FINAL YEAR PROJECT - FINAL REVIEW

25-04-2022

DESIGN AND DEVELOPMENT OF IOT BASED HEALTH CARE MONITORING DEVICE FOR EARLY DIAGNOSIS OF HEART DISEASE USING AI

PROJECT GUIDE - DR. GEETHA C

GROUP 9B

TEAM MEMBERS

HEMANGANI - 110118031

AKSHYAH - 110118006

Timelines And Goalposts Achieved

Review – 0: *January*

1. Formulation of problem statement
2. Literature review
3. Study of heart diseases
4. Framed solution timeline
5. Dataset collection

Review – 1: *February*

1. Sensor selection and purchase
2. Fabrication, testing, calibration, and thresholding of sensors
3. Circuit design and implementation
4. C programming for data acquisition from sensors and display
5. Pre-processing, and analysis of data and feature engineering
6. Design and development of machine learning algorithms
7. Comparison with existing methods
8. Training, testing, and debugging of the ML model

Review – 2: *March*

1. Addition of blood pressure using BPT algorithm
2. Developed fall detection algorithm
3. Incorporation of Wi-Fi module using NodeMCU ESP8266
4. Configuring and programming NodeMCU
5. Successful implementation of wireless communication and display in browser
6. Integrating python and C codes
7. Design of the wearable hand sensor glove
8. Documentation

Final Review: *April*

1. Integration of GPS
2. Designing of PCB
3. Compact fabrication with inbuilt power supply
4. Performance Analysis and verification of results
5. Testing and submission

AIM:

To design a Wearable IOT healthcare monitoring and Heart Disease Prediction device using AI.

OBJECTIVES:

This project aims at designing and developing an IOT based healthcare monitoring and Heart Disease Prediction device using AI and the sub objectives are:

- To design and develop a wearable IOT healthcare monitoring system.
- Successful Data acquisition and transfer from the sensors.
- To process the data acquired and develop a machine learning model for diagnosing of Heart Disease and other ailments using AI.
- To integrate the hardware with software and analyze and validate the system.

The further improvements in the project include making Web/app user interface design and implementation for better understanding and visualization purposes.

METHODOLOGY:

The approach for the proposed project is as follows:

- **Literature survey** :
    - To learn and understand the previous work done in the related area and the feasibility of our idea and implementation.
    - To understand the concept of heart diseases and the vital signs that affect, for better selection of features for the AI model.
- **Building the wearable sensors** :
    - Sensors to be used - body movement sensor (mems ADXL), pressure sensor (MPX10DP), temperature (LM35), heartbeat (SPO2), GPS (VK-16E), GSM/GPRS, humidity sensor, wi-fi (esp8266).
    - Selection of components and design specifications.
    - Calibration and testing of individual sensors.
- **Collect Data :**
    - Data acquisition from sensors.
    - Collection of datasets from online sources (UCI machine learning repository, Hungarian heart disease dataset, Framingham, and Public Health).
- **Build a AI model :**
    - Preprocessing of the data (data cleaning, feature selection and extraction).
    - Design of suitable neural network architecture.
    - Proper parameters selection and optimization employed.
    - Training, testing and debugging of the model.

INTRODUCTION:

People all over the world are prone to chronic illnesses like chronic respiratory disease, heart disease, and diabetes. Among them, heart disease with disparate features or symptoms complicates diagnosis. Because of the emergence of smart wearable gadgets and the "Internet of Things" (IoT), solutions have become necessary for diagnosis. The hardware components (sensors) collect data from different patients. The heart feature extraction from signals is done to get significant features.

Furthermore, the feature extraction of other attributes is also gathered. All these features are collected and subjected to the diagnostic system using Machine Learning, Deep Learning and artificial intelligence techniques. Here we are planning to design a hand wearable device. IoT-based healthcare systems and to provide a systematic review of its enabling technologies, services, and applications. the IoT-technology has helped healthcare professionals to monitor and diagnose several health issues, measure many health parameters, and provide diagnostic facilities at remote locations.

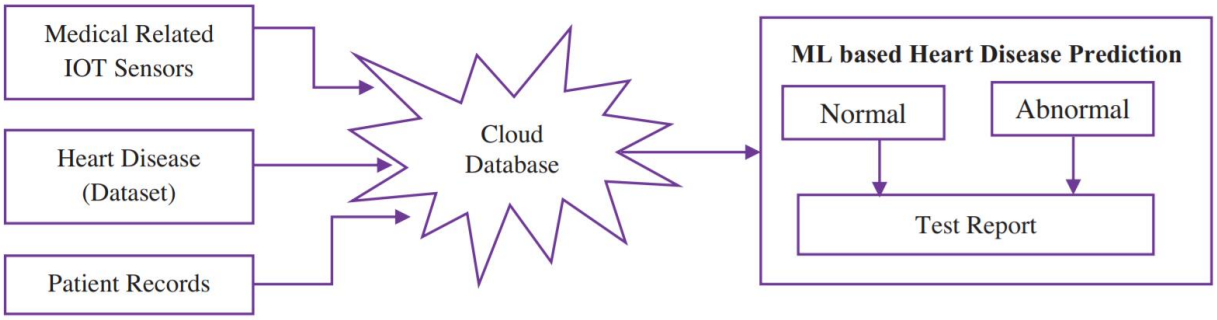
The major disease caused by human death nowadays is heart disease, due it happens suddenly and without significant symptoms, leads patient to miss the best time for first aid. With the development of IoT technology combined with the healthcare industry. It is providing technical support for clinic staff to predict and monitor heart disease patients remotely. In this paper, the main goal is to review the most relevant and latest papers to find the advantages and disadvantages and gaps in this area.

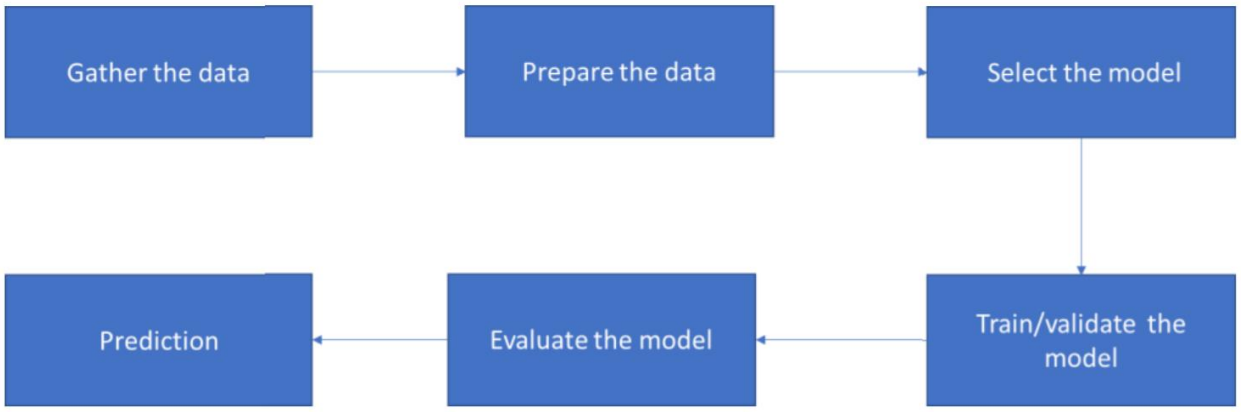
Furthermore, compare the different proposed method's performance and present the best framework for heart disease continuous prediction and monitoring. Many researchers have been already providing the use of different types of machine learning algorithms to predict and diagnose heart disease. However, most of the previous researchers use the data collected from the dataset.

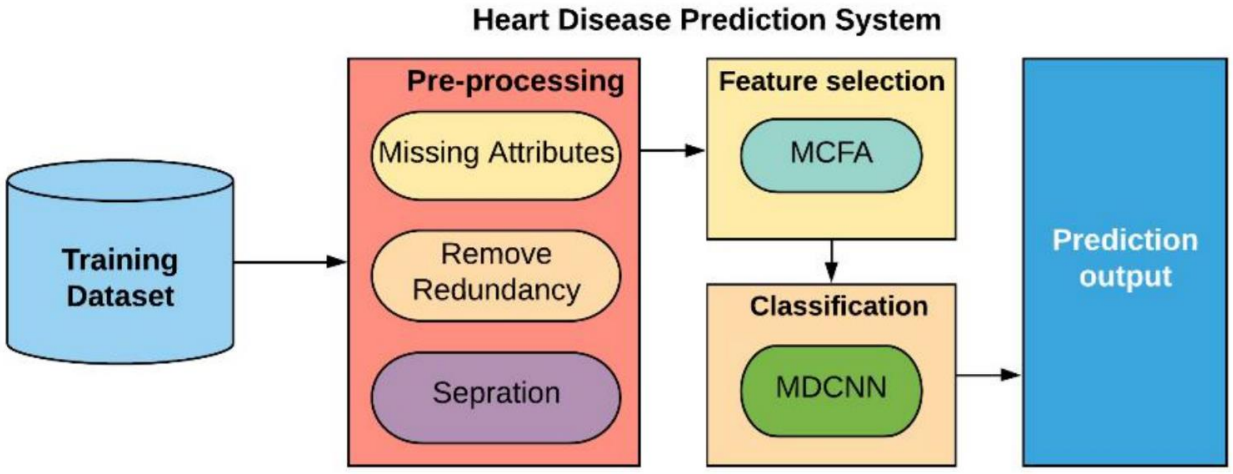
REFERENCE:

- A. A. Mutlag, M. K. Abd Ghani, N. Arunkumar, M. A. Mohammed, and O. Mohd, "Enabling technologies for fog computing in healthcare IoT systems," Future Generation Computer Systems, vol. 90, pp. 62–78, 2019.
- B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare," Future Generation Computer Systems, vol. 78, pp. 659–676, 2018.
- A. M. Rahmani, T. N. Gia, B. Negash et al., "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach," Future Generation Computer Systems, vol. 78, pp. 641–658, 2018.
- S. Sharma and S. Singh, "Heart disease diagnosis using genetic and particle swarm optimization," International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 8, 2014.
- M. A. Khan, "An IoT framework for heart disease prediction based on MDCNN classifier," IEEE Access, vol. 8, pp. 34717–34727, 2020.

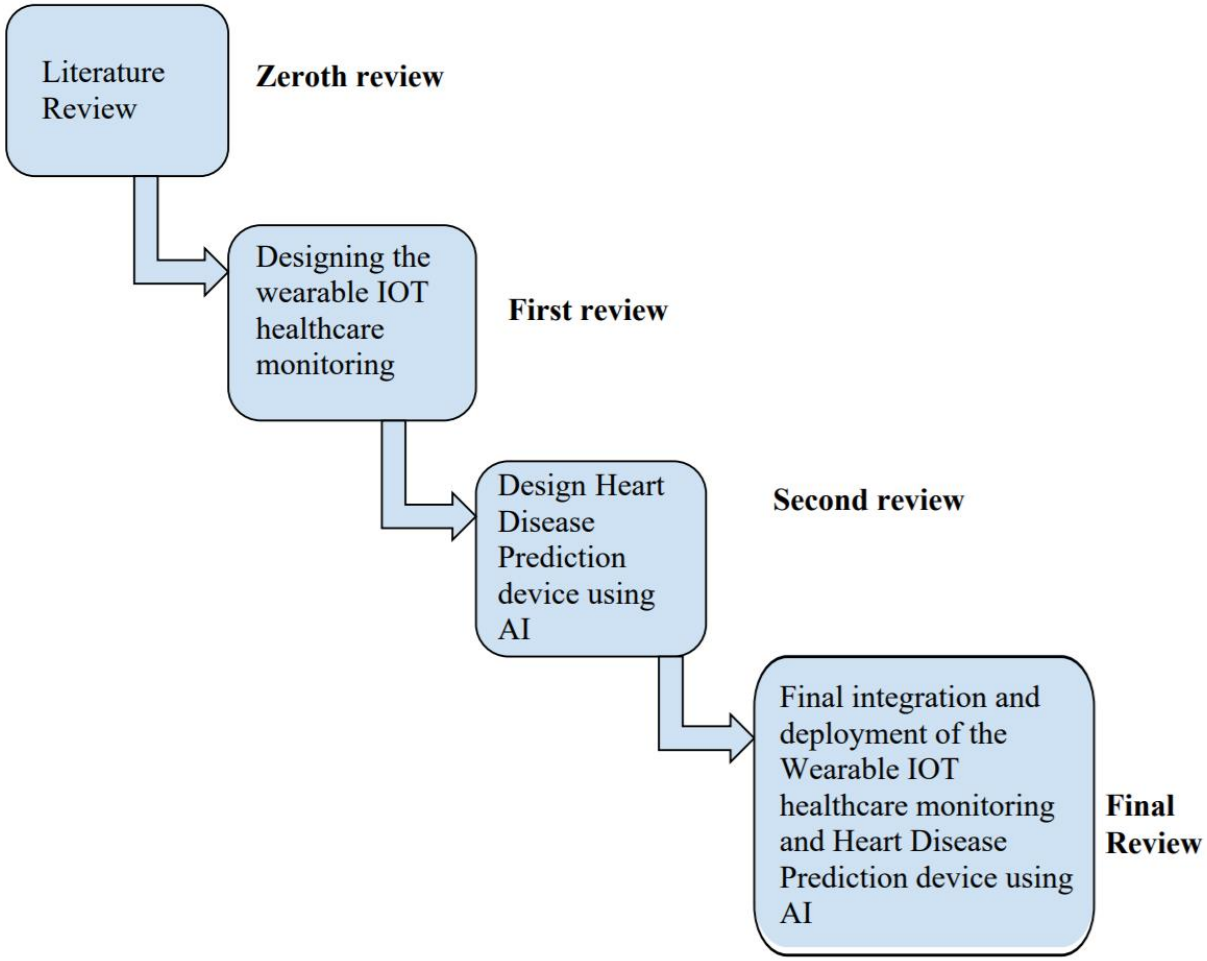
Flowchart and block diagrams of the system to be designed







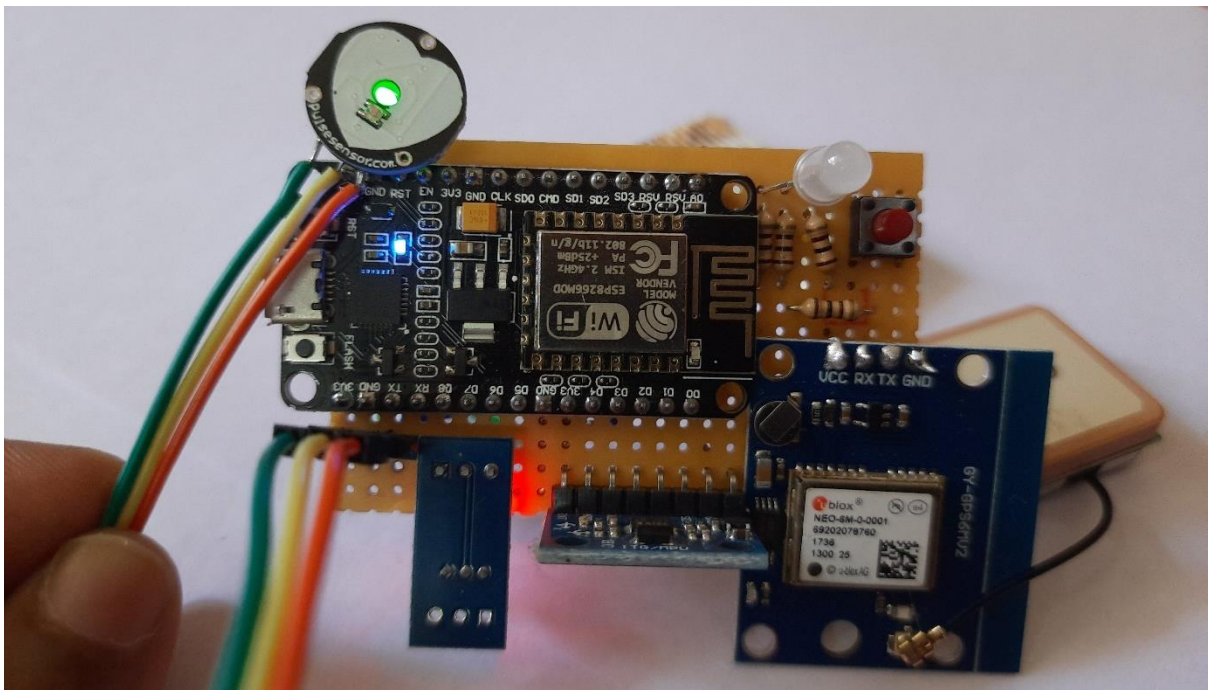
WORK PLAN:



CONTRIBUTIONS:

Literature Survey	Hemangani, Akshyah
Documentation	Hemangani, Akshyah
Fabricating the wearable sensors glove and interfacing	Hemangani
Data Acquisition	Hemangani
Data Processing and Model Construction and Testing and Debugging	Hemangani
Diagnosis and Decision making	Hemangani

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of 5CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 6 features that can be used to predict possible heart disease. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia, or already established disease) need early detection and management wherein a machine learning model can be of great help.



