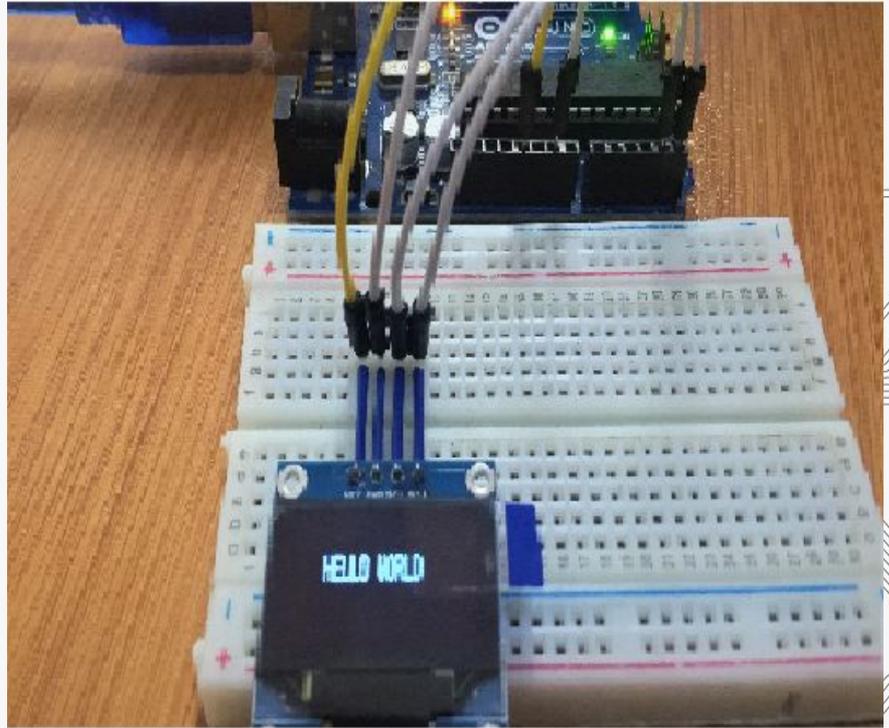
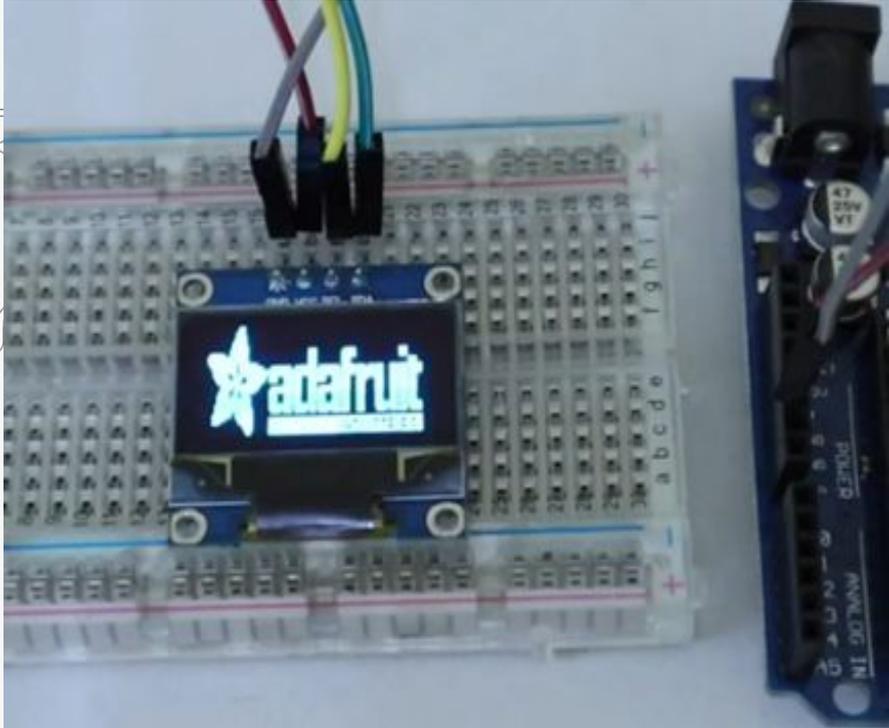


Interfacing of OLED



OLED

- OLED (Organic Light-Emitting Diode) is a self light-emitting technology composed of a thin, multi-layered organic film placed between an anode and cathode. In contrast to LCD technology, OLED does not require a backlight.
- OLED possesses high application potential for virtually all types of displays and is regarded as the ultimate technology for the next generation of flat-panel displays.

