iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.63034

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

GURARDIANPI-Intelligent Home Security and Automation System

Mohammad[1], Shaik Abdul Rahman[2], Mohammed Abdul Rahman[3], B Sarala[4]
[1, 2, 3]*Electronics and Communication, M. V. S. R. Engineering College Osmania University, Nadergul, Hyderabad, India*
[4]*Guide MTech, Ph.D. Professor, Department of Electronics and Communication Engineering Maturi Venkata Subba Rao Engineering College, Hyderabad*

Abstract: ***The field of home automation and security has advanced significantly, allowing for the control and monitoring of household systems using a variety of technologies. This paper introduces GuardianPi, a cutting-edge intelligent home security and automation solution. GuardianPi uses a Raspberry Pi to integrate a variety of sensors, including DHT11 for temperature and humidity, RFID for secure entry, and ultrasonic sensor for burglar detection. Users can remotely operate electrical appliances and monitor environmental conditions using a web interface that is accessible from any location in the world. The system delivers email notifications for fire hazards, unlawful entry, and break-ins, allowing for quick and effective reactions to possible threats. GuardianPi's innovative hardware design and efficient software execution provide a complete and cost-effective solution for modern home control.***

Keywords: Home security, Raspberry Pi, RFID, Home automation, Sensors, Remote access, Threat Alerts, IoT.

I. INTRODUCTION

In the current day, integrating technology into our daily lives has become critical. The demand for efficient and secure home management systems has accelerated research and development in home automation and security. GuardianPi is a unique solution that integrates home security and automation into a single platform. Unlike previous solutions which primarily provide either security or automation capabilities, Guardian pi effortlessly incorporates both, together with added smart features.

Home automation systems frequently lack effective security capabilities, while traditional security systems do not offer the convenience of managing home appliances. GuardianPi overcomes these constraints by providing a full system that incorporates remote access control, fire warnings, break-in and burglar detection, alongside real-time temperature and humidity monitoring.

The system's core is a Raspberry Pi, which is integrated with multiple sensors and modules to effectively monitor and control the home environment. Users interact with the system through a secure web interface, which allows them to log in, view real—time data, and remotely control appliances. GuardianPi's unique combination of functions and user-friendly design makes it an affordable and effective option for modern home management.

II. LITERATURE REVIEW

GuardianPi combines home security and automation with a Raspberry Pi to create a comprehensive system. The DHT11 sensor monitors temperature and humidity, the real time data is updated on the webpage, when temperatures exceed certain limits, a fire alert is sent by SMTP email. The RFID module provides secure entrance. A proper RFID tag triggers the solenoid lock, whilst faulty tags or if the person standing in front of the door for longer than the stipulated time, generates an email alert for the user, signifying the potential risk of break-in. Home automation is controlled via a relay module attached to the Raspberry Pi's GPIIO pins.

GuardianPi overcomes the limitations of standalone systems by integrating security and automation. Using the Raspberry Pi, the system is both inexpensive and efficient, removing the need for additional peripherals. This research demonstrates the feasibility of merging different smart home technologies into a single platform, thereby enhancing both safety and convenience.

III. SYSTEM OVERVIEW

This section provides an overview of the system's operational workflow, Fig 1 briefly depicts the overview of system's working. It begins with user interaction via a web-based interface, accessible from any internet enabled device.

©IJRASET: All Rights are Reserved

Upon logging in, users gain access to a dashboard real-time data on temperature and humidity levels, alongside controls foe household appliances, User commands such as toggling appliances states, are transmitted to the database for storage and processing, The Raspberry Pi acting as the central processing unit, retrieves status values from the database. Through programmed logic, it orchestrates the control of electrical appliances based on the value retrieved from the database. A '0 value indicates the raspberry pi to turn OFF the corresponding mapped electrical appliance and a '1 indicates the raspberry pi to turn ON the corresponding electrical appliance.

