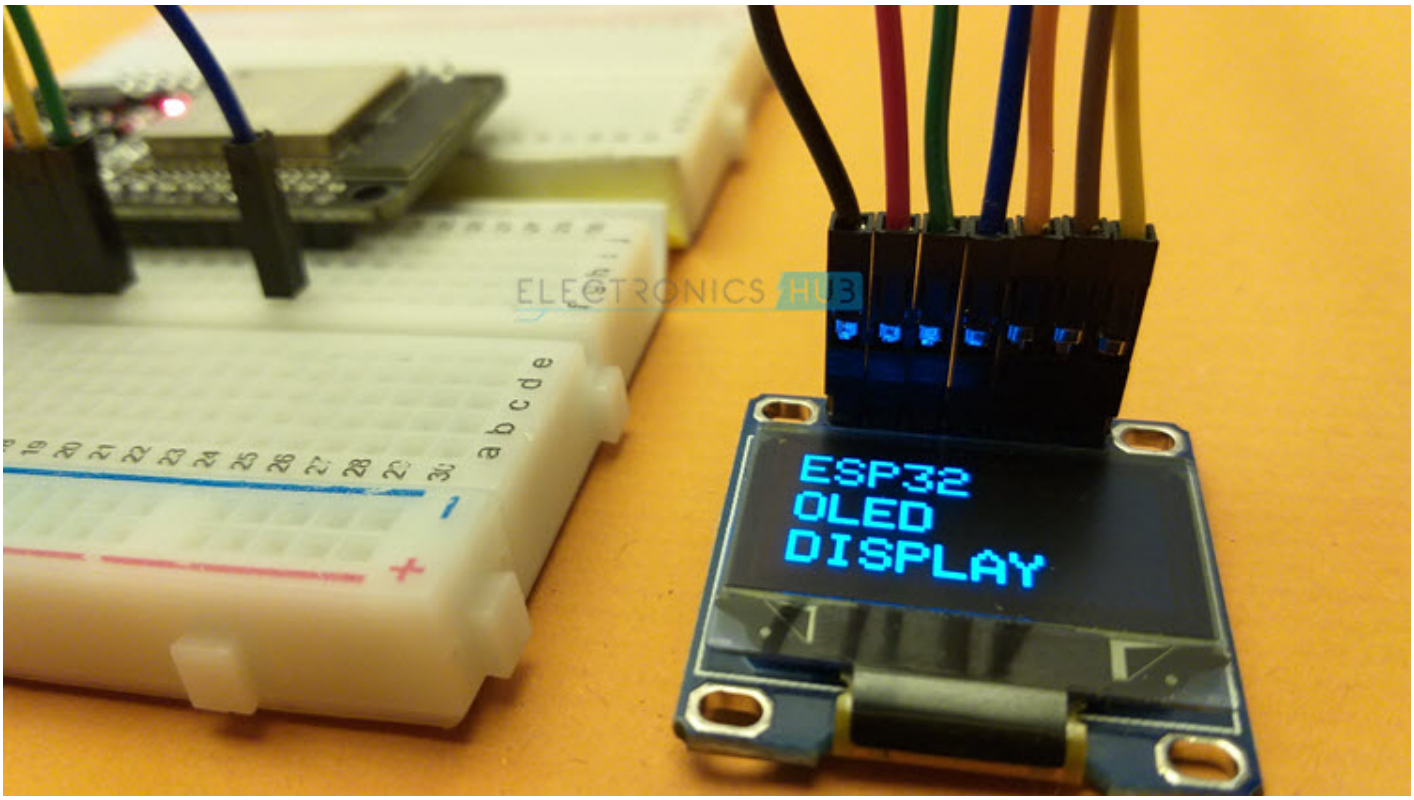
Если вы видите что-то необычное, просто сообщите мне.

Как взаимодествовать с OLED дисплеем через ESP32?

В этой инструкции, мы изучим как взаимодействовать с OLED дисплеем через ESP32 плату разработки. Графический OLED дисплей используемый в проекте основан на SSD1306 OLED драйвере IC и взаимодействует через SPI. Его можно испопльзовать для отображения текста, картинки графики и так далее.



