Interfacing of OLED

ТΜ

Δ

ACTIVITY BASED LEARNING





- 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.

www.able@cation.com

Working of OLED

ASED LEARNING

www.able@cation.com

 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



www.able@cation.com

BASED LEARNING

Pin 1: GND Pin 2: 3.3V to 5V Pin 3: SCL - Serial Clock Pin 4: SDA - Serial Data

 \bigcirc

Components Required

Arduino UNO Board

TY BASED LEARNING

• 0.96" I2C OLED Display

TM

• Breadboard

www.able@cation.com

• Connecting Wires





Connection Diagram

ТΜ

Δ

ACTIVITY BASED LEARNING





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

www.ablkart.com

- 5v-----VCC
- GND-----GND

www.able@cation.com

- SDA of arduino-----SDA
- SCL of arduino-----SCL

Install Required Libraries

Adafruit_GFX.h

www.able@cation.com

SED LEARNING

<u>https://github.com/adafruit/Adafruit-GFX-Library</u>

- Adafruit_SSD1306.h <u>https://www.github.com/adafruit/Adafruit_SSD1306</u>
- 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.

ACTIVITY BASED LEARNING





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

| | Verify/Compile | %R ¥U | Manage Libraries | 10b Arduino 1.8.5 |
|---|-------------------------|-------------|--|---------------------|
| CORER | Upload Using Programmer | ៤ អប | Add .ZIP Library | |
| <pre>sketch_feb10b 1 void setup() { 2 // put your</pre> | Show Sketch Folder | €#S %K | Arduino libraries Bridge EEPROM | |
| 3 4} 5 | Add File | | Esplora Ethernet Firmata GSM HID | |
| 6 <pre>void loop() { 7 // put your</pre> | main code here, to | o ru | Keyboard LiquidCrystal Mouse | |
| <u>j</u> | 0 | | | |



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

6

WWW

| SSD1306 oled driver 128x32 displays | r library for monochrome 128x6 | 54 and 128x32 displays SSD1306 oled driver libri | ary for monochrome 128x64 and |
|--|--|--|--------------------------------------|
| More_info | <u>11</u> | | |
| Select version | Install | | |
| Adafruit SSD1306 W SSD1306 oled driver 64x48 display by mca More_info | Vemos Mini OLED by Adafruit + ı r library for Wemos D1 Mini OLE iuser. | mcauser D shield This is based on the Adafruit library, with | additional code added to support the |
| | | | |
| OakOLED by Brian 1 An Adafruit GFX driv More Info | Taylor ver for the Oak OLED (an SSD13 | 06 with no reset line) Install this as the display | library for Adafruit_GFX |
| | | | |
| | | | |
| | | | 1 |

• Then search for the Adafruit GEX a

www.abled

TM

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

| | | | | Library Man | ager | | |
|--|---|-------------------------------------|--|---------------------------------|--|--------------------------------|------|
| Type All | | Topic | All | 0 | adafruit gfx | | |
| Adafruit GFX Library Adafruit GFX graphi to the display library t More info | by Adafruit cs core librar for your hardw | Version ry, this is are. | 1.3.6 INSTALLE the 'core' clas | D s that all our oth | er graphics libraries derive from. Ir | stall this library in addition | 0~ |
| Adafruit ImageRead Companion library fr your hardware (e.g. A More Info | Ser Library by or Adafruit_C dafruit_IL1934 | y Adafru GFX to lo 11). | it ad images fron | n SD card. Install | this library in addition to Adafruit_GFX | and the display library for | |
| Adafruit NeoMatrix Adafruit_GFX-comp <u>More info</u> | by Adafruit atible library | for Neo | Pixel grids Adaf | ruit_GFX-compati | ale library for NeoPixel grids | | 1209 |
| GUIslice by Calvin H GUIslice embedded Arduiro, FSP8265 / N | ass Version 0 touchscreen adeMCU_ESP3 |). 10.4 IN GUI libr 12. Feath | STALLED ary in C for Ard r MO_oR552_ST | uino & Raspberr M32. MSStark | PI Supports AdaTruit-GFX and TFT_e5 | SPI graphics drivers on | |
| | | | | (1-1) | | Close | |

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.

