

How to generate and display self made Custom characters on 16x2 lcd

EG Projects

This tutorial is about making/building your own characters/special images and then displaying them on character 16×2 lcd. Generating custom characters or special character images and displaying on lcds of any sizes (16×1,16×2,8×1,8×2,20×1,20×2,40×1,40×2 etc) is not a very hard task. One must go through the internal structure of lcd control set in order to know how the character lcd works? Once you know how the lcd works and what features it offers. Then you can easily build and display characters of your desire on lcd screen in a defined matrix.

It requires only knowledge of CG-RAM(character generated ram) of character lcd and the lcd chip set controller to build and display self made characters on 16×2 lcd display. Most of the lcds contains HD4478 controller in them. HD4478 controller is build by Hitachi and its the most popular controller used in the character lcd's. Almost all of the character lcd's resides one, but new competitors are also present in market.

By custom characters i mean characters that are not present in the ASCII character set of lcd controller. Like heart symbol, smiley 😊 😊 ;D :-d 😊 etc. We have to declare these custom characters in CG-RAM of lcd by our own. This tutorial will teach you how to declare custom characters in CG-RAM and then call display them one by one to display on 16×2 lcd screen.

CG-RAM is the main component in making custom characters. **CG** stands for **custom generated** and RAM you all know random access memory. This CG-RAM stores our custom characters once we declare them in our code. I will come on it later. As you know once we write any type of code we need a memory to store it and a controller to run it. In the 16×2 lcd custom character case its same. We write code(arrays) of character's which we want to display on lcd. Then we store them in a memory on lcd. In our case this memory is named as CG-RAM(Character generated RAM).

CG-RAM size is 64 Bytes. You can create 8 characters at a time and load them in cg-ram. Each character occupiees 8-bytes. Eight characters each of eight byte (8-characters * 8-Bytes) is equal to 8×8=64 Bytes. CG-RAM address in lcd memory starts from 0x40(Hexadecimal) or 64 in decimal.

You can place your first character in address ranging from 0x40 to 0x47. The address moves on by 8 bytes for each character for 2nd character address starts from 0x48 and goes to 0x4F. Last eight't character address starts from 0x78 and goes to 0x7F.

We generate and put our self made custom characters at these addresses. Once we generate our characters at these address now we can print them on lcd at any time by just sending simple commands to lcd controller. Below is the table in which characters addresses and their printing commands are given.

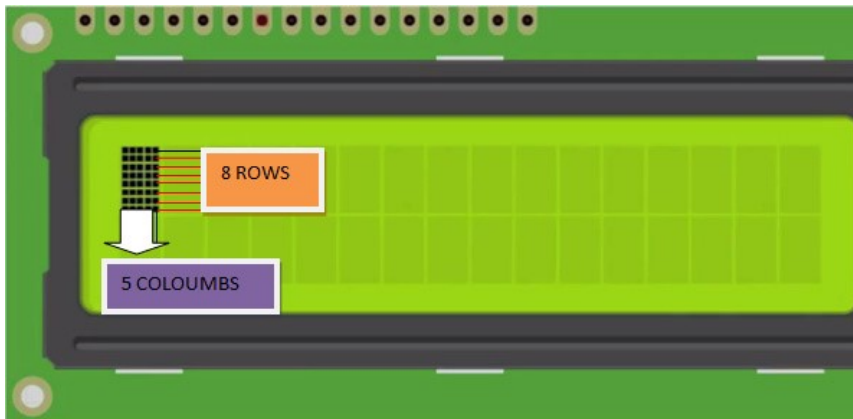
CG-RAM Characters www.microcontroller-project.com	CG-RAM Address (Hexadecimal)	Commands to display Generated Characters (Decimal)
1 st Character	0x40	0
2 nd Character	0x48	1
3 rd Character	0x50	2
4 th Character	0x58	3
5 th Character	0x60	4
6 th Character	0x68	5
7 th Character	0x70	6
8 th Character www.microcontroller-project.com	0x78	7

CG-RAM character addresses and commands

In the diagram above you can see starting addresses for each characters with their displaying commands. First character is generated at address 0x40 to 0x47 and is printed on lcd by just sending simple command 0 to lcd. Second character is generated at address 0x48 to 0x55 and is printed by sending 1 to lcd. You can see all character addresses and commands in the above diagram.

How to Generate and place Custom Characters in CG-RAM?

Character lcd's displays each character in 5×7 or 5×8 matrix. Where 5 represents number of coulombs and 8 are number of rows. We create our custom character in 5×7 matrix format. The eighth row is not used because it is dedicated for cursor. In the below picture you can see each individual and dedicated slot of character is a square matrix of pixels. Pixels are arranged in 5×8 format.



