Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Project Report On

Heart Beat Sensor Using Arduino

Submitted to

School Of Physical Sciences

1) Miss. Utkarsha Payghan
2) Miss. Payal Patil
3) Miss. Harshada Padule

Under the guidence of

Prof. Gavhane Sir

In partial fulfilment of the degree of

Master Of Science

Department Of Physics (Applied Electronics)

Declaration

We hereby declare that student of second year of course “ M. Sc. – 2 nd “kindly submit that we have completed from time to time the seminar/ project work described in this report by our own skill for the project “ Heat Beat Sensor Using Arduino “

And the following students associated with us for this work however quantum of our contribution has been approved by teacher.

Name of Students	**Seat No.**
1) Miss. Payghan U. U.	**015625**
2) Miss. Patil P. K.	**015624**
3) Miss. Padule H. S.	**015622**

Certificate

This is to certify that

Name- 1) Miss. Payghan U. U. Seat No- 015625

2) Miss. Patil P. K. Seat No- 015624

3) Miss. Padule H. S Seat No- 015622

has successfully carried out the project **" Heart Beat Sensor Using Arduino "**assigned to them as a part of M. Sc. 2 nd curriculum. As per the rules & regulations of Punyashlok Ahilyadevi Holkar Solapur University, Solapur during academic year 2022-2023.

Place: Solapur

Date:

Project Guide

Prof. Gavhane Sir

Examiner

Head

Department Of Physics

(Applied Electronics)

Prof. B. J. Lokhande

Director

School Of Physical Sciences

Acknowledgment

It is our immense pleasure to present our project on **“ Heart Beat Sensor Using Arduino “** under the guidence of Prof. Gavhane Sir and would like to thank for his valuable support.

We also want to extend our grateful thanks to our director & H. O. D. Dept. Of Physical Sciences & our subject teacher for their indirect valuable help to us to complete our project in time without any problem.

Last but not least we express our sincere thanks to the institute Department Of Physical Sciences, Punyashlok Ahilyadevi Holkar Solapur University, Solapur for providing such a platform for implementing the ideas in our mind.

Index :-

. Title

. Abstract

. Aim

. Introduction

. Principle & working

. Diagram & components required

. Code

. Application

Title:-

Heart Beat Sensor Using Arduino

Abstract :-

Heartbeat sensor is an electronic device that is used to measure the heart rate ie. Speed of the heartbeat. Monitoring body temp, heart rate & blood pressure are the basic things that we do in order to keep us healthy.

In order to measure the body temp, we use thermometers & shygmomanometer to monitor the blood pressure.

Heart rate can be monitored in two ways: one way is to manually check the pulse either at wrists or neck & the other way is to use a heartbeat sensor.

In this project, we have designed a heartrate monitor system using arduino & heartbeat sensor.

Aim:-

To measure the heart rate ie. Speed of the heartbeat.

Introduction :-

Monitoring heart rate is very important for athletes, patients as it determines the condition of the heart (just heart rate). There are many ways to measure heart rate and the most precise one is using an Electrocardiography

But the more easy way to monitor the heart rate is to use a Heartbeat Sensor. It comes in different shapes and sizes and allows an instant way to measure the heartbeat.

Heartbeat Sensors are available in Wrist Watches (Smart Watches), Smart Phones, chest straps, etc. The heartbeat is measured in beats per minute or bpm, which indicates the number of times the heart is contracting or expanding in a minute.

Principle :-

The principle behind the working of the Heartbeat Sensor is Photoplethysmograph. According to this principle, the changes In the volume of blood in an organ is measured by the changes in the intensity of the light passing through that organ.

Usually, the source of light in a heartbeat sensor would be an IR LED and the detector would be any Photo Detector like a Photo Diode, an LDR (Light Dependent Resistor) or a Photo Transistor.

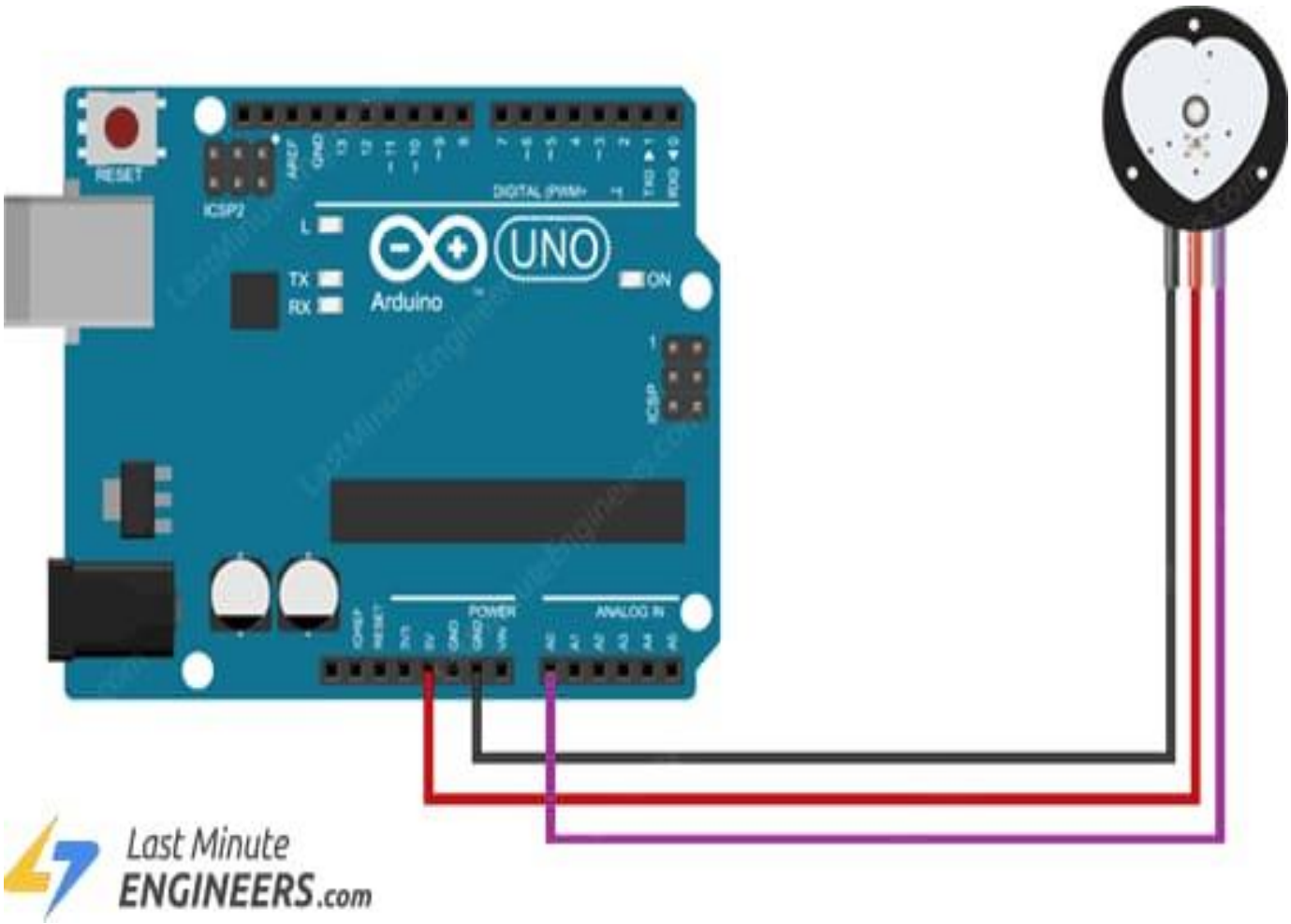
With these two i.e. a light source and a detector, we can arrange them in two ways: A Transmissive Sensor and a Reflective Sensor.

