IoT based Home Automation and Security System using Android Voice Assistant Working with Firebase RDB and FCM

Akshay Saini[1], Prof. Sangeeta Thakur[2]

[1]*Student Master of Technology, Department of Electronics and Communication Engineering SIRDA Group of Institution H.P. Technical University Hamirpur India.*

[2]*Asst. Professor, Department of Electronics and Communication Engineering SIRDA Group of Institution H.P. India*

---***---

Abstract - *IOT based home automation and security system is very less expensive and useful solution to control appliances and monitor home security. This kind of system can be used by senior citizens, handicap person or any other individual who is not available to do these tasks manually or not present in home. This system can be controlled easily by voice commands and manual switching through toggles in Android app. The app also provides voice feedbacks on each command. The system is to make easier to control the appliances from anywhere in the world with internet access. The system also has security features to monitor and report any unwanted movement in home when it is armed to the user through firebase push notifications. The third main feature of the system is to monitor and detection of any fire, Temp and LPG leak in home. The system uses a Raspberry Pi Model 3B+ to control and monitor using Android app.*

Key Words: Raspberry Pi, Android, Voice Assistant, Firebase, FCM, Cloud Messaging, Real Time DB, Python, Kotlin, Push Notification.

1. INTRODUCTION

Automation and security are two major fields of electronics and with increase and advancement of technology these fields are also advancing from simple home automation. The main use of this project is to reduce efforts and manual work. Aim of this project is to provide a less expensive working home automation and security system based on internet of thing. The system does not require any other major hardware then a Raspberry Pi and an Android phone. Basic working of this project is to use internet as the communication medium between Android device and Raspberry Pi. Android application has three main functionalities first is to provide voice recognized commands to the user and control the appliance by controlling Raspberry pi. Second functionality of the application is to provide button interface in any case user don't want or can't use the voice command feature. The third functionality of the app is to get data from the database which raspberry sent and show it to the user this data contains temperature sensor data, motion sensor alerts, fire sensor alerts and LPG gas leakage.

2. System Working Flow:

The working flow of the developed system has Android app as first part which is the interface between user and the hardware. In Fig-1 it shows the working block diagram of the developed system. The Android takes user commands in voice and toggle button interface. The toggle button interface and voice commands also provide functionality for arming the system for security.

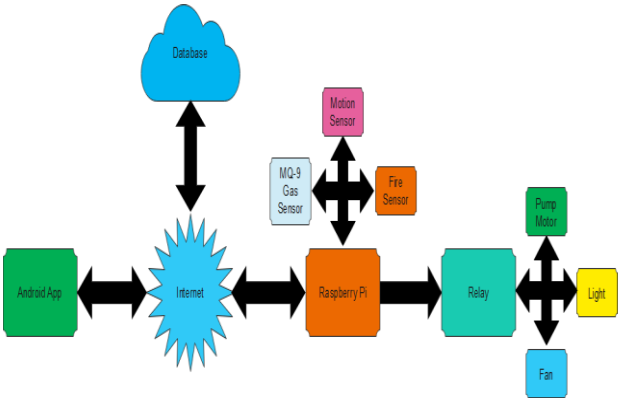


Fig-1: System Block Diagram

Android app and Raspberry are connected to a central real time database through internet which instantly updates the connected hardware with new data. Multiple devices can be connected to this DB. When user set a voice command or toggle a switch in app the particular device flag in the database will be set to true or false according to the on/off command. The Raspberry Pi is connected to internet through Wi-Fi will constantly listen to change in the database and it control the relay according to the appliance connected which is preset. The command follows like '*Turn on the light*', '*Turn on the fan*', '*Turn off the light*'. From the raspberry part it also makes changes in the database for the sensor values and send push notification alerts to the Android app using FCM. The push notifications are used to alert user if there are some critical conditions in the house like fire or LPG leak it will also alert the user if system detects any unwanted movement in the house if it is armed for monitoring the security all system can managed through the android app no need for reprogram the Raspberry or Android app when need to change any functionality.

2.1 Android App Flow:

When android app starts it connects to Firebase database using Firebase SDK after that it creates the database listeners and checks for the sensor data and update the app UI according to the data received. Basically, all the initialization process is done within fraction of second. Buttons are for voice command or a simple toggle button if its voice button it launches speech to text speech recognizer service and after inputting the voice command it will return the data to the app and we check the data and update the database according to the command. If button was a toggle button it will update the database directly.

