A

Project Report

on

PULSE RATE SENSING WITH ARDUINO

Submitted to
K.K. Wagh Institute of Engineering Education and Research, Nashik.
In partial fulfillment of the requirement of Second year

Robotics and Automation

By

“Niraj Khankari”

Under the guidance of

Prof.D.B.Ghorpade

Department of Robotics and Automation

K.K. Wagh Institute of Engineering Education and Research,

Nashik 422003

2023-2024

SAVITRIBAI PHULE PUNE UNIVERSITY

K.K. WAGH INSTITUTE OF ENGINEERING EDUCATION AND RESEARCH, NASHIK.(Autonoumous)

DEPARTMENT OF ROBOTICS AND AUTOMATION
2023-2024

CERTIFICATE

This is certify that

"Niraj Khankari "

student of Second year (Robotics and Automation) was examined in the Project Based Learning entitled

"PULSE RATE SENSING WITH ARDUINO"

On 21 / April /2024

K.K. WAGH INSTITUTE OF ENGINEERING EDUCATION AND RESEARCH, NASHIK

(Prof. Devyani Ghorpade) **EXAMINER_____**

K.K. Wagh Institute of Engineering Education and Research, Nashik –422003

2023-2024

CERTIFICATE

This is certify that the Micro project report entitled

"PULSE RATE SENSING WITH ARDUINO"

Submitted By

"Niraj Khankari "

is approved for the partial fulfillment of the Second Year Robotics and Automation of Savitribal Phule Pune University.

Prof. D.B.Ghorpade
Project Guide

Prof. (Dr.) P.J. Pawar
Head of Department

ACKNOWLEDGEMENT

I feel great pleasure in expressing my deepest sense of gratitude and sincere thanks to my guide Prof.D.B.Ghorpade for his innovative guidance before and during the project work, and continuous help without whom this would have been a very difficult task.

I would also like to extend my heartfelt gratitude to the Head of Department of Robotics & Automation Engineering, Prof.Dr P.J.Pawar who provided a lot of valuable support and sincere thanks to the respected Principal, Dr. K.N.Nandurkar for his constant support and encouragement. I express my sincere thanks for the guidance, valuable assistance, and co-operation extended to all the teaching and non-teaching staff members of the department. Last but not least I pay thanks and appreciation to my colleagues who helped me in developing the project report and willingly helped me out with their abilities to complete the seminar successfully.

Submitted by:-
Niraj Khankari

INDEX

CHAPTER 1

RATIONALE

In this project, we will interface Pulse Sensor with Arduino to Measure Pulse Rate (BPM) or Heart Beat value. The Pulse rate will be displayed on 16×2 LCD Display.

A pulse sensor is a hardware device that can be used to measure heart rate in real-time. When paired with an Arduino microcontroller, you can create a simple yet effective heart rate monitor. This sensor is quite easy to use and operate. Place your finger on top of the sensor and it will sense the heartbeat by measuring the change in light from the expansion of capillary blood vessels.

The Pulse Sensor is a 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.

The essence is an integrated optical amplifying circuit and noise eliminating circuit sensor. Clip the Pulse Sensor to your earlobe or fingertip. Then it into your Arduino, you are now ready to read heart rate.

The front of the sensor comes with the heart logo. This is where you place your finger. On the front side, you will see a small round hole, from where the green LED shines. Just below the LED is a small ambient light photosensor APDS9008 which adjust the brightness in different light conditions

On the back of the module you will find MCP6001 Op-Amp IC, a few resistors, and capacitors. This makes up the R/C filter network. There is also a reverse protection diode to prevent damage if you connect the power leads reverse..

CHAPTER 2

AIMS BENEFITS OF MICRO PROJECT

The Pulse Sensor is a 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.

The Heartbeat rate information knowing is very useful while doing exercise, studying, etc. But, the heartbeat rate can be complicated to calculate. To overcome this problem, the pulse sensor or heartbeat sensor is used. This is a plug & play sensor mainly designed for Arduino board which can be used by makers, students, developers, artists who can utilize the heartbeat information into their projects. This sensor uses an easy optical pulse sensor along with amplification & cancellation of noise to make a circuit. By using this circuit, we can get fast and reliable heartbeat readings. This circuit can be operated with 4mA current and 5V voltage to use in mobile applications.

CHAPTER 3

OVERCOME OUTCOME ADDRESED

The step-by-step process of creating a Pulse Rate Monitor using an Arduino and a Pulse Sensor. This project is not only educational but also highly practical, offering a cost-effective solution for monitoring your heart rate in real-time. Whether you're a healthcare professional, a fitness enthusiast, or simply curious about electronics, this project provides valuable insights into both health monitoring and Arduino programming.