Инструкция

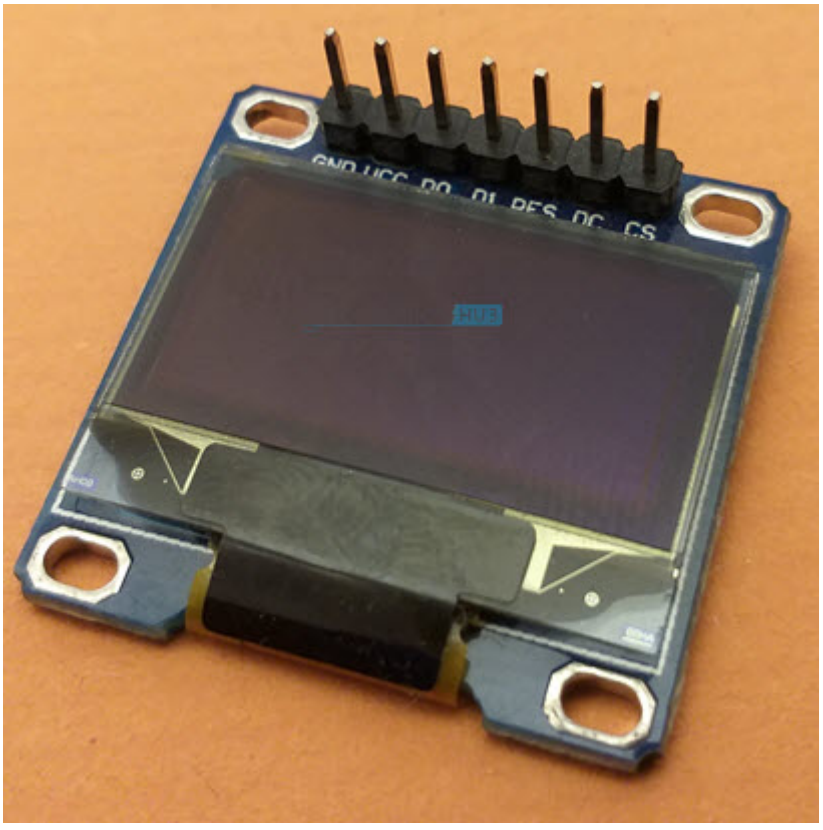
OLED или органический светодиод это улучшенная технология, которая исользует пленку с органическим основанием между двумя электродами(анодом и катодом) и когда подается напряжение между электродами, органическая пленка испускает свет.

Главное преимущество OLED дисплея, в том, что она испускает свой собственный свет, и не требует другого источника подсветки. По этой причине OLED дисплеи часто имеют лучший контраст, яркость и углы обзора при сравнении с LCD дисплеями.

Другое важное свойство OLED дисплея это уровень черного цвета. Так как каждый пиксель испускает свой свет в OLED дисплее, чтобы получить черный свет, достаточно отключить пиксель.

Краткое описание SSD1306 OLED дисплея

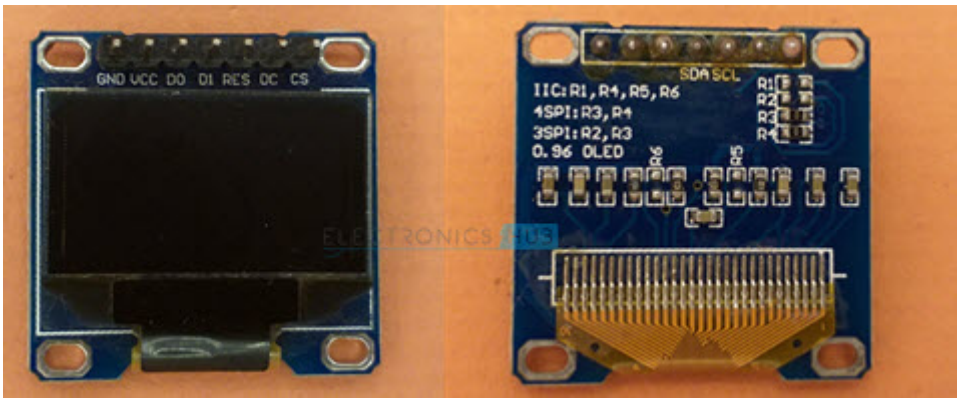
Хотите использовать OLED дисплелй в ваших DYI проектах? Хотите отображать важную информацию, вроде IP адреса, Адреса Web сервера? Тогда модуль SSD1306 отличный выбор!