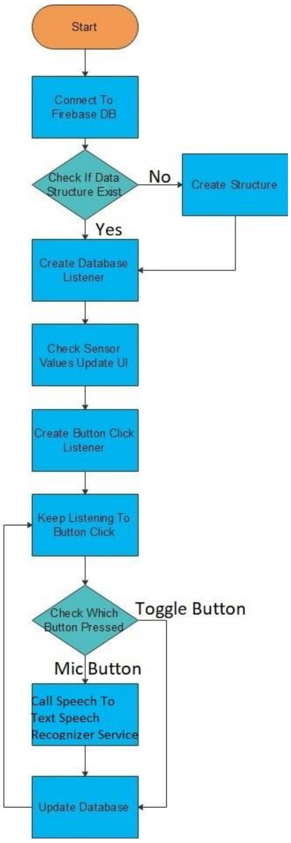

Start
Connect To Firebase DB
Check If Data Structure Exist
No
Create Structure
Yes
Create Database Listener
Check Sensor Values Update UI
Create Button Click Listener
Keep Listening To Button Click
Check Which Button Pressed
Toggle Button
Mic Button
Call Speech To Text Speech Recognizer Service
Update Database

Fig-2: Android App Flow

App will create button listeners and when user press the button it will check if the

2.2 Raspberry Pi Python Program Flow:

Raspberry pi starts program by initializing and creating pin objects to take tata from the sensor connected to those pins. The program runs in a while loop which continuously check any change in database and writes the sensor value to the DB or send push notification to the app through FCM.

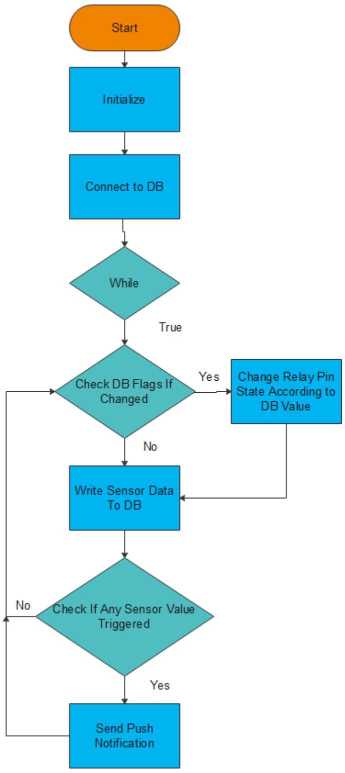

Start
Initialize
Connect to DB
While
True
Check DB Flags If Changed
Yes
Change Relay Pin State According to DB Value
No
Write Sensor Data To DB
No
Check If Any Sensor Value Triggered
Yes
Send Push Notification

Fig-3: Raspberry Python Program Flow

3. Used Components in Developed Project:

In below tables all components are listed according to their quantity and configuration used in this project.

Table-1: Hardware Components Used

Component Name	Quantity	Description
Raspberry Pi 3 B+	1	1GB Model
Android Smartphone	1	Android API 21 and above
Flame Sensor	1	
4 Relay Board	1x4	
PIR Motion Sensor HC-SR501	1	
MQ-9 Carbon Monoxide, Methane and LPG Gas Sensor	1	
5v Power Supply	1	
Jumper Wires		

Table-2: Software Components Used

Name	Version	Description
Android Studio	Arctic Fox | 2020.3.1	Windows, Mac or Linux
Kotlin	1.3.50	Android

		Programing Language
Python	3	Raspberry Programing Language
Firebase DB		Realtime database
Fire Cloud Messaging		Push Notification
Android App		Phone Application

3.1 Hardware Requirements

As we can see Table-2 shows with software's and programming languages used for development of this for the project.

