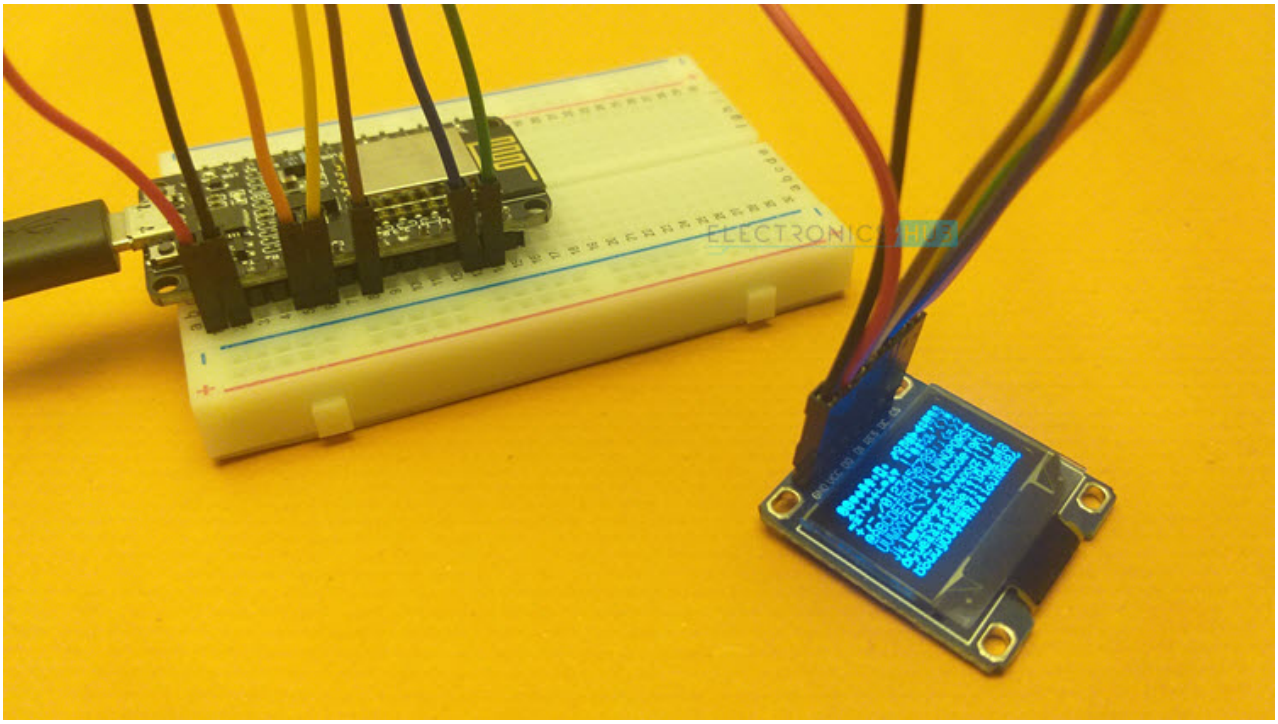
How to Interface OLED Display with NodeMCU ESP8266?

electronicshub.org/nodemcu-esp8266-oled-display

In this tutorial, we will learn how to connect an SSD1306 OLED Display with ESP8266. This particular OLED Display is a 7 Pin SPI type. So, we will use NodeMCU development board instead of regular ESP-01 Module. We will learn the pins needed for interfacing OLED Display with ESP8266 NodeMCU, display bitmap image and understand how ESP8266 NodeMCU OLED Display interface works.

If you want to interface OLED Display with ESP32 Development Board, then I made a dedicated tutorial. Check it out.



Introduction

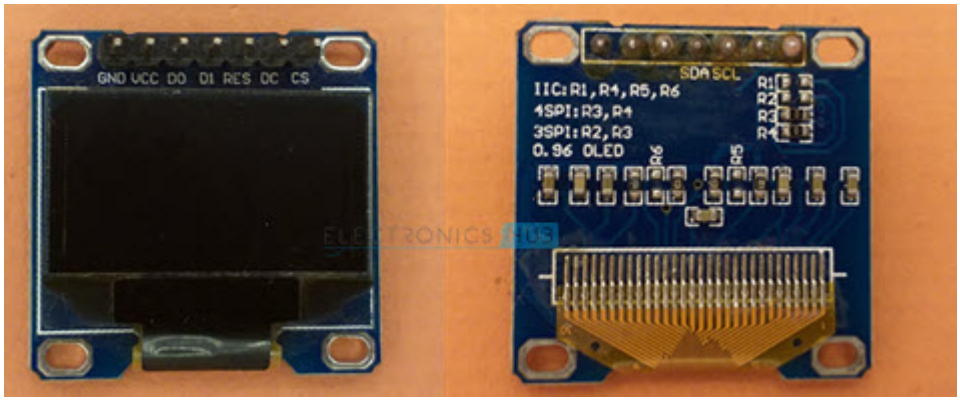
Organic Light Emitting Diode or OLED in short is a display technology which is quite popular in Mobile Phones and Televisions. OLED Displays have the ability to generate its own backlight at individual pixel level. This is in contrast to LCD Displays, which require a separate backlight to light up the panel.

Speaking of contrast, since individual pixel emits its own light, it can be turned off as well to produce deep black image. So, the contrast ratio of an OLED Display is much better than a regular LCD.

Due to lower power consumption (only pixels which are lit up draw current), OLED displays are also popular in battery operated devices like Smart Watches, Health Trackers and other wearables.

A Brief Note on SSD1306 OLED Display

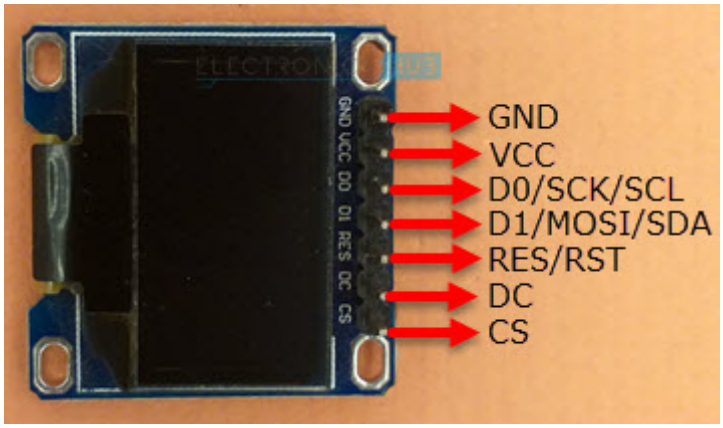
For DIY projects, there are some OLED Displays available in the market, usually with a resolution of 128×64 pixels or 128×32 pixels. Also, there are two types of OLED Displays based on the communication interface they provide. One type uses I²C and the other uses SPI.