Этот модуль состоит из монохромного OLED дисплея с разрешением 128 на 64 пикселя. Диагональ такого дисплея 0.96. Этот дисплей использует SSD1306 OLED драйвер.

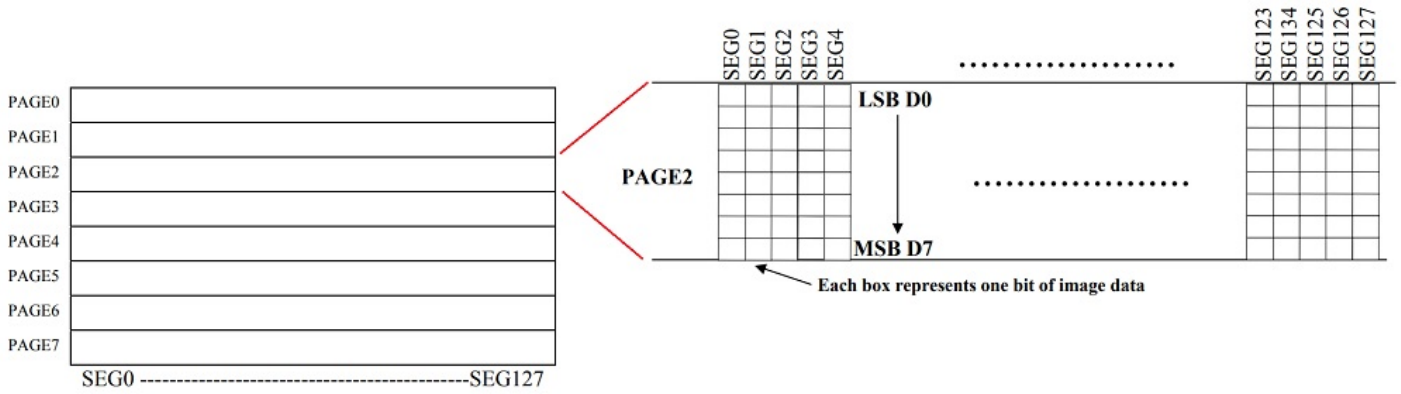
SSD1306 OLED Displays have three types of communication interfaces:

- -bit 6800 Parallel Interface
- 3 or 4 wire SPI
- I2C Of these, the I2C and SPI type OLEDs are very common. It is possible of change the configuration from SPI to I2C and vice-versa (you have solder / de-solder some SMD resistors). The model that I have is using 4-wire SPI Communication.



The SSD1306 OLED Driver IC has 128 x 64 bits Graphic Display Data RAM (GDDRAM). It is divided into eight pages (PAGE 0 to PAGE 7) and each page has 128 Segments. Again, each segment consists of 8-bits and each bit represents one pixel of the display.

So, 8 Pages * 128 Segments * 8 Bits = 8192 Bits (1024 Bytes).

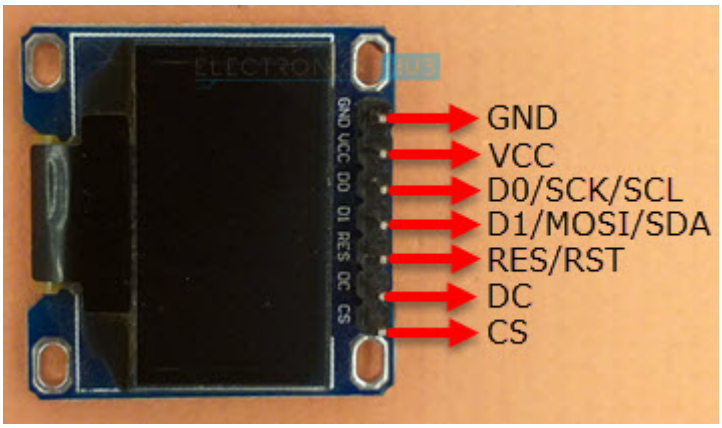


Pinout of OLED Display Module

The following table shows the Pinout of 7-pin SPI based OLED Display Module.