After processing user commands, the Raspberry Pi interfaces with various sensors deployed throughout the home, including those for fire and intrusion detection. These sensors continuously monitor environmental conditions and security parameters, sending relevant data back to the Raspberry Pi for analysis.

The system's capacity to seamlessly integrate user interaction, sensor feedback forms the foundation of its operational efficiency and reliability.

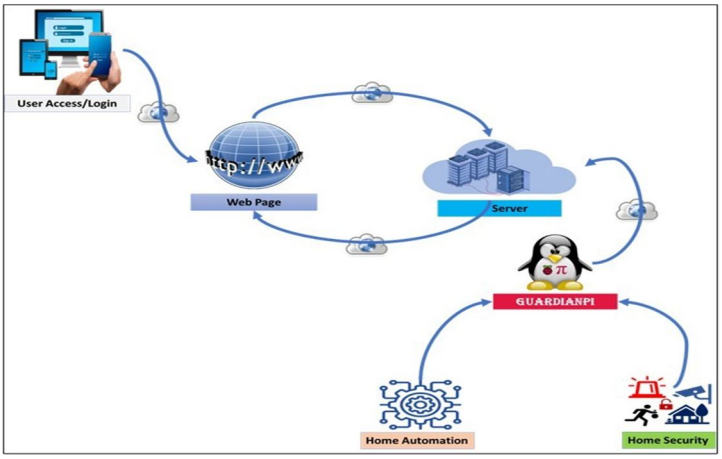


Fig 1: Operational Workflow of the system

Sections A, B, and C delve into the foundational aspects of the GuardianPi system, Comprising its user interface, data flow and management and database integration.

A. User Interface and Interaction

The system has a user-centric web interface accessible via standard web browsers. When the user redirects to the website, a login page is displayed, Fig 2 shows the login page of the system. Upon login, users gain access to a dashboard from which they may monitor environmental variables such as humidity and temperature as well as manage home security settings and control connected appliances remotely.

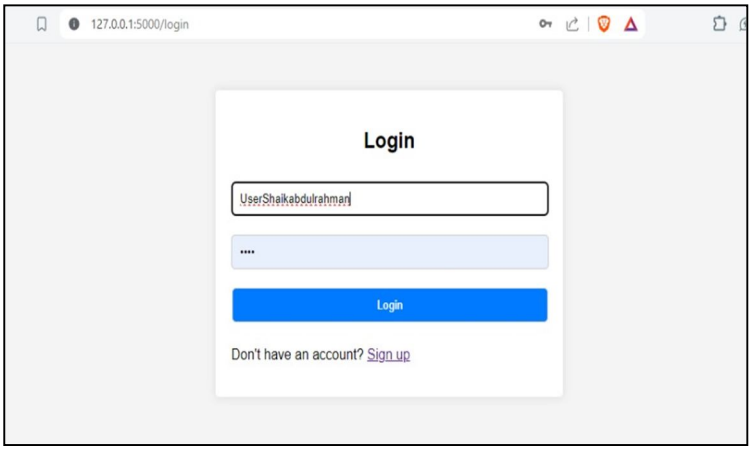


Fig 2 User login Page

This interface serves as the primary point of interaction, enabling seamless engagement with the GuardianPi system.

B. *Data Flow and Management*

GuardianPi has a sophisticated data flow architecture that is maintained behind the scenes to ensure seamless communication between the web interface, Raspberry Pi controller, and interconnects sensors and actuators. User commands are securely transmitted to the Raspberry Pi, which processes and implements corresponding actions. Simultaneously, the Raspberry Pi receives environmental data from the inbuilt sensors and analyses and stores it to provide insights into the home environment.

C. *Database Integration*

Integral to the functionality of the system, is its robust database integration, enabling the persistent storage and retrieval of critical operational data. The Raspberry Pi interacts with this database to update records in real-time, ensuring the synchronization between user interaction and system responses. This bidirectional data flow enhances overall responsiveness and operational efficiency with the GuardianPi ecosystem.

