

0.96 inch IIC Serial OLED Module User Manual



General Description:

The OLED is an advanced display which is self light emissive, high contrast, high response, low power consumptive, and large view angle, no backlight is required. Each OLED display is made of 128x64 individual OLEDs, each one is turned on or off by the controller chip.

This miniature display can be connect directly to any 3V or 5V microcontroller which has IIC interface, without needing any kind of level shifter. so it's very easy to get up and running in your project.

Features:

1. Compatible with Raspberry Pi Zero, Zero W, 3B, 3B+, STM32, Arduino, beaglebone black, and any microcontroller which has IIC interface.
2. Driver IC: SSD1306, Display is made of 128x64 individual OLED pixels.
3. OLED makes its own light, no backlight is required.
4. Compact size and lightweight, easy to assembly anywhere.
5. Comes with detailed documents, three kinds of ASCII character library, annotated sources and friendly technology support.

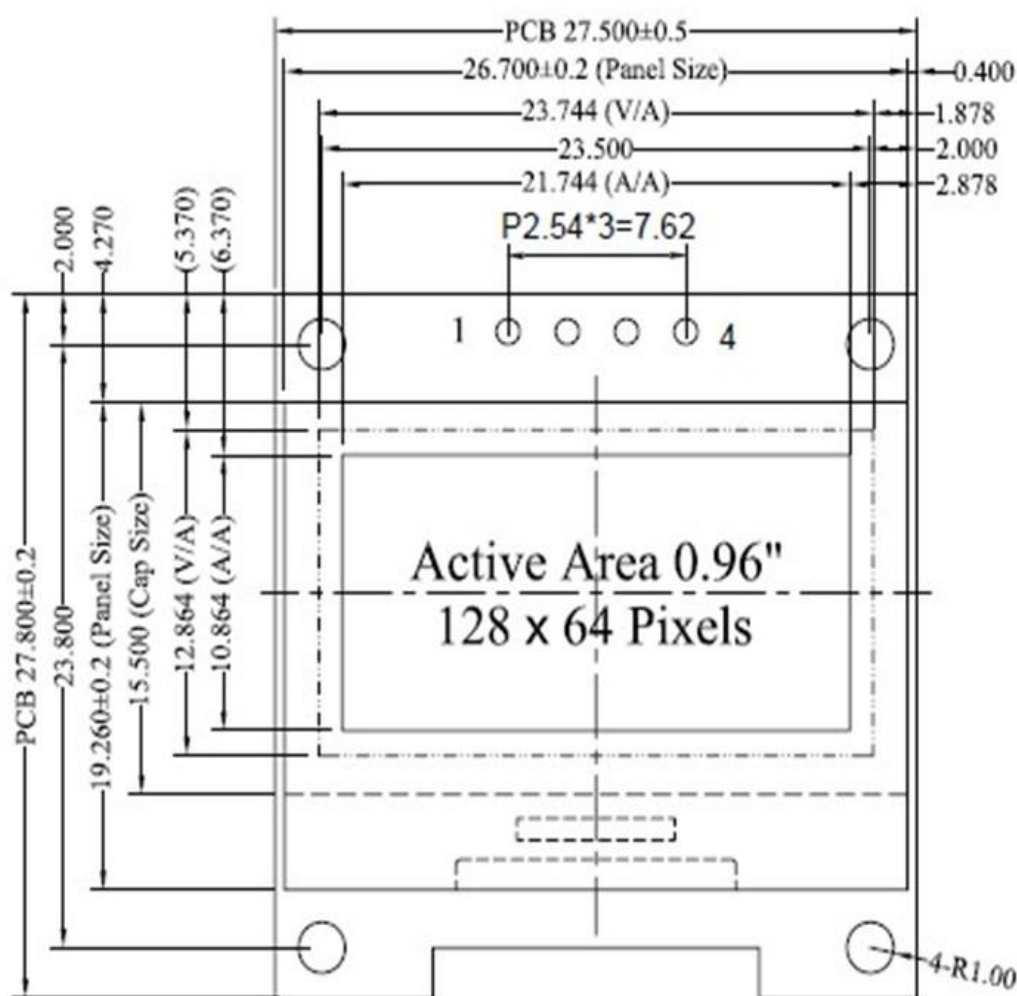
Note:

