AI-Integrated Smart Home Automation System Using Raspberry Pi and Google Assistant for Predictive Lights and Fans

Ganji jashwanth[1], Nasika Nagendra Prasad[2], Manikanetula Naga Venkata Krishnasai[3], Palepu karthik[4], Pothula Venkatalaksmi Ravi Kumar[5], U. Sravya[6]
[1,2,3,4,5]UG scholar, [6]Assistant Professor
Department of EEE, St.Ann's College of Engineering and Technology,Chirala,India.
Email id : jashwanthganji@gmail.com[1], prasadnasika435@gmail.com[2,] mkrishnasai2003 @gmail.com[3], karthiklucky0000@gmail.com[4], pvlravikumar@gmail.com[5], sravyaummareddy@gmail.com[6]

Article Received: 09 April 2025 *Article Accepted: 20 April 2025* *Article Published: 12 May 2025*

Citation
Ganji jashwanth, Nasika Nagendra Prasad, Manikanetula Naga Venkata Krishnasai, Palepu karthik, Pothula Venkatalaksmi Ravi Kumar, U. Sravya, "AI-Integrated Smart Home Automation System Using Raspberry Pi and Google Assistant for Predictive Lights and Fans", Journal of Next Generation Technology (ISSN: 2583-021X), 5(3), pp. 52-67, May 2025. DOI: 10.5281/zenodo.15715820

Abstract

The Voice-Controlled AI-Enabled Home Automation System enhances modern homes' convenience, safety, and energy efficiency by combining voice recognition, machine learning, and the growing Internet of Things. The system, which is outfitted with a range of sensors, such as light, temperature, humidity, infrared, and current sensors, uses real-time data to control home appliances in an intelligent manner. The system's main component is a Raspberry Pi, which can be controlled remotely using a smartphone app and has local speech recognition capabilities. Data transfers for continuous monitoring and analysis, together with alerts for overload and light failure, are sent to the ThingSpeak cloud platform. This state-of-the-art solution promotes smarter living by automating repetitive processes and optimizing resources. The AI-powered, voice-activated home automation system revolutionizes modern living by fusing IoT, machine learning, and advanced voice recognition. Depending on changing environmental data, the system uses a range of sophisticated sensors, including light, temperature, humidity, infrared, and current sensors, to intelligently regulate household appliances. Essentially, a Raspberry Pi coordinates local speech recognition and enables seamless remote control through a dedicated mobile application. Real-time data transmission to the ThingSpeak cloud facilitates anomaly identification, predictive analytics, and continuous monitoring, including overload and lighting system failures. This cutting-edge automation framework enhances convenience, maximizes energy efficiency, and fortifies home security, marking the next phase in intelligent living environments.

Keywords: Electric Vehicle Charging Station (EVCS), Photovoltaic , Wind Energy, Fuzzy Logic Controller MPPT.

I. INTRODUCTION

Smart home automation has quickly become a game-changing way to increase security, energy efficiency, and comfort in homes. Intelligent home settings may now react to environmental inputs and user behaviors on their own thanks to the convergence of embedded systems, artificial intelligence (AI), and the Internet of Things (IoT) [1][2].Real-time monitoring and management of household appliances is made possible by the integration of IoT devices, which range from sensors to actuators. These gadgets use wireless protocols to connect with one another and produce an ongoing flow of environmental data that may be examined for automation purposes [3][4]. Gubbi and associates. Apple Siri, Amazon Alexa, and Assistant let users to operate gadgets without the need for human intervention [6][7]. Voice interfaces greatly improve accessibility and convenience, particularly for the elderly and disabled, according to studies [8][9]. Because of its low cost, GPIO flexibility, and Linux-based open ecosystem, Raspberry Pi has emerged as the go-to edge computing platform for smart home projects [10]. Robust environmental sensing and automation are made possible when combined with sensors such as DHT11 (temperature/humidity), LDR (light), IR (motion), and ACS712 (current) [11][12]. Fans, lights, and HVAC systems can all be controlled finely using such hardware configurations. Predictive control using machine learning models on sensor data enables systems to gradually adjust to user preferences. For instance, using usage patterns and environmental factors, supervised learning systems can predict when to turn on devices [13][14]. It is well established that predictive automation improves system responsiveness and lowers wasteful energy use [15]. Platforms such as ThingSpeak and Blynk provide remote access and monitoring, enabling customers to observe and manage their home systems from any location [16]. Additionally, historical data analytics and anomaly detection—such as identifying unusual power usage or component failure—are made possible by cloud integration [17]. One of the biggest concerns with smart homes is still security. To avoid unwanted access, data must be encrypted and verified before being sent between systems and devices [18].

