IoT based Smart Home Automation System using Raspberry Pi

Manas Singh Mehra
Bhavan Vidyalaya Panchkula,
Haryana, India

Abstract:- **At its very basic, home automation means automatic and electronic control of household utilities, security devices and appliances. An integrated approach to home automation implies the user and associated home appliances communicate with one another seamlessly. The aim of this research paper on Smart Home Automation is to explore the technical challenges of setting up a cost-effective home automation system with a web-based interface and assess its effectiveness in operating household utilities and security devices. Consequently, this paper evaluates the setting up of a smart home automation system employing Python Webserver with Flask based user interface and a Raspberry Pi. Flask is a lightweight Python web framework which enables quick creation of web-based applications using a single Python file, whereas the Raspberry Pi with its powerful CPU allows connection of multiple sensors simultaneously. The paper studies key aspects of sensor and software integration, analyses communication protocols and dwells on code development facets. Overall, the smart home automation system was seen to function smoothly and a cost economical setup of this kind besides enhancing home security and entertainment; can also assist the elderly in leading a better quality of life.**

Keywords:- Home Automation, Internet of Things based smart homes, Raspberry Pi, Python Webserver Flask, Sensors

I. INTRODUCTION

The term **Internet of Things** (IoT) refers both to the *collective network* of linked devices and the *technology* that facilitates communication between them. Home automation is one of the best examples of IoT as it seamlessly integrates consumer electronic gadgets as cameras, sensors, lights, fans, air-conditioners, etc. to each other via the internet. *The wireless network approach towards IoT based smart homes - employing a Raspberry device and Python Webserver with Flask - offers seamless, cost-effective and scalable solutions which can be easily maintained and preserved*. It results in numerous benefits as security and continuous monitoring of our homes, provision of timely healthcare and medical services to the elderly, energy conservation and addresses entertainment, comfort and lifestyle aspects also. This research paper on Smart Home Automation examines the practical trials and effectiveness of an economical home automation system. Through a Raspberry Pi and Python web-based interface home automation system, the users operate their home devices and security sensors. This paper reinforces the convenience, security, entertainment and health care benefits that accrue from utilizing simple IoT devices.

II. BACKGROUND

A significant amount of technological development has taken place in this field of home automation. Initial mobile based home automation systems utilized Java based boards, central computer and servers. Later Bluetooth, RFID, cellular network and Wi-Fi technologies were utilized in home automation system along with Arduino and Raspberry Pi integrated to Android phones. In recent times voice control over internet, cloud networking and wireless communication have been utilized for home automation systems. Latest trends include Machine Learning algorithms and Block-Chain technologies for enhancing communication, reliability, security and integrity of the network. The above systems are however complex, expensive and preclude the Do - It - Yourself user. A Raspberry Pi with Python interface on the other hand is an economical, efficient and scalable method with excellent user experience.

III. STRUCTURE OF PAPER

The paper is organized in following parts: -

a. Proposed system architecture.
b. Execution of the project.
c. Analysis of results.
d. Conclusion.

PROPOSED SYSTEM ARCHITECTURE

IV. SOFTWARE & HARDWARE COMPONENTS

The various components employed in the home automation system for a cost effective and open-source structure are: -

a. Raspberry Pi 4

Raspberry Pi is a low cost, small size computer that runs on Linux OS. Raspberry Pi is used as the local webserver and its General-Purpose Input/Output (GPIO) pins are controlled through a simple webpage.

b. Setting Up of Raspberry Pi

Raspberry Pi is set up before it can be used. And this is done in the following manner: -

i. Raspberry Pi with Raspbian OS is loaded and configured using the Raspi-config command.

ii. Post this the Secure Shell (SSH), Virtual Network Computing (VNC) and camera are enabled. 'SSH' command makes a secure encrypted link amongst two hosts across an insecure network. Using **Putty** - an SSH client - Raspberry Pi's **command line** is remotely accessed to execute the commands.

iii. VNC on the other hand is a graphical desktop-sharing system, which is used to remotely control another computer.

c. Python Flask

Apart from Raspbian OS, Python Flask was the other software used. Flask is a Python web framework, which offers helpful tools and features that simplifies the process of creating web applications. Flask was directly installed from the Raspbian OS's command line using: *sudo apt get-install python3-flask.*

d. Other Hardware Components

Other hardware components integrated were: -

i. IR, fire detection and PIR sensors.

ii. Webcam and LEDs/ lights.

iii. 5 V Eight Channel Relay module.

iv. Misc. components to include jumper wires, power supply etc.