Working of OLED

- The main component in an **OLED display** is the **OLED emitter** - an organic (carbon-based) material that emits light when electricity is applied. The basic structure of an **OLED** is an emissive layer sandwiched between a cathode (which injects electrons) and an anode (which removes electrons).

Pin Diagram of OLED



Pin 1: GND

Pin 2: 3.3V to 5V

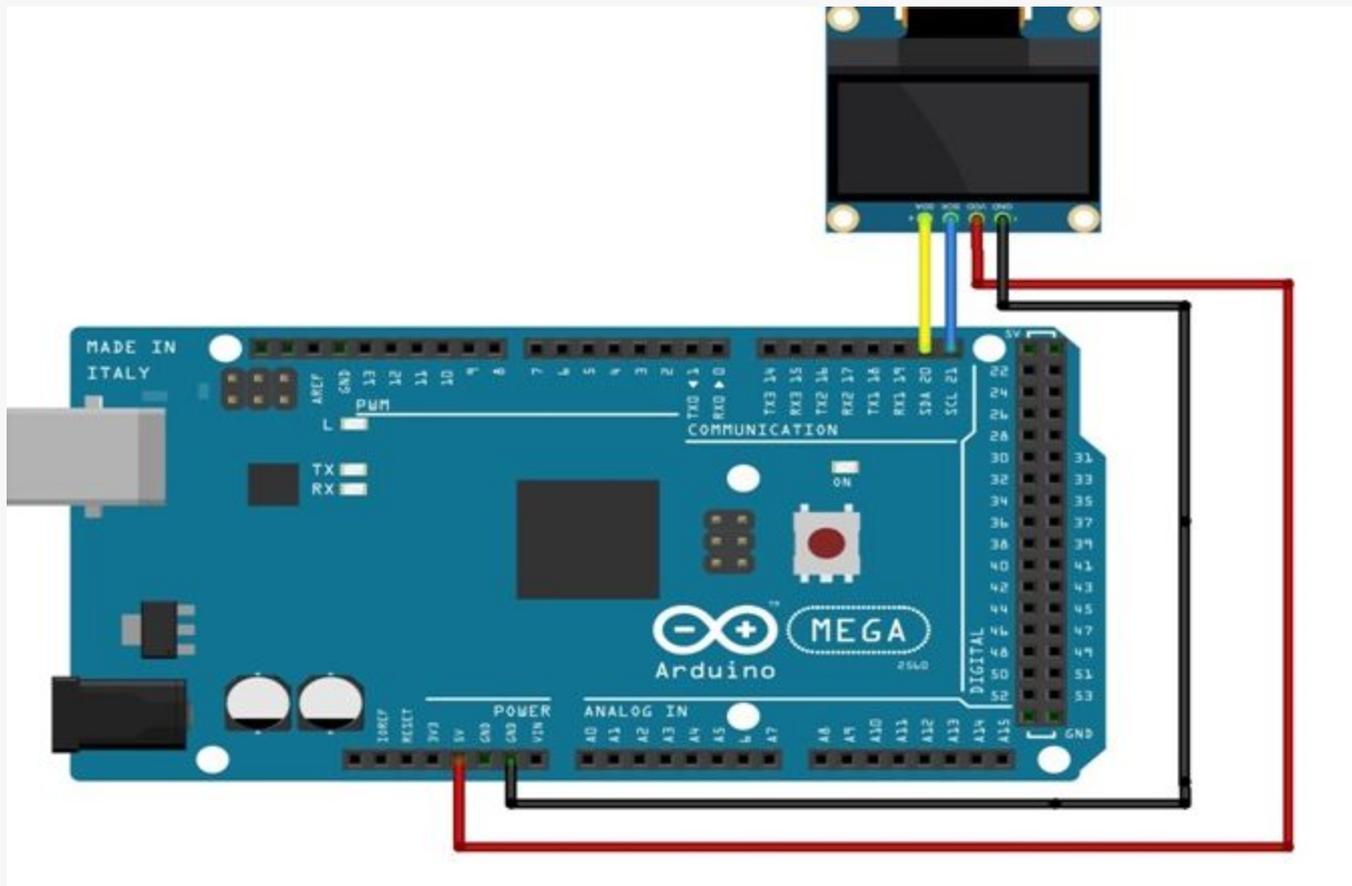
Pin 3: SCL - Serial Clock

Pin 4: SDA - Serial Data

Components Required

- Arduino UNO Board
- 0.96" I2C OLED Display
- Breadboard
- Connecting Wires

Connection Diagram



Connections

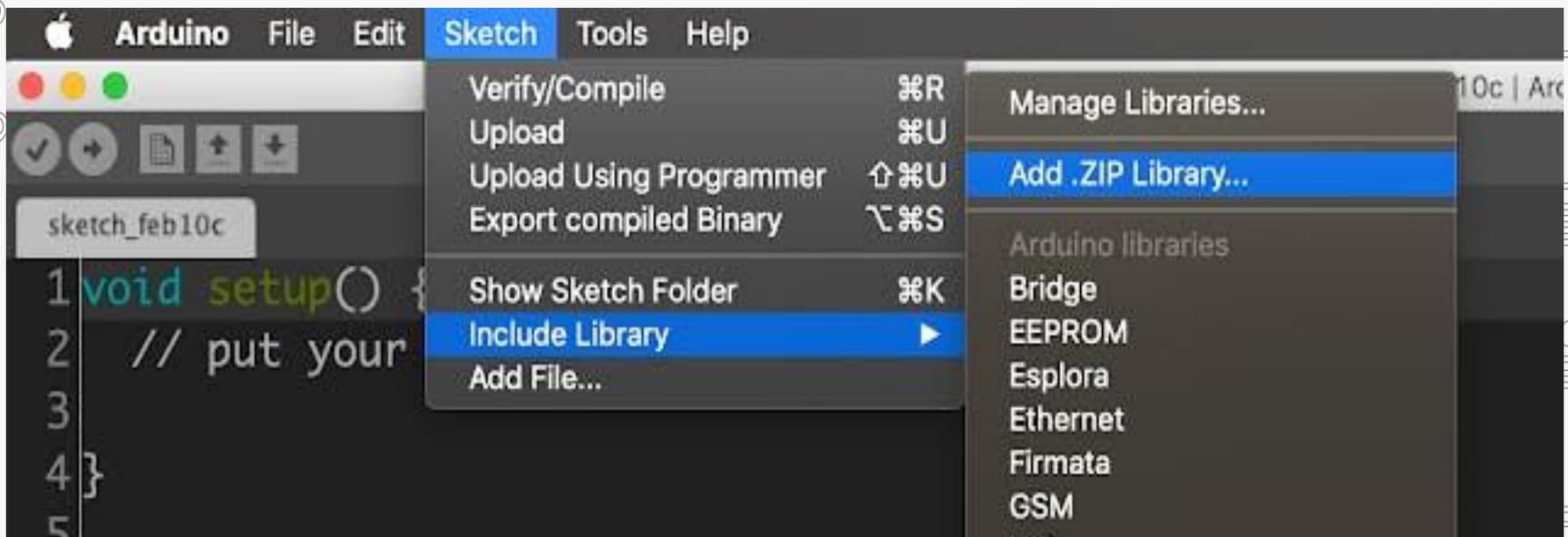
Connect the pins of OLED with the following pins of Arduino Mega , as shown below.