In this project, we will use an SPI type OLED Display with 128×64 pixel resolution and measuring 0.96” diagonally. This OLED display is based on SSD1306 OLED Driver IC.

You have to note the SSD1306 OLED Driver IC can be configured to communicate either in I²C Interface or in SPI Interface. So, the same OLED Panel and same OLED Driver IC can result in two devices: one is a 4 pin I²C type and the other is a 7 pin SPI type.

In fact, you can even modify one type to another simply by soldering / de-soldering some SMD resistors.

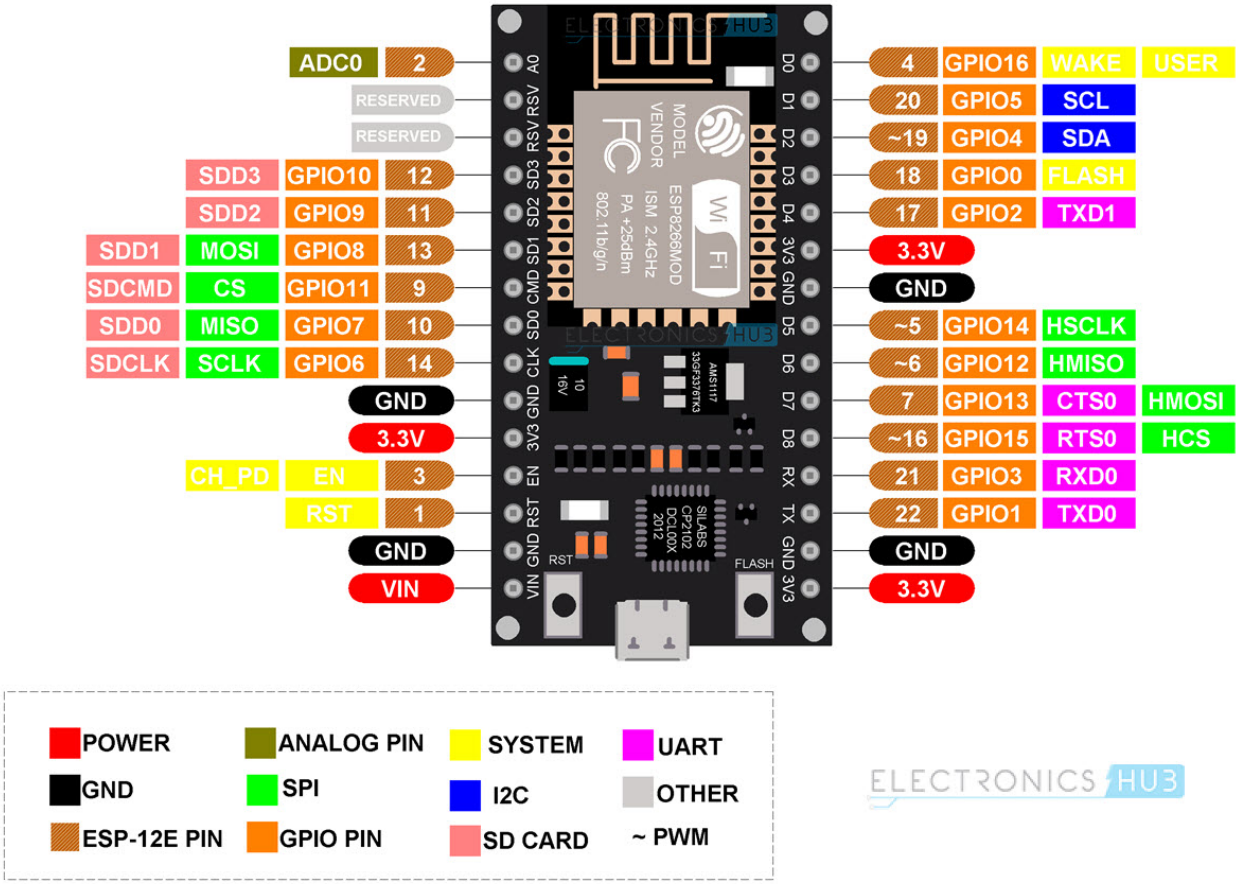


Identifying SPI Pins in ESP8266 NodeMCU

As mentioned before, this particular OLED is of SPI type. So, we have to search and configure SPI pins in ESP8266. Since, the ESP-01 Module doesn’t fanout all the pins of ESP8266EX SoC, we will be using the NodeMCU Development Board.

If you remember the ESP8266 NodeMCU Pinout tutorial, ESP8266 has two SPI interfaces: SPI and HSPI. But SPI interface is already used by the ESP8266EX SoC for connecting with SPI Flash IC in the ESP-12E Module.

So, we are left with HSPI interface.



Take a look at the pinout diagram of ESP8266 NodeMCU. HSPI Pins are as follows:

HSPI	GPIO	NodeMCU
HSPI_MISO	GPIO 12	D6
HSPI_MOSI	GPIO 13	D7
HSPI_CLK	GPIO 14	D5
HSPI_CS	GPIO 15	D8

ESP8266 NodeMCU OLED Display Interface

Now that we have identified the SPI Pins of ESP8266 NodeMCU, we can now start with interfacing OLED display with ESP8266. But if you notice the pinout of SPI OLED Display, there are two more pins other than the SPI Pins.

They are RST (Reset) and DC (Data / Command) pins. Since these pins are just control pins, we can use any GPIO Pins of ESP8266 NodeMCU. As GPIO 4 (D2) and GPIO 5 (D1) are free, let us use them for this purpose.

