VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI, KARNATAKA, INDIA

NITTE EDUCATION TRUST

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

An Autonomous Institution with A$^+$ Grade UGC by NAAC UGC, Approved by UGC, AICTE, Government of Karnataka, Yelahanka, Bengaluru-560064, Karnatka, India.

Mini Project Report On

"IoT Gateway using Raspberry Pi"

A Mini Project report submitted in partial fulfillment of the requirement for the award of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

2021-2022

Submitted By

S Bharath	**1NT18EC134**
Shaik Abdul Aleem	**1NT18EC143**

Under the Guidance of

Prof. Sitaram Yaji
Professor
Department of Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology
Yelahanka, Bangalore-560064

NITTE EDUCATION TRUST | NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

An Autonomous Institution with A^+ Grade UGC by NAAC UGC, Approved by UGC, AICTE,
Government of Karnataka, Yelahanka, Bengaluru-560064, Karnatka, India.

Department of Electronics and Communication Engineering

Certificate

*This is to certify that **S Bharath (1NT18EC134) , Shaik Abdul Aleem (1NT18EC143)** has submitted the Mini Project report entitled **"IoT Gateway using Raspberry Pi"** in fulfillment for the award of Bachelor of Engineering in Electronics and Communication Engineering from Visvesvaraya Technological University, Belagavi during the year 2021-2022. It is certified that all the corrections, suggestions indicated for internal assessment have been incorporated in the report. The Mini Project report has been approved as it satisfies the academic requirements in respect of work prescribed for the aforesaid degree.*

Prof. Sitaram Yaji	Dr. Ramachandra A C	Dr. H. C. Nagaraj
Guide	**HOD**	**Principal**

External Viva Voce

Name of Examiners **Signature with Date**

1) ____________________

2) ____________________

ACKNOWLEDGEMENT

It is my proud privilege and duty to acknowledge the kind help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without valuable suggestions, cooperation and guidance.

I wish to record my sincere gratitude to Management, and **Dr. H. C. Nagaraj,** Principal, Nitte Meenakshi Institute of Technology, Bengaluru for the permission provided to accomplish this Mini Project.

My sincere thanks to **Dr. Ramachandra A. C,** Professor and Head, Department of Electronics and Communication Engineering for his valuable suggestions and guidance.

My sincere gratitude to **Dr. Rajesh N, Dr. Sunil S. Harakannanavar** and **Dr. Thimmaraja Yadava G,** Mini Project Coordinators**,** Department of Electronics and Communication Engineering, for their valuable suggestions in preparing this report.

I express my sincere gratitude to our beloved guide**, Mr. Sitaram Yaji,** Professor, Department of Electronics and Communication and Engineering for his/her support and guidance.

I am extremely great full to faculty members of Department of Electronics and Communication Engineering, and friends for their support and encouragement in successful completion of this Mini Project.

Name of Student	**USN**
S Bharath	**1NT18EC134**
Shaik Abdul Aleem	**1NT18EC143**

TABLE OF CONTENTS

List of Figures :

ABSTRACT

The Internet of Things (IoT) is a network of physical objects that are fitted with sensors, software and other technologies. Connected to the Internet, these ‘things’ are able to exchange real time data with other connected devices and systems over networks. These connected devices combine with automated systems to gather IoT data that can be analyzed to assist with tasks or learn how to improve a process.

IoT Gateways are emerging as an essential component in building a robust IoT and for delivering computational power in edge computing scenarios. Edge computing distributes the load on a system by performing data processing at the data source, or “edge”, rather than relying on a central server for the bulk of the work.

In this project we have used credit card sized single board computer Raspberry Pi as IoT gateway for temperature monitoring and control as application using DHT11 sensor and Relays. We have used Google Firebase as Cloud Storage. Through an Android App we can monitor the temperature and control the relays in 2 modes (Manual/Automatic). The obtained results are presented.

1. INTRODUCTION

IoT GATEWAY:

There's one device that is key to tying internet of things systems together. The IoT gateway. An loT   gateway device bridges the communication gap between IoT devices sensors, equipment, systems and the cloud. By systematically connecting the field and the cloud, IoT gateway devices offer local processing and storage solutions, as well as he ability to autonomously control field devices based on data input by sensors.

Overview:

An loT gateway performs several critical functions from translating protocols to encrypting processing, managing and filtering data. If you imagine an loT ecosystem, a gateway sits between devices and sensors to communicate with the cloud. Refer fig l.

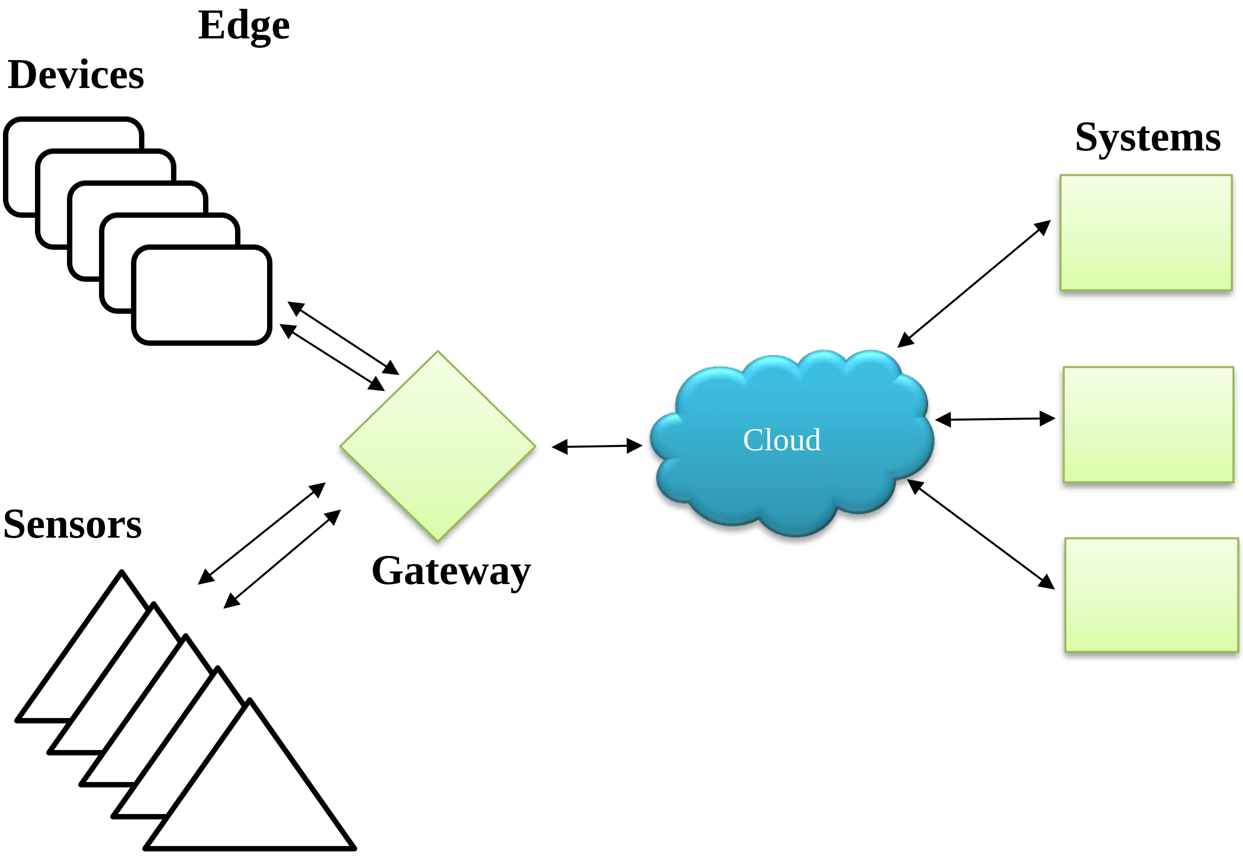


Fig 1 : IoT Gateway

2. LITERATURE SURVEY

Title	Author Name	Citation	Algorithm / Techniques Used	Conclusion
Mobile wireless sensor network gateway: A raspberry Pi implementation with a VPN backend to OpenStack	Eduard-Florentin Luchian; Adrian Taut; Iustin-Alexandru Ivanciu; Gabriel Lazar; Virgil Dobrota	E. Luchian, A. Taut, I. Ivanciu, G. Lazar and V. Dobrota, "Mobile wireless sensor network gateway: A raspberry Pi implementation with a VPN backend to OpenStack," *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2017, pp. 1-5, doi: 10.23919/SOFTCOM.2017.8115561.	**Openstack Cloud System , MWSNG Cloud Connectivity Through Vpn**	A Raspberry Pi device, acting as Compute Node in a private cloud proved to be a feasible solution to act as gateway in a mobile wireless sensor network and to provide seamless connectivity over various access technologies.
Implementation of IoT based PV monitoring system with message queuing telemetry transfer protocol and smart utility network	Chang-Sic Choi; Jin-Doo Jeong; Jinsoo Han; Wan-Ki Park; Il-Woo Lee	C. -S.Choi, J. -D. Jeong, J. Han, W. -K. Park and I. -W. Lee, "Implementation of IoT based PV monitoring system with message queuing telemetry transfer protocol and smart utility network," *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, pp. 1077-1079, doi: 10.1109/ICTC.2017.819085.	**Message Queuing Telemetry Transfer (MQTT) Smart Utility Network(SUN)**	This paper introduces IoT gateway based on raspberry Pi, MQTT protocol and SUN communication, and introduces the implementation of IoT based PV monitoring system that monitors PV panel information.
Design and Development of Modbus/MQTT Gateway for Industrial IoT Cloud Applications Using Raspberry Pi	Changqing Sun; Kun Guo; Zhaoxia Xu; Jianhui Ma; Dairong Hu	C. Sun, K. Guo, Z. Xu, J. Ma and D. Hu, "Design and Development of Modbus/MQTT Gateway for Industrial IoT Cloud Applications Using Raspberry Pi," *2019 Chinese Automation Congress (CAC)*, 2019, pp. 2267-2271, doi: 10.1109/CAC48633.2019.8997492.	**MQTT protocol, Modbus, Amazon AWS, Alibaba Cloud**	A prototype gateway using inexpensive credit-sized mini-computer Raspberry Pi and a plug-in printed circuit board with integrated circuit chips or components for data and signal processing. Powered by the MQTT messaging protocol and JSON data format.