- 5V-----VCC
- GND-----GND
- SDA of arduino-----SDA
- SCL of arduino-----SCL

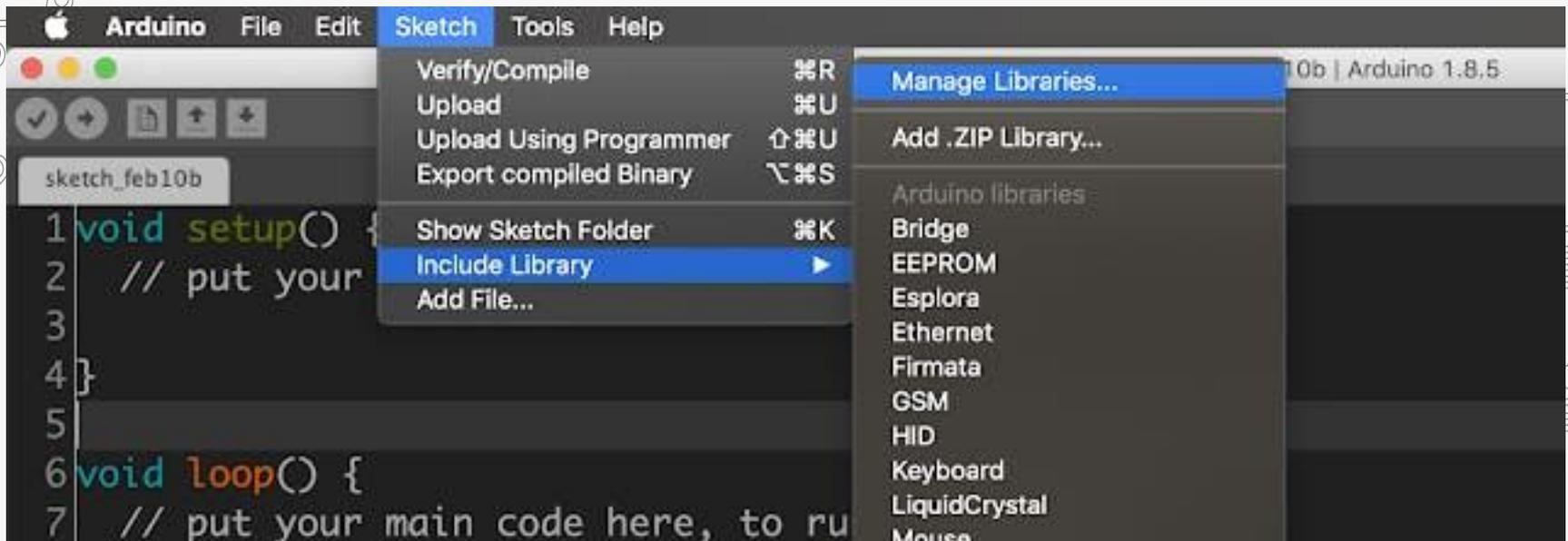
Install Required Libraries

- Adafruit_GFX.h
<https://github.com/adafruit/Adafruit-GFX-Library>
- Adafruit_SSD1306.h https://www.github.com/adafruit/Adafruit_SSD1306
- Wire.h this library will be installed by default.