CHAPTER 4

PROPOSED METHODOLOGY

We studied about different sources from internet and futher process by studying specifications about different components we needed in project .

Components like ARDUINO UNO, Pulse Sensor ,LCD Display etc.

Pulse Sensor Technical Specifications

Physical Characteristics

- Dimensions: Approximately 0.625″ (15.875mm) in diameter
- Weight: Lightweight, usually around a few grams
- Material: Biocompatible materials for safe skin contact

Electrical Characteristics

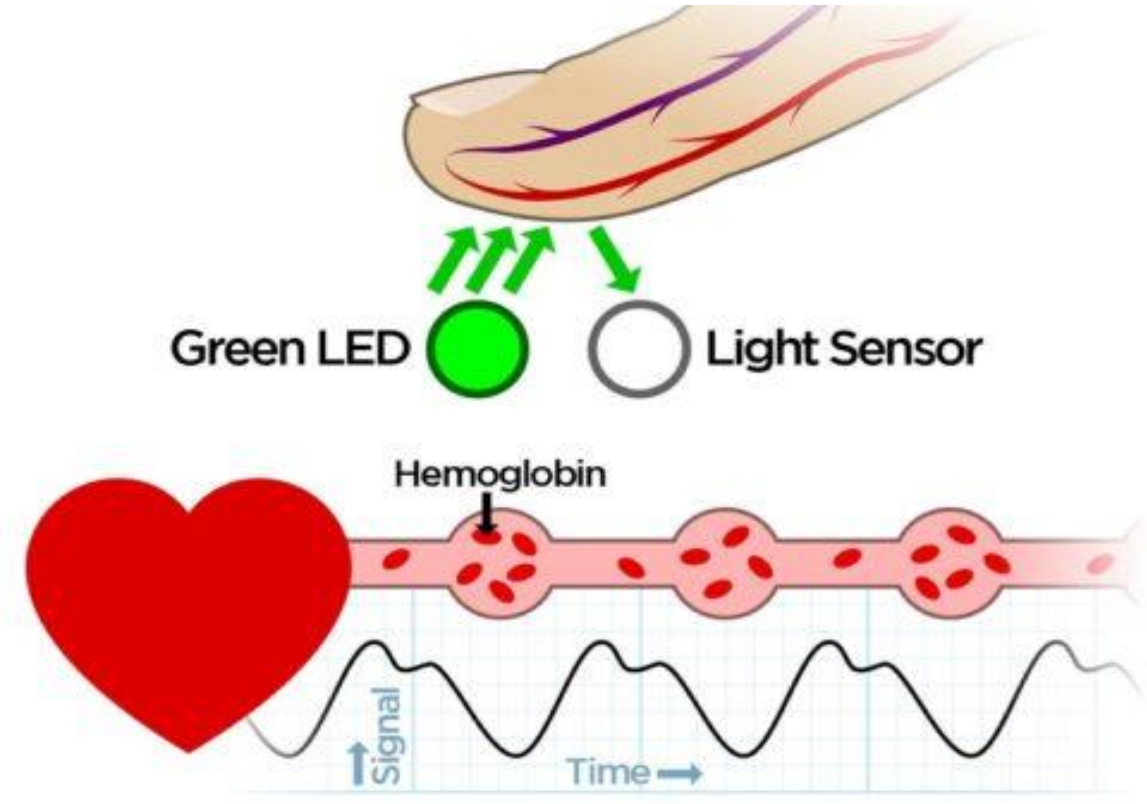
- Operating Voltage: 3V – 5.5V
- Current Consumption: Typically around 4mA
- Output Signal: Analog (0.3V to VCC)
- Signal Range: 0-1023 (10-bit ADC output of Arduino)

Sensing Technology

- Sensor Type: Photoplethysmogram (PPG)
- Wavelength: Typically around 565nm (Green LED)

We also studied about Working of Pulse Sensor

The Pulse Sensor works on the principle of Photoplethysmography (PPG), which is a non-invasive method for measuring changes in blood volume under the skin. The sensor essentially consists of two main components: a light-emitting diode (LED) that shines light into the skin and a photodetector that measures the amount of light that is reflected back. Here's a detailed explanation of its working:



1. Light Emission: A green LED emits light into the skin.
2. Reflection & Detection: The light interacts with blood and is partially reflected back, captured by a photodetector.
3. Heart Rate: Changes in reflected light create a waveform that correlates with heartbeats.
4. Oxygen Level: The amount of reflected light also indicates blood oxygen levels, as oxygenated blood absorbs more green light.
5. Signal Filtering: A Low Pass Filter cleans up the noisy, raw signal from the photodetector.
6. Amplification: An operational amplifier boosts the filtered signal for better accuracy.
7. Data Reading: Finally, an Arduino reads the amplified signal and software algorithms translate it into heart rate and potentially blood oxygen levels.

