A minor project report on :

HEALTHCARE MONITORING AND MANAGEMENT SYSTEM

submitted in partial fulfilment of the requirements for the degree of
B. Tech
In
Electronics and Electrical Engineering
By

Souvik Karmakar	**1807228**
Indrashis Mitra	**1807274**
Kinjal Sarkar	**1807277**
Pratyay Basu	**1807291**

under the guidance of
Prof. K.B. Ray

School of Electronics Engineering
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
(Deemed to be University)
BHUBANESWAR
April 2021

CERTIFICATE

This is to certify that the project report entitled **"HEALTHCARE MONITORING AND MANAGEMENT SYSTEM"** submitted by

Souvik Karmakar	**1807228**
Indrashis Mitra	**1807274**
Kinjal Sarkar	**1807277**
Pratyay Basu	**1807291**

in partial fulfilment of the requirements for the award of the **Degree of Bachelor of Technology** in **Electronics and Electrical Engineering** is a bonafide record of the work carried out under my(our) guidance and supervision at School of Electronics Engineering, KIIT (Deemed to be University).

Signature of Supervisor 1
Prof.K.B.Ray
School of Electronics Engineering
KIIT (Deemed to be University)

The Project was evaluated by us on _____________

EXAMINER 1 EXAMINER 2
EXAMINER 3 EXAMINER 4

ACKNOWLEDGEMENT

We feel immense pleasure and feel privileged in expressing our deepest and most sincere gratitude to our supervisor **Prof.K.B.Ray**, for her excellent guidance throughout our project work. Her kindness, dedication, hard work and attention to detail have been a great inspiration to us. Our heartfelt thanks to you mam for the unlimited support and patience shown to us. We would particularly like to thank her for all her help in patiently and carefully correcting all our manuscripts.

We are also very thankful to **Prof.Suman Roy, Prof. Pravat Biswal**
Prof. M. Ramana, FIC project (EEE), Associate Dean Prof. **Dr. Amlan Datta** and **Dr. Suprava Patnaik**, Dean (School Of Electronics) for their support and suggestions during our course of the project work in the final year of our undergraduate course.

STUDENT SIGNATURE :-

Roll Number	Name	Signature
1807228	**Souvik Karmakar**	Souvik Karmakar
1807274	**Indrashis Mitra**	Indrashis Mitra
1807277	**Kinjal Sarkar**	Kinjal Sarkar
1807291	**Pratyay Basu**	Pratyay Basu

Date:-

ABSTRACT

Nowadays Internet of Things is bringing a revolution in the infrastructure of technologies. IoT based health monitoring system is essentially a patient monitoring system in which he can be supervised 24*7 . Remote Patient Monitoring arrangement enables observation of patients outside of customary clinical settings (e.g. at home), which expands access to human services thus bringing down costs. Healthcare is given extreme importance by each country with the advent of the novel corona virus. Recently there has been a spike in the use of smartphones and along with that, wearable sensor remote health monitoring has evolved quickly . IoT not only helps in preventing the spread of disease but also in getting a proper diagnosis, even if the doctor is present at a remote distance By facilitating effortless interaction among various modules, IoT has enabled us to implement various complex systems such as smart home appliances, smart traffic control systems, etc.

Health monitoring systems are one of the most notable applications of IoT. Many types of designs and patterns have already been put into use to monitor a person's health .

Several life-threatening diseases can be easily monitored by IoT based systems. Cardiovascular Disease (CVD) is a common disease which is the cause behind most of the deaths in the world. The number of heart bits per minute is denoted as the heart rate of the patient. It is also referred to as the pulse rate of the body. The rate changes with illness, due to damage to body, heart, and exercise. Hence heart rate is essential in determining one's health condition.

PROPOSED SYSTEM: The main objective of this project is to develop, design and implement a smart patient healthcare monitoring system. The sensors used here are embedded in the body of the patient to sense the parameters like the heartbeat and temperature. These sensors are connected to a master unit, that calculates the values of all of them. These values are then transmitted by leveraging IoT cloud technology, to the base . From the base station, these can be easily accessed by the doctor present at some other location. Thus based on the temperature and heartbeat values, the doctor can decide the state of the patient and appropriate measures can be taken.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS / ABBREVIATIONS

Symbol / Abbreviations	Description
MLX	Melexis
MEMS	Micro-electromechanical systems
GPRS	General Packet Radio Service
Wifi	**Wireless Fidelity**
MAX	Maxim integrated
ESP	Espressif Systems
IC	Integrated Circuit
I2C	Inter-Integrated Circuit

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Presently, with the information and technology revolution , smartphone-based health monitoring systems are becoming more widespread. These systems can be used to collect real-time information about health and give feedback to doctors. The use of monitoring systems can lower medical expenses for the country in the long run. At present, the combination of mobile internet with a health service system using open-source design has become very easy due to widespread access to mobile internet services. Using a smart device, doctors and patients can continuously observe the heart rate and can get important data and take proper steps to prevent severe damages . Heart rate and body temperature are some of the most important traits of the human body which are major contributors to determining a patient's health condition.

The normal pulse rate of a healthy adult is 60 to 100 beats per minute. The average human pulse rate is 70 beats per minute for males and 75 beats per minute are for females. Females aged 12 and older have faster heart rates than males.

1.2 ORGANIZATION OF THE REPORT

This report has been divided into 7 chapters: -

Chapter 1 – Introduction

Chapter 2 – Embedded systems – Metrics monitoring

Chapter 3 – Embedded systems – Fall detection system

Chapter 4 – IOT – Website monitoring

Chapter 5 – IOT – Blynk

Chapter 6 – Conclusion & Future Scope

Chapter 7 – Planning & References

CHAPTER 2

EMBEDDED SYSTEMS: METRICS MONITORING

OBJECTIVE: Monitor patient parameters remotely to increase efficacy of healthcare management systems

IDEATION:

We need to monitor the patient parameters from remote distances using various sensors. The data given out by the sensors are then sent over to cloud for further access via a Wi-Fi module (inbuilt or externally connected). This is being done to reduce the critical time of testing patient parameters before any major operation.

Here we are measuring the temperature and pulse parameter of a patient remotely.

2.1 Components required :-

1. MLX 90614 Temperature Sensor.
2. MAX 30102 Pulse Rate Sensor.
3. Jumper wires**.**
4. Nodemcu ESP8266 board

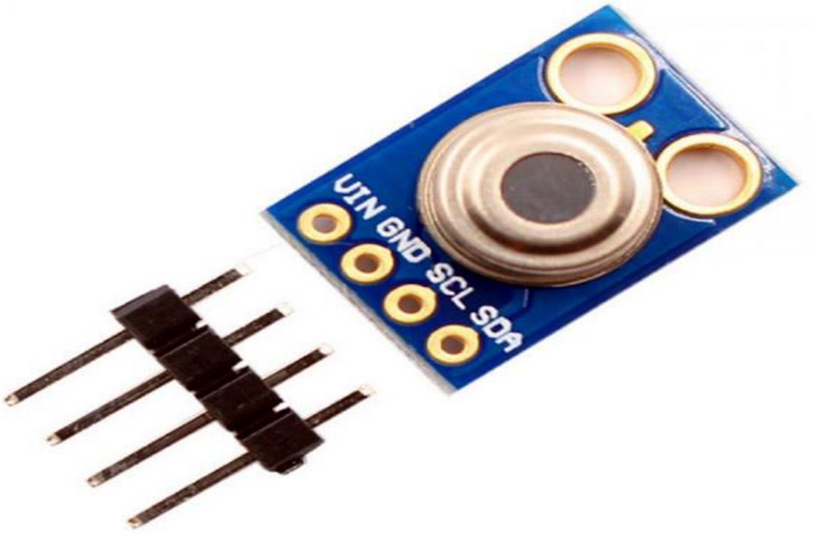


Fig 2.1 MLX90614 temperature sensor module

(a) Product Description: - (MLX90614)

The MLX90614 is a **Contactless IR Digital Temperature Sensor.** It is used to measure the particular object’s temperature ranging from -70° C to 382.2°C. It measures the temperature of the object without any physical contact and communicates to the microcontroller using the I2C protocol.

(b) Module Features:-

- Standard I2C interface with built 2x pull up resistors
- Can operate at voltage ranges from 3v to 5v.
- It can operate at the current of 2mA
- Has an I2C communication protocol:
- Working temperature ranges from -40 to +125 ° C .
- Can sense the temperature range from 70 to +380 °C
- Must maintain a measuring distance of 1 cm, while measuring the temperature,
- The PCB Dimensions IS 11mm x 17 mm

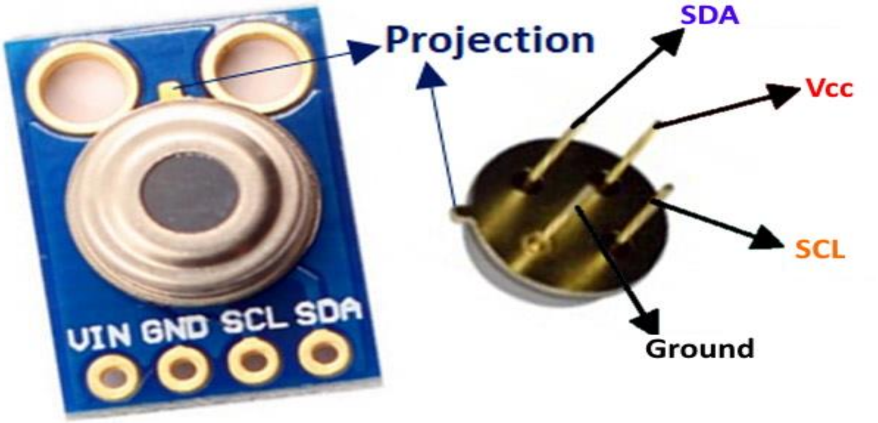


Fig 2.2 MLX90614 Pin-out Configuration

Pin No.	Pin Name	Description
1	**V_{dd} (Power supply)**	V_{dd} can be used to power the sensor, typically using 5V
2	**Ground**	The metal can also act as ground
3	**SDA – Serial Data**	Serial data pin used for I2C Communication
4	**SCL – Serial Clock**	Serial Clock Pin used for I2C Communication

Table 3.1

2.3 MLX90614 Features and Benefits

- The size is small and the cost is also low.
- Integration is easy.
- Factory calibrated in wide temperature ranges from: -40 to 125 °C for sensor temperature and -70 to 380 °C for object temperature.
- High accuracy of 0.5°C over wide temperature range (0 - +50 °C)
- Medical accuracy of 0.1°C in a limited temperature range available on request
- The measurement resolution of 0.01°C
- SMBus compatible digital interface for fast temperature readings and building sensor networks
- For continuous reading, customizable PWM output is there.
- Adaptation for 8 to 16V applications is simple
- Power saving mode
- Automotive-grade

2.4 Advantages of MLX90614:-

- Our project is related to the medical field so we chose the temp sensor which gives good accuracy. MLX 90614 's accuracy is near about 0.02° C so it is far better than other temp sensors.
- It is a contact-less infrared temp sensor and in this pandemic situation contact-less is the best option. It can sense the temp of an object from a distance of 2cm to 5cm approx.
- Its object temp range is near about -70 to 380 °C, so it can sense the wide range of temp.
- MLX 90614 consumes low voltage and low current. Its operating voltage is near about 3 V to 5 V and the required supply current is near about 2 mA.

2.5 Applications:-

- It applies to non-contact temperature measurements with high precision.
- Thermal Comfort sensor for Mobile Air Conditioning control system.
- It is used in temperature sensing element for residential, commercial, and industrial building air conditioning Windshield defogging
- It can also detect the automotive blind angle.
- Industrial temperature control of moving parts.
- Temperature control in printers and copiers
- Home appliances with temperature control
- Healthcare
- Livestock monitoring
- Movement detection
- Multiple zone temperature control – up to 127sensors can be read via common 2 wires
- Thermal relay/alert
- Body temperature measurement

2.6

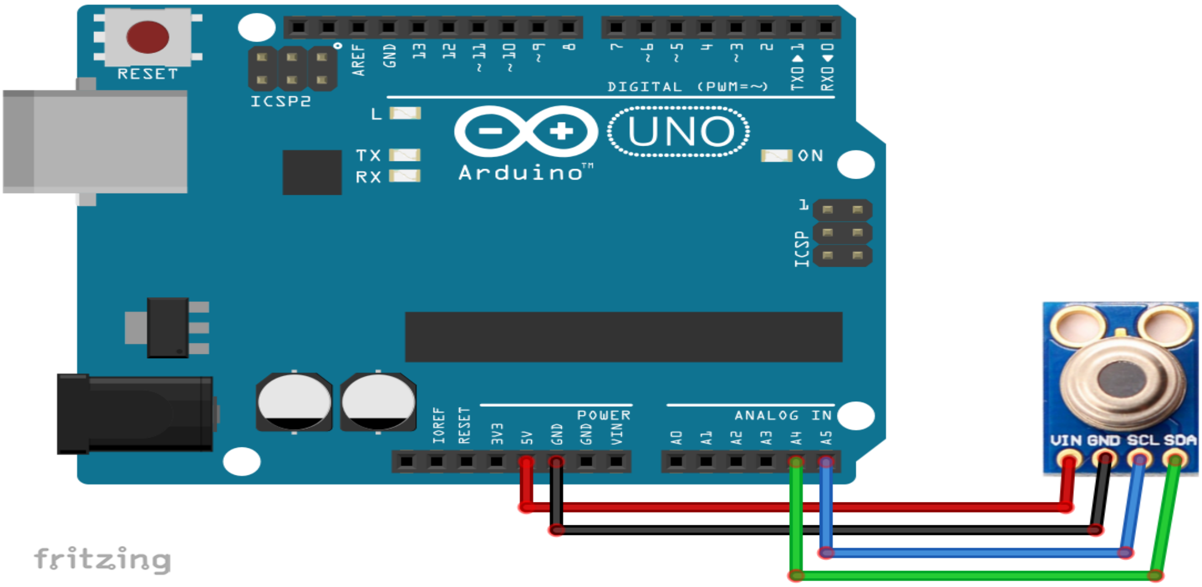
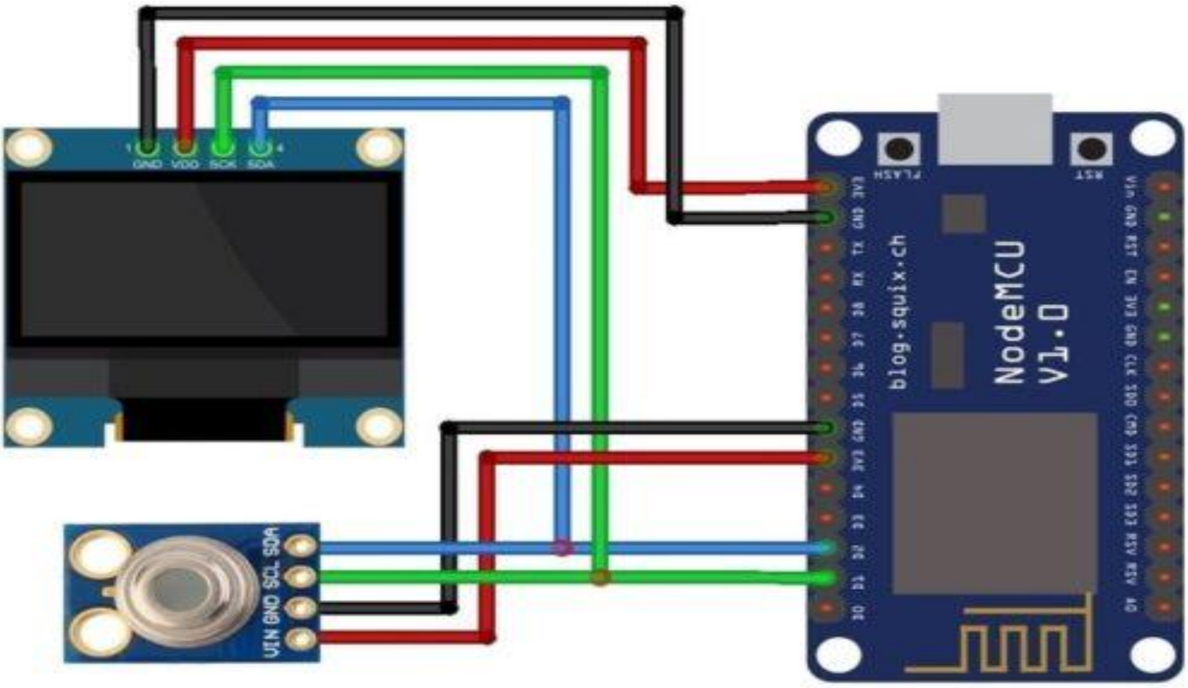


