

Displaying an Image on Graphical LCD using 8051 Microcontroller

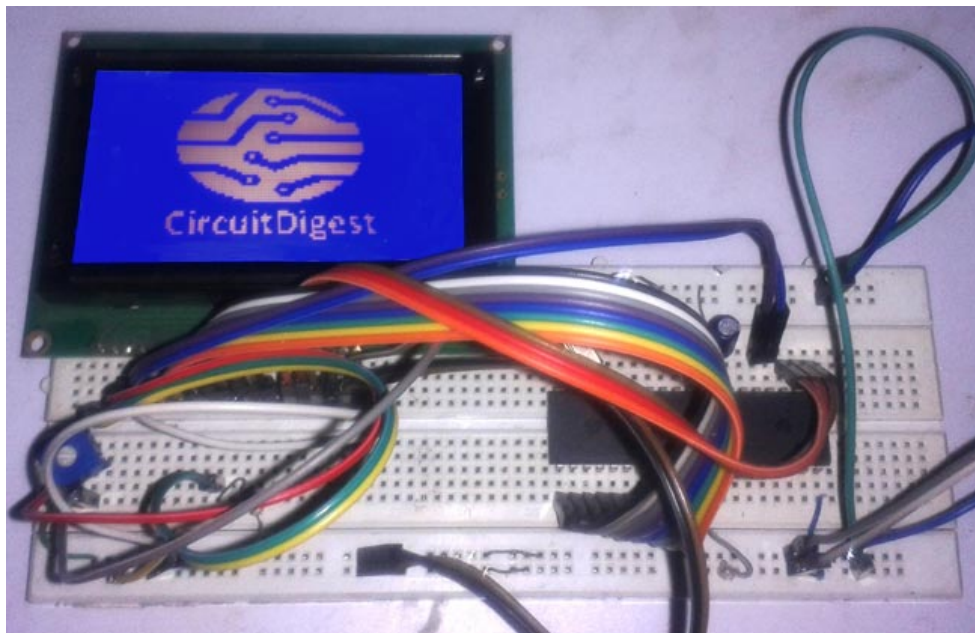
Published June 7, 2016

4

S

Saddam (/users/saddam)

Author



Displaying an Image on Graphical LCD using 8051 Microcontroller

In our daily life, we see many types of devices for displaying Text, Images and Graphics. LCDs are one of the most popular Display Device in Electronics and used in mostly all the projects which display some kind of information. There are many types of LCDs used in Electronic Projects. We have already used 16X2 LCD in many of our projects and also used [TFT LCD with Arduino](http://circuitdigest.com/microcontroller-projects/arduino-tft-lcd-interfacing-tutorial) (<http://circuitdigest.com/microcontroller-projects/arduino-tft-lcd-interfacing-tutorial>). You can find all our 16X2 LCD related project by following [this link](http://circuitdigest.com/search/node/16x2%20lcd) (<http://circuitdigest.com/search/node/16x2%20lcd>), including **interfacing with 8051, AVR, Arduino** and many more.

In this tutorial we are going to do **Graphic LCD interfacing with 8051 microcontroller**. In this project, we will show how to **Display an Image on Graphical LCD (GLCD)**.

Components Required:

- Graphical LCD

- AT89c52 8051 Microcontroller
- 7805 voltage regulator
- 1000uf capacitor
- 10 uF capacitor
- 10K resistor
- 10K POT
- Crystal Oscillator 12 MH
- Connecting wire
- Bread Board
- Burg strips male
- Power supply
- LED
- 220 Ohm resistor
- 1K resistor

Graphical LCD:



A simple 16x2 LCD has 16 pins but **Graphical LCD** has 20 pin. **Pin description** is given below as per its datasheet:

Pin No.	Pin Name	Description	Function
1	VSS	Ground	0 Volt
2	VDD	Power Supply	5 Volt

3	V0	LCD Contrast adjustment	
4	RS	Command/data Register selection	RS=0: Command Selection and RS=1: Data Selection
5	R/W	Read/Write Register	R/W=0: Write Selection and R/W=1: Read Selection
6	E	Enable Signal	
7	DB0	Data input/output pin (DB0-DB7)	8 Bit (DB0-DB7)
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	CS1	Chip Select	CS1 = 1, Chip Select Signal for IC1
16	CS2	Chip Select	CS2 = 1, Chip Select Signal for IC2
17	RST	Reset	Reset GLCD
18	VEE	Negative Voltage for LCD Driver	
19	A	Back light LED	5 Volt
20	K	Back light LED	Ground