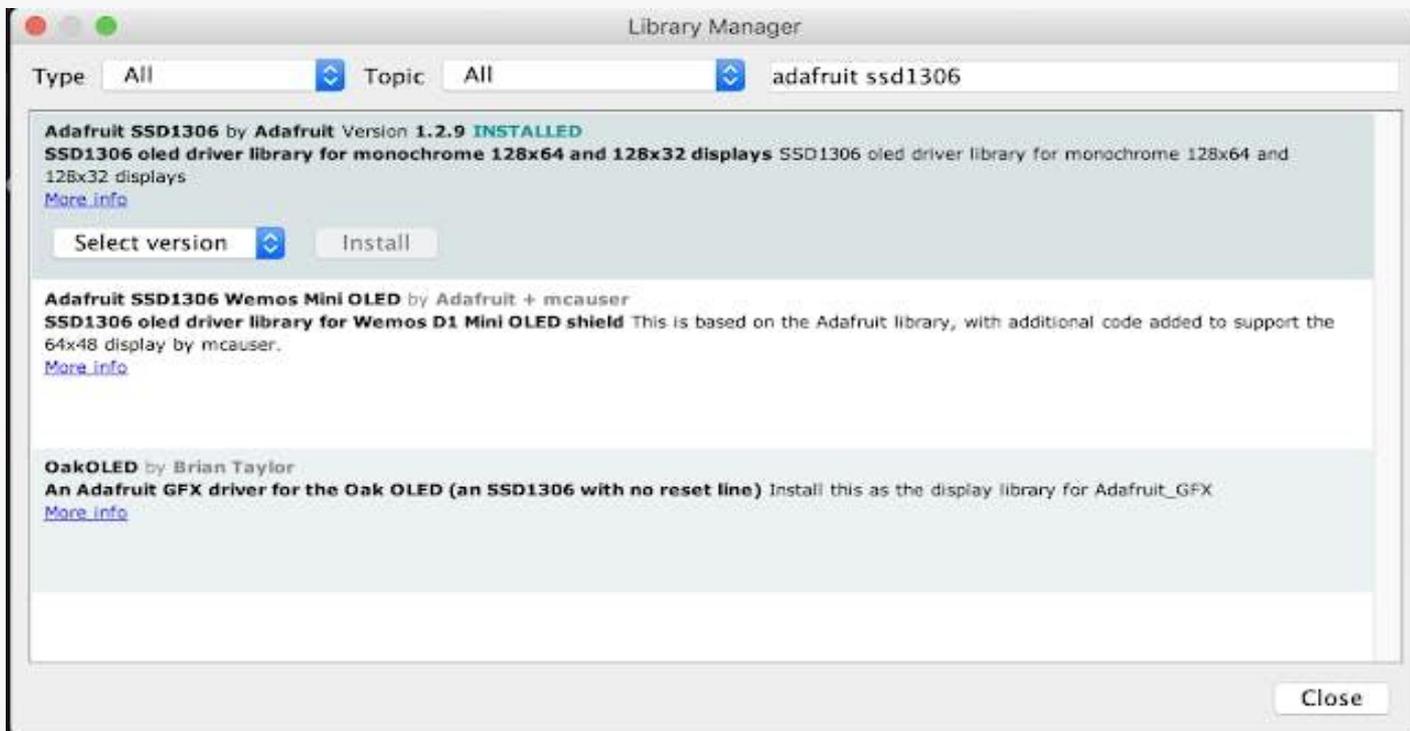
- Either you can download the libraries from github and add the ZIP file in Arduino IDE using add.zip library option.