Pin (Alternative Names)	Description
GND	Ground
VCC	Power Supply
D0 (SCK, SCL, CLK)	Clock
D1 (MOSI, SDA)	Data
RES (RST)	Reset

Pin (Alternative Names)	Description
DC (A0)	Data / Command Selection
CS	Chip Select



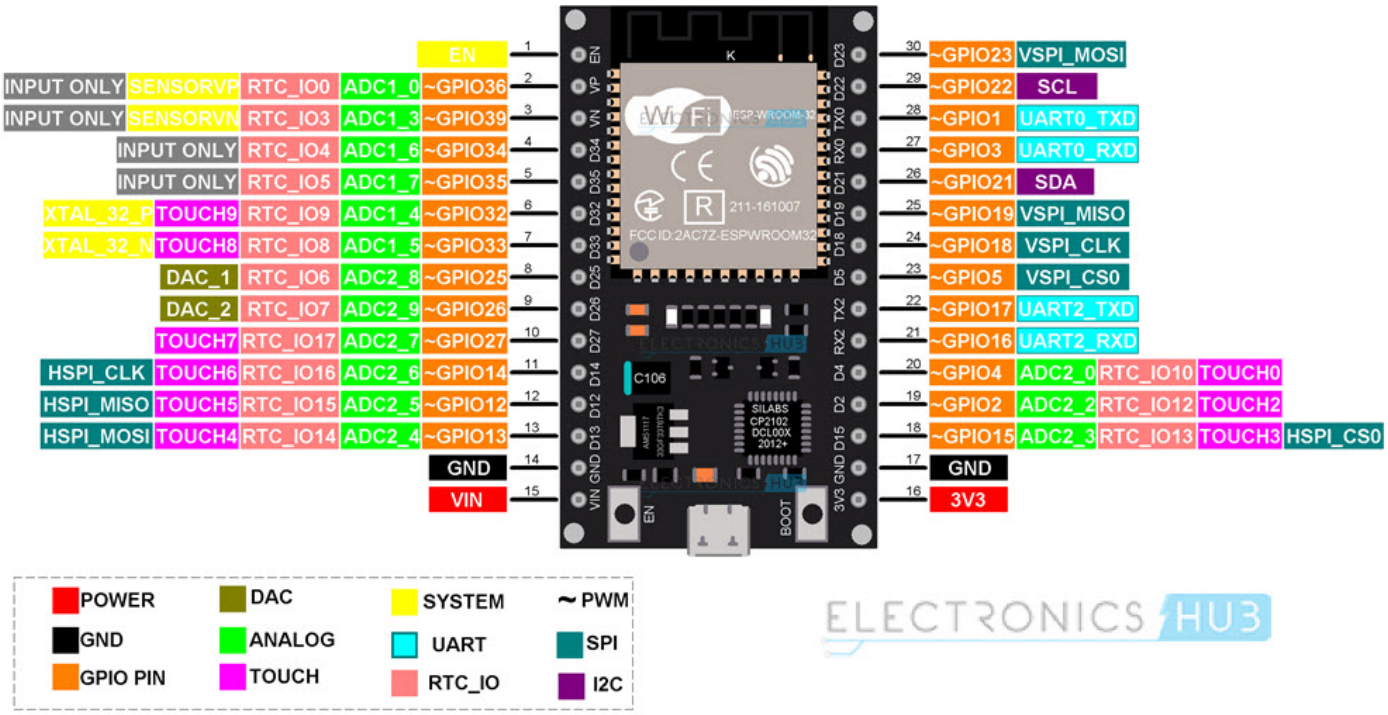
Power Supply

The SSD1306 OLED Driver IC runs on VDD = 1.65V to 3.3V and the actual OLED Panel runs on VCC = 7V to 15V. The OLED Display Module takes care of these wide ranges of voltage requirements with a charge pump circuit (for Panel) and regulator (for Driver IC) from a single power supply (usually between 3V and 5V).