Pulse Sensor PinOut

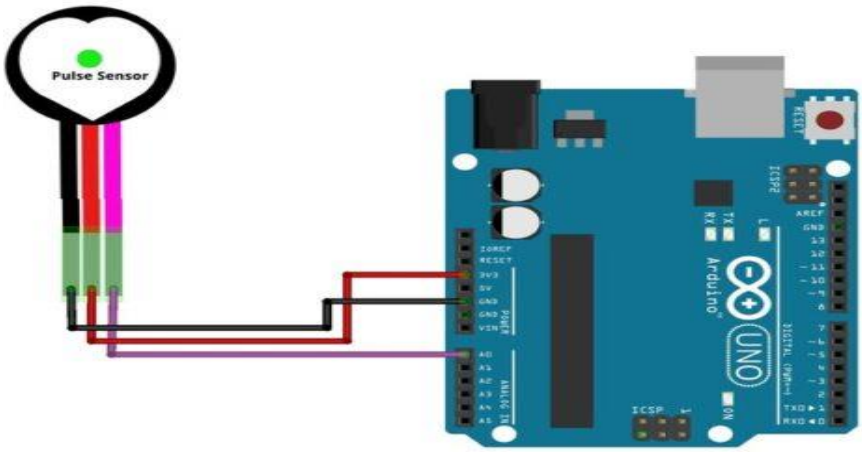
The pulse sensor has three pins: VCC, GND & Analog Pin.

Interfacing Pulse Sensor with Arduino

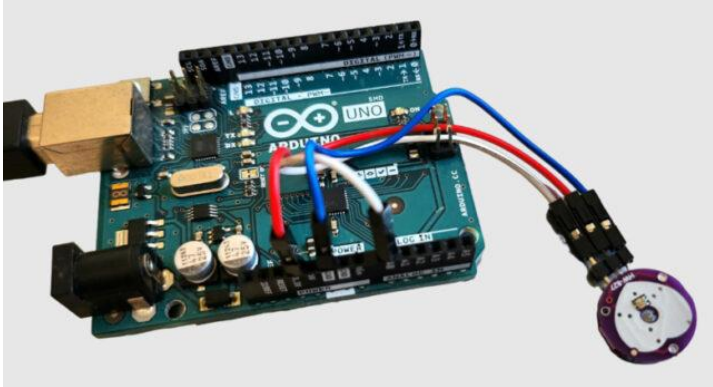
Let us interface the Pulse Sensor with Arduino and start measuring the Pulse Rate/Heart Beat/BPM Value.

Hardware Wiring Diagram

The connection diagram between Pulse Sensor and Arduino .



Using the Jumper Wires directly connect the Pulse Sensor with Arduino.



CHAPTER 5

RESOURCES REQUIRED

Sr.NO.	Components	Quantity	Price (Rupee)
1	Arduino Uno	1	495
2	Pulse Sensor	1	330
3	LCD Display	1	190
4	I2C Module	1	230
5	Female Connector	1	10
6	Jumper Wires	10-12	100
7	Soldering	-	50
TOTAL AMOUNT			1395

CHAPTER 6

ACTION PLAN

Sr.No.	Details of Activity	Plan Start Date	Plan Finish Date	Name of Responsible Team Members
1	Components Buying	8th January	22nd January	Niraj Khankari
2	Coding	23 th Jan.	12th Feb.	Niraj Khankari
3	Problems	-	-	Niraj Khankari
4	Buying I2C Module and Soldering	14 th February	21st February	Niraj Khankari
5	Working on Code Compiling	22nd Feb.	26th Feb.	Niraj Khankari
6	Presentation & Report Making	1st March	19th March	Niraj Khankari
7	Project Working/ Testing	20th March	1st April	Niraj Khankari
8	Problem Solving	2nd April	15th April	Niraj Khankari
9	Submission of Project	21th April	-	Niraj Khankari,

Skilled Developed

Chapter 7

I learned about the working and importance of arduino and pulse sensor .

I understand about the Arduino Code error and also solving some problems .

Learned about soldering of components and developed physical buying ability of components in shops and interaction skills in electronics components shop.