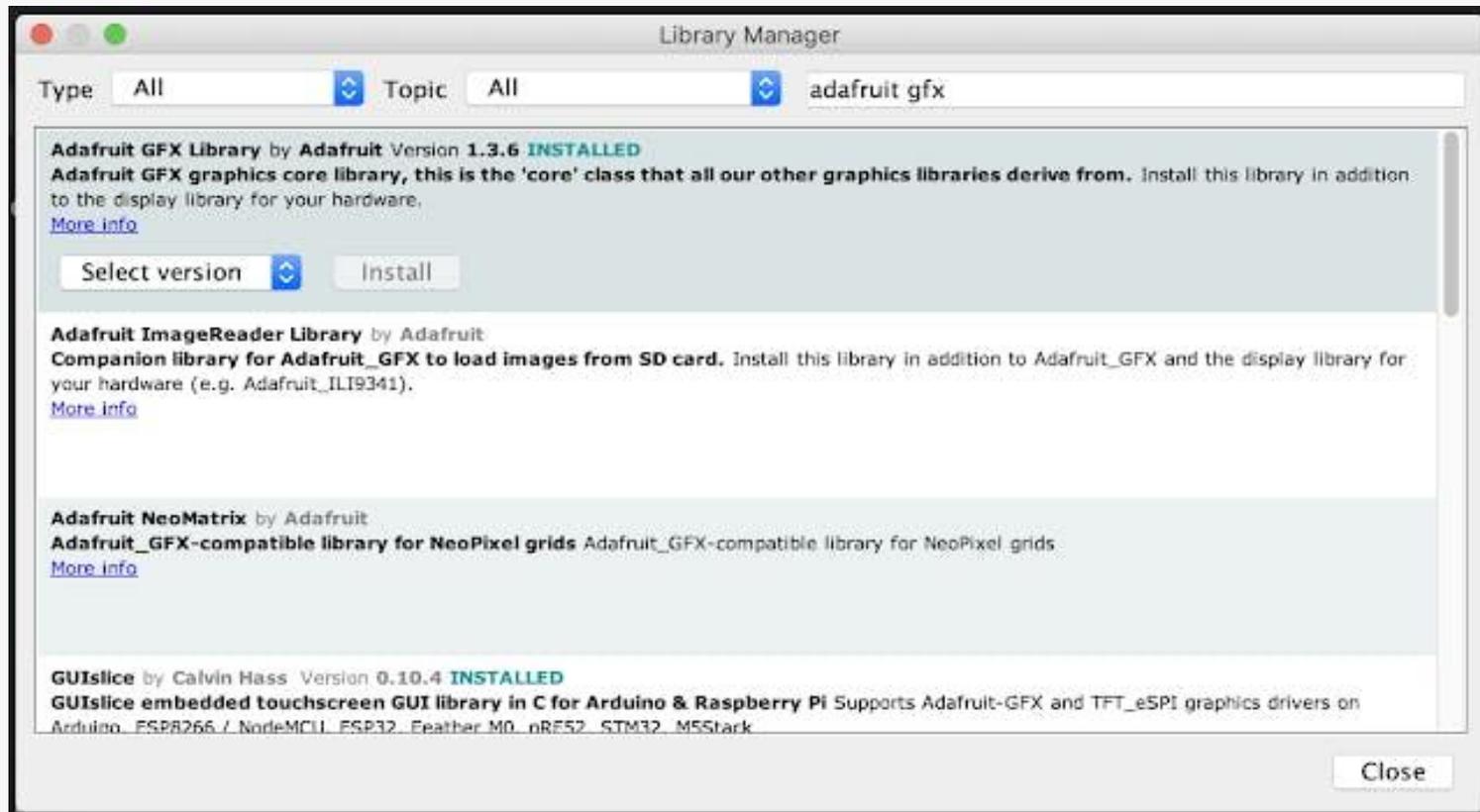
- Or you can install the libraries directly from Arduino IDE from Manage Libraries.
- Follow the below steps to install.



- Open manage library from sketch -> include library -> manage libraries.
- Then search for AdafruitSSD1306 and click on the install button.



- Then search for the Adafruit GFX and click the install button. Now we have successfully installed the required libraries for this project.



Adafruit Example Code

- Let's upload the adafruitSSD1306 library example program and test our setup. Open the example program from File -> Examples -> Adafruit SSD1306 -> ssd1306_128x32_i2c, because We are using OLED display with I2C interface with the screen size of 128x64.



Code for interfacing

ssd1306_128x32_i2c | Arduino 1.8.19

File Edit Sketch Tools Help



ssd1306_128x32_i2c

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

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET      4 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

#define NUMFLAKES      10 // Number of snowflakes in the animation example

#define LOGO_HEIGHT    16
#define LOGO_WIDTH     16
static const unsigned char PROGMEM logo_bmp[] =
{ B00000000, B11000000,
  B00000001, B11000000,
  B00000001, B11000000,
  B00000011, B11100000,
  B11110011, B11100000,
  B11111110, B11111000,
  B01111110, B11111111,
```



ssd1306_128x32_i2c | Arduino 1.8.19

File Edit Sketch Tools Help



ssd1306_128x32_i2c