So, finally, the connection between ESP8266 NodeMCU and SPI OLED Display looks something like this:

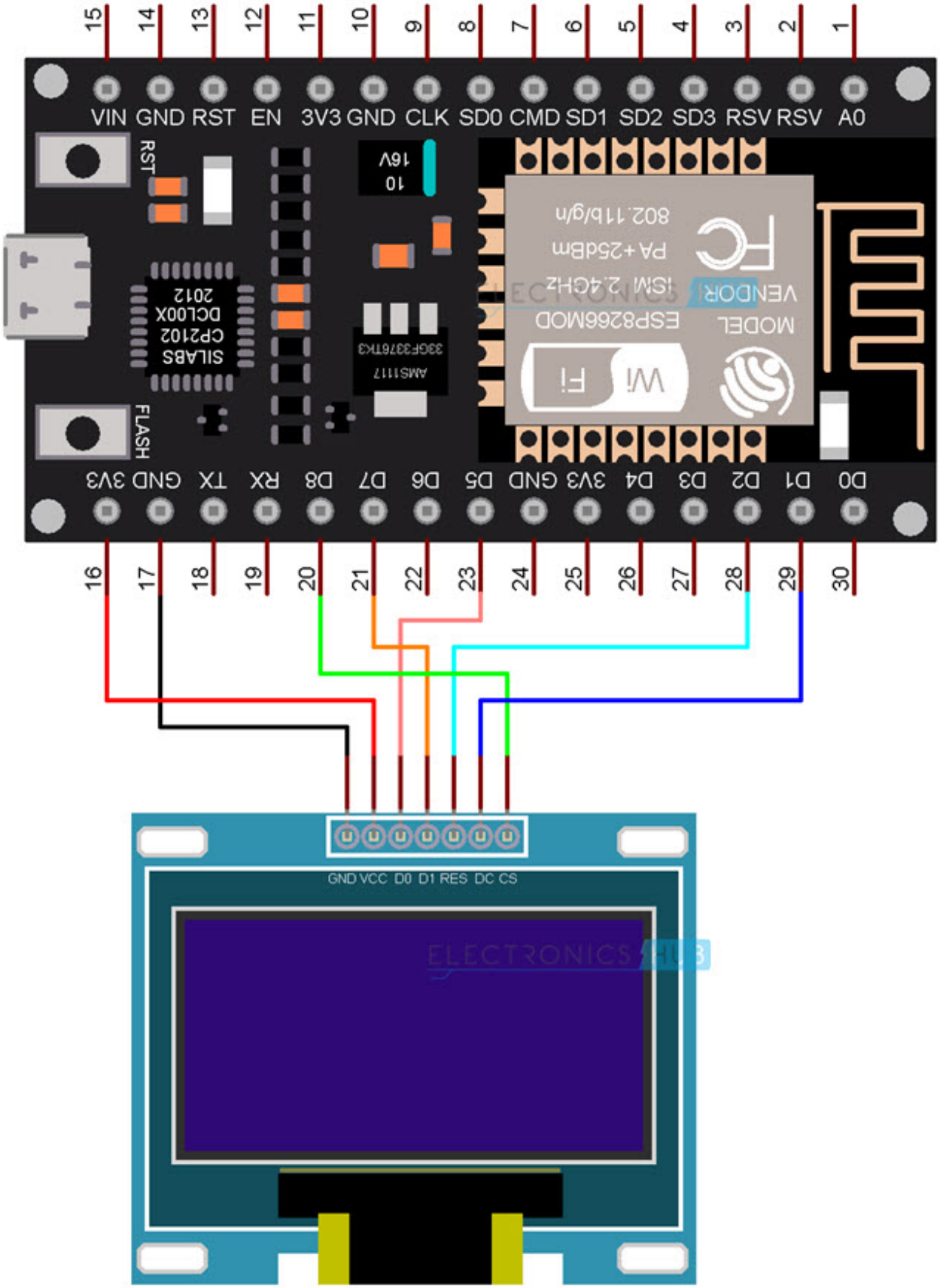
OLED Display	ESP8266 NodeMCU
GND	GND
VCC	3.3V
D0 (SCK / CLK)	GPIO 14 (D5)
D1 (MOSI)	GPIO 13 (D7)
RES	GPIO 4 (D2)
DC	GPIO 5 (D1)
CS	GPIO 15 (D8)

Components Required

- ESP8266 NodeMCU Development Board
- OLED Display
- Breadboard
- Connecting Wires
- Micro-USB Cable

Circuit Diagram

The following image shows the circuit diagram of ESP8266 NodeMCU OLED Display Interface. This circuit is specific to SPI OLED Display.