This makes the OLED Display Module to be connected to different boards like Arduino (with 5V logic) and ESP32 (with 3.3V logic).

ESP32 OLED Display Interface

Let us now see how to interface an OLED Display with ESP32. First thing to understand is that the communication interface is SPI. So, look at the Pinout of ESP32 and identify the SPI Pins.



From the above image, HSPI and VSPI are available on ESP32 Development Board for SPI Interface. Let us use the VSPI peripheral. The pins for VSPI in ESP32 are:

VSPI Pin	GPIO Pin
VSPI_MOSI	GPIO 23
VSPI_MISO	GPIO 19

VSPIPin	GPIO Pin
VSPI_CLK	GPIO 18
VSPI_CS	GPIO 5

> NOTE: ESP32 has totally 4 SPI Peripherals. (SPI0, SPI1, HSPI and VSPI). SPI0 is dedicated to SPI Flash IC. SPI1 shares the hardware with SPI0. This leaves HSPI and VSPI for interfacing SPI Devices.

The following table shows the connections between ESP32 and OLED Display Module. In total, we have to make seven connections as this is an SPI OLED Display.

OLED Display	ESP32
GND	GND
VCC	3.3V
D0 (SCK)	GPIO 18
D1 (MOSI)	GPIO 23
RES	GPIO 17
DC	GPIO 16

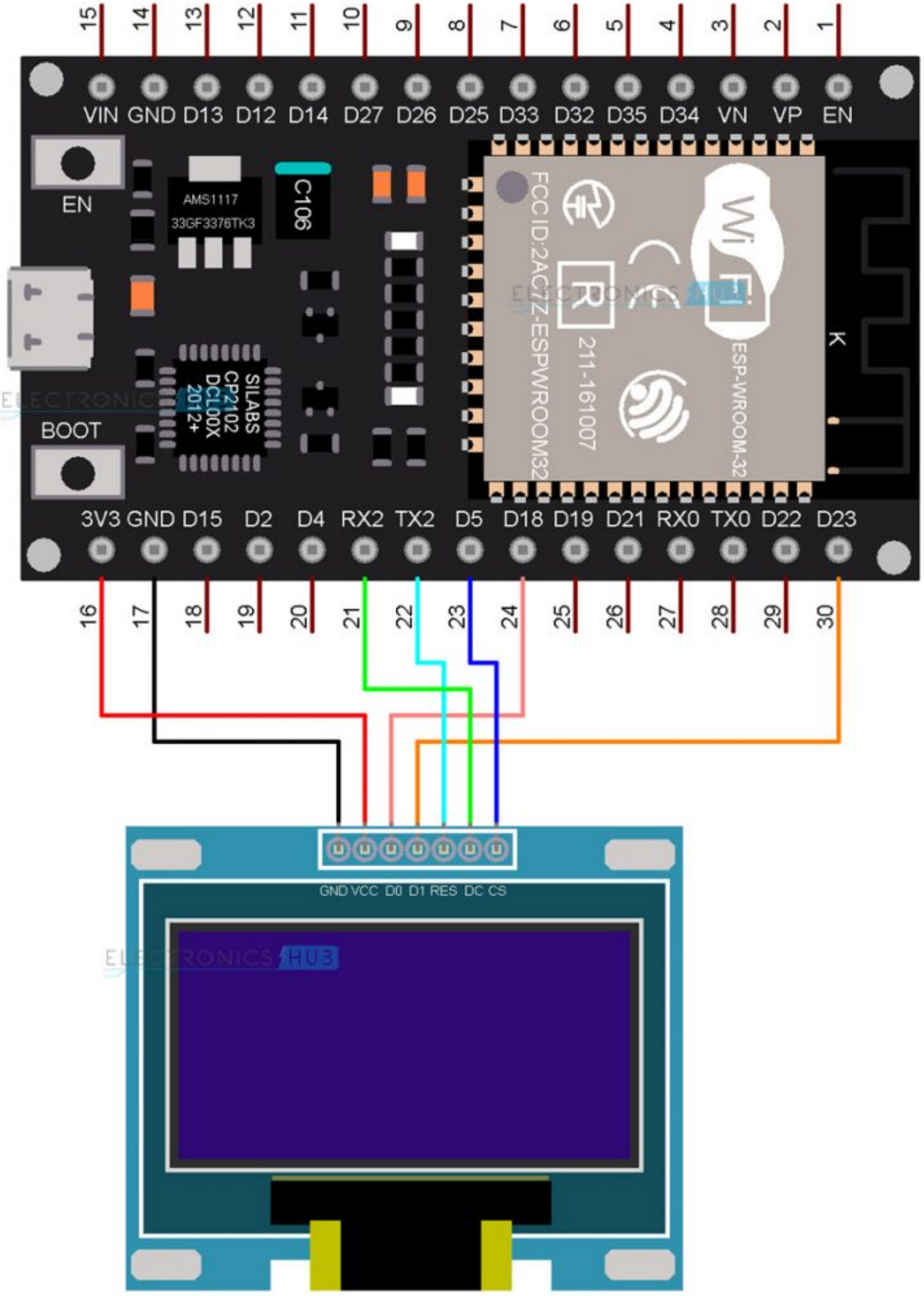
OLED Display	ESP32
CS	GPIO 5

Components Required

- ESP32 DevKit Development Board
- OLED Display Module
- Breadboard
- Connecting Wires
- Micro USB Cable

Circuit Diagram

The following image shows the circuit diagram for Interfacing SPI OLED Display with ESP32.


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
AMS1117
33GF3376TK3
C106
BOOT
SILABS
CP2102
DCL00X
2012+
FCC ID:2AC7Z-ESPWROOM32
211-161007
ESP-WROOM-32
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
GND VCC D0 D1 RES DC CS