Fig 2.3 Arduino UNO and Nodemcu interfacing with MLX90614



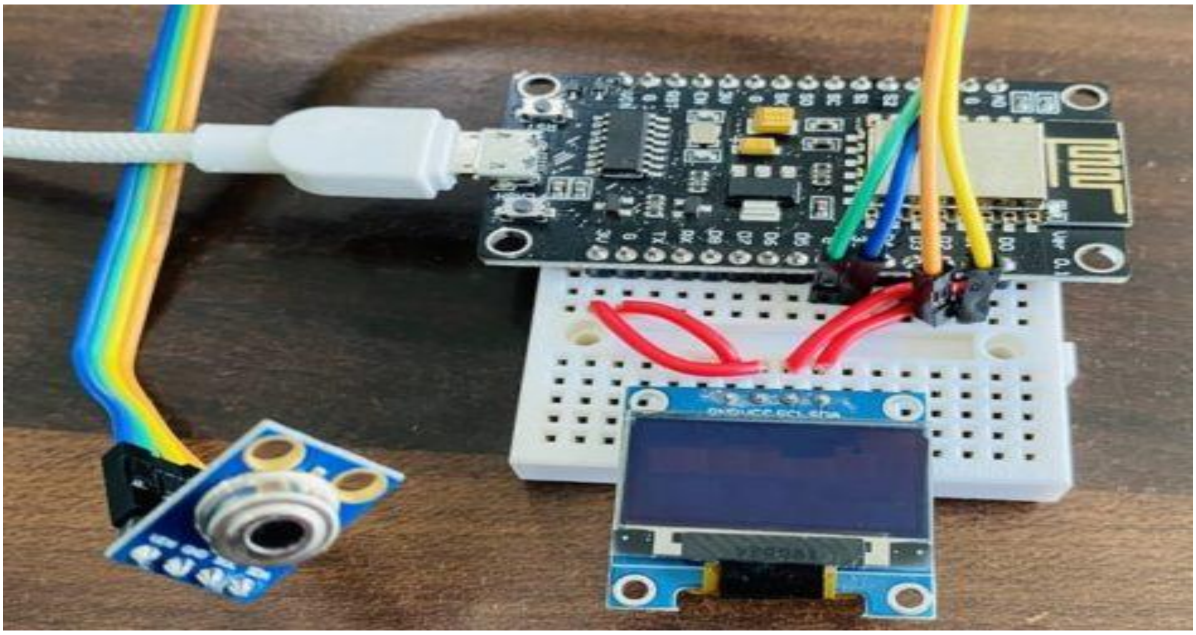


Fig 2.4 Connection diagram of MLX90614 with nodemcu board

SDA	D2
SCL	D1
VCC	3.3V
GND	GND

Table 3.2

2.7 Board Schematic representation :-

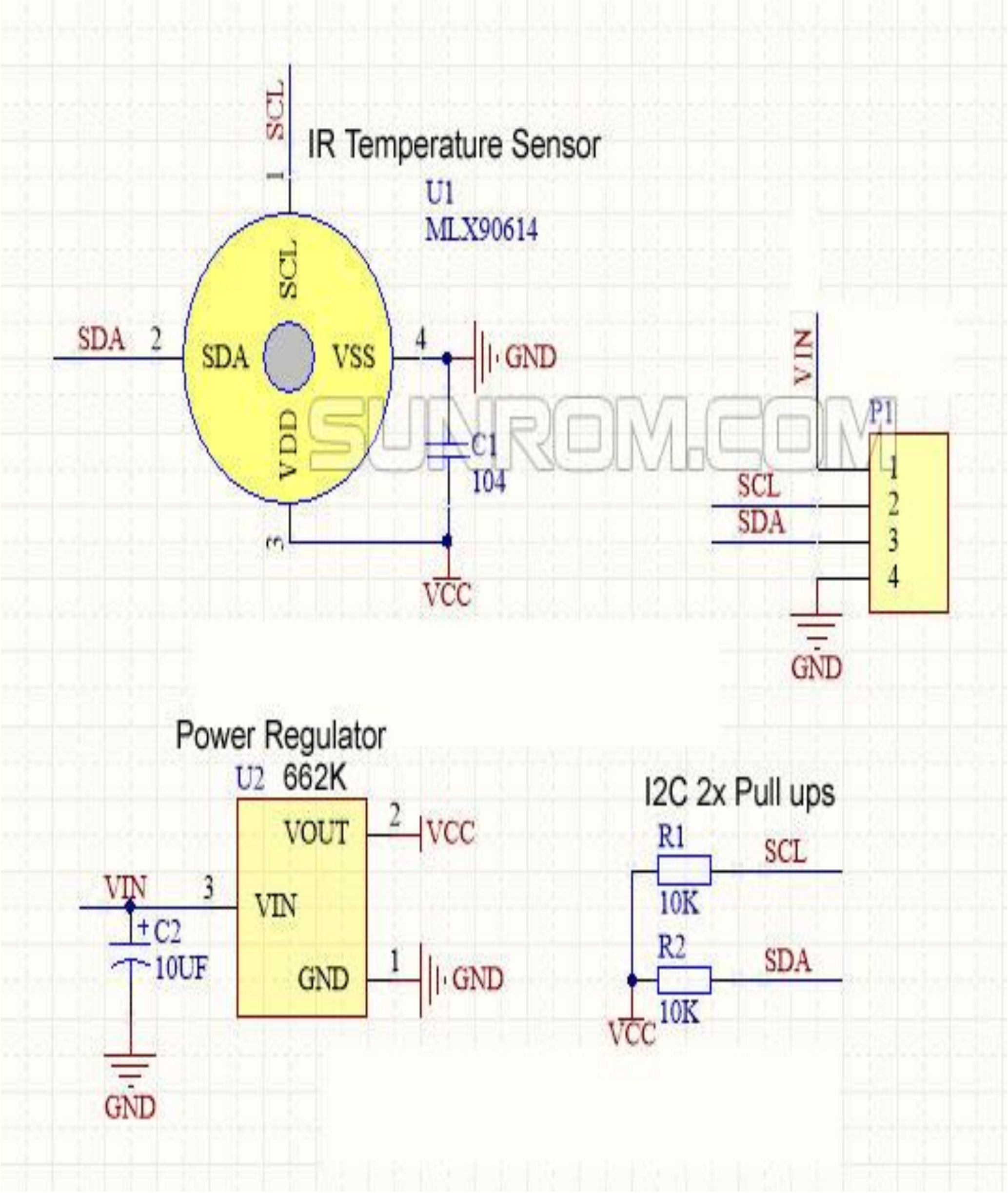


Fig 2.5 - Board schematic representation

2.8 MAX 30102 Pulse Rate Sensor:-

Product Description:-

The MAX30102 is a merged pulse oximeter and heart rate monitor biosensor module. It is a combination of a red LED and an infrared LED, photodetector, optical components, and low-noise electronic circuitry with ambient light suppression. - The MAX30102 has a 1.8 V power supply and a separate 5.0V power supply for internal LEDs for heart rate and blood oxygen accession in wearable devices, worn on the fingers, earlobe, and wrist. The standard I2C-compatible communication interface can transmit the collected values to the KL25Z and other microcontrollers for heart rate and blood oxygen calculation. Besides, the chip can also shut down the module through software, the standby current is close to zero, and the power supply is always maintained. Because of its excellent performance, the chip is largely used in the Samsung Galaxy S series mobile phones. Compared with the previous generation MAX30100, the chip combines a glass cover to effectively eliminate external and internal light interference, and has the best reliable performance. The Main Parameters: LED peak wavelength: 660nm/880nm LED power supply voltage: 3.3~5V Detection signal type: light reflection signal (PPG)Output signal interface: I2C communication interface voltage: 1.8~3.3V~5V (optional)Board reserved assembly hole size: 0.5 x 8.5mm Pin Description: VIN: main power input terminal 1.8-5V3-bit pad: Select the pull-up level of the bus, depending on the pin master voltage, select 1.8v or 3_3v (this terminal contains 3.3V and above)SCL: the clock connected to the I2C bus; SDA: data connected to the I2C bus; INT: Interrupt pin of the MAX30102 chip; RD: RED LED ground terminal of MAX30102 chip, generally not connected; IRD: The IR LED ground of the MAX30102 chip is generally not connected; GND: Ground wire.

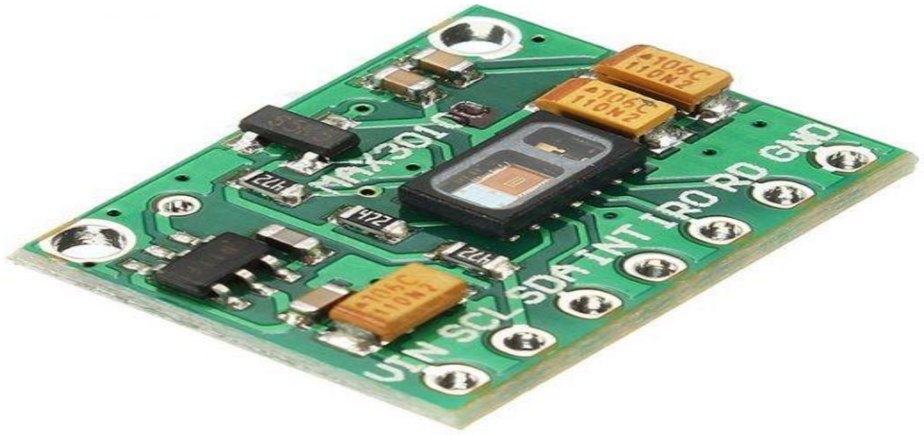


Fig 2.6 MAX 30102 pulse rate sensor

2.9 Detailed Pin description :-

Pin No.	Pin Name	Description
1	**Vin**	Main power supply input 1.8V-5V
2	**GND**	Ground Wire
3	**SDA – Serial Data**	Data connected to the I2C bus.
4	**SCL – Serial Clock**	The clock connected to the I2C bus.

Table 3.3

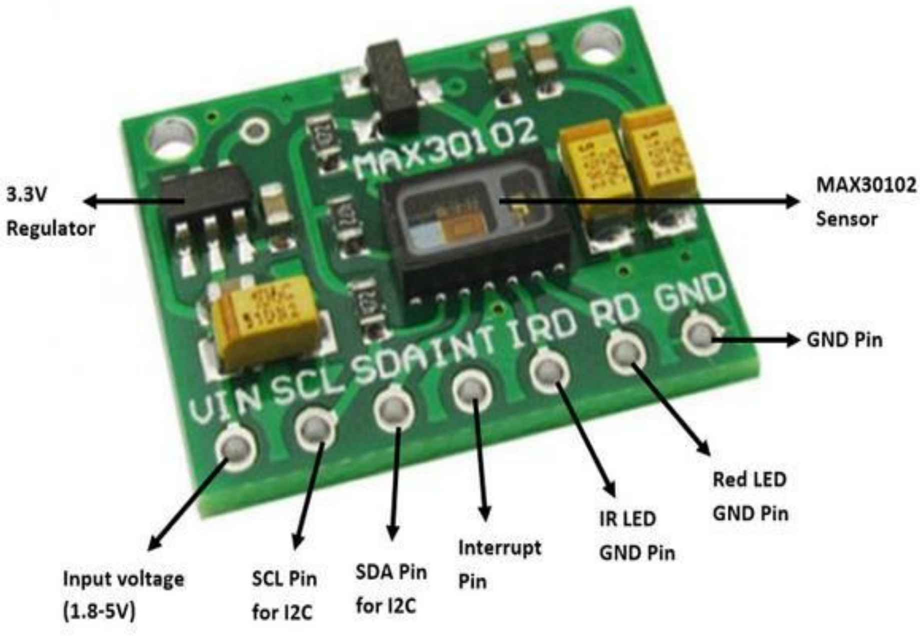


Fig 2.7 MAX 30102 Pin out configuration

2.10 Advantages of MAX3102 in compare to other pulse rate sensor:-

● The MAX30102 integrates red and IR LEDs to modulate LED pulses for oxygen saturation (SpO_2) and heart rate measurements.

● Space savings: Maintains a very small solution size without sacrificing optical or electrical performance; Integrates internal LEDs, photodetectors, optical elements, and low-noise electronics with ambient light rejection.

2.11 Specifications:

- Small in size and also highly integrated.
- It can also detect non-chest based heart-rate/SpO_2 detection
- Power consumption is very low.
- The Peak Wavelength of LED is 660nm/880The supply voltage of LED is 3.3~5V
- Interface: I2C Interface
- Operating Temperature Range is between -40°C and +85°C
- Dimension: 20.3 x 15.2mm
- Weight is just 1.1g

2.11 Principle description: -

Photo-dissolution method:

The measurement of pulse and blood oxygen saturation is performed by using human tissue to cause different light transmittance when the blood vessel beats.

Light source: A specific wavelength of light-emitting diode selective for oxyhaemoglobin (HbO_2) and haemoglobin (Hb) in arterial blood.

Light transmittance is converted into an electrical signal:

The change in the volume of the arterial pulsation causes the light transmittance of the light to change. At this time the light reflected by the human tissue is received by the photoelectric transducer, converted into an electrical signal, and amplified and output.

2.12 Arduino UNO and Nodemcu Interfacing:-

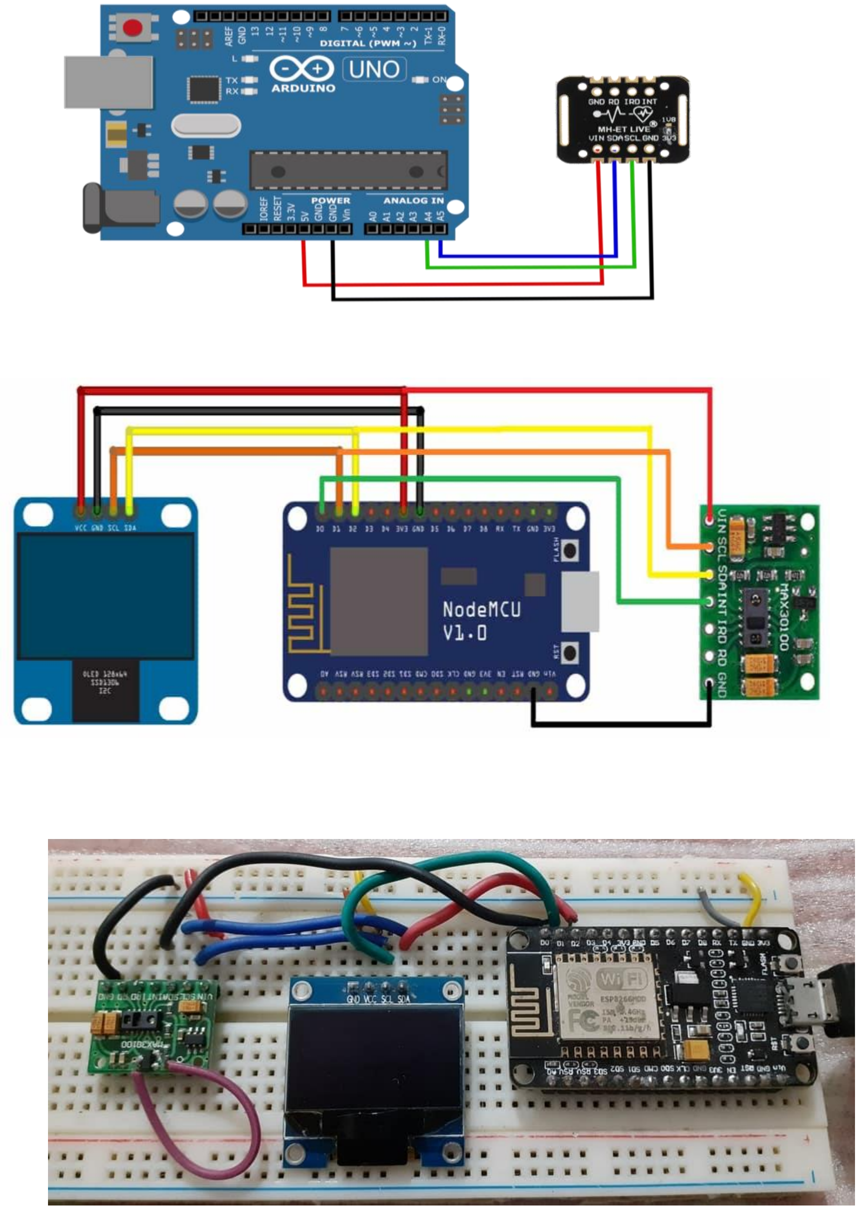


Fig 2.8 Connection diagram of MAX30102 with Nodemcu

SDA	D2
SCL	D1
VCC	3.3V
GND	GND

Table 3.4

2.13 -

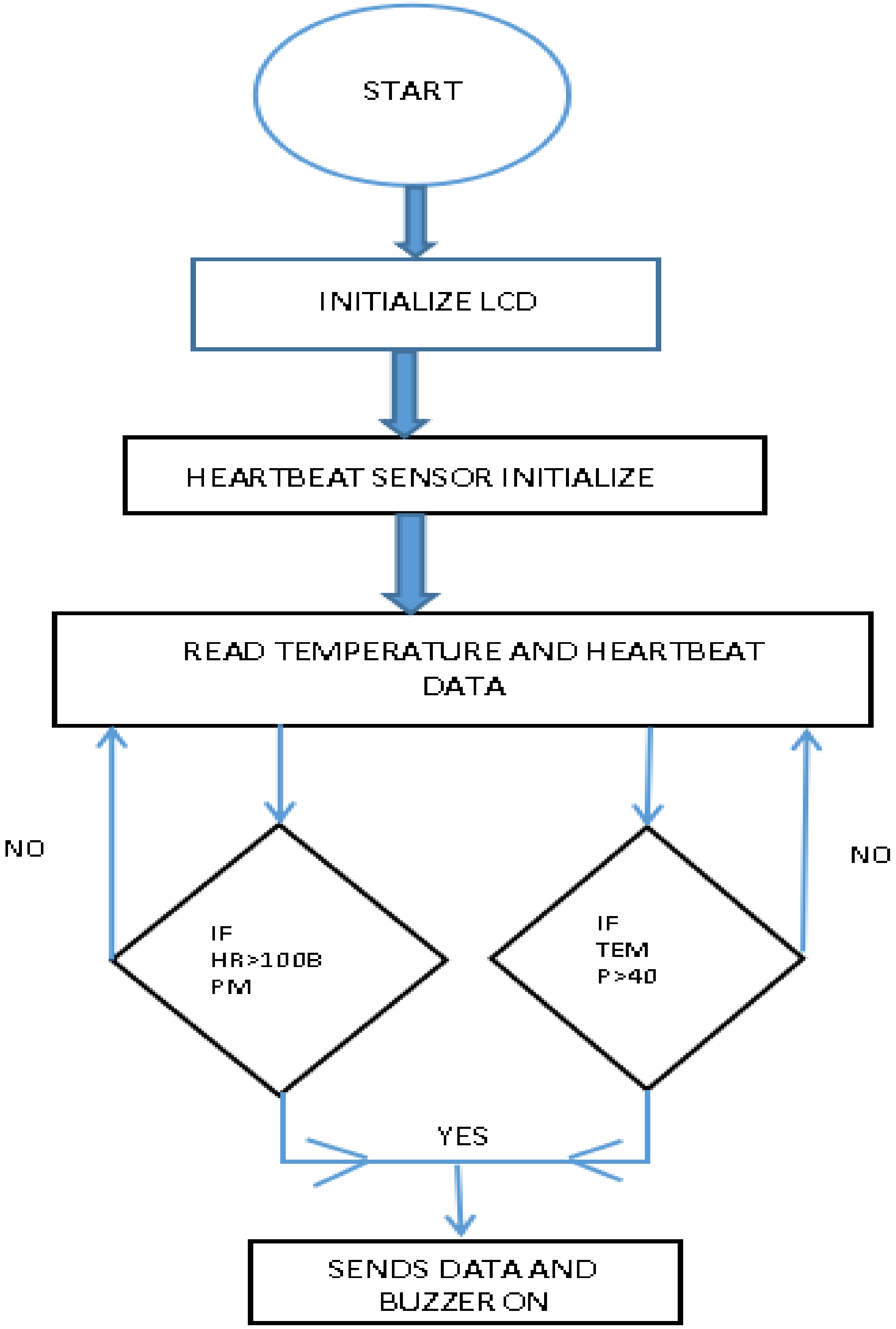


Fig 2.9 Flow diagram (For monitoring parameters)

