# DISPLAY DE LCD 16x2

O chip que se encontra dentro do display é um HITACHI HD44780 ou algum outro que seja compatível com o mesmo. Possui 14 pinos:

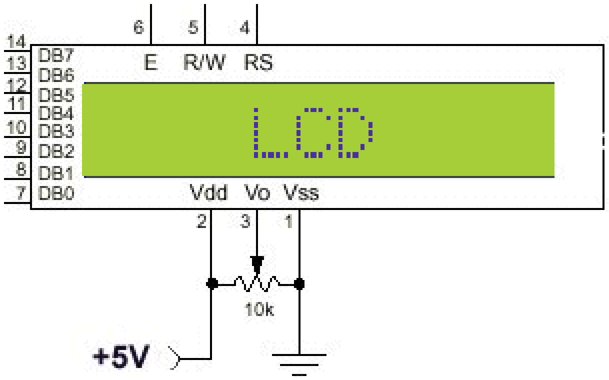
| Número | Símbolo | Função |
|---|---|---|
| 1 | Vss | 0v Power Supply (GND Level) |
| 2 | Vdd | Power Supply for Logic Circuit |
| 3 | Vo | Is for adjusting the contrast of the display. Usually, when this pin is grounded the pixels will be the darkest |
| 4 | RS | Data/Instruction select |
| 5 | R/W | Determines if we read from or write to the LCD |
| 6 | E | Enables or disables the LCD module |
| 7-14 | DB0-DB7 | Bi-directional data bus |

**OBS:** Quando o pino E está ‘baixo’, o display se encontra desabilitado e os valores que se encontram nos pinos RS, R/W e no barramento de dados serão ignorados.

## Register Select (RS):

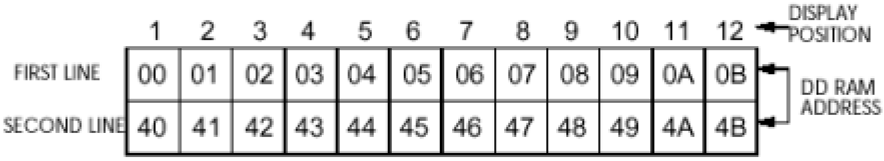
RS = ‘0’ quando se envia para o display uma instrução (instruction register)
RS = ‘1’ quando se envia para o display um caractere (data register)

Segue abaixo um esquemático do display:



## Display Data RAM (DDRAM):

Display data RAM é pra onde você manda os caracteres (ASCII code) que você quer ver na tela do LCD. Sua capacidade é de 80 caracteres (bytes). O endereço da DDRAM é a posição do cursor (ex: onde o dado que você enviou será mostrado). Abaixo são mostrados os endereços da DDRAM:



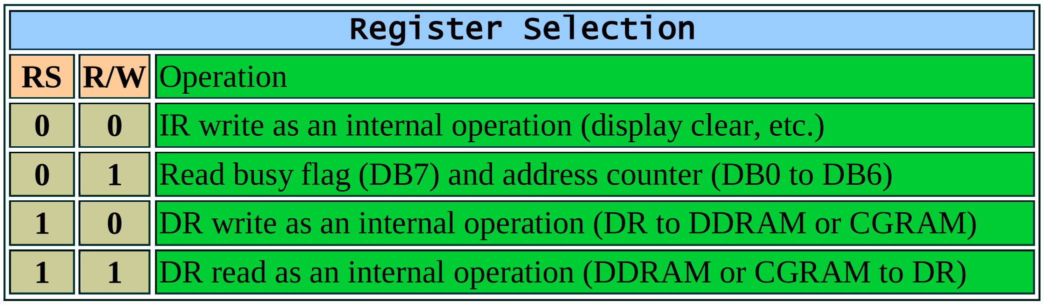
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FIRST LINE | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B |
| SECOND LINE | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B |

## Character Generator RAM (CGRAM):

No CGRAM o usuário pode definir seus próprios caracteres por programação. CG RAM é de 64 bytes.