IV. FEATURES

A. *Temperature and Humidity Monitoring*

GuardianPi gives users real-time insights into their home environment with its extensive temperature and humidity monitoring features. Through appropriate sensors, it tracks temperature and humidity levels, ensuring optimal living conditions. Users can conveniently access this information via the web interface, allowing them to stay informed about any fluctuations that may require attention. Additionally, the system is equipped to send alerts to users in the event of abnormal readings enabling proactive measures to maintain potential issues.

B. *Fire Alert System*

An essential safety component of the system is the fire alert feature, which recognizes and reacts to possible fire threats. Utilizing modern sensors and appropriate algorithms, GuardianPi monitors for signs of smoke or rapid increase in temperature, indicative of fire. Upon detection, it promptly alerts users, allowing them to take immediate action to address1 the situation. By providing early warnings and facilitating rapid response, GuardianPi enhances home safety and minimizes the risk of fire related damage.

C. *Burglar Alert System*

To prevent unauthorized entry and enhance home security, GuardianPi has a powerful burglar alarm system. Using advanced sensors and detecting techniques, GuardianPi keeps an eye out for any unwanted entry attempts and suspicious behavior. GuardianPi rapidly notifies users in the case of a security breach, enabling them to take appropriate action, such as notifying authorities or initiating deterrent measures. GuardianPi gives people confidence in the safety of their homes and peace of mind with its proactive security measures.

D. *RFID Entry System*

The RFID entry system implemented by GuardianPi offers secure and convenient access control to the home. RFID tags and cards can be registered by users to verify their identification and make sure that only those with permission are allowed in. The RFID entry system from GuardianPi blends in perfectly with the entire security setup, offering dependable access control without sacrificing ease of use. By eliminating the need for traditional keys and enhancing security, GuardianPi's RFID entry system enhances the overall user experience and reinforces home security.

E. *Remote Control of Electrical Appliances*

GuardianPi enables users to remotely control electrical appliances within their home, offering unparalleled convenience and flexibility. Users may easily toggle the state of connected appliances, such as lights, fans, or electronic devices. The Raspberry Pi's seamless integration with relay modules, ensures reliable and responsive control, allowing users to customize their preferences. Whether controlling home appliances or giving entry into the house from remote location, GuardianPi puts user in control of their home's energy consumption and comfort.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

F. Email Alerts

Email alerts are one of the ways GuardianPi keeps users informed and equipped to handle important situations, in the event when fire sensed by the system, it promptly sends fire alert email to the user, upon detection of intrusion, GuardianPi quickly notifies through sending an email about the potential risk of break-in. By delivering timely notifications directly to users' inboxes, GuardianPi ensures that users stay informed and proactive in managing their home's security and environmental integrity.

V. IMPLEMENTATION

The system's core is a Raspberry Pi, which is integrated with multiple sensors and modules to effectively monitor and control the home

A. Software Implementation

This section provides a detailed description of the python scripts used for various functionalities and also covers the working of the web-based interface.

The chunk of code shown in the Fig.3 is the python code for IoT integration, this code connects the Raspberry Pi to various IoT Devices. It sends and receives data, enabling remote control and monitoring. Libraries such as 'requests' or 'paho-mqtt' can be used for this purpose. The script shown in Fig. 4 is for image capture and Notification Setup, it uses 'picamera' library, the captured images are then emailed to the user using the 'smtplib' library for notifications. The Fig. 5 shows the python script for intrusion detection loop with email alert, a continuous loop checks sensor inputs for any intrusion signals. The python script shown in Fig.6 reads data from the ultrasonic sensor using the 'RPi.GPIO' and 'time' libraries. Fig.7 is the python code for multithread main RFID Intrusion Update, multithreading is used to handle multiple tasks concurrently. The python script shown in Fig. 8 is to read RFID, Update database, fetch status using libraries such as 'MFRC522'.