In a Transmissive Sensor, the light source and the detector are place facing each other and the finger of the person must be placed in between the transmitter and receiver.

Reflective Sensor, on the other hand, has the light source and the detector adjacent to each other and the finger of the person must be placed in front of the sensor.

A simple heartbeat sensor consists of a sensor and a control circuit. The sensor part of the heartbeat sensor consists of an IR LED and a Photo Diode.

The control circuit consists of an op-amp IC and few other components that help in connecting the signal to a microcontroller.



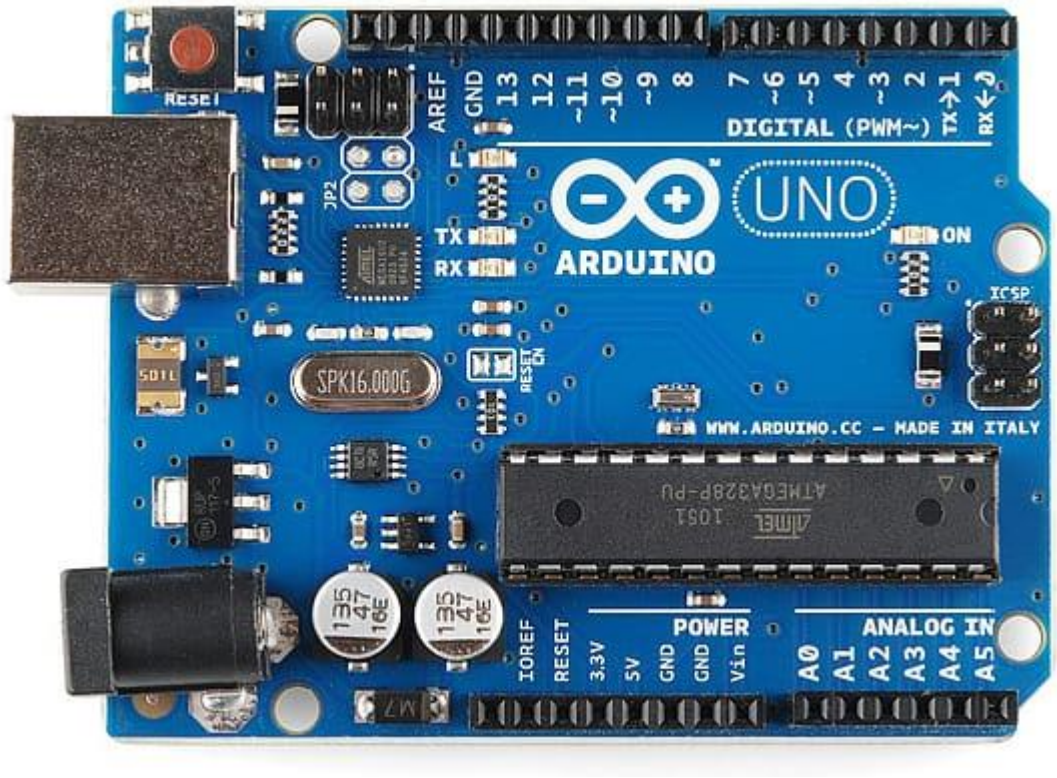
Last Minute ENGINEERS.com

Components Required :-

Arduino UNO

Connecting Wires

Sensor



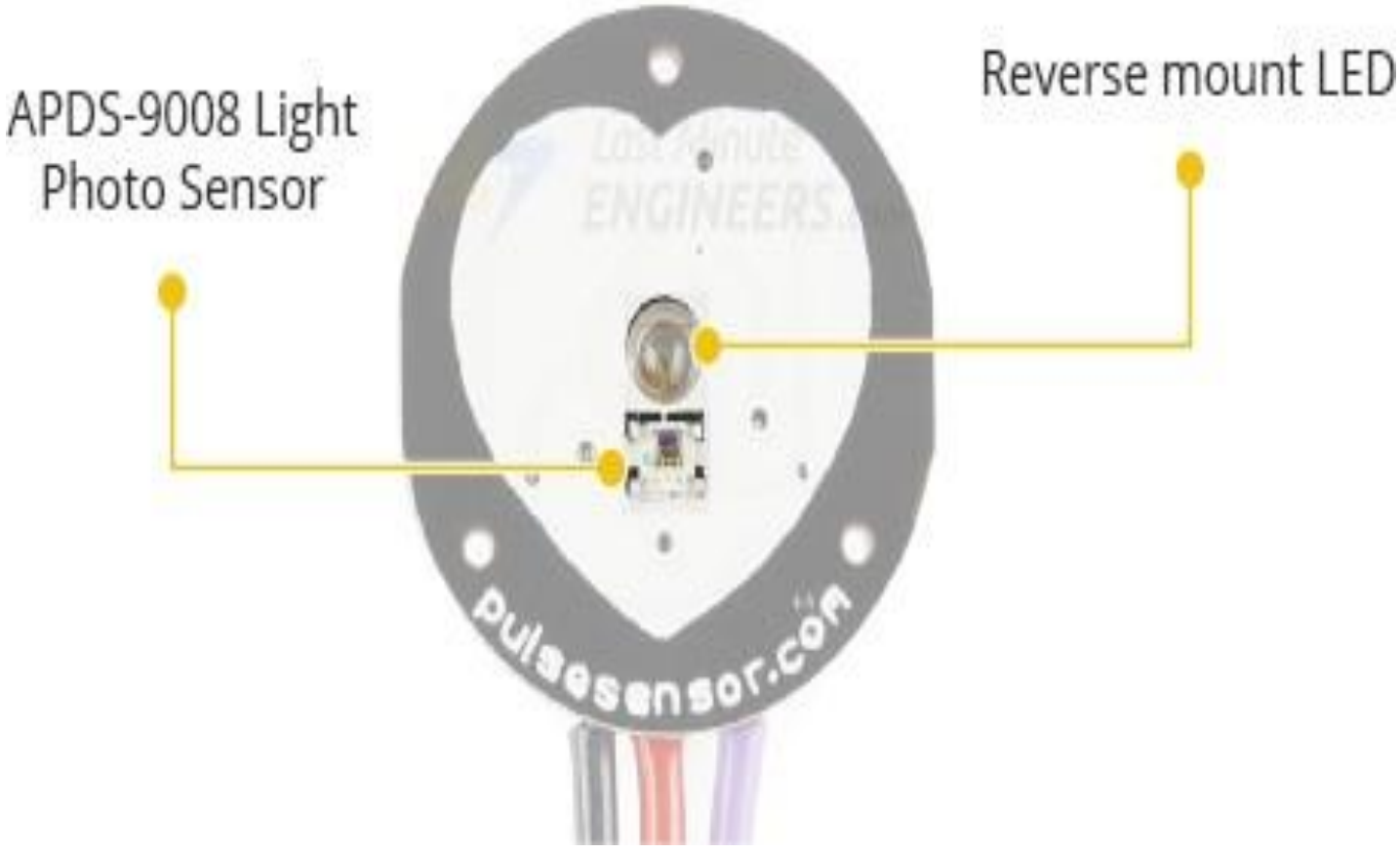
Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