Dataset creation

Attribute Information:

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. RestingBP: resting blood pressure [mm Hg]
5. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
6. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
7. HeartDisease: output class [1: heart disease, 0: Normal]

This dataset was created by combining different datasets already available independently but not combined before. In this dataset, 5 heart datasets are combined over 6 common features which makes it the largest heart disease dataset available so far for research purposes.

The five datasets used for its curation are:

1. Cleveland: 303 observations
2. Hungarian: 294 observations
3. Switzerland: 123 observations
4. Long Beach VA: 200 observations
5. Stalog (Heart) Data Set: 270 observations

Total: 1190 observations

Duplicated: 272 observations

Final dataset: 918 observations

Every dataset used can be found under the Index of heart disease datasets from UCI Machine Learning Repository on the following link:

https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/

Acknowledgments

Creators:

1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Centre, Long Beach, and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

Donor:

David W. Aha (aha '@' ics.uci.edu) (714) 856-8779

This dataset is to be used for building a predictive machine learning model for early-stage heart disease detection.

Processed Dataset

	A	B	C	D	E	F	G	H	I	J
1	age	anaemia	high_blood_pres	sex	smoking	height	weight	Alcohol	active	diagnosis of heart disease (Target Variable)
2	75	0	1	1	0	110	80	0	1	1
3	55	0	0	1	0	140	90	0	1	1
4	65	0	0	1	1	130	70	0	0	1
5	50	1	0	1	0	50;	00	0	1	1
6	65	1	0	0	0	100	60	1	0	1
7	90	1	1	1	1	120	80	0	1	1
8	75	1	0	1	0	130	80	0	1	1
9	60	1	0	1	1	130	90	1	1	1
10	65	0	0	0	0	110	70	0	0	1
11	80	1	1	1	1	110	60	0	1	1
12	75	1	1	1	1	120	80	0	1	1
13	62	0	1	1	1	120	80	0	0	1
14	45	1	0	1	0	120	80	0	1	1
15	50	1	1	1	0	110	70	0	1	1
16	49	1	1	0	0	130	90	0	0	0
17	82	1	0	1	0	120	80	0	1	1
18	87	1	0	1	0	130	70	0	1	1
19	45	0	0	1	0	110	70	0	1	1
20	70	1	1	0	0	100	70	0	0	1
21	48	1	0	0	0	120	70	1	0	1
22	65	1	1	0	0	120	80	0	0	0
23	65	1	1	0	0	130	80	1	0	1
24	68	1	1	1	1	145	85	0	1	1
25	53	0	0	1	0	110	60	1	1	0
26	75	0	1	0	0	150	90	1	0	1
27	80	0	0	1	1	30;	00	1	1	1
28	95	1	1	0	0	130	90	0	0	1

Machine learning

Dataset:

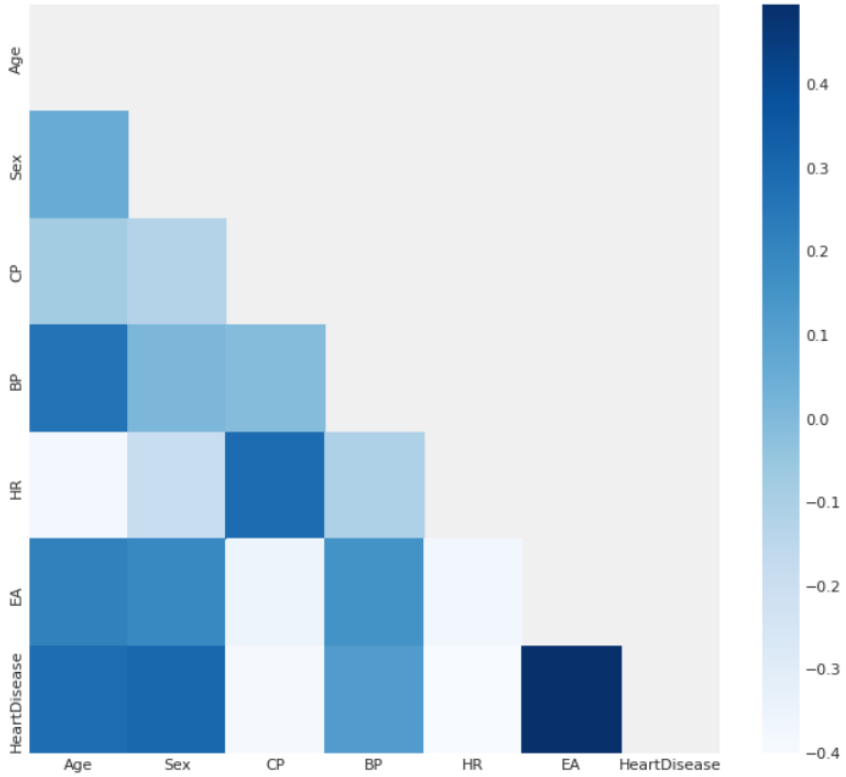
Rows = 917
Columns = 7

	Age	Sex	CP	BP	HR	EA	HeartDisease
0	40	M	ATA	140	172	N	0
1	49	F	NAP	160	156	N	1
2	37	M	ATA	130	98	N	0
3	48	F	ASY	138	108	Y	1
4	54	M	NAP	150	122	N	0

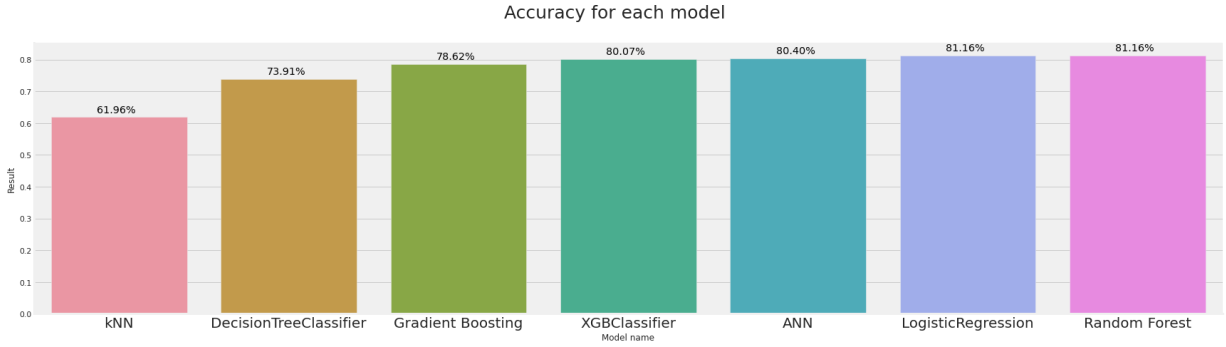
	count	mean	std	min	25%	50%	75%	max
Age	917.0	53.509269	9.437636	28.0	47.0	54.0	60.0	77.0
BP	917.0	132.540894	17.999749	80.0	120.0	130.0	140.0	200.0
HR	917.0	136.789531	25.467129	60.0	120.0	138.0	156.0	202.0
HeartDisease	917.0	0.552890	0.497466	0.0	0.0	1.0	1.0	1.0

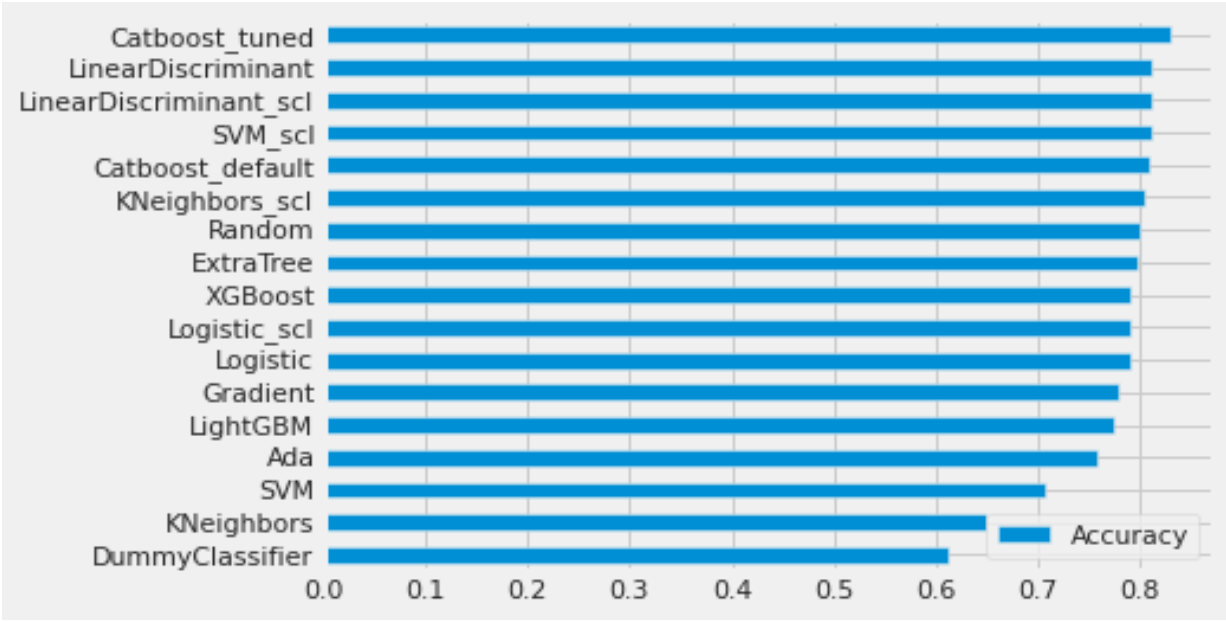
Distribution of the features:

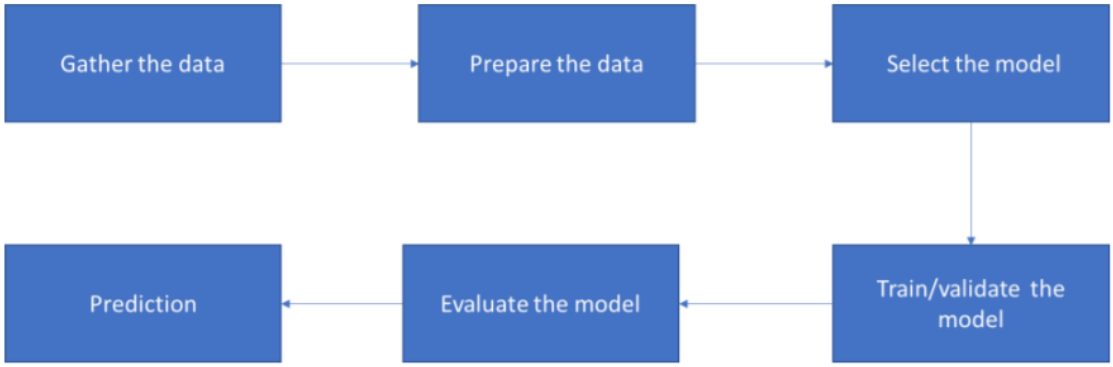
Heatmap:



Models Evaluating:

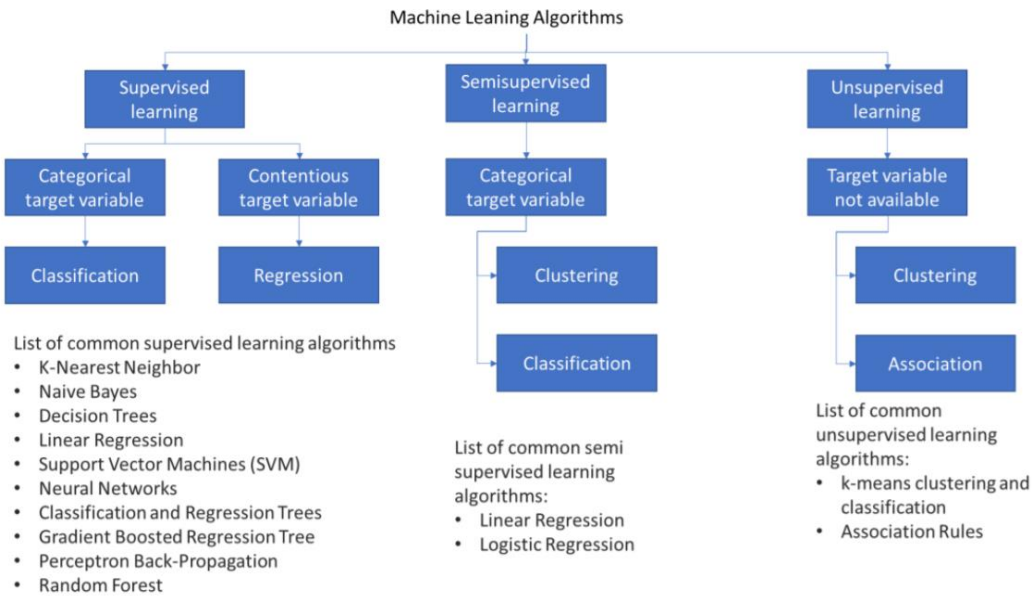






- Developed model to classify heart disease cases.
- Made the detailed exploratory analysis.
- Decided which metric to use.
- Analysed both target and features in detail.
- Transformed categorical variables into numeric so we can use them in the model.
- Used pipeline to avoid data leakage.
- Looked at the results of each model and selected the best one for the problem on hand.
- Made hyperparameter tuning of the Catboost with Optuna to see the improvement.
- Looked at the feature importance.