The web-based interface allows users to control electrical appliances and monitor home security through a user-friendly dashboard. Users can log in, view real-time data, and toggle appliances on or off. The workflow of the website is shown in Fig.9. The interface communicates with SQLite database using python flask scripts to store and retrieve appliance statuses.

```python
from sensors import *
from picamera import PiCamera
import time
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
import os
import RPi.GPIO as gpio
from mysql import connector
import threading
from mfrc522 import SimpleMFRC522


data = {}


def get_db_connection():
    conn = connector.connect(
        host='up-us-sjo1-mysql-1.db.run-on-seenode.com',
        port=11550,
        user='db-hlx8d21axvv7',
        password='QPDMAVVtbc39RG014R0ytGsO',
        database='db-hlx8d21axvv7'
    )
    return conn
```

Fig. 3 Python Code for IoT integration with Raspberry Pi

```python
def capture_img():

    with PiCamera() as camera:
        camera.resolution=(320,320)
        time.sleep(4)
        camera.capture('intrusion.jpg')


    print('Image captured successfully.')
    return True


def email_conn():
    sender_email = "mohammad.ahmed1774@gmail.com"
    receiver_email = "mohammad.ahmed1774@gmail.com"
    password = "kehx negx kqvn luhw"

    # Create the email message
    msg = MIMEMultipart()
    msg['From'] = sender_email
    msg['To'] = receiver_email
    msg['Subject'] = 'intrusion detected'
    message = f'CLICK ON THIS LINK TO SEE THE LIVE FEED/n{URL}+/camera_feed'

    # Attach message
    msg.attach(MIMEText(message, 'plain'))

    with open('intrusion.jpg', 'rb') as f:
        img_data = f.read()
        image = MIMEImage(img_data, name='intrusion.jpg')
        msg.attach(image)
```

Fig. 4 Python code for image capture and notification setup

```python
    # Connect to the SMTP server
    with smtplib.SMTP('smtp.gmail.com', 587) as server:
        server.starttls()
        server.login(sender_email, password)
        server.sendmail(sender_email, receiver_email, msg.as_string())

    print('mail sent')

def main():
    ultra_dist = UltraSensor()
    distance = ultra_dist.update_distance()

    while True:

        dist = ultra_dist.update_distance()
        try:
            #gpio.add_event_detect(BUTTON_PIN,gpio.FALLING, callback= button_pressed,bouncetime=200)
            if dist < 0.5:
                time.sleep(10)
                dist = ultra_dist.update_distance()
                if dist < 0.9:
                    capture = capture_img()
                    if capture:
                        email_conn()
        finally:
```

Fig. 5 Python for intrusion detection loop with email alert

```python
def intrusion_detect():

    time.sleep(2)

    try:
        usensor = UltraSensor()
        while True:

            dist = usensor.update_distance()
            print(dist)
            if dist < 0.5:
                print('intrusion detected starting countdown')
                time.sleep(5)
                dist = usensor.update_distance()
                if dist < 0.9:
                    capture = capture_img()
                    if capture:
                        email_conn()
            else:
                print('no person')
                #intrusion_detect()

    except:
        intrusion_detect()
```

Fig. 6 Python code for ultrasonic sensor integration and intrusion alert

```python
def main1():
    thread1 = threading.Thread(target=init_rfid)
    thread2 = threading.Thread(target=intrusion_detect)
    thread3 = threading.Thread(target = app_update)


    thread1.start()
#    thread2.start()
    thread3.start()


    print('both functions completed')
```