Roman et al. [19] emphasize how crucial it is to use lightweight encryption and security mechanisms in IoT systems with limited resources. Additionally, the adoption of smart homes depends on user-centric design. To serve a wide variety of users, systems need to be inclusive, flexible, and easy to use [20]. Voice interaction, cloud analytics, and local intelligence (at the edge) are all seen to be a potent mix for next-generation home automation systems. Manual controls or set schedules are the foundation of traditional home automation systems, which might be ineffective and fail to take current circumstances into account. It is not flexible enough to adjust to changing environmental conditions or user preferences. usually demand for a skilled setup and pricey hardware. Voice-Activated Devices (like Google Home and Alexa) restricted to vocal commands and lack the ability to infer human demands from sensor data. frequently depends on cloud computing, which could cause network and privacy problems. limited ability to integrate local appliances and sensors. The primary limitations include the inability to anticipate user demands based on sensor data and the restriction to voice requests. frequently depends on cloud computing, which could cause network and privacy problems. limited ability to integrate local appliances and sensors. This paper presents a Raspberry Pi–based, voice-controlled smart home system that combines IoT sensors, machine learning, and real-time cloud integration for intelligent control of lighting and fan systems[21]-[23]. The goal is to deliver a scalable, cost-effective, and energy-efficient automation framework suitable for modern households.

II. SYSTEM ARCHITECTURE

The AI-Integrated Smart Home Automation System With Raspberry Pi And Google Assistant For Predictive Lights And Fans is depicted in the block diagram in Figure 1. The primary goal is to create a voice-activated automation system that uses sensor data and spoken commands to control home appliances. In order to forecast the requirement for lighting, cooling, and other appliance usage, it is necessary to incorporate IoT sensors (temperature, humidity, LDR, IR, and current sensors). integrating a microphone with a Raspberry Pi to enable local speech control and recognition. to make it possible to monitor and control the system remotely using a mobile application. Assure efficiency and safety by providing real-time alerts for overloads and light failures. Utilize machine learning to include environmental condition-based predictive automation..

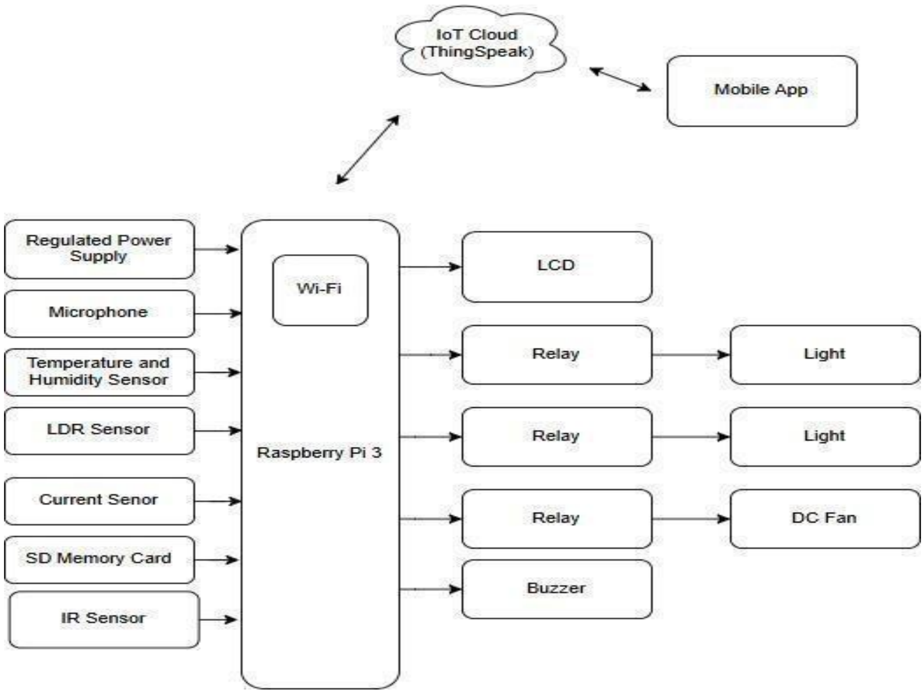


Figure 1:Block Diagram

The suggested solution combines cloud-based monitoring, machine learning, sensor integration, and voice recognition to provide intelligent and automated control of household appliances. Users can operate appliances remotely by utilizing a Raspberry Pi with speech-to-text software to handle voice instructions locally. In order to support decision-making about appliance operation, sensors for temperature, humidity, light intensity, infrared presence, and current flow simultaneously collect real-time environmental data. In order to enhance energy efficiency, the system evaluates this data to decide whether to turn appliances on or off.

Machine learning algorithms are used to process previous data and forecast future usage trends in order to further improve performance. With the help of this predictive feature, the system may gradually adjust to user preferences, guaranteeing more precise and effective appliance control. Real-time updates, user notifications, and remote data monitoring are made possible via integration with the ThingSpeak cloud platform. Users can access fault alarms, control features, and system status using a specialized mobile application. In order to prevent harm and guarantee safety, the system also keeps an eye out for situations like overloads or component failures and sends out instant alerts. Software programming, cloud integration, and hardware interfacing are all included in the system's structured development process. Using GPIO or I2C communication protocols, the Raspberry Pi serves as the main controller and is connected to a microphone and other environmental sensors. Python modules that can translate voice input to text and match it to preset commands are used for speech recognition.