1. All OLED has been tested twice before shipment. so if it cannot display, Please double check your codes (especially the IIC address) and pin connections. If the pin connections are incorrect, the OLED will burn up in a minute.
2. We offer three kinds of OLED, all blue, all white, yellow and blue. For yellow and blue display, there is a two-pixel gap between the yellow pixels and the blue pixels for a good presentation. The gap doesn't count as rows of pixels.
3. IIC default address is 0x78, but for some usage cases it use 7 bit IIC address mode, So it should set to 0x3C like in Linux and adafruit's library.
4. IIC selectable address is 0x7A, but for some usage case it use 7 bit IIC address mode, So it should set to 0x3D like in Linux and adafruit's library

Hardware Description

1、Dimensions Description:

Module Size: 27.5(L)mm × 27.8(W)mm × 3.0(H)mm
View Area : 23.74mm × 13.204 mm
Effective Area: 128mm × 64 dots
Pixel Size: 0.15mm × 0.15 mm
Pixel Pitch: 0.17mm × 0.17 mm



2、Interface Desceiption:

VCC: 3.3-5V

Support: support@inno-maker.com www.inno-maker.com/wiki
Bulk Price: sales@inno-maker.com

GND: Ground
SCL: Serial Clock signal
SDA: Serial Data signal

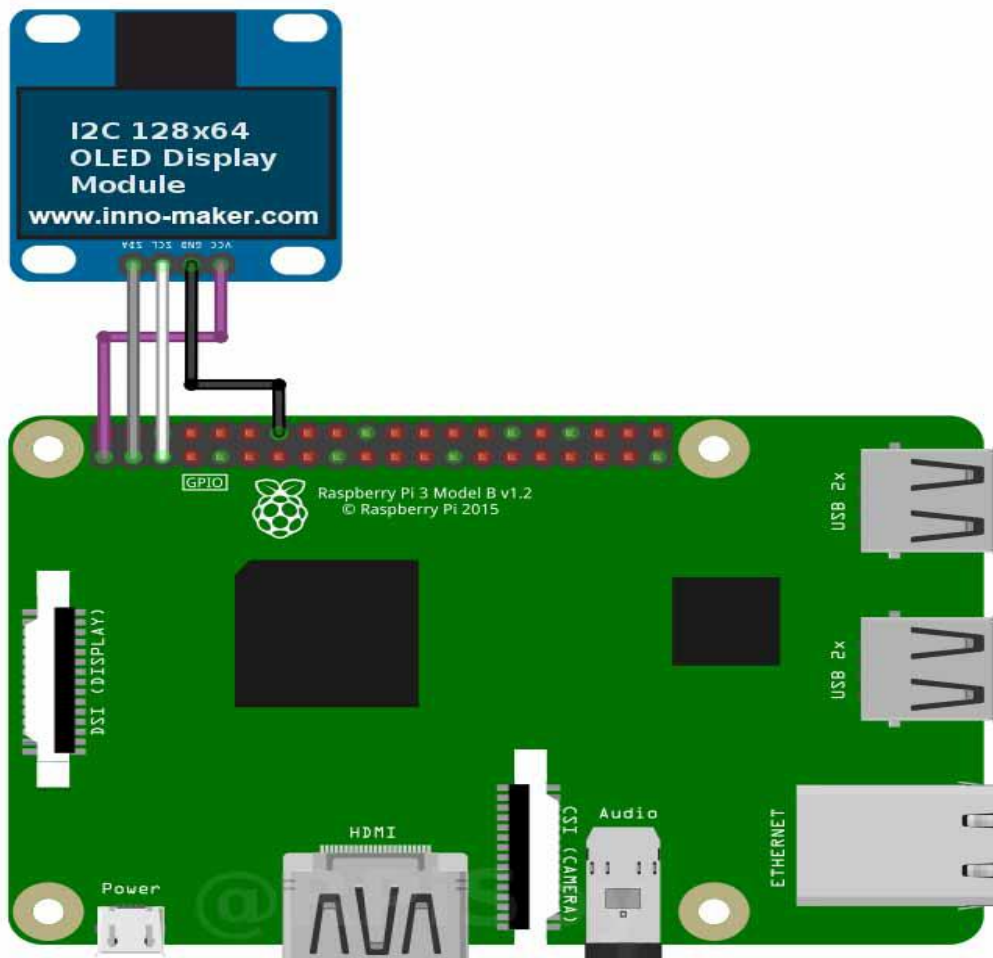
If the pin connections are incorrect, the OLED will burn up in a minute.