CHAPTER 3

EMBEDDED SYSTEMS : FALL DETECTION SYSTEM

3.1 Objective (II) : Development of a Wearable-Sensor-Based Fall Detection System for the aged people.

3.1.1 Components Required:-

1. MPU6050 sensor board
2. Jumper wires.
3. Nodemcu ESP8266 board**.**

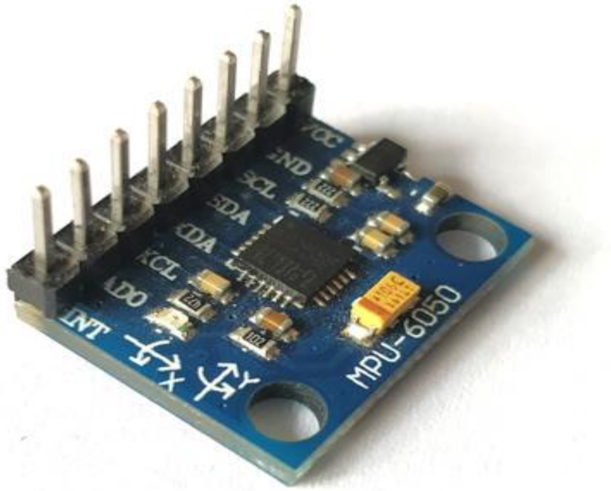


Fig 3.1 MPU6050 - Accelerometer and Gyroscope Module

3.2 Sensor Description :-

The **MPU6050** devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon together with an onboard Digital Motion Processor (DMP) capable of processing complex 9-axis Motion Fusion algorithms. This particular property helps us to measure acceleration, velocity, orientation, displacement and many other motion-related parameters of a system or object. This MPU6050 module also has a (DMP) Digital Motion Processor inside it which is powerful enough to perform complex calculations and thus does not require a microcontroller to function.

This module also has two auxiliary pins which can be optionally used to interface external IIC modules like a magnetometer. Since the IIC address of the module is configurable more than one MPU6050 sensor can be interfaced to a Microcontroller using the AD_0 pin. This module is well documented and also has revised libraries available hence it’s very easy to use with famous platforms like Arduino and ESP8266. So, if we want a sensor to control motion for your RC Car, Drone, Self-balancing Robot, Humanoid, Biped , or something like that then this sensor might be the right choice.

3.3 Pin Description:-

Table 4.1

Pin No.	Pin Name	Description
1	**V_{cc}**	Provides power for the module, can be +3V to +5V. Typically +5V is used
2	**Ground**	Connected to Ground
3	**Serial Clock (SCL)**	Used for providing clock pulse for I2C Communication
4	**Serial Data (SDA)**	Used for transferring Data through I2C communication

5	**Auxiliary Serial Data (XDA)**	Can be used to interface other I2C modules with MPU6050. It is optional
6	**Auxiliary Serial Clock (XCL)**	Can be used to interface other I2C modules with MPU6050. It is optional
7	**AD0**	If more than one MPU6050 is used a single MCU, then this pin can be used to vary the address
8	**Interrupt (INT)**	Interrupt pin to indicate that data is available for MCU to read

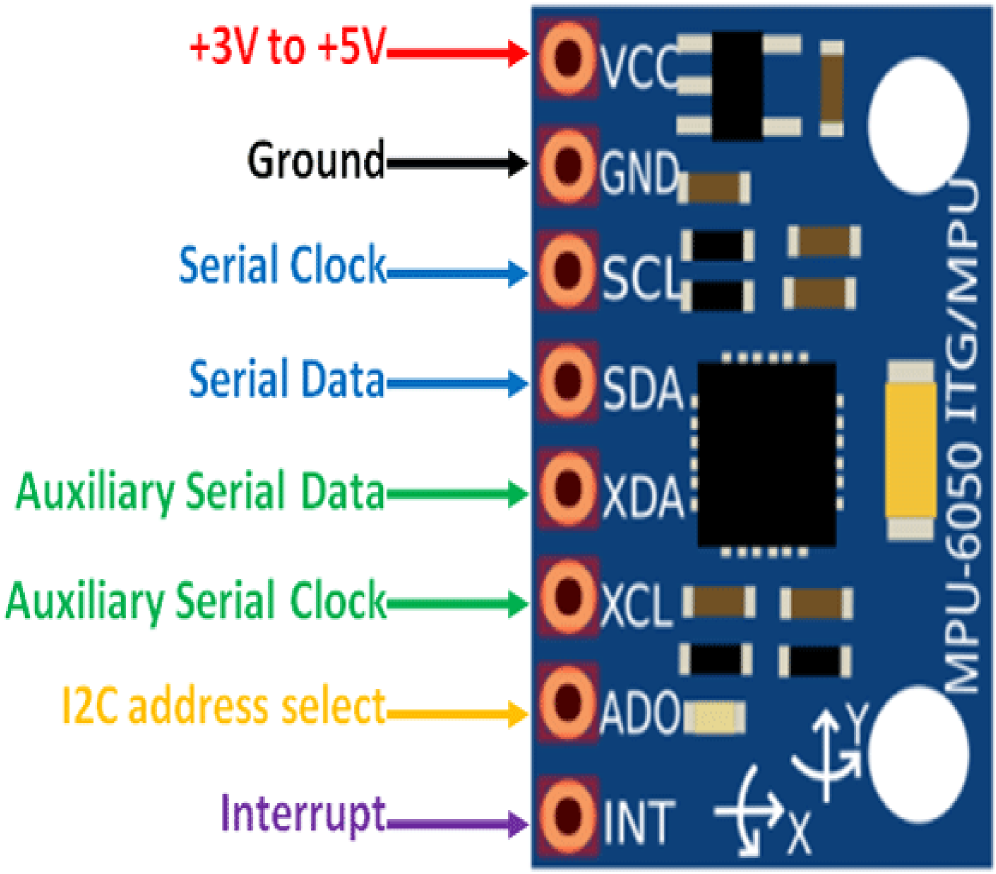


Fig 3.2 MPU6050 PINOUT

3.4 Using procedure of MPU6050:-

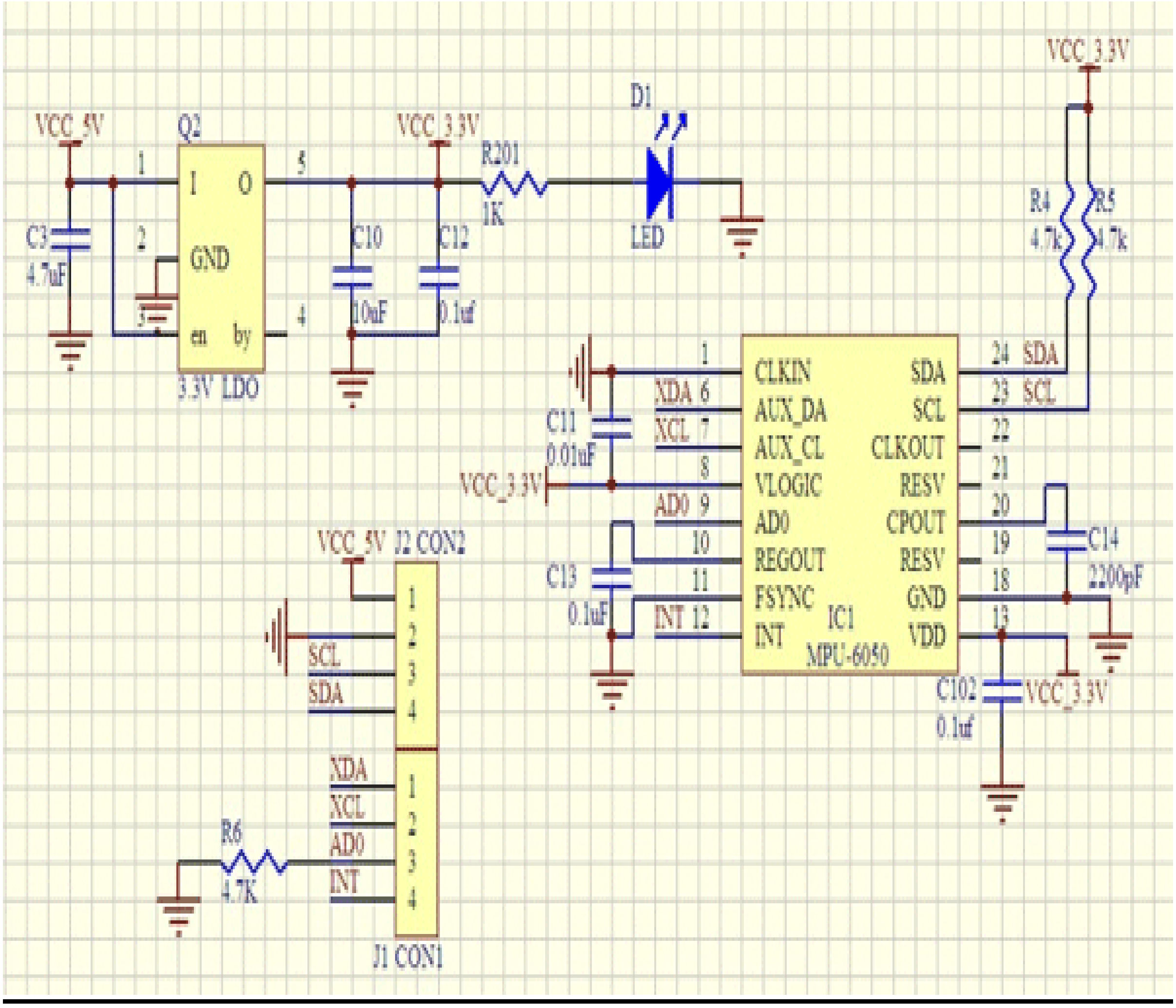


Fig 3.3 Board Schematic

The hardware of the MPU6050 module is very simple, it actually comprises the MPU6050 as the main components as shown above. Since it works on 3.3V, a voltage regulator is also required. The IIC lines are pulled high using a 4.7k resistor and the interrupt pin is pulled down using another 4.7k resistor.

The MPU6050 module allows us to read data from it through the IIC bus. Any change in motion will be reflected on the mechanical system which will in turn vary the voltage. Then the IC has a 16-bit ADC which it uses to accurately read these changes in voltage and stores it in the FIFO buffer and makes the INT (interrupt) pin to go high. This means that the data is ready to be read, so we use a MCU to read the data from this FIFO buffer through IIC communication. As easy as it might sound, you may face some problem while actually trying to make sense of the data. However there are lots of platforms like Arduino using which you can start using this module in no time by utilizing the readily available libraries .

3.5 MPU6050 Features:-

↙ MEMS 3-axis accelerometer and 3-axis gyroscope values combined
↙ Power Supply is 3-5V
↙ Communication: I2C protocol
↙ Built-in 16-bit ADC provides high accuracy
↙ Built-in DMP provides high computational power
↙ Can be used to interface with other IIC devices like magnetometer
↙ Configurable IIC Address
↙ In-built Temperature sensor is there

3.6 Interfacing MPU6050 with Arduino UNO and Nodemcu board:-

It is easy to interface the MPU6050 with Arduino due to the library developed by a person named Jeff Row-berg.

As soon as this library is added to Arduino IDE, follow the below schematics for establishing an IIC connection between your Arduino UNO or Node MCU ESP8266 and MPU6050 sensor

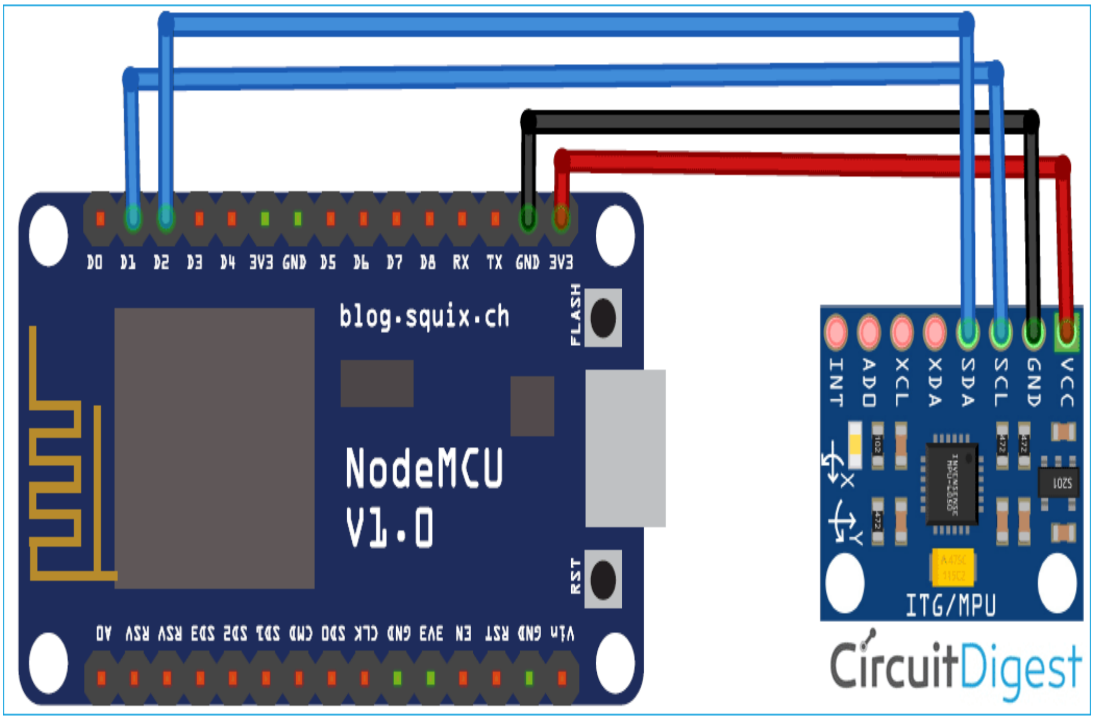


Fig 3.4 Circuit Connection of MPU6050

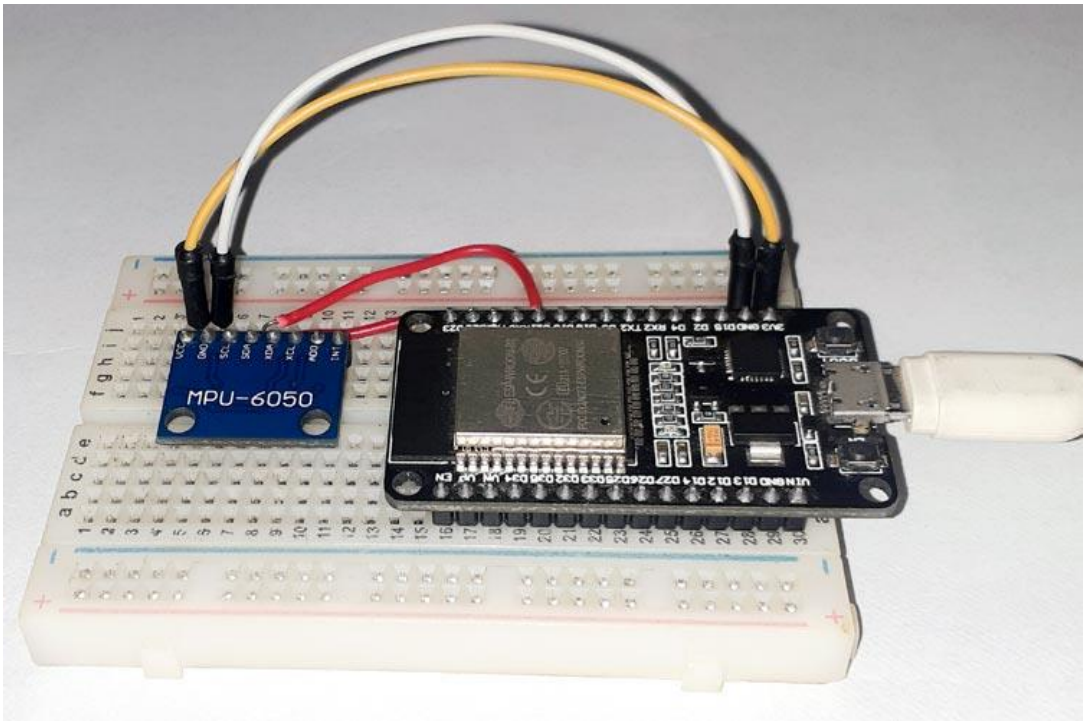


Fig 3.5

3.7 MODEL_ACTIVITY

Taking in account that the fact where direction of gravity is invariably perpendicular to the ground, and that the orientation of the vest worn on the body is supposed to be the same as that of the trunk, we will be using a Cartesian coordinate system OXYZ for the upper trunk, the origin of which is close to the neck of the human body, and is parallel with the geodetic coordinate system OXYZ, as depicted in Figure 3.6.