Continuously gathered sensor data is utilized to build machine learning models that forecast the need for appliance control based on historical usage and current conditions. A smartphone application that connects to the ThingSpeak platform for data retrieval and command execution is made to enable remote access to system functionalities. ThingSpeak also serves as a cloud-based analytics tool, providing visualizations and enabling alerts based on sensor inputs and system events.

A. System Modules

There are four functional modules that make up the system architecture. The microphone and Raspberry Pi are part of the voice recognition module, which processes voice commands. The sensor module keeps an eye on the conditions of the appliance and the surroundings using temperature, humidity, LDR, IR, and current sensors. Relays are used by the control module to process voice and sensor inputs and control appliances. For data logging, analysis, and remote control through the mobile application, the cloud integration module links the system to ThingSpeak.

B. Implementation Steps

The first step in implementation is hardware setup, which involves connecting the Raspberry Pi to peripherals like a microphone and sensors. To guarantee precise readings, sensors are calibrated. Python libraries are used to create voice recognition capabilities, and control logic is written to react to user commands and sensor inputs. ThingSpeak is set up to enable remote monitoring and alerting services as well as to receive real-time data from the system. After that, a mobile application is created or modified to communicate with ThingSpeak and give users feedback and control. To confirm accuracy and operation, the system is put through a rigorous calibration and testing process.

C. Hardware Tools

A Raspberry Pi (ideally a Raspberry Pi 4), a microphone to record voice commands, a DHT11 or DHT22 sensor to measure temperature and humidity, an LDR to detect light, an IR sensor to detect motion, an ACS712 current sensor, relay modules to switch appliances, and a smartphone to access mobile applications are among the hardware components needed for the system.

D. Software Tools

The software stack consists of Python libraries like SpeechRecognition for voice processing, Scikit-learn for machine learning method implementation, and the Raspberry Pi OS as the operating system. Data analytics and cloud-based monitoring are done with the ThingSpeak platform. MIT App Inventor or a comparable program is used to create the mobile application, and the Arduino IDE may be used for sensor configuration or testing if necessary.