Understand about the specifications of components used in project ,problem solving method , not to be dependent on others while working on anything .

I developed skilled to make presentation & report for future situations of project .

Troubleshooting Pulse Sensor Reading

*To achieve stable and accurate readings from the Pulse Sensor, consider the following points:

1)Check Operating Voltage: Ensure that the sensor is operating within its stable voltage range, which is between 3.3V and 5.5V. Using voltages outside this range can lead to unstable readings.

2)Secure the Sensor: If you notice fluctuations in the readings, consider taping the sensor to your finger. This helps maintain constant blood flow, leading to more stable results.

3)Be Patient: The sensor may take some time to produce stable output. Patience is key to obtaining reliable readings.

5)Avoid External Light: Make sure to minimize exposure to external light sources when taking measurements, as they can interfere with the sensor's ability to accurately detect pulse.

6)Use a Low Pass Filter: If you're still experiencing noise in the readings, implementing a software-based low pass filter in your code can help smooth out the data.

APPLICATIONS

The applications of pulse rate sensor include the following.

- This sensor is used for Sleep Tracking
- This sensor is used for Anxiety monitoring
- This sensor is used in remote patient monitoring or alarm system
- This sensor is used in Health bands
- This sensor is used in complex gaming consoles

CONCLUSION

Pulse Rate Monitor using an Arduino and a Pulse Sensor. This project is not only educational but also highly practical, offering a cost-effective solution for monitoring your heart rate in real-time. Whether you're a healthcare professional, a fitness enthusiast, or simply curious about electronics, this project provides valuable insights into both health monitoring and Arduino programming. Applications of this technology for your personal or professional use.

CODE FOR ARDUINO UNO

1)

```
#define USE_ARDUINO_INTERRUPTS true // Set-up low-level interrupts for most acurate BPM math.
#include <PulseSensorPlayground.h> // Includes the PulseSensorPlayground Library.
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h> // F Malpartida's NewLiquidCrystal library
#define I2C_ADDR 0x27 // Define I2C Address for controller
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
#define BACKLIGHT 3

LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);
// Variables
const int PulseWire = 0; // PulseSensor PURPLE WIRE connected to ANALOG PIN 0
const int LED13 = 13; // The on-board Arduino LED, close to PIN 13.
int Threshold = 550; // Determine which Signal to "count as a beat" and which to ignore.
// Use the "Gettting Started Project" to fine-tune Threshold Value beyond default setting.
// Otherwise leave the default "550" value.
PulseSensorPlayground pulseSensor; // Creates an instance of the PulseSensorPlayground object called "pulseSensor"
void setup() {

Serial.begin(9600); // For Serial Monitor
```

```
lcd.begin(20,4);
lcd.setBacklightPin(BACKLIGHT,POSITIVE);
lcd.setBacklight(HIGH);

// Configure the PulseSensor object, by assigning our variables to it.
pulseSensor.analogInput(PulseWire);
pulseSensor.blinkOnPulse(LED13); //auto-magically blink Arduino's LED with
heartbeat.
pulseSensor.setThreshold(Threshold);

// Double-check the "pulseSensor" object was created and "began" seeing a
signal.
if (pulseSensor.begin()) {
Serial.println("We created a pulseSensor Object !"); //This prints one
time at Arduino power-up, or on Arduino reset.
lcd.setCursor(0,0);
lcd.print(" Heart Rate Monitor");

}
}

void loop() {

int myBPM = pulseSensor.getBeatsPerMinute(); // Calls function on our
pulseSensor object that returns BPM as an "int".
// "myBPM" hold this BPM value now.
if (pulseSensor.sawStartOfBeat()) { // Constantly test to see if "a beat
happened".
Serial.println("♥ A HeartBeat Happened ! "); // If test is "true", print a
message "a heartbeat happened".
Serial.print("BPM: "); // Print phrase "BPM: "
Serial.println(myBPM); // Print the value inside of myBPM.
lcd.setCursor(0,2);
lcd.print("HeartBeat Happened !"); // If test is "true", print a message
"a heartbeat happened".
lcd.setCursor(5,3);
```