Algorithms used:



Components

xcluma Heart-Beat Pulse Sensor Module

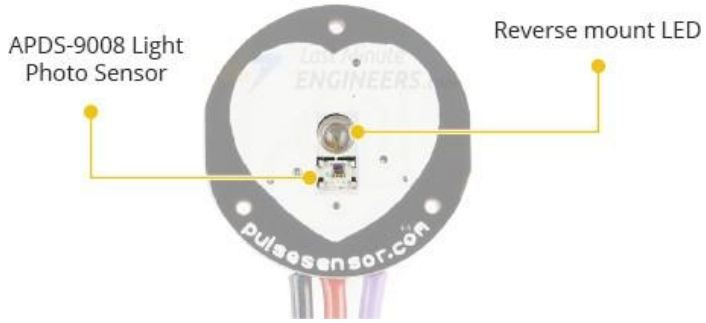
Description:

Pulse sensor is used to test the heart rate. Sensors can be worn on the finger or earlobe and can be connected with an Arduino line via the internet. It also has an open-source app program, which can put the heart rate in real time displayed by the graph. Essence is an integrated optical heart rate sensor amplifier and noise elimination circuit. The Pulse Sensor is a well-designed low-power plug-and-play heart-rate sensor for the Arduino. It can be used to incorporate live heart-rate data into the project. And the best part is that this sensor plugs right into Arduino and easily clips onto a fingertip or earlobe. It is also super small (button-shaped) with holes, so it can be sewn into fabric.

Product Dimensions:8 x 4 x 10 cm; 4 Grams

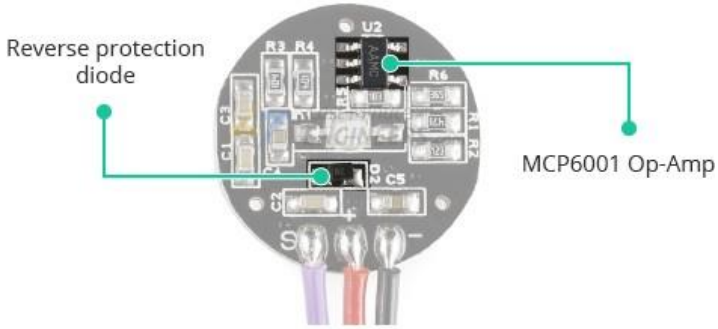
Hardware Overview:

The front of the sensor is the side with the heart logo. This is where we place our finger. On the front side there is a small round hole, from where the King Bright's reverse mounted green LED shines.



Just below the LED is a small ambient light photo sensor – APDS-9008 from Avago, similar to that used in cell phones, tablets and laptops, to adjust the screen brightness in different light conditions.

On the back of the module is the rest of the components including a microchip's MCP6001 Op-Amp and a bunch of resistors and capacitors that make up the R/C filter network. There is also a reverse protection diode to prevent damage if the power leads are accidentally reversed.



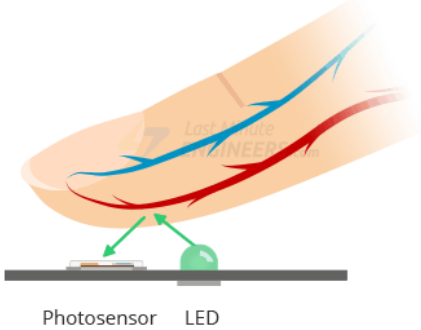
The module operates from a 3.3 to 5V DC Voltage supply with an operating current of < 4mA.

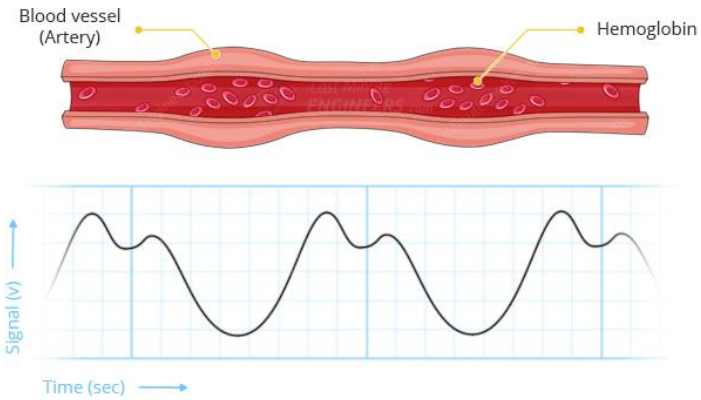
Technical specifications:

Maximum Ratings	VCC	3.0 – 5.5V
	IMax (Maximum Current Draw)	< 4mA
	VOut (Output Voltage Range)	0.3V to Vcc
Wavelength	LED Output	565 nm
	Sensor Input	525 nm
Dimensions	L x W (PCB)	15.8mm (0.625″)
	Lead Length	20cm (7.8″)

Pulse Sensor Working:

A pulse sensor or any optical heart-rate sensor, for that matter, works by shining a green light (~ 550 nm) on the finger and measuring the amount of reflected light using a photosensor. This method of pulse detection through light is called Photoplethysmogram.





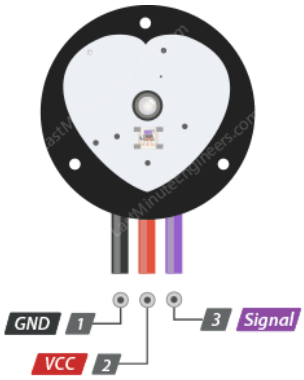
The oxygenated haemoglobin in the arterial blood has the characteristic of absorbing green light. The redder the blood (the higher the haemoglobin), the more green light is absorbed. As the blood is pumped through the finger with each heartbeat, the amount of reflected light changes, creating a changing waveform at the output of the photosensor.

As we continue to shine light and take photo sensor readings, we quickly start to get a heart-beat pulse reading.

This signal from the photosensor is generally small and noisy, therefore the signal is passed through an R/C filter network and then amplified using an Op Amp to create a signal that is much larger, cleaner and easier to detect.

Pulse Sensor Pinout:

The sensor comes with a 24” flat ribbon cable with 3 male header connectors. The following diagram shows the pinout.



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.

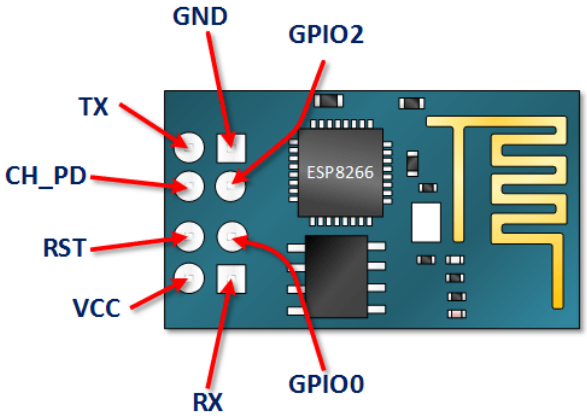
Wiring Pulse Sensor with Arduino:

The module can be powered from 3.3 or 5V. The positive voltage connects to ‘+’ and ground connects to ‘-‘. The 3rd ‘S’ wire is the analog signal output from the sensor and this will connect to the A0 analog input of an Arduino.

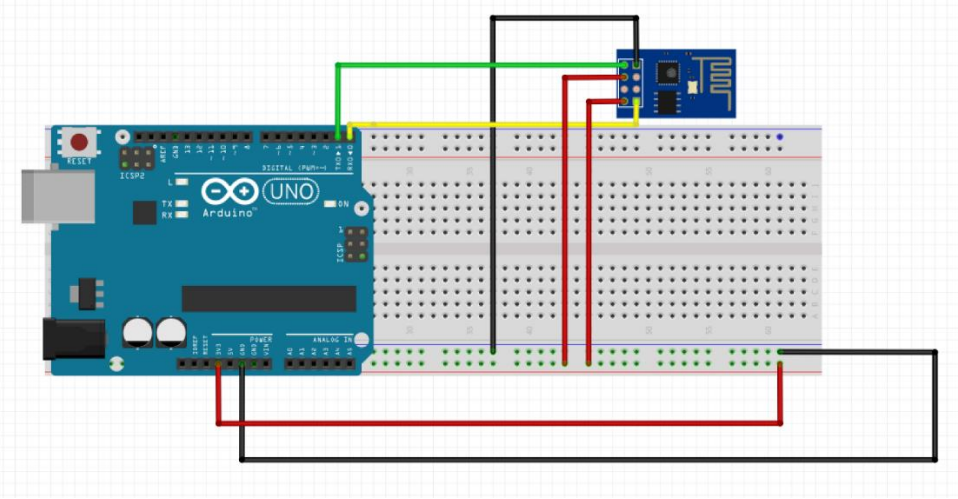
ESP8266

The ESP8266 is a low-cost Wi-Fi microchip, with a full TCP/IP stack and microcontroller capability. We can use ESP8266 in the projects for communication. It has 3 different modes. Now we will connect ESP8266 to Arduino Uno R3 and then we will set up our microchip with AT commands.

First, I did use ESP8266 and Arduino UNO R3. ESP8266 can be used in 2 different styles. In the first option we can use it as a receiver or transmitter. Second option is using ESP8266 as a microcontroller.



We will use just 5 of these 8 pins. RX and TX are our transmitter and receiver pins. We'll use them for serial communication. Last 3 pins are about the power system of ESP8266 which are GND, VCC, and CH_PD (CH_EN). As we know that GND and VCC pins are ground and voltage pins. We will use CH_PD (CH_EN) pins for enabling our microchip. Also, we can understand the right direction of the microchip by using the golden line at the right side of it.



We have to use a serial monitor at 115200 baud rate because this rate is the default option for ESP8266. Secondly, in the line options we should use Both NL & CR. If we do these steps correctly, we are ready to start to set up.

As the first step of setup, we should type "AT" in the serial monitor screen. Then we should get an "OK" response from this screen. If we get this, we will have an available ESP8266. Then we will type "AT+GMR" command to reach our version information.

Basic	
Command	**Description**
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT commands echo or not
AT+RESTORE	Factory Reset
AT+UART	UART configuration, [@deprecated]
AT+UART_CUR	UART current configuration
AT+UART_DEF	UART default configuration, save to flash
AT+SLEEP	Sleep mode
AT+RFPOWER	Set maximum value of RF TX Power
AT+RFVDD	Set RF TX Power according to VDD33

AT Commands Table for **ESP8266**

We can see that we all work at 115200 baud rate. This baud rate is too fast for us and we might want to change it to 9600 which is the optimum baud rate that is used by other components. For this change we will use the "AT+UART_DEF" command which can be seen on the table. But we will have some changes on this command. This command changes default UART configurations of our device. So, we should give some configuration parameters. "AT+UART_DEF=9600,8,1,0,0" we will use this command to change baud rate of ESP8266.

Commands	Description	Type
AT+RST	restart module	basic
AT+CWMODE	wifi mode	wifi
AT+CWJAP	join AP	wifi
AT+CWLAP	list AP	wif
AT+CWQAP	quit AP	wifi
AT+CIPSTATUS	get status	TCP/IP
AT+CIPSTART	set up TCP or UDP	TCP/IP
AT+CIPSEND	send data	TCP/IP
AT+CIPCLOSE	close TCP or UDP	TCP/IP
AT+CIFSR	get IP	TCP/IP
AT+CIPMUX	set multiple connections	TCP/IP
AT+CIPSERVER	set as server	TCP/IP

AT Commands for WiFi Configuration

The last step of the ESP8266 setup is about WiFi connection. In the table we can see AT commands for WiFi configuration. Usage of ESP8266 we need to know these commands for WiFi communication. The first command "AT+RST" is resetting our device. It is not changing settings, just reset the device.

"AT+CWMODE" this command is allowing us to change our device's usage mode. This command takes 1 parameter which should be integer.

"AT+CWMODE=mode" mode is representing the parameters and it looks like:

Modes:

1- Station mode (client)

2- AP mode (host)