Fig. 7 Python code for multithread main: RFID intrusion update

```python
def read_rfid():
    reader = SimpleMFRC522()

    try:
        print('place your card')
        id, text = reader.read()
        data['id'] = id
        data['username'] = text
        print(id,text)

    finally:
        gpio.cleanup()

    #print(username)
        conn = get_db_connection()
        cursor = conn.cursor()
        status = 1
        cursor.execute('UPDATE users SET app5 = %s WHERE rfid = %s',(status,id))
        conn.commit()
        conn.close()

        time.sleep(1)
        conn = get_db_connection()
        cursor = conn.cursor()
        cursor.execute('SELECT * FROM users WHERE rfid = %s',(data['id'],))
        status = cursor.fetchone()
        conn.close()
```

Fig. 8 Python code for read RFID, update database fetch status

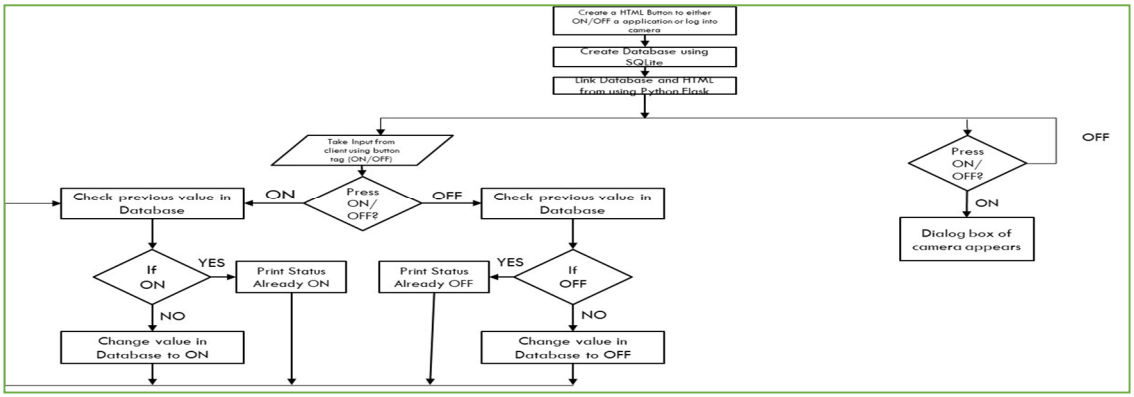


Fig. 9 Workflow of the Web-Based Interface

B. Hardware Implementation

The Hardware implementation involves connecting various sensors and components to the Raspberry Pi to ensure they work together to provide the desired functionality. The ultrasonic sensor is connected to the GPIO pins of the Raspberry Pi, with VCC and GND providing power. The RFID sensor is also connected to the GPIO pins for data transfer, with appropriate power connections. The solenoid lock, which secures entry points, is controlled by the Raspberry Pi through a relay module, which allows it to lock or unlock based on RFID input. The Pi camera is connected via the dedicated camera interface port on the Raspberry Pi, enabling image capture for security alerts. The relay module, which controls high-power electrical appliances, is connected to the GPIO pins, with the Raspberry Pi sending signals to switch appliances on or off. The bulbs and other appliances are interfaced with the relay module, allowing the Raspberry Pi to control their power states effectively. Overall, the system integrates these components to create a comprehensive home security and automation solution.

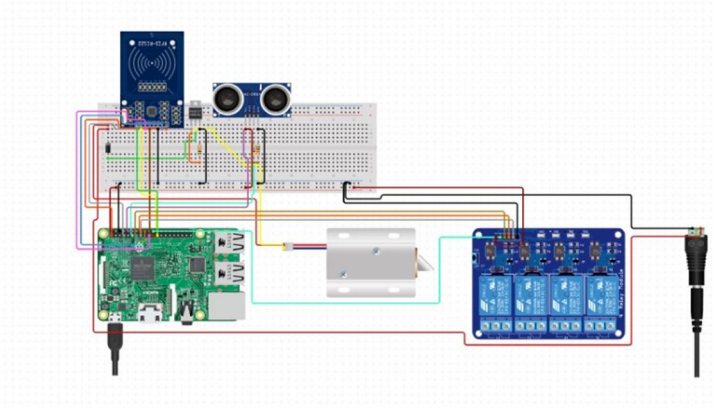


Fig. 10 Hardware Realization of GuardianPi with sensor and relay connections to the Raspberry Pi

VI. RESILT & DISCUSSION

The seamless integration of the web interface with the Raspberry Pi microprocessor was achieved through the use of both server-end and processor-side scripts. Users can access the system by going to the stated URL on any internet-enabled device, such as a phone or a computer. Upon successful authentication, users gain control over the connected appliances.