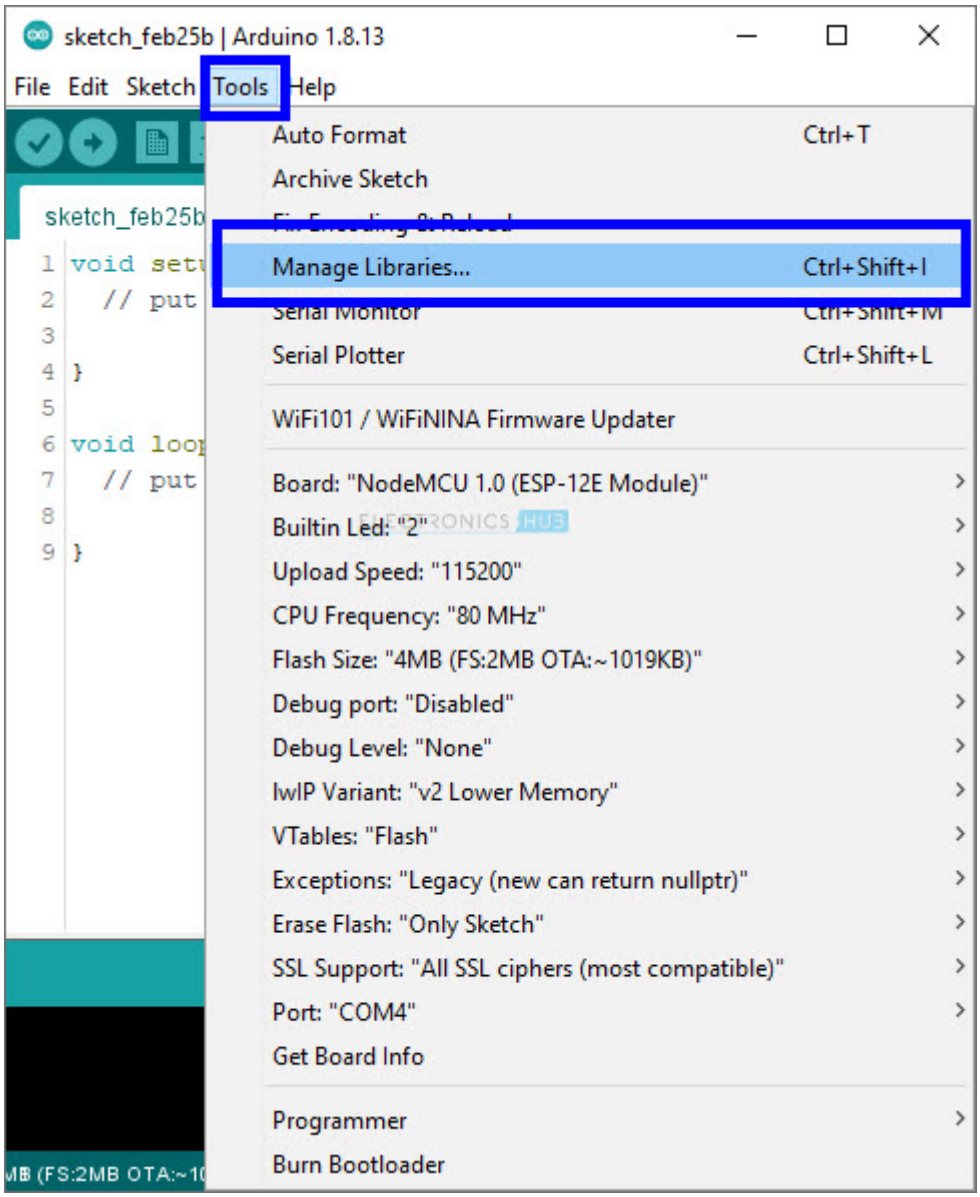


Preparing Arduino IDE

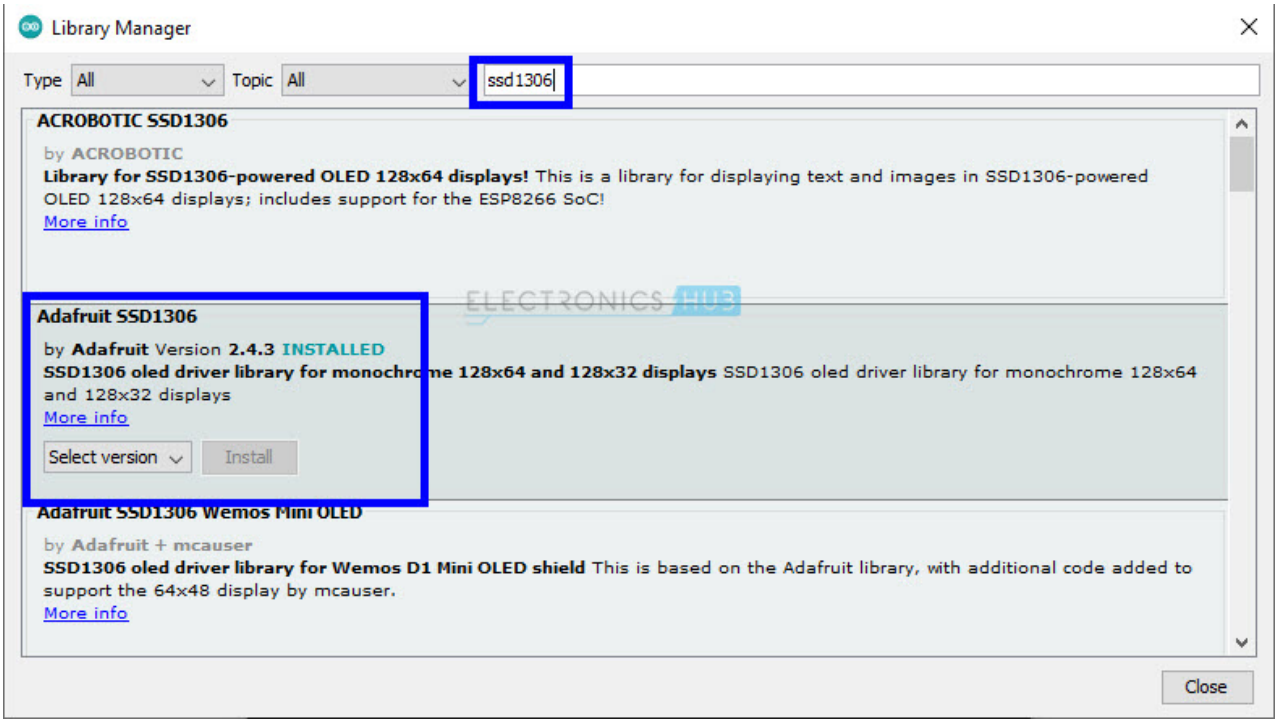
If you worked with OLED Displays (either I²C or SPI type) before, then you probably would have downloaded the necessary libraries for Arduino IDE. Let me take you through those steps once again.

Specifically, you have to download two libraries. One is for the OLED Driver SSD1306 and the other one is for displaying some basic graphics.