3、IIC address selection

Default IIC device address: 0x78 (8 bit IIC address mode) 0x3C (7 bit IIC address mode)
Selectable IIC device address: 0x7A (8 bit IIC address mode) 0x3D (7 bit IIC address mode)

Select the IIC address by the resistance on the back of the board.

4、connect the pins to your Raspberry Pi



5、connect the pins to your Arduino

VCC : goes to 5V

GND : goes to ground

SCL : to I2C SCL (on the Uno, this is A5 on the Mega it is 21 and on the Leonardo digital 3)

RST to digital 4 (you can change this pin in the code, later)

SDA : to I2C SDA (on the Uno, this is A4 on the Mega it is 20 and on the Leonardo digital 2)

6、connect the pins to other mcu

For ease of porting, we provide the driver which is designed by simulation of mcu's any two IO to I2C time sequence. Please refer to the codes of STM32 demo.

Software Description

You can download the OLED software from our wiki link : <http://www.inno-maker.com/wiki>, which comes with demo codes, three kinds of ASCII character library, and demo picture and gadgets.

The drivers of OLED include three files. The code is very simple to adapt to any other microcontroller. We offer STM32 demo and Raspberry pi demo for your reference.

The main function of demo include two parts.

The first part is show you how to display character string to screen.

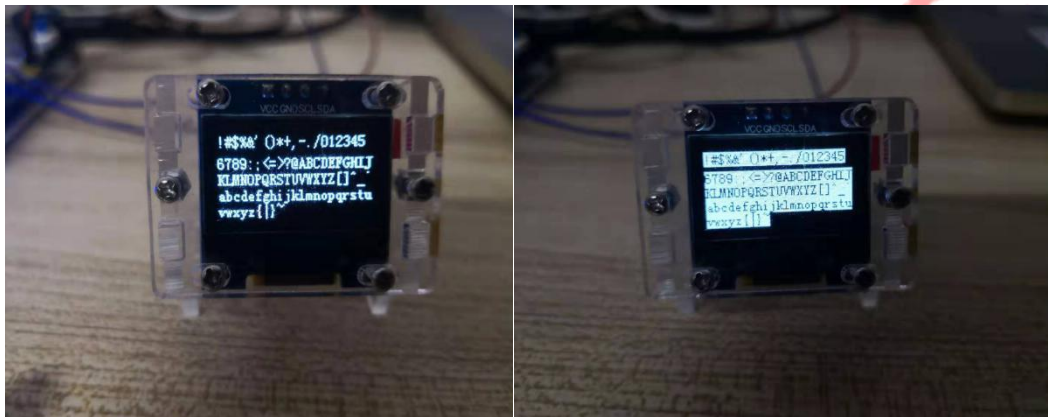
(1)

```
//Display "Hello world" normal mode
Ssd1306ShowString(0,0," Hello World ",16,1);
Ssd1306ShowString(0,16,"welcome to: www.inno-maker.com ",16,1);
Ssd1306Refresh();
MsDelay(3000); //delay 3S to show
//Display "Hello world" highlight mode
Ssd1306ShowString(0,0," Hello World ",16,0);
Ssd1306ShowString(0,16,"welcome to: www.inno-maker.com ",16,0);
Ssd1306Refresh();
MsDelay(3000); //delay 3S to show
//clean all display data for next step
OledCleanAll();
Ssd1306Refresh();
```