Showing an Image on Graphical LCD using 8051:

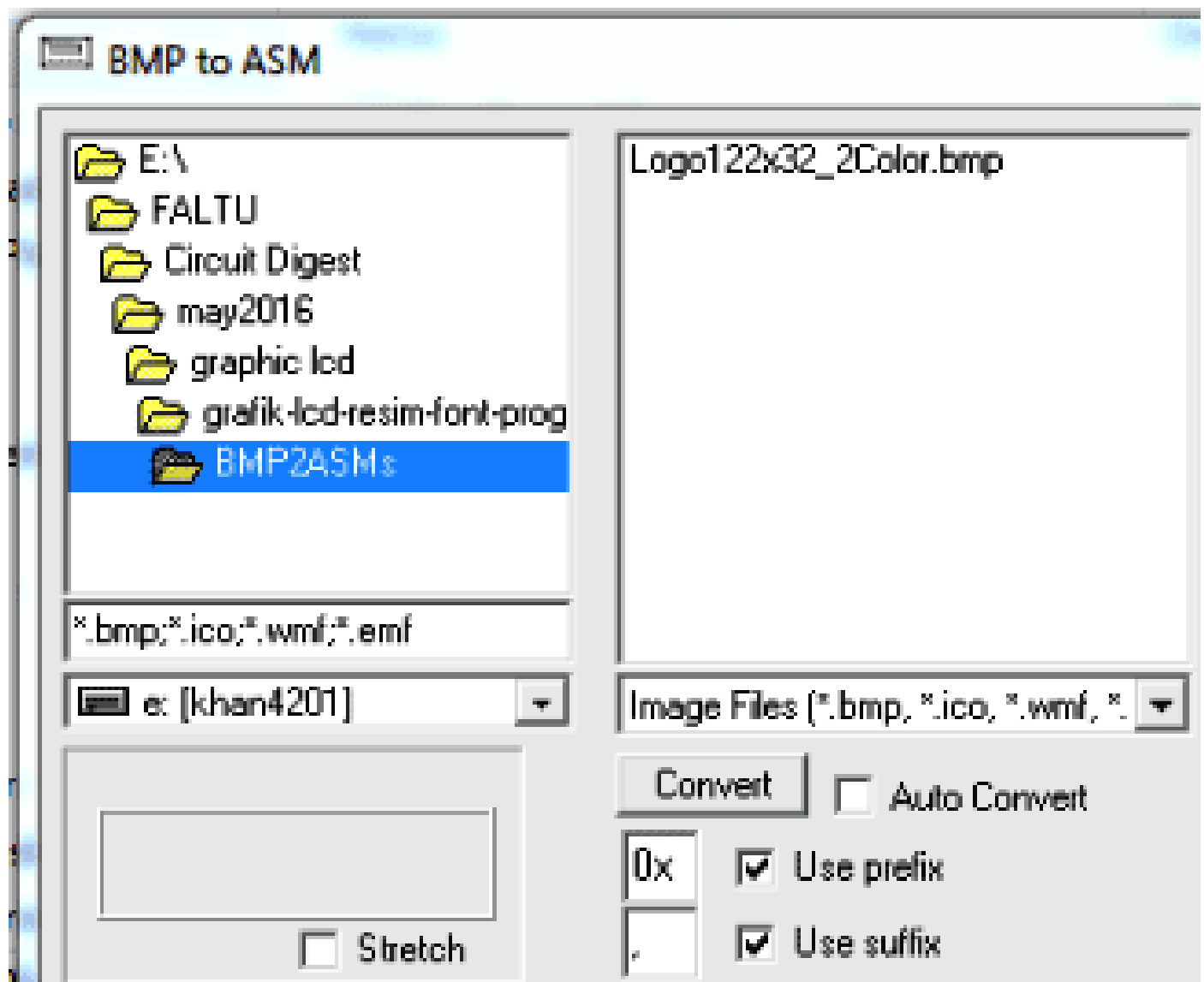
To show an Image on Graphical LCD, first we need to convert that image into Assembly Code, so that 8051 Microcontroller can understand and read it. So we need to follow the below **Steps for Converting Image into HEX code**:

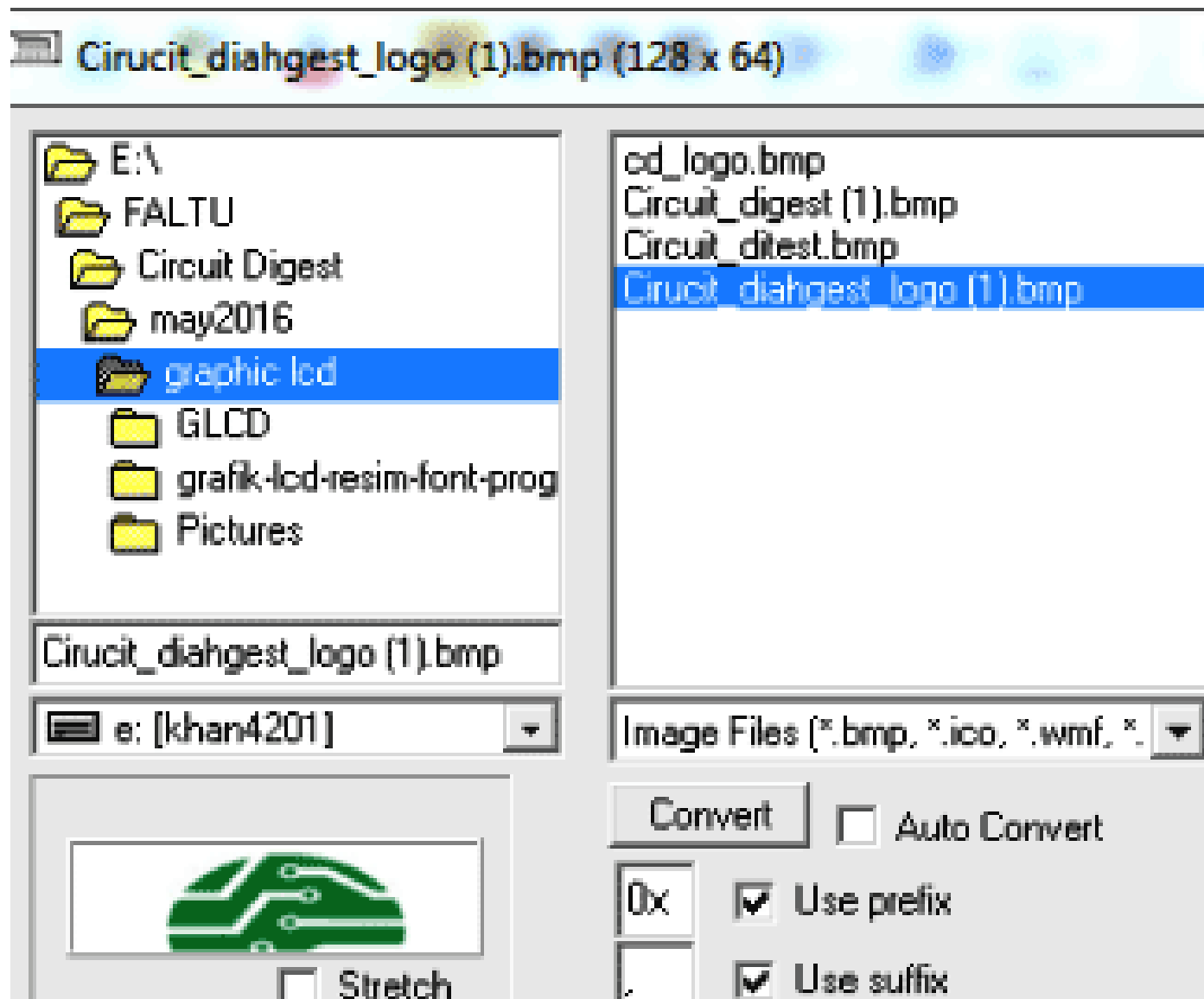
Step 1: First we need to download a application which convert Image (BMP format) into Assembly Code. So download the **BMP2ASM Image Conversion Application** from [this link \(http://circuitdigest.com/sites/default/files/BMP2ASM.zip\)](http://circuitdigest.com/sites/default/files/BMP2ASM.zip), just Right Click on the link and then click Save link as...