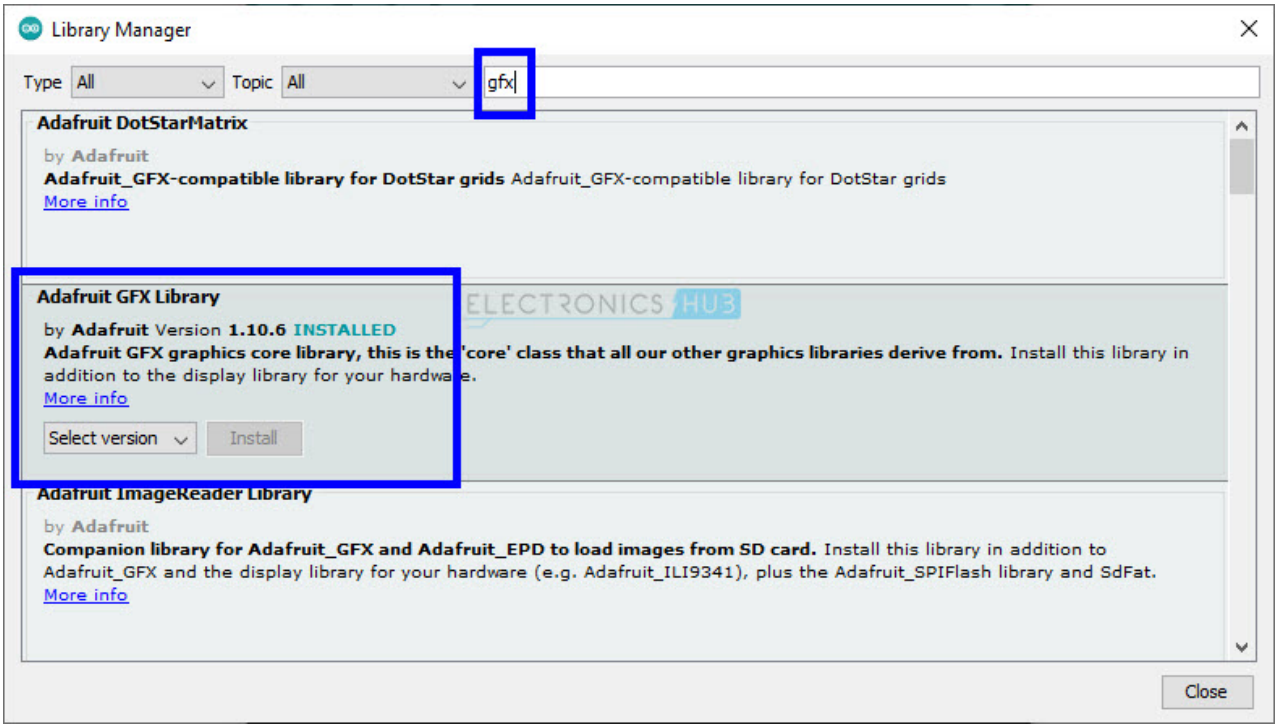
Open Arduino IDE and open library manager by selecting Tools -> Manage Libraries. . .



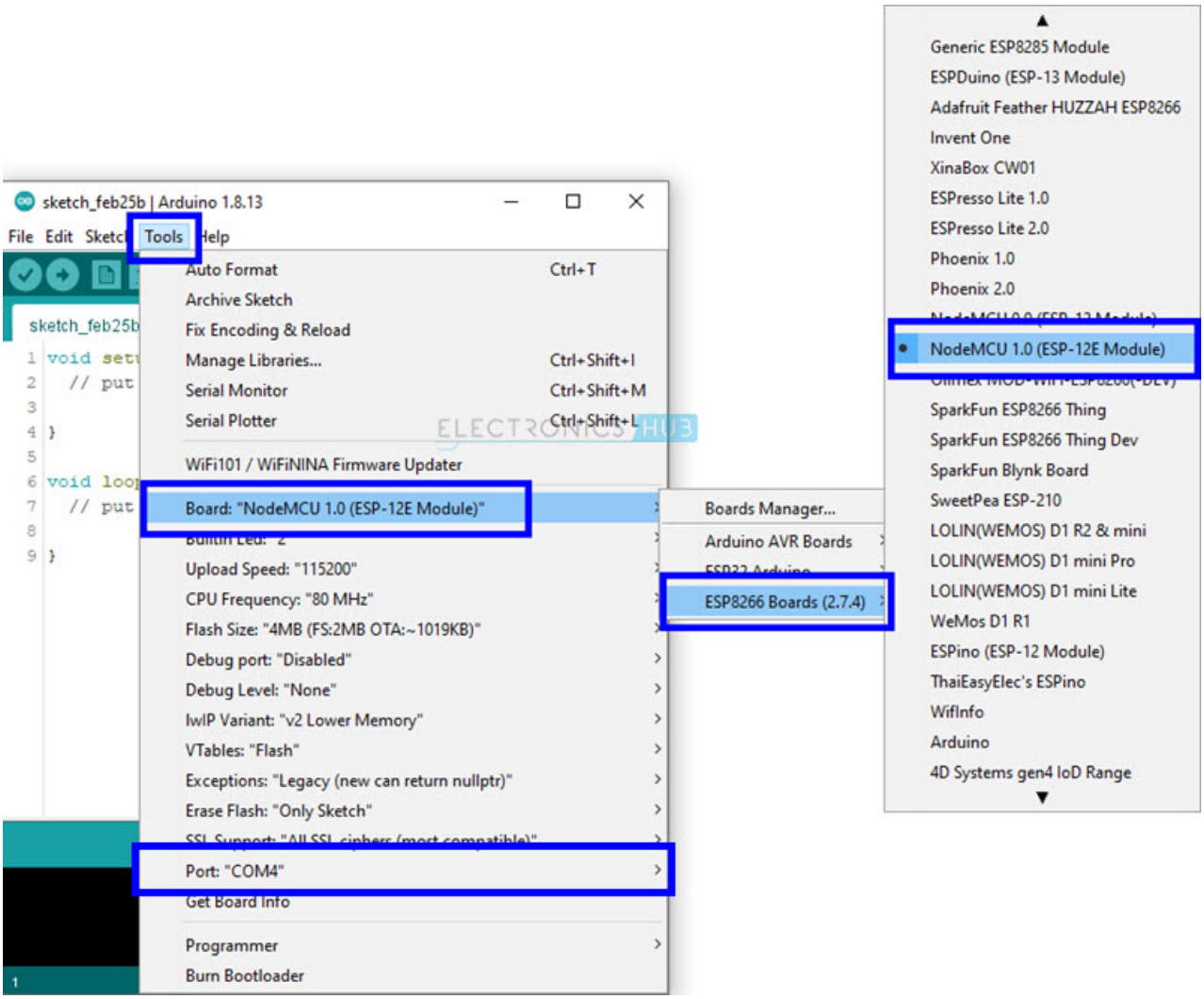
Search for ‘ssd1306’ and install ‘Adafruit SSD1306’ library. As you can see from the image, I already installed it.



Next, search for ‘gfx’ and install ‘Adafruit GFX’ library. After this, close the library manager.



Make sure that NodeMCU board is selected. If not, then you can select this board by going to Tools -> Board -> ESP8266 Boards and click on NodeMCU 1.0 (ESP-12E Module). Also, select the correct COM Port (if you already made the connections and connected NodeMCU to the computer).



We are now ready to display some stuff on the OLED Display.

Displaying Text

First, we will write a small program which will display some basic text related info. This includes simple text, inverted text, scrolling text, ASCII Characters and also two sizes of fonts.