Lcd 5×8 Matrix Display

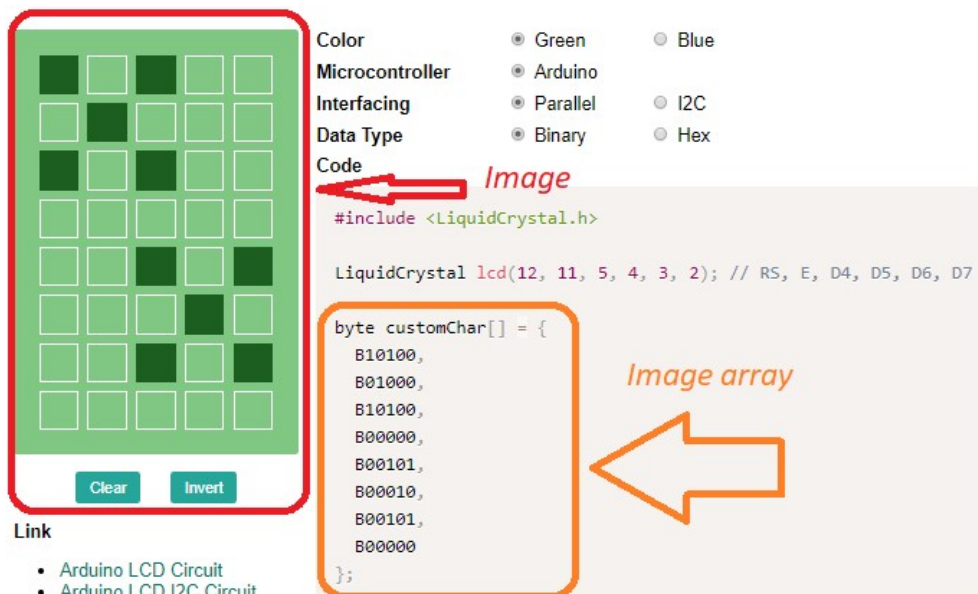
Declaring character bits against CG-RAM addresses

Here is a simple example on how to generate pixels array for letter 'b' and then place it in CG-RAM of character 16×2 lcd display.

The Array for generating letter 'b' is

```
char b[7]={0x10,0x10,0x16,0x19,0x11,0x1E};
```

We now have some online websites through which you can get array of your desired image in 5×7 or 5×8 format. One of the most popular is [maxpomer](https://maxpomer.com/). You can graphically input your desired image on a matrix in maxpomer and then with a single button press you get the array of the image. The array can further be used in your code.



maxpomer custom character generator

In the above picture you can see the interface of maxpomer. Pretty simple and easy to use. In just seconds you can generate the matrix array for your input character. Array can be generated in binary as well as hexadecimal format.

What individual bits represent in array really needs to be understood. '0' represents pixel is off and '1' represent pixel is on. Like for the letter 'b' whose array was highlighted above. Its representation is given below. For each row we have a binary or hexadecimal representation. In binary representation we ignore the 3 MSB or most significant bits, because the coulombs of the matrix are 5. So 5 LSB least significant bits are considered. Also in hexadecimal representation the 3 MSB don't matter. In the below picture the array of letter 'b' is placed against address ranging from **0x40 to 0x47**. Row-1 is placed at address 0x40, row-2 is placed at 0x41, then row-3 is placed at 0x42 and so on until row-7 against address 0x47.

CG-RAM ADDRESS	Byte Address	C1	C2	C3	C4	C5	Pattern for b is
Starting Address of CG-RAM.	0 0 0 0	1	0	0	0	0	Row1 0x10= 000 10000
0 1 0 0 0 0 0	0 0 0 1	1	0	0	0	0	Row2 0x10= 000 10000
4 0	0 0 1 0	1	0	1	1	0	Row3 0x16= 000 10110
Hexadecimal=0x40	0 0 1 1	1	1	0	0	1	Row4 0x19= 000 11001
Decimal=64	0 1 0 0	1	0	0	0	1	Row5 0x11= 000 10001
	0 1 0 1	1	0	0	0	1	Row6 0x11= 000 10001
	0 1 1 0	1	1	1	1	0	Row7 0x1E= 000 11110
	0 1 1 1	0	0	0	0	0	Cursor position

First three bits are treated as Dont cares.
* Rest five bits are loaded at the address

CG-RAM address with Character commands

In the above figure byte address is the address of particular byte of character. Since each custom character is made up of 8 bytes. So each byte address with the value loaded in it is given. The above method is universal. You can generate self made characters in character lcd of any size using the same above method.

Sample c++ code to generate custom character in cg-ram

```
char b[7]={0x10,0x10,0x16,0x19,0x11,0x11,0x1E}; //'b' character pattern placed in code
lcdcommand(0x40); //Notifying CG-RAM for incoming custom character and its address
while(i!=7) //Placing 'b' By sending 'b' pattern to cg-ram
{
  lcddata(b[i]);
  i++;
}
lcdcommand(0); //We placed 'b' at first address of CG-RAM so in order to print it send 0 to lcd
```

The above task is just a simple demo. We properly have to select the data and command registers of lcd in order to place the custom image in CG-RAM.

Important: Just to let you know that commands(like 0x40 address of cg-ram) are send to command register of lcd and data like 'b' pattern is send to data register of lcd.

The above code can be divide in to steps

- Send cg-ram character starting address where you want to create character. I send 0x40 in the above code which means i am placing character in first location.
- Now put your character at this address. Send the 'b' character array string defined above one by one to data register of lcd.
- To print the generated character at 0x40. Send command 0 to command register of lcd.

Visit some projects which i created using different microcontrollers to display custom self made characters on lcd screen. These projects will help you to understand what command and data registers means and how to select them? Projects contains free source code and circuit diagrams.