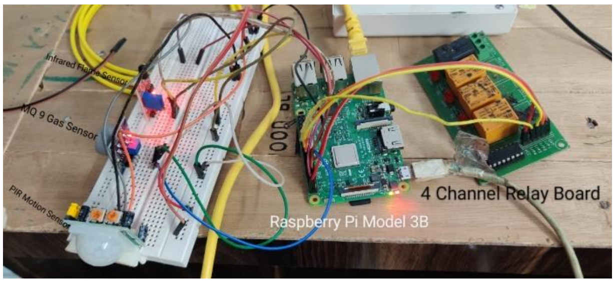


Fig 4: Actual Setup

3.1.1 Raspberry Pi

The Raspberry Pi used in the project is Raspberry pi 3 Model B with 1 GB RAM. If needed the newer models above it can also be used. Raspberry Pi is a small size inexpensive device with features listed below.

- 1 HDMI Port
- 40 GPIO Pins
- 4 USB Ports
- Broadcom BCM2837 chipset 1.2GHz Quad-Core ARM Cortex-A53 (64Bit)
- 1 GB Ram
- Ethernet Port
- HDMI Port
- Wireless Lan 802.11 b/g/n
- Bluetooth 4.1 (Classic and LE)
- Storage MMC

3.1.2 Android Smartphone:

To provide interface with hardware system uses an Android smartphone. The device needs to have Lollipop version of android and above for better working more than 2 GB RAM is best suitable.

3.1.3 Flame Sensor:

Project uses an infrared flame sensor to detect any fire in the house or where the system is used.

3.1.4 Relay Board:

The system uses 4 channel relay board to control the high voltage house hold appliances like Lights, TV, Fridge, Fan, Water Pump etc.

3.1.5 PIR Motion Sensor HC-SR501:

Project has a motion sensor to detect the movement in the room this motion sensor has two purposes when user in not present at home and system is armed it detects unwanted motions and Raspberry send a push notification to the user about the movement. In other case if user is at home and system is not armed the same system can be used to turn lights on and off according to the user movements.

3.1.6 MQ-9 Carbon Monoxide, Methane and LPG Gas Sensor

This sensor is used for detection of any harmful or flammable gases like LPG and report it immediately to the user.

3.1.7 Power Supply 5V:

Any 5V 2-2.5A power supply to drive the project.

3.1.8 Jumper Wires

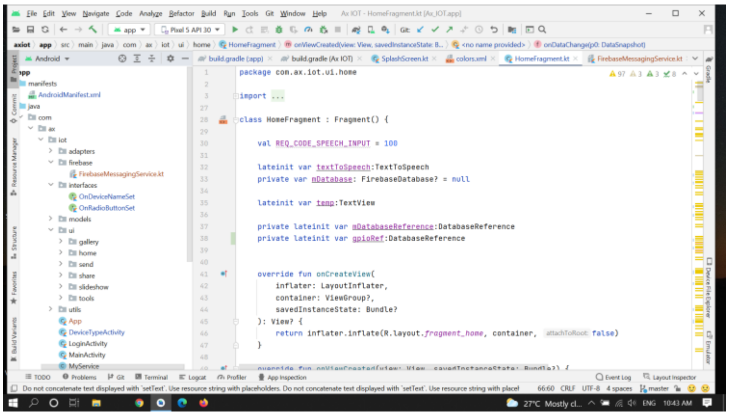
Jumper wires to connect the components to the Raspberry Pi.

3.2 Software Requirements:

As we can see Table-2 shows with software's and programming languages used for development of this for the project.

3.2.1 Android Studio:

Android studio is used to develop the android app. The app can be developed in both JAVA and Kotlin languages.



```
package com.ax.iot.ui.home

import ...

class HomeFragment : Fragment() {

    val REQ_CODE_SPEECH_INPUT = 100

    lateinit var textToSpeech:TextToSpeech
    private var mDatabase: FirebaseDatabase? = null

    lateinit var temp:TextView

    private lateinit var mDatabaseReference:DatabaseReference
    private lateinit var gpioRef:DatabaseReference


    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_home, container,  attachToRoot: false)
    }
```

Fig-5: Android Studio

3.2.2 Kotlin:

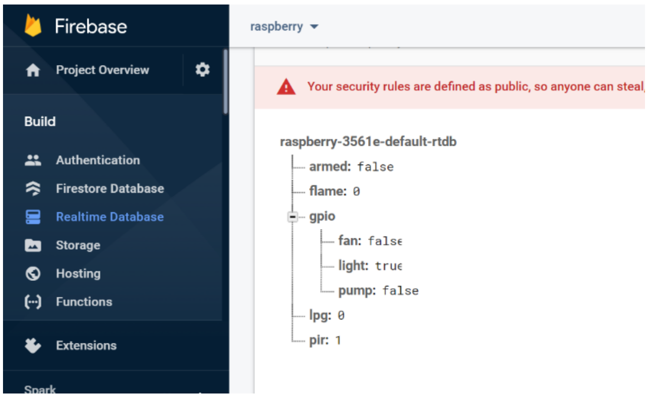
Kotlin is official native language of Android the app is developed in Kotlin.