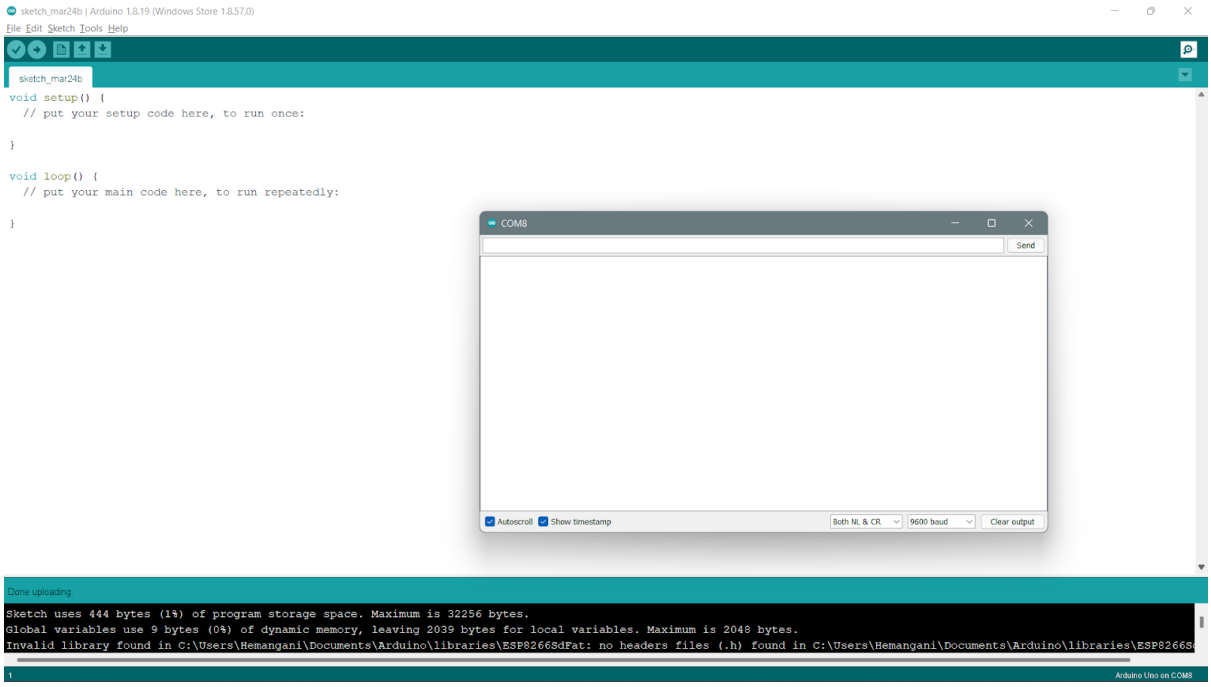
3- AP and Station mode

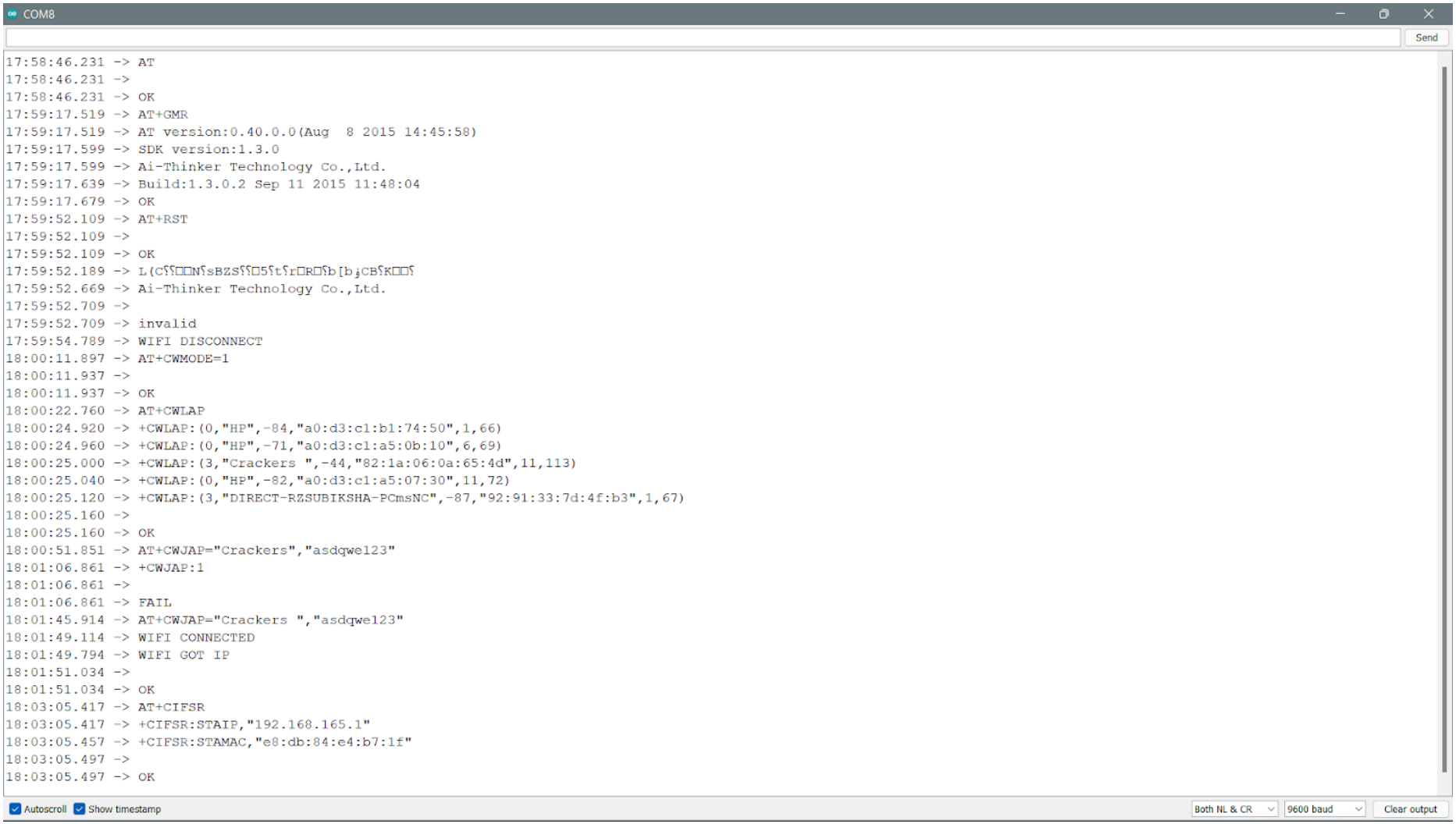
That means if we want to use ESP8266 as a client of our system, we should use the "AT+CWMODE=1" command for it.

Third command is about WiFi spots. With this command we can scan WiFi spots which are located nearby. "AT+CWLAP" can list all available WiFi spots for us.

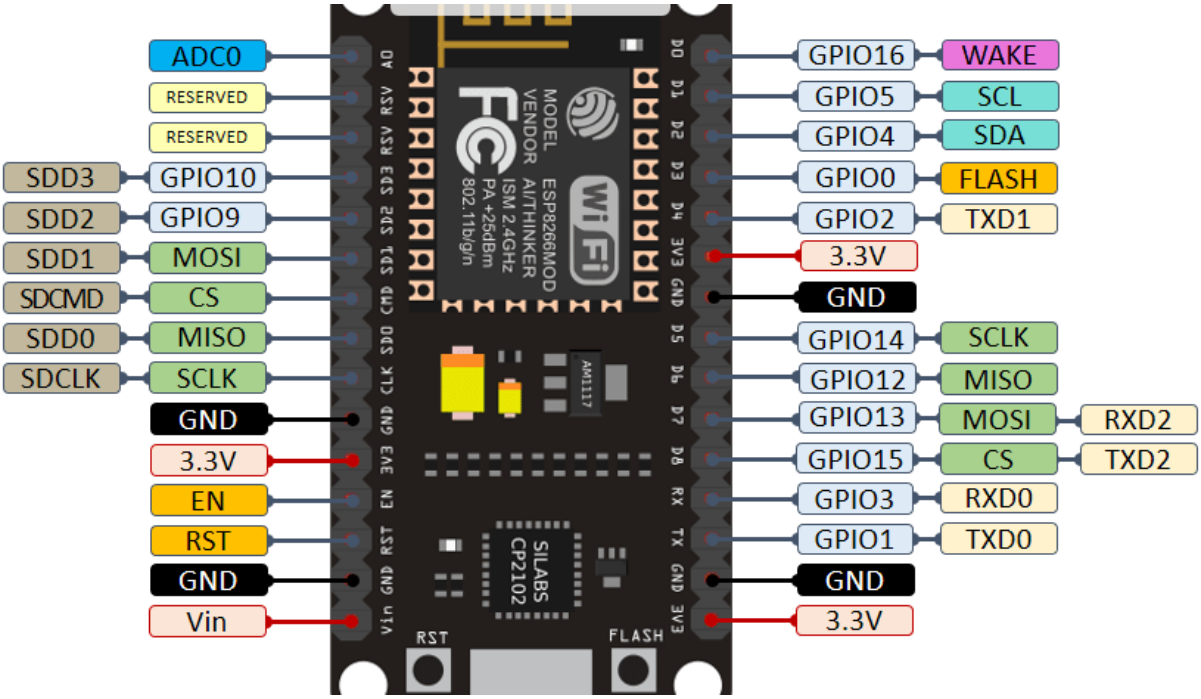
The next step of setup is joining a WiFi. I did set my device's mode as station mode then I listed all available WiFi spots. Now we will join WiFi without password. For this step we will use the "AT+CWJAP" command. But we know that for joining a WiFi we need SSID and Password. We can see all available SSIDs when we scan WiFi. Now we should change this command to "AT+CWJAP="YOUR_SSID_NAME","YOUR_PASSWORD".

We will use the "AT+CIFSR" command to find our IP. We can use this IP to communicate to our device from another system. Also, this command returns us the MAC Address of ESP8266.



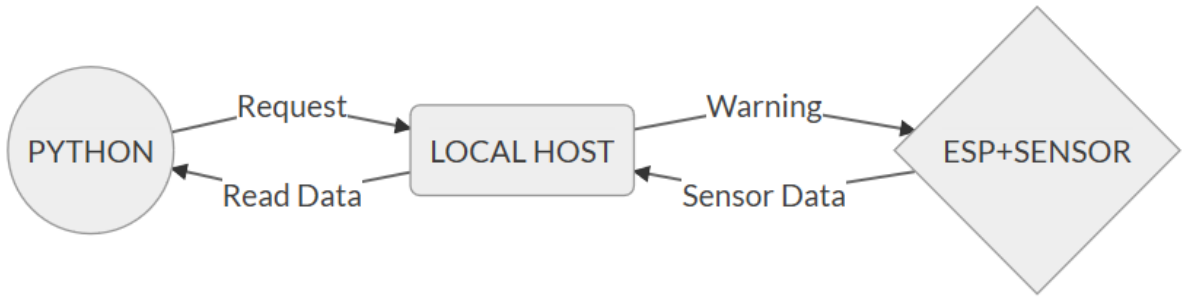

COM8
Send
17:58:46.231 -> AT
17:58:46.231 ->
17:58:46.231 -> OK
17:59:17.519 -> AT+GMR
17:59:17.519 -> AT version:0.40.0.0(Aug 8 2015 14:45:58)
17:59:17.599 -> SDK version:1.3.0
17:59:17.599 -> Ai-Thinker Technology Co.,Ltd.
17:59:17.639 -> Build:1.3.0.2 Sep 11 2015 11:48:04
17:59:17.679 -> OK
17:59:52.109 -> AT+RST
17:59:52.109 ->
17:59:52.109 -> OK
17:59:52.189 -> L(C§§□□N§sBZS§§□5§t§r□R□§b[b¿CB§K□□§
17:59:52.669 -> Ai-Thinker Technology Co.,Ltd.
17:59:52.709 ->
17:59:52.709 -> invalid
17:59:54.789 -> WIFI DISCONNECT
18:00:11.897 -> AT+CWMODE=1
18:00:11.937 ->
18:00:11.937 -> OK
18:00:22.760 -> AT+CWLAP
18:00:24.920 -> +CWLAP:(0,"HP",-84,"a0:d3:c1:b1:74:50",1,66)
18:00:24.960 -> +CWLAP:(0,"HP",-71,"a0:d3:c1:a5:0b:10",6,69)
18:00:25.000 -> +CWLAP:(3,"Crackers ",-44,"82:1a:06:0a:65:4d",11,113)
18:00:25.040 -> +CWLAP:(0,"HP",-82,"a0:d3:c1:a5:07:30",11,72)
18:00:25.120 -> +CWLAP:(3,"DIRECT-RZSUBIKSHA-PCmsNC",-87,"92:91:33:7d:4f:b3",1,67)
18:00:25.160 ->
18:00:25.160 -> OK
18:00:51.851 -> AT+CWJAP="Crackers","asdqwe123"
18:01:06.861 -> +CWJAP:1
18:01:06.861 ->
18:01:06.861 -> FAIL
18:01:45.914 -> AT+CWJAP="Crackers ","asdqwe123"
18:01:49.114 -> WIFI CONNECTED
18:01:49.794 -> WIFI GOT IP
18:01:51.034 ->
18:01:51.034 -> OK
18:03:05.417 -> AT+CIFSR
18:03:05.417 -> +CIFSR:STAIP,"192.168.165.1"
18:03:05.457 -> +CIFSR:STAMAC,"e8:db:84:e4:b7:1f"
18:03:05.497 ->
18:03:05.497 -> OK
Autoscroll
Show timestamp
Both NL & CR
9600 baud
Clear output

Installing ESP8266 Into Arduino IDE And, Python Communication



ESP8266 12-E NodeMCU Pinout Diagram

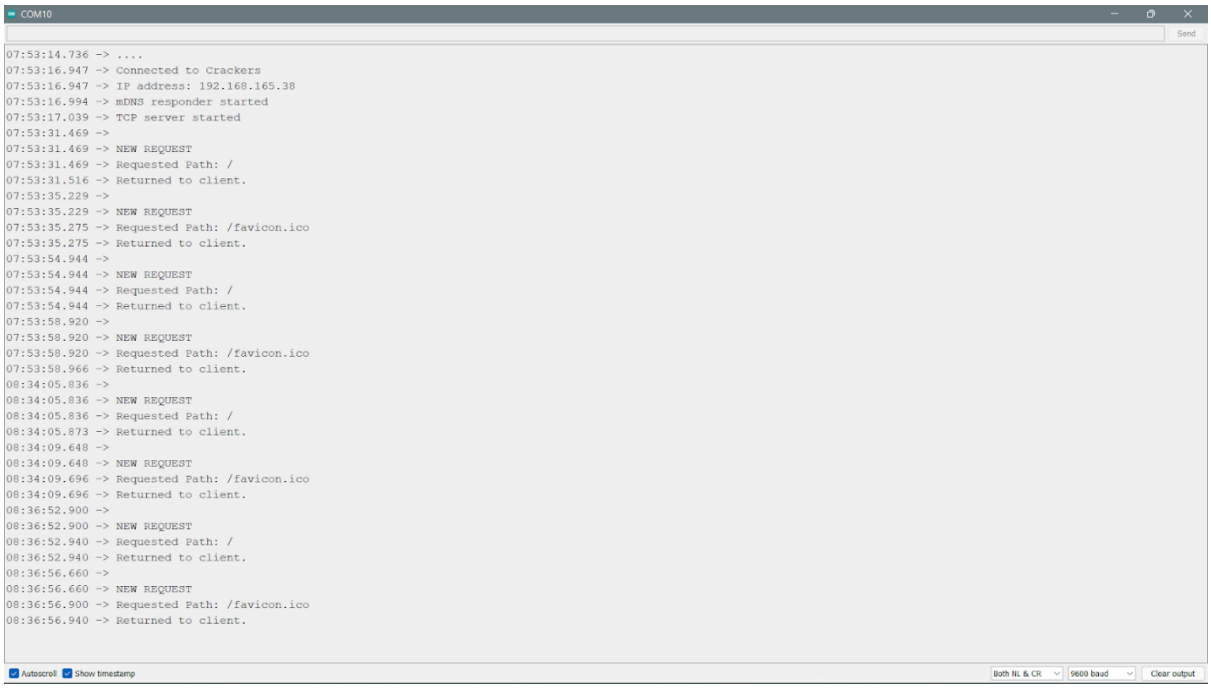
Created a simple mDNS communication system. Our esp connects to our wifi and creates a localhost server and starts to wait for a request. Every time our python sends a request to that localhost, esp runs the desired code and then returns the result as an http request. Finally, python reads that returned data as http request and grab those variables from it. With this, esp can return strings, datas and arrays. Python code will understand their datatype.



While code is being uploaded, we can open the serial port, the details are printed when uploading is done. Learn the IP of esp and note that. ESP's IP on local; changes by wifi to wifi, not session to session, so when we close and open it later, it will not be changed.

We can also view the raw data with a browser while pasting the ip of ESP on a browser. Or we can make an application to read it, or can even use another language.

Serial monitor window



Browser window

HEART RATE = 82, BP sys = 121, BP dia = 79, SPO2 = 96.8, TEMPERATURE = 37.2C, BODY POSTURE = UPRIGHT

Measure of SpO2, Heart Rate and BP Trends (BPT)

High physiological stress and high altitudes can be a reason for varying levels of oxygen. The human body is generally capable of adapting itself to such extreme conditions but hypoxemia is always a possibility.