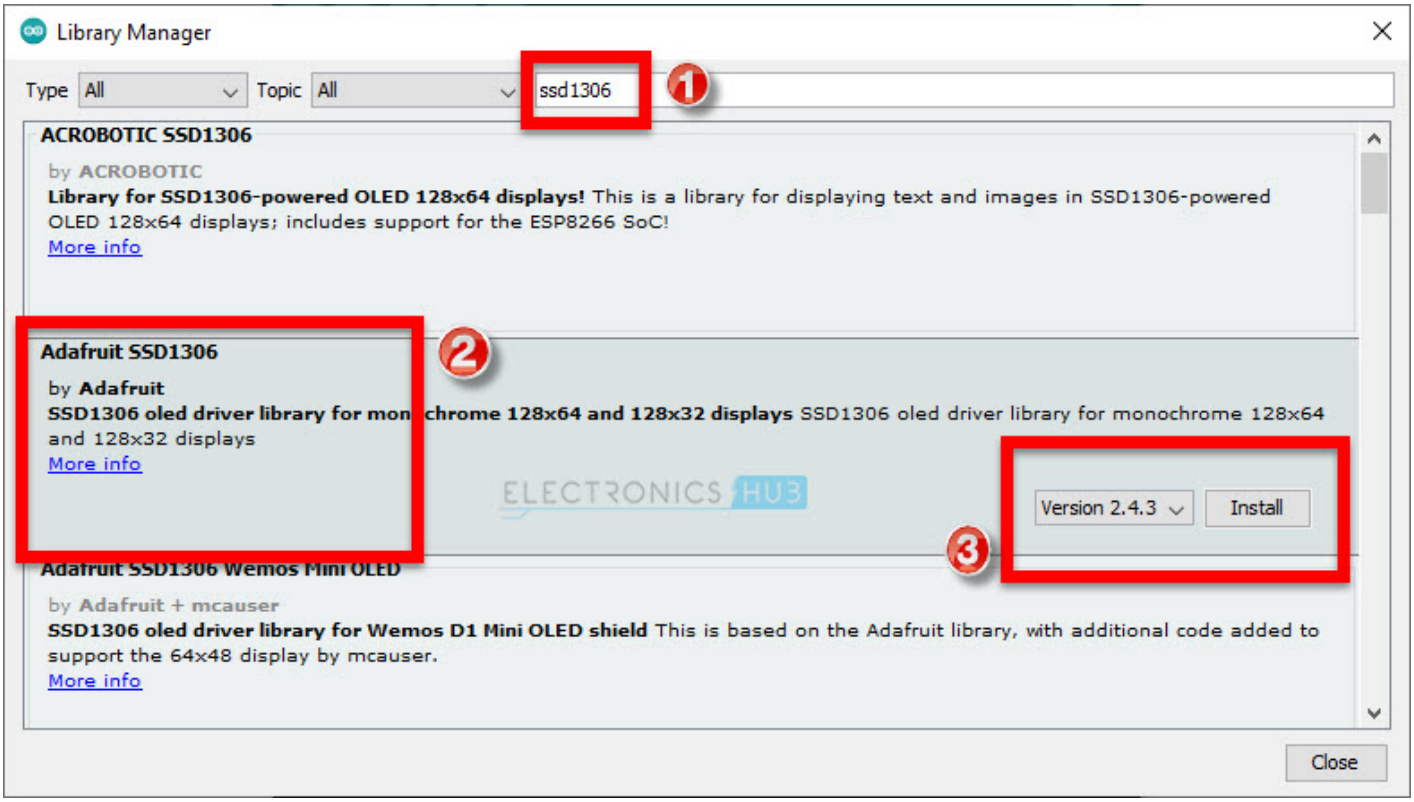
Preparing Arduino IDE

Before writing the code, you need to download some libraries for Arduino IDE related to SSD1306 OLED Display.

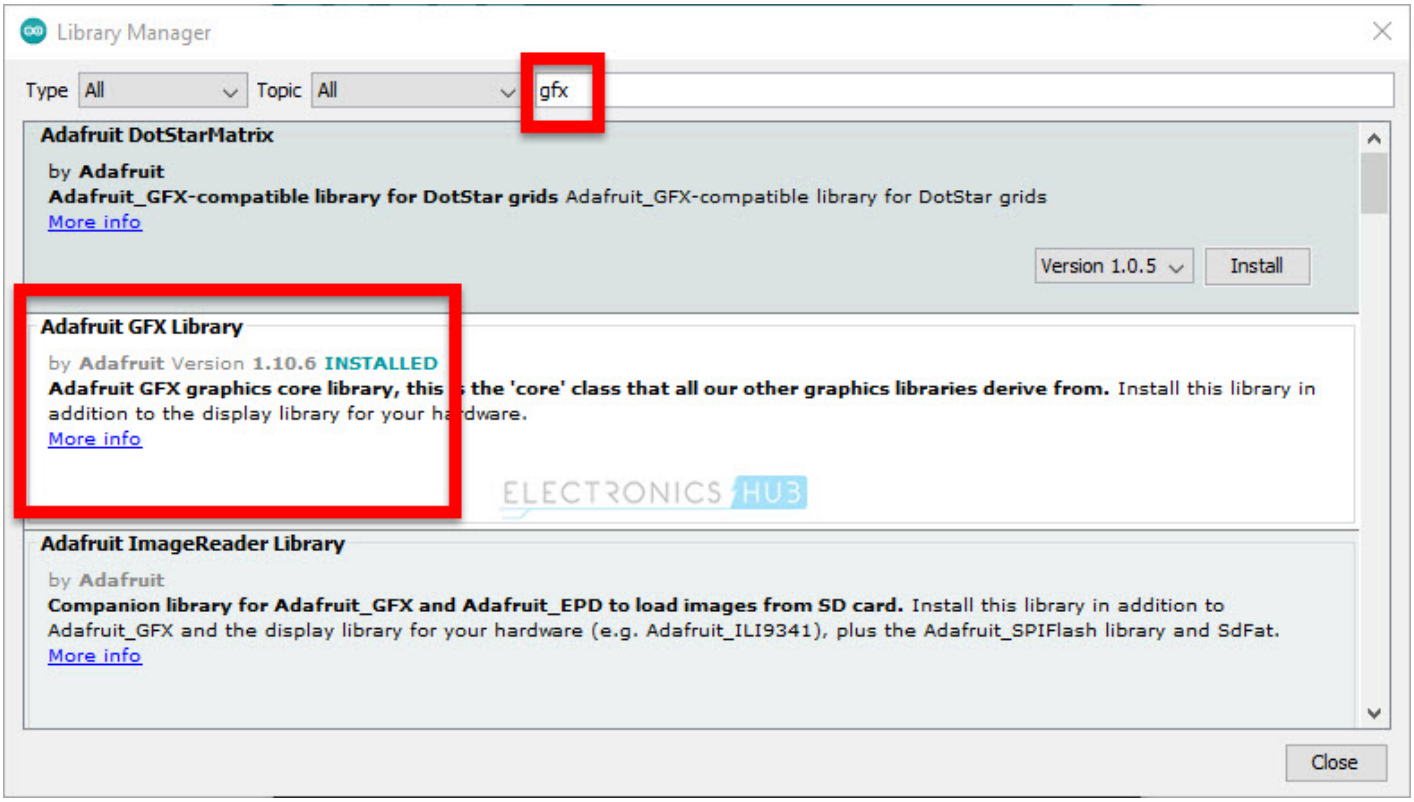
I made a dedicated tutorial on how to install ESP32 Board in Arduino IDE. You can check out that tutorial first. Now, open the Arduino IDE and go to Tools -> Manage Libraries. . .