LITERATURE SURVEY

Title	Author Name	Citation	Algorithms / Techniques Used	Conclusion
A Performance Evaluation of Raspberry Pi Zero W Based Gateway Running MQTT Broker for IoT	Diana Bezerra Correia Lima; Rubens Matheus Brasil da Silva Lima; Douglas de Farias Medeiros; Renata Imaculada Soares Pereira; Cleonilson Protasio de Souza	D. B. C. Lima, R. M. B. da Silva Lima, D. de Farias Medeiros, R. I. S. Pereira, C. P. de Souza and O. Baiocchi, "A Performance Evaluation of Raspberry Pi Zero W Based Gateway Running MQTT Broker for IoT," *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 0076-0081, doi: 10.1109/IEMCON.2019.8936206.	**Edge-computing gateway (GTW),IoT standard MQTT and MQTT-SN(For sensor networks) protocols**	A study about the performance evaluation of the Raspberry Pi Zero W working as an IoT gateway running MQTT protocol was described. The experimental results show that the QoS level chosen affects its performance but not in an extreme way.
UAV and IoT Integration: A Flying Gateway	Aya Moheddine; Fabio Patrone; Mario Marchese	A. Moheddine, F. Patrone and M. Marchese, "UAV and IoT Integration: A Flying Gateway," *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 121-122, doi: 10.1109/ICECS46596.2019.8965135.	**LoRa-based gateway, LTE cellular networks**	A new approach for the integration of UAVs in the IoT domain. To design, install and test a flying gateway system using Raspberry Pi and Rak2245 Pi-Hat boards, offering an opportunity for the users to reach IoT devices installed in remote areas out of the coverage of terrestrial gateways.
A Flexible/Scalable IoT Server Node testbed, from Gateway to Edge Computing. A Smart Home Use Case	S. Blionas ; G. Doukas; K. Doukas; N. D. Tselikas	S. Blionas, G. Doukas, K. Doukas and N. D. Tselikas, "A Flexible/Scalable IoT Server Node testbed, from Gateway to Edge Computing. A Smart Home Use Case," *2019 Panhellenic Conference on Electronics & Telecommunications (PACET)*, 2019, pp. 1-6, doi: 10.1109/PACET48583.2019.8956284.	**IoTSN Testbed , Zigbee Technology**	An implementation of a Raspberry IoTSN setup was demonstrated with custom IoT nodes controlled by an Android application.

LITERATURE SURVEY

Title	Author Name	Citation	Algorithms / Techniques Used	Conclusion
Design and Implementation of Cloud-based Single-channel LoRa IIoT Gateway Using Raspberry Pi	Changqing Sun; Fuquan Zheng; Guangxu Zhou; Kun Guo	C. Sun, F. Zheng, G. Zhou and K. Guo, "Design and Implementation of Cloud-based Single-channel LoRa IIoT Gateway Using Raspberry Pi," *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 5259-5263, doi: 10.23919/CCC50068.2020.9189480.	**LoRa, HTTP RESTful protocol and IoT-standard MQTT protocol.**	A prototype of LoRa-based IoT gateway using inexpensive credit-sized mini-computer Raspberry Pi and a plug-in printed circuit board with the necessary integrated circuits for data processing. The system consists of devices or nodes, gateways, and cloud services.
Design of a Smart Gateway for Edge Enabled IoT Applications	Azim Uddin Chowdhury; Mohammad Mamun Elahi	A. U. Chowdhury and M. M. Elahi, "Design of a Smart Gateway for Edge Enabled IoT Applications," *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 417-420, doi: 10.1109/TENSYMP50017.2020.9230843.	**MQTT and HTTP**	We have designed a smart IoT gateway that collects, stores, and analyses sensor data at the edge and reduces bandwidth required to send data to the cloud by utilizing lightweight protocols.
Design and Implementation of a Web Platform Prototype Based an IoT Gateway Using Raspberry Pi for Livestock Monitoring	Ana María Balladares Ocaña; Roger Idrovo Urgilés; José Antonio Soria Pérez	A. M. Balladares Ocaña, R. I. Urgilés and J. A. Soria Pérez, "Design and Implementation of a Web Platform Prototype Based an IoT Gateway Using Raspberry Pi for Livestock Monitoring," *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, 2021, pp. 1-6, doi: 10.1109/ETCM53643.2021.9590657.	**LoRa (Long Range) technology LoRa/GPS communication, HTTP RESTful protocol and the IoT standard MQTT protocol**	A prototype of the web platform for monitoring livestock in real-time based on LoRa technology, applied to the field of Smart Agriculture IoT, has been presented, including hardware and software design.

3. OBJECTIVES OF PROJECT

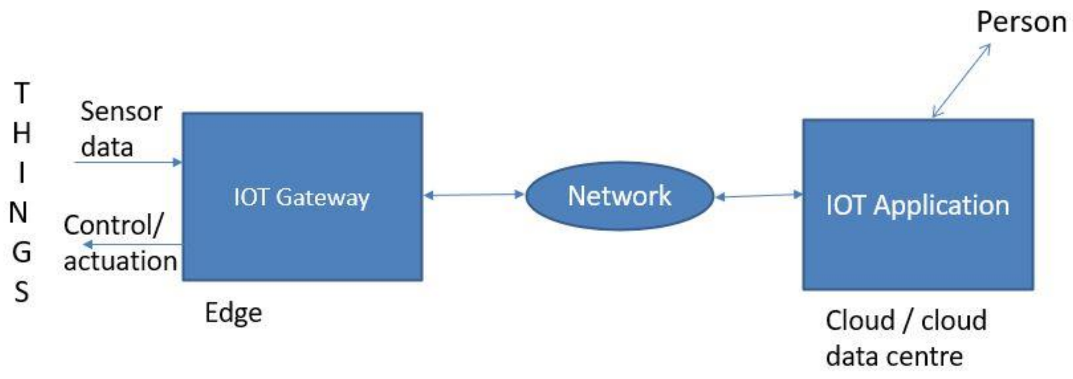
The following are the proposed objectives of the mini project

- To implement an IoT solution using cloud

- To use Raspberry Pi as Gateway

- To show control and actuation (edge, cloud)

4. BLOCK DIAGRAM / DESIGN METHODOLOGY



IOT Application - Visualization, IOT data analytics , informed action (automatic, manual, semi)

