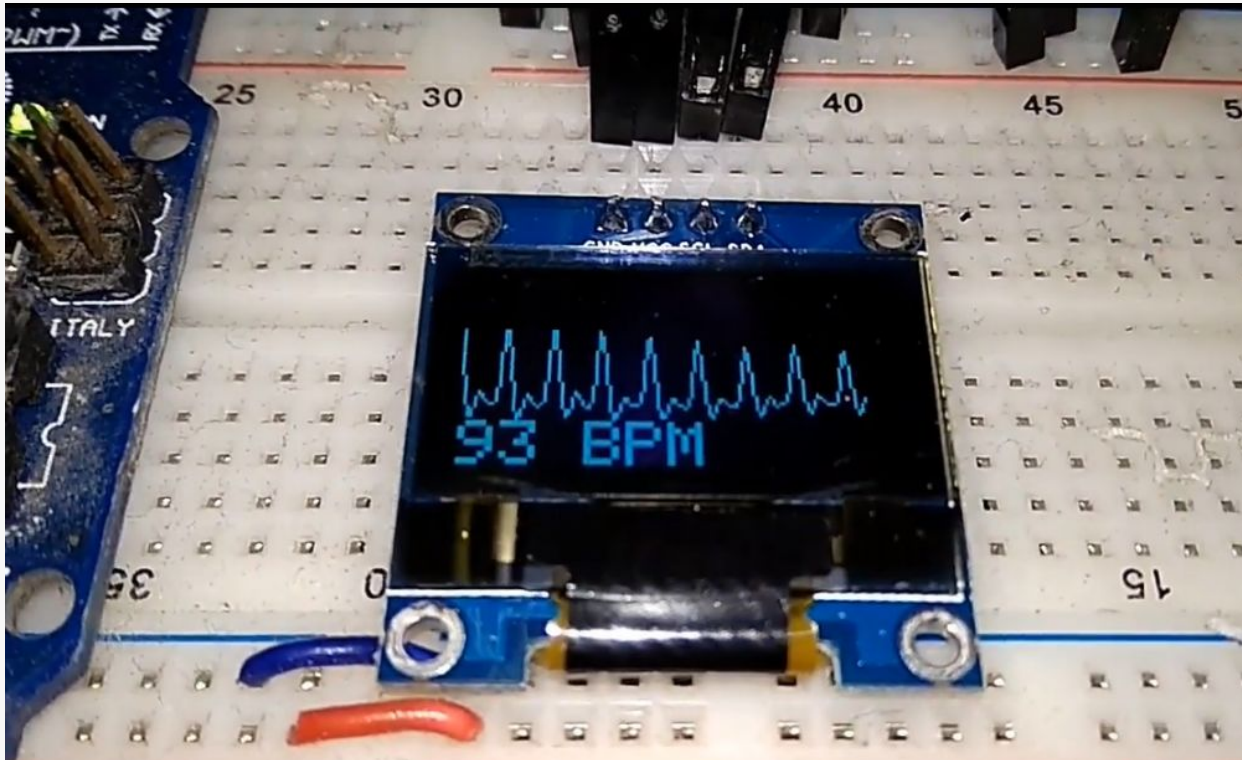


ECG display using Pulse Rate Sensor



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

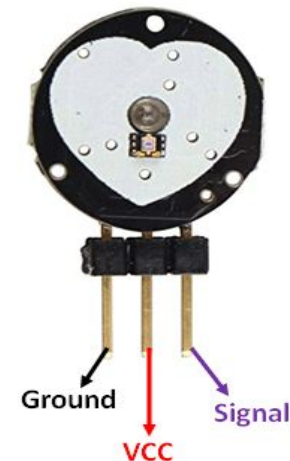
Pin 2: 3.3V to 5V

Pin 3: SCL - Serial Clock

Pin 4: SDA - Serial Data

Pulse Rate Sensor

- **Pulse Sensor** is a well-designed plug-and-play heart-rate **sensor** for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects. It also includes an open-source monitoring app that graphs your **pulse** in real time.



Working of Pulse Rate sensor

- The working of the **Pulse/Heart beat sensor** is very simple. The sensor has two sides, on one side the LED is placed along with an ambient light sensor and on the other side we have some circuitry. This circuitry is responsible for the amplification and noise cancellation work. The LED on the front side of the sensor is placed over a vein in our human body. This can either be your Finger tip or you ear tips, but it should be placed directly on top of a vein.
- Now the LED emits light which will fall on the vein directly. The veins will have blood flow inside them only when the heart is pumping, so if we monitor the flow of blood we can monitor the heart beats as well. If the flow of blood is detected then the ambient light sensor will pick up more light since they will be reflect ted by the blood, this minor change in received light is analysed over time to determine our heart beats.