Upon logging in, users are greeted with a dashboard interface, providing access to real-time data on temperature and humidity levels and controls for various home appliances. The dashboard as shown in Fig. 11 offers an straightforward and user-friendly interface, making it easy for users to interact with the system.

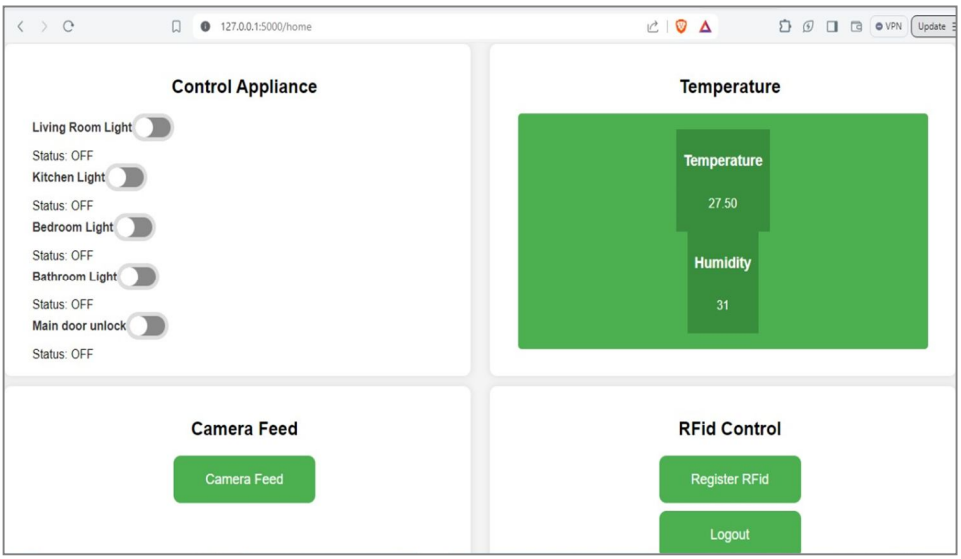


Fig. 11 Dashboard interface of Guardian Pi System

The setup of the GuardianPi system, demonstrated in Fig. 12 showcases the practical implementation of the project. This includes the Raspberry Pi connected to several sensors and a relay module, demonstrating the meticulous integration and planning involved. The physical model proves the feasibility of the system, with each component functioning as intended.



Fig. 12 Physical Model of the GuardianPi System

The system's alert mechanism is depicted in Fig. 13, which shows screenshots of the email notifications received by the user. These alerts confirm the system's capability to detect intrusions and other anomalies sending real-time notifications to the user. This ensures that users are immediately informed of any potential security breaches or environmental changes.

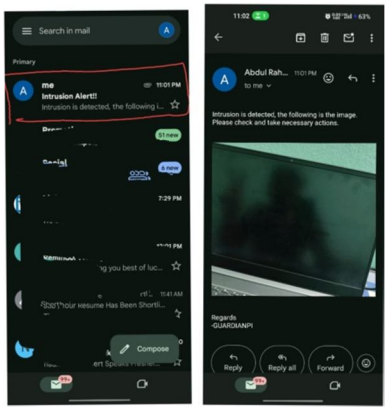


Fig. 13 Screenshots of alert emails received by the User.