Fig 2 : IoT Solution using Cloud

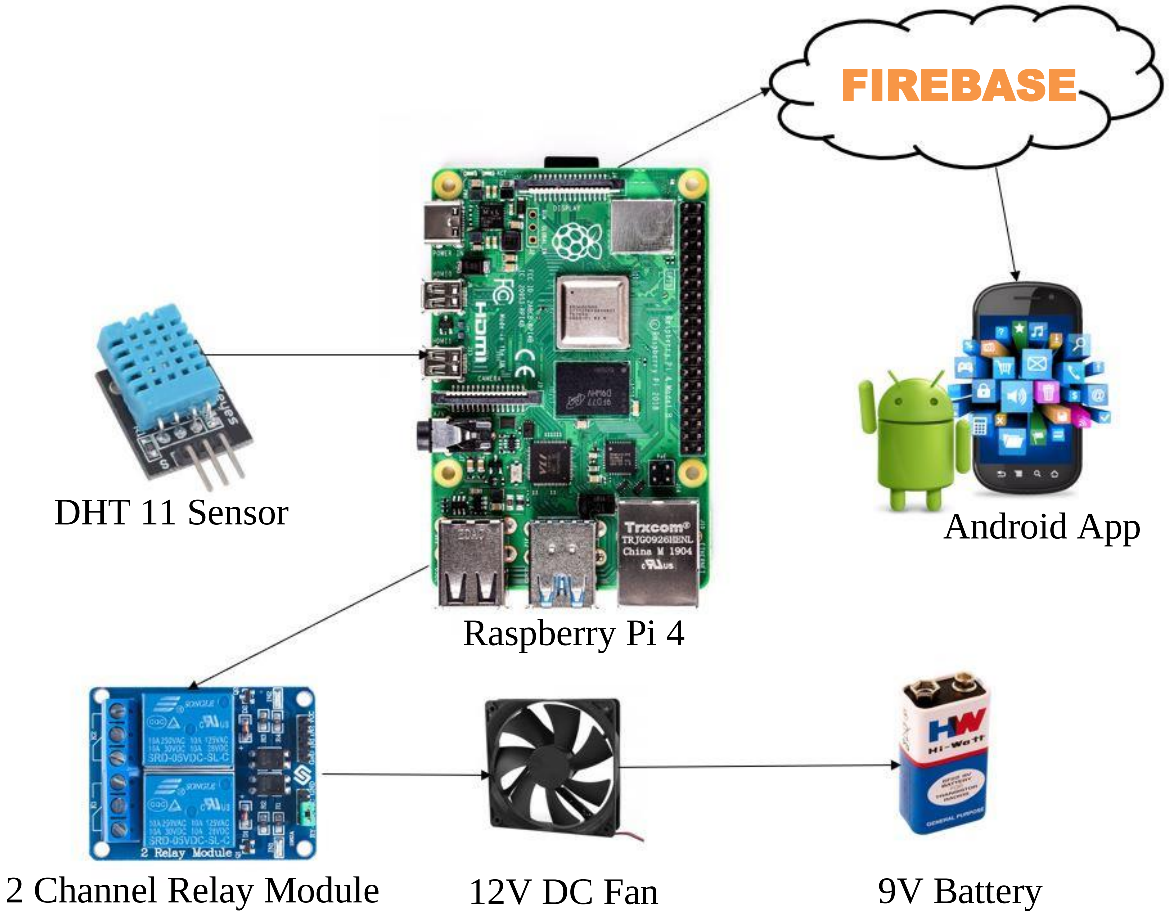


Fig 3 : IoT Gateway Application Architecture

CIRCUIT DIAGRAM

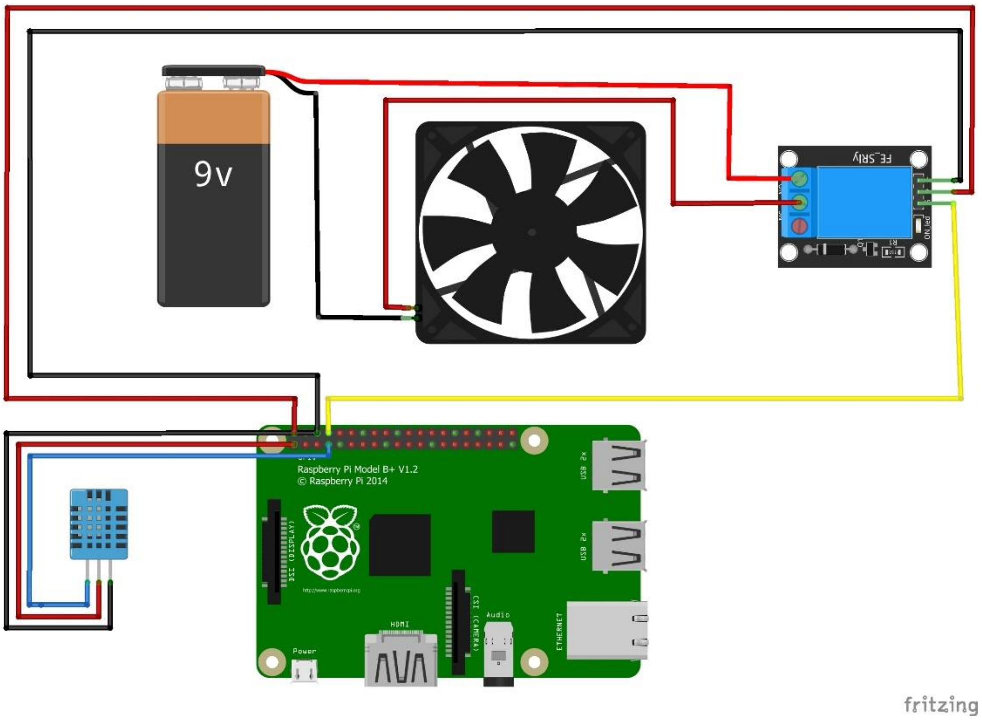


Fig 4 : Circuit Diagram of IoT Application

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

4.1 Raspberry Pi 4 Model B:

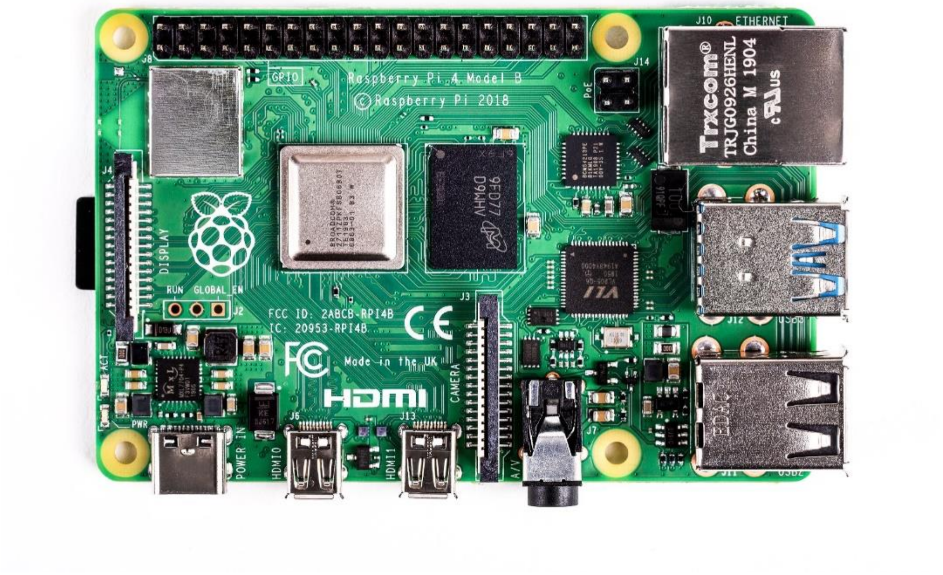


Fig 5 : Raspberry Pi 4 Model B

Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.

This product’s key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decode at up to 4Kp60, up to 8GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on).

The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

4.1.1 Specifications:

- Processor : Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- Memory : 4GB SDRAM
- Connectivity : 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless,
  Bluetooth 5.0, BLE
  Gigabit Ethernet
  2 USB 3.0 ports;
  2 USB 2.0 ports.
- GPIO : Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- Video and Sound : 2 × micro-HDMI ports (up to 4kp60 supported)
  2-lane MIPI DSI display port
  2-lane MIPI CSI camera port
  4-pole stereo audio and composite video port
- Multimedia : H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
  OpenGL ES 3.1, Vulkan 1.0
- SD Card Support : Micro-SD card slot for loading operating system and data storage
- Input Power : 5V DC via USB-C connector (minimum 3A*)
  5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature : 0 – 50 degrees C ambient

4.1.2 Raspberry Pi 4 GPIO Pinout:

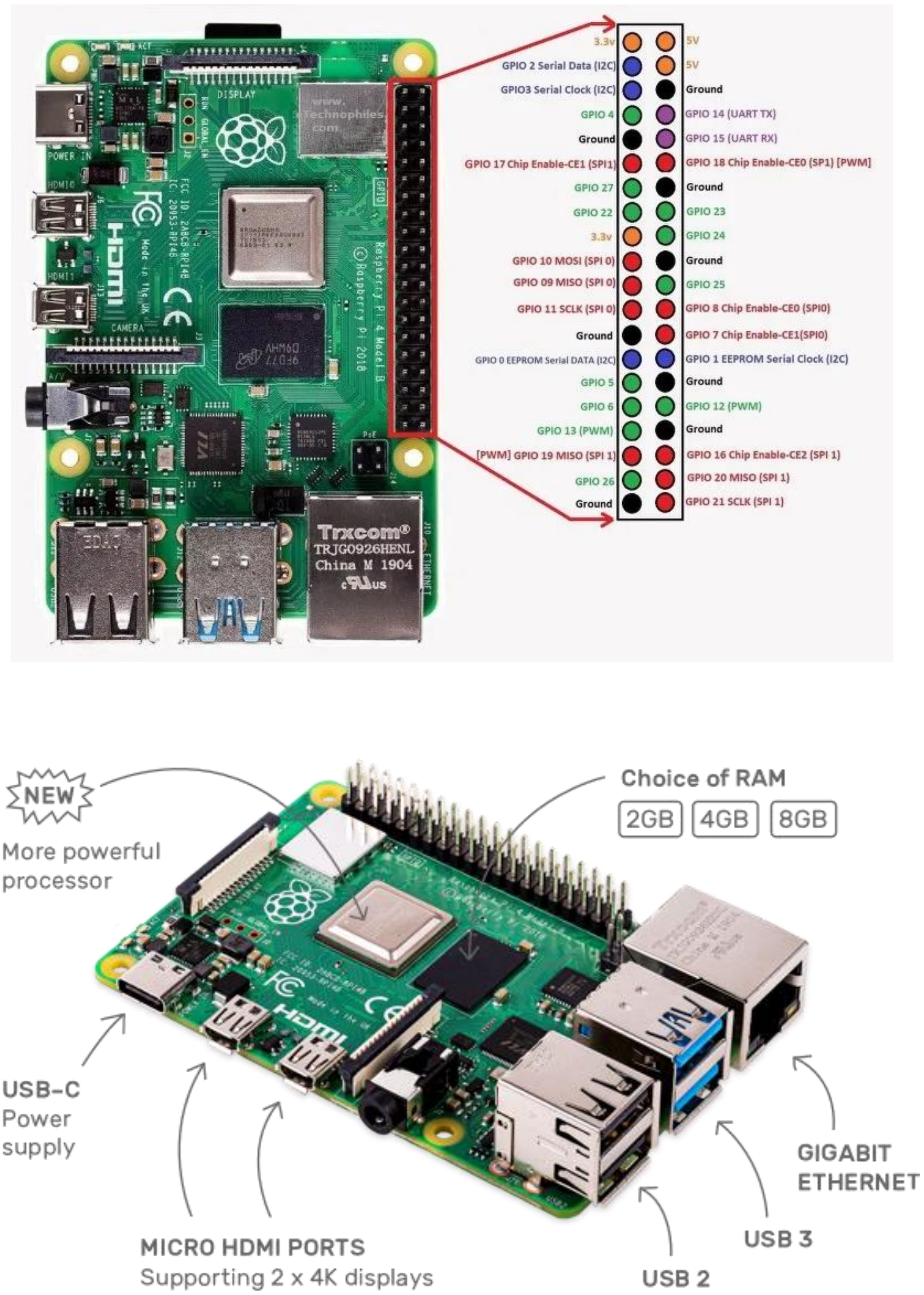


Fig 6 : Raspberry Pi 4 Pinout Configuration

4.2 HT11 Sensor :

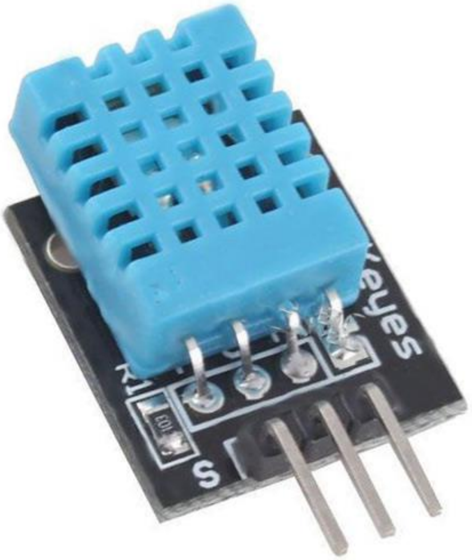


Fig 7 : DHT11 Sensor

DHT11 digital temperature and humidity sensor is a calibrated digital signal output of the temperature and humidity combined sensor. It uses a dedicated digital modules capture technology and the temperature and humidity sensor technology to ensure that products with high reliability and excellent long-term stability. Sensor includes a resistive element and a sense of wet NTC temperature measurement devices, and with a high-performance 8-bit microcontroller connected .