(2)

```
//display 12*6 font lib normal mode
Ssd1306ShowString(0,0,"!#$%&'()*+,-./012345",12,1);
Ssd1306ShowString(0,16,"6789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~",12,1);
Ssd1306Refresh();
MsDelay(3000); //delay 3S to show
//display 12*6 font lib highlight mode
Ssd1306ShowString(0,0,"!#$%&'()*+,-./012345",12,0);
Ssd1306ShowString(0,16,"6789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~",12,0);
Ssd1306Refresh();
MsDelay(3000); //delay 3S to show
//clean all display data for next step
OledCleanAll();
Ssd1306Refresh();
```



(3)

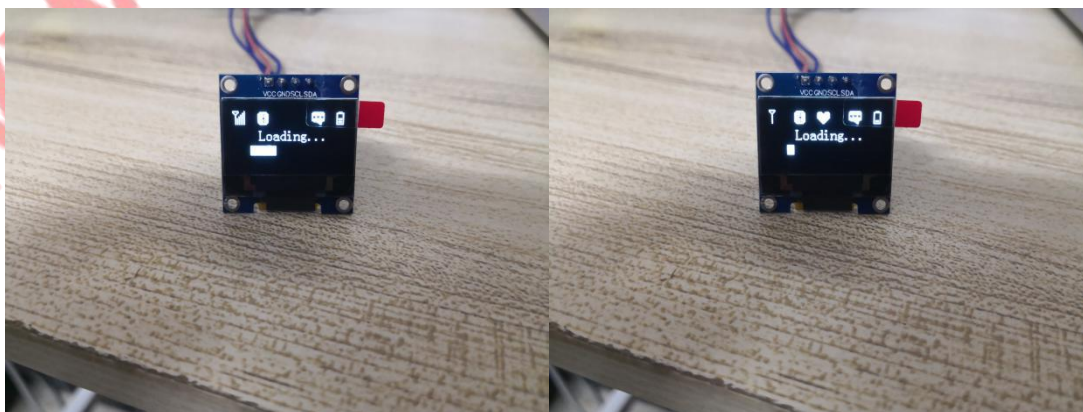
```
// display 16*8 font lib normal mode
Ssd1306ShowString(0,0," !#$%&'()*+,-./0",16,1);
Ssd1306ShowString(0,16,"123456789:;<=>?@",16,1);
Ssd1306ShowString(0,32,"ABCDEFGHIJKLMNOPQRSTUVWXYZ",16,1);
Ssd1306ShowString(0,48,"QRSTUVWXYZ[\]^_`",16,1);
Ssd1306Refresh();
MsDelay(3000); //delay 3S to show

OledCleanAll();
Ssd1306Refresh();
//display 6*8 font lib highlight mode
Ssd1306ShowString(0,0," !#$%&'()*+,-./0",16,0);
Ssd1306ShowString(0,16,"123456789:;<=>?@",16,0);
Ssd1306ShowString(0,32,"ABCDEFGHIJKLMNOPQRSTUVWXYZ",16,0);
Ssd1306ShowString(0,48,"QRSTUVWXYZ[\]^_`",16,0);
Ssd1306Refresh();
MsDelay(3000); //delay 3S to show
OledCleanAll();
```



The second part of code is show you how to display a bitmap to the OLED. You will see some bitmaps update dynamically which often used in project.
For example: battery, antenna, progress Bar, heart symbol, message.

```
//below is application business logic.
while(1)
{
    OledCleanAll();
    UpdateBTStatus();
    ProgressBarUpdate();
    UpdateBatteryStatus();
    UpdateheartStatus();
    UpdateAntStatus();
    UpdateMSGStatus();
    Ssd1306Refresh(); // update to lcd
    MsDelay(1000); //delay 1S to show
}
} « end main »
```