This is a unique and easy way to measure PPG and calculate spO2, heart rate and Blood Pressure trending (BPT) with surprisingly high accuracy.

There are several Pulse boards available with different form factors and applications. ProtoCentral also provides the popular MAX30102-based breakout board in a variety of versions, but Pulse Express stands out by integrating the MAX32664D Biometric sensor hub.

Typically, PPG is the output from the optical sensors and it is up to the user to calculate other vitals such as Heartrate, spO2 etc, but the MAX32664D sensor hub does all the vitals calculations and gives us the final output.

- Built in the shape and size of a finger, it is ideally suited to measuring vitals with a Velcro strap hole to connect the finger on board.
- Internal algorithm for measuring Pulse Heart Rate, Pulse Blood Oxygen Saturation (SpO2), Estimated Blood Pressure trending.
- Integrates a high-sensitivity pulse oximeter and heart rate sensor (MAX30102) and biometric sensor hub (MAX32664D).
- In-built accelerometer for robust detection and compensation of motion artifacts.

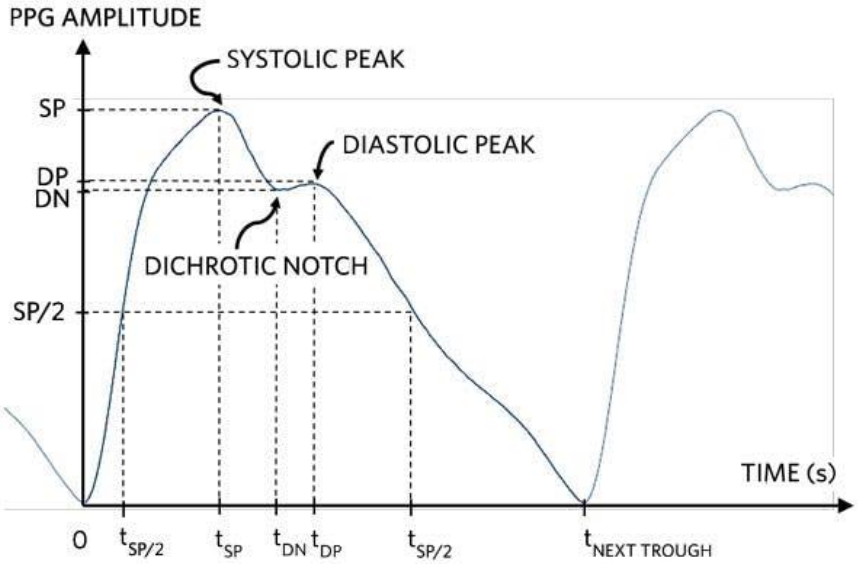
Pulse oximetry (percentage of SpO2 concentration in blood) has been used as a key health indicator for many decades. Although the original academic development was made in 1935, the modern basis for determining the SpO2 concentration using light sources and photosensor(s) was developed by Takuo Aoyagi and Michio Kishi in 1972.

When commercially feasible, SpO2 concentration measurement devices have made huge gains in medical applications. Since 1987, the Standard of Care (SoC) for the administration of a general anaesthetic has included pulse oximetry. All modern hospital bedside equipment include an SpO2 module based on the same fundamentals, albeit with minor modifications.

Pulse oximetry is used to measure the level of oxygen (oxygen saturation) in the blood. It's a simple, painless measure of how well oxygen is being sent out from our heart to parts of our body, such as our arms and legs. It can be used to check if there is enough oxygen in the blood and to check the health of a person with any condition that affects blood oxygen levels.

Blood Pressure Trend is NOT the same as absolute blood pressure measurement. BPT uses an algorithm to look at changes in the shape of the PPG signal and correlate them to changes in BP from a given calibrated baseline BP. This is still useful because it is not possible to take continuous BP recordings from traditional cuff-based sphygmomanometer devices.

Changes in BP or the BP trend (BPT) prove to be valuable in cardiovascular care and monitoring of high-risk patients.



Protocentral Pulse Express comes with all the basic things for us to get started with reading PPG signal and get basic measurements It consists of a MAX32664-D Biometric Sensor Hub along with a MAX30102 pulse sensor and logic level converters. We simply have to hook it up with Arduino as shown in the following table.

Max32664 pin label	Arduino Nano Connection	Pin Function
SDA	A4	Serial Data
SCL	A5	Serial Clock
Vin	3.3V	Power
GND	Gnd	Gnd
MFIO Pin	D5	MFIO
RESET Pin	D4	Reset

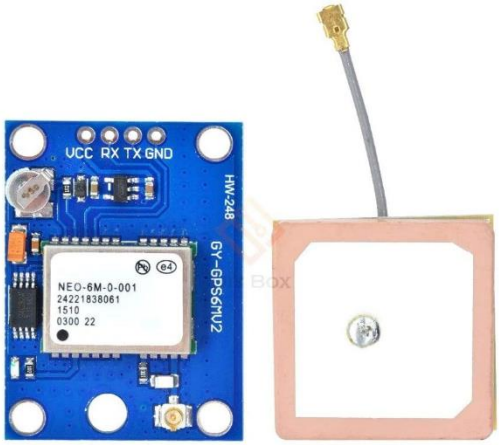
We have designed the ProtoCentral Pulse Express with integrated high-sensitivity optical sensors (MAX30102) and MAX32664D biometric sensor hub to read PPG signals and perform finger based heart rate, spo2 and blood oxygen saturation measurements.

The board is connected to the MC using a standard I2C interface. Using the ProtoCentral Pulse Express Arduino Library uploaded to the NANO interfaced with the Pulse Express Pulse Oximeter and the Heart Rate Monitor makes it much easier to read the PPG and the measured SPO2, the heart rate and the estimated BP. The output is seen on the serial monitor and the serial plotter.

This code configures the sensor in algorithm mode to enable the sensor to start calculating the Spo2, BP systolic and diastolic and HR values.

The board will start in calibration mode once we upload the library, we will have to keep our finger on the sensor until the calibration progress reaches 100%. The board will switch to estimation mode once the calibration process is completed. It takes 10 to 20 seconds for the algorithm to gather enough samples to give good values.

NEO-6M GPS Module



Description:
This GPS module uses the latest technology to give the best possible position information, allowing for better performance. The NEO-6M GPS module is shown in the figure below. It comes with an external antenna and does not come with header pins. The heart of the module is a NEO-6M GPS chip from u-blox. It can track up to 22 satellites on 50 channels and achieves the industry's highest level of sensitivity i.e. -161 dB tracking, while consuming only 45mA supply current. The u-blox 6 positioning engine also boasts a Time-To-First-Fix (TTFF) of under 1 second. One of the best features the chip provides is Power Save Mode(PSM). It allows a reduction in system power consumption by selectively switching parts of the receiver ON and OFF. This dramatically reduces power consumption of the module to just 11mA making it suitable for power sensitive applications like GPS wristwatch. The necessary data pins of NEO-6M GPS chip are broken out to a "0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART.

Product Dimensions: 16 x 12.2 x 2.4 mm
Hardware Overview Technical specifications:

• Standalone GPS receiver

• Under 1 second time-to-first-fix for hot and aided starts

• SuperSense ® Indoor GPS: -162 dBm tracking sensitivity

• Anti-jamming technology

• Support SBAS (WAAS, EGNOS, MSAS, GAGAN)

• U-blox 6 50 channel positioning engine with over 2 million effective correlators

• 5Hz position update rate

• Operating temperature range: -40 TO 85°C

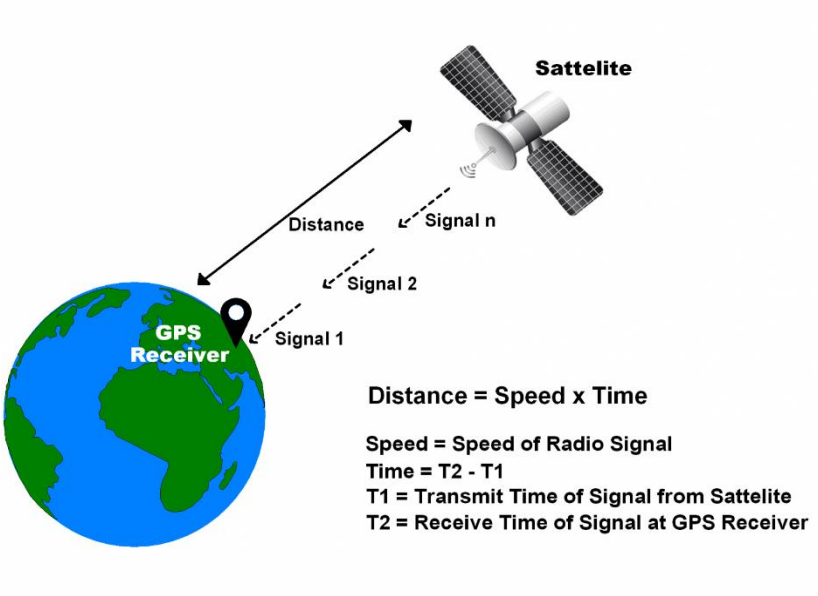
• UART TTL socket

• EEprom to store settings

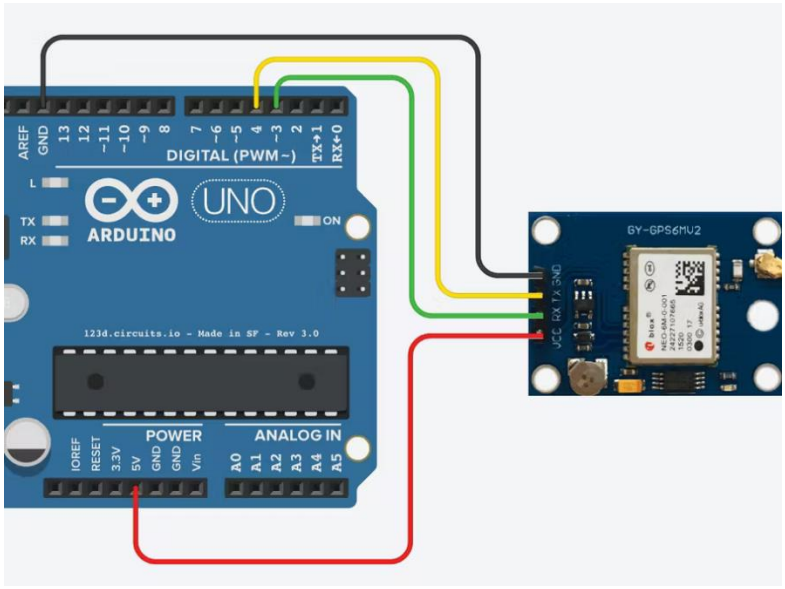
• Build in 18X18mm GPS antenna

GPS sensor working:

The Global Positioning System (GPS) is a satellite-based navigation system made up of at least 24 satellites. GPS works in any weather conditions, anywhere in the world, 24 hours a day, with no subscription fees or setup charges.GPS satellites circle the Earth twice a day in a precise orbit. Each satellite transmits a unique signal and orbital parameters that allow GPS devices to decode and compute the precise location of the satellite. GPS receivers use this information and trilateration to calculate a user's exact location. Essentially, the GPS receiver measures the distance to each satellite by the amount of time it takes to receive a transmitted signal. With distance measurements from a few more satellites, the receiver can determine a user's position and display it. To calculate your 2-D position (latitude and longitude) and track movement, a GPS receiver must be locked on to the signal of at least 3 satellites. With 4 or more satellites in view, the receiver can determine your 3-D position (latitude, longitude and altitude). Generally, a GPS receiver will track 8 or more satellites, but that depends on the time of day and where you are on the earth.



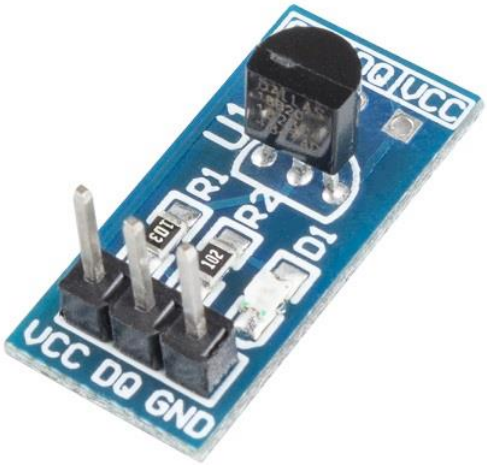
Arduino interfacing:



The NEO-6M GPS module has four pins: VCC, RX, TX, and GND. The module communicates with the Arduino via serial communication using the TX and RX pins

- The module GND pin is connected to Arduino GND pin
- The module RX pin is connected to Arduino pin 3
- The module TX pin is connected to Arduino pin 4
- The module VCC pin is connected to Arduino 5V pin

DS18B20 Temperature Sensor Module



Description:

DS18B20 is 1-Wire interface Temperature sensor manufactured by Dallas Semiconductor Corp. The unique 1-Wire® Interface requires only one digital pin for two way communication with a microcontroller. The sensor comes usually in two form factors. One that comes in TO-92 package looks exactly like an ordinary transistor. Other one in a waterproof probe style which can be more useful when you need to measure something far away, underwater or under the ground

Product Dimensions: 21 x10 mm
Hardware Overview:

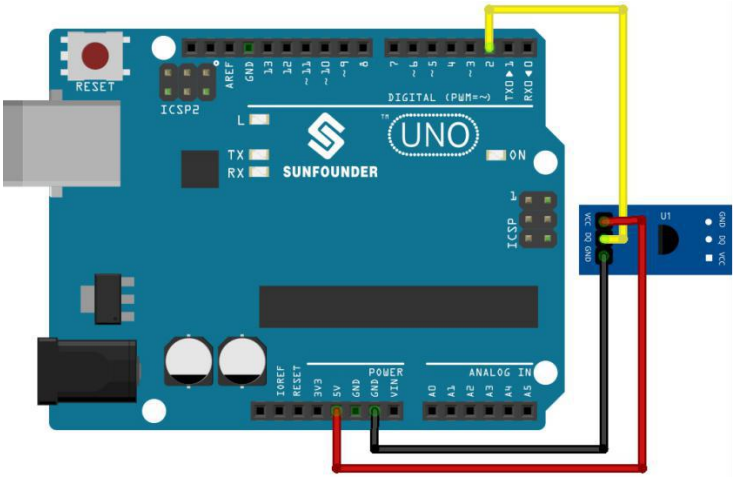
DS18B20 digital temperature sensor works on a single bus and it has 64-bit ROM to store the serial number of component. It can get quite a few DS18B20 sensors connected to a single bus in parallel. With a microcontroller, you can control so many DS18B20 sensors that are distributed around a wide range. So the sensor is widely used in these fields, including HVAC environmental control, temperature detection of building, instrument and machine, and process monitoring and control.