3.2.3 Python:

Python is used for development of Raspberry Pi program to interface with database and read it and act accordingly.

3.2.4 Firebase DB:

Firebase DB is the central database used in the developed project. It is a real-time database syncs between all connected device immediately if any change occurs.



Your security rules are defined as public, so anyone can steal

```
raspberry-3561e-default-rtdb
├─ armed: false
├─ flame: 0
├─ gpio
│   ├─ fan: false
│   ├─ light: true
│   └─ pump: false
├─ lpg: 0
└─ pir: 1
```

Fig-6: Firebase DB Structure

3.2.5 Firebase Cloud Messaging:

FCM is used to send push notification from Raspberry Pi to the Android app. The notification can be alert or any data notification.

3.2.6 Android Application:

Android application is developed in Kotlin and uses Firebase SDK and google voice assistant with voice to text API.

4. Results:

After running the actual project when user input the voice commands or toggled the switches from the app the desired result was implemented. The app changed the flag values in database from true to false or false to true according to command given. The Raspberry which was continually listening to the database set the GPIO pin to HIGH on getting true from the DB and on LOW after getting false from the DB. Raspberry Pi also checks the sensor values and write data to the database every 1 second (This time can be changed according to need). The sensor data then is read by the Android app the app has listeners implemented whenever DB Raspberry changed the value of the sensor in the database the app showed it immediately. Raspberry also sent the push notification to the android whenever a certain case occurred on the controller for which it was programmed.

5. Conclusion:

This project was to show a cheap and fast way to control, secure and monitor the home. The system is faster than any GSM based system. Other major advantage of this system is it is based on IOT and can be controlled from anywhere in the world.

6. Future Scope:

The project is fully upgradable for the future. As the advancement in the technology in future I propose to use small ESP8266 Wi-Fi modules with relay to connect appliances with the system on same network to control appliances wirelessly. Other security features like biometric scanning can also be implemented to open doors and a camera can be added to check the video feed on motion detection. In future I am also trying to make changes in the app to dynamically add other devices.

References:

[1] D. Pavithra, R. Balakrishnan, in 2015 global conference on communication technologies (GCCT), IEEE, **2015**, pp. 169-173.

[2] S. Alani, S. N. Mahmood, S. Z. Attaallah, H. S. Mhmood, Z. A. Khudhur, A. A. Dhannoon, International Journal of Electrical Computer Engineering **2021**, 11.

[3] S. F. Islam, M. I. Hasan, M. Akter, M. S. Uddin, Journal of Computer Communications **2021**, 9, 143-157.

[4] S. Kiran, S. B. Sriramoju, Indian Journal of Public Health Research Development **2018**, 9.

[5] I. G. M. N. Desnanjaya, I. N. A. Arsana, Indonesian Journal of Electrical Engineering Computer Science **2021**, 22, 1295-1302.

[6] D. Shah, Procedia Computer Science **2016**, 79, 328-336.

[7] F. Ullah, F. Al-Turjman, A. Nayyar, Environmental Technology Innovation **2020**, 20, 101091.

[8] P. S. N. Reddy, K. T. K. Reddy, P. A. K. Reddy, G. K. Ramaiah, S. N. Kishor, in 2016 International conference on signal processing, communication, power and embedded system (SCOPES), IEEE, **2016**, pp. 285-290.

[9] Y. Irawan, R. Wahyuni, in Journal Of Physics: Conference Series, Vol. 1783, IOP Publishing, **2021**, p. 012094.

[10] Y. H. Lin, International Transactions on Electrical Energy Systems **2019**, 29, e12014.

[11] P. Kanade, S. Kanade, International Research Journal of Engineering Technology **2020**, 7, 1044-1049.