| | Examples from Custom Libraries ACROBOTIC SSD1306 Adafruit Circuit Playground Adafruit GFX Library Adafruit SSD1306 Adafruit TouchScreen ArduinoJson Bifrost library for HC-SR04 Encoder ESP8266wifi-master FastLED | ****** | OLED_featherwing ssd1306_128x32_i2c ssd1306_128x32_spi | |
|----------------------|--|------------|--|-----|
| www.ablec@cation.com | | \bigcirc | 0 0 www.ablkart.con | n (|

Code for interfacing

LEARNING

Δ

ACTIVI

0

 \bigcirc

 \bigcirc

BASED

ТΜ

| S ssd1306_128x32_i2c Arduino 1.8.19 | - 0 | \times |
|--|-----|----------|
| File Edit Sketch Tools Help | | |
| | | P |
| ssd1306_128x32_i2c | | |
| #include <wire.b></wire.b> | | |
| <pre>#include <adafruit gfx.b=""></adafruit></pre> | | |
| = #include <adafruit ssd1306.h=""></adafruit> | | - I |
| | | |
| #define SCREEN WIDTH 128 // OLED display width, in pixels | | |
| <pre>#define SCREEN_HEIGHT 32 // OLED display height, in pixels</pre> | | |
| | | |
| // Declaration for an SSD1306 display connected to I2C (SDA, SCL pins) | | |
| <pre>#define OLED_RESET 4 // Reset pin # (or -1 if sharing Arduino reset pin)</pre> | | |
| 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 | | |
| s_{const} unsigned char product \log_{const} \log_{const} | | |
| E00000000, B1000000, | | |
| B00000001, B11000000, | | |
| B00000011 B11100000 | | |
| B11110011 B11100000 | | |
| B1111110, B11111000, | | |
| B01111110, B11111111, | | |
| | | |
| | | |
| | | |
| | | |
| | | |

 \bigcirc



File Edit Sketch Tools Help

0

| FI | le Edit Sketch Tools Help | |
|----|--|-----|
| | | ø |
| ſ | ssd1306_128x32_i2c | |
| 2 | static const unsigned char PROGMEM logo_bmp[] = | |
| { | (B0000000, B11000000, | |
| | B0000001, B11000000, | 1.1 |
| | B0000001, B11000000, | 1 |
| | B0000011, B11100000, | |
| | B11110011, B11100000, | _ |
| | B1111110, B11111000, | |
| | B01111110, B11111111, | |
| | B00110011, B10011111, | 10 |
| | B00011111, B11111100, | 17 |
| 7 | B00001101, B01110000, | |
| | B00011011, B10100000, | |
| | B00111111, B11100000, | |
| | B00111111, B11110000, | |
| | B01111100, B11110000, | |
| | B01110000, B01110000, | |
| | B0000000, B00110000 }; | |
| | | |
| V | roid setup() { | |
| | Serial.begin(9600); | |
| | | |
| | <pre>// SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally</pre> | |
| | | |

 \bigcirc

if(!display.begin(SSD1306 SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x32



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();



D

Ø

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)



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

```
void loop() {
}
```

```
/*void testdrawline() {
    int16 t i;
```

display.clearDisplay(); // Clear display buffer



ø

-

Ssd1306 128x32 i2c | Arduino 1.8.19 File Edit Sketch Tools Help

B + +

$\bigcirc \bigcirc$

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.startscrolldiagleft(0x00, 0x07);



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.startscrolldiagleft(0x00, 0x07);





```
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);
```



ssd1306_128x32_i2c

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

```
// Draw each snowflake:
for(f=0; f< NUMFLAKES; f++) {</pre>
  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++) {</pre>
  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);
```



Code for displaying image

D

ø

Displaying_Image | Arduino 1.8.19

File Edit Sketch Tools Help

Displaying_Image

#include <Adafruit SSD1306.h>

#define OLED_Address 0x3C // 0x3C device address of I2C OLED. Few other OLED has 0x3D
Adafruit SSD1306 oled(128, 64); // create our screen object setting resolution to 128x64

// 'pp', 128x64px

const unsigned char myBitmap [] PROGMEM = {

TY BASED LEARNING

0xff, 0xff,



File Edit Sketch Tools Help

Displaying_Image

0xff, 0xff, 0xff, 0xff, 0xff, 0xc0, 0x03, 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() { }