## Registers:

O HDD44780 tem dois registradores de 8 bits, um é o instruction register (IR) e o outro é o data register (DR). O IR grava códigos de instruções, como limpar display e shift do cursor, e informação de endereço para DDRAM e CGRAM. O DR grava temporariamente dados para serem escritos na DDRAM ou CGRAM e dados para serem lidos da DDRAM ou CGRAM. O dado escrito no DR é automaticamente escrito na DDRAM ou CGRAM por uma operação interna. Esses dois registradores podem ser selecionados pelo register select (RS):



Register Selection

| RS | R/W | Operation |
|---|---|---|
| 0 | 0 | IR write as an internal operation (display clear, etc.) |
| 0 | 1 | Read busy flag (DB7) and address counter (DB0 to DB6) |
| 1 | 0 | DR write as an internal operation (DR to DDRAM or CGRAM) |
| 1 | 1 | DR read as an internal operation (DDRAM or CGRAM to DR) |

## Busy Flag (BF):

Quando o busy flag é ‘1’, o HD44780 está no modo de operação interno e a próxima instrução não será aceita. Quando RS = ‘0’ e R/W = ‘1’, o busy flag é colocado na saída do DB7. A instrução seguinte deve ser escrita após ter-se assegurado de que o busy flag é ‘0’.

## Address Counter (AC):

O AC atribui endereços para DDRAM e CGRAM. Quando o endereço de uma instrução setada no DDRAM/CGRAM é escrito dentro do IR, o endereço da informação é mandado do IR para o AC. A seleção tanto do DDRAM como do CGRAM é determinada também simultaneamente pela instrução. Depois de escrever na (ou ler da) DDRAM ou CGRAM, o AC é automaticamente incrementado ou decrementado por 1.

Para mostrar um caractere, a posição do dado é escrita no IR (endereço na DDRAM). O código do caractere é então escrito no DR e o LCD mostra o caractere correspondente na posição especificada. O LCD pode ainda incrementar ou decrementar a posição no display depois de cada caractere ter sido mostrado, só que para isso acontecer deve-se ter uma string de caracteres.
Para que um caractere seja escrito no display, o mesmo deve ser inicializado antes.

## LCD Commands:

```
+----+-----+-----------------------------------------+
| RS | P0  | Descrição                               |
+----+-----+-----------------------------------------+
| 0  | 38H | 8 bits, 2 linhas, 5 x 8 font            |
| 0  | 0FH | display on, cursor on, blink on         |
| 0  | C0H | posiciona o cursor na na linha 1, coluna 0|
| 0  | 80H | posiciona o cursor na na linha 0, coluna 0|
| 0  | 84H | posiciona o cursor na na linha 0, coluna 4|
| 0  | C8H | posiciona o cursor na na linha 1, coluna 8|
| 0  | 07H | inc. cursor position, scroll on         |
| 0  | 01H | clear display                           |
| 1  | xxH | escreve dado xx (ASCII) no display 8 bits |
+----+-----+-----------------------------------------+
```

A informação, tanto de configuração quanto de dado, só será escrita no display, no evento de uma transição de 1 para 0 no bit EN (P3.6). A transição deverá levar, no mínimo, 450 ns para ocorrer (ex. colocar o sinal EN em '1', executar uma rotina de delay para esperar por 450 ns, colocar o sinal EN de volta em '0' para realizar a escrita no display). Um delay em torno de 5 us é suficiente.

The commands for HD44780 chip are shown in the table below.

| INSTRUCTION | Decimal | Hexadecimal |
|---|---|---|
| **Function set (8-bit interface, 2 lines, 5*7 Pixels)** | **56** | **38** |
| **Function set (8-bit interface, 1 line, 5*7 Pixels)** | **48** | **30** |
| **Function set (4-bit interface, 2 lines, 5*7 Pixels)** | **40** | **28** |
| **Function set (4-bit interface, 1 line, 5*7 Pixels)** | **32** | **20** |
| **Entry mode set** | **See Below** | **See Below** |
| **Scroll display one character right (all lines)** | **28** | **1E** |
| **Scroll display one character left (all lines)** | **24** | **18** |
| **Home (move cursor to top/left character position)** | **2** | **2** |
| **Move cursor one character left** | **16** | **10** |
| **Move cursor one character right** | **20** | **14** |
| **Turn on visible underline cursor** | **14** | **0E** |

| | | |
|---|---|---|
| **Turn on visible blinking-block cursor** | **15** | **0F** |
| **Make cursor invisible** | **12** | **0C** |
| **Blank the display (without clearing)** | **8** | **08** |
| **Restore the display (with cursor hidden)** | **12** | **0C** |
| **Clear Screen** | **1** | **01** |
| **Set cursor position (DDRAM address)** | **128 + addr** | **80+ addr** |
| **Set pointer in character-generator RAM (CG RAM address)** | **64 + addr** | **40+ addr** |
| **Read DDRAM/CGRAM & Check Busy Flag** | **See Below** | **See Below** |

### Entry mode set command

This command sets cursor move direction and display shift ON/OFF. There are 4 possible function set commands;04, 05, 06, and 07. This command changes the direction the cursor moves by setting the address counter to increment or decrement.