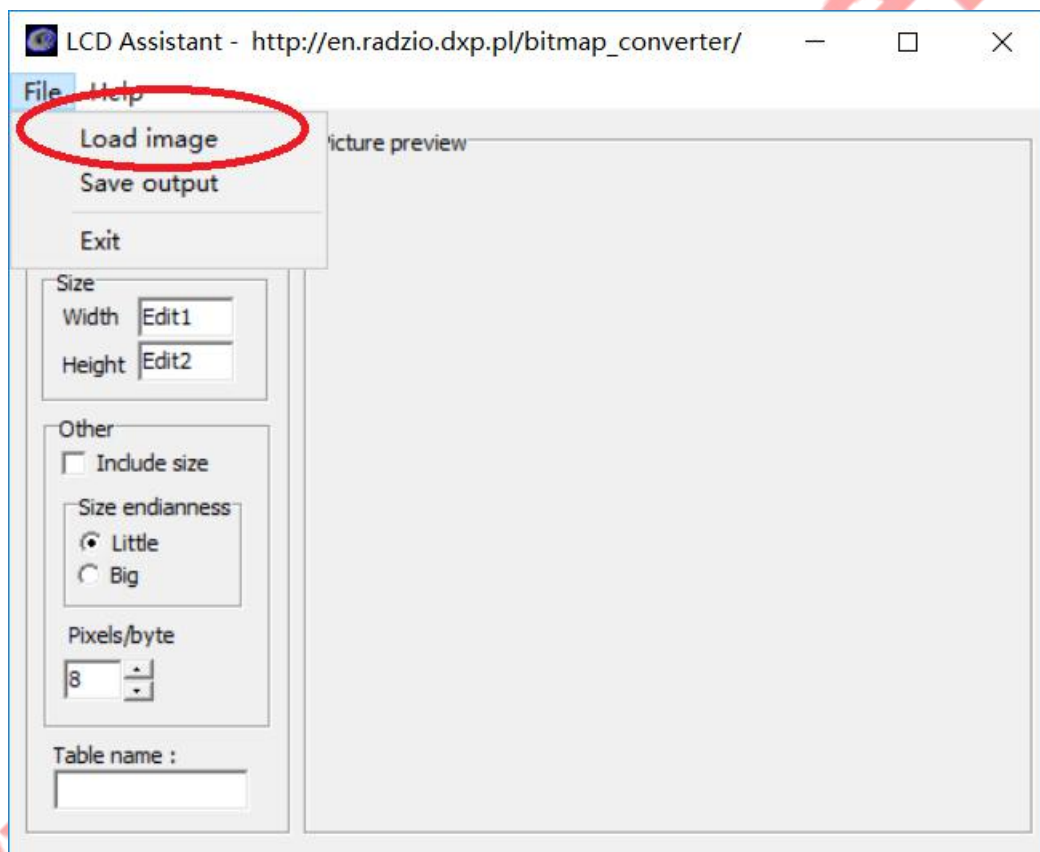


How to use LCD Assistant

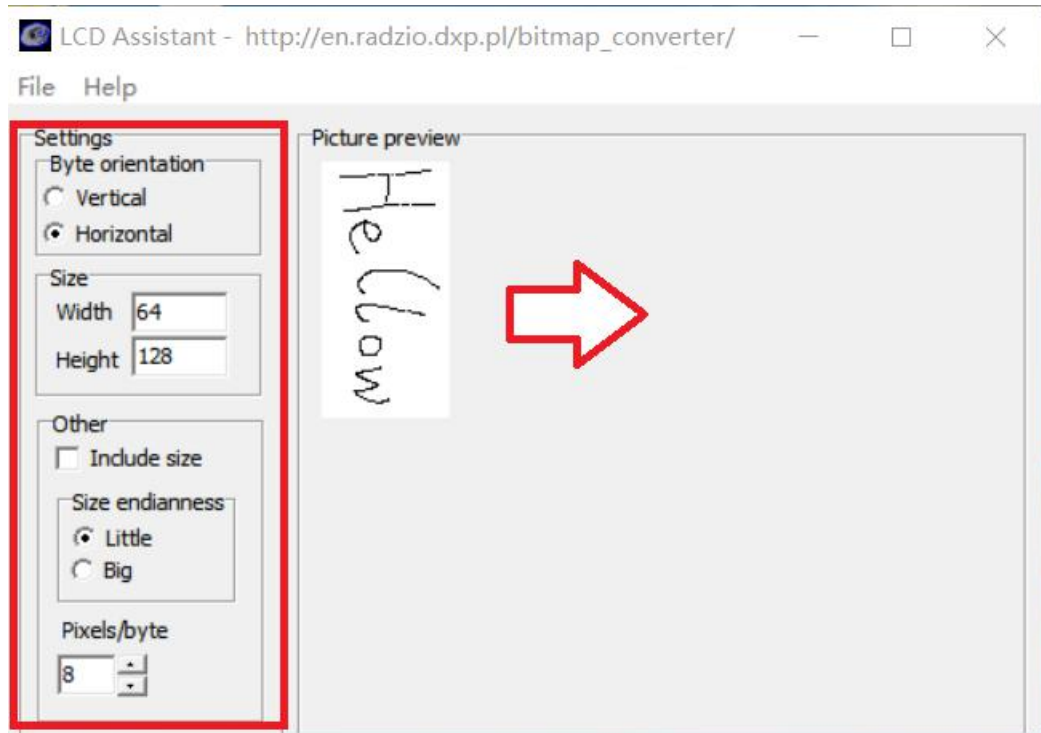
Speak frankly, I think lcd assistant is not a very friendly tool. I prefer PCtoLCD2002 but it is only available in Chinese version. If you have better tools, please kindly email to me (calvin@inno-maker.com), and I will share it to everyone.