Technical specifications:

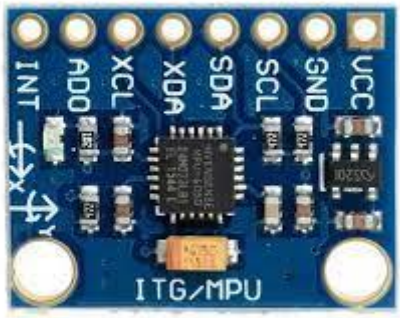
- Working Voltage: 3V~5.5V
- Detected Temperature Range: -55°C~+125°C (-67°F~+257°F) deviation ±2°C ,-10°C~+85°C deviation ±0.5°C

Arduino interfacing:

- VCC: 3.3V-5V working voltage
- DQ: data input/ output pin
- GND: ground



MPU6050 Accelerometer and Gyroscope Sensor



Description:

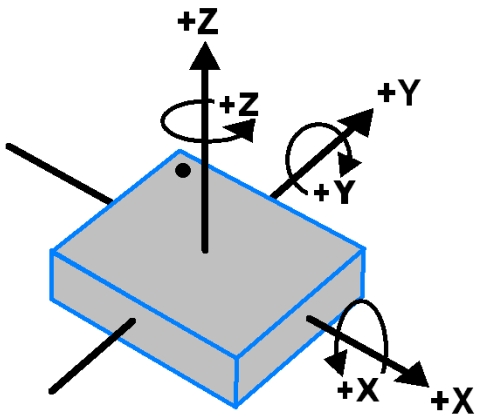
MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package. It has I2C bus interface to communicate with the microcontrollers.

Product Dimensions: 3 × 2 × .2 cm

Hardware Overview:

Technical specifications:
Chip built-in 16bit AD converter, 16-bit data output
Driver Chip: MPU6050
Operating Voltage: 3-5V DC
Communication: I2C/IIC Protocol
Gyro Range: ± 250, 500, 1000, 2000 °/s
Accelerometer Range: ± 2 ± 4 ± 8 ± 16 g

Working:

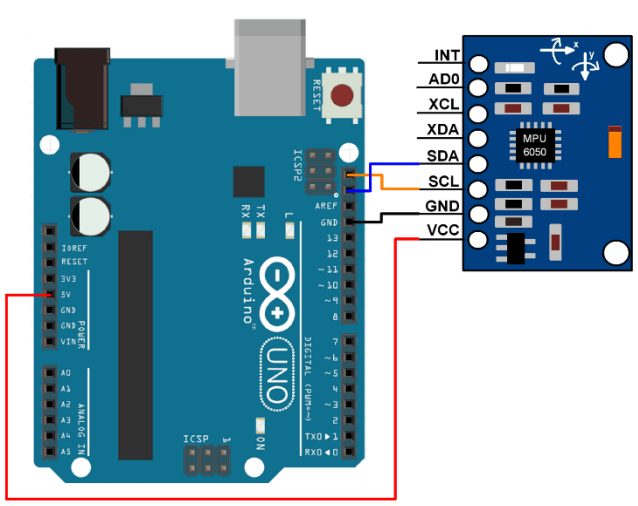


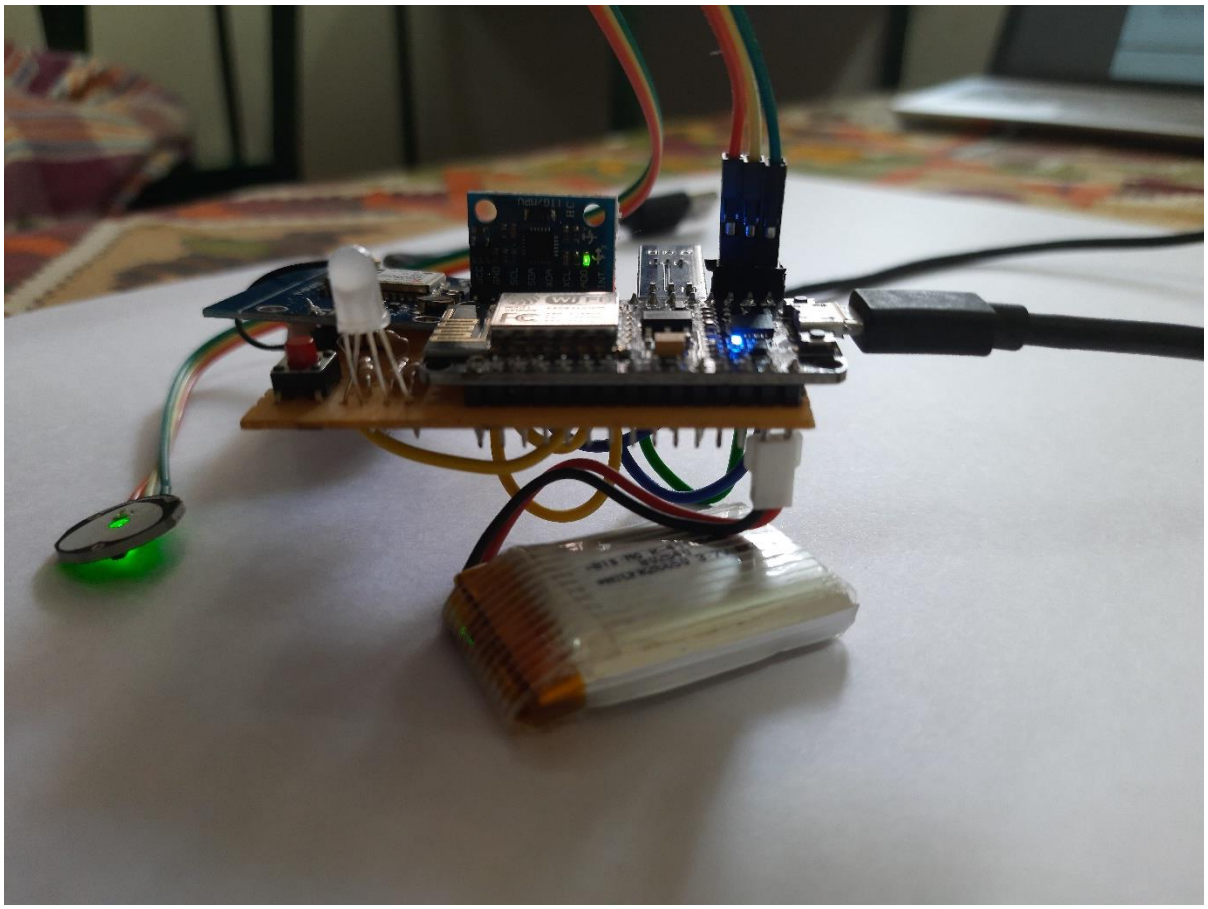
MPU-6050
Orientation & Polarity of Rotation

- When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.
- The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate.
- This voltage is digitized using 16-bit ADC to sample each axis.
- The full-scale range of output are +/- 250, +/- 500, +/- 1000, +/- 2000.
- It measures the angular velocity along each axis in degree per second unit.

Arduino interfacing:

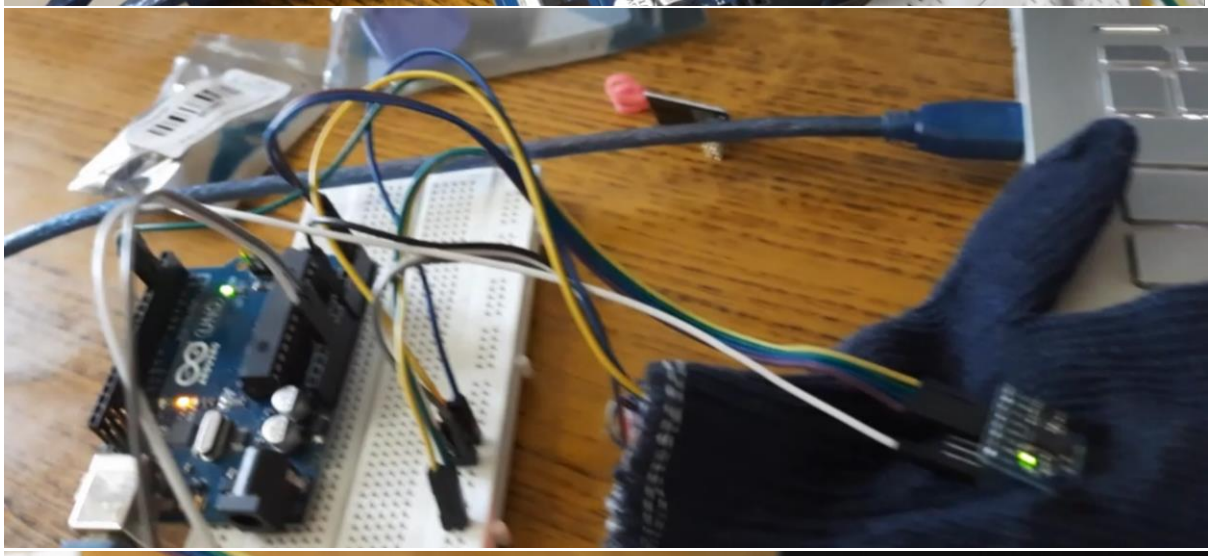
The MPU-6050 module has 8 pins,
INT: Interrupt digital output pin.
AD0: I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.
XCL: Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.
XDA: Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.
SCL: Serial Clock pin. Connect this pin to microcontrollers SCL pin.
SDA: Serial Data pin. Connect this pin to microcontrollers SDA pin.
GND: Ground pin. Connect this pin to ground connection.
VCC: Power supply pin. Connect this pin to +5V DC supply.