When the address counter is set to DECREMENT, strings sent to the LCD will be printed in reverse order. The same thing applies to the CG RAM, as well.

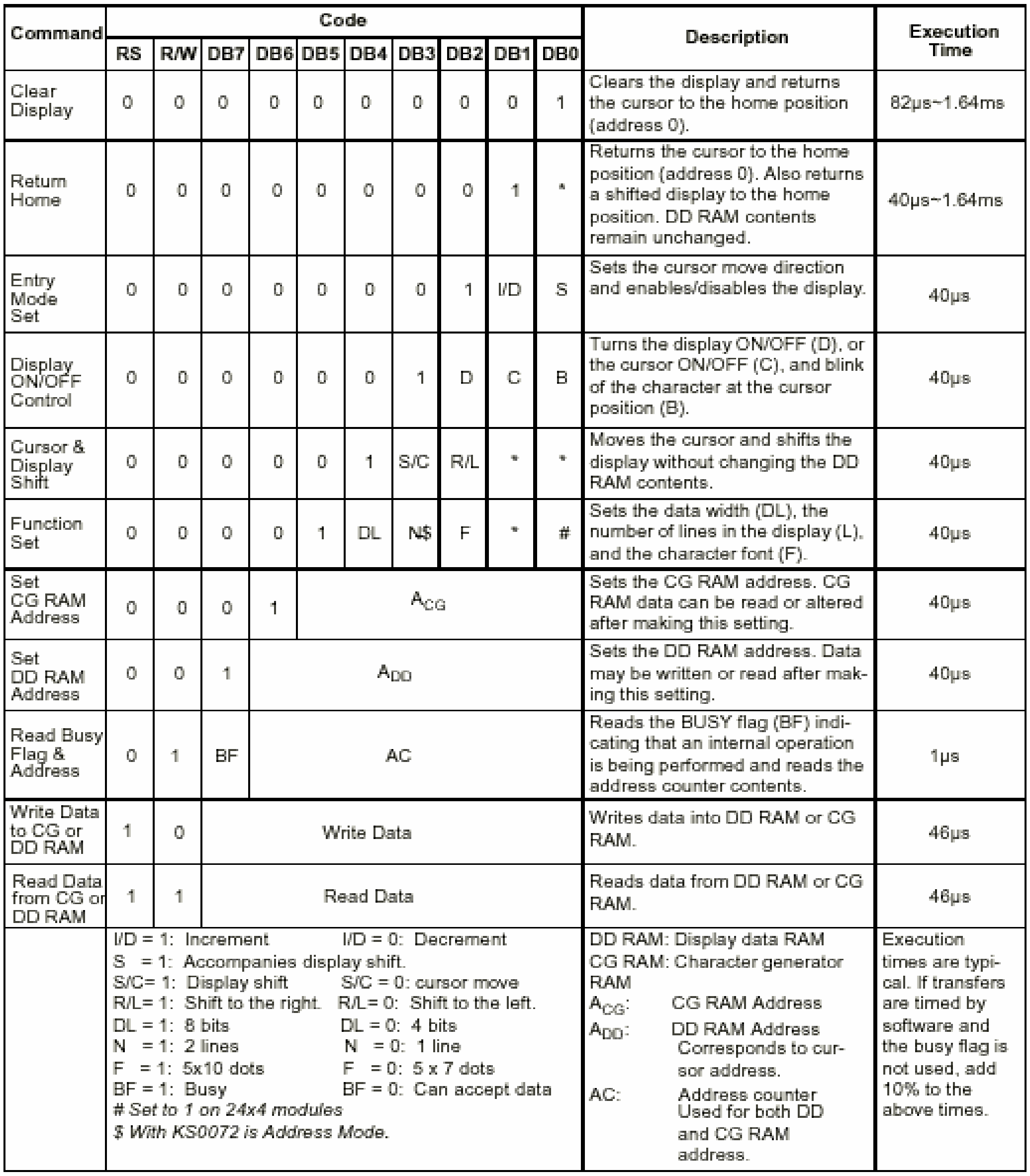
http://www.geocities.com/dinceraydin/lcd/commands.htm

### Checking the Busy Flag

To check the state of the busy flag and read the address counter:

1. Set R/W Pin of the LCD HIGH(read from the LCD)
2. Select the instruction register by setting RS pin LOW
3. Enable the LCD by Setting the enable pin HIGH
4. The most significant bit of the LCD data bus is the state of the busy flag(1=Busy,0=ready to accept instructions/data).The other bits hold the current value of the address counter.