E. INITIAL DESIGN CONSIDERATIONS

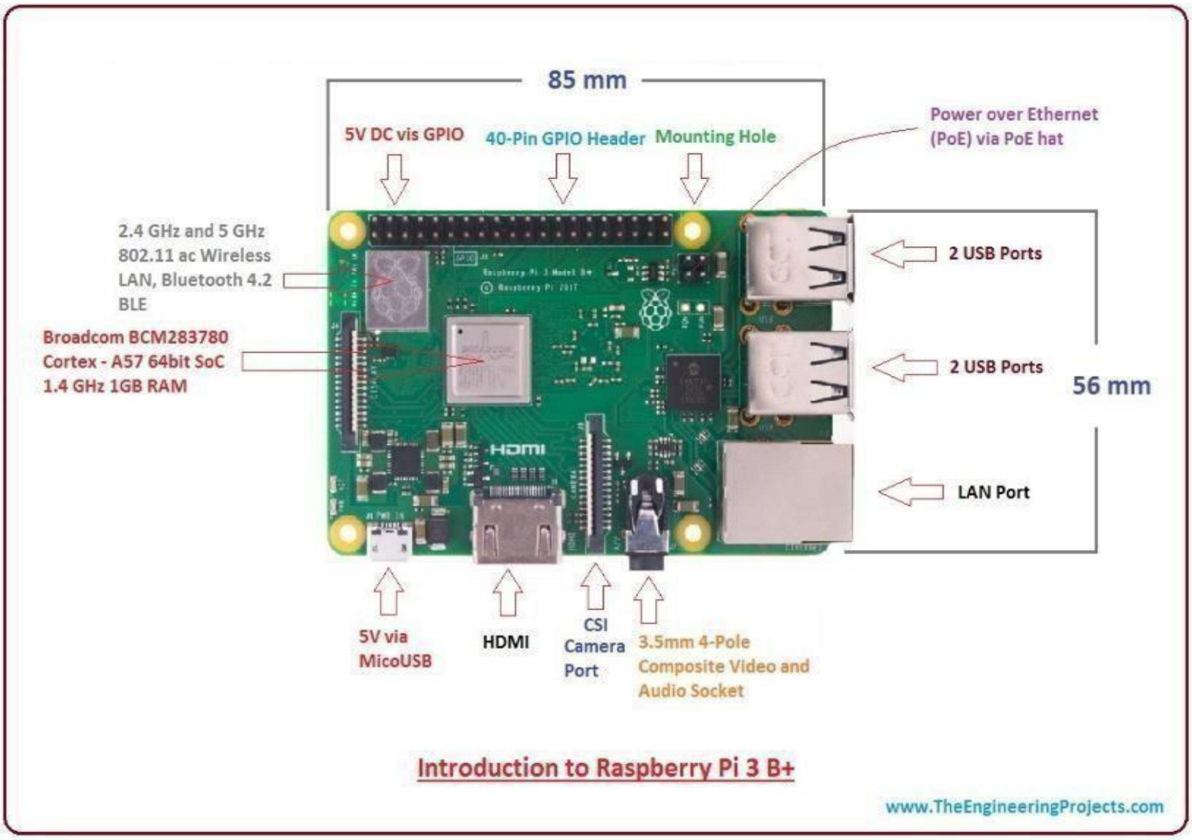


Figure 2:Hardware Layout of Raspberry Pi3B+

III. A BRIEF DESCRIPTION OF THE COMPONENTS ON THE PI:

A. Raspberry Pi Usb Plug And Play Desktop Microphone:

Constructed from aluminum and polycarbonate, it is incredibly lightweight and robust, and its advanced digital USB offers exceptional clarity. As a single USB plug-and-play connection, it is easy to use. Unwanted background noise is eliminated with a noise-cancelling microphone. When the microphone is turned on, the power switch glows. To maintain the desired position, the microphone pivots on the base. A flexible neck allows you to change your speech height and direction as needed.

Figure 3:Raspberry pi USB plug and play desktop microphone

Table 1 : Specifications of microphone

Good Capture Range (m)	1
Sensitivity (dBm):	-3dB at 1.5V, Reduction
Frequency response(kHz)	100 ~ 16
Material:	Plastic and metal
Cable Length (cm):	90
Microphone length(cm)	13
Shipping Weight	0.04 kg
Shipping Dimensions	11 × 10 × 6 cm

B. LCD

Two registers, such as a data register and a command register, are present in a 16x2 LCD. The primary function of the register select (RS) is to switch between registers. The register set is referred to as a command register when it is set to "0." Likewise, a register set that has a value of 1 is referred to as a data register..

Command Register

The main function of the command register is to store the instructions of command which are given to the display. So that predefined tasks can be performed such as clearing the display, initializing, set the cursor place, and display control. Here commands processing can occur within the register.

Data Register

In this case, the information that will be displayed on the LCD screen is the character's ASCII value. Every time we provide data to the LCD, it is transmitted to the data register, where the process begins. The data register will be chosen when register set = 1.The commands of LCD 16X2 include the following.

The clear LCD screen will be the LCD command for Hex Code-01.
The LCD command for Hex Code-02 will be going back home.
The LCD command for Hex Code-04 will be decrement cursor.

The LCD command for Hex Code-06 will be the increment cursor.
The LCD command for Hex Code-05 will be Shift display right.
Shift display left will be the LCD command for Hex Code-07.
The LCD command for Hex Code-08 will be "Display off, cursor off."
The LCD command for Hex Code-0A will be display off and cursor on.
The LCD command for Hex Code-0C will be display on, cursor off.
The LCD instruction for Hex Code-0E will be "Cursor blinking, Display on."
The LCD instruction for Hex Code-0F will be "Cursor blinking, Display on."
Shift the cursor position to the left will be the LCD instruction for Hex Code-10.
Shift the cursor position to the right will be the LCD instruction for Hex Code-14.
Shift the entire display to the left is the LCD command for Hex Code-18.
Shift the entire display to the right is the LCD command for Hex Code-1C.
The LCD instruction for Hex Code-80 will be to force the cursor to the first line.
The LCD command for Hex Code-C0 will be Force cursor to begin (second line).
The LCD instruction for Hex Code-38 will be a 5x7 matrix with two lines.

C. DHT11–TEMPERATURE AND HUMIDITY SENSOR

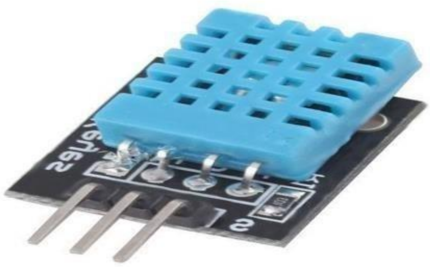


Figure 4:DHT11–Temperature and Humidity Sensor

The DHT11 Sensor is very simple to set up because it is factory calibrated and delivers serial data. This sensor's connection diagram is displayed below. As you can see, a 5K pull-up resistor is employed, and the data pin is connected to an MCU I/O pin. The temperature and humidity values are serially output from this data pin. There are pre-made libraries for DHT11 that will help you get started quickly if you're attempting to interface it with Arduino.

The datasheet provided below will be useful if you are attempting to interface it with another MCU. The data pin's output will look like this: 8 bits of integer humidity data + 8 bits of humidity decimal data + 8 bits of integer temperature data + 8 bits of fractional temperature data + 8 bits of parity bit. As seen in the timing diagram below, the I/O pin must be briefly held low and then held high in order to request that the DHT11 module provide these data. The duration of each host signal is explained in the DHT11 datasheet, with neat steps and illustrative timing diagrams