INT
AD0
XCL
XDA
SDA
SCL
GND
VCC
MPU
6050
Arduino
UNO

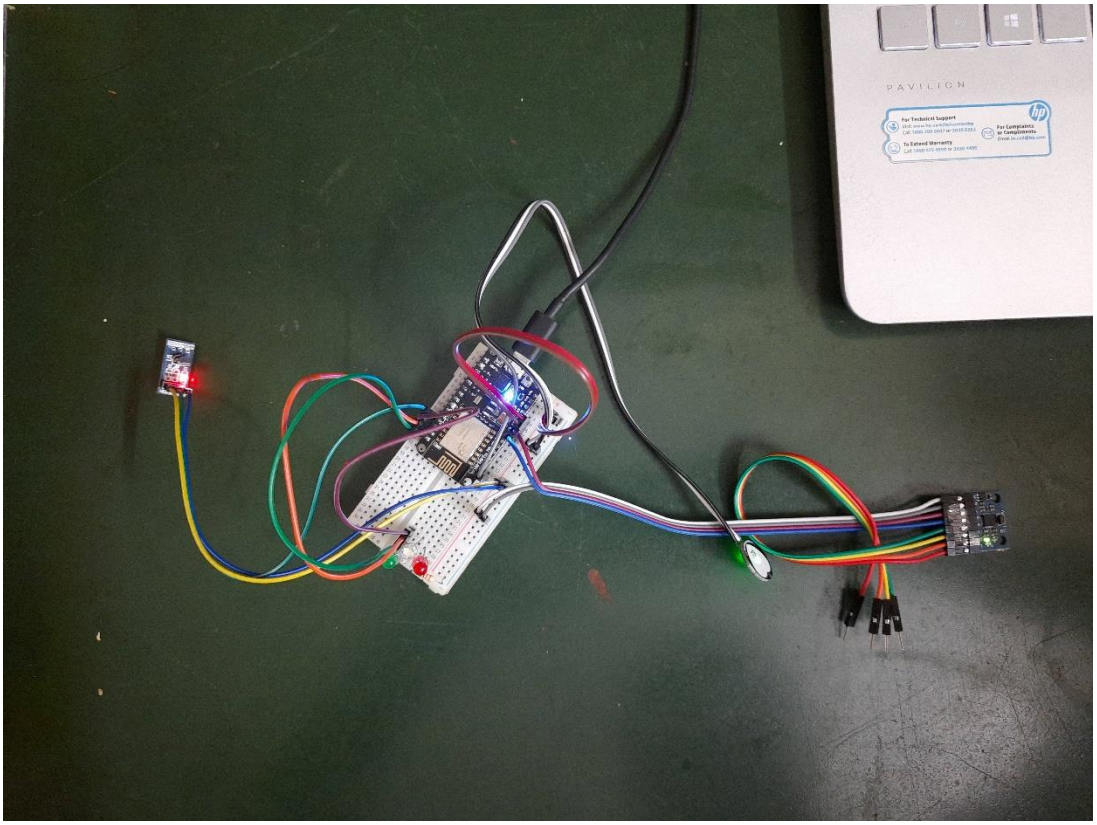


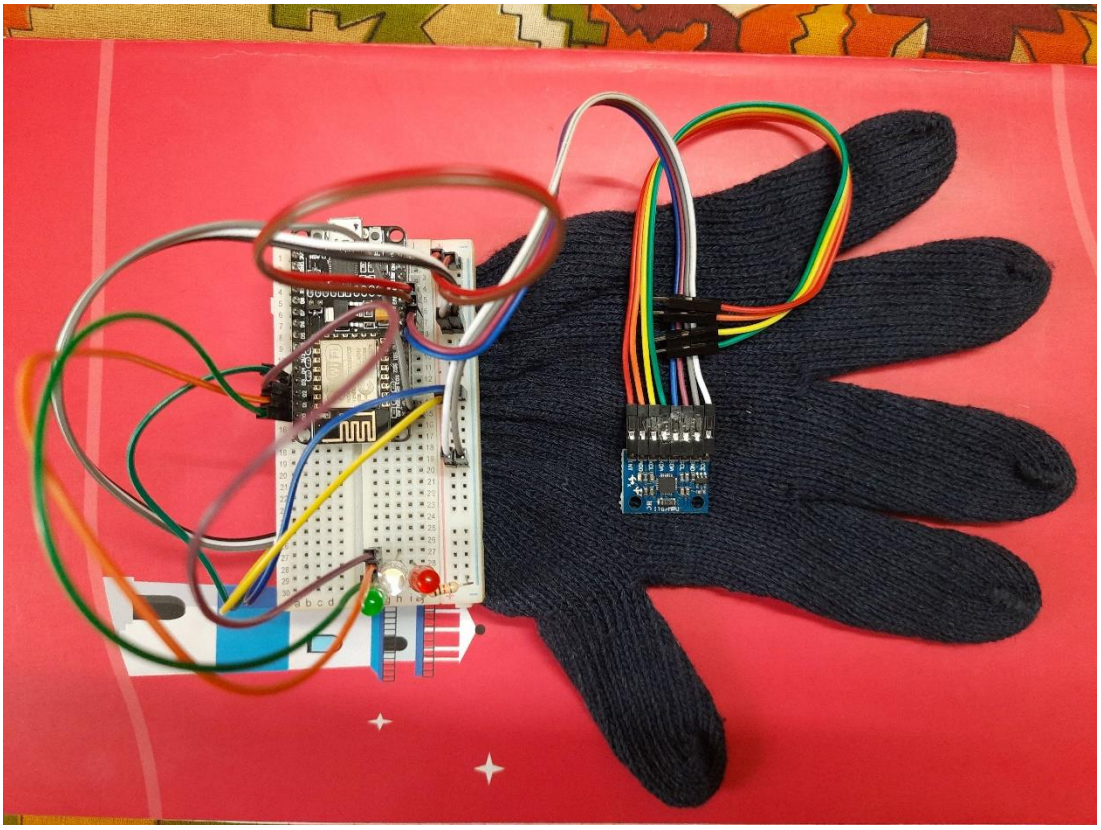
Circuit Images

Review 1

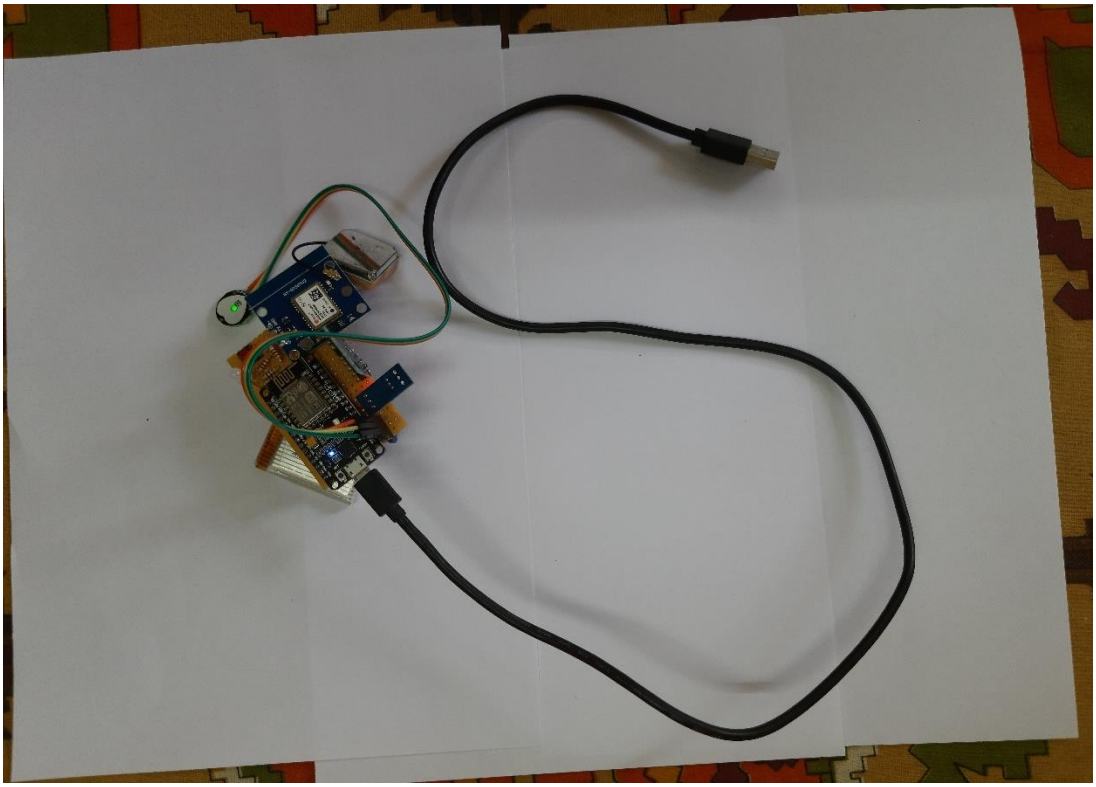


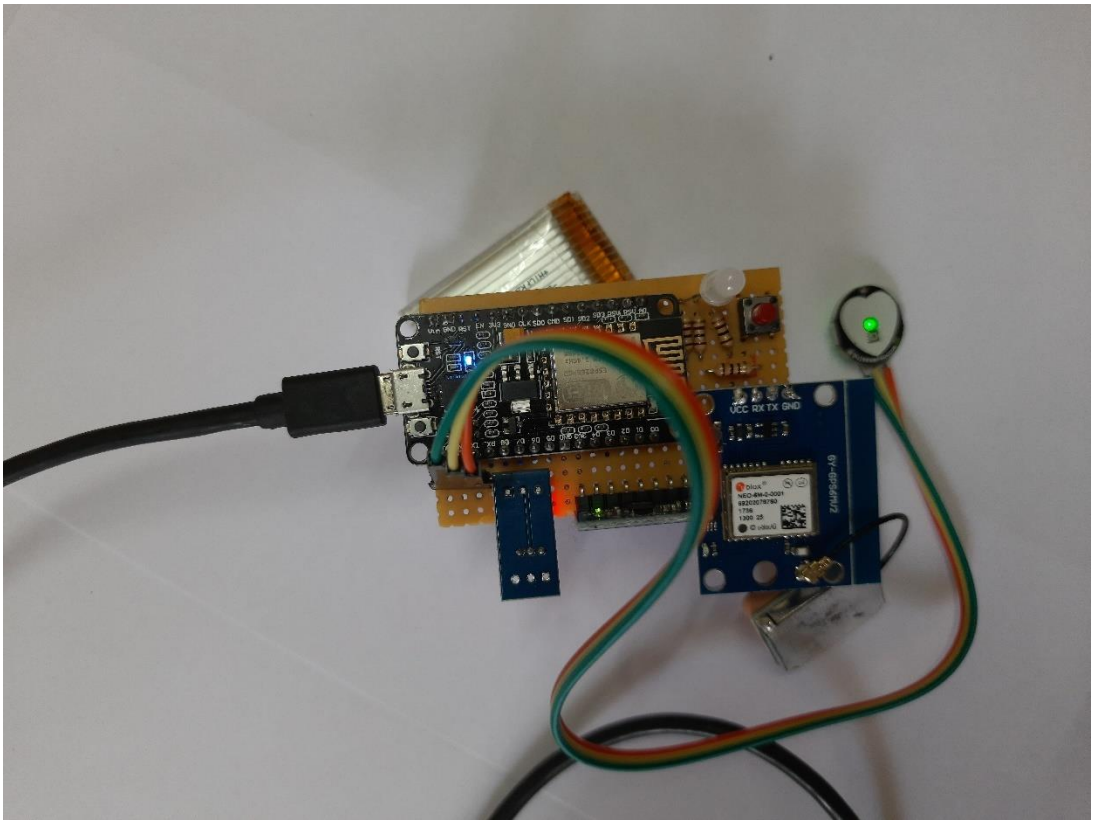
Review 2



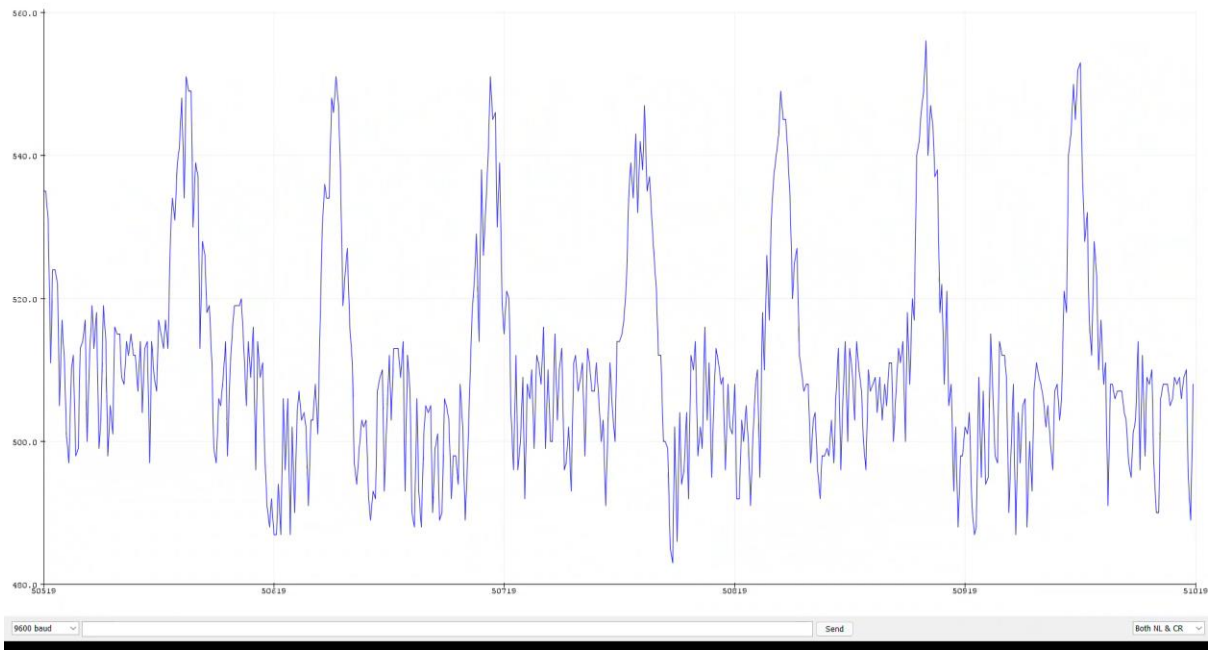


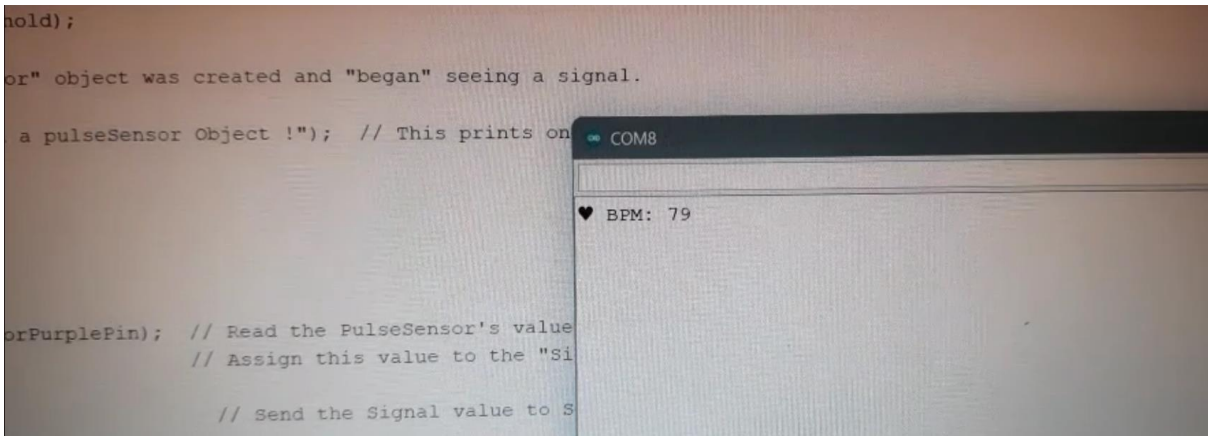
Final Review



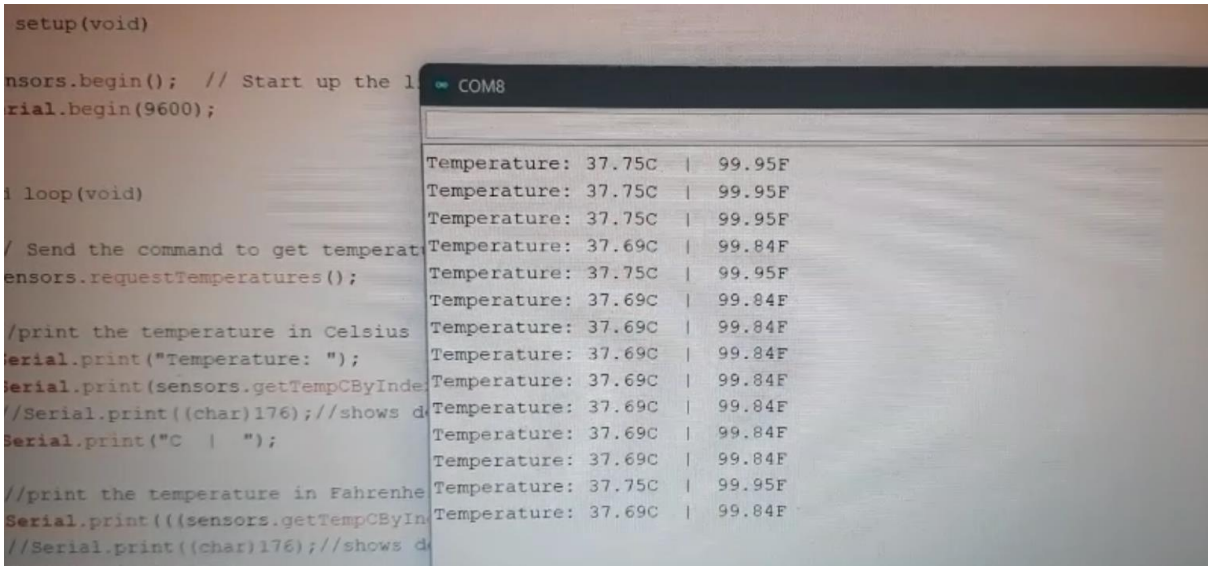


Heart rate sensor output:



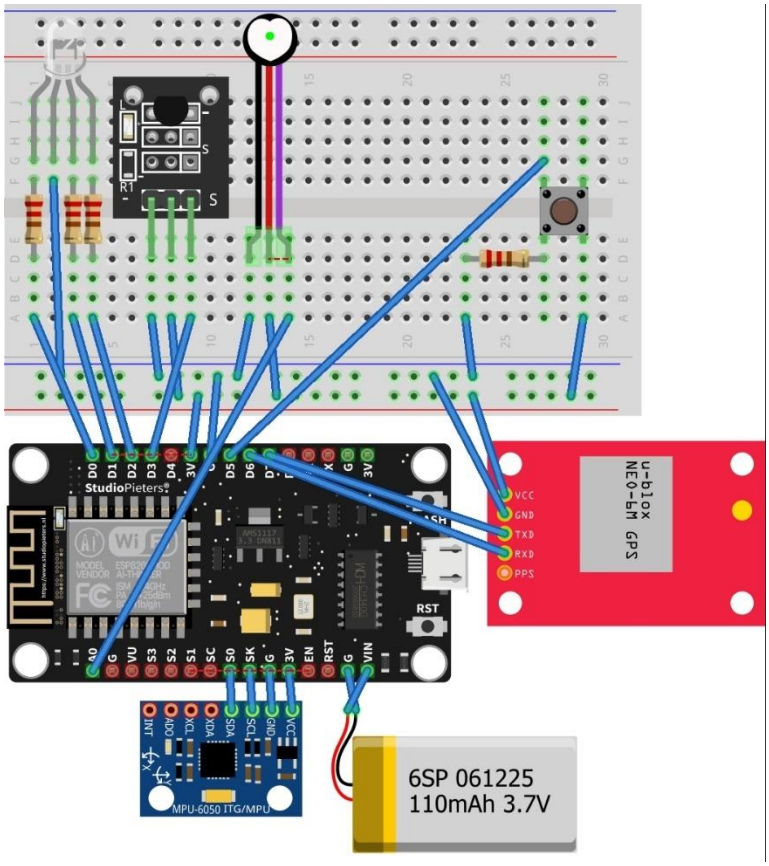


Temperature sensor output:

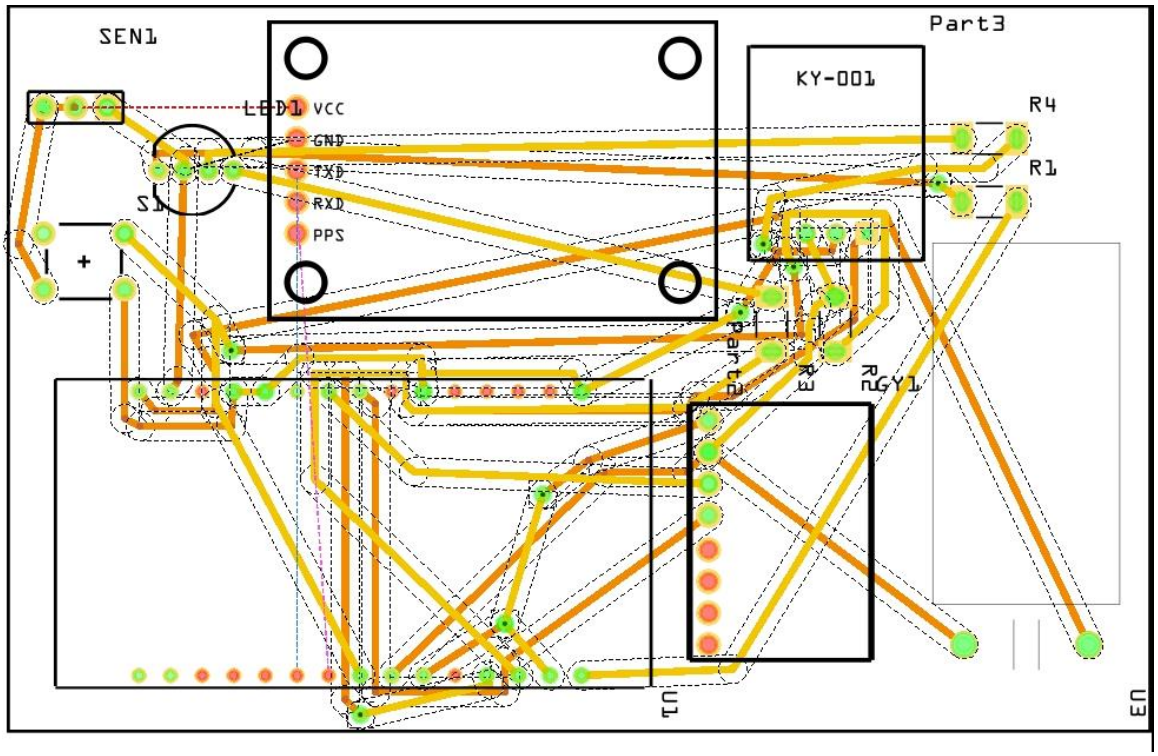


PCB DESIGN

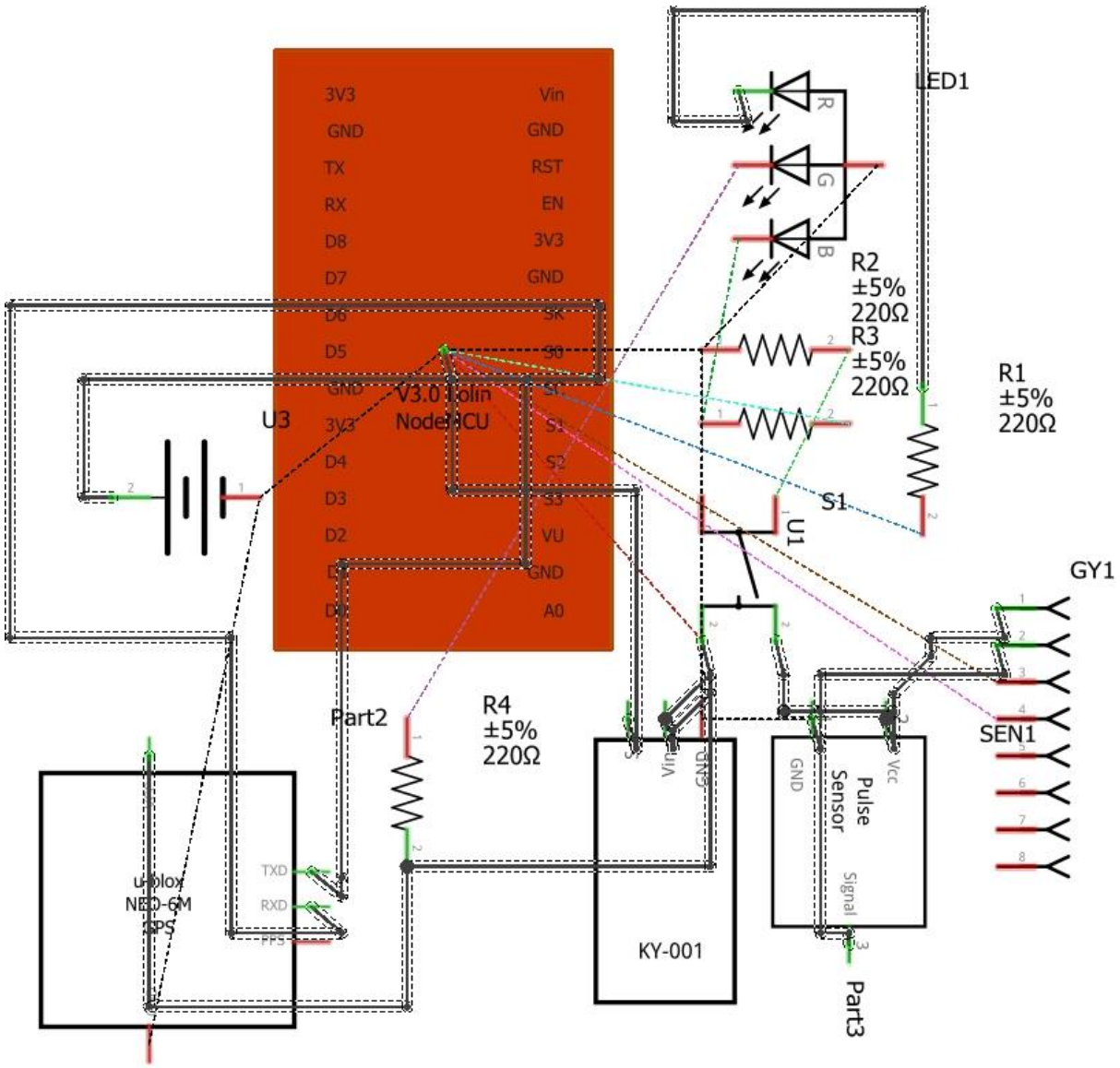
Bread board



PCB



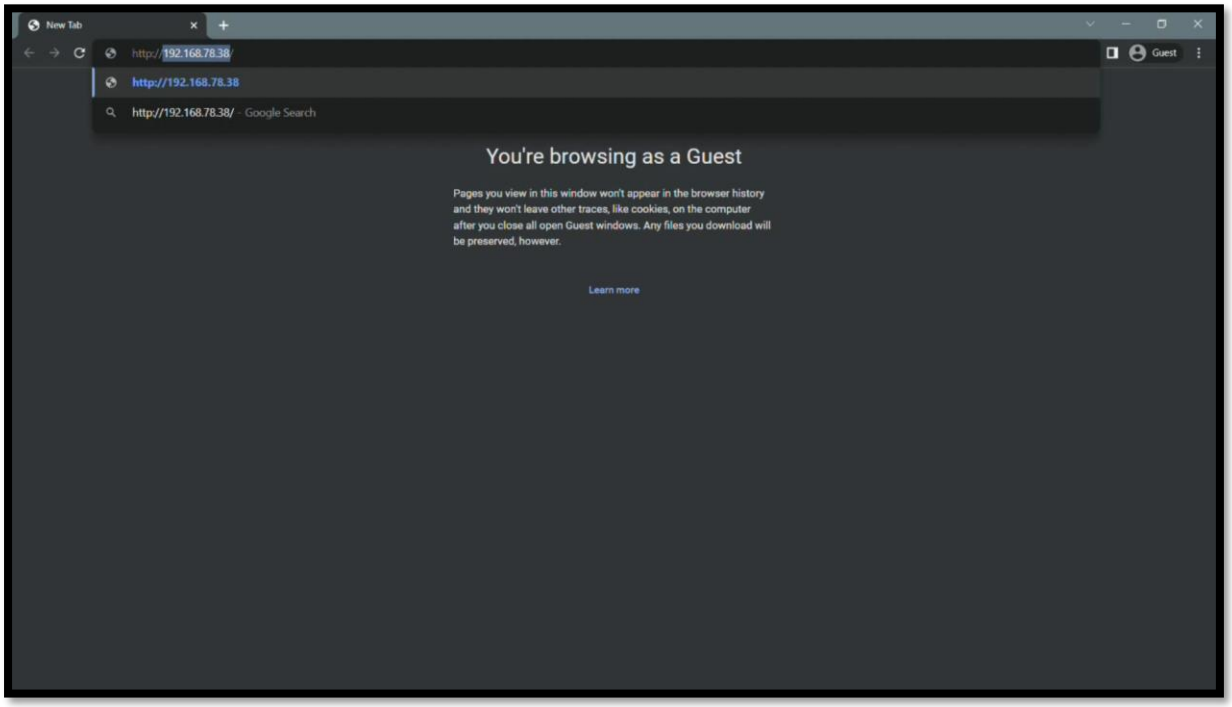
SCHEME

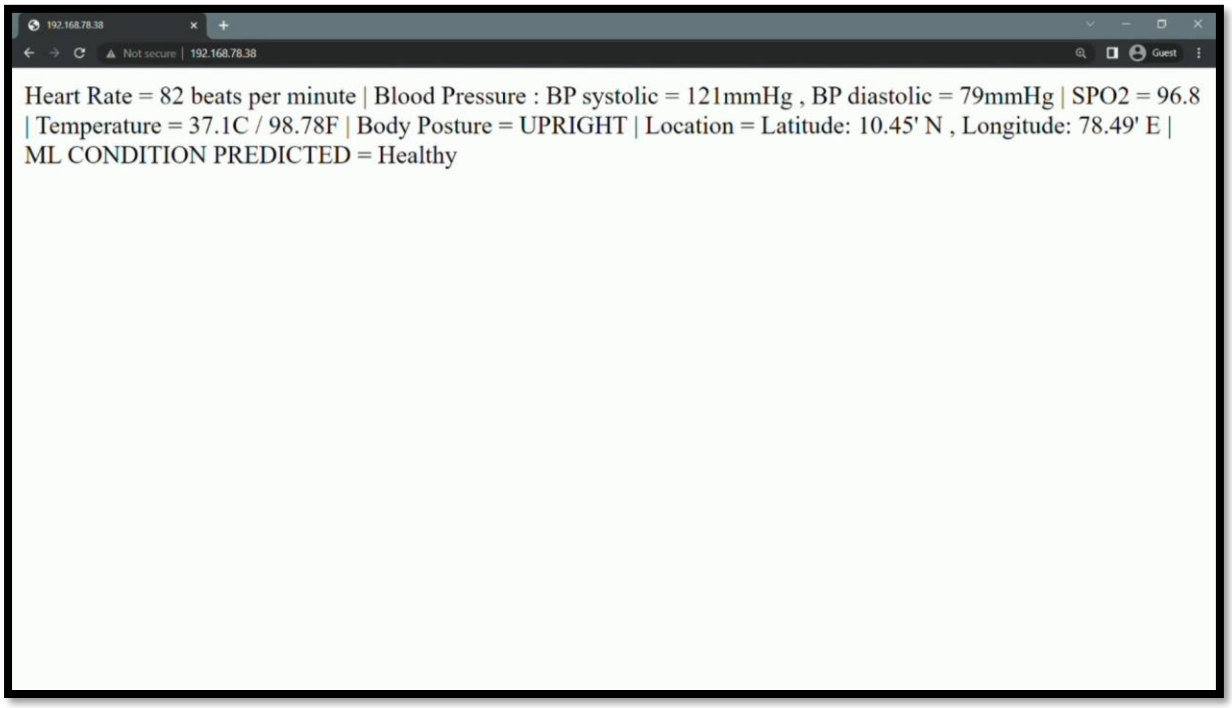

3V3
GND
TX
RX
D8
D7
D5
GND
3V3
D4
D3
D2
Vin
GND
RST
EN
3V3
GND
S3
S2
S1
VU
GND
A0
V3.0 Lolin
NodeMCU
U3
LED1
R
G
B
R2
±5%
220Ω
R3
±5%
220Ω
R1
±5%
220Ω
S1
U1
GY1
SEN1
Part2
R4
±5%
220Ω
u-blox
NEO-6M
GPS
TXD
RXD
KY-001
Pulse
Sensor
Signal
Vcc
GND
Part3

Assembly List

Label	Part Type	Properties
GY1	MPU-6050 Board GY-521	row double; form ♀ (female); package THT; variant variant 1; pins 8; hole size 1.0mm,0.508mm; pin spacing 0.1in (2.54mm)
LED1	RGB LED - (4 legs)	package 5 mm [THT]; polarity common anode; rgb RGB; pin order rgb
Part2	u-blox NEO-6M GPS Breakout	variant variant 1
Part3	KY-001 Temperature Sensor Module	chip label KY-001; editable pin labels false; variant variant 1; package Breakout board; sensor DS18B20; dimensions 18.5mm x 15mm [0.728in x 0.591in]; pins 3; temperature range -55°C to 125°C [-57°F to 257°F]; operating voltage 3.0V to 5.5V; accuracy range ±0.5°C; pin spacing 300mil
R1	220Ω Resistor	resistance 220Ω; package 2010 [SMD]; tolerance ±5%
R2	220Ω Resistor	resistance 220Ω; package 2010 [SMD]; tolerance ±5%
R3	220Ω Resistor	resistance 220Ω; package 2010 [SMD]; tolerance ±5%
R4	220Ω Resistor	resistance 220Ω; package 2010 [SMD]; tolerance ±5%
S1	Pushbutton	package [THT]
SEN1	Pulse Sensor	variant variant 1; package SIP-3
U1	NodeMCU V3.0	variant variant 1; chip ESP8266
U3	LIPO-110mAh	variant 110mAh; package lipo-110

Results and conclusion:





In this project we have successfully designed and tested a wearable sensor glove used to monitor the vitals of the patient wirelessly from any location. If any unforeseen situation may come up the sensor data and the location of the patient is sent to the emergency contact of the patient or to any nearby hospital which helps timely rescuing of the patient.