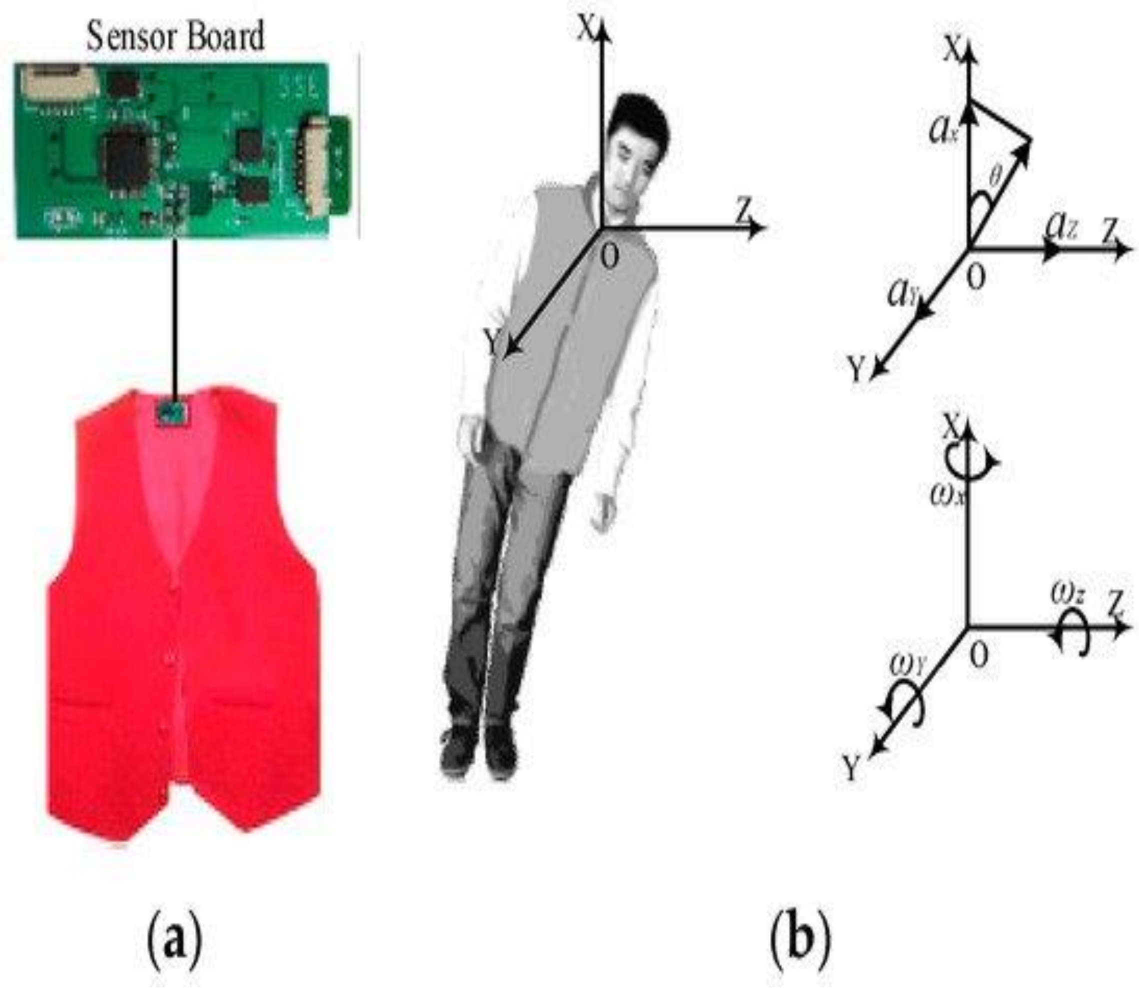


Fig 3.6

Fig 3.6(a) The placement of the sensor board
3.6(b). The geodetic coordinate OXYZ

At a time t, accelerations along the X, Y, and Z axes are denoted as Ax(t), Ay(t) and Az(t) respectively, namely A(t) = {Ax(t), Ay(t), Az(t)}. The resultant acceleration α(t) can be calculated using Equation (1):

$$A(t)= \sqrt{Ax(t)^2 + Ay(t)^2 + Az(t)^2}$$

Since Ax(t), Ay(t) and Az(t) contain an approximation of the gravitational component of the acceleration on every axis, the trunk angle (namely θ(t)) can be calculated using Equation (2):

$$\theta(t)= \cos^{-1}\left(\frac{Ax(t)}{\sqrt{Ax(t)^2 + Ay(t)^2 + Az(t)^2}}\right)$$

The x-axis is perpendicular to the gravitational direction in the lying position and parallel to the gravitational direction in the standing position. A fall usually means that the trunk changes from a standing position to a lying position, and the θ(t) increases from about 0° to about 90°.

Meanwhile, the tri-axial angular velocities of the trunk can be collected by gyroscope. Wx(t), Wy(t), Wz(t) are the angular velocity at time t in the X, Y, and Z axes respectively, namely W(t) = {Wx(t), Wy(t), Wz(t)}. The resultant angular velocity W(t) can be calculated using Equation (3):

$$W(t) = \sqrt{Wx(t)^2 + Wy(t)^2 + Wz(t)^2}$$

3.8 Data Acquisition :-

Fig 3.7 shows the sensor board, which is about 65 mm × 40 mm × 7 mm (length × width × thickness), and is appropriate for use in a vest. The board contains a class 2 Bluetooth module and a low power microcontroller. The default transmission rate of the module is 1,15,200 bps with the maximum range being 10m. The tri-axial accelerometer can measure upto ±16 g with full-scale reading of the gyroscope being ±2000°/s. The sampled data from both of these are read and transmitted to an Android smartphone.

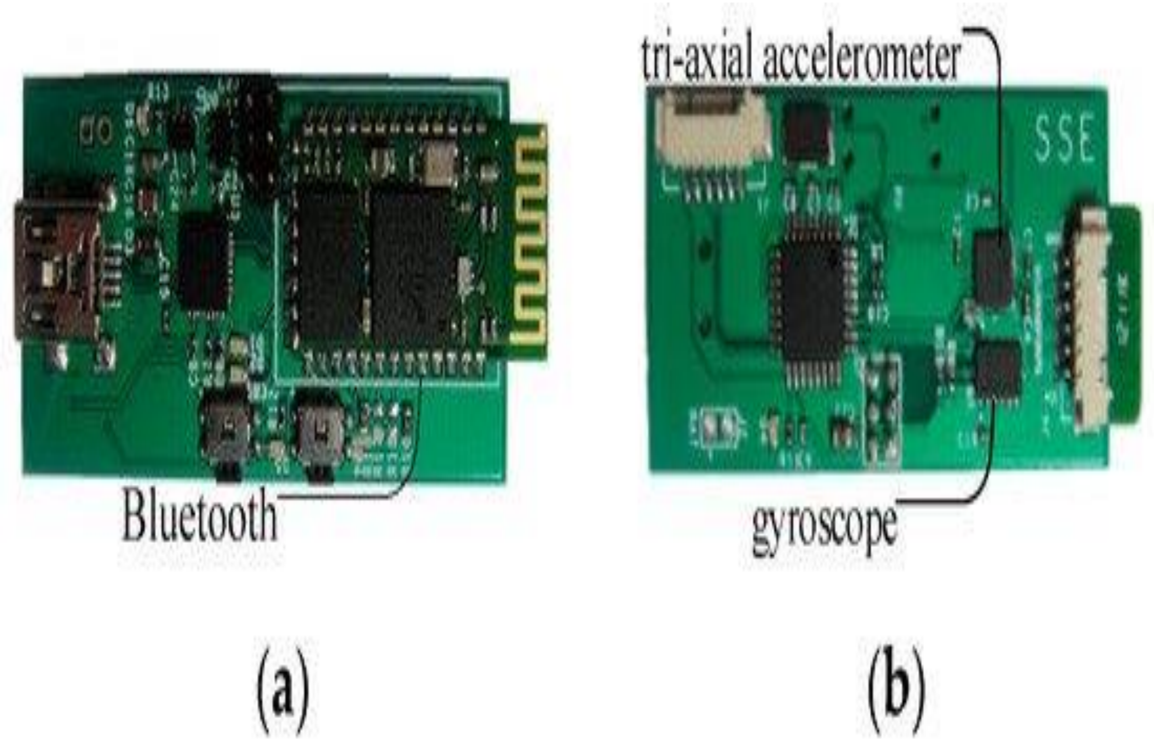


Fig 3.7(a & b) Front & back view of sensor board with Bluetooth, tri-axial accelerometer and gyroscope

Most frequencies of human activities are less than 20 Hz, so the sampling frequency from human activities can be set to 100 Hz. The board can acquire tri-axial accelerations and angular velocities which can be sent directly to a smartphone.

As falls are usually characterized by rapid acceleration and great angular velocity, hence 4 subcategories of ADLs and 2 kinds of falls are proposed so that the difference between ADLs and falls can be found out. ADLs include Sitting down (Sd), Squatting down (Sq),Walking (Wk), and Bowing (Bw). Falls include Backward fall (Bw-Fall) and Sideward fall (Sw-Fall) .

3.9 **Flow Diagram :-**

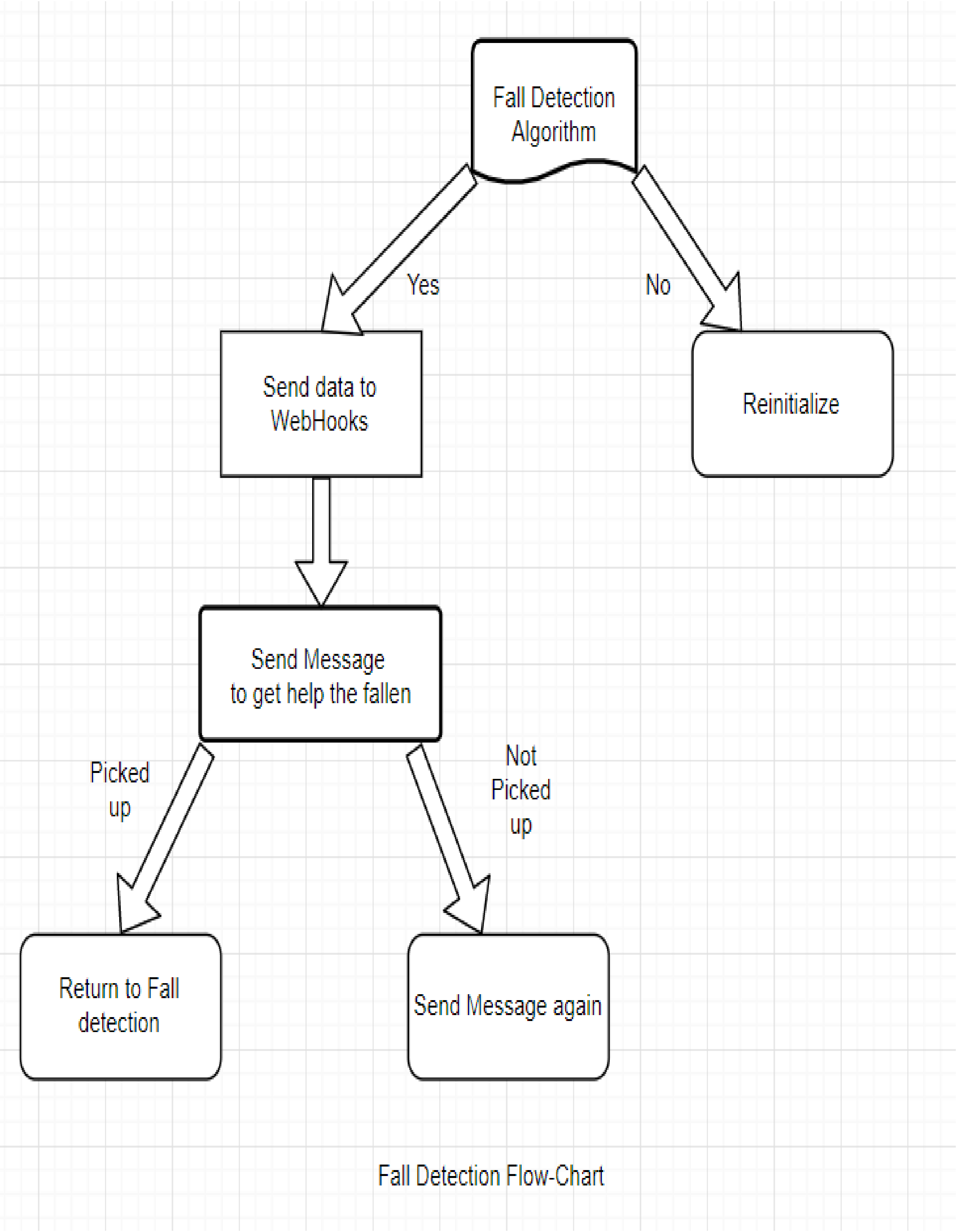


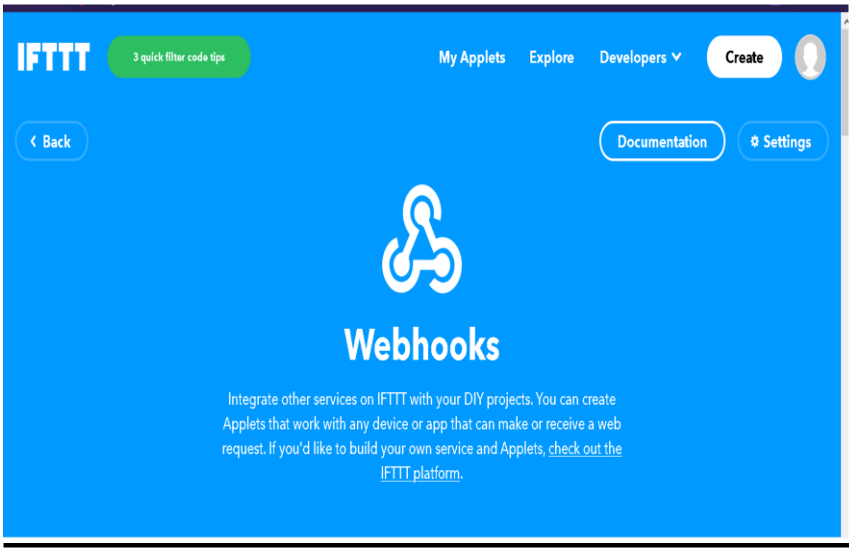
Fig 3.8 Flow diagram

3.10 IFTTT Integration:-

IFTTT (**If This Then That**) is a web-based service by which chains of conditional statements, called applets can be created. Using these applets, we can send Emails, post on Twitter, play music, exchange SMS, receive and send notifications, etc. The purpose of using IFTTT in this project is to send SMS notifications to the mobile phone when the system detects a fall.