The word “uno” means “one” in Italian and was chosen to mark a major redesign of the Arduino hardware and software. The Uno board was the successor of the

Duemilanove release and was the 9th version in a series of USB-based Arduino board.

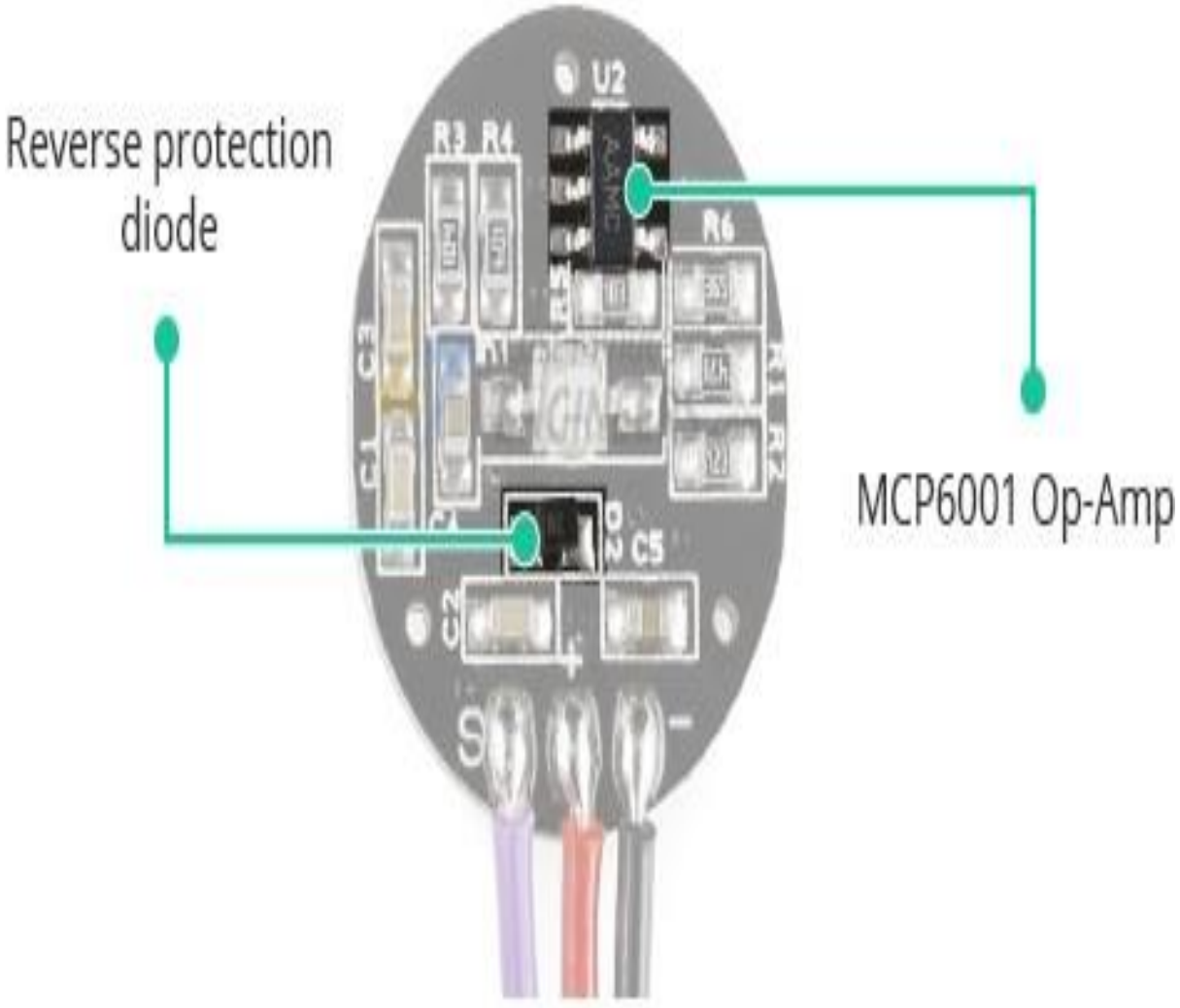
The front of the sensor, with the heart logo, is where you put your finger. You'll also notice a tiny circular opening through which the Kingbright's reverse mounted green LED shines.

Just beneath the circular opening is a small ambient light photo sensor – APDS-9008 from Avago. This sensor is similar to the ones used in cell phones, tablets, and laptops to adjust the screen's brightness based on the ambient lighting conditions.



On the back of the module are an MCP6001 Op-Amp from Microchip and a few resistors and capacitors that make up the R/C filter network. Additionally, there is a reverse protection diode to prevent damage in the event that the power leads are accidentally reversed.