V. BLOCK DIAGRAM

The block diagram of the proposed architecture as given below depicts the integration of hardware sensors in to the Raspberry Pi computer. The sensors based on environmental inputs send the data to Raspberry Pi which further processes the data. Corresponding values are then fed to the Relay Panel which changes the state of the devices as per the received input.

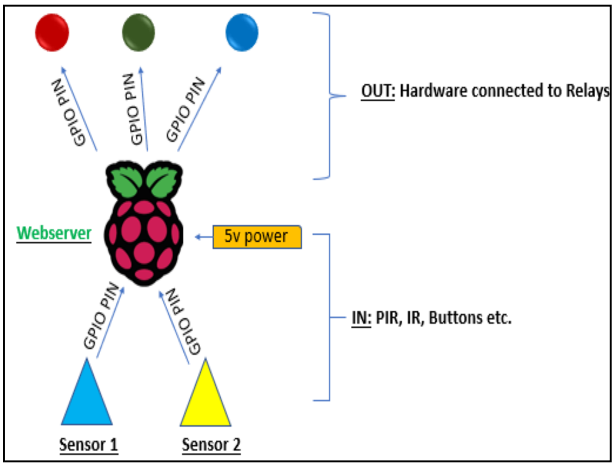


Fig. 1: Proposed Architecture

EXECUTION OF THE PROJECT

VI. HARDWARE INTEGRATION

Hardware integration of electronic components with the Raspberry Pi was undertaken and project was set up for remote access. The integration of the sensors with Raspberry Pi was as below: -