The results indicate a robust and reliable system that effectively integrates various sensors and the Raspberry Pi for home automation and security purposes. The successful operation of the systems as shown in the above figures illustrates its viability and potential for further development in smart home technologies.

VII. CONCLUSIONS

This paper discusses the development and successful implementation GuardianPi, a unified home security and automation system. Users can remotely operate electrical appliances, monitor environmental conditions like temperature and humidity, and get security alerts- all through a user-friendly digital platform. This demonstrates that the system is practical and reliable. GuardianPi uniquely combines home automation and security features into a single platform, utilizing the capabilities of the Raspberry Pi. Through meticulous hardware and software design, the system efficiently manages multiple electrical loads and incorporates a range of smart features, such as remote entry, fire alerts and burglar alerts. Throughout the development process, significant research and effort were put into solving technical challenges, ensuring a seamless and user-friendly experience.

This project not only improves home convenience and security, but also lays a solid foundation for future advancements in smart home technologies. GuardianPi represents a significant step towards integrating comprehensive home management systems into everyday life.

VIII. FUTURE SCOPE

GuardianPi's flexible and scalable architecture allows for considerable improvements to increase home security and automation. By integrating advanced technologies, the system can provide more advanced features, improve user experience, and increase overall efficiency. The following improvements outline possible future developments:

A. Automated Emergency Calls

By integrating a GSM module, the system can automatically contact emergency services in critical situations.

B. Facial Recognition and Biometric Authentication

Incorporating facial recognition software, the system can enhance security by allowing only authorized individuals to access it.

C. Voice Control Integration

Adding a voice recognition module to this system, it can be operated using simple voice commands with popular voice assistants.

D. Geofencing and Location-Based Automation

By integrating GPS functionality to this module, the system can perform tasks based on the user's location, such as turning off appliances when the user leaves home.