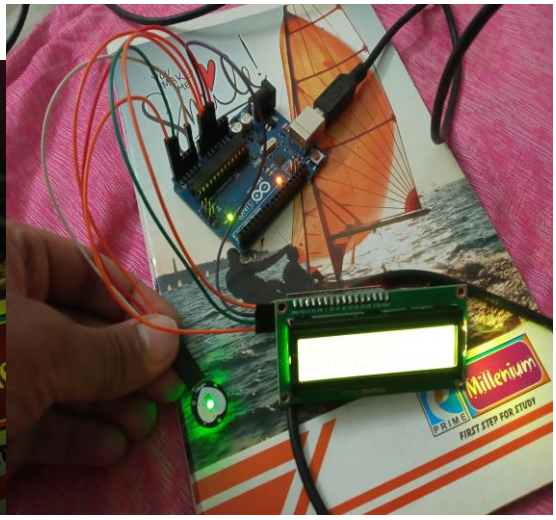
```
lcd.print("BPM: "); // Print phrase "BPM: "
lcd.print(myBPM);
}
delay(20); // considered best practice in a simple sketch.
}
```

2)

```
#include <Wire.h> // library for I2C
#include <LiquidCrystal_I2C.h> // Library for I2C LCD
LiquidCrystal_I2C lcd(0x27, 16, 2); // Initialize 16x2 LCD


int pulseSensor = A0; // Pin connected to the Pulse Sensor
int LED13 = 13; // Pin connected to onboard LED
int threshold = 550; // Determines at what point the LED will turn on
// Variables to calculate Beats Per Minute (BPM)
int pulses = 0;
int beatsPerMinute = 0;
int beatInterval = 0;
unsigned long lastTime = 0;


void setup()
{
  pinMode(LED13, OUTPUT);
  lcd.init(); // Initialize I2C LCD
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Pulse Rate Sensor");
  lcd.setCursor(0, 1);
  lcd.print("with I2C LCD");
  Serial.begin(9600); // Initialize Serial communication
}


void loop()
{
  int pulseReading = analogRead(pulseSensor); // Read the Pulse Sensor
```

```
if (pulseReading > threshold) {
  // If the reading is higher than the threshold, turn on the LED
  digitalWrite(LED13, HIGH);
  // and add 1 to the no. of pulses
  pulses++;
}
else {
  // If the reading is lower than the threshold, turn off the LED
  digitalWrite(LED13, LOW);
}
unsigned long time = millis(); // Get current time
if (time - lastTime >= 1000) {
  // If 1 second has passed, calculate the BPM
  // by counting the no. of pulses in the last second
  lastTime = time;
  beatInterval = (time - lastTime)/pulses;
  beatsPerMinute = 60000/beatInterval;
  // Display the BPM on the I2C LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("BPM:");
  lcd.setCursor(5, 0);
  lcd.print(beatsPerMinute);
  // Display the pulse sensor reading on Serial monitor
  Serial.print("Pulse Sensor Reading: ");
  Serial.print(pulseReading);
  Serial.print("\t");
  // Display the BPM on Serial monitor
  Serial.print("BPM: ");
  Serial.println(beatsPerMinute);
  pulses = 0;
}}
```

Working Results



So, working of this project with the Result is showcasing that how the project will work in real-time .This is just the model type of the idea use in digital watches, mobiles, and many different types of the devices use to measure the pulse rate of Heart or blood flow in veins in body .

The project is showing the Heart beat Rate (BPM) in the display connected to the setup with Arduino Uno.

REFERENCES

Google,Youtube ,AI,etc.

1) https://techatronic.com/heart-beat-sensor-using-arduino-bpm-monitor/

2)https://www.elprocus.com/pulse-sensor-working-principle-and-its-applications/

3)https://www.google.com/searchq=yutube+videos+on+pulse+rate+with+arduino&oq=yutube+videos+on+pulse+rate+with+arduino+&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTEyMzAzajBqMagCALACAA&sourceid=chrome&ie=UTF-8#

4)https://www.google.com/searchq=yutube+videos+on+pulse+rate+with+arduino&oq=yutube+videos+on+pulse+rate+with+arduino+&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTEyMzAzajBqMagCALACAA&sourceid=chrome&ie=UTF-8#

5)

https://www.google.com/searchq=yutube+videos+on+pulse+rate+with+arduino&oq=yutube+videos+on+pulse+rate+with+arduino+&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTEyMzAzajBqMagCALACAA&sourceid=chrome&ie=UTF-8#

6)

https://www.google.com/search?q=yutube+videos+on+pulse+rate+with+arduino&oq=yutube+videos+on+pulse+rate+with+arduino+&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTEyMzAzajBqMagCALACAA&sourceid=chrome&ie=UTF-8#

7)https://www.google.com/searchq=yutube+videos+on+pulse+rate+with+arduino&oq=yutube+videos+on+pulse+rate+with+arduino+&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTEyMzAzajBqMagCALACAA&sourceid=chrome&ie=UTF-8#