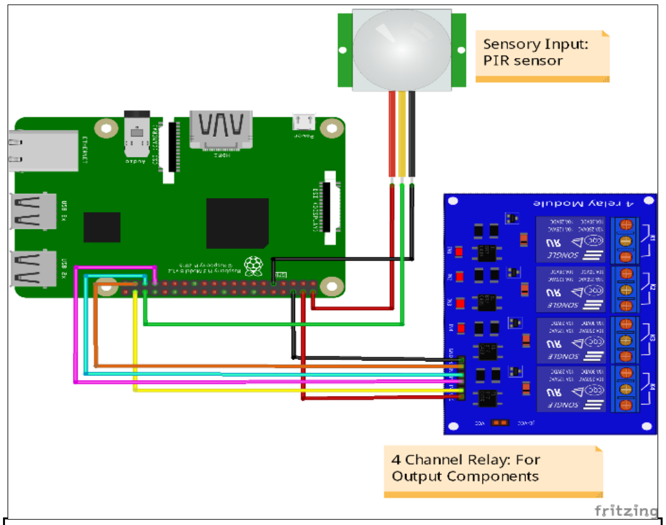

Sensory Input: PIR sensor
4 Channel Relay: For Output Components
fritzing

Fig. 2: Raspberry Pi & Sensor Integration

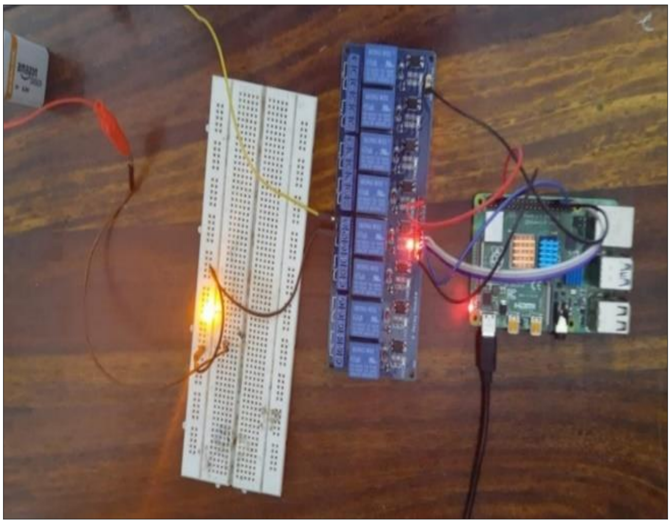


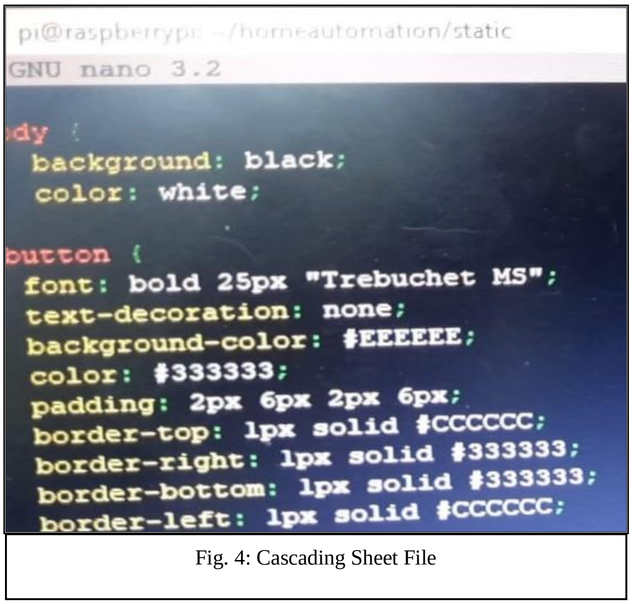
Fig 3: Raspberry Pi & Sensor Integration

VII. SOFTWARE CODES

The codes in the system architecture were applied as per steps depicted below: -

a. **Step1: Raspbian Command Line**. All code sequences were completed using the Raspbian command line. SSH was enabled in the Pi device and the command line was remotely accessed from another device.

b. **Step 2: Web Page**. An HTML (Hyper Text Markup Language) file for setting up the basic structure of the web page was created and linked to a CSS (Cascading Style Sheet) file to improve the aesthetics of the web page.



```
pi@raspberrypi: ~/homeautomation/static
GNU nano 3.2

dy {
 background: black;
 color: white;

button {
 font: bold 25px "Trebuchet MS";
 text-decoration: none;
 background-color: #EEEEEE;
 color: #333333;
 padding: 2px 6px 2px 6px;
 border-top: 1px solid #CCCCCC;
 border-right: 1px solid #333333;
 border-bottom: 1px solid #333333;
 border-left: 1px solid #CCCCCC;
```

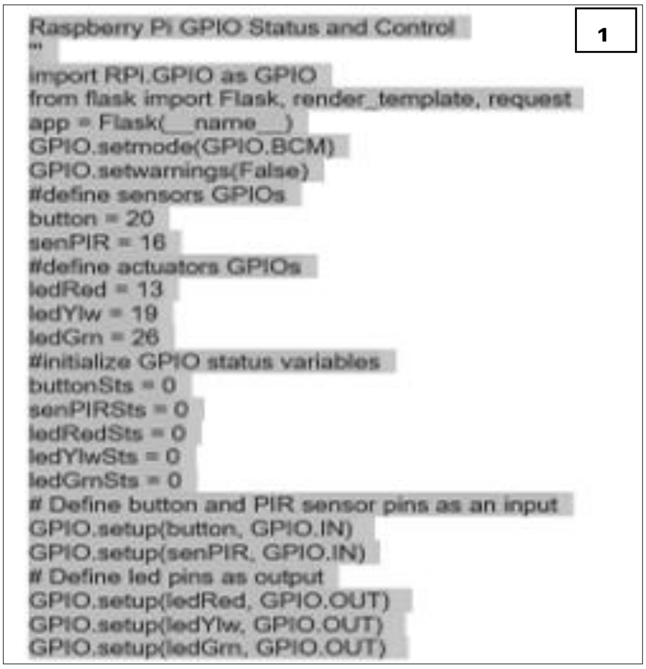
Fig. 4: Cascading Sheet File

c. **Step 3: Python Code**.

i. Python code was set up first by importing Flask and RPI.GPIO. The General-Purpose Input/output pins on the Pi were defined according to the completed hardware connections.