4.2.1 DHT11 Specifications

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

4.2.2 DHT11 Pinout Configuration :

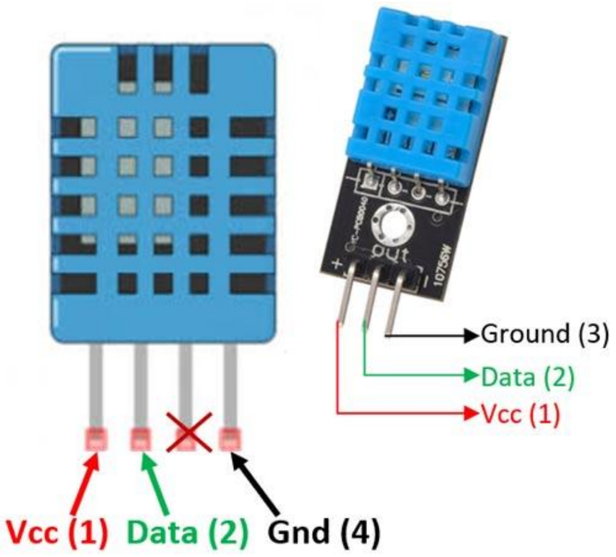


Fig 8 : DHT11 Pinout Configuration

For DHT11 Sensor

No.	Pin Name	Description
1	Vcc	Power Supply 3.5 to 5.5v
2	Data	Outputs both Temperature and Humidity through serail data
3	NC	No Connection and hence not used
4	Ground	Connected to ground of the circuit

For DHT11 Sensor Module

No.	Pin Name	Description
1	Vcc	Power Supply 3.5 to 5.5v
2	Data	Outputs both Temperature and Humidity through serail data
3	Ground	Connected to ground of the circuit

4.3. 12v DC Fan :

Fig 9 : 12V DC Fan

A DC motor is any motor within a class of electrical machines whereby direct current electrical power is converted into mechanical power. Most often, this type of motor relies on forces that magnetic fields produce. Regardless of the type, DC motors have some kind of internal mechanism, which is electronic or electromechanical. In both cases, the direction of current flow in part of the motor is changed periodically. The speed of a DC motor is controlled using a variable supply voltage or by changing the strength of the current within its field wind rings. While smaller DC motors arecommonly used in the making of appliances, tools, toys, and automobile mechanisms, such as electric car seats, larger DC motors are used in hoists, elevators, and electric vehicles. A 12v DC motor is small and inexpensive, yet powerful enough to be used for many applications. Because choosing the right DC motor for a specific application can be challenging, it is important to work with the right company. A prime example is METMotors, which has been creating high-quality permanent magnet DC motors for more than 45 years.

4.3.1. Features

- Rated voltage:12V.
- No-Load Speed:35000±10% RPM/MIN; No-Load Current: 0.85A.
- Diameter 38.5mm.
- Length 57mm.
- Shaft 3.17mm.
- Weight: 255g(APPROX).

4.4. Channel 5V Relay:

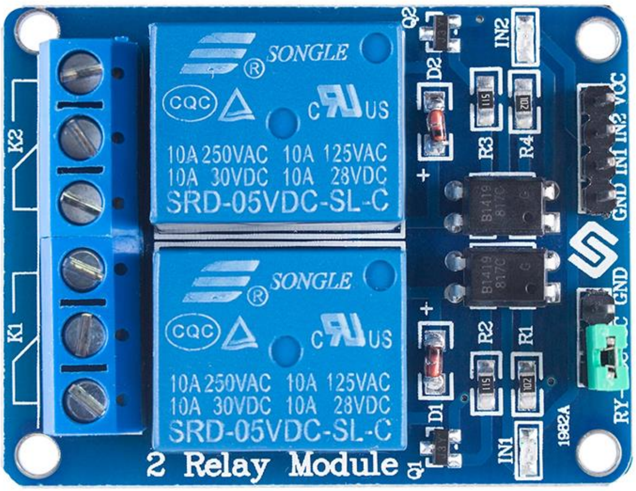


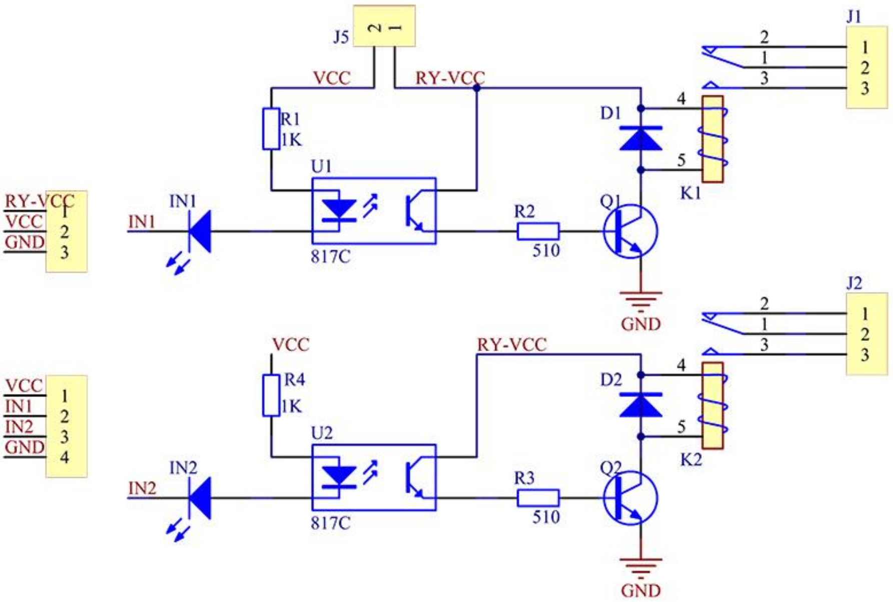
Fig 10 : 2 Channel 5V Relay Module

This is a LOW Level 5V 2-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller.

4.4.1. Features:

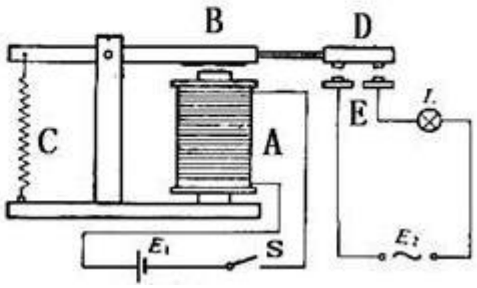
- Relay Maximum output: DC 30V/10A, AC 250V/10A
- 2 Channel Relay Module with Optocoupler LOW Level Triger expansion board, which is compatible with Arduino and Raspberry Pi
- Standard interface that can be controlled directly by microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic)
- Relay of high quality loose music relays SPDT. A common terminal, a normally open, one normally closed terminal
- Optocoupler isolation
- Good anti-jamming

4.4.2. Schematic:



In the below figure, A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one.

When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on.



Add a certain voltage to the coil and some currents will pass through the coil thus generating

the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open contact (or you may say releasing the former and the normally closed contact). After the coil is de-energized,

the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

Input:

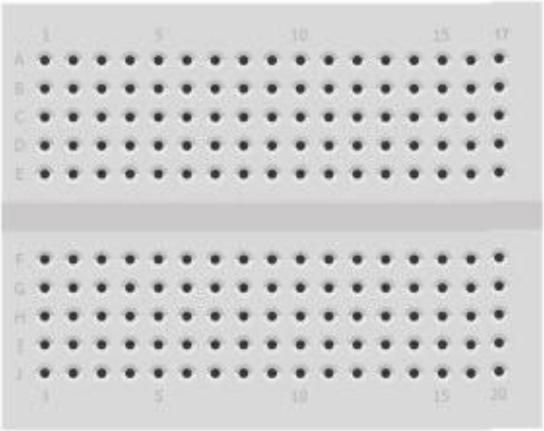
VCC: Connected to positive supply voltage (supply power according to relay voltage)
GND : Connected to negative supply voltage

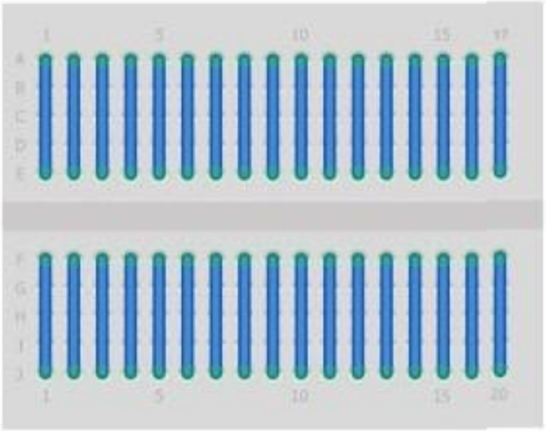
IN1: Signal triggering terminal 1 of relay module
IN2: Signal triggering terminal 2 of relay module

Output:

Each submodular of the relay has one NC (Normally close), one NO (Normally open) and one COM (Common). So there are 2 NC, 2 NO and 2 COM of the channel relay in total. NC stands for the normal close port contact and the state without power; No stands for the normal open port contact and the state with power. COM means the common port. You can choose NC port or NO port according to whether power or not.