We have provided some demo pictures and LCD ASSISTANT software to you, refer to:
http://www.inno-maker.com/wiki/doku.php?id=0.96_inch_iic_serial_oled_module or
<http://www.inno-maker.com/wiki>

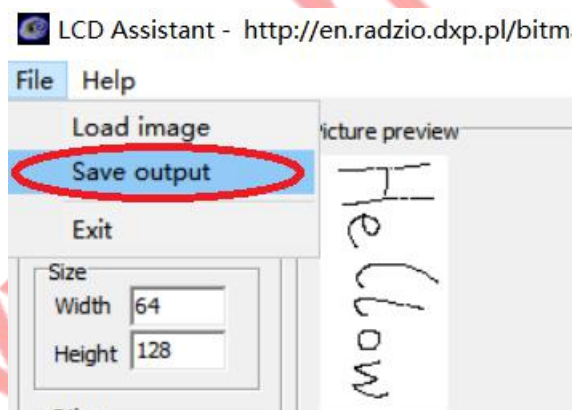
Step 1: load a picture. We use "hellow.bmp" as an example.



Step 2: Select the follow option. This settings only fit for the codes we provide.
You might change the settings for your code or other libraries.



Step 3: Save the output to a "hellow.txt" file.



Step 4: Copy the output array of data to "gui.c" and enter the following code.

```
//-----
// File generated by LCD Assistant
// http://en.radzio.dxp.pl/bitmap_converter/
//-----

const unsigned char hellow [] = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0xC0, 0x00,
0x00, 0x00, 0x01, 0xEF, 0xD7, 0xC0, 0x00, 0x00, 0x7F, 0xFE, 0x00, 0x40, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xFE, 0xF7, 0xFF, 0x80,
0x00, 0x00, 0x00, 0x7E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x80, 0x00, 0x00, 0x00, 0x00,
0x00, 0x1F, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x1F, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x74, 0x70, 0x00, 0x00, 0x00,
0x00, 0x00, 0xC4, 0x18, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x84, 0x08, 0x00, 0x00, 0x00,
0x00, 0x01, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00, 0x02, 0x06, 0x04, 0x00, 0x00, 0x00, 0x00,
0x00, 0x02, 0x02, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x02, 0x08, 0x00, 0x00, 0x00,
0x00, 0x02, 0x01, 0x98, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0xF0, 0x00, 0x00, 0x00,
0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,
}
```