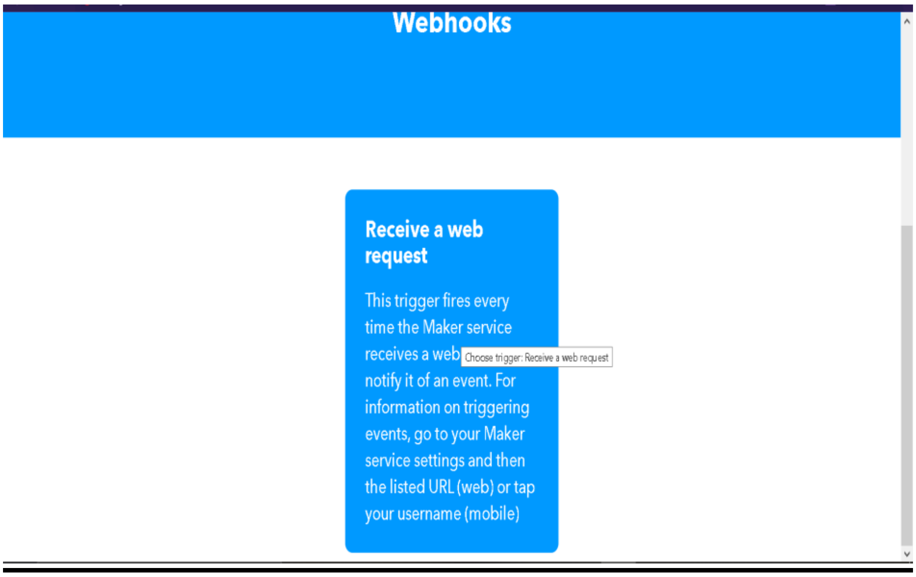
To use the IFTTT, we need to sign in to our IFTTT account in case we already have an account or we need to create an account.The steps are as follows:-

- Click on the documentation tab for getting the key which we will use in the programming



Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

- Then we will click on create
- After that click on “this”
- We need to now search for the webhooks and activate it

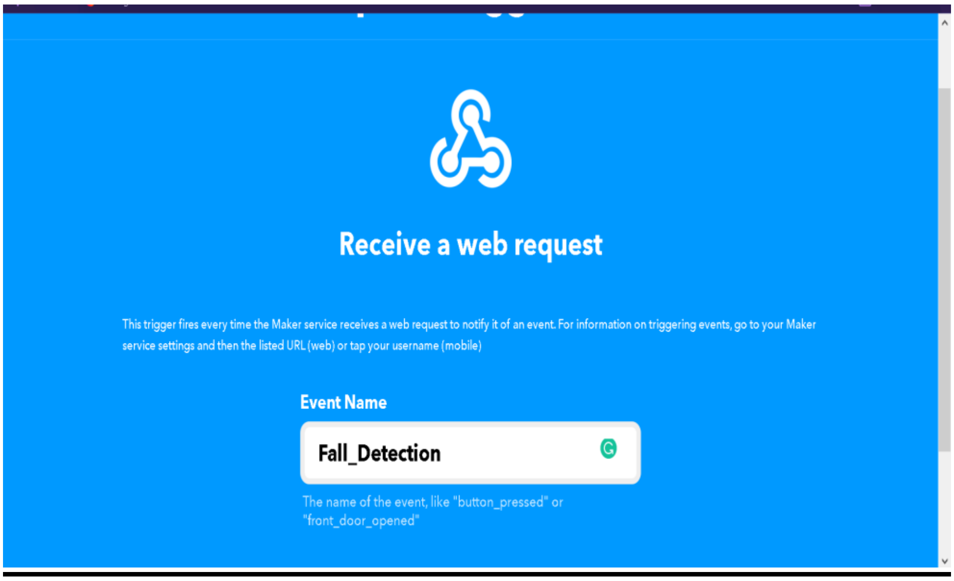


Webhooks

Receive a web request

This trigger fires every time the Maker service receives a web notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

- After clicking on the webhooks, now we want to select the "Receive a web request" part and allocate it a name as per our wish. In this case, we have given the name as "fall detect"



Receive a web request

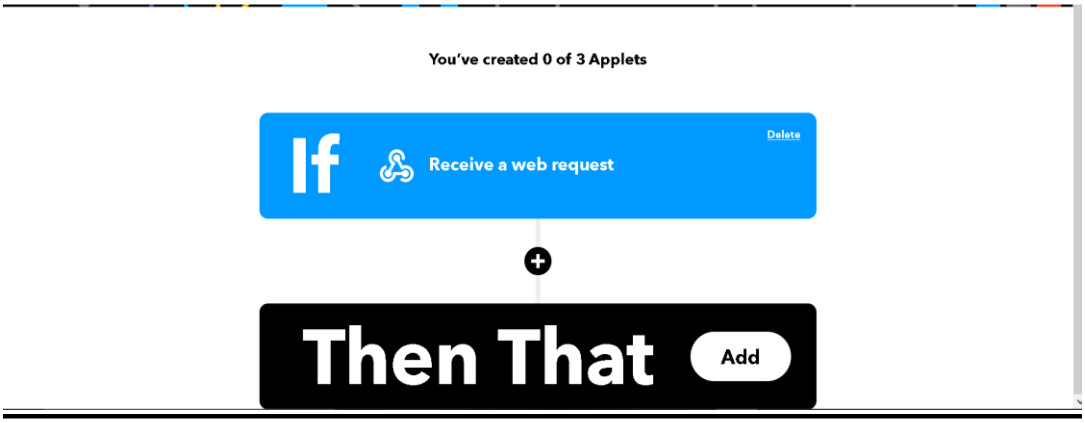
This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Event Name

Fall_Detection

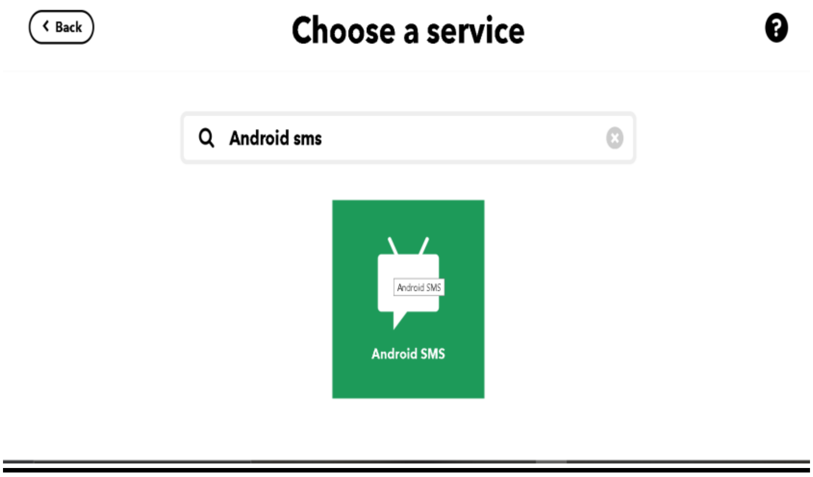
The name of the event, like "button_pressed" or "front_door_opened"

- Now we will click on “that”



You’ve created 0 of 3 Applets

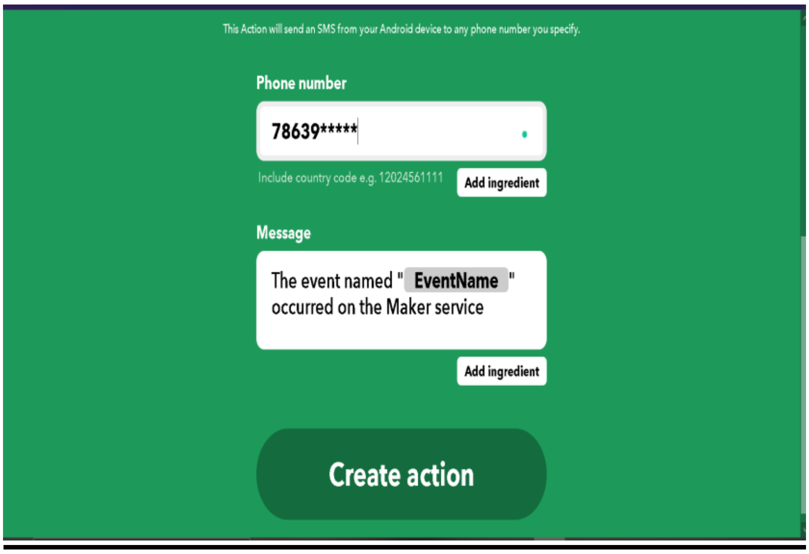
- We need to search for Android SMS option



Choose a service

Android sms

- We need to enter the required phone number song with the country code



This Action will send an SMS from your Android device to any phone number you specify.

Phone number

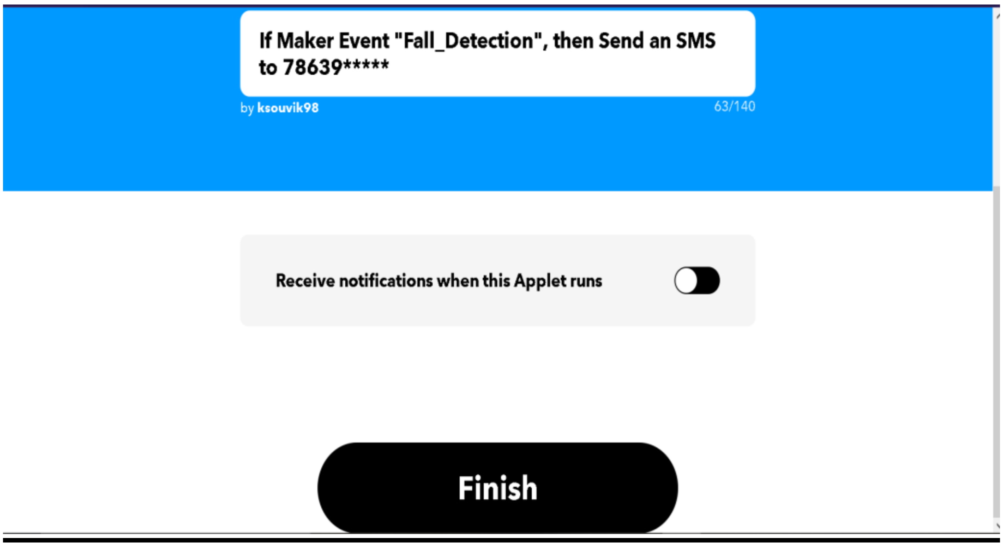
78639*****

Include country code e.g. 12024561111

Message

The event named " EventName " occurred on the Maker service

- After that we will click on create action

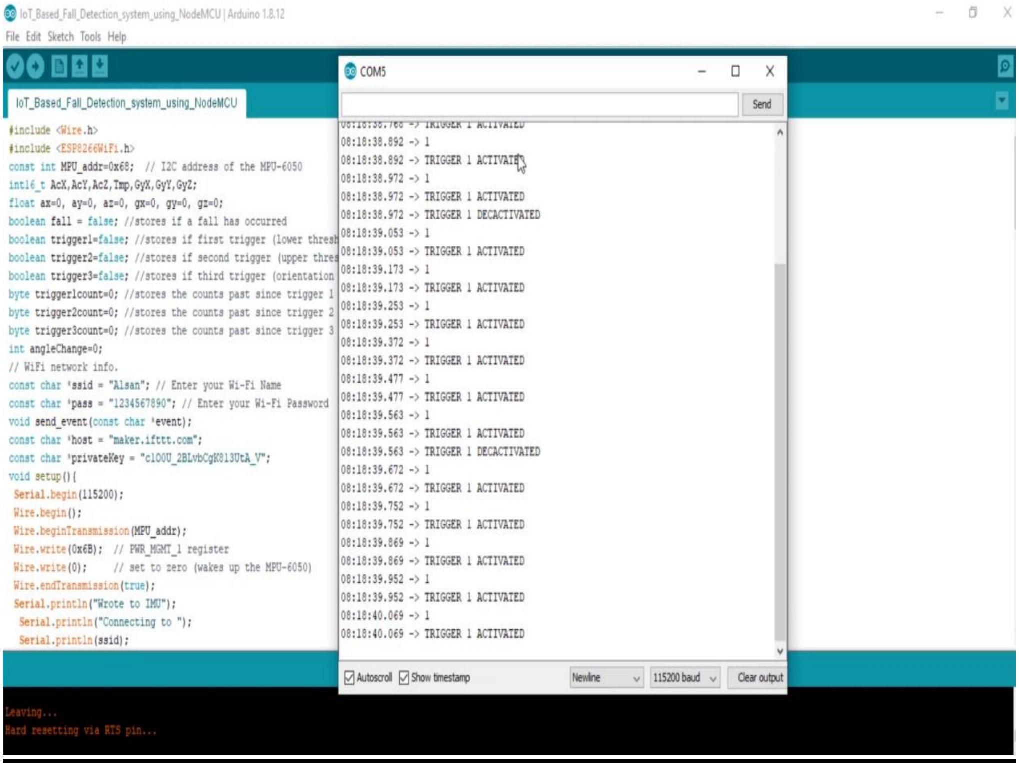


If Maker Event "Fall_Detection", then Send an SMS to 78639*****

by ksouvik98 63/140

Receive notifications when this Applet runs

- Then Click on Create Action.



IoT_Based_Fall_Detection_system_using_NodeMCU | Arduino 1.8.12
File Edit Sketch Tools Help
IoT_Based_Fall_Detection_system_using_NodeMCU
COM5
Send

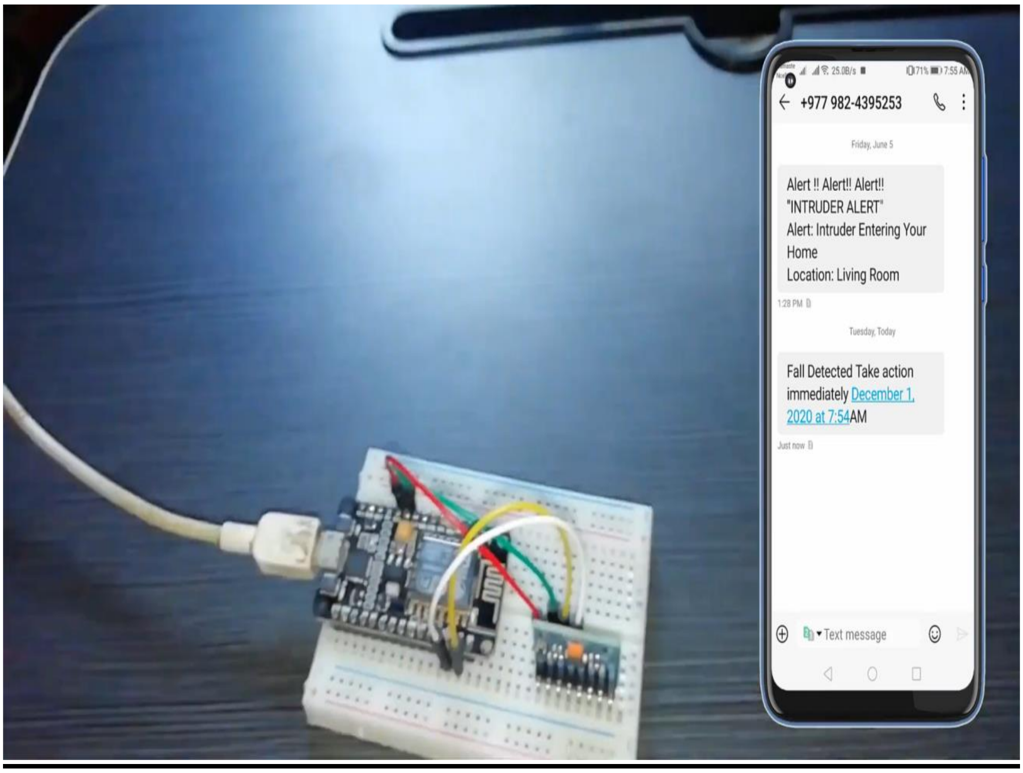
```
#include <Wire.h>
#include <ESP8266WiFi.h>
const int MPU_addr=0x68;  // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;
boolean fall = false; //stores if a fall has occurred
boolean trigger1=false; //stores if first trigger (lower thresh
boolean trigger2=false; //stores if second trigger (upper thres
boolean trigger3=false; //stores if third trigger (orientation
byte trigger1count=0; //stores the counts past since trigger 1
byte trigger2count=0; //stores the counts past since trigger 2
byte trigger3count=0; //stores the counts past since trigger 3
int angleChange=0;
// WiFi network info.
const char *ssid = "Alsan"; // Enter your Wi-Fi Name
const char *pass = "1234567890"; // Enter your Wi-Fi Password
void send_event(const char *event);
const char *host = "maker.ifttt.com";
const char *privateKey = "c1O0U_2BLvbCgK813UtA_V";
void setup(){
 Serial.begin(115200);
 Wire.begin();
 Wire.beginTransmission(MPU_addr);
 Wire.write(0x6B);  // PWR_MGMT_1 register
 Wire.write(0);     // set to zero (wakes up the MPU-6050)
 Wire.endTransmission(true);
 Serial.println("Wrote to IMU");
  Serial.println("Connecting to ");
  Serial.println(ssid);
```

```
08:18:38.768 -> TRIGGER 1 ACTIVATED
08:18:38.892 -> 1
08:18:38.892 -> TRIGGER 1 ACTIVATED
08:18:38.972 -> 1
08:18:38.972 -> TRIGGER 1 ACTIVATED
08:18:38.972 -> TRIGGER 1 DECACTIVATED
08:18:39.053 -> 1
08:18:39.053 -> TRIGGER 1 ACTIVATED
08:18:39.173 -> 1
08:18:39.173 -> TRIGGER 1 ACTIVATED
08:18:39.253 -> 1
08:18:39.253 -> TRIGGER 1 ACTIVATED
08:18:39.372 -> 1
08:18:39.372 -> TRIGGER 1 ACTIVATED
08:18:39.477 -> 1
08:18:39.477 -> TRIGGER 1 ACTIVATED
08:18:39.563 -> 1
08:18:39.563 -> TRIGGER 1 ACTIVATED
08:18:39.563 -> TRIGGER 1 DECACTIVATED
08:18:39.672 -> 1
08:18:39.672 -> TRIGGER 1 ACTIVATED
08:18:39.752 -> 1
08:18:39.752 -> TRIGGER 1 ACTIVATED
08:18:39.869 -> 1
08:18:39.869 -> TRIGGER 1 ACTIVATED
08:18:39.952 -> 1
08:18:39.952 -> TRIGGER 1 ACTIVATED
08:18:40.069 -> 1
08:18:40.069 -> TRIGGER 1 ACTIVATED
```

☑ Autoscroll ☑ Show timestamp Newline 115200 baud Clear output

```
Leaving...
Hard resetting via RTS pin...
```

Fig 3.9 Illustration of IFTTT steps



Friday, June 5

Alert !! Alert!! Alert!!
"INTRUDER ALERT"
Alert: Intruder Entering Your Home
Location: Living Room

1:28 PM

Tuesday, Today

Fall Detected Take action immediately December 1, 2020 at 7:54AM

Just now

Fig 3.10 Connection diagram

CHAPTER-4

IOT- WEBSITE MONITORING

Objective:- To effectively monitor the patient parameters via a website.

IDEATION:-The parameters that are being monitored by the sensors are checked via a website that will display the sensor data in a graphical way for better understanding and to keep check on the threshold limits of the parameters.

4.1 Introduction to Thinger.io:-

Thinger.io is a free, cloud IoT Platform that helps to prototype, scale and manage connected products in an easy way. The aim is to regularize IoT usage ,thereby making it reachable to the whole world, and modernizing the evolution of big IoT projects.