4.5. Bread Board:





Breadboard can be derived from two terms namely bread & board. Initially, this was used to cut the bread into pieces. Further, it was called a breadboard & it was used in electronics projects and electronic devices in the year 1970. A breadboard is also known as a solderless board because the component used on the breadboard does not need any soldering to connect to the board, so it can be reused.

The arrangement of different components on a breadboard can be done by inserting their terminals into the breadboard, so it is frequently known as a plugboard. Breadboard definition is a plastic board in rectangular shape that includes a lot of small holes in it to allow you to place different components to build an electronic circuit is known as a breadboard. The connection on the breadboard is not permanent but they can be connected without soldering the components.

If you make any mistake while connecting the components, you can place or remove the components effortlessly. For beginners of electronics, this device is very helpful to make mini-projects. If a designer builds a simple circuit that they desire to analyse, then a breadboard gives a quick solution.

4.5.1. Specifications & Features

- Distribution Strips are two
- Wire Size is 21 to 26 AWG wire
- Tie Points are two hundred
- Withstanding Voltage is 1,000V AC
- Tie points within IC are 630
- Insulation Resistance is DC500V or 500MΩ
- Dimension is 6.5*4.4*0.3 inch
- Rating is 5Amps
- ABS plastic through color legend
- ABS heat Distortion Temperature is 183° F (84° C)Hole or Pitch Style is 2.54mm

4.6. 9V Battery :

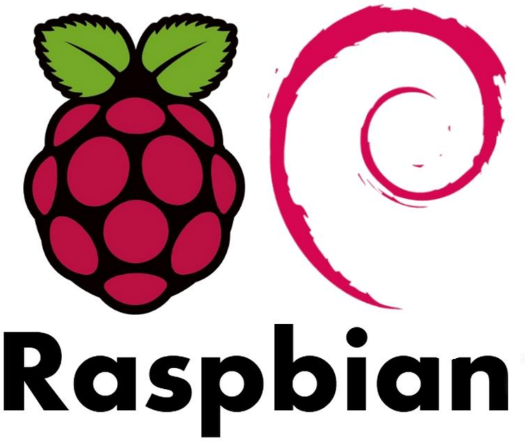
It is an affordable, reliable, dedicated low-power solution to provide sufficient energy to your application. Ideally used in circuits with low power consumption so that it can work for longer durations.

4.6.1. Features:

- Model Number: 9V 6F 22
- Battery Type: Zinc Carbon
- Size: 6F22 006P
- Jacket: Metal
- Single Battery Dimensions (mm): L- 26. 5, H - 48. 5, W - 17. 5 (Max)
- Nominal Voltage(V) :9V
- Discharge Resistance(Ω): 620
- Cut-off Voltage(V): 5.4

SOFTWARE REQUIREMENTS

4.7. Raspbian OS:



Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make your Raspberry Pi run. A Raspbian image is a file that we downloaded onto an SD card which in turn can be used to boot our Raspberry Pi and Via APC into the Raspbian operating system.

Debian is very lightweight and makes a great choice for the Pi. The Raspbian includes tools for browsing, python programming and a GUI desktop. The Raspian desktop environment is known as the "Lightweight XII Desktop Environment" or in short LXDE. This has a fairly attractive user interface that is built using the X Window System software and is a familiar point and click interface.

To connect the raspberry pi with the Raspbian OS:

Flash the SD Card with the Raspian OS.

1. To prepare the card for use with the Pi we will need to put a OS on the card.

   The OS must be Flashed into SDCard.

2. The Rasbian Os was Downloaded from URL http://www.raspberrypi.org/downloads/

3. Unzip the contents of the Zip file into a folder on your machine, one of the unzipped files would be a img file which is what needs to be flashed on to the SD card [In case there are more than one file, the current version of the zip has only this file and none other]

4. Flashing from Linux instructions

   a. Start the terminal on your Linux OS

   b. Insert the empty SD Card into the card reader of your machine.

   c. Type sudo fdisk -l to see all the disks listed. Find the SD card by its size,and note the device address (/dev/sdX, where X is a letter identifying the storage device. Some systems with integrated SDcard readers may use /dev/mmeblkX-format, just change the target in the following instructions accordingly)

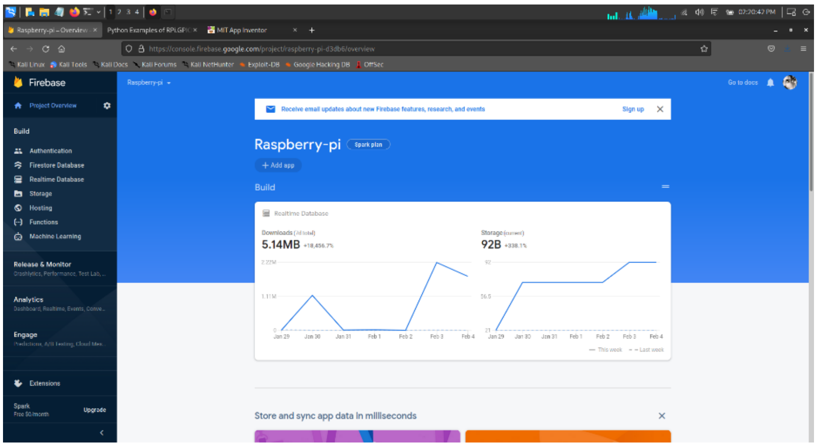
   d. Use cd to change to the directory with the img file you extracted from the Zip archive

5. Type sudo dd if=imagefilename.img of=/dev/sdX bs-2M to write the file imagefilename img to the SDcard connected to the device address. Replace imagefilename.img with the actual name of the file extracted from the Zip archive. This step takes a while, so be patient! During flashing, nothing will be shown on the screen until the process is fully complete.

6. Login into the Raspbian OS using Default userid and password

   a. Userid: pi
   b. Password: raspberry

4.8. Google Firebase :



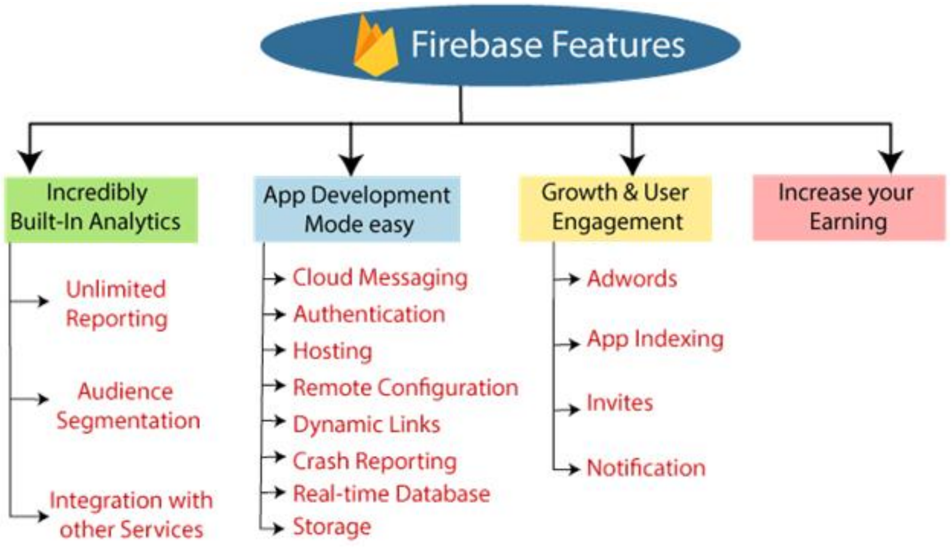
Receive email updates about new Firebase features, research, and events

Store and sync app data in milliseconds

Google Firebase is a Google-backed application development software that developers to Develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

4.8.1. Features of Firebase

Firebase has several features that make this platform essential. These features include unlimited Reporting, cloud messaging, authentication and hosting, etc. Let’s take a look at these features to understand how these features make Firebase essential:



4.9. Pyrebase :

Pyrebase

A simple python wrapper for the Firebase API.

Installation

pip install pyrebase

Getting Started

Python Version

Pyrebase was written for python 3 and will not work correctly with python 2.

Add Pyrebase to your application

For use with only user based authentication we can create the following configuration:

```
import pyrebase
      config = {
     "apiKey": "apiKey",
"authDomain": "projectId.firebaseapp.com",
"databaseURL": "https://databaseName.firebaseio.com",
"storageBucket": "projectId.appspot.com"
}
     firebase = pyrebase.initialize_app(config)
```

We can optionally add a service account credential to our configuration that will allow our server to authenticate with Firebase as an admin and disregard any security rules.

```
import pyrebase

config = {
"apiKey": "apiKey",
"authDomain": "projectId.firebaseapp.com",
"databaseURL": "https://databaseName.firebaseio.com",
"storageBucket": "projectId.appspot.com",
"serviceAccount": "path/to/serviceAccountCredentials.json"
}
      firebase = pyrebase.initialize_app(config)
```

Adding a service account will authenticate as an admin by default for all database queries.

Use Services

A Pyrebase app can use multiple Firebase services.

firebase.auth() - Authentication

firebase.database() - Database

firebase.storage() – Storage

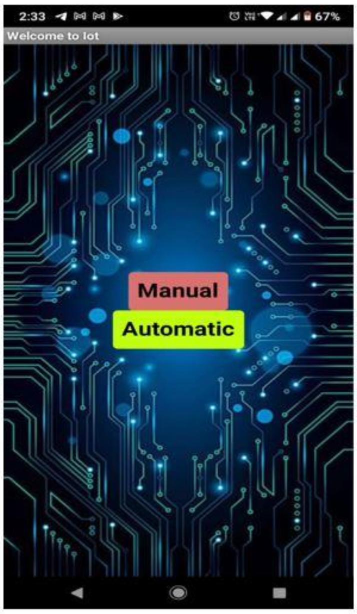
4.10. RPi :

The RPI.GPIO library is another Python-only library. It provide basic interactions with the GPIO pins, but no implementation of any connection protocol yet. The projects python files can be downloaded from Pypi.org, the projects home page is hosted on Scourceforge.