Arduino-IDE-OLED-Library-1

A Library Manager window will pop-up. In the search bar, type “ssd1306” and from the results select the “Adafruit SSD1306” option and click on install. This library is written specifically for monochrome OLED Displays based on SSD1306 Driver IC. The supported resolutions are 128 x 32 and 128 x 64.



After installing SSD1306 Library, search for “gfx” and install “Adafruit GFX Library”. This is a graphics library by Adafruit for displaying basic graphics like lines, circles, rectangles etc.



Close the library manager window after downloading all the necessary libraries. Now, make sure that ESP32 Board is selected in Arduino IDE (Tools -> Board -> ESP32 Arduino -> ESP32 Dev Module).

Testing ESP32 OLED Display

After making all the necessary connections, we will now proceed to write a test code for ESP32 to display some text and graphics on the OLED Display. In this code, I am testing various features of the OLED Display like displaying normal text, inverted text, scrolling text, displaying ASCII Characters, setting font size.

I also added the code for displaying graphics like rectangle, filled rectangle, rounded rectangle, filled rounded rectangle, circle, filled circle, triangle and filled triangle.

Finally, I took the “Electronics Hub” logo and converted it into a bitmap and displayed it on the OLED Display.

```
#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   23
#define OLED_CLK    18
#define OLED_DC     16
#define OLED_CS     5
#define OLED_RESET  17

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(1000);

  display.clearDisplay();
  display.drawBitmap(0, 0, electronicshub_logo, SCREEN_WIDTH, SCREEN_HEIGHT, SSD1306_WHITE);
  display.display();
  delay(1000);



}
void loop()
{
  AllPixels();
  TextDisplay();
  InvertedTextDisplay();
  ScrollText();
  DisplayChars();
  TextSize();
  DrawRectangle();
  DrawFilledRectangle();
  DrawRoundRectangle();
  DrawFilledRoundRectangle();
  DrawCircle();
  DrawFilledCircle();
  DrawTriangle();
  DrawFilledTriangle();
}
```

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

    }
    display.display();
    delay(30);
  }

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

    }
    display.display();
    delay(30);
  }

}

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(4000);
}
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);
}

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(2000);
}

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);
}
```

Conclusion

A simple tutorial on how to interface SPI OLED Display Module with ESP32 DevKit Board. You learned the pinout of SSD1306 OLED Display, necessary connections for SPI Interface with ESP32, download libraries for Arduino IDE and display some text, graphics and image on the OLED Display using ESP32.

Revision #3
Created 7 April 2023 18:35:45 by gasick
Updated 1 September 2023 06:49:40 by gasick