- **Free platform for IoT**: the makers have come up with a free premium account with only few restrictions to start training and make a mock-up of our scalable product. A Premium Server can easily be deployed in a short span of time.
- **Plain but Robust**: To connect a device, we need very few code lines . Then we can start recovering data or controlling its functions with the web-based console and control many devices in a simple way.
- **Hardware nullifidian:** There is no specification for any device or manufacturer. We can use as per our choice or requirement.
- **Extremely expandable & methodical architecture:** due to the distinctive communication model, wherein the IoT **server provides resources** to recover data only when required, one instance can oversee many IoT devices with low computational load, bandwidth and latencies.

- **Open-Source**: most of the platform modules, libraries and APP source code are available in the Github repository to be downloaded and modified with MIT license.

  Thinger.io is an Open-source cloud-based IoT Platform developed by INTERNET OF THINGER SL, a Spanish company whose objective is to provide an efficient, consistent and easy to use technology for IoT.

  Thinger.io Platform project started life in early 2015, despite some of the first source code lines was made on 2014, as a side-project by Ph.D. Alvaro Luis Bustamante when he was working as researcher at the University Carlos III of Madrid (Spain), when he was studying different solutions to work online with cheap electronic devices, and he found that all the existent platforms were very difficult to use, inefficient or doesn't had enough capacities to create IoT projects in a simple way.

4.2 Main Features of thinger.io:-

The platform is made up of two products - a Backend server and a web-based frontend. Both of these are vital in making work with any smartphone or computer,really simple and easy. The main features provided are:-

- **Join devices:** Adaptable with any device, irrespective of its specifications or maker. We can create two - way **transmission** with Arduino, Raspberry Pi, Linux or MQTT devices . Edge technologies like Sigfox or LoRaWAN or other internet API data resources are also supported..
- **Store Device Data:** Just a couple clicks to create a Data Bucket a store IoT data in a scalable, efficient and affordable way, that also allows real-time data aggregation.
- **Show instantaneous** or **Stored Data** in a variety of widgets such as donut charts,time series, gauges, or custom made renditions.
- **Activate events and information** using an implanted Node-RED rule engine
- **Expand with user-defined attributes** diverse plugins are there to amalgamate IoT projects

- **Customization of appearance** can be done using the reconditioned frontend allowing introduction of branding colours, logos and web domain.

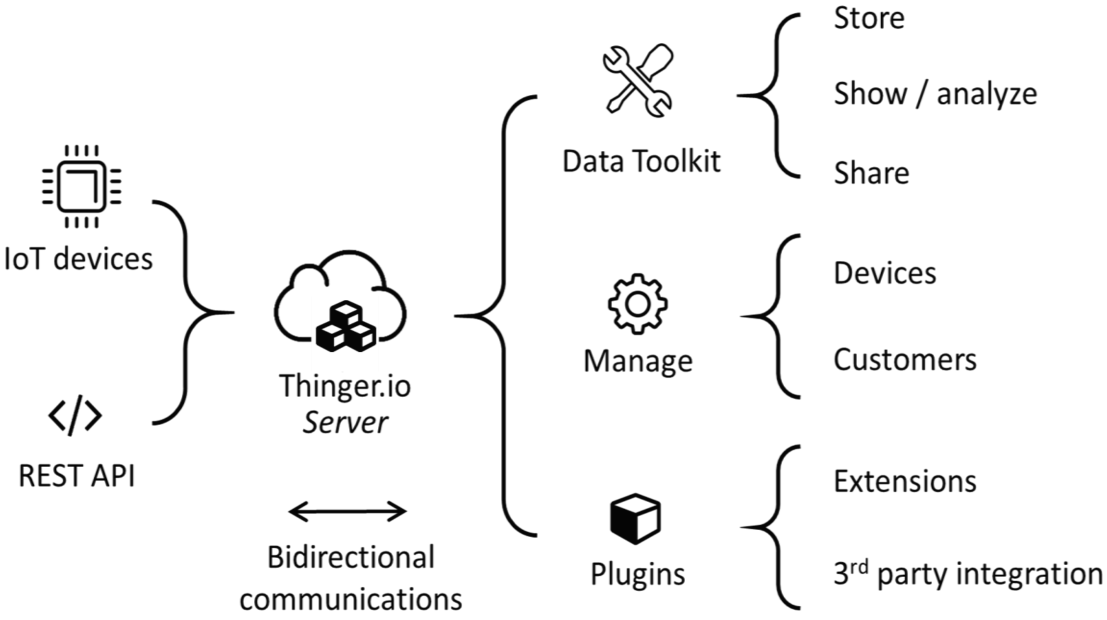


Fig 4.1 Connectivity of thinger.io platform

The core of Thinger.io IoT Platform is designed to be light, in order to increase elasticity and decrease learning and common configuration of IoT networks. Plugins offer custom functions that can be deployed on-demand, allowing each user to complement their IoT server as its specific needs.

Dashboards are a means to visualize data giving all users ability to understand the analytics essential for their business or project. Generic users can use this to participate and understand the analytics by data compilation and seeing trends. We plan to make a web based dashboard for effective and easy monitoring and understanding of the various health parameters of the patient, such as blood pressure and temperature.

Thinger.io dashboard system is a feature that allows creating nice data representation interfaces within minutes in a very simple way. No coding is required, just selecting different widgets from a list and using drag drop technology to configure the layout of the dashboard, then using the configuration forms it is possible to set the data sources, sampling interval, and other behaviours of each widget. The main types of these widgets are:

- **Real-time** data representation
- **Historical** data representation from buckets
- **Control** device functions or change values with On/Off buttons or sliders

Why this approach?

We could have used normal website coding (using HTML,CSS) to design a website for displaying the parameters. But we preferred using this since we found that it was open source,easy to use and most importantly, it was convenient to create simple,attractive dashboards using this platform. Moreover, the array of devices supported by thinger.io is humongous. So, a wide variety of devices can be easily integrated in the project, irrespective of the manufacturers or properties.

4.3 Working Procedure:-

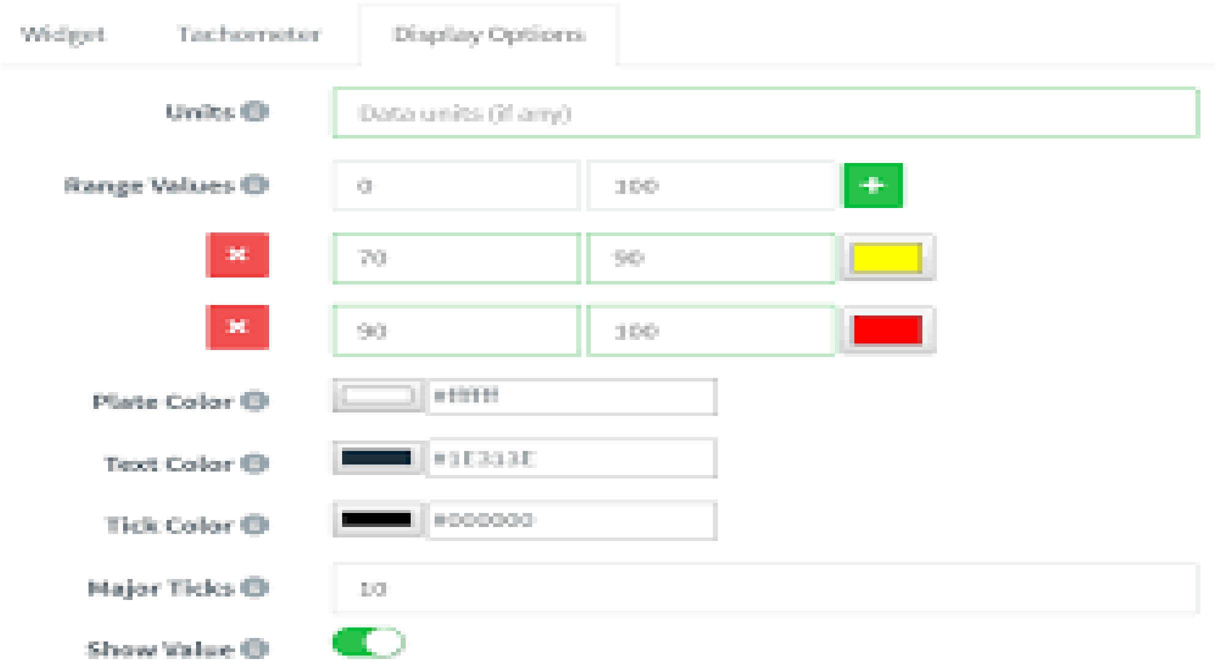
1. Create device – create a device on thinger.io by registering a new device. By logging into console dashboard, go to Devices ->Add Device.

   Add the device credentials such as identifier, description and credentials, like Wi-Fi username, etc. as mentioned in the code.

If everything is successful a message appears and the device is displayed in the list of devices.

2. Add a display widget -while editing on the dashboard, a new button Add Widget appears, where the type can be selected based on requirement.

   Widget selection is done like this –



Units ⓘ Data units (if any)

Range Values ⓘ 0 100 +

✖ 70 90 🟨

✖ 90 100 🟥

Plate Color ⓘ #ffffff

Text Color ⓘ #1E313E

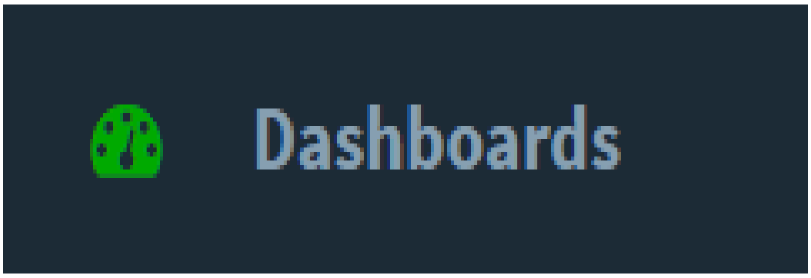
Tick Color ⓘ #000000

Major Ticks ⓘ 10

Show Value ⓘ ●

Fig 4.2 Illustration of widget selection

3. To manage all our dashboards, it is necessary to access to the Dashboards section, by clicking in the following menu item :



Then click on the Add Dashboard button that will open a new interface for entering the dashboard details.

Now we need to set up a few metrics:

- Id: A distinct identifier for the dashboard.
- Name: An illustrative name of the dashboard, in a more friendly way than its identifier.
- Dashboard description: Fill here any description or detailed information you need to keep about the dashboard.

4. Creating a data bucket – it is a conceptual storage for time series info over time. This data can be plotted onto dashboards, or distributed in various formats for offline processing. Data buckets feature and Add Bucket needs to be done, in order to create a data bucket.

Here are some dashboard layouts created using the platform :-

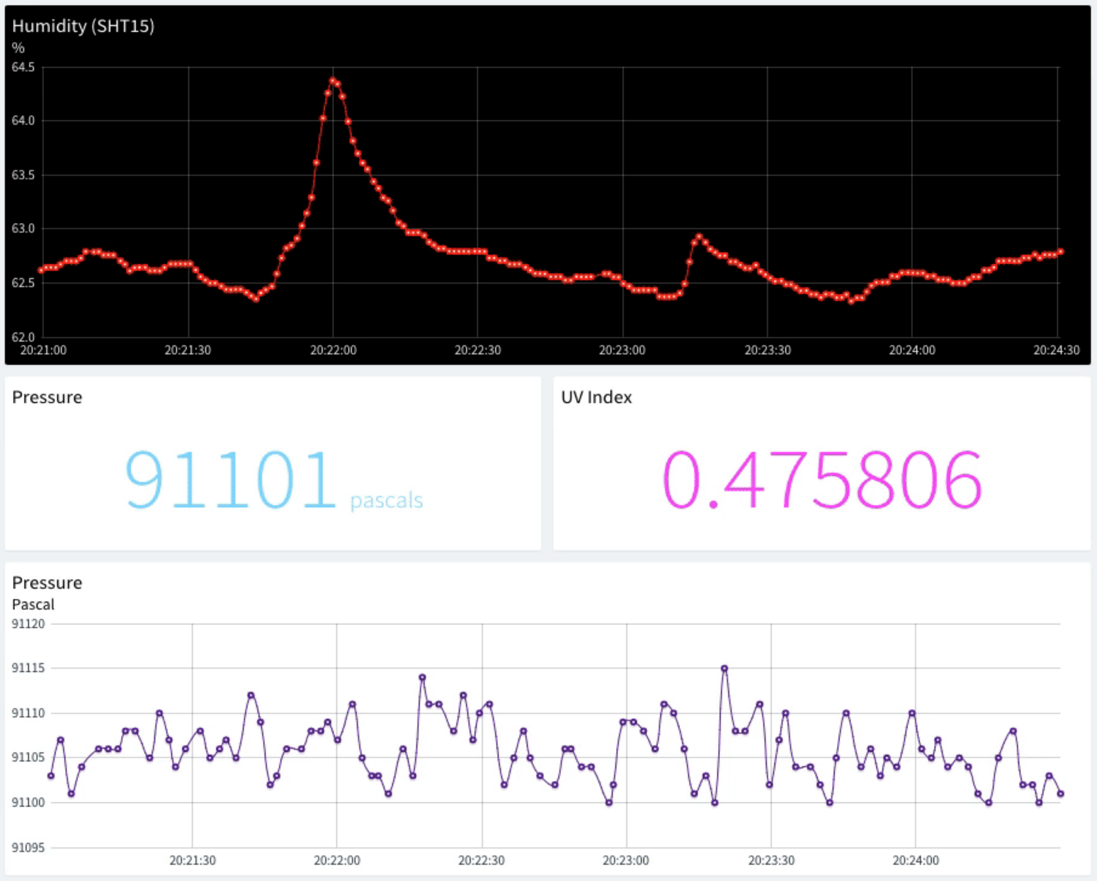


Fig 4.3 Dashboard layout

Here is an example dashboard with some widgets defined, like time series charts, donut charts, maps, or single values.

Once created, dashboards can be shared with third parties through a link or configured as templates to analyse data from different devices of the same type.

The following subsections describe the different parameters for each widget type.

Time Series Chart

A time-series chart is a graph that can show variation in values with respect to time. It is quite useful to display time-series data, like temperature variable that changes over time. It is possible to plot a single variable or multiple values in the same chart.

Donut Chart

A donut chart is a graph that can display a value, normally in form of a rounded percentage. In this sense, this is quite useful when you have a know variable that oscillates between a maximum and minimum value. In this case, it is only possible to only represent a single variable, that can be both updated in real-time from a device, or from a data bucket.

Text/Value

The text/value widget is an useful widget to display any arbitrary data, specially text values that cannot be represented with other widgets. As any other widget, can display data both from connected devices or data buckets.

Clock

This widget is just a clock widget that can display the current time both in the local time zone or in UTC, which can be useful when monitoring processes in real-time. This widget takes the current time just from our computer.

A Dashboard Tab is an additional work page that can be added to a dashboard to organize the visualization of data and simplify navigation between related pannels. The widgets and data sources of each tab can be completely independent of the others but all the tabs will share the same configuration settings (column number, background image, widgets border-radius, etc).

This feature also has the advantage of keeping all the tabs of a dashboard open even if they are not being visualized, so the data of the devices shown in real-time will not be lost when changing from one tab to another.

