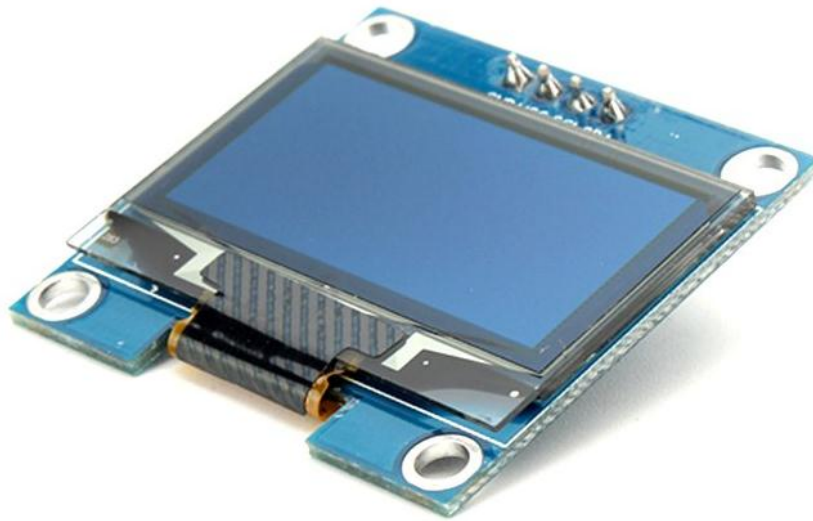


Data Specs

1.3" OLED Display Module – I²C Interface

This is an OLED monochrome (Blue) 128x64 dot matrix display module with I²C Interface. Comparing to LCD, OLED screens are way more competitive, which has a number of advantages such as high brightness, self-emission (no need backlight), high contrast ratio, wide viewing angle, wide temperature range, and low power consumption. It is compatible with Arduino. It is easy to use with the help of the U8glib library for Arduino.

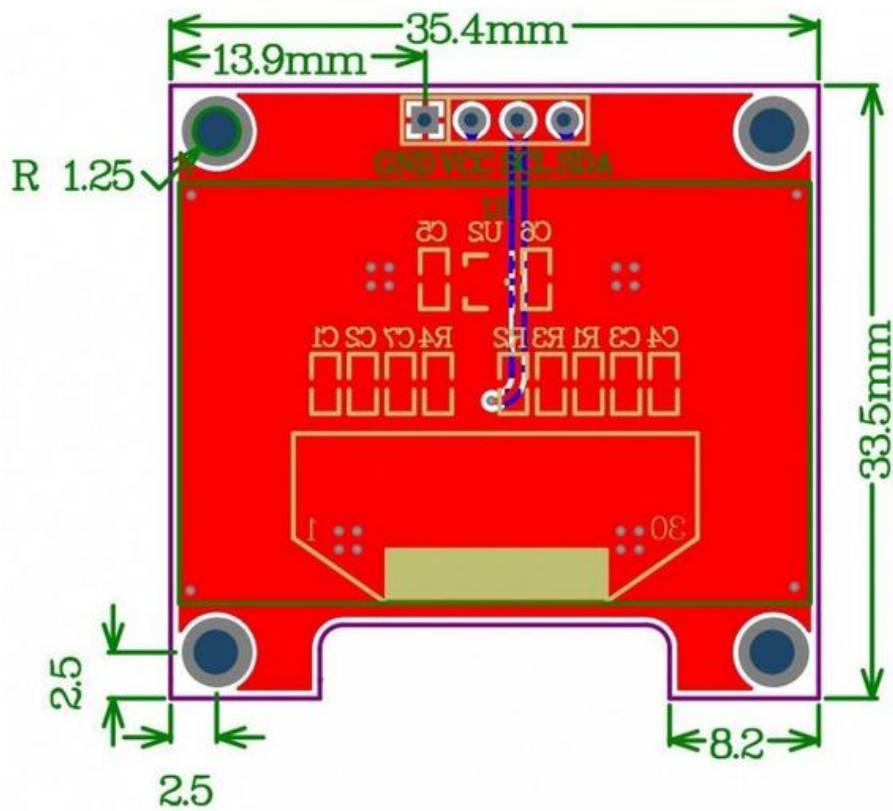
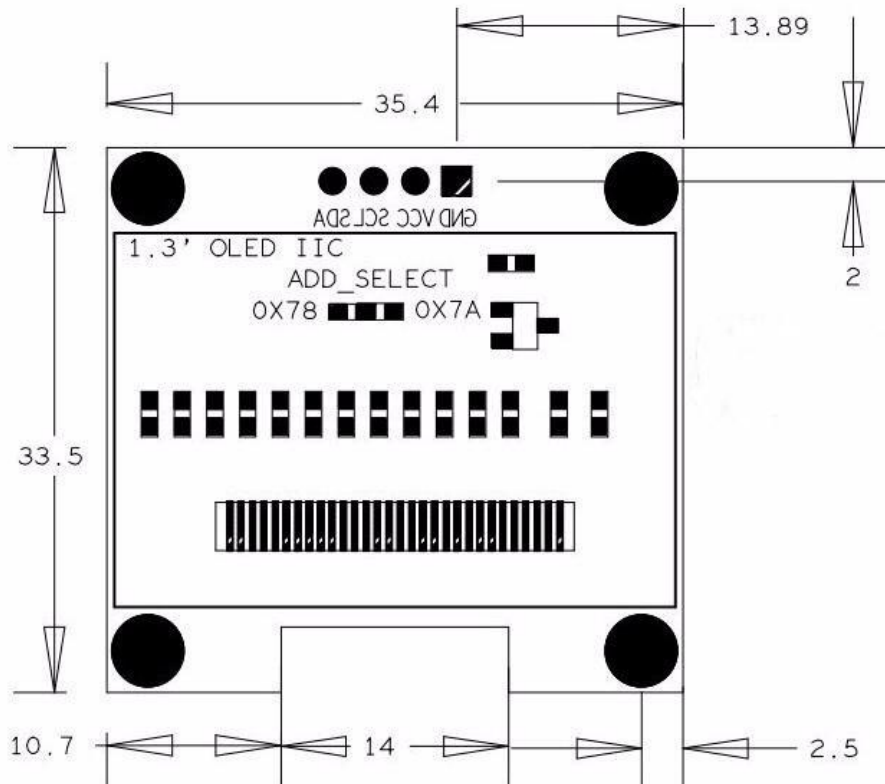