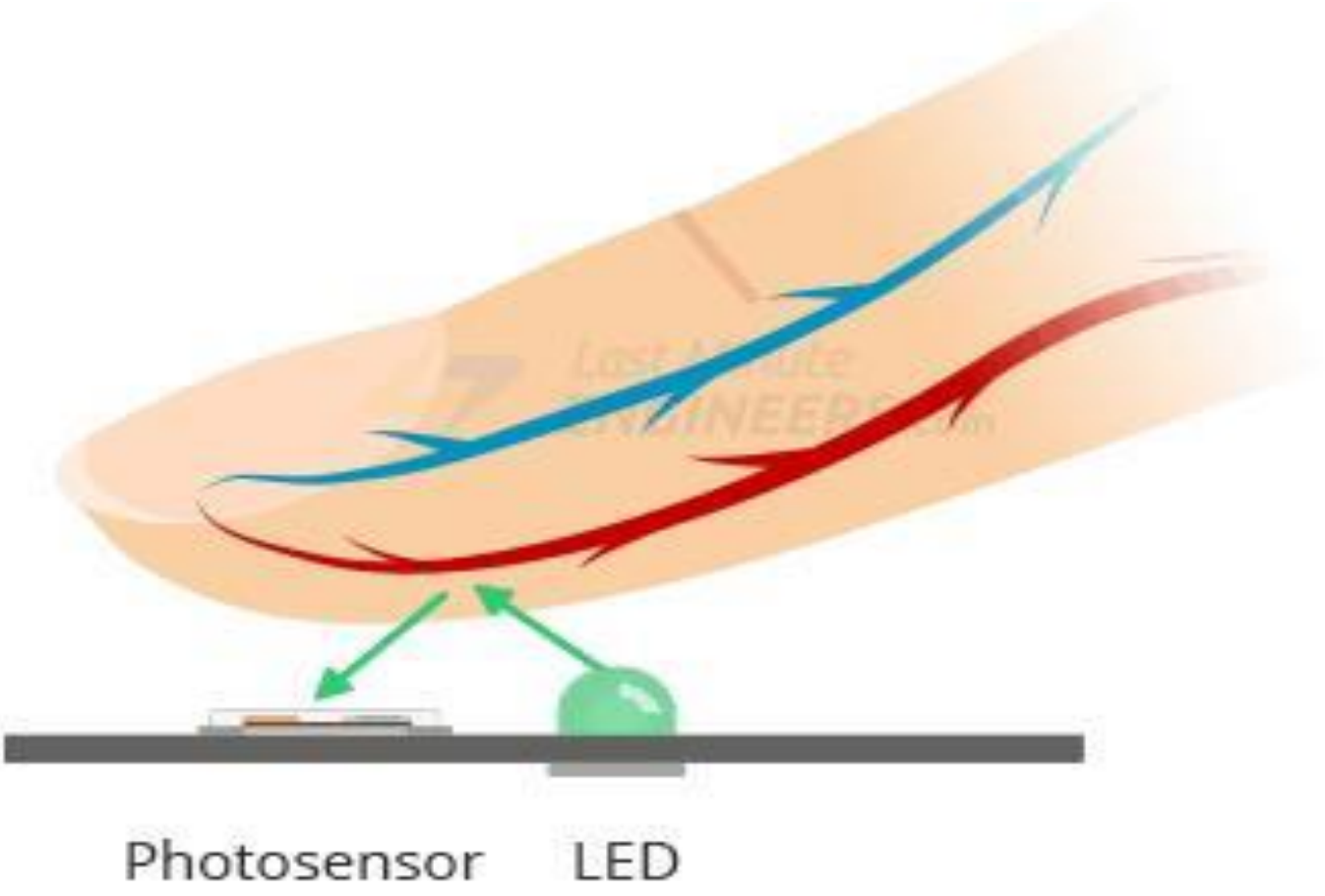
The module requires a DC power supply ranging from 3.3 to 5V and draws less than 4mA of current.



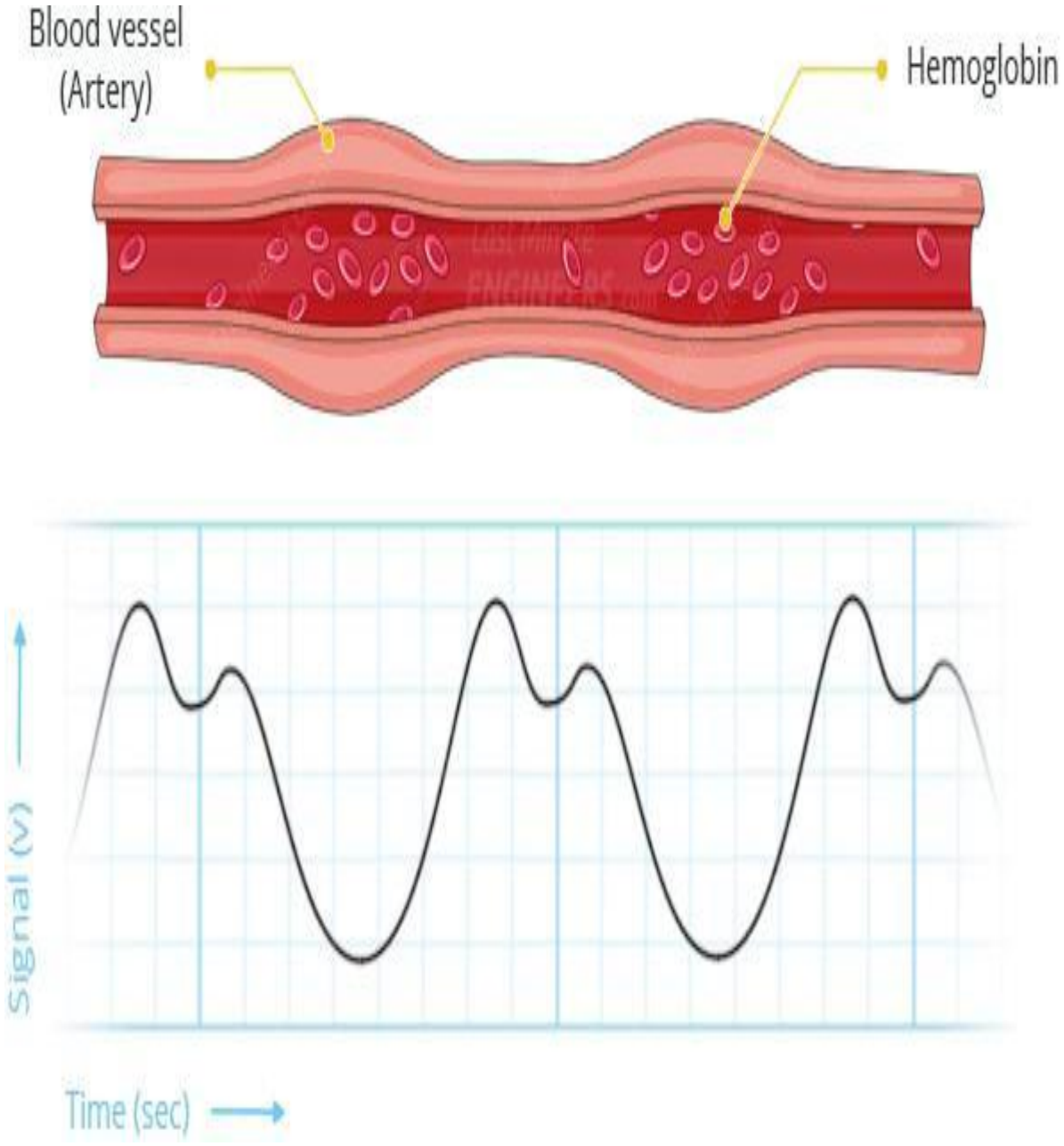
How Does a Pulse Sensor Work?