```
static const unsigned char PROGMEM logo_bmp[] =
{ B00000000, B11000000,
  B00000001, B11000000,
  B00000001, B11000000,
  B00000011, B11100000,
  B11110011, B11100000,
  B11111110, B11111000,
  B01111110, B11111111,
  B00110011, B10011111,
  B00011111, B11111100,
  B00001101, B01110000,
  B00011011, B10100000,
  B00111111, B11100000,
  B00111111, B11110000,
  B01111100, B11110000,
  B01110000, B01110000,
  B00000000, B00110000 };

void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x32
```



ssd1306_128x32_i2c | Arduino 1.8.19

File Edit Sketch Tools Help

ssd1306_128x32_i2c

```
void setup() {
  Serial.begin(9600);

  // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x32
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }

  // Show initial display buffer contents on the screen --
  // the library initializes this with an Adafruit splash screen.
  display.display();
  delay(2000); // Pause for 2 seconds

  // Clear the buffer
  display.clearDisplay();

  // Draw a single pixel in white
  display.drawPixel(10, 10, SSD1306_WHITE);

  // Show the display buffer on the screen. You MUST call display() after
  // drawing commands to make them visible on screen!
  display.display();
}
```



ssd1306_128x32_i2c | Arduino 1.8.19

File Edit Sketch Tools Help



ssd1306_128x32_i2c

```
// Show initial display buffer contents on the screen --
// the library initializes this with an Adafruit splash screen.
display.display();
delay(2000); // Pause for 2 seconds

// Clear the buffer
display.clearDisplay();

// Draw a single pixel in white
display.drawPixel(10, 10, SSD1306_WHITE);

// Show the display buffer on the screen. You MUST call display() after
// drawing commands to make them visible on screen!
display.display();
delay(2000);
// display.display() is NOT necessary after every single drawing command,
// unless that's what you want...rather, you can batch up a bunch of
// drawing operations and then update the screen all at once by calling
// display.display(). These examples demonstrate both approaches...

/*testdrawline(); // Draw many lines

testdrawrect(); // Draw rectangles (outlines)
```



ssd1306_128x32_i2c | Arduino 1.8.19

File Edit Sketch Tools Help



ssd1306_128x32_i2c

```
testdrawstyles(); // Draw 'stylized' characters*/
testscrolltext(); // Draw scrolling text
testdrawbitmap(); // Draw a small bitmap image

// Invert and restore display, pausing in-between
display.invertDisplay(true);
delay(1000);
display.invertDisplay(false);
delay(1000);

testanimate(logo_bmp, LOGO_WIDTH, LOGO_HEIGHT); // Animate bitmaps
}

void loop() {
}

/*void testdrawline() {
  int16_t i;

  display.clearDisplay(); // Clear display buffer
```



```
void testscrolltext(void) {  
  display.clearDisplay();  
  
  display.setTextSize(2); // Draw 2X-scale text  
  display.setTextColor(SSD1306_WHITE);  
  display.setCursor(10, 0);  
  display.println(F("ABL"));  
  display.display(); // Show initial text  
  delay(100);  
  
  // Scroll in various directions, pausing in-between:  
  display.startscrollright(0x00, 0x0F);  
  delay(2000);  
  display.stopscroll();  
  delay(1000);  
  display.startscrollleft(0x00, 0x0F);  
  delay(2000);  
  display.stopscroll();  
  delay(1000);  
  display.startscrolldiagright(0x00, 0x07);  
  delay(2000);  
  display.startscrolldiaqlleft(0x00, 0x07);  
}
```



ssd1306_128x32_i2c | Arduino 1.8.19

File Edit Sketch Tools Help



ssd1306_128x32_i2c

```
void testscrolltext(void) {
  display.clearDisplay();

  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(10, 0);
  display.println(F("ABL"));
  display.display();      // Show initial text
  delay(100);