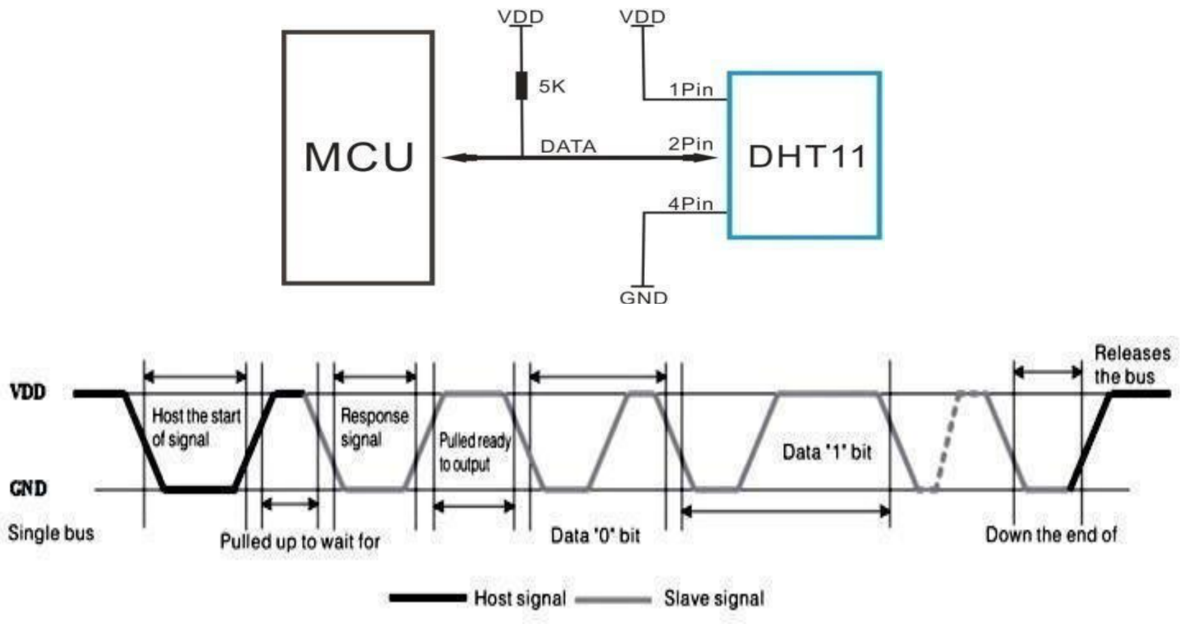


Figure 5: DTH11 Sensor connecting with MCU

Table 2: PIN IDENTIFICATION AND CONFIGURATION

No:	Pin Name	Description
For DHT11 Sensor		
1	Vcc	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	NC	No Connection and hence not used
4	Ground	Connected to the ground of the circuit
For DHT11 Sensor module		
1	Vcc	Power supply 3.5V to 5.5V

2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

D. IR(Infrared)PROXIMITY SENSOR

An infrared sensor is an electronic device that scans for infrared signals within a standardized frequency range and transforms them into electrical signals on its digital output pin, which is often an OUT PIN. Touchscreen phones and other gadgets employ proximity sensors. During the call, the display is turned off so that it won't be impacted even if the cheek touches the touchscreen.

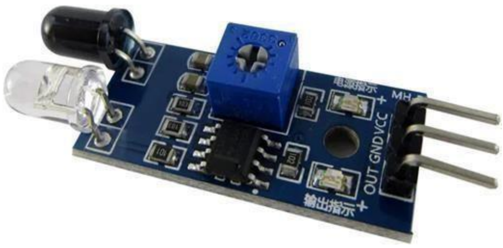


Figure 6: IR Sensor

This multifunctional infrared sensor (IR proximity sensor) from Easy Electronics can also be utilized as an encoding sensor, a barrier sensor, and a line sensing robotic sensor. A digital output of 1 or 0 is produced when an object with a logical zero (0V) output is positioned in front of the sensor. Use the built-in LED indications to check if the device is receiving adequate power, then start playing with your reasoning.

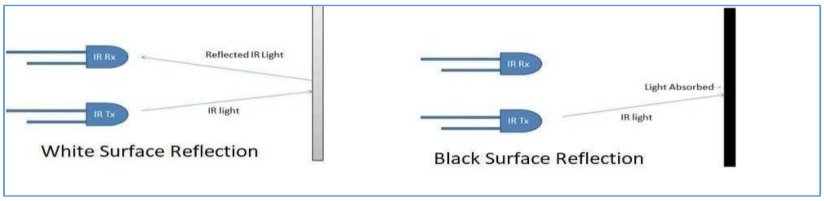
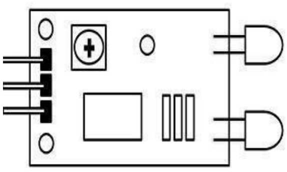


Figure 7: Working Principle



Pin Name	Description
VCC	Power Supply Input
GND	Power Supply Ground
OUT	Active High Output

Figure 8:Pin configuration

E. FOUR-CHANNEL RELAY MODULE

Since there are four relays on the same module, the four-channel can be utilized to switch numerous loads simultaneously. This is helpful for setting up a central hub that can power several distant loads. It is helpful for projects like home automation, where a microcontroller can be used to control the module from a central location. The module can be installed in the main switchboard and connected to loads in different areas of the house.