Step 2: Then, select the image which you want to display on the **Graphical LCD** and **convert it into BMP**, (if it is not in BMP format already) using any application like MS Paint, Photoshop etc. Or you can find many online website for Image format conversion. We have below BMP image, with size 128x64:

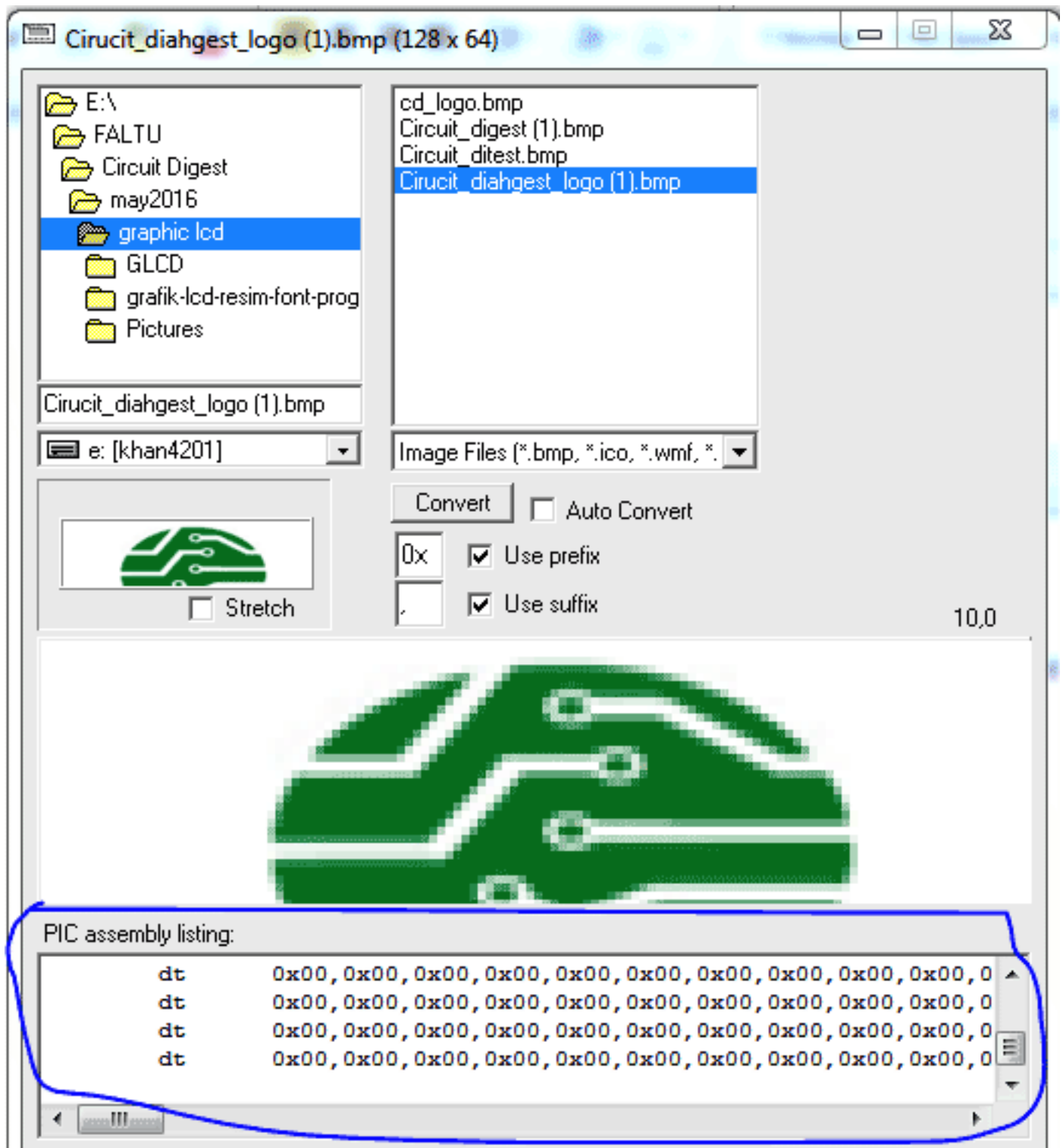


Step 3: Now extract the BMP2ASM.zip file, which we have downloaded in Step 1 and open Bmp2asm.exe within it, and **select** the BMP image.





Step 4: Now press "Convert" in BMP2ASM application window.



Step 5: Then Copy the generated code and paste in the program of 8051 in Keil uVision. Do some changes and compile the code.