Code

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
```

```
#define OLED_MOSI 13
#define OLED_CLK 14
#define OLED_DC 5
#define OLED_CS 15
#define OLED_RESET 4
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);

void setup()
{
Serial.begin(115200);
if(!display.begin(SSD1306_SWITCHCAPVCC))
{
Serial.println(F("SSD1306 allocation failed"));
for(;;);
}

display.clearDisplay();
display.display();
delay(100);
}

void loop()
{
AllPixels();
TextDisplay();
InvertedTextDisplay();
ScrollText();
DisplayChars();
TextSize();
```

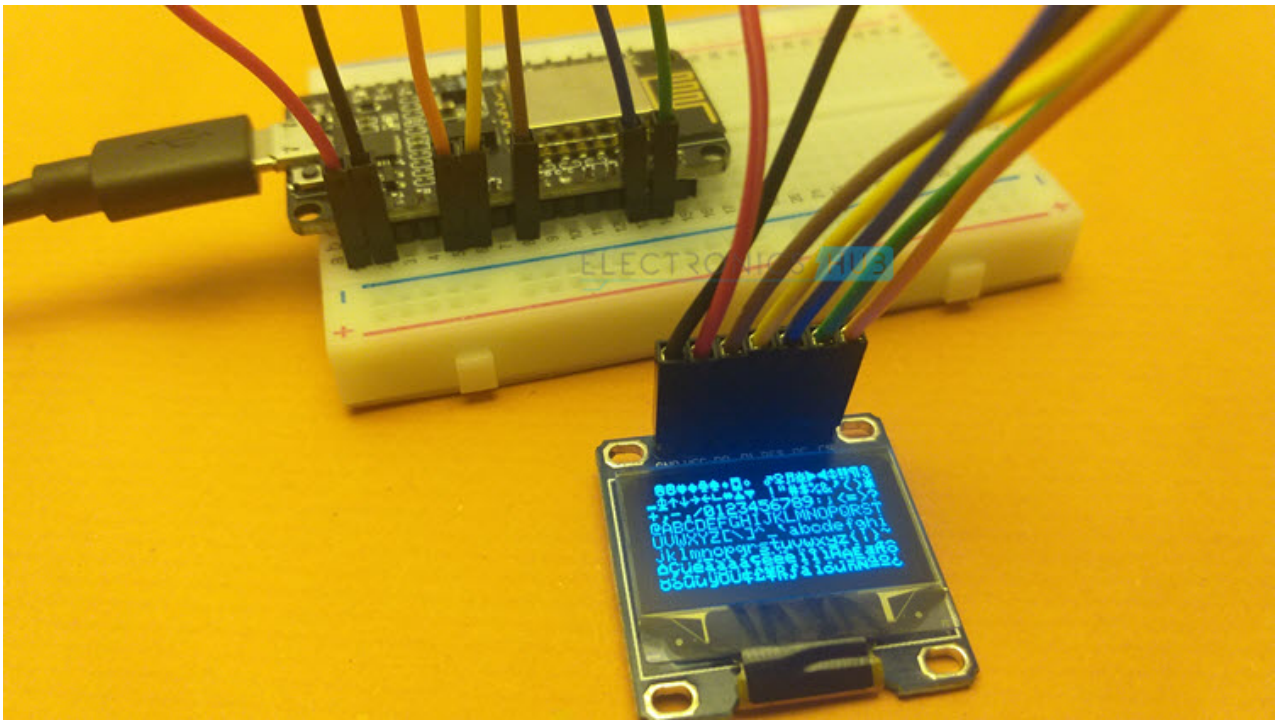
```
void AllPixels()
{
int i;
int j;
display.clearDisplay();
for(i=0;i<128;i++)
{
for(j=0;j<64;j++)
{
display.drawPixel(i, j, SSD1306_WHITE);

}
display.display();
delay(20);
}

for(i=0;i<128;i++)
{
for(j=0;j<64;j++)
{
display.drawPixel(i, j, SSD1306_BLACK);

}
display.display();
delay(20);
}

}

void TextDisplay()
```

```
{
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(5,28);
display.println("Electronics Hub");
display.display();
delay(3000);
}

void InvertedTextDisplay()
{
display.clearDisplay();
display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
display.setCursor(5,28);
display.println("Electronics Hub");
display.display();
delay(3000);
}

void ScrollText()
{
display.clearDisplay();
display.setCursor(0,0);
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.println("This is a");
display.println("Scrolling");
display.println("Text!");
```

```
display.display();
delay(100);
display.startscrollright(0x00, 0x0F);
delay(2000);
//display.stopscroll();
//delay(1000);
display.startscrollleft(0x00, 0x0F);
delay(2000);
//display.stopscroll();
//delay(1000);
display.startscrolldiagright(0x00, 0x0F);
delay(2000);
display.startscrolldiagleft(0x00, 0x0F);
delay(2000);
display.stopscroll();
}

void DisplayChars()
{
display.clearDisplay();

display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0, 0);
display.cp437(true);

for(int16_t i=0; i<256; i++)
{
if(i == '\n')
{
```

```
    display.write(' ');
    }
    else
    {
    display.write(i);
    }
    }

    display.display();
    delay(3000);
    }
    void TextSize()
    {
    display.clearDisplay();

    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0,0);
    display.println(F("Size: 1"));
    display.println(F("ABC"));

    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.println("Size: 2");
    display.println(F("ABC"));

    display.display();
    delay(3000);
    }
```

view raw ESP8266-NodeMCU-OLED-Text.ino hosted with ❤ by GitHub



Displaying Graphics

Next, we will use the GFX library to display some basic shapes like rectangle, filled rectangle, round rectangle, filled round rectangle, circle, filled circle, triangle and filled triangle.

Code

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

#define OLED_MOSI 13
#define OLED_CLK 14
#define OLED_DC 5
#define OLED_CS 15
#define OLED_RESET 4
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
```

```
OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);

void setup()
{
Serial.begin(115200);
if(!display.begin(SSD1306_SWITCHCAPVCC))
{
Serial.println(F("SSD1306 allocation failed"));
for(;;);
}

display.clearDisplay();
display.display();
delay(100);
}

void loop()
{
DrawRectangle();
DrawFilledRectangle();
DrawRoundRectangle();
DrawFilledRoundRectangle();
DrawCircle();
DrawFilledCircle();
DrawTriangle();
DrawFilledTriangle();
}


void DrawRectangle()
{
```