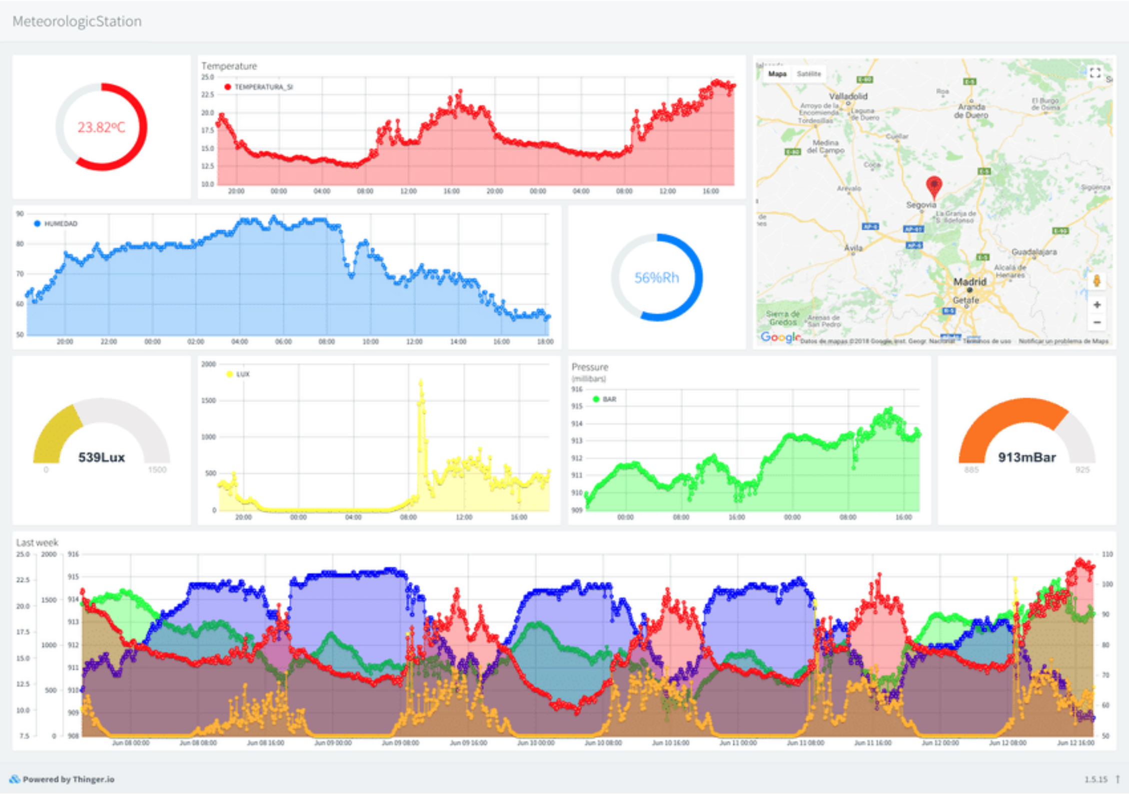
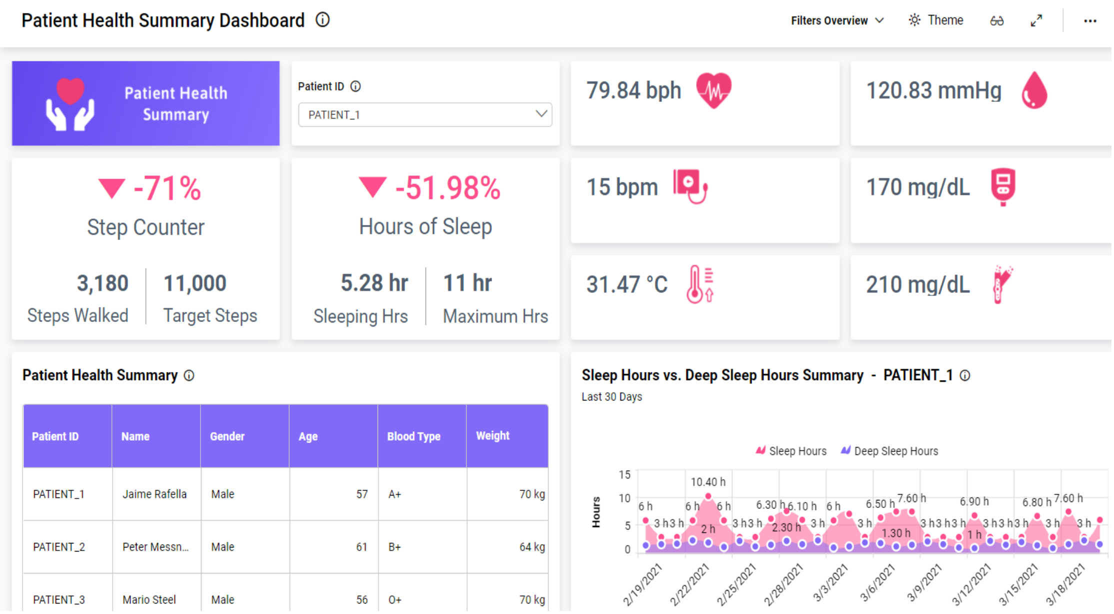


Fig 4.4 (a), (b) An example of how a dashboard monitor looks like


Patient Health Summary Dashboard ⓘ

Filters Overview ⌄ Theme

Patient Health Summary

Patient ID ⓘ

PATIENT_1

79.84 bph

120.83 mmHg

▼ -71%
Step Counter

3,180 Steps Walked | 11,000 Target Steps

▼ -51.98%
Hours of Sleep

5.28 hr Sleeping Hrs | 11 hr Maximum Hrs

15 bpm

170 mg/dL

31.47 °C

210 mg/dL

Patient Health Summary ⓘ

Patient ID	Name	Gender	Age	Blood Type	Weight
PATIENT_1	Jaime Rafella	Male	57	A+	70 kg
PATIENT_2	Peter Messn...	Male	61	B+	64 kg
PATIENT_3	Mario Steel	Male	56	O+	70 kg

Sleep Hours vs. Deep Sleep Hours Summary - PATIENT_1 ⓘ

Last 30 Days

(b)

CHAPTER 5

IOT - BLYNK

OBJECTIVE:- Using Blynk interface to create an user friendly GUI for the patient parameters

IDEATION:-

The parameters that are being monitored by the different sensors are visualized remotely via the Blynk app. This allows a quality GUI experience for the customers i.e. the relatives and family members of the patient. By suing the Blynk app anyone who is shared with the QR code can download the app and it will contains the required project for metrics monitoring. This approach will reduce the amount of panic within family members and will provide a suitable environment and time duration for the best possible.

5.1 About Blynk Community :-

Blynk was originally developed to cope up with the emerging field of Internet of Things(IOT). It can control the hardware remotely, can display the data of the sensor , capable of storing data, visualize it and various other important tasks.

There are three major components in the Blynk platform:

- **Blynk App** - It allows us to create good interactable interfaces for our projects using different kind of widgets
- **Blynk Server** - It is responsible for all the communications between the android smartphone and hardware. We can use our Blynk Cloud or run our private Blynk server locally. It is an open-source platform that can easily handle thousands of devices . It can also be launched on a Raspberry Pi

- **Blynk Libraries** - It contains all the popular hardware platforms - enabling communication with the server and thereby processing all the incoming and outgoing commands.

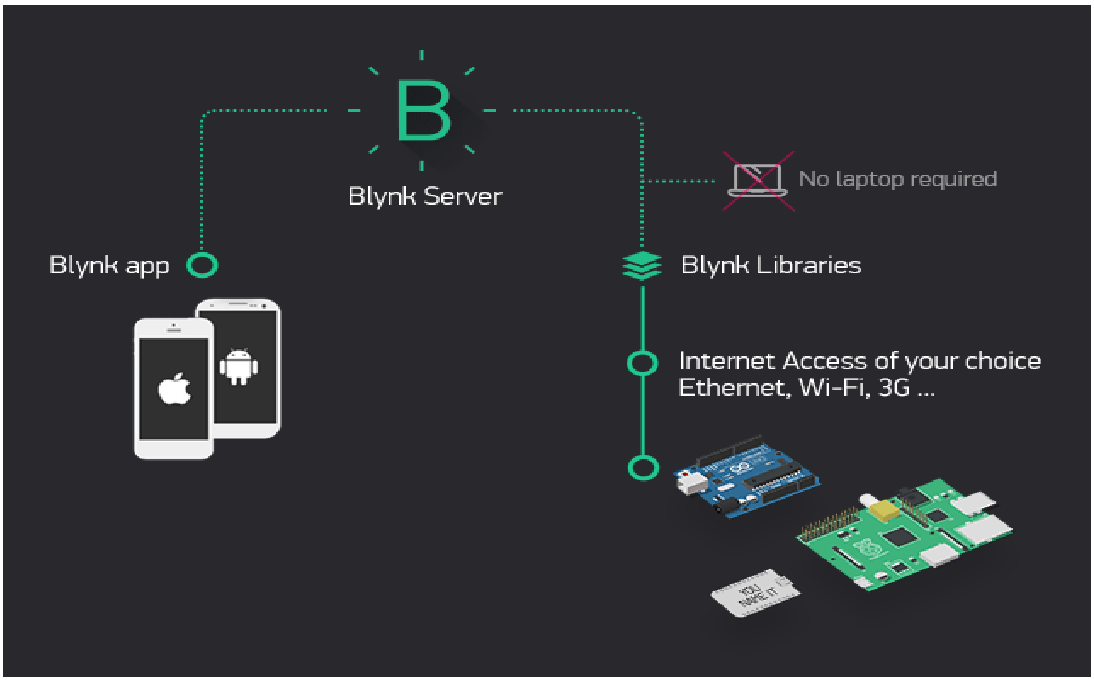


Fig 5.1 Organization of Blynk cloud

5.2 Blynk Features:-

- It has a similar API & UI for all the hardware & devices that it supports.
- It has the provision of connecting to the cloud by using:
- WiFi
- Bluetooth and BLE
- Ethernet
- USB (Serial)
- GSM
- It has a set of easy-to-use different widgets
- It has direct pin manipulation which requires no code to be written
- It is easily integrable and we can add new functional properties by using virtual pins
- It can monitor history data by the help of the SuperChart widget
- It is also capable of device-to-device communication using the Bridge Widget

- Blynk can be useful in sending emails, tweets, notifications, etc.

5.3 Introduction to Blynk account and interface:-

1. Create a Blynk Account

As we have downloaded the Blynk Android App, now we need to create a New Blynk account. This account that we have created is totally a separate account from that which is used for the Blynk Forums. It is recommended to use a **real** email address to simplify the procedures further.

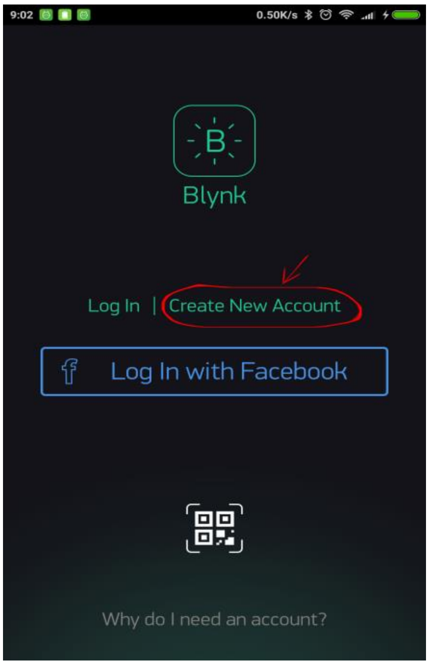


Fig 5.2 Illustration of account creation

2.Create a New Project

As we have successfully logged into our account that we have created, now we will proceed by creating a new project in the Blynk Android App.

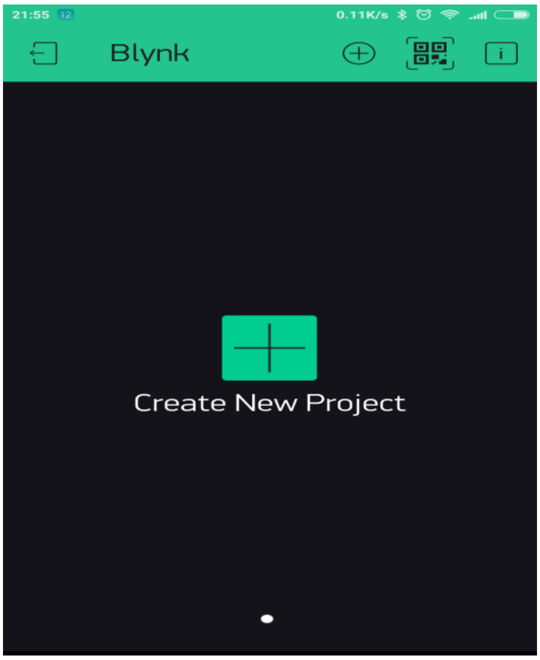
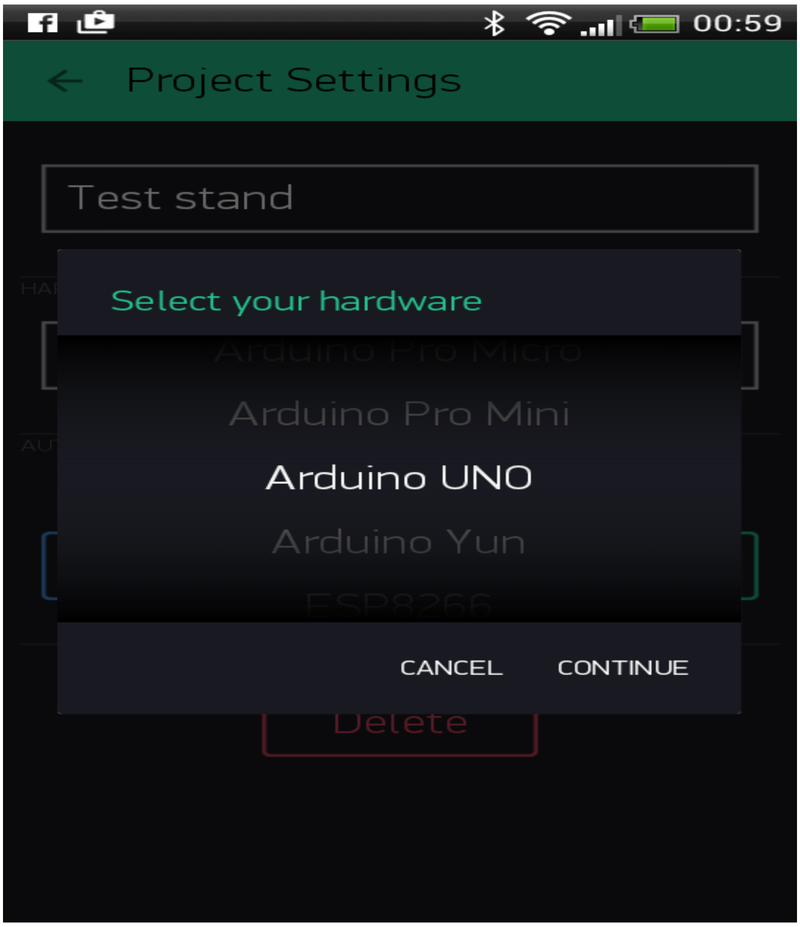


Fig 5.3 Creation of project

3. Choose Your Hardware

Select the hardware tools we want to use



Test stand

Arduino UNO

Fig 5.4 Selection of hardware

4. Auth Token

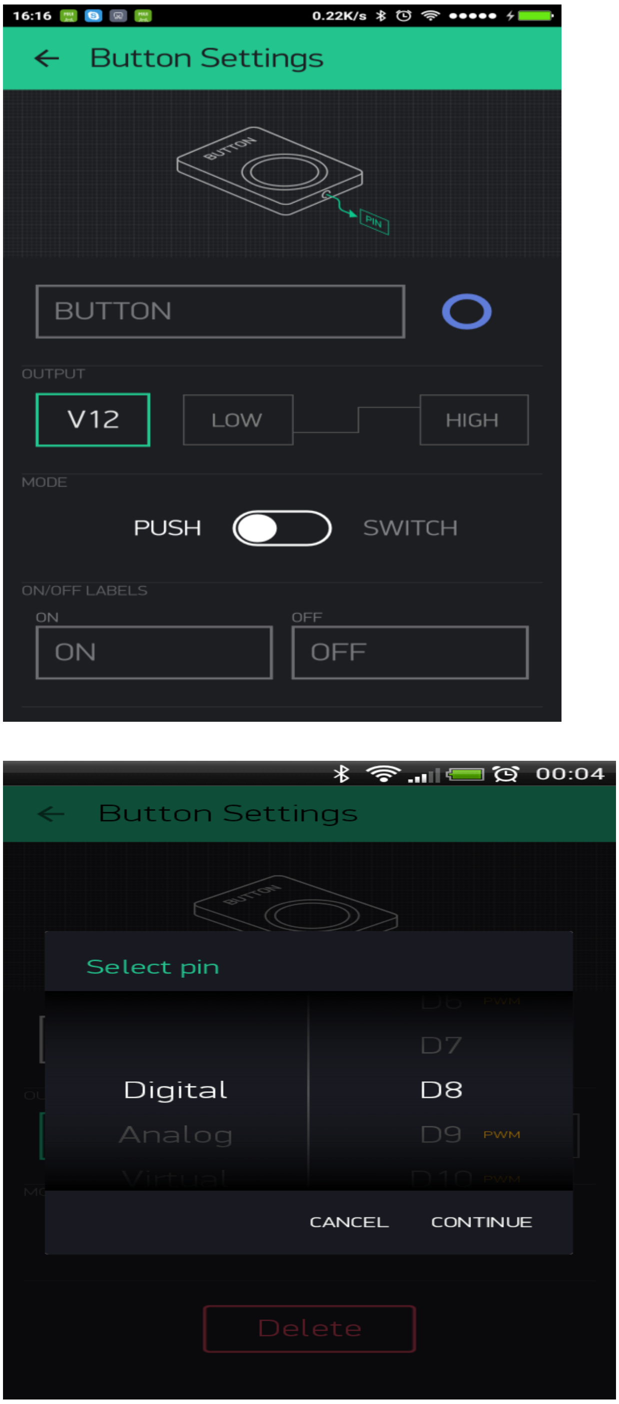
It is a distinct identifier needed to connect the hardware to the android smartphones. Each and every new project that we will create has its own unique Auth Token. We’ll receive Auth Token automatically on our email after we have created our project.We need the auth token for setting up a link for our project. For that we need to go to the devices section and choose the devices as per our requirement

5. Add a Widget(as required):

We need to add a button to our empty project canvas to get control over our LED. Tap anywhere on the canvas to open the widget box. All the available widgets are placed here. Just pick a suitable button as per requirement.

Drag and Drop feature - Tap and hold the selected widget to drag it to the new location

Widget Settings - Each of the widget has its own customised settings. Single tap on the widget to get access to these settings.



16:16 0.22K/s

Button Settings

BUTTON

OUTPUT

V12 LOW HIGH

MODE

PUSH SWITCH

ON/OFF LABELS

ON: ON

OFF: OFF

00:04

Button Settings

Select pin

Digital / Analog / Virtual

D6 PWM / D7 / D8 / D9 PWM / D10 PWM

CANCEL CONTINUE

Delete

Fig 5.5 Widget selection

6. Run The Project

After making necessary changes to the Settings then we will press the **PLAY** button. On pressing this button it will switch us from the Edit mode to Play mode where we can now interact with the specified hardware. While we are in the PLAY mode, now we cannot set up any new widgets, we need to press **STOP** and go back to the EDIT mode.There we will get a message telling us that the “Arduino is offline”.