SKU: [DSP-1170](#)

Brief Data:

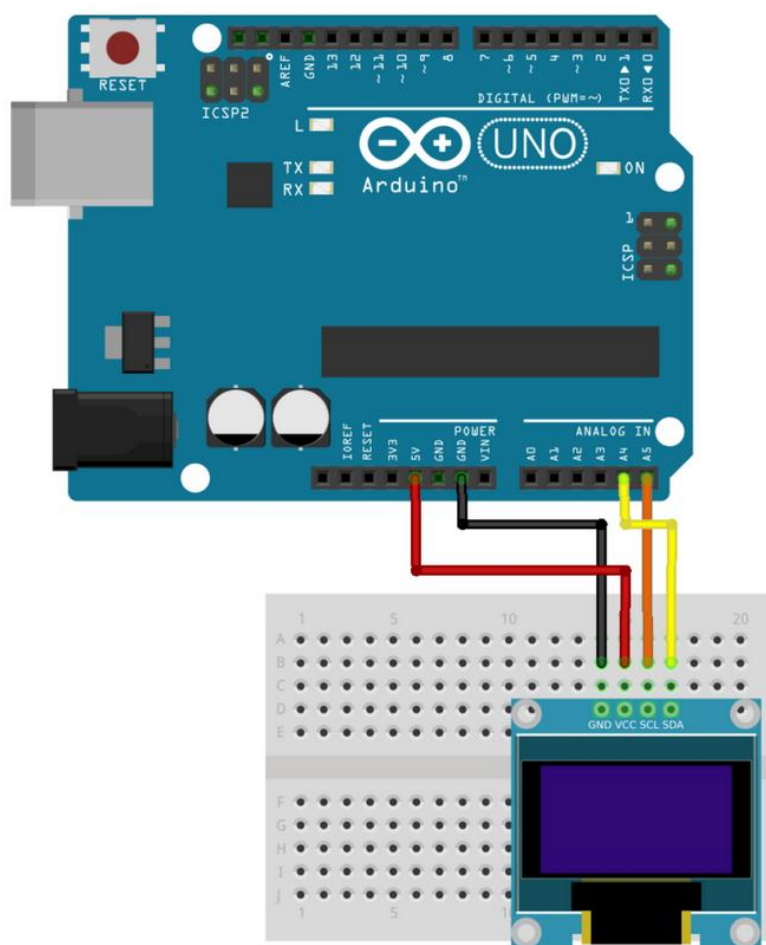
- Size: 1.3 inch.
- Light Color: Blue.
- Resolution: 128x64.
- Controlling Chip: SSH1106.
- Interface Type: I²C.
- Angle of view: > 160°.
- Display Area: 29.42x14.7 mm.
- Driving Voltage: 3.3~5V.
- Working current: 25mA (Max).
- Operating Temperature: -40~70 °C

Mechanical Dimension:



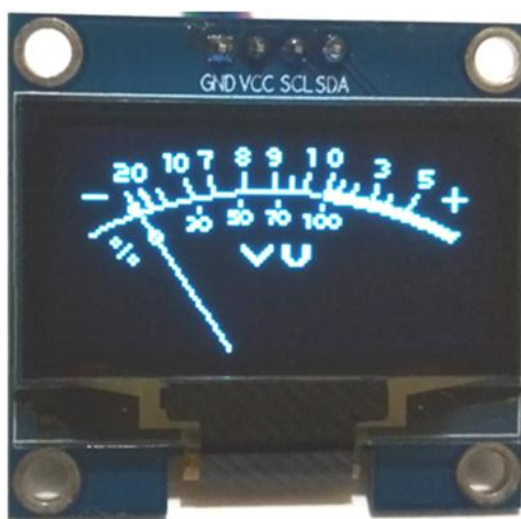
Arduino Connection Examples:

Connect the 1.3" OLED Module to Arduino Uno controller board as shown in schematic drawing below:



Arduino Sketch to Simulate Analog VU Meter:

Copy and paste the below Arduino IDE and upload to the Arduino Uno. After successfully upload the sketch, the similar image will be shown on the OLED Module:



Arduino Sketch to Simulate Analog VU Meter:

```
/* OLEDMeter was written to utilize any 128x64 display. I have only seen marginal attempts to
 * animate meters and I hope this one will set a standard. Please feel free to modify and share
 * this code for any 128x64 LCD or OLED. OLEDMeter sketch was written for use with I2C SH1106.
 * This code must be modified to work with other display devices.
 *
 * Working portion of code was taken from Adafruit Example Sound Level Sketch for the
 * Adafruit Microphone Amplifier
 * https://learn.adafruit.com/adafruit-microphone-amplifier-breakout/measuring-sound-levels
 *
 * Remaining code was written by Greg Stievenart with no claim to or any images or information
 * provided in this code. Freely published May 26, 2016.
 *
 * Software to convert background mask to 128x64 at: http://www.ablab.in/image2glcd-software/
 *
 * IMPORTANT: Sound source must be grounded to the Arduino or other MCU's to work. Usually the
 * base sleeve contact on TRS or TRRS connector is the ground.
 */

#include <Wire.h> // required to run I2C SH1106
#include <SPI.h> // required to run I2C SH1106
#include <Adafruit_GFX.h> // https://github.com/adafruit/Adafruit-GFX-Library
#include <Adafruit_SH1106.h> // https://github.com/wonho-maker/Adafruit\_SH1106

#define OLED_RESET 4 // reset required for SH1106

Adafruit_SH1106 display(OLED_RESET); // reset required for SH1106

int analogInput = A0; // analog input for outside audio source
int hMeter = 65; // horizontal center for needle animation
int vMeter = 85; // vertical center for needle animation (outside of display
limits)
int rMeter = 80; // length of needle animation or arch of needle travel

const int sampleWindow = 50; // sample window width in mS (50 mS = 20Hz)
unsigned int sample;

// VU meter background mask image:
static const unsigned char PROGMEM VUMeter[] = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x03, 0x00, 0x60, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x09, 0x04, 0x80, 0x21, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x01, 0x98, 0x08, 0x06, 0x03, 0x80, 0x21, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xA4, 0x10, 0x09, 0x00, 0x80, 0x21, 0x20, 0x07, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0xA4, 0x10, 0x06, 0x03, 0x00, 0x20, 0xC0, 0x00, 0x80, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x71, 0x80, 0xA4, 0x10, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x0A, 0x40, 0x98, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x3C, 0x00, 0x00,
  0x00, 0x00, 0x3A, 0x40, 0x00, 0x00, 0x02, 0x01, 0x00, 0x40, 0x80, 0x07, 0x00, 0x20, 0x00, 0x00,
  0x00, 0x00, 0x42, 0x40, 0x00, 0x08, 0x02, 0x01, 0x08, 0x40, 0x80, 0x00, 0x00, 0x38, 0x00, 0x00,
  0x00, 0x00, 0x79, 0x80, 0x04, 0x08, 0x02, 0x01, 0x08, 0x81, 0x10, 0x00, 0x00, 0x04, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x04, 0x08, 0x02, 0x01, 0x08, 0x81, 0x11, 0x04, 0x00, 0x38, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x02, 0x04, 0x02, 0x01, 0x08, 0x81, 0x21, 0x04, 0x00, 0x00, 0x08, 0x00,
  0x00, 0x00, 0x00, 0x84, 0x02, 0x04, 0x0F, 0xFF, 0xFF, 0xC3, 0xE2, 0x04, 0x00, 0x00, 0x08, 0x00,
  0x00, 0x00, 0x00, 0xC2, 0x01, 0x07, 0xF0, 0x00, 0x00, 0x3B, 0xFE, 0x08, 0x40, 0x40, 0x08, 0x00,
  0x00, 0xFE, 0x00, 0x62, 0x01, 0xF8, 0x00, 0x00, 0x00, 0x03, 0xFF, 0xE8, 0x40, 0x80, 0x7F, 0x00,
  0x00, 0x00, 0x00, 0x21, 0x1E, 0x00, 0x04, 0x00, 0x80, 0x00, 0x7F, 0xFE, 0x80, 0x80, 0x08, 0x00,