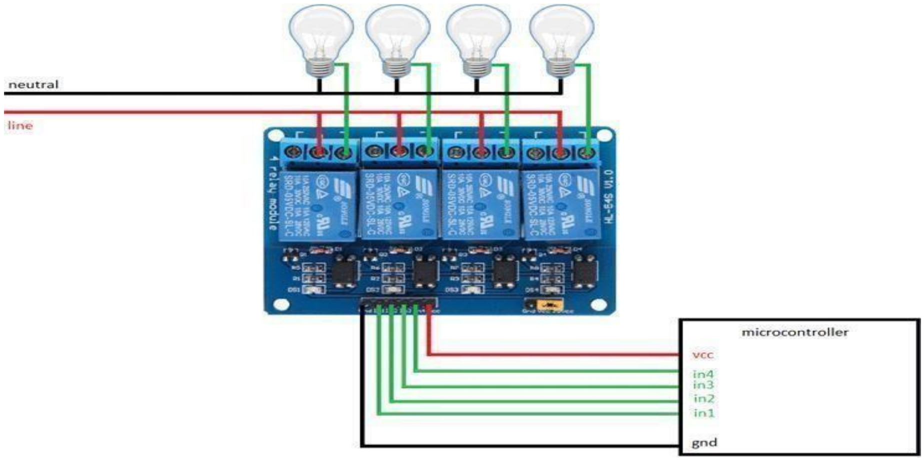


Figure 9: Connections of four channel relay module

Four distinct loads—represented by lightbulbs—have been attached to the relay's NO terminals in this diagram. Each relay's common terminal has been linked to the live wire. The load is powered and connected to the live wire when the relays are turned on. By connecting the load to the NC terminal, which keeps it turned on until the relay is activated, this configuration can be reversed.

F. ACS712 CURRENT SENSOR MODULE

Figure 10:ACS712 Current sensor module

It is a 5A, 20A, and 30A module that measures both AC and DC current. Because it outputs analog voltage, it offers separation from the load and is simple to integrate with MCU. The ACS712 Module interfaces with microcontrollers quite easily.

The diagram below would provide greater context.

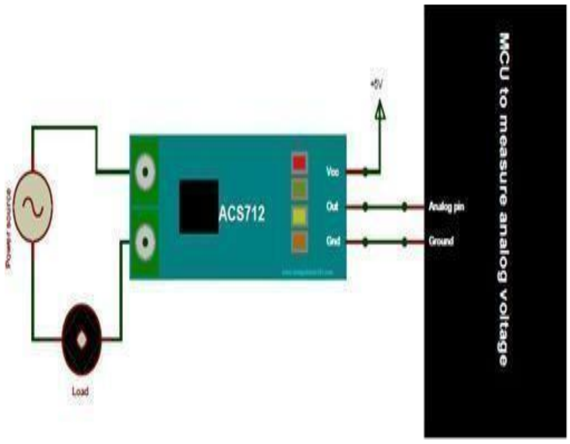


Figure 11: Connections ACS712 Current sensor module

Before we can program our microcontrollers to read current from the ACS712 Module, there are a few things we need to know. The output voltage will be +2.5V (Vcc/2) by default when there is no current flowing through the module terminals; if current flows in one direction, the value will rise from 2.5V, and if it flows in another direction, the value will fall from 2.5V. We can measure both AC and DC current in this method thanks to the module. Assuming that the microcontroller you are using has a 10-bit ADC and runs at 5V with a 5V reference voltage for ADC conversion, the microcontroller will be able to read ADC values between 0 and 1024. The output voltage from ADC values can then be determined using the formulae below.

(ADC Value/1023)*5000 = Vout (mV)

Once the output voltage has been determined, we can use the following formulas to determine the current value from the voltage.

The wire's current (A) is equal to (Vout(mv)-2500)/Scale factor.Note that the value of scale factor changes for every module based on its range. The values of scale factor for all three modules are given in the specifications above.

IV. SOFTWARE SETUP

A. SET UP THONNY

Press the BOOTSEL button on your Pico and use a USB cable to connect it to your PC. Go to Run > Select interpreter, then pick Raspberry Pi Pico (MicroPython).Installing or updating firmware is also a smart idea. This will install MicroPython on your Pico if it hasn't already been installed or update it with the most recent version.