Now your code is ready to upload in 8051 Microcontroller.


```
#include<reg51.h>
#include<intrins.h>
#define lcdport P2

sbit rs=P1^0;
sbit rw=P1^1;
sbit en=P1^2;
sbit cs1=P1^3;
sbit cs2=P1^4;
```

After it, we have created a delay function.

```
void delay(int itime)
{
    int i,j;
    for(i=0;i<itime;i++)
        for(j=0;j<125;j++);
}
```

Function *void lcd_disp_on()* is used for Turning On the display.

Function *void setCursorY(int y)* is created for setting the column in GLCD and Function *void setCursorX(int x)* is created for setting the page on the GLCD. Complete Code file is given in below Code Section.

After setting Column and Page, we have written a function for sending command and data to GLCD.

```
void lcdprint(char dat,int size)
{
    unsigned int i;
    for(i=0;i<size;i++)
    {
        if(c<64)
        {
            lcdport=dat;
            cs1=1;
            ....
            ....
        }
    }
}
```

In *void main()* function, we have cleared GLCD and then set Column and Page. After it, send data to LCD by using *void lcdprint(char dat,int size)* function.

```
void main()
{
    int x,y;
    P3=0xff;
    while(1)
    {
        lcdclear();
        for(y=0;y<8;y++)
        {
            for(x=0;x<128;x++)
            {
                lcd_disp_on();
                setCursorX(y);
                setCursorY(x);
                lcdprint(image[y][x],x);
            }
        }
        ....
        ....
    }
}
```

Flow of Code:

1. First when we Power up the system, program clears the GLCD and Turns On the display.
2. Then set cursor to Column, from where we want to write the Data.
3. Then set cursor to Page, from where we want to write the Data.
4. Now program sends Data to the selected location one by one till 128X8 Times. Because a GLCD has 8 pages and 128 Columns.

Code

Download the Code file for "Displaying an Image on Graphical LCD using 8051 Microcontroller" from below link, Right Click on Showing_image_on_GLCD_8051.c (http://circuitdigest.com/sites/default/files/Showing_image_on_GLCD_8051.c)

Video

Graphic LCD Interfacing with 8051 Microcontroller

**Tags**

[8051 \(/Tags/8051\)](#) [Interfacing \(/Tags/Interfacing\)](#) [Graphical LCD \(/Tags/Graphical-Lcd\)](#)

[Embedded \(/Tags/Embedded\)](#)

Comments

Submitted by [abdullah](#) on Thu, 11/03/2016 - 20:54

[Permalink \(/comment/7821#comment-7821\)](#)

model of graphical lcd (/comment/7821#comment-7821)

sir can you tell me the model of glcd and also about the circuit simulator which you had used. i'm using proteus and i am facing trouble in choosing graphical lcd

Submitted by [Abhishek \(/users/abhishek\)](#) on Fri, 12/02/2016 - 19:37

[Permalink \(/comment/8799#comment-8799\)](#)

You can use any 128x64 (/comment/8799#comment-8799)

You can use any 128x64 Graphical LCD, its JHD12864E Graphical LCD. You can simulate it in Proteus

Submitted by [s.m. shamim](#) on Thu, 01/26/2017 - 23:35

[Permalink \(/comment/10370#comment-10370\)](#)

JHD12864E Graphical LCD (/comment/10370#comment-10370)

JHD12864E Graphical LCD Display not found in proteus to simulate. So what can I do now?????

Submitted by [Amir Abbass Za... \(/users/amir-abbass-zahir-mobarhan\)](#) on Wed, 09/20/2017 - 12:52

[Permalink \(/comment/18210#comment-18210\)](#)

Thanks for such projects. (/comment/18210#comment-18210)

Thanks for such projects.

Log in ([/user/login?destination=/microcontroller-projects/graphic-lcd-interfacing-with-8051-microcontroller%23comment-form](#)) or register ([/user/register?destination=/microcontroller-projects/graphic-lcd-interfacing-with-8051-microcontroller%23comment-form](#)) to post comments





3M™ TwinAx High Speed Cable Solutions (<https://bit.ly/3HH4BTG>)

3M™ TwinAx High Speed Cable Solutions: Thin, low profile cable with extremely tight bend radii

(<https://bit.ly/3HH4BTG>)



Waterproof USB Type-C® Connectors (<https://bit.ly/40fSBjE>)

Amphenol Communications Solutions connectors are tailored for emerging product designs

(<https://bit.ly/40fSBjE>)



AUTOMATE® Type A Mini-FAKRA Cable Assemblies (<https://bit.ly/3HEMQV6>)

Amphenol RF's AUTOMATE Type A Mini-FAKRA quad-port jack is designed with a compact, modular housing

(<https://bit.ly/3HEMQV6>)



MKP Series Class X2 EMI/RFI Suppression Capacitors (<https://bit.ly/40b3xil>)

This makes it easier to implement noise suppression with an optimal footprint and low BOM cost.