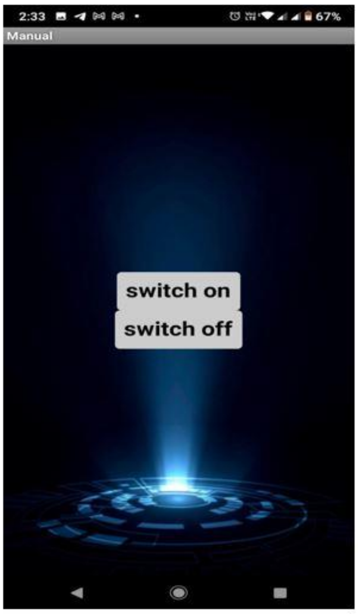
```
import RPi.GPIO as GPIO
import time

ledPin= 21
GPIO.setmode(GPIO.BOARD)
GPIO.setup(ledPin , GPIO.OUT)
while True:
  GPIO.output(ledPin , GPIO.HIGH)
  time.sleep(1)
  GPIO.output(ledPin , GPIO.LOW)
  time.sleep(1)
```

4.11. Android App:

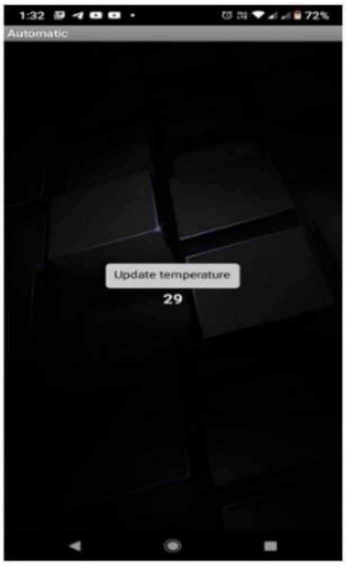
An Android app is a software application running on the Android platform. Because the Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS.



Screen 1



Manual Screen



Automatic Screen

Fig 11: Android App Images

5. IMPLEMENTATION OF PROJECT

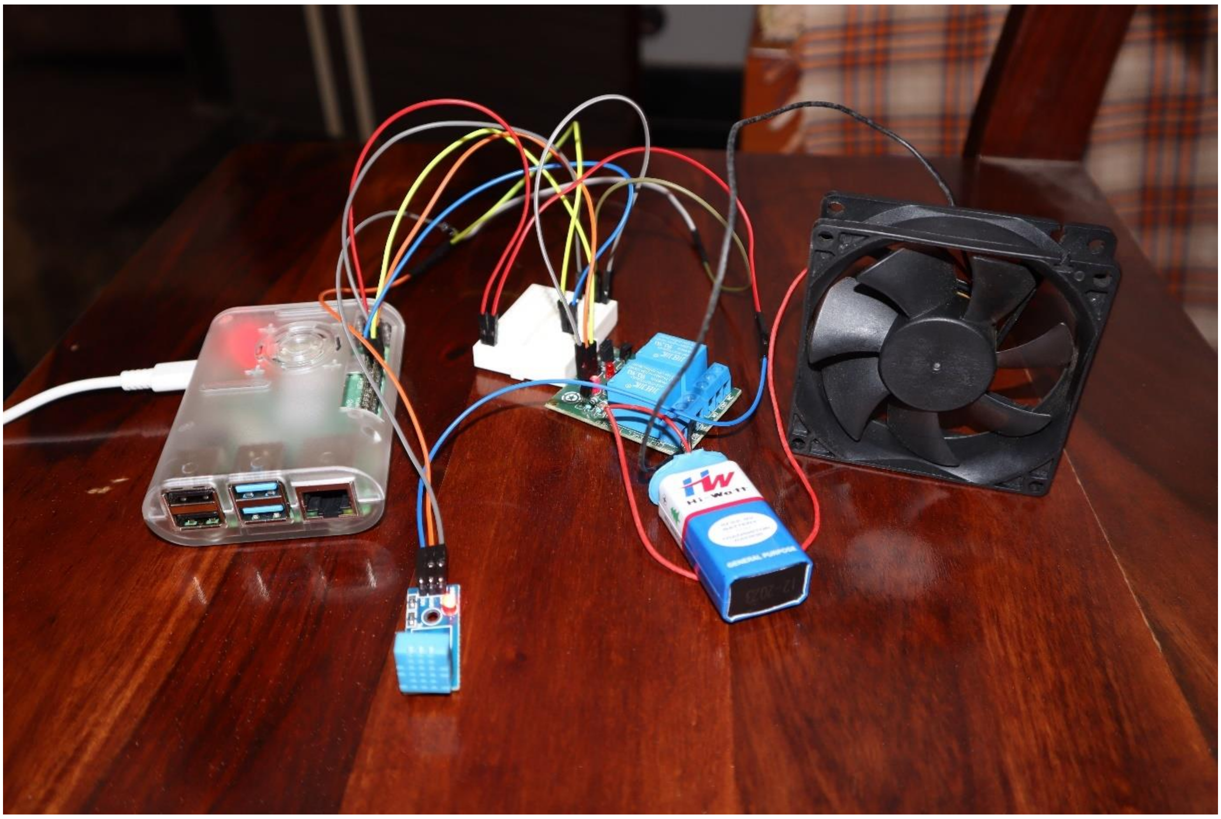
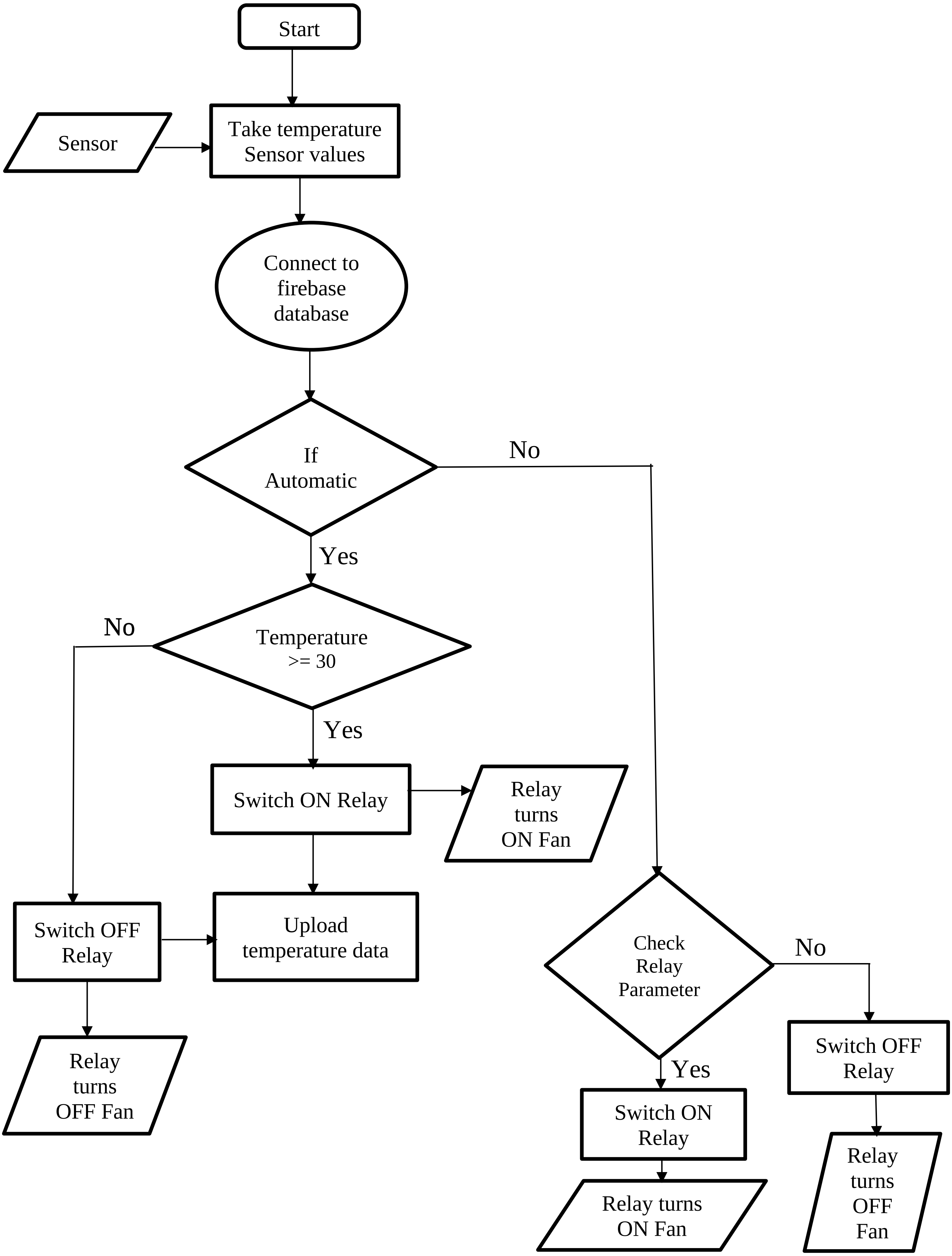


Fig 12 : Snapshot of Project

- This project presents the design and simulation of the temperature monitoring and control based on the room temperature
- A temperature sensor has been used to measure the temperature of the room and the fan is turned ON or OFF according to the room temperature
- If the temperature is greater than or equal to 30 degrees Relay is ON, it uploads the sensor data to firebase and relay turns ON the fan
- If the temperature is below 30 degrees Relay is OFF, it uploads the sensor data to firebase and relay turns OFF the fan

5.1. FLOWCHART



Source Code :