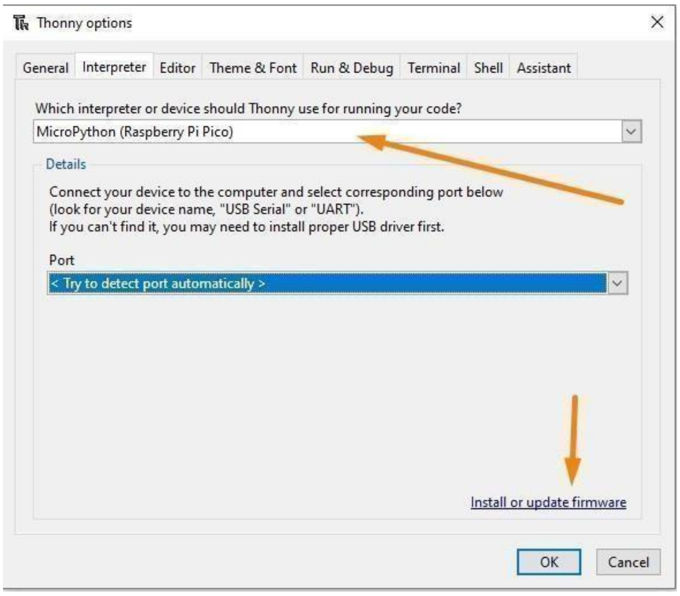


Figure 12: Set up thonny

B. REPL INTERFACE (Shell)

We can begin running code in the REPL right away by entering the following code in the shell tab: "Hello, World!" is printed.
Your Pico will receive the command, perform it, and show the message back to you.We can also take control of the on-board LED by executing the following code:

```
from machine import

led = Pin(25, Pin.OUT)

led.toggle(
```

This code will toggle the LED. If you keep executing led.toggle() the LED will keep changing state.

LED Coding:

```
MicroPython v1.13-290-g556ae7914 on
Type "help()" for more information.
>>> print("hello, world!")
 hello, world!
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.toggle()
>>> led.toggle()
```

Create a new script with File>New and paste in the following code:

Save the script - you will be prompted to save to your computer OR the pico.
Select save to Pico and name the file main.py
Return to the REPL and press Ctrl+D (or use the Stop/Restart button) to restart your Pico. The

LED should flash at a steady rate **and** the shell should begin printing multiples of thirteen.

```
from machine import

Pin from time import

sleep led = Pin(25,

Pin.OUT) n = 0


   print("13 x {} = {}".format(n, 13*n)) # print the thirteen-times

   n = n + 1

   sleep(0.5
```

THINGSPEAK IOT PLATFORM

ThingSpeak is an Internet of Things platform that allows users to collect data in real time, such as location, climate, and other device data. You can summarize information, see data online in charts, and study relevant data in various ThingSpeak channels. IoT:bit (micro:bit) and other hardware/software platforms can be integrated with ThingSpeak. You can input sensor data (such as temperature, humidity, light intensity, noise, motion, raindrop, distance, and other device information) to ThingSpeak using IoT:bit.

Figure 13:ThingSpeak IOT Platform

V. RESULTS AND DISCUSSION

By combining voice control, cloud-based monitoring, and real-time sensor data, the Voice-Controlled AI-Enabled Home Automation System has shown promise in improving home automation. By effectively automating appliance control, the system optimizes energy use according to user preferences and environmental conditions. Furthermore, by using machine learning, the system can become more sophisticated over time and provide predictive control for household appliances.