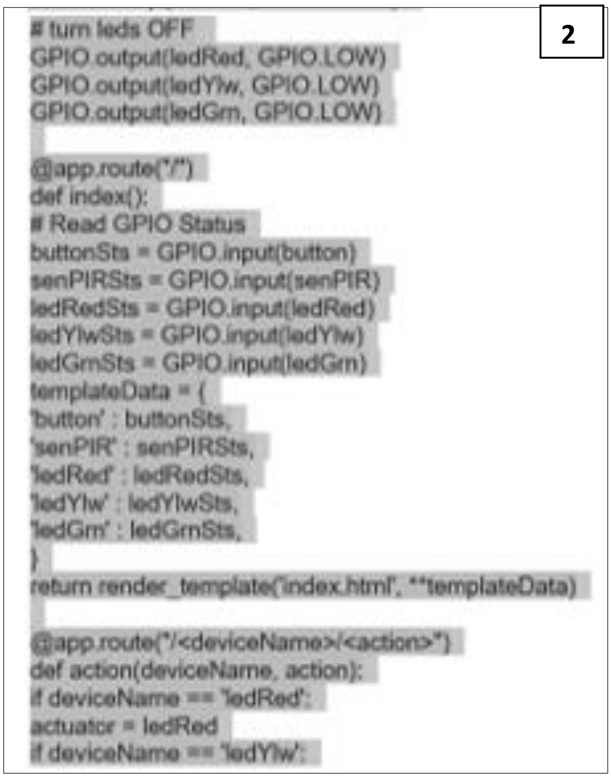
ii. For sensor results the status of input pins were noted and the data sent to HTML web page via the 'render_template' function. The sensor data was displayed on the webpage when the server was run.

iii. For hardware appliances (lights, fan etc.) connected to the defined output pins on the Pi, hyperlinks attached to the webpage (via the html file) along with conditional statements were used. The python webserver was run and port communication established via the '*app.run ()*' function.



1

```
Raspberry Pi GPIO Status and Control
'''
import RPi.GPIO as GPIO
from flask import Flask, render_template, request
app = Flask(__name__)
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
#define sensors GPIOs
button = 20
senPIR = 16
#define actuators GPIOs
ledRed = 13
ledYlw = 19
ledGrn = 26
#initialize GPIO status variables
buttonSts = 0
senPIRSts = 0
ledRedSts = 0
ledYlwSts = 0
ledGrnSts = 0
# Define button and PIR sensor pins as an input
GPIO.setup(button, GPIO.IN)
GPIO.setup(senPIR, GPIO.IN)
# Define led pins as output
GPIO.setup(ledRed, GPIO.OUT)
GPIO.setup(ledYlw, GPIO.OUT)
GPIO.setup(ledGrn, GPIO.OUT)
```