Working of project

- In this project we will learn how to design ECG Display using Pulse Sensor with OLED & Arduino. We will use 0.96" OLED Display with 128×64 resolution for BPM & ECG waveform display. The I2C OLED uses only 2 wire, i.e SDA & SCK for serial communication.
- The Arduino Sketch running over the device implements the various functionalities of the project. These functions are reading sensor data, converting them into strings, passing them to I2C communication and displaying measured pulse rate/beats per minute (BPM**) in I2C OLED Display.

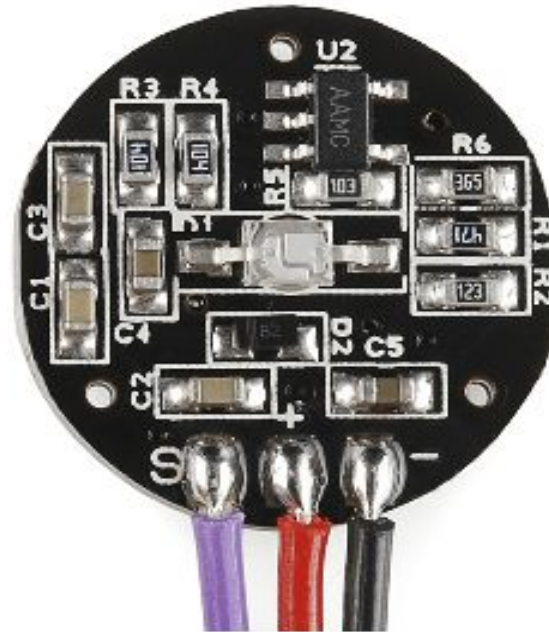
Working of project

- We need 2 different power supply, each for OLED & Pulse Sensor. We can get these supplies from Arduino Mega Board as it has 3.3 V & 5V output voltage. OLED can be connected to 3.3V & Pulse Sensor to 5V.
 - Similarly Arduino acts as a processor and converts code into predefined output. The OLED Display works on the principle of I2C Communication. It requires only 2 wires for displaying the defined output.
- ** The **Full form** of **BPM** is **Beats per Minute**. Heart rate is the speed of the heartbeat measured by the number of beats (contractions) of the heart per minute (**BPM**).

Front Side (to Skin)