  0x00, 0x00, 0x03, 0x31, 0xE0, 0x00, 0x04, 0x00, 0x80, 0x04, 0x01, 0xFF, 0xC1, 0x00, 0x08, 0x00,
  0x00, 0x00, 0x07, 0x1E, 0x00, 0x40, 0x00, 0x00, 0x00, 0x04, 0x00, 0x1F, 0xFA, 0x00, 0x08, 0x00,
  0x00, 0x00, 0x07, 0xF0, 0x00, 0x40, 0x3B, 0x07, 0x60, 0x00, 0x00, 0x01, 0xFF, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x03, 0x80, 0x00, 0x00, 0x34, 0x81, 0x90, 0xCC, 0xC0, 0x00, 0x3F, 0xC0, 0x00, 0x00,
  0x00, 0x00, 0x0C, 0x00, 0x03, 0x30, 0x0C, 0x82, 0x90, 0x53, 0x20, 0x00, 0x07, 0xF8, 0x00, 0x00,
  0x00, 0x00, 0x70, 0x40, 0x00, 0xC8, 0x3B, 0x02, 0x60, 0x53, 0x20, 0x00, 0x00, 0xFE, 0x00, 0x00,
  0x00, 0x01, 0x80, 0x20, 0x01, 0xC8, 0x00, 0x00, 0x00, 0x4C, 0xC0, 0x00, 0x00, 0x3F, 0x80, 0x00,
  0x00, 0x06, 0x00, 0x00, 0x03, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xE0, 0x00,
  0x00, 0x08, 0x00, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xFC, 0x00,
  0x00, 0x30, 0x00, 0x12, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x00,
  0x00, 0x00, 0x40, 0x12, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00,
```

```

0x00, 0x00, 0xA0, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x44, 0x00, 0x00, 0x00, 0x02, 0x02, 0x30, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x03, 0x06, 0x30, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x01, 0x8C, 0x30, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x22, 0x00, 0x00, 0x00, 0x00, 0x00, 0xD8, 0x30, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x05, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x19, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 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() {
    pinMode(analogInput, INPUT);                                     // analog input for outside audio
    source
    display.begin(SH1106_SWITCHCAPVCC, 0x3C);                     // needed for SH1106 display
    display.clearDisplay();                                       // clears display from any library
    info displayed
}

void loop(){
    /*****
    Start of code taken from Adafruit Example Sound Level Sketch for the
    Adafruit Microphone Amplifier
    *****/

    unsigned long startMillis = millis();                         // start of sample window
    unsigned int PeaktoPeak = 0;                                   // peak-to-peak level
    unsigned int SignalMax = 0;
    unsigned int SignalMin = 1024;

    while ( millis() - startMillis < sampleWindow ){

        sample = analogRead(analogInput);
        if (sample < 1024) {

            if (sample > SignalMax){

                SignalMax = sample;                               // saves just the max levels
            }

            else if (sample < SignalMin){

                SignalMin = sample;                               // saves just the min levels
            }
        }
    }
}

```

```

PeaktoPeak = SignalMax - SignalMin; // max - min = peak-peak amplitude
float MeterValue = PeaktoPeak * 330 / 1024; // convert volts to arrow information

/*****
End of code taken from Adafruit Sound Level Sketch
*****/

MeterValue = MeterValue - 34; // shifts needle to zero position
display.clearDisplay(); // refresh display for next step
display.drawBitmap(0, 0, VUMeter, 128, 64, WHITE); // draws background
int a1 = (hMeter + (sin(MeterValue / 57.296) * rMeter)); // meter needle horizontal coordinate
int a2 = (vMeter - (cos(MeterValue / 57.296) * rMeter)); // meter needle vertical coordinate
display.drawLine(a1, a2, hMeter, vMeter, WHITE); // draws needle
display.display();
}

```

This particular tutorial is intended to allow you to wire up your display, load a sketch and thus let you prove to yourself that the module is working properly. Once you become more familiar with U8glib Library, SPI, and I2C communication, you will be able to make modifications that better fit your project.