The theory behind optical heart-rate sensors is very simple. If you've ever shined a flashlight through your fingers and observed your heartbeat pulsing, the concept of optical heart-rate pulse sensors can be easily grasped.

A pulse sensor, like any other optical heart-rate sensor, works by shining a green light (~ 550nm) on the finger and measuring the amount of reflected light with a photosensor.



This optical pulse detection technique is known as a Photoplethysmogram.



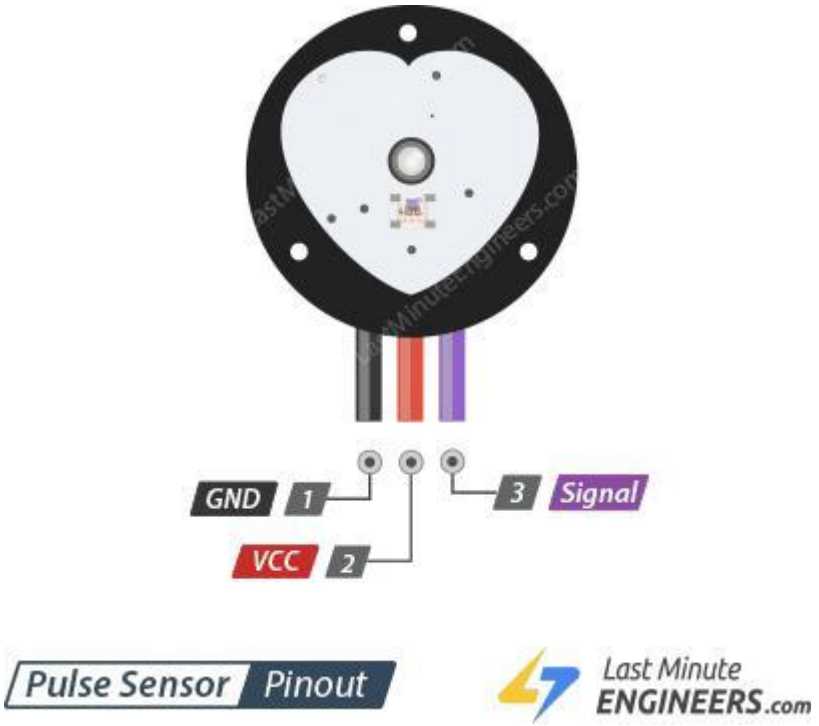
The oxygenated hemoglobin in arterial blood has the property of absorbing green light. The redder the blood (the higher the hemoglobin), the greater the absorption of green light. With each heartbeat, blood is pumped through the finger, causing a change in the amount of reflected light, which in turn produces a waveform at the photosensor's output.

As you keep shining light and taking photosensor readings, you quickly begin to obtain a heart-beat pulse reading.

This signal from the photosensor is typically small and noisy; therefore, it is passed through an R/C filter network and then amplified with an Op-Amp to create a signal that is significantly larger, cleaner, and easier to detect.

Pulse Sensor Pinout :-

The sensor comes with a 24” flat ribbon cable with three male header connectors. The pinout is shown in the figure below.



S (Signal) is the signal output. Connects to analog input of an Arduino

+ (VCC) is the VCC pin. Connects to 3.3 or 5V.

– (GND) is the Ground pin.

Wiring a Pulse Sensor to an Arduino

Connecting the Pulse Sensor to an Arduino is a breeze. You only need to connect three wires: two for power and one for reading the sensor value.