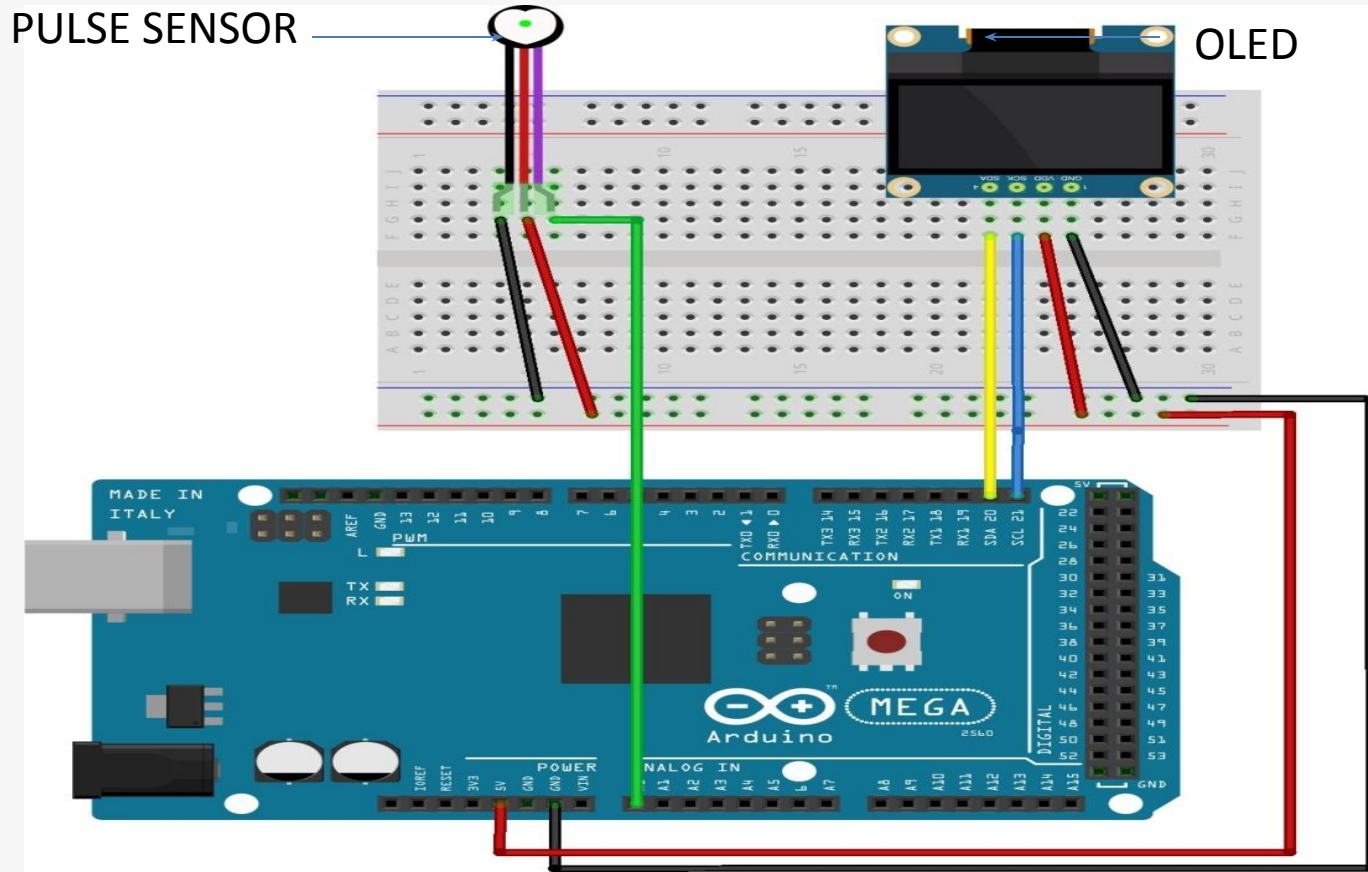
Backside



Components Required

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

Connection Diagram



Connections

- Connect SDA pin of OLED with SDA pin of Arduino.
- Connect SCK pin of OLED with SCL pin of Arduino.
- Connect Vcc and GND pin of OLED with +5V and GND pin of Arduino respectively.
- Connect S pin of pulse sensor with Ao pin of Arduino.
- Connect Vcc and GND pin of pulse sensor with +5V and GND pin of Arduino respectively.



Code

ECG_display.ino | Arduino 1.8.19

File Edit Sketch Tools Help



ECG_display.ino

```
#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
```

```
int a=0;
int lasta=0;
int lastb=0;
int LastTime=0;
int ThisTime;
bool BPMTiming=false;
bool BeatComplete=false;
int BPM=0;
#define UpperThreshold 560
#define LowerThreshold 530

void setup() {
  oled.begin(SSD1306_SWITCHCAPVCC, OLED_Address);
  oled.clearDisplay();
  oled.setTextSize(2);
}
```

```
void loop()
{
  if(a>127)
  {
    oled.clearDisplay();
  }
}
```



ECG_display.ino | Arduino 1.8.19

File Edit Sketch Tools Help

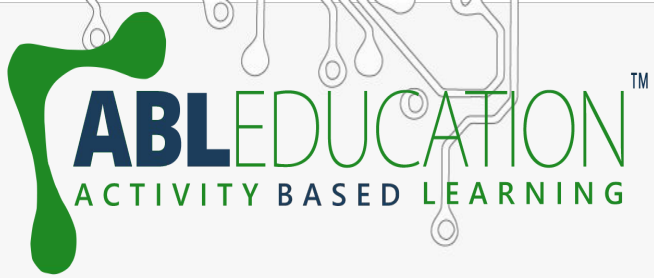


ECG_display.ino

```
void loop()
{
  if(a>127)
  {
    oled.clearDisplay();
    a=0;
    lasta=a;
  }

  ThisTime=millis();
  int value=analogRead(0);
  oled.setTextColor(WHITE);
  int b=60-(value/16);
  oled.writeLine(lasta, lastb, a,b,WHITE);
  lastb=b;
  lasta=a;

  if(value>UpperThreshold)
  {
    if(BeatComplete)
    {
      BPM=ThisTime-LastTime;
      BPM=int(60/(float(BPM)/1000));
      BPMTiming=false;
      BeatComplete=false;
      tone(8,1000,250);
    }
  }
}
```



ECG_display.ino | Arduino 1.8.19

File Edit Sketch Tools Help



ECG_display.ino

```
if (value > UpperThreshold)
{
  if (BeatComplete)
  {
    BPM = ThisTime - LastTime;
    BPM = int(60 / (float(BPM) / 1000));
    BPMTiming = false;
    BeatComplete = false;
    tone(8, 1000, 250);
  }
  if (BPMTiming == false)
  {
    LastTime = millis();
    BPMTiming = true;
  }
}
if ((value < LowerThreshold) & (BPMTiming))
  BeatComplete = true;

oled.writeFillRect(0, 50, 128, 16, BLACK);
oled.setCursor(0, 50);
oled.print("BPM:");
oled.print(BPM);

oled.display();
a++;
}
```

Project Link : <https://youtu.be/mEJqAk4G5Xc>