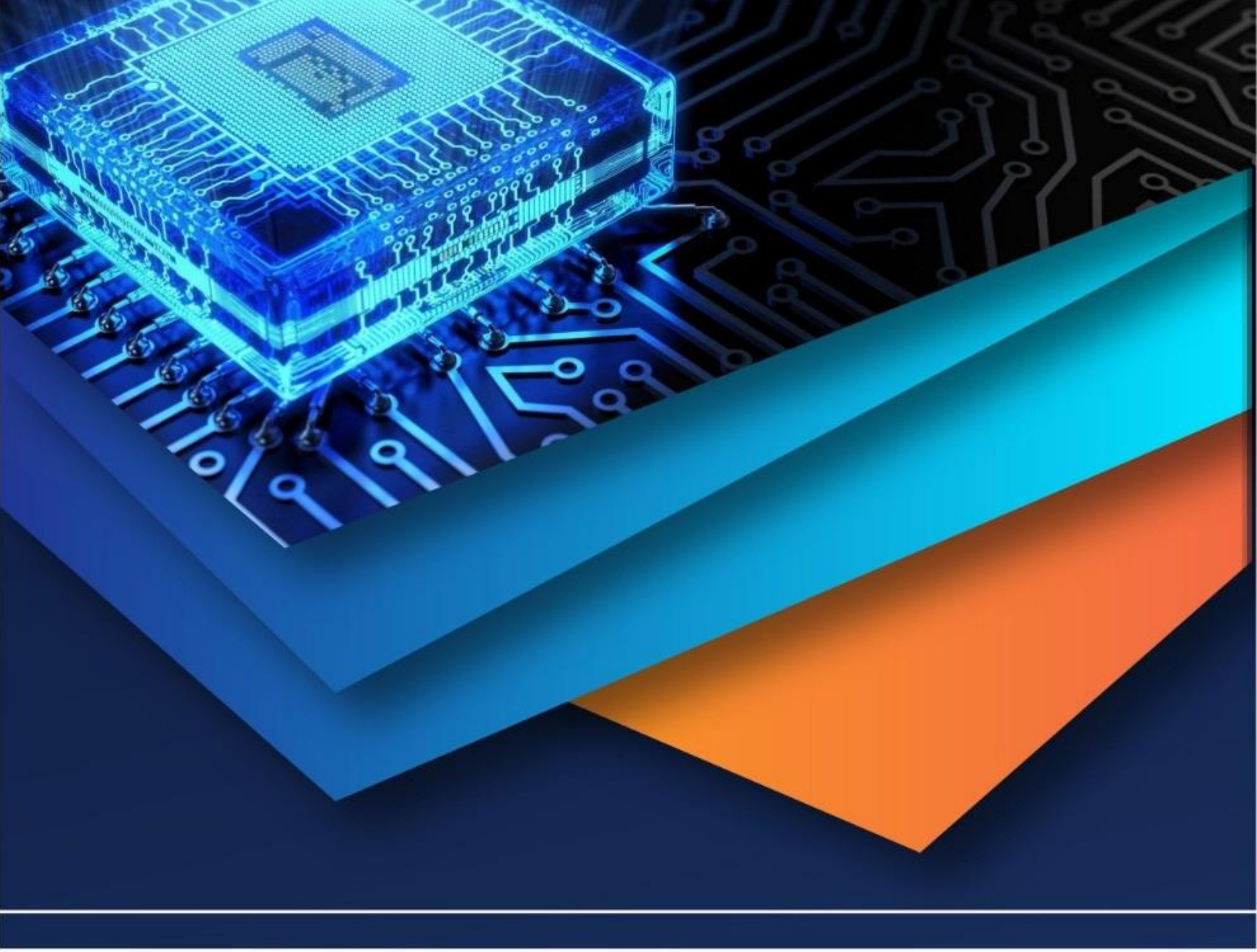
E. Energy Management

Integrating energy monitoring sensors and smart meters to this module, the system can monitor and optimize power consumption, schedule appliance operation and integrate renewable energy sources.

REFERENCES

[1] Siraj Ahmed, Mir Mohammed Ali, Mohammed Abdul Sami Siddiqui "Realization of a Method to Control Electrical Appliances through Web Interface" International Journal of Science and Research (IJSR) Vol. 4 Issue 11, PP. 2530-2532, Nov 2015.
[2] Aziz and M. S. Hussain, "IoT-based Smart Home Automation: A Review, "IEEE IoT Journal, vol. 7, pp.1-15, Jan. 2020.
[3] P. Verma and S. Sharma, "Wireless Sensor Networks for Home Automation: A Review, "Journal of Engineering Research & Technology, vol. 13, no. 8, pp. 1549-1555, Aug, 2020
[4] Isna K. and S. d. Sawant, "Integration of Cloud Computing and Internet of Things," International Journal of Advanced Research in Computer and Communication Engineering, vol/issue: 5($), Apr 2016
[5] S. Hilton. (2012, 14 January). Progression from M2M to the Internet of Things: an introductory blog. Available: http://blog.bosch-si.com/progression-fromm2m-to-internet-of-things-an-introductory-blog/.
[6] Anju, Neeraj Gupta, Arun Kumar Singh, "Home Automation Systems", 2011, International Journal of Engineering Science and Computing, ISSN 2321- 3361, Vol. 6 Issue 5, pp. 4828-4833, May 2016.
[7] H. Brooke Stauffer "Smart Enabling System for Home automation", 1991, IEEE Transactions on Consumer Electronics, Vol. 37(2), pp. 29-35.
[8] A. Alheraish, "Design and Implementation of Home Automation System", 2004, IEEE Transactions on Consumer Electronics, Vol. 50(4), pp. 1087-1092
[9] Documentation of Raspberry pi from https://www.raspberrypi.org/documentation
[10] S. R. Bharanialankar, C. S. Manikanda Babu, "Intelligent Home Appliance Status Intimation Control and System Using GSM", 2014, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4(4), pp.554 - 556.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)