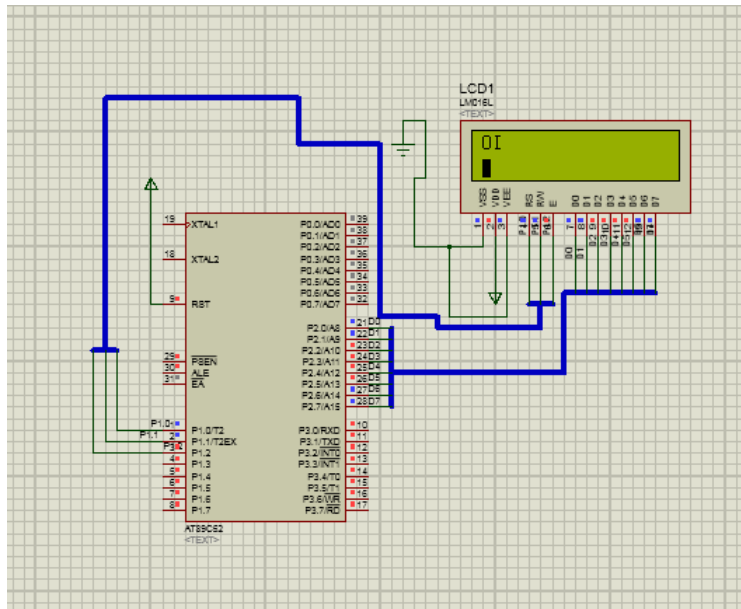


Figura 6: Teste com o display

## 5.2 Teclado

Após a integração do teclado no hardware do projeto e após feito a função responsável por varrer o teclado que retorna o número digitado por seu valor em ASCII, foi acrescentado na função que escreve o número no display o armazenamento desses valores na memória, utilizando três posições para esse armazenamento (20h à 23h) através desses registradores conseguimos escrever o número correspondente que foi armazenado como segue a imagem a seguir.

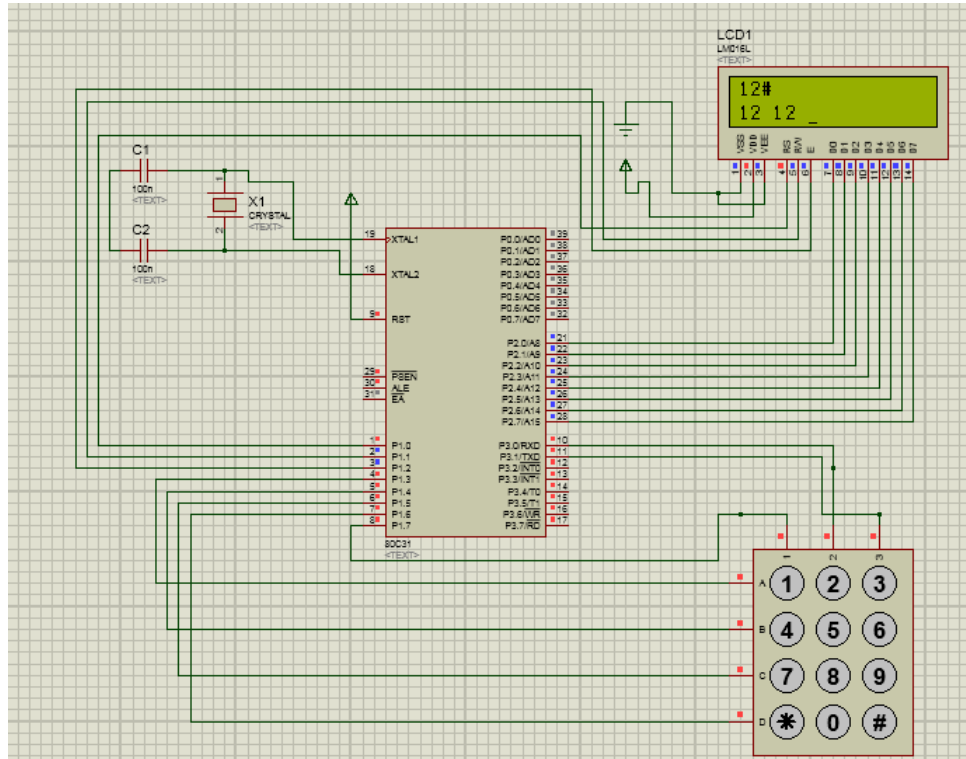


Figura 7: Teste com o display

## 6 Conclusão

Apesar de ser um microcontrolador antigo o 8051 possui um grande potencial, tendo as funções e características básicas dos processadores atuais, sendo de grande importância para o entendimento do funcionamento de um microcontrolador, o assembly que o 8051 implementa possui funções que foram essenciais na implementação desse projeto e facilitaram a programação, facilidade essa que não tínhamos quando fizemos o primeiro projeto dessa disciplina no processador BIP, processador didático que não possuía acesso a periféricos, sendo essa a principal diferença entre um processador em si e um microcontrolador. Mas a linguagem assembly é muito mais complexa que uma linguagem de programação de alto nível como o C e o Java, por exemplo, pois o assembly é uma linguagem de baixo nível que trabalha com bits e acesso direto a memória, por este motivo o 8051 assim como outros microcontroladores podem ser programados com outras linguagens de alto nível, o 8051 por exemplo, implementa a linguagem C, facilitando a programação desses microcontroladores. Devido ao tempo reduzido que tivemos para fazer esse projeto e por alguns problemas encontrados para integrar nosso sistema de cadastramento do projeto anterior, não conseguimos finalizar este projeto, pois tivemos problema para acessar e ler a opção que o usuário digitaria, 1 para consultar a quantidade e 0 para cadastrar.

## Referências

- [1] <http://microcontrolandos.blogspot.com.br/2012/10/biblioteca-mikroc-keypad.html>.
- [2] <http://repositorio.uniceub.br/bitstream/123456789/3303/2/20268113.pdf>
- [3] <http://www.arnerobotics.com.br/eletronica/MicrocontroladorPICpratica5.htm>[http : //serialink.com.br](http://serialink.com.br)
- [4] <ftp://ftp.dca.fee.unicamp.br/pub/docs/ea079/complementos/Lcd.pdf>
- [5] <http://www.ebah.com.br/content/ABAAAj2YAD/tutorial-lcd?part=3>
- [6] <http://www.angelfire.com/ultra/lcd/gerador.htm>
- [7] [http://wiki.sj.ifsc.edu.br/wiki/index.php/MIC29004-2015-](http://wiki.sj.ifsc.edu.br/wiki/index.php/MIC29004-2015-1)  
*1Microprocessadores – Engenharia* *Emílio Nicolo*, 8051 *detalhado*, 8 *edição*
- [8]