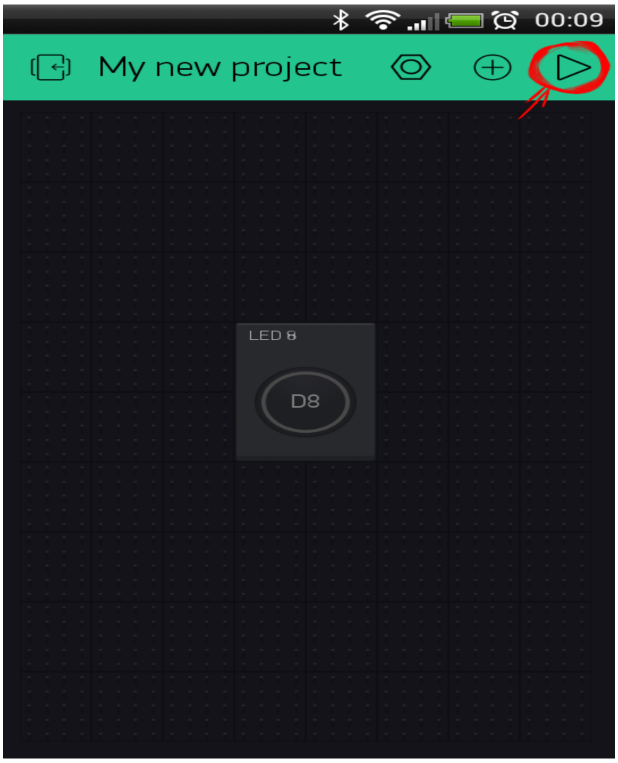


Fig 5.6 Running the project

5.4 Blynk application in metrics monitoring

With the Blynk we are able to plot the graphical representation of data which will be given by the sensors that we are using,.

Blynk uses **virtual pin** concept through which we can push data from the nodemcu to the blynk **mobile app widgets** and also get the data from the widgets to the nodemcu via the ESP8266 Wi-Fi module present. **Virtual Pin** is a concept invented by Blynk Inc. to provide exchange of any data between hardware and Blynk mobile app.

Operation/ Working:-

1. Open Blynk App on mobile and create an account.

2. Create a New Project

3. Select the working board (here NODEMCU)

4. Select the widget as per necessity (here it will be Super chart and Value Display widgets)

5. Tap the widgets for editing its properties. We can use digital or analog or virtual pins as per requirement in the project.

6. After uploading the sensors data in the NODEMCU from Arduino IDE we can monitor the patient parameters on the widgets on the smartphone at a remote distance.

The thing to understand here is every sensor connected to Nodemcu will send its data which we will store in a value and send it to the blynk app via the virtual pin (V1 for eg.)

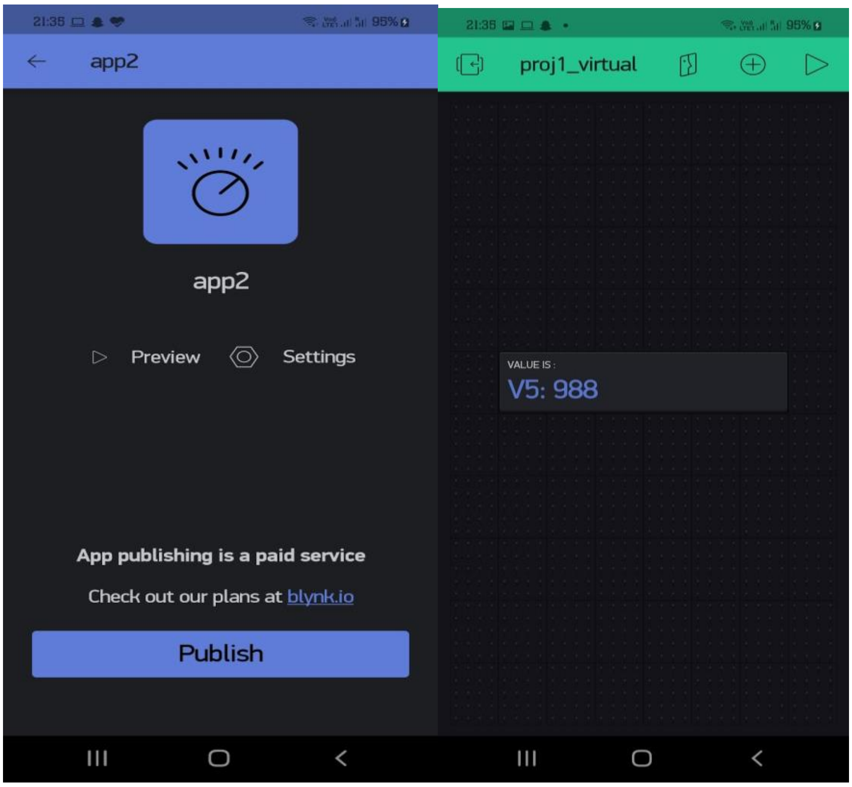
Let us store the incoming data from the pulse sensor (for eg.) in a variable name : **pulse_val**

```
Blynk_Write(V1){

Blynk.VirtualWrite(pin,pulse_val);

}
```

5.5 Blynk Community (My Apps)

- Blynk community has recently launched My Apps for blynk projects
- We need to enter the app name ,icon style ,theme ,color and other attributesfor the app to look smart.
- Then we need to choose the projects we want to access inside the app.
- Blynk community comes up with a exciting features of BlynkFaces which allows us to include multiple projects inside our customised app.
- Then we need to select the type of auth token (Dynamic or static)

- Static auth token is used for devices that connect to the internet over the Ethernnet cable or cellular network.
- Auth token is sent to the registered email id in the project itself
- We can publish this app also for others to use and monitor my project readings.



App publishing is a paid service

Check out our plans at blynk.io

Fig 5.7 Blynk MyApp Layout

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Summary:-

Health-care is given extreme importance nowadays by each country with the advent of the novel corona-virus. Thus in this regard, an IoT controlled healthcare monitoring system is most probably the best solution for such an epidemic. Internet of Things (IoT) is the new revolution of the internet which is a growing research field, especially in health related services..

“Smart IoT based health-care system ” is the project where we have mainly focused on two objectives first one was a smart health monitoring system, to collect the health history of patients with a unique ID and store it in a database so that doctors need not spend much of their time in search of the report and give analysis right from the dashboard. Any health-care that is being done will be updated and reflected in the dashboard itself. And the other one is the Wearable-Sensor-Based Fall Detection System for aged people, to monitor the movements of them, and recognize a fall from normal daily activities by using sensors, and automatically sends a request or an alert for help to the caregivers so that they can pick up the patient.

However, there are few shortcomings to this too. A basic knowledge of the operation is to be learnt by the caregivers . Also both the caregiver and the wearer should know how to protect the sensors from water damage or any physical damage .

6.2 Future Scope:-

Though the medical sector was unhurried in the first stage to adopt IoT technology as compared to other sectors, the new uproar Internet of Medical Things (IoMT) proved to be revolutionary in today’s world. It is set to transform how to keep people healthy and safe, while bearing costs in mind.

Internet of Medical Things (IoMT) is a combination of medical devices and applications that connect healthcare IT systems via different network technologies. The technology can lower needless visits to hospitals along with decreasing the load on the health care sector by interconnecting patients and physicians. Also it makes secure transmission of medical data possible.

As per experts, the global IoT market was worth $22.5 billion in 2016 and is expected to reach $72.02 billion by 2021. Contributing greatly to the healthcare domain, the IoMT market is filled with smart devices like medical/vital monitors at home, community, clinic along with wearables, while connecting real-time location, telehealth, and other services.

Some of the ways in IOT in medical services can help in upcoming time are :

- Reducing waiting time in emergency rooms
- Tracking of patients, inventory and staff
- Augmenting drug management techniques
- Ensuring the provision of essential hardware
- Converting data into activities

(IOMT) benefits include :

REDUCTION IN ERRORS:

The data that is generated from the connected devices helps in taking effective, accurate decisions and also ensures smooth operations with lower costs and wastage.

FASTER DIAGNOSIS : Due to continuous monitoring and real-time tracking of patient data, doctors can diagnose/detect disease at an early stage for effective treatment.

COST EFFICIENT : Usage of IoT devices enables doctors to monitor patients in real-time. Thus, this process can help patients cut down unnecessary visits to doctors and stay at hospitals..

REMOTE MEDICAL ASSISTANCE : For lone patients in a medical emergency, it is quite impossible to contact a doctor who is far away. IoT applications in healthcare and other related devices make it possible to do that easily.Moreover, health personnel can also check the patients on-the-go.

ALERTS AND TRACKING : On-time alerts become quite critical in life-threatening situations. To combat those kinds of situations, IoMT devices and applications can gather essential data and transfer it to medical personnel for real-time tracking. Also, these devices can send notifications regarding the conditions of the patient irrespective of place or time.

REMOTE REPORTING AND MONITORING : Connected devices makes it easier for doctors and physicians to monitor patients' health. Real-time monitoring can save lives in an emergency such as diabetic attacks, asthma attacks, heart failure,etc. Through the use of a smart medical device connected to the smartphone app, collection of medical and other necessary health data is not tough. Collected data from these devices can easily be accessed and used by an authorized person, who could be a doctor, insurance company, participating health firm or an external consultant, regardless of their location, time, or device.

As per the latest innovations and requirements in medical technologies, we must form an organized network of smart devices which would collect, sync and manage huge amounts of data effectively and cost efficiently.

In future the scope of automation in health industries is regarded as one of the most revolutionary schemes ever deployed in the healthcare industries. We look forward to this upcoming boost in the infrastructure of the healthcare structure to build a greater nation.

6.3 Cost Analysis-

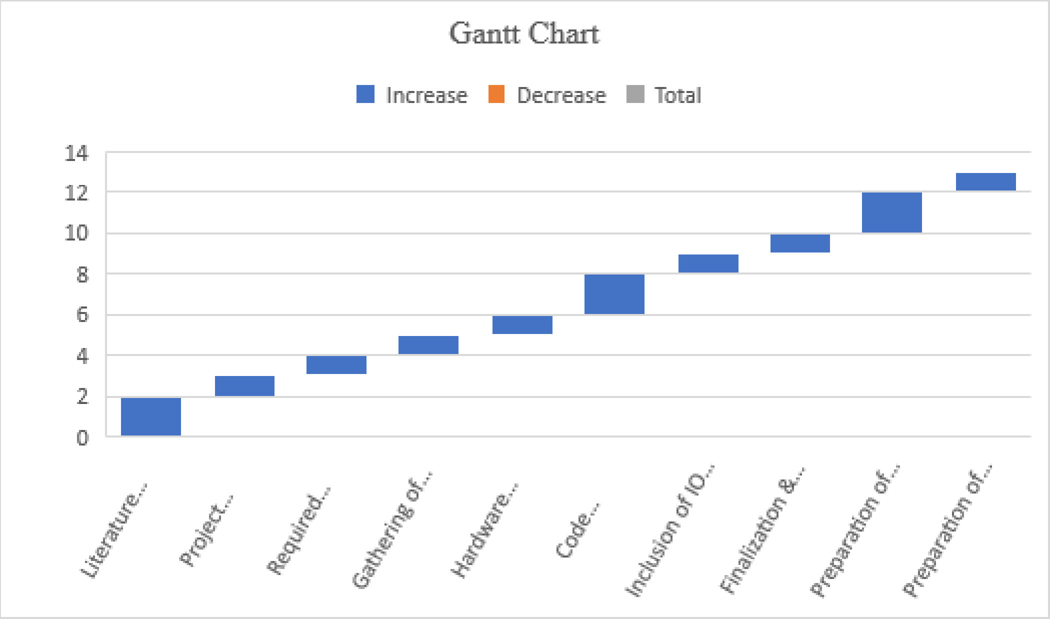
Equipment	**Cost**
MLX 90614 Temperature Sensor.	Rs. 1299
MAX 30102 Pulse Rate Sensor.	Rs. 699
MPU6050 sensor board	Rs. 213
NodeMCU ESP8266 Wi-Fi development board	Rs. 450
Jumper wires	Rs. 199
Total	Rs. 2860

CHAPTER 7
PLANNING & REFERENCES

7.1 Planning and project management

S.No.	Activity	Starting Week	Number of Weeks
1.	Literature Review	1st week of January	2
2.	Project Finalization	3rd week of January	1
3.	Required software setup	4th week of January	1
4.	Gathering of Hardware & Formation of codes	1st week of February	1
5.	Hardware assembly calibration	2nd week of February	1
6.	Code Integration & Debugging	3rd week of February	2
7.	Inclusion of IOT and Blynk	1st week of March	1
8.	Finalization & modification of Website and mobile application	1st week of March	1
9.	Preparation of project report	2nd-3rd week of March	2
10.	Preparation of Project presentation	4th week of March	1

The Gantt Chart is shown below :-



7.2 REFERENCES –

[1]. Md. Milon Islam, Ashikur Rahaman and Md. Rashedul Islam. 2020. Development of Smart Healthcare Monitoring System in IOT Environment *Springer Nature Singapore Pte Ltd*.

[2]. Deepak N, Rajendra Prasad C H and Sanjay Kumar Ch. 2018. Patient Health Monitoring using IOT *International Journal of Innovative Technology and Exploring Engineering* 8.

[3]. Hari Kishore K, Surendra Nath K V, Hari Krishna K V N, Pavan Kumar D, Manikanta V and Fazal Noor Basha. 2019. IOT Based Smart Health Monitoring Alert Device *International Journal of Innovative Technology and Exploring Engineering* 8.

[4]. Tamilselvi V, Sribalaji S, Vigneshwaran P, Vinu P, GeethaRamani J. IoT based health monitoring system. In: 2020 6th International conference on advanced computing and communication systems (ICACCS). IEEE; 2020. p. 386–9

[5]. Riazul Islam SM, Kwak Daehan, Humaun Kabir M, Hossain M, Kwak Kyung-Sup. The Internet of Things for health care: a comprehensive survey. *IEEE Access. 2015;3:678–708.*

[6].Li, Q.; Stankovic, J.A.; Hanson, M.A.; Barth, A.T.; Lach, J.; Zhou, G. Accurate, fast fall detection using gyroscopes and accelerometer derived posture information. In Body Sensor Networks; International Workshop on Wearable & Implantable Body Sensor Networks: Berkeley, CA, USA, 2009; pp. 138–143.

[7].Wang, J., Zhang, Z., Li, B., Lee, S., Sherratt, R.An enhanced fall detection system for elderly person monitoring using consumer home networks IEEE Transactions on Consumer Electronics2014601232910.1109/TCE.2014.67809212-s2.0-84898032338
[8]. Islam M S, Islam M T, Ali F. Almutairi, GankokBeng, Amid N M N. 2019. Monitoring of the Human Body Signal through the Internet of Things (IoT) Based LoRa Wireless Network *System Journal of Applied Sciences 9*

[9]. Desai MR, Toravi S. A smart sensor interface for smart homes and heart beat monitoring using WSN in IoT environment. *In: 2017 International conference on current trends in computer, electrical, electronics and communication (CTCEEC). IEEE; 2017. p. 74–7*

[10]. Prakash H Patil, Pratyush Singh, Swatee Biradar and Prasad Rane. 2013. Design and Implementation of Wireless Patient Health Monitoring System *International Journal of Engineering Research and Technology 2* (6).

[11]. He, J.; Hu, C.; Wang, X.Y. A Smart Device Enabled System for Autonomous Fall Detection and Alert. Int. J. Distrib. Sens. Netw. 2016, 12, 1–10

[12]. Wang, J.; Zhang, Z.; Li, B.; Lee, S.; Sherratt, R.S. An Enhanced Fall Detection System for Elderly Person Monitoring using Consumer Home Networks. *IEEE Trans. Consum. Electron.* **2014**, *60*, 23–28.

[13]. Rahaman A, Islam M, Islam M, Sadi M, Nooruddin S. Developing IoT based smart health monitoring systems: a review. *Rev Intell Artif. 2019;33:435–440. doi: 10.18280/ria.330605.*

[14]. Becker, C.; Schwickert, L.; Mellone, S.; Bagalà, F.; Chiari, L.; Helbostad, J.L.; Zijlstra, W.; Aminian, K.; Bourke, A.; Todd, C.; et al. Proposal for a multiphase fall model based on real-world fall recordings with body-fixed sensors. In *Zeitschrift Für Gerontologie Und Geriatrie; Springer International Publishing: Berlin, Germany,* 2012; pp. 707–715.

[15]. Bourke, A.K.; Van De Ven, P.; Gamble, M.; O’Connor, R.; Murphy, K.; Bogan, E.; McQuade, E.; Finucane, P.; O´Laighin, G.; Nelson, J. Evaluation of waist-

mounted tri-axial accelerometer-based fall-detection algorithms during scripted and continuous unscripted activities. *J. Biomech.* **2010**, *43*, 3051–3057.

[16]. Vavoulas, G.; Pediaditis, M.; Spanakis, E.G.; Tsiknakis, M. The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones. In Proceedings of the IEEE 13th International Conference on Bioinformatics and Bioengineering, Chania, Greece, 10–13 November 2013; pp. 1–4.

[17]. F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, “Barometric pressure and triaxial accelerometery-based falls event detection,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 18, no. 6, pp. 619–627, 2010

[18]. S. D. Bersch, C. M. J. Chislett, D. Azzi, R. Kusainov, and J. S. Briggs, “Activity detection using frequency analysis and off the-shelf devices: fall detection from accelerometer data,” in Proceedings of the 5th International Conference on Pervasive Computing Technologies for Healthcare and Workshops (Pervasive Health ’11), pp. 362–365, Dublin, Ireland, May 2011.

SELF DECLARATION FOR PLAGIARISM CHECK

We, Souvik Karmakar(1807228), Indrashis Mitra(1807274), Kinjal Sarkar(1807277) and Pratyay Basu(1807291) are declaring that our Project report on "**HEALTHCARE MONITORING AND MANAGEMENT SYSTEM**" has plagiarism well within the limits prescribed to us. We take the full responsibility of it.