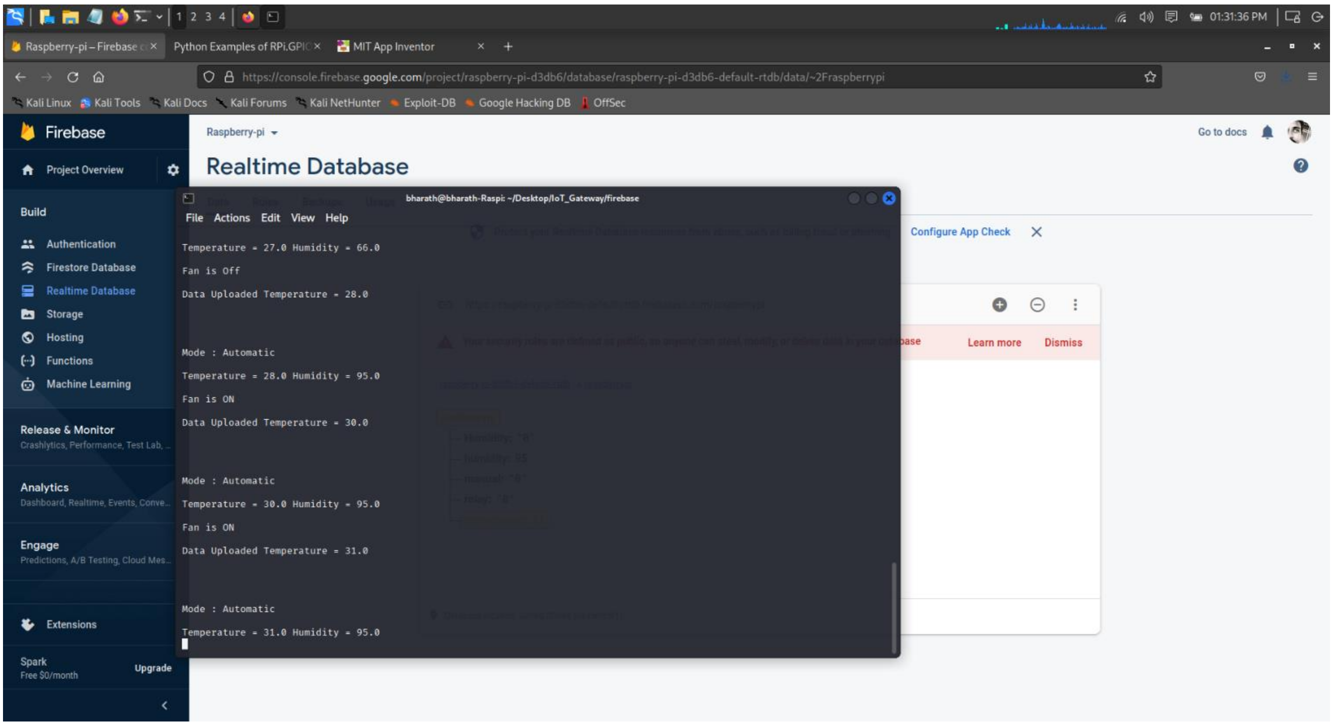
```
import RPi.GPIO as GPIO # To access the GPIO Pins of Raspberry pi

import pyrebase # APi to communicate with google firebase cloud

from time import sleep # To create a delay

import time

import sys # To access system tools

import Adafruit_DHT #To use the dht sensor

sensor=Adafruit_DHT.DHT11 # specify the type of sensor


# Details To connect to the cloud

config = {

  "apiKey": "AIzaSyAazprjzmWs0sv6RjmvLA0rim-q1m89VeQ ",

  "authDomain": "raspberry-pi-d3db6.firebaseapp.com",

  "databaseURL": "https://raspberry-pi-d3db6-default-rtdb.firebaseio.com/",

  "storageBucket": "raspberry-pi-d3db6.appspot.com"

}


#initialize the communication with the "firebase" servers using the previous config data.

firebase = pyrebase.initialize_app(config)


#Set the mode to use gpio pins

GPIO.setmode(GPIO.BOARD)


temp=4 # board 7 Gpio number 4
```

```
relay=8 # board 8


#set GPIO direction as Output
GPIO.setup(relay,GPIO.OUT)


# function to take values from sensor
def sense(pin):
    humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
    return humidity,temperature


#function to send connect signal to the relay
def relay_on(pin):
    GPIO.output(pin,False)


#function to send disconnect signal to relay
def relay_off(pin):
    GPIO.output(pin,True)


#Print the data on to terminal
def print_database(d):
    manual=d.child("raspberrypi").child("manual").get().val()
    humidity=d.child("raspberrypi").child("humidity").get().val()
    relay=d.child("raspberrypi").child("relay").get().val()
    temperature=d.child("raspberrypi").child("temperature").get().val()
    if manual=="1":
        print("\n\n\n\nMode : Manual")
```

```
else:
        print("\n\n\n\nMode : Automatic")
    print(f"\nTemperature = {temperature} Humidity = {humidity}")


if _name_ == '_main_':


    try:
        while True:
            # sense
            hum,tem=sense(temp)
            # connect to cloud database
            database = firebase.database()
            # print database to terminal
            #det=database.get().val()
            print_database(database)
            pi=database.child("raspberrypi")
            manual=pi.child("manual").get().val()
            database.child("raspberrypi").update({"temperature":tem,"humidity":hum})
            if int(manual)==1:
                servo=database.child("raspberrypi").child("relay").get().val()
                if int(servo)==1:
                    relay_on(relay)
                    print("Fan is  ON")
                else:
                    relay_off(relay)
                    print("Fan is  OFF")
```

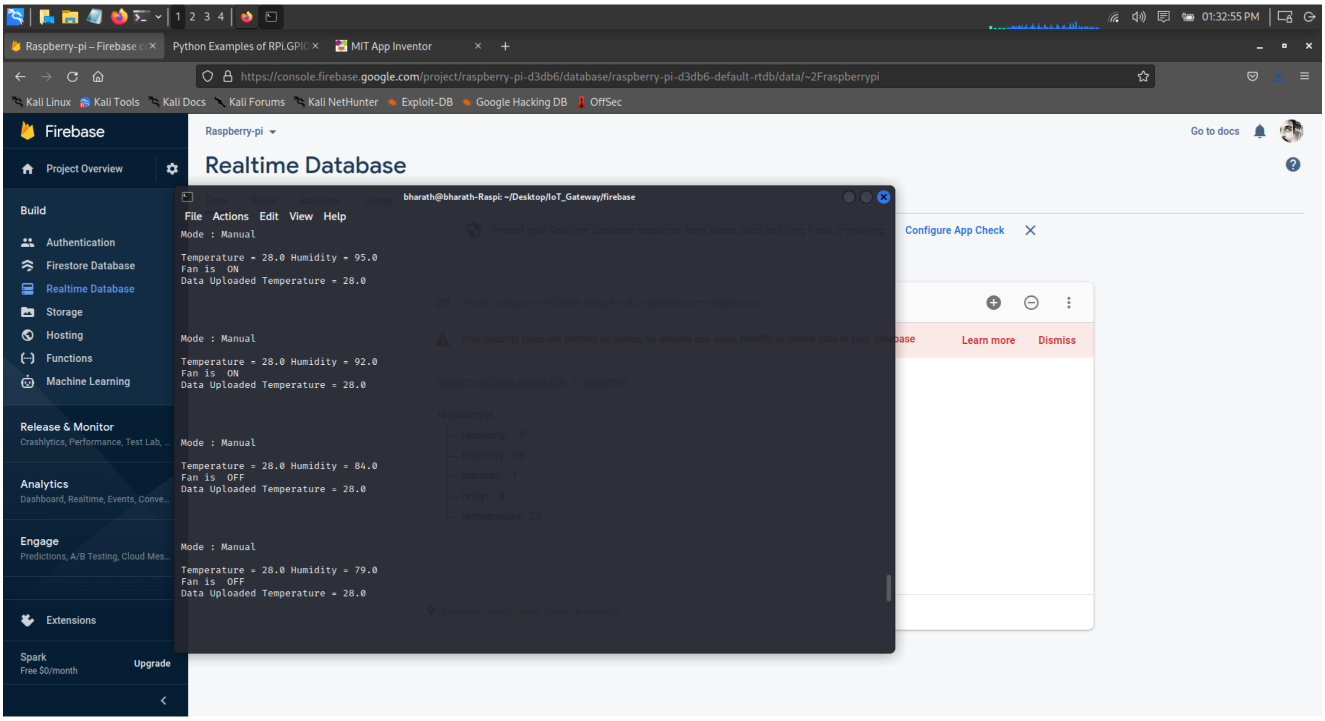
```
else:
            if tem>=30:
               relay_on(relay)
               print("\nFan is ON\n")
            else:

               relay_off(relay)
               print("\nFan is Off\n")
         print("Data Uploaded Temperature =",tem)

   # Handle Exceptions
   except Exception as e :
      print(e)
   except KeyboardInterrupt:
      print("Server stopped by User")
      GPIO.cleanup()
```