### Complete command reference:



| Command | Code | | | | | | | | | | Description | Execution Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RS | R/W | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 | | |
| Clear Display | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Clears the display and returns the cursor to the home position (address 0). | 82μs~1.64ms |
| Return Home | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | * | Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged. | 40μs~1.64ms |
| Entry Mode Set | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | S | Sets the cursor move direction and enables/disables the display. | 40μs |
| Display ON/OFF Control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B). | 40μs |
| Cursor & Display Shift | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | * | * | Moves the cursor and shifts the display without changing the DD RAM contents. | 40μs |
| Function Set | 0 | 0 | 0 | 0 | 1 | DL | N$ | F | * | # | Sets the data width (DL), the number of lines in the display (L), and the character font (F). | 40μs |
| Set CG RAM Address | 0 | 0 | 0 | 1 | $A_{CG}$ | | | | | | Sets the CG RAM address. CG RAM data can be read or altered after making this setting. | 40μs |
| Set DD RAM Address | 0 | 0 | 1 | $A_{DD}$ | | | | | | | Sets the DD RAM address. Data may be written or read after making this setting. | 40μs |
| Read Busy Flag & Address | 0 | 1 | BF | AC | | | | | | | Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents. | 1μs |
| Write Data to CG or DD RAM | 1 | 0 | Write Data | | | | | | | | Writes data into DD RAM or CG RAM. | 46μs |
| Read Data from CG or DD RAM | 1 | 1 | Read Data | | | | | | | | Reads data from DD RAM or CG RAM. | 46μs |
| | I/D = 1: Increment I/D = 0: Decrement<br>S = 1: Accompanies display shift.<br>S/C= 1: Display shift S/C = 0: cursor move<br>R/L= 1: Shift to the right. R/L= 0: Shift to the left.<br>DL = 1: 8 bits DL = 0: 4 bits<br>N = 1: 2 lines N = 0: 1 line<br>F = 1: 5x10 dots F = 0: 5 x 7 dots<br>BF = 1: Busy BF = 0: Can accept data<br>*# Set to 1 on 24x4 modules*<br>*$ With KS0072 is Address Mode.* | | | | | | | | | | DD RAM: Display data RAM<br>CG RAM: Character generator RAM<br>$A_{CG}$: CG RAM Address<br>$A_{DD}$: DD RAM Address Corresponds to cursor address.<br>AC: Address counter Used for both DD and CG RAM address. | Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times. |

## Defining Custom Characters:

### Character-Generator ROM and RAM:

When you send the ASCII code for a character like “A” to an LCD module, the module’s controller looks up the appropriate 5x8-pixel pattern in ROM (read-only memory) and displays that pattern on the LCD. That character-generator ROM contains 192 bit maps corresponding to the alphabet, numbers, punctuation, Japanese Kanji characters, and Greek symbols. The ROM is part of the main LCD controller (e.g., HD44780, KS0066, etc.), is mask-programmed, and cannot be changed by the user. The manufacturers do offer alternative symbols sets in ROM for European and Asian languages, but most U.S. distributors stock only the standard character set shown in the LCD Serial Backpack manual. Alphanumeric LCD controllers do not allow you to turn individual pixels on or off—they just let you pick a particular pattern (corresponding to an ASCII code) and display it on the screen. If you can't change the ROM and you can't control pixels, how do you create graphics on these LCDs? Easy. There's a 64-byte hunk of RAM (random-access memory) that the LCD controller uses in the same way as character-generator (CG) ROM. When the controller receives an ASCII code in the range that's mapped to the CG RAM, it uses the bit patterns stored there to display a pattern on the LCD. The main difference is that you can write to CG RAM, thereby defining your own graphic symbols.

Each byte of CG RAM is mapped to a five-bit horizontal row of pixels, and LCD characters are typically eight rows high, so 64 bytes of CG RAM is enough to define eight custom characters. These characters correspond to ASCII codes 0 through 7, which normally serve as control codes for marking the beginning of a serial transmission or ringing the bell on a terminal. Since these have no meaning to an LCD module, the designers appropriated them for CG RAM.