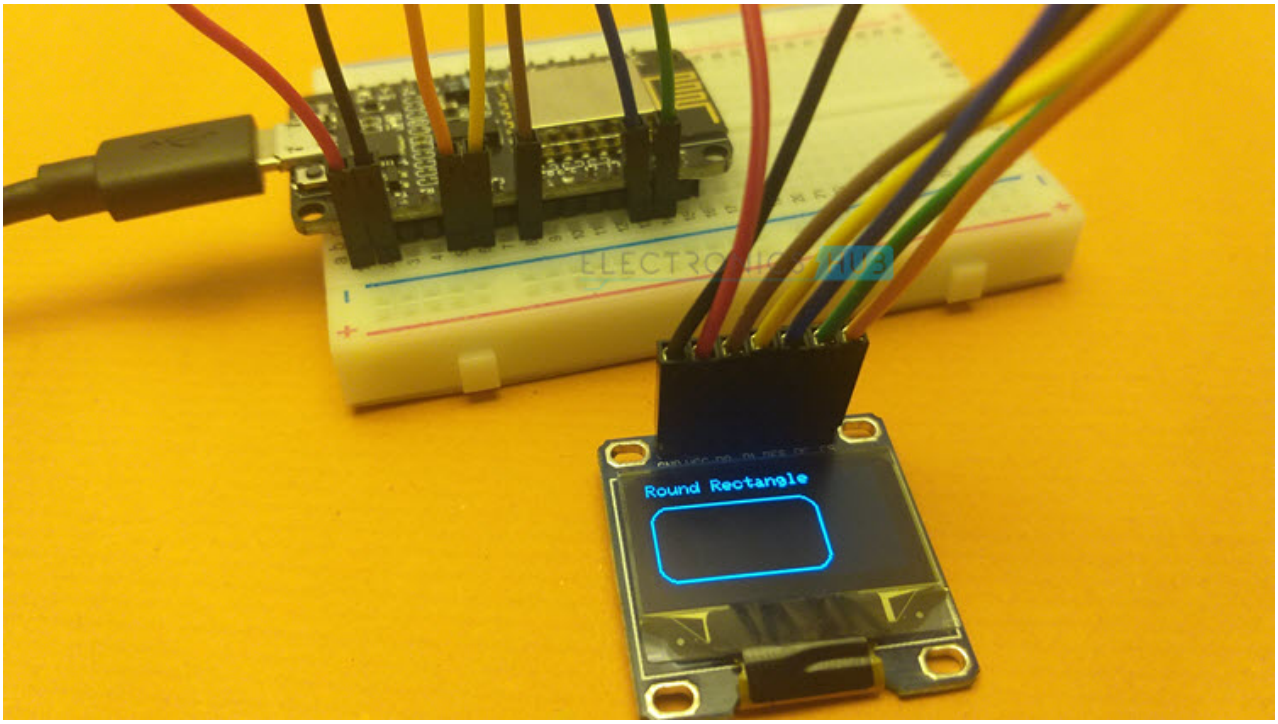
```
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.println("Rectangle");
  display.drawRect(0, 15, 90, 45, SSD1306_WHITE);
  display.display();
  delay(2000);
  }

  void DrawFilledRectangle()
  {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.println("Filled Rectangle");
  display.fillRect(0, 15, 90, 45, SSD1306_WHITE);
  display.display();
  delay(2000);

  }

  void DrawRoundRectangle()
  {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,0);
  display.println("Round Rectangle");
```

```
display.drawRoundRect(0, 15, 90, 45, 10, SSD1306_WHITE);
display.display();
delay(5000);
}

void DrawFilledRoundRectangle()
{
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0,0);
display.println("Filled Round Rect");
display.fillRoundRect(0, 15, 90, 45, 10, SSD1306_WHITE);
display.display();
delay(2000);

}

void DrawCircle()
{
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0,0);
display.println("Circle");
display.drawCircle(30, 36, 25, SSD1306_WHITE);
display.display();
delay(2000);
}
void DrawFilledCircle()
```

```
{
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0,0);
display.println("Filled Circle");
display.fillCircle(30, 36, 25, SSD1306_WHITE);
display.display();
delay(2000);

}

void DrawTriangle()
{
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0,0);
display.println("Triangle");
display.drawTriangle(30, 15, 0, 60, 60, 60, SSD1306_WHITE);
display.display();
delay(2000);
}

void DrawFilledTriangle()
{
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(0,0);
```

```
    display.println("Filled Triangle");
    display.fillTriangle(30, 15, 0, 60, 60, 60, SSD1306_WHITE);
    display.display();
    delay(2000);
    }
```

view raw ESP8266-NodeMCU-OLED-Graphics.ino hosted with ❤ by GitHub



Displaying Bitmap Image

Finally, to demonstrate the bitmap displaying ability of SSD1306 OLED Display, let us take a small bitmap image and generate equivalent bytecode for the image (using tools like GIMP or any online tool).

We have to insert this byte code in our code and use the 'drawBitmap' function to display the image on the OLED Display.

Code

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

```
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

#define OLED_MOSI 13
#define OLED_CLK 14
#define OLED_DC 5
#define OLED_CS 15
#define OLED_RESET 4
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);

const unsigned char electronicshub_logo [] PROGMEM = {
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, 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, 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,
```