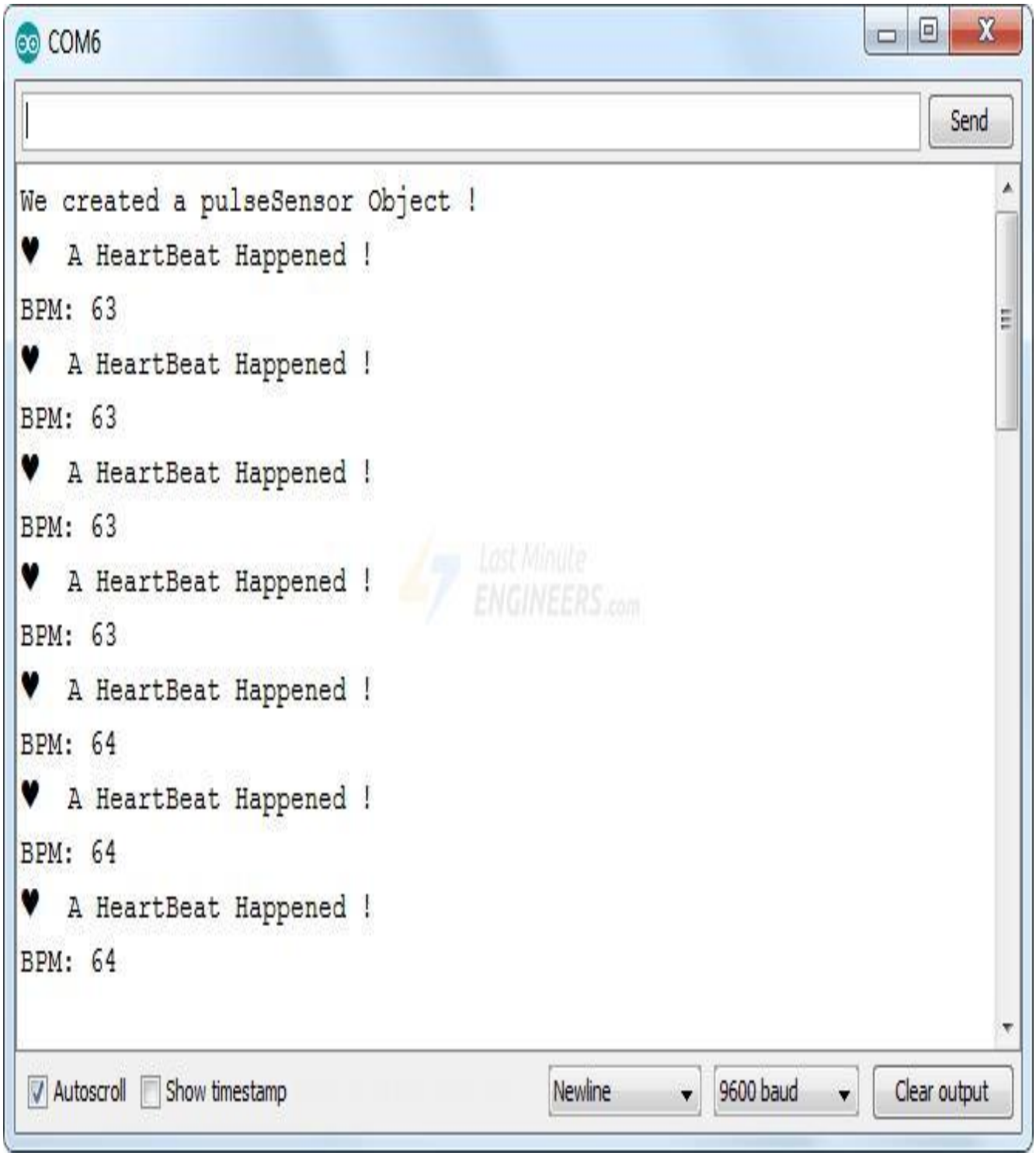
The module can be supplied with either 3.3V or 5V. Positive voltage is connected to '+,' while ground is connected to '-.' The third 'S' wire is the analog signal output from the sensor, which will be connected to the Arduino's A0 analog input.