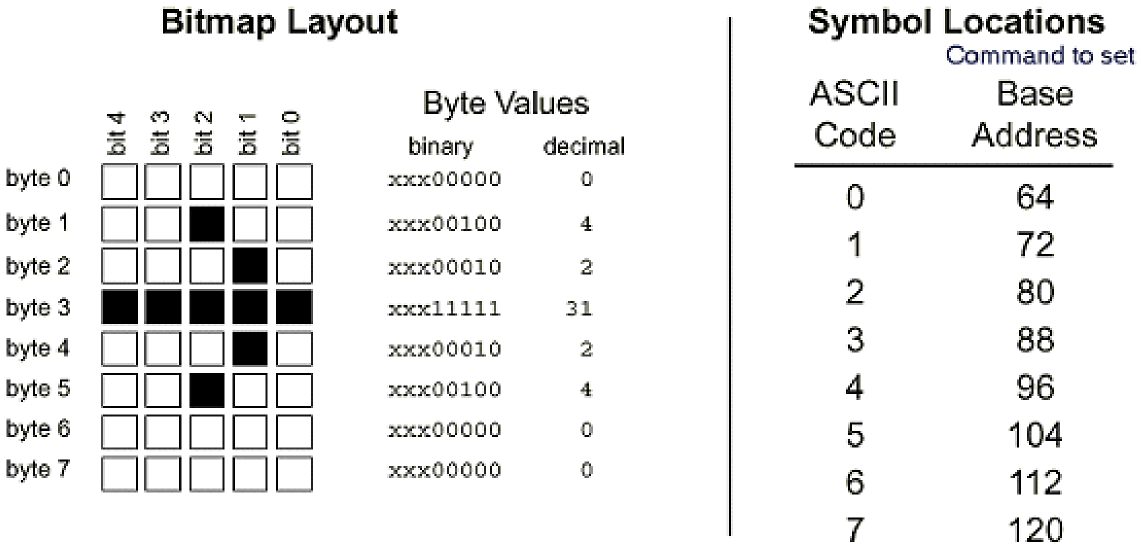
When an LCD is first powered up, CG RAM contains random bits garbage. If necessary,you may clear CG RAM by writing 00 into each CG RAM cell.

**Writing to CG RAM:**

Writing to CG RAM is a lot like moving the cursor to a particular position on the display and displaying characters at that new location. The steps are:

- Reset RS and R/W pins of the LCD to prepare the LCD to accept instructions
- Set the CG RAM address by sending an instruction byte from 64 to 127 (locations 0–63 in CG RAM).
- Switch to DATA MODE by changing setting the RS pin
- Send bytes with the bit patterns for your symbol(s). The LCD controller automatically increments CG RAM addresses, just as it does cursor positions on the display.
- To leave CG RAM,switch to COMMAND MODE to set address counter to a valid display address (e.g. 128, 1st character of 1st line); the clear-screen instruction (byte 1); or the home instruction (byte 2). Now bytes are once again being written to the visible portion of the display.
- To see the custom character(s) you have defined, print ASCII codes 0 through 7.

**Bitmap layout example:**



**Bitmap Layout**

| | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Byte Values binary | Byte Values decimal |
|---|---|---|---|---|---|---|---|
| byte 0 | □ | □ | □ | □ | □ | xxx00000 | 0 |
| byte 1 | □ | □ | ■ | □ | □ | xxx00100 | 4 |
| byte 2 | □ | □ | □ | ■ | □ | xxx00010 | 2 |
| byte 3 | ■ | ■ | ■ | ■ | ■ | xxx11111 | 31 |
| byte 4 | □ | □ | □ | ■ | □ | xxx00010 | 2 |
| byte 5 | □ | □ | ■ | □ | □ | xxx00100 | 4 |
| byte 6 | □ | □ | □ | □ | □ | xxx00000 | 0 |
| byte 7 | □ | □ | □ | □ | □ | xxx00000 | 0 |

**Symbol Locations**

| ASCII Code | Command to set Base Address |
|---|---|
| 0 | 64 |
| 1 | 72 |
| 2 | 80 |
| 3 | 88 |
| 4 | 96 |
| 5 | 104 |
| 6 | 112 |
| 7 | 120 |

**Simple Animation:**

You can create a sort of animation by rapidly printing a series of custom characters at the same screen position.

## Links:

1) http://www.geocities.com/dinceraydin/lcd/index.html
2) http://www.inf.pucrs.br/~eduardob/disciplinas/topicos_sdacii_pos/8051/Exemplos_C/display.c