  // Scroll in various directions, pausing in-between:
  display.startscrollright(0x00, 0x0F);
  delay(2000);
  display.stopscroll();
  delay(1000);
  display.startscrollleft(0x00, 0x0F);
  delay(2000);
  display.stopscroll();
  delay(1000);
  display.startscrolldiagright(0x00, 0x07);
  delay(2000);
  display.startscrolldiaqlleft(0x00, 0x07);
```



ssid1306_128x32_i2c | Arduino 1.8.19

File Edit Sketch Tools Help



ssid1306_128x32_i2c

```
void testanimate(const uint8_t *bitmap, uint8_t w, uint8_t h) {
  int8_t f, icons[NUMFLAKES][3];

  // Initialize 'snowflake' positions
  for(f=0; f< NUMFLAKES; f++) {
    icons[f][XPOS]   = random(1 - LOGO_WIDTH, display.width());
    icons[f][YPOS]   = -LOGO_HEIGHT;
    icons[f][DELTAY] = random(1, 6);
    Serial.print(F("x: "));
    Serial.print(icons[f][XPOS], DEC);
    Serial.print(F(" y: "));
    Serial.print(icons[f][YPOS], DEC);
    Serial.print(F(" dy: "));
    Serial.println(icons[f][DELTAY], DEC);
  }

  for(;;) { // Loop forever...
    display.clearDisplay(); // Clear the display buffer

    // Draw each snowflake:
    for(f=0; f< NUMFLAKES; f++) {
      display.drawBitmap(icons[f][XPOS], icons[f][YPOS], bitmap, w, h, SSD1306 WHITE);
    }
  }
}
```



```
display.clearDisplay(); // Clear the display buffer

// Draw each snowflake:
for(f=0; f< NUMFLAKES; f++) {
    display.drawBitmap(icons[f][XPOS], icons[f][YPOS], bitmap, w, h, SSD1306_WHITE);
}

display.display(); // Show the display buffer on the screen
delay(200);        // Pause for 1/10 second

// Then update coordinates of each flake...
for(f=0; f< NUMFLAKES; f++) {
    icons[f][YPOS] += icons[f][DELTAY];
    // If snowflake is off the bottom of the screen...
    if (icons[f][YPOS] >= display.height()) {
        // Reinitialize to a random position, just off the top
        icons[f][XPOS] = random(1 - LOGO_WIDTH, display.width());
        icons[f][YPOS] = -LOGO_HEIGHT;
        icons[f][DELTAY] = random(1, 6);
    }
}
}
}
```



```
display.clearDisplay(); // Clear the display buffer
```

```
// Draw each snowflake:
```

```
for(f=0; f< NUMFLAKES; f++) {
```

```
    display.drawBitmap(icons[f][XPOS], icons[f][YPOS], bitmap, w, h, SSD1306_WHITE);
```

```
}
```

```
display.display(); // Show the display buffer on the screen
```

```
delay(200); // Pause for 1/10 second
```

```
// Then update coordinates of each flake...
```

```
for(f=0; f< NUMFLAKES; f++) {
```

```
    icons[f][YPOS] += icons[f][DELTAY];
```

```
    // If snowflake is off the bottom of the screen...
```

```
    if (icons[f][YPOS] >= display.height()) {
```

```
        // Reinitialize to a random position, just off the top
```

```
        icons[f][XPOS] = random(1 - LOGO_WIDTH, display.width());
```

```
        icons[f][YPOS] = -LOGO_HEIGHT;
```

```
        icons[f][DELTAY] = random(1, 6);
```

```
    }
```

```
}
```

```
}
```

```
}
```




Displaying_Image | Arduino 1.8.19

File Edit Sketch Tools Help



Displaying_Image

```
0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0x80, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x03, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x07, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xe0, 0x0f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0x1f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
};
void setup(){
  oled.begin(SSD1306_SWITCHCAPVCC, 0x3C); //or 0x3C
  oled.clearDisplay(); //for Clearing the display
  oled.drawBitmap(0, 0, myBitmap, 128, 64, WHITE); // display.drawBitmap(x position, y position, bitmap data, bitmap width, bitmap height, color)
  oled.display();
}
void loop() { }
```

Project Link : <https://youtu.be/7xWjEkj3oxs>