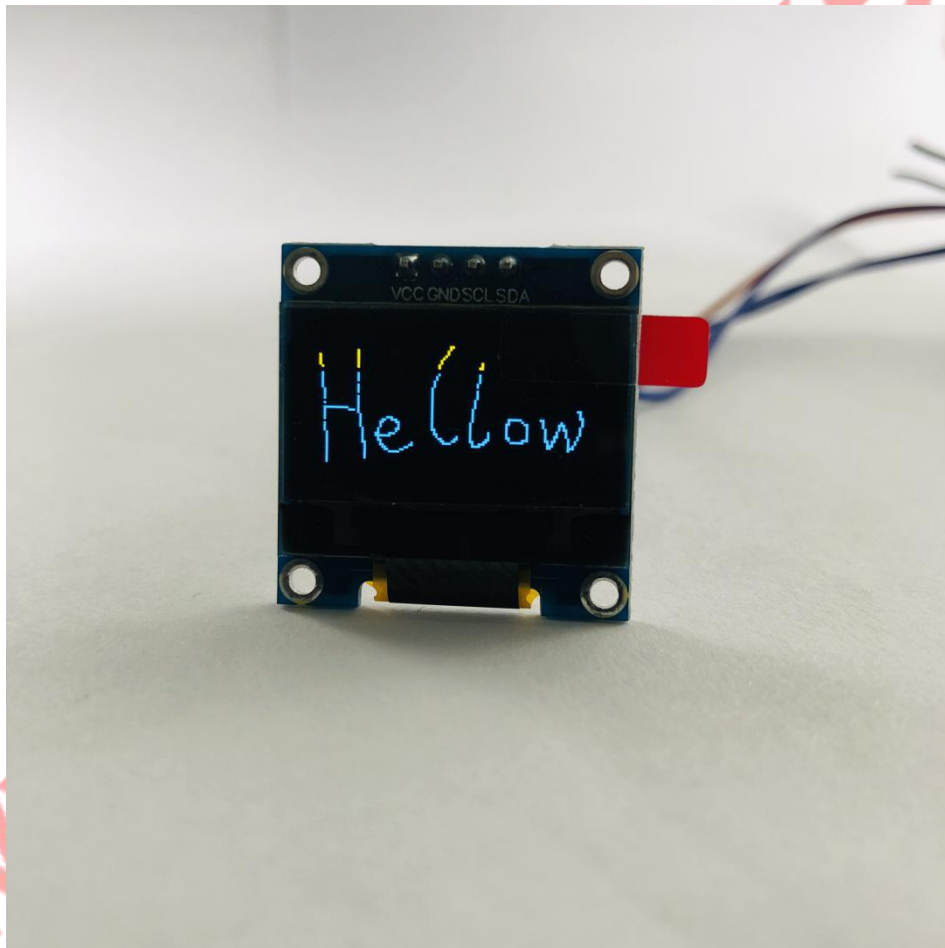
```
const FONT_CHAR_INFO hellow_descriptors[] =
{
    {128,0},
};

const FONT_INFO hellow_info =
{
    8,
    '0',
    hellow_descriptors,
    hellow,
};
```

Step 5: Call the functions to display "hellow".

```
void ShowHellow()  
{  
    DrawBitMapString(0,0,&hellow_info,"0");  
}
```

```
ShowHellow(); // show "hellow"  
Ssd1306Refresh(); // refresh display
```



Reference for OLED I2c Display with Raspberry :

<https://www.raspberrypi-spy.co.uk/2018/04/i2c-oled-display-module-with-raspberry-pi>

<https://circuitdigest.com/microcontroller-projects/ssd1306-oled-display-with-raspberry-pi>

<https://tech.scargill.net/ssd1306-with-python/>

<https://www.youtube.com/watch?v=AW2Hu0lep0s>

Reference for OLED I2c Display with Arduino:

<https://www.instructables.com/id/Monochrome-096-i2c-OLED-display-with-arduino-SSD13/>

<https://startingelectronics.org/tutorials/arduino/modules/OLED-128x64-I2C-display/>

Version Descriptions

Version	Description	Author	Date	E-mail
V1.0.0.0	First edition	Calvin	2019.03.01	calvin@inno-maker.com

If you have any suggestions, ideas, codes and tools please feel free to email to me. I will update the user manual and record your name and E-mail in list. Look forward to your letter and kindly share.