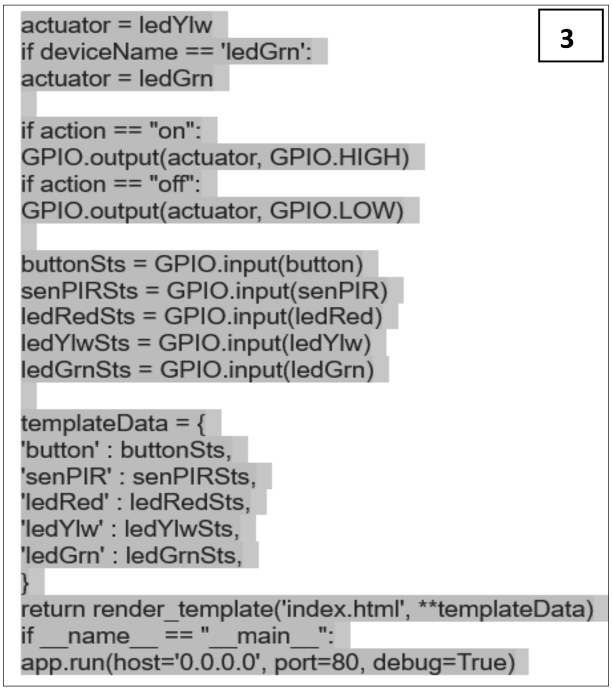


2

```
# turn leds OFF
GPIO.output(ledRed, GPIO.LOW)
GPIO.output(ledYlw, GPIO.LOW)
GPIO.output(ledGrn, GPIO.LOW)

@app.route("/")
def index():
# Read GPIO Status
buttonSts = GPIO.input(button)
senPIRSts = GPIO.input(senPIR)
ledRedSts = GPIO.input(ledRed)
ledYlwSts = GPIO.input(ledYlw)
ledGrnSts = GPIO.input(ledGrn)
templateData = {
'button' : buttonSts,
'senPIR' : senPIRSts,
'ledRed' : ledRedSts,
'ledYlw' : ledYlwSts,
'ledGrn' : ledGrnSts,
}
return render_template('index.html', **templateData)

@app.route("/<deviceName>/<action>")
def action(deviceName, action):
if deviceName == 'ledRed':
actuator = ledRed
if deviceName == 'ledYlw':
```



3

```
actuator = ledYlw
if deviceName == 'ledGrn':
actuator = ledGrn

if action == "on":
GPIO.output(actuator, GPIO.HIGH)
if action == "off":
GPIO.output(actuator, GPIO.LOW)

buttonSts = GPIO.input(button)
senPIRSts = GPIO.input(senPIR)
ledRedSts = GPIO.input(ledRed)
ledYlwSts = GPIO.input(ledYlw)
ledGrnSts = GPIO.input(ledGrn)

templateData = {
'button' : buttonSts,
'senPIR' : senPIRSts,
'ledRed' : ledRedSts,
'ledYlw' : ledYlwSts,
'ledGrn' : ledGrnSts,
}
return render_template('index.html', **templateData)
if __name__ == "__main__":
app.run(host='0.0.0.0', port=80, debug=True)
```

Fig. 5: Python Code

VIII. WEB PAGE INTERFACE

Raspberry Pi communicates with the user via the web interface, who can view and control the sensors by scrutinizing their current state. The server, client applications and the interface were set in Python code. The information on existing state of sensors and appliances is always stored in SQL database of Raspberry Pi and the database is continuously synched with the user interface. For example, the user can press the On/Off button in user interface or use the voice command to send a signal to the Raspberry Pi.

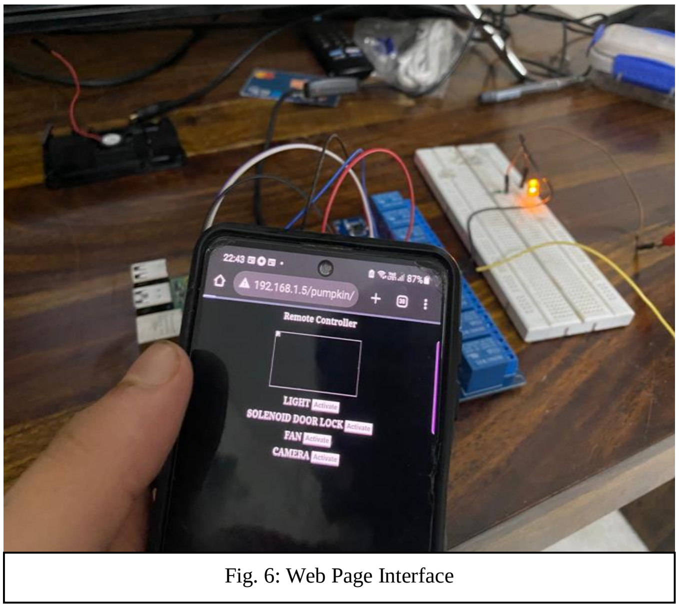


Fig. 6: Web Page Interface

ANALYSIS OF RESULTS

IX. EXAMINATION OF RESULTS

The smart home IoT automation system was accomplished and functioned smoothly as described in this paper. The structure enabled seamless control of sensors and appliances from a Python based user interface. The IoT system was overall found to be flexible, economical, efficient and scalable low-cost home automation solution. The interface design was user friendly and structure was fairly secure as it operates under the home Wi-Fi network. The key observations are as tabulated below: -