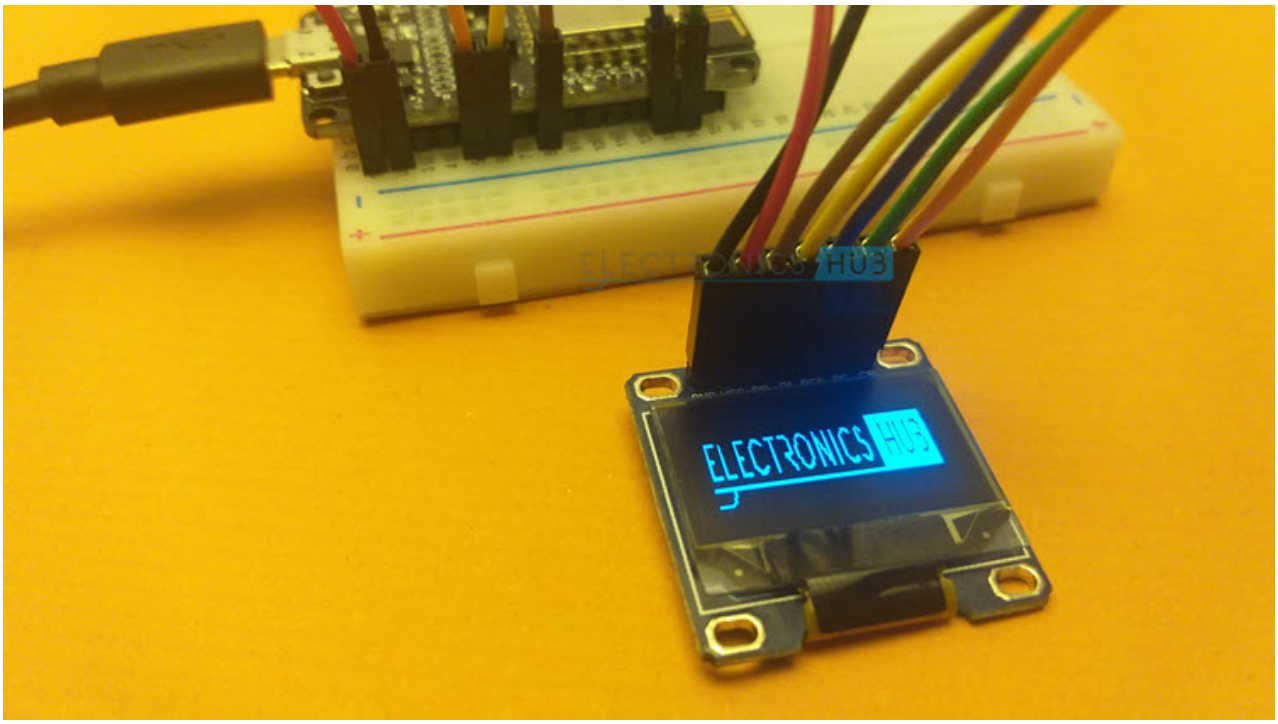
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x03, 0xe0, 0x07, 0xc3, 0x87, 0xcf, 0x03, 0x02, 0x11, 0x07, 0x04, 0x0f, 0xbd, 0xbd, 0xc7, 0x80,
0x03, 0xe4, 0x07, 0xc3, 0xc7, 0xcf, 0x07, 0x82, 0x11, 0x07, 0x8e, 0x0f, 0xbd, 0xbd, 0xc7, 0x80,
0x02, 0x04, 0x04, 0x06, 0xc1, 0x01, 0x84, 0xc2, 0x11, 0x0d, 0x8a, 0x0f, 0xbd, 0xbd, 0xf3, 0x80,
0x02, 0x04, 0x04, 0x04, 0x61, 0x00, 0x88, 0x42, 0x11, 0x08, 0x8a, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x0c, 0x21, 0x00, 0x88, 0x43, 0x11, 0x10, 0x08, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x00, 0x88, 0x23, 0x11, 0x10, 0x08, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x00, 0x88, 0x23, 0x11, 0x10, 0x08, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x00, 0x90, 0x23, 0x11, 0x10, 0x08, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x00, 0x90, 0x22, 0x91, 0x10, 0x08, 0x0f, 0xbd, 0xbd, 0xf7, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x00, 0x90, 0x22, 0x91, 0x10, 0x0e, 0x0f, 0xbd, 0xbd, 0xc7, 0x80,
0x03, 0x84, 0x07, 0x08, 0x01, 0x00, 0x90, 0x22, 0x91, 0x10, 0x06, 0x0f, 0x81, 0xbd, 0xc7, 0x80,

0x03, 0x84, 0x07, 0x08, 0x01, 0x07, 0x90, 0x22, 0xd1, 0x10, 0x03, 0x0f, 0x81, 0xbd, 0xf7, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x07, 0x10, 0x22, 0x51, 0x10, 0x01, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x06, 0x10, 0x22, 0x51, 0x10, 0x01, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x02, 0x10, 0x22, 0x51, 0x10, 0x01, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x02, 0x08, 0x22, 0x31, 0x10, 0x01, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x08, 0x01, 0x01, 0x08, 0x22, 0x31, 0x10, 0x01, 0x0f, 0xbd, 0xbd, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x0c, 0x21, 0x01, 0x08, 0x42, 0x31, 0x10, 0x01, 0x0f, 0xbd, 0xdb, 0xfb, 0x80,
0x02, 0x04, 0x04, 0x04, 0x61, 0x01, 0x88, 0x42, 0x31, 0x08, 0x99, 0x0f, 0xbd, 0xdb, 0xf3, 0x80,
0x02, 0x04, 0x04, 0x06, 0xc1, 0x00, 0x84, 0xc2, 0x11, 0x0d, 0x8b, 0x0f, 0xbd, 0xc3, 0xc7, 0x80,
0x03, 0xe7, 0xc7, 0xc3, 0xc1, 0x00, 0x87, 0x82, 0x11, 0x07, 0x8e, 0x0f, 0xbd, 0xe7, 0xc7, 0x80,
0x03, 0xe7, 0xc7, 0xc3, 0x81, 0x00, 0x83, 0x02, 0x11, 0x07, 0x06, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xff, 0xff, 0xff, 0x80,
0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80,
0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x80,
0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```
0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x07, 0xf0, 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,
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, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00
};

void setup()
{
Serial.begin(115200);
if(!display.begin(SSD1306_SWITCHCAPVCC))
```

```
{
Serial.println(F("SSD1306 allocation failed"));
for(;;);
}

display.clearDisplay();
display.display();
delay(100);

display.clearDisplay();
display.drawBitmap(0, 0, electronicshub_logo, SCREEN_WIDTH,
SCREEN_HEIGHT, SSD1306_WHITE);
display.display();
//delay(5000);

}

void loop()
{

}
```

view raw ESP8266-NodeMCU-OLED-Bitmap.ino hosted with ❤ by GitHub



Conclusion

A simple project demonstrating ESP8266 NodeMCU OLED Display Interface. You learned how to connect ESP8266 with an SPI type OLED Display, download the necessary library files, display different types of text file, some basic graphics and also a bitmap image.

Leave a Reply

Your email address will not be published. Required fields are marked *