(<https://bit.ly/40b3xil>)



Gar/Barracuda Series Vehicular Antennas (<https://bit.ly/3Y8VjVV>)

Laird Connectivity's Gar and Barracuda products offer reliability, durability, and performance

(<https://bit.ly/3Y8VjVV>)



Front-End Modules for IoT and Smart Energy (<https://bit.ly/3Y4awrx>)

Skyworks' Front-End Modules for IoT and Smart Energy featuring SKY85716-11

(<https://bit.ly/3Y4awrx>)



SSM10N954L 12 V Common Drain N-channel MOSFET for Battery Protection (<https://bit.ly/3DpRK65>)

Toshiba's 12 V low on-resistance common-drain MOSFETs help battery-driven devices

(<https://bit.ly/3DpRK65>)



AVR® DB Family of Microcontrollers (<https://bit.ly/3rZV7uA>)

Microchip's AVR DB family of microcontrollers is designed for demanding analog applications

(<https://bit.ly/3rZV7uA>)



Join 100K+ Subscribers

Your email is safe with us, we don't spam.





Type your email address

Subscribe

Be a part of our ever growing community.



Semicon Media is a unique collection of online media, focused purely on the Electronics Community across the globe. With a perfectly blended team of Engineers and Journalists, we demystify electronics and its related technologies by providing high value content to our readers.

 (<https://www.facebook.com/circuitdigest/>)  (<https://twitter.com/CircuitDigest>) 
(<https://www.youtube.com/channel/UCy3CUAIYgZdAOG9k3IPdLmw>)  (<https://www.linkedin.com/company/circuit-digest/>)

COMPANY

[Privacy Policy \(/privacy-policy\)](/privacy-policy)[Cookie Policy \(/cookie-policy\)](/cookie-policy)[Terms of Use \(/terms-of-use\)](/terms-of-use)[Contact](#)[Us \(/contact\)](#)[Advertise \(/advertise\)](#)

PROJECT

[555 Timer Circuits \(/555-timer-circuits\)](/555-timer-circuits)[Op-amp Circuits \(/op-amp-circuits\)](/op-amp-circuits)[Audio Circuits \(/audio-](/audio-circuits)[circuits\)](#)[Power Supply Circuits \(/smps-power-supply-circuits\)](/smps-power-supply-circuits)[Arduino Projects \(/arduino-projects\)](/arduino-projects)[Raspberry Pi Projects \(/simple-raspberry-pi-projects-for-beginners\)](/simple-raspberry-pi-projects-for-beginners)[MSP430 Projects \(/msp430-](/msp430-projects)[projects\)](#)[STM32 Projects \(/stm32-projects-and-tutorials\)](/stm32-projects-and-tutorials)[ESP8266 Projects \(/esp8266-projects\)](/esp8266-projects)[PIC Projects \(/pic-microcontroller-projects\)](/pic-microcontroller-projects)[AVR Projects \(/avr-microcontroller-projects\)](/avr-microcontroller-projects)[8051 Projects](#)[\(/8051-microcontroller-projects\)](/8051-microcontroller-projects)[ESP32 Projects \(/esp32-projects\)](/esp32-projects)[\[iot-projects\\)\]\(#\)](/internet-of-things-</div><div data-bbox=)[PCB Projects \(/diy-pcb-projects\)](/diy-pcb-projects)[\[projects\\)\]\(#\)](/arduino-esp8266-</div><div data-bbox=)[All Microcontroller Projects \(/microcontroller-projects\)](/microcontroller-projects)

OUR NETWORK

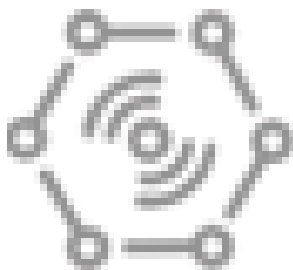


CIRCUIT DIGEST

(<https://circuitdigest.com>)

COMPONENTS¹⁰¹

(<https://components101.com>)



IOT DESIGN PRO

(<https://iotdesignpro.com>)

Copyright © 2022 Circuit Digest (/). All rights reserved.