Results Table		
Sensor	***Observation***	***Expenses***
Raspberry Pi	DIY installation	Reasonable
Software	Python and HTML codes used	Open Source
Relays	Worked as expected	Cheap
IR	At times did not measure readings and also had a limited range.	Low-priced
PIR	Low sensitivity was observed	Low-priced
Camera	5 MP camera Inexpensive camera was used	Budget price Cost increases if high-end cameras used

X. CONCLUSION AND FUTURE DEVELOPMENTS

A home automation setup of this kind can be established in an economical manner using open-source software. This setup will enhance security of homes, management of home appliances as lighting, heating, entertainment etc. in a simple, flexible, robust and cost-effective manner. *Future developments* of this system would include incorporating a decision-making ability about the status of each device of the home especially for elderly and patient care. A ***machine learning algorithm*** can be applied for such decision making about the status of home devices and sensors. Security and integrity of home automation system can be enhanced of IoT system using ***blockchain technology*** to ensure secure communication.

REFERENCES

[1] D. Pavithra and R. Balakrishnan, "IoT based monitoring and control system for home automation," 2015 Global Conference on Communication Technologies (GCCT), 2015, pp. 169-173, doi: 10.1109/GCCT.2015.7342646.

[2] R. K. Kodali, V. Jain, S. Bose and L. Boppana, "IoT based smart security and home automation system," 2016 International Conference on Computing, Communication and Automation (ICCCA), 2016, pp. 1286-1289, doi: 10.1109/CCAA.2016.7813916.

[3] M. Al-Kuwari, A. Ramadan, Y. Ismael, L. Al-Sughair, A. Gastli and M. Benammar, "Smart-home automation using IoT-based sensing and monitoring platform," 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), 2018, pp. 1-6, doi: 10.1109/CPE.2018.8372548.

[4] Madihalli, and Chougala. 2019. "Advanced Home Automation System Using Raspberry Pi". Jetir.Org. https://www.jetir.org/papers/JETIR1906923.pdf.

[5] Kiran D. Suthar, S.K. Hadia, Brijesh Shah, "Home Automation Using Raspberry Pi with Android Application", International Journal of Research in Advent Technology, VOLUME-7 ISSUE-3, MARCH 2019, page no. 1044-1046, www.ijrat.org

[6] Sagar, Saurav and Choudhary, Ujjwal and Dwivedi, Rinky, "Smart Home Automation Using IoT and Raspberry Pi" (April 4, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020.

[7] Divya Purohit & Moumita Ghosh, "Challenges and Types of Home, Automation Systems" International Journal of Computer Science and Mobile Computing, Vol.6 Issue.4, April- 2017, pg. 369-375

[8] Abhilash Sinha, Ramasai Tatikonda. "IoT based home automation using Raspberry Pi." International Journal of Advance Research, Ideas and Innovations in Technology 5.1 (2019)

[9] Sandesh D. Patil, Jitendra M. Mahajan, Sushant M. Nagare, "Home Automation using Raspberry PI", IJACEN, Volume-7, Issue-5, pp 1-5 ,2019

[10] Venkatesh, K & Hemaswathi, S & Balakrishnan, Rajalingam & Rajkumar, P. (2018). "IoT Based Home Automation Using Raspberry Pi. Journal of Advanced Research in Dynamical and Control Systems". 10.1721-1728.

[11] Irwin Soni, "Design and implementation of home automation system using Raspberry Pi" A Project Presented to the Faculty of California State Polytechnic University, Pomona 2018.

[12] P Bhaskar Rao et al, "Raspberry Pi home automation with wireless sensors using smart phone" International Journal of Computer Science and Mobile Computing, Vol.4 Issue.5, May- 2015, pg. 797-803.

[13] https://towardsdatascience.com/python-webserver-with-flask-and-raspberry-pi-398423cc6f5d

[14] https://www.hackster.io/ahmedibrrahim/smart-home-automation-iot-using-raspberry-pi-and-python-47fb62#team