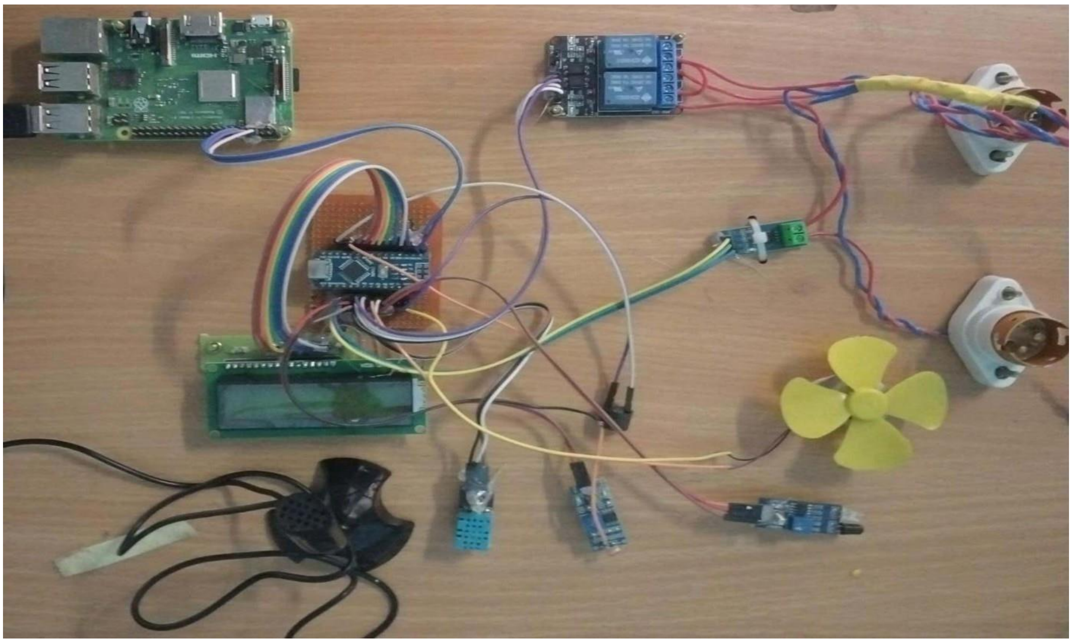


Figure 14: Hardware setup of proposed system

VI. CONCLUSION

The suggested system gathers data on the home environment in real time by using a network of sensors to monitor variables including temperature, humidity, motion, and ambient light. The system uses clever algorithms to process this data, learning from user activity patterns and environmental changes to make well-informed judgments about when and how to operate devices. To save energy and increase comfort, lights and fans, for instance, can be programmed to turn on or off automatically depending on occupancy and surrounding conditions. AI-based speech recognition technologies facilitate voice interaction, allowing users to control systems and appliances with straightforward spoken commands. Cloud connectivity gives users control over their home environment even when they are not there by enabling data synchronization, remote access, and continuous monitoring. A secure and dependable system is enhanced by extra safety measures like alerts for anomalous activity or environmental dangers. The technology offers a customized and effective smart living solution as it grows more accurate and adaptive with continued use.

References

[1]. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). *Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications*. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376.

[2]. Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). *Internet of Things for Smart Cities*. IEEE Internet of Things Journal, 1(1), 22–32.

[3]. Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). *Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges*. In 10th International Conference on Frontiers of Information Technology (pp. 257–260).
[4]. Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., & Qiu, D. (2014). *Security of the Internet of Things: Perspectives and Challenges*. Wireless Networks, 20(8), 2481–2501.
[5]. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). *Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions*. Future Generation Computer Systems, 29(7), 1645–1660.
[6]. Porambage, P., Schmitt, C., Gurtov, A., & Ylianttila, M. (2016). *PAuthKey: A Pervasive Authentication Protocol and Key Establishment Scheme for Wireless Sensor Networks in Distributed IoT Applications*. International Journal of Distributed Sensor Networks, 12(3), 1–14.
[7]. Kocabas, O., & Soyata, T. (2017). *Towards Privacy-Preserving Data Analytics for IoT Edge Devices with Built-in AI Capabilities*. Computer Science Review, 27, 34–46.
[8]. Pal, D., Funilkul, S., Charoenkitkarn, N., & Kanthamanon, P. (2018). *Internet-of-Things and Smart Homes for Elderly Healthcare: An End-User Perspective*. IEEE Access, 6, 10483–10496.
[9]. Rashid, M. M., Mehedi, J. M., & Reaz, M. B. I. (2019). *Voice Controlled Smart Home Automation System for Disabled People*. IEEE Access, 7, 133969–133976.
[10]. Upton, E., & Halfacree, G. (2014). *Raspberry Pi User Guide*. John Wiley & Sons.
[11]. Ahmed, A. F., Hassan, R. A., & El-Sayed, A. H. (2019). *IoT Based Smart Home Automation Using Raspberry Pi*. International Journal of Scientific & Engineering Research, 10(5), 558–564.
[12]. Mehra, M. S. (2022). *IoT Based Smart Home Automation System Using Raspberry Pi*. International Journal of Engineering Research & Technology (IJERT), 11(7), 377–381.
[13]. Hussain, M., Alhamid, M. F., Alhadlaq, A., & Hossain, M. S. (2017). *A Framework for Cloud-Based Context-Aware Personalized Services*. IEEE Communications Magazine, 55(2), 44–50.
[14]. Xu, L. D., He, W., & Li, S. (2014). *Internet of Things in Industries: A Survey*. IEEE Transactions on Industrial Informatics, 10(4), 2233–2243.
[15]. Srinivasan, K., & Lee, J. (2015). *Energy-Aware Scheduling for IoT Applications on Heterogeneous Systems*. ACM Transactions on Embedded Computing Systems (TECS), 14(3), 1–26.
[16]. MathWorks. (2023). *ThingSpeak™ IoT Platform Documentation*. [Online]. Available: https://www.mathworks.com/help/thingspeak
[17]. Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2012). *A Review of Smart Homes – Past, Present, and Future*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6), 1190–1203.
[18]. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). *Security, Privacy and Trust in Internet of Things: The Road Ahead*. Computer Networks, 76, 146–164.
[19]. Roman, R., Najera, P., & Lopez, J. (2011). *Securing the Internet of Things*. Computer, 44(9), 51–58.
[20]. Zheng, Y., & Ni, L. M. (2015). *Smart Sensing and Context-Aware Computing*. IEEE Computer, 48(2), 16–18.

[21]. Farooq Sunar Mahammad, Sai Shreya Sudireddy.(2021). An Efficient Home Automation and Security System using Arduino and 1-Sheeld, Journal of Next Generation Technology, 1(1), 29-33.

[22]. C.Peter Devedoss, R.Dharini Rathika, S.Kavitha, M.Padmapriya, B.Ragavi, “Smart Farming AI Powered Assistant”, Journal of Next Generation Technology (ISSN: 2583-021X), 5(2), pp. 81-92. April 2025.

[23]. R. A. Karthika, R. A, P. Muthukumar and S. P. John, "Real-Time Detection of Physical Altercations Using AI Frameworks," *2024 International Conference on Advancement in Renewable Energy and Intelligent Systems (AREIS)*, Thrissur, India, 2024, pp. 1-6, doi: 10.1109/AREIS62559.2024.10893639.