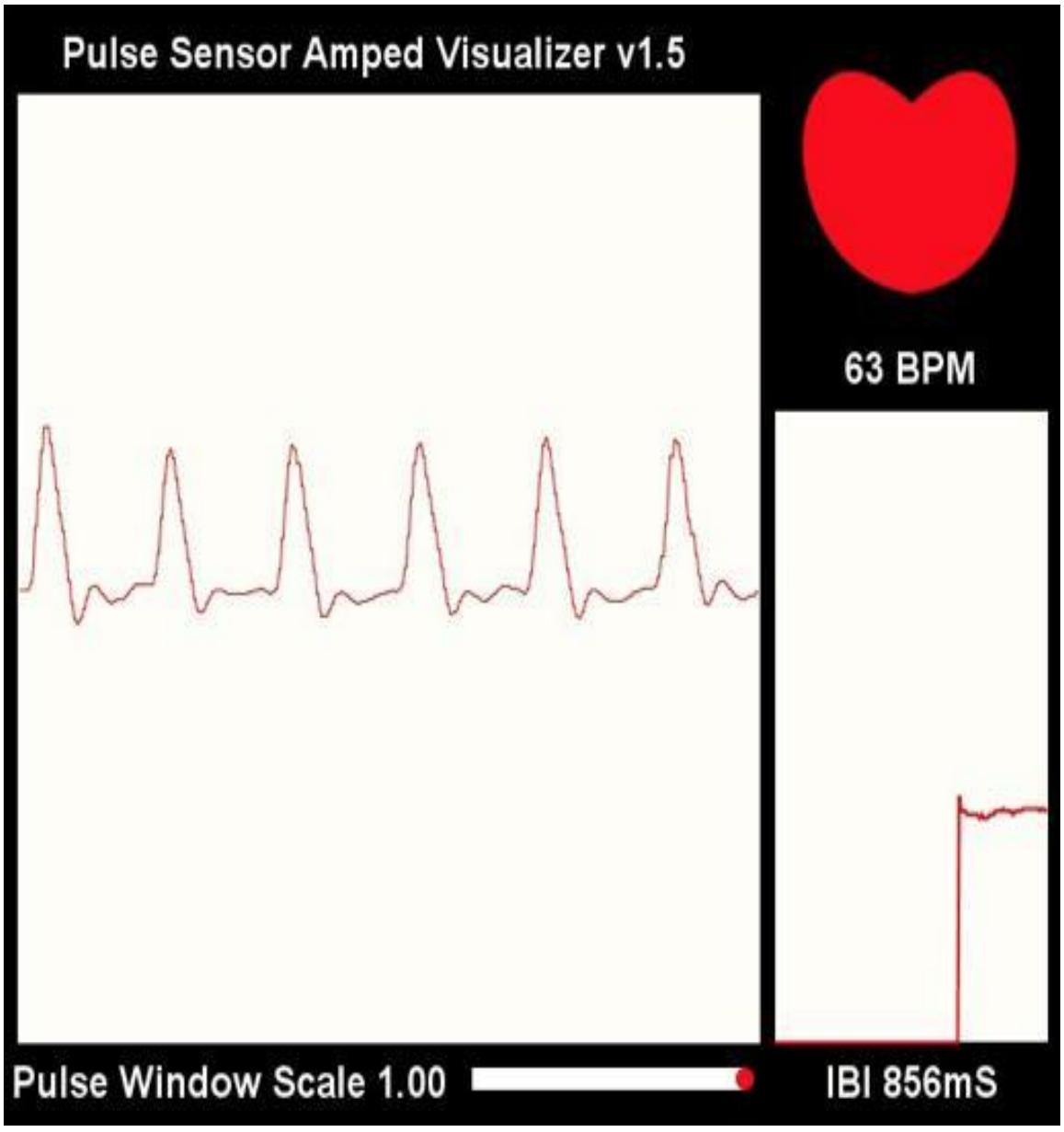
Program 1:-

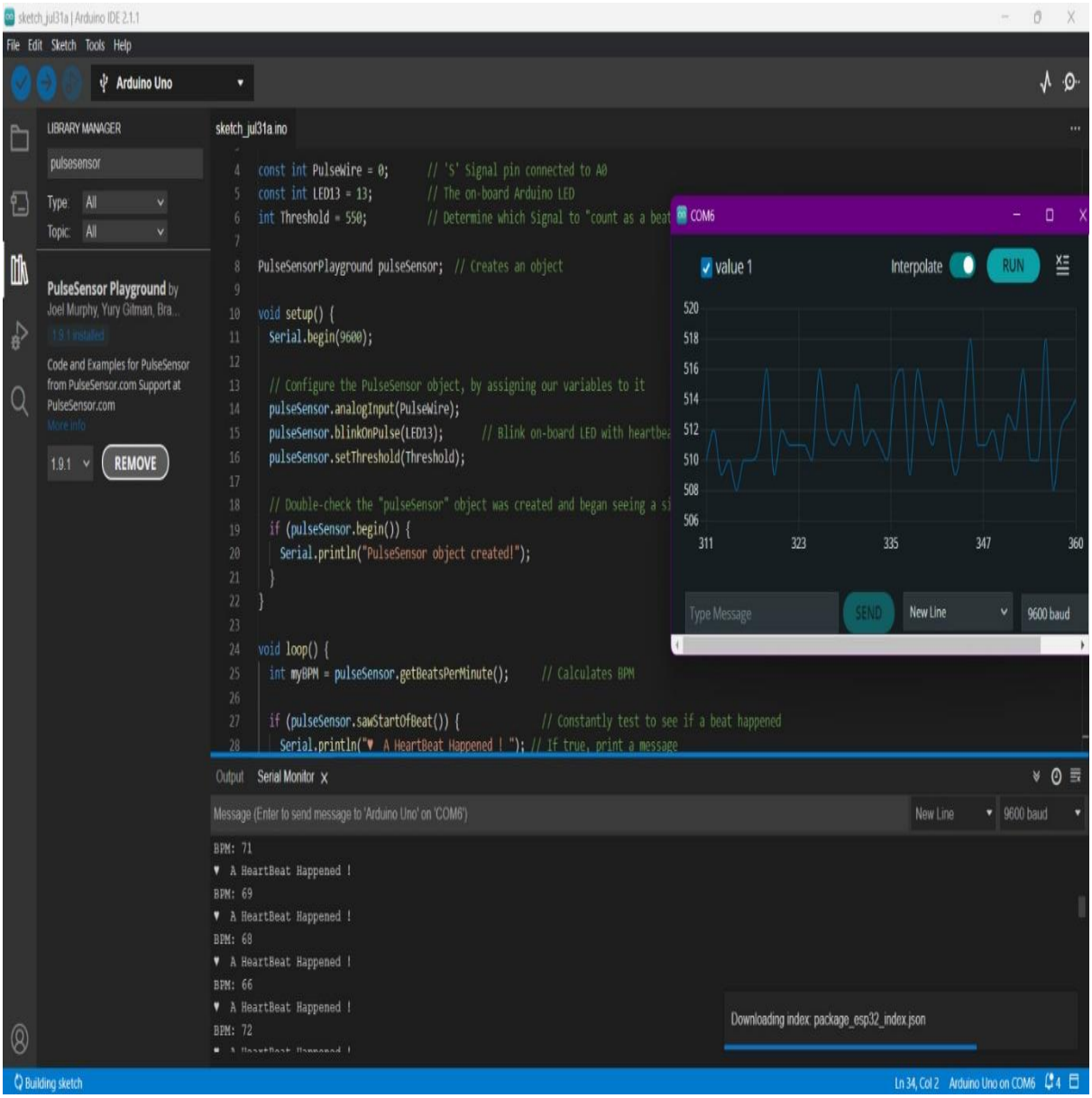
```
Int const PULSE_SENSOR_PIN = 0;   // 'S' Signal pin connected to A0
Int Signal;              // Store incoming ADC data. Value can range from 0-1024
Int Threshold = 550;      // Determine which Signal to "count as a beat" and which to ignore.

Void setup() {
        pinMode(LED_BUILTIN,OUTPUT);  // Built-in LED will blink to your heartbeat
        Serial.begin(9600);          // Set comm speed for serial plotter window
}

Void loop() {

        Signal = analogRead(PULSE_SENSOR_PIN); // Read the sensor value

        Serial.println(Signal);              // Send the signal value to serial plotter

        If(Signal > Threshold){              // If the signal is above threshold, turn on the LED
                digitalWrite(LED_BUILTIN,HIGH);
        } else {
                digitalWrite(LED_BUILTIN,LOW);     // Else turn off the LED
        }
        Delay(10);
}
```