6. EXPERIMENTAL RESULTS AND ANALYSIS



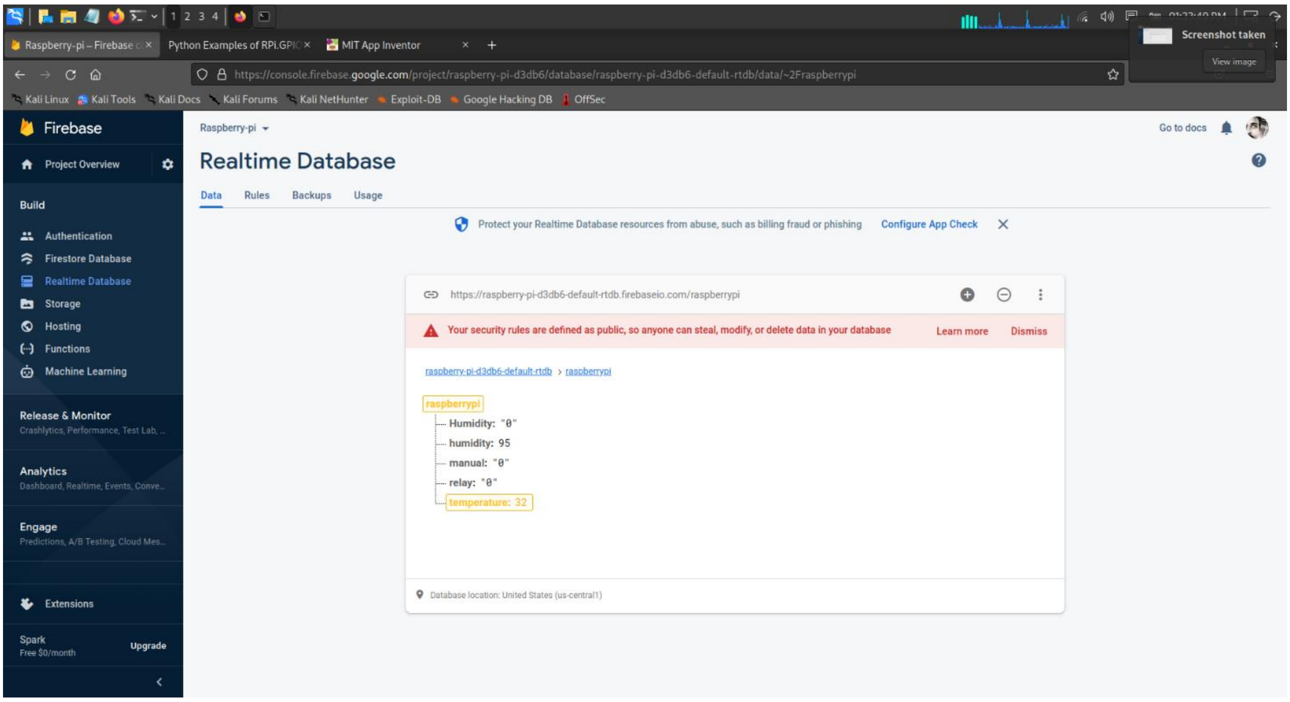
```
Temperature = 27.0 Humidity = 66.0

Fan is Off

Data Uploaded Temperature = 28.0


Mode : Automatic

Temperature = 28.0 Humidity = 95.0

Fan is ON

Data Uploaded Temperature = 30.0


Mode : Automatic

Temperature = 30.0 Humidity = 95.0

Fan is ON

Data Uploaded Temperature = 31.0


Mode : Automatic

Temperature = 31.0 Humidity = 95.0
```

AUTOMATIC MODE RESULTS



```
Mode : Manual

Temperature = 28.0 Humidity = 95.0
Fan is  ON
Data Uploaded Temperature = 28.0



Mode : Manual

Temperature = 28.0 Humidity = 92.0
Fan is  ON
Data Uploaded Temperature = 28.0



Mode : Manual

Temperature = 28.0 Humidity = 84.0
Fan is  OFF
Data Uploaded Temperature = 28.0



Mode : Manual

Temperature = 28.0 Humidity = 79.0
Fan is  OFF
Data Uploaded Temperature = 28.0
```

MANUAL MODE RESULTS



Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://raspberry-pi-d3db6-default-rtdb.firebaseio.com/raspberrypi

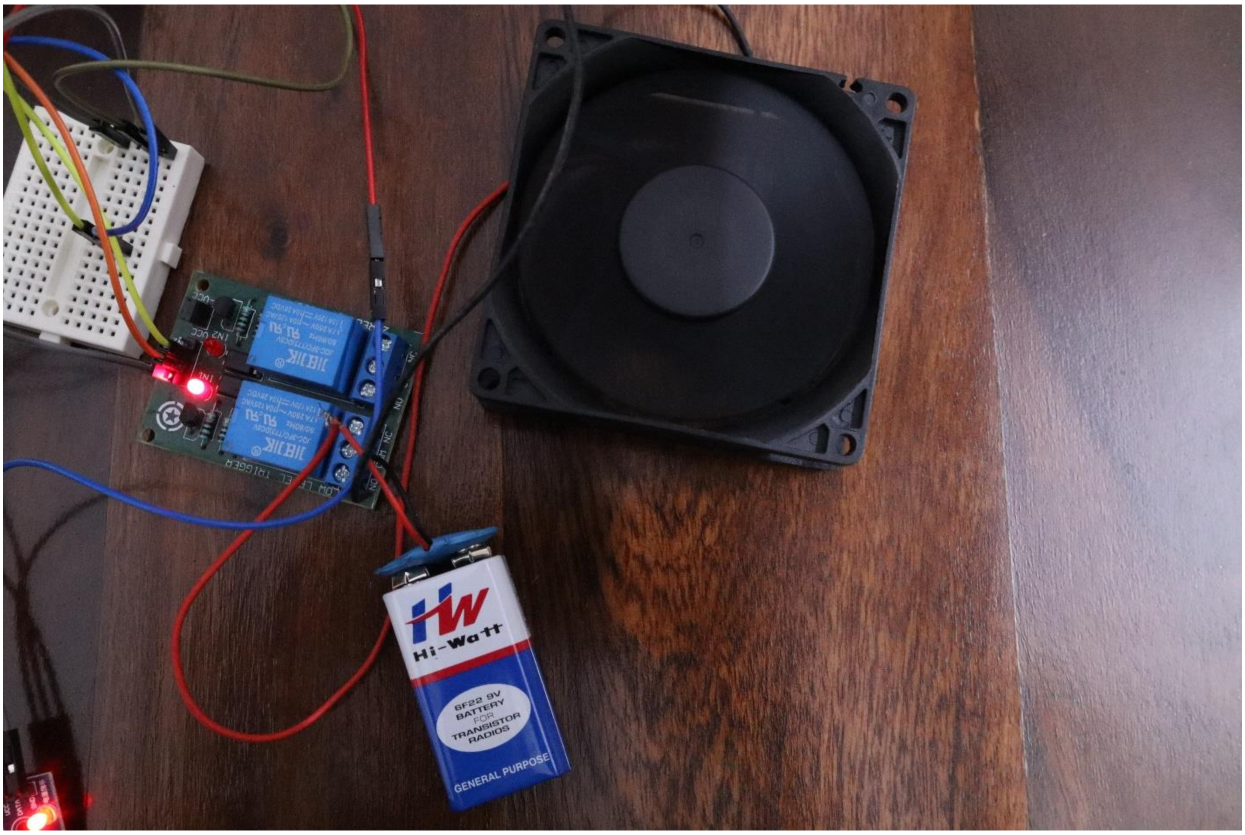
Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

raspberry-pi-d3db6-default-rtdb > raspberrypi

```
raspberrypi
├── Humidity: "0"
├── humidity: 95
├── manual: "0"
├── relay: "0"
└── temperature: 32
```

Database location: United States (us-central1)

CLOUD DATA



TEMPERATURE >= 30 ---- FAN ON

7. CONCLUSION

A simple IoT gateway is implemented for temperature monitoring and control using Raspberry Pi as IoT Gateway

The different hardware components and software designs are clearly described.

The Relay can be controlled manually or automatically through android App.

This project is also implemented to reduce power consumption.

In future more sensors and Actuators can be integrated with the Gateway and more processing can be done at the gateway level.

REFERENCES

[1]. E. Luchian, A. Taut, I. Ivanciu, G. Lazar and V. Dobrota, "**Mobile wireless sensor network gateway: A raspberry Pi implementation with a VPN backend to OpenStack,**" *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2017, pp. 1-5, doi: 10.23919/SOFTCOM.2017.8115561.

[2]. C. -S.Choi, J. -D. Jeong, J. Han, W. -K. Park and I. -W. Lee, "**Implementation of IoT based PV monitoring system with message queuing telemetry transfer protocol and smart utility network,**" *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, pp. 1077-1079, doi: 10.1109/ICTC.2017.819085

[3]. C. Sun, K. Guo, Z. Xu, J. Ma and D. Hu, "**Design and Development of Modbus/MQTT Gateway for Industrial IoT Cloud Applications Using Raspberry Pi,**" *2019 Chinese Automation Congress (CAC)*, 2019, pp. 2267-2271, doi: 10.1109/CAC48633.2019.8997492.

[4]. D. B. C. Lima, R. M. B. da Silva Lima, D. de Farias Medeiros, R. I. S. Pereira, C. P. de Souza and O. Baiocchi, "**A Performance Evaluation of Raspberry Pi Zero W Based Gateway Running MQTT Broker for IoT,**" *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 0076-0081, doi:10.1109/IEMCON.2019.8936206.

[5]. A. Moheddine, F. Patrone and M. Marchese, "**UAV and IoT Integration: A Flying Gateway,**" *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 121-122, doi: 10.1109/ICECS46596.2019.8965135.

[6]. S. Blionas, G. Doukas, K. Doukas and N. D. Tselikas, "**A Flexible/Scalable IoT Server Node testbed, from Gateway to Edge Computing. A Smart Home Use Case,**" *2019 Panhellenic Conference on Electronics & Telecommunications (PACET)*, 2019, pp. 1-6, doi: 10.1109/PACET48583.2019.8956284.

[7]. C. Sun, F. Zheng, G. Zhou and K. Guo, "**Design and Implementation of Cloud-based Single-channel LoRa IIoT Gateway Using Raspberry Pi,**" *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 5259-5263, doi: 10.23919/CCC50068.2020.9189480.

[8]. A. U. Chowdhury and M. M. Elahi, "**Design of a Smart Gateway for Edge Enabled IoT Applications,**" *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 417-420, doi: 10.1109/TENSYMP50017.2020.9230843.

[9]. A. M. Balladares Ocaña, R. I. Urgilés and J. A. Soria Pérez, "**Design and Implementation of a Web Platform Prototype Based an IoT Gateway Using Raspberry Pi for Livestock Monitoring,**" *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, 2021, pp. 1-6, doi: 10.1109/ETCM53643.2021.9590657.