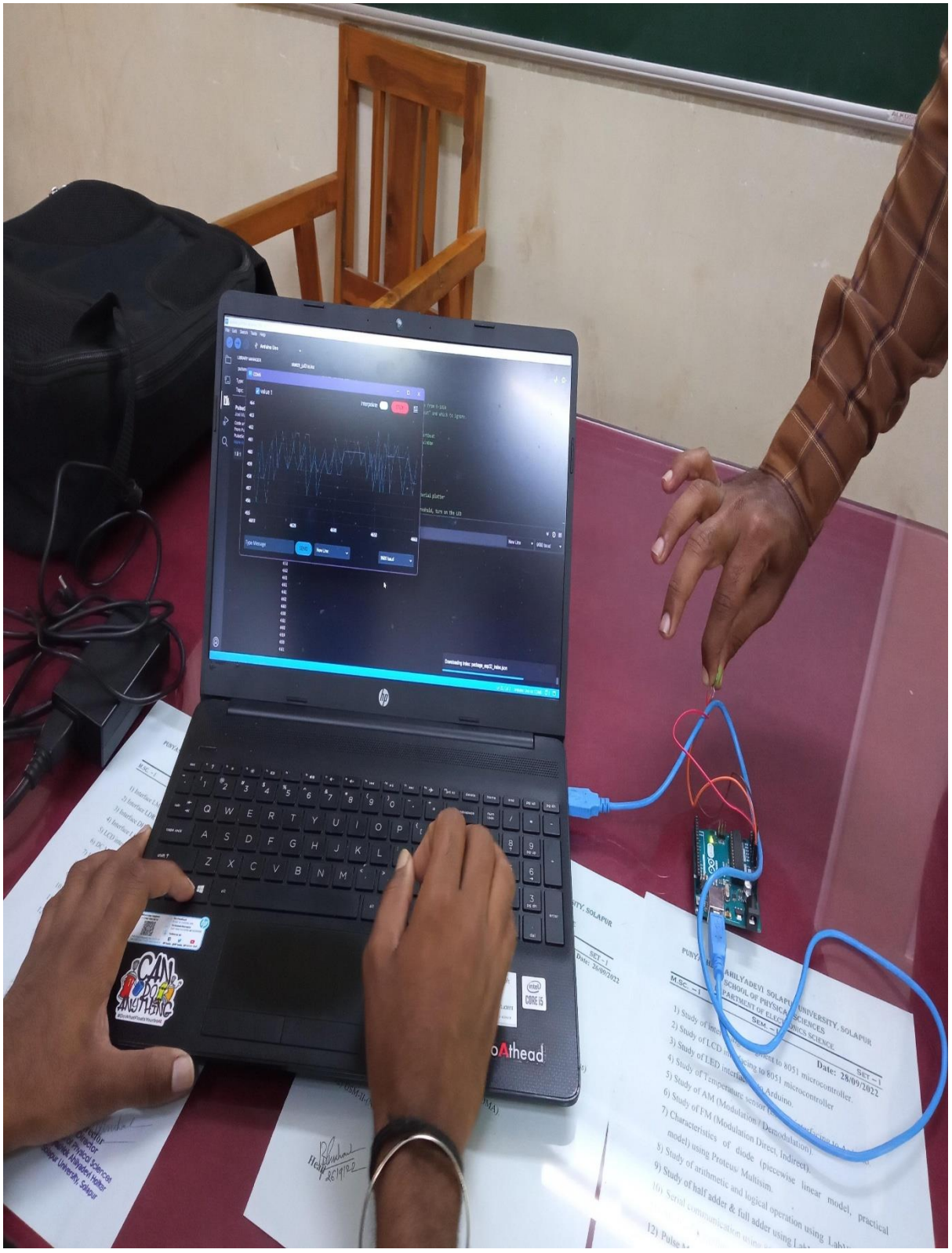
Program 2 :-

```
#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math
#include <PulseSensorPlayground.h>     // Includes the PulseSensorPlayground Library

Const int PulseWire = 0;       // 'S' Signal pin connected to A0
Const int LED13 = 13;          // The on-board Arduino LED
Int Threshold = 550;           // Determine which Signal to "count as a beat" and which to ignore

PulseSensorPlayground pulseSensor;  // Creates an object

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

        // Configure the PulseSensor object, by assigning our variables to it
        pulseSensor.analogInput(PulseWire);
        pulseSensor.blinkOnPulse(LED13);       // Blink on-board LED with heartbeat
        pulseSensor.setThreshold(Threshold);

        // Double-check the "pulseSensor" object was created and began seeing a signal
        If (pulseSensor.begin()) {
                Serial.println("PulseSensor object created!");
        }
}

Void loop() {
        Int myBPM = pulseSensor.getBeatsPerMinute();     // Calculates BPM

        If (pulseSensor.sawStartOfBeat()) {            // Constantly test to see if a beat happened
                Serial.println("❤  A HeartBeat Happened ! "); // If true, print a message
                Serial.print("BPM: ");
                Serial.println(myBPM);                  // Print the BPM value
                }

        Delay(20);
```


COM6
Send
We created a pulseSensor Object !
♥ A HeartBeat Happened !
BPM: 63
♥ A HeartBeat Happened !
BPM: 63
♥ A HeartBeat Happened !
BPM: 63
♥ A HeartBeat Happened !
BPM: 63
♥ A HeartBeat Happened !
BPM: 64
♥ A HeartBeat Happened !
BPM: 64
♥ A HeartBeat Happened !
BPM: 64
Autoscroll
Show timestamp
Newline
9600 baud
Clear output


Pulse Sensor Amped Visualizer v1.5
63 BPM
Pulse Window Scale 1.00
IBI 856mS


sketch_jul31a | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Arduino Uno
LIBRARY MANAGER
pulsosensor
Type: All
Topic: All
PulseSensor Playground by
Joel Murphy, Yury Gitman, Bra...
1.9.1 installed
Code and Examples for PulseSensor from PulseSensor.com Support at PulseSensor.com
More info
1.9.1
REMOVE
sketch_jul31a.ino
const int PulseWire = 0; // 'S' Signal pin connected to A0
const int LED13 = 13; // The on-board Arduino LED
int Threshold = 550; // Determine which Signal to "count as a beat
PulseSensorPlayground pulseSensor; // Creates an object
void setup() {
Serial.begin(9600);
// Configure the PulseSensor object, by assigning our variables to it
pulseSensor.analogInput(PulseWire);
pulseSensor.blinkOnPulse(LED13); // Blink on-board LED with heartbea
pulseSensor.setThreshold(Threshold);
// Double-check the "pulseSensor" object was created and began seeing a si
if (pulseSensor.begin()) {
Serial.println("PulseSensor object created!");
}
}
void loop() {
int myBPM = pulseSensor.getBeatsPerMinute(); // Calculates BPM
if (pulseSensor.sawStartOfBeat()) { // Constantly test to see if a beat happened
Serial.println("♥ A HeartBeat Happened ! "); // If true, print a message
COM6
value 1
Interpolate
RUN
520
518
516
514
512
510
508
506
311
323
335
347
360
Type Message
SEND
New Line
9600 baud
Output Serial Monitor X
Message (Enter to send message to 'Arduino Uno' on 'COM6')
New Line
9600 baud
BPM: 71
♥ A HeartBeat Happened !
BPM: 69
♥ A HeartBeat Happened !
BPM: 68
♥ A HeartBeat Happened !
BPM: 66
♥ A HeartBeat Happened !
BPM: 72
Downloading index: package_esp32_index.json
Building sketch
Ln 34, Col 2 Arduino Uno on COM6 4


Date: 28/09/2022
SET – 1

Applications of Heart Rate Monitor using Arduino :-

- A simple project involving Arduino UNO and Heartbeat Sensor Module is designed here which can calculate the heart rate of a person.

- This project can be used as an inexpensive alternative to Smart Watches and other expensive Heart Rate Monitors.

Conclusion :-

Upload the code to arduino UNO & power on the system. The arduino asks us to place our finger in the sensor. Place any finger on sensor (except the thumb) on the sensor. Based on data from sensor, arduino calculates heart rate & displays the heartbeat in bpm.

References :-

https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/#google_vignette

https://www.electronicshub.org/heartbeat-sensor-using-arduino-heart-rate-monitor/#Applications_of_Heart_Rate_Monitor_using_Arduino