IntelliHome: The automated household infrastructure

A Project Report

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE WITH SPECIALIZATION IN

Internet of Things

Submitted by:

Ananya Singh (20BCS4585)
Siddharth Singh(20BCS6103)

Under the Supervision of:
Dr. Geeta Rani

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI - 140413, PUNJAB

November, 2023

CERTIFICATE

This is to certify that major project entitled "**IntelliHome: The automated household Infrastructure"** in field of Major Project is a Bonafede work carried out in the seventh and eighth semester by "*Ananya Singh"," Siddharth Singh*" who carried out the project work under the supervision of "*Dr. Geeta Rani"* in partial fulfillment for the award of the degree of Bachelor of Engineering in Computer Science with specialization in Internet of things during the academic year 2023-2024.

(Signature of the HOD)

NAME & SIGNATURE

(Signature of the Supervisor)

NAME & SIGNATURE

ABSTRACT

The integration of automation technologies into household infrastructure has revolutionized the way we interact with and manage our living spaces. This project explores the concept of automated household infrastructure, which encompasses the deployment of smart systems and devices to enhance convenience, energy efficiency, security, and overall quality of life. The objective of this study is to provide an overview of the various automated technologies available for residential environments, along with their benefits and potential challenges.

The paper begins by outlining the key components of automated household infrastructure, including smart appliances, lighting systems, climate control, security systems, entertainment setups, and more. It delves into the mechanisms that enable seamless communication and coordination among these components, such as Internet of Things (IoT) platforms, sensor networks, and centralized control systems. Efficiency and energy conservation are major focal points of automated household infrastructure. The ability to remotely monitor and regulate energy usage not only reduces utility costs but also contributes to a more sustainable environment. Moreover, the paper discusses the importance of data security and user privacy in the context of the interconnected devices that collect and exchange personal information.

The benefits of automated household infrastructure extend beyond efficiency and security. The paper highlights the convenience and enhanced quality of life offered by features like voice-activated assistants, automated routines, and personalized settings tailored to individual preferences. Additionally, the potential challenges associated with technology obsolescence, interoperability issues, and the digital divide are addressed.

In conclusion, the paper underscores the transformative impact of automated household infrastructure on modern living. The integration of automation technologies has the potential to revolutionize domestic life by simplifying tasks, improving resource management, and creating more comfortable and secure living environments. As technology continues to evolve, further advancements in automated household infrastructure are anticipated, prompting the need for ongoing research and dialogue to maximize its benefits while mitigating potential drawbacks.

TABLE CONTENT

1. INTRODUCTION

The rapid advancement of technology has significantly altered the way we live, work, and interact with our surroundings. One prominent aspect of this technological evolution is the automation of household infrastructure, which entails the integration of smart systems and devices to streamline and enhance various aspects of domestic life. From managing energy consumption and security to providing unprecedented convenience, the concept of automated household infrastructure has reshaped the traditional idea of home management.

In the not-so-distant past, household tasks required substantial manual effort and attention. From adjusting thermostat settings to physically checking whether doors were locked, these mundane activities consumed time and energy. However, with the proliferation of interconnected devices, sensor networks, and intelligent algorithms, homes are becoming more intelligent and responsive, allowing residents to delegate routine tasks to technology.

This paper delves into the realm of automated household infrastructure, aiming to illuminate its transformative potential, benefits, challenges, and implications for modern living. By exploring the convergence of technologies such as the Internet of Things (IoT), artificial intelligence (AI), and data analytics, we can uncover how these innovations have paved the way for a more interconnected and efficient home environment. From optimizing resource usage to enabling remote control and personalization, automated household infrastructure promises to redefine the way we experience and manage our living spaces.

Through a comprehensive examination of the components, functions, and impacts of automated household infrastructure, we aim to shed light on its far-reaching implications for homeowners, manufacturers, policy makers, and society at large.

As this paradigm shift in home management continues to gather momentum, it is imperative to understand the dynamics at play, address potential challenges, and envision a future where our homes become not just smart, but intelligent companions that cater to our needs and aspiration

1.1. Problem Definition

Houses without automated infrastructure can face a range of challenges that impact convenience, energy efficiency, security, and overall quality of life. Here are some problems commonly experienced in non-automated households:

1. **Inefficient Energy Usage**: Without automation, residents might forget to turn off lights, appliances, or thermostats when not needed, leading to unnecessary energy consumption and higher utility bills.

2. **Security Concerns:** Manual security systems are more prone to human error. Forgetting to lock doors, close windows, or activate alarms can leave the house vulnerable to intrusions.

3. **Time-Consuming Routine Tasks:** Residents must manually perform routine tasks like adjusting thermostat settings, turning on/off lights, and locking doors, which can be time-consuming and add to daily stress.

4. **Limited Remote Control:** In the absence of automation, controlling household devices remotely isn't possible. This lack of remote control can cause inconvenience, especially when homeowners are away.

5. **Lack of Personalization:** Non-automated homes don't have the capability to personalize settings based on individual preferences and occupancy patterns, leading to less comfortable living conditions.

6. **Difficulty in Maintenance:** Without automated diagnostics, identifying issues with appliances or systems becomes challenging, potentially leading to delayed maintenance or unexpected breakdowns.

7. **Wasted Resources:** Manual resource management can lead to wastage, such as over-irrigating plants, overusing water, or not optimizing energy consumption.

8. **Limited Accessibility for Elderly and Disabled**: Houses lacking automation might pose challenges for elderly or disabled individuals who struggle with physical tasks, such as opening doors or adjusting lighting

9. **Lack of Data-Driven Insights:** Without automation, homeowners miss out on valuable data regarding energy consumption, occupancy patterns, and usage trends, which could inform smarter decisions.

10. **Increased Human Error**: The absence of automation increases the likelihood of human error, leading to potential safety hazards, such as forgetting to turn off a stove.

In summary, households lacking automated infrastructure often grapple with inefficiency, inconvenience, security vulnerabilities, and missed opportunities for energy savings and personalization. As technology continues to evolve, the benefits of automation offer a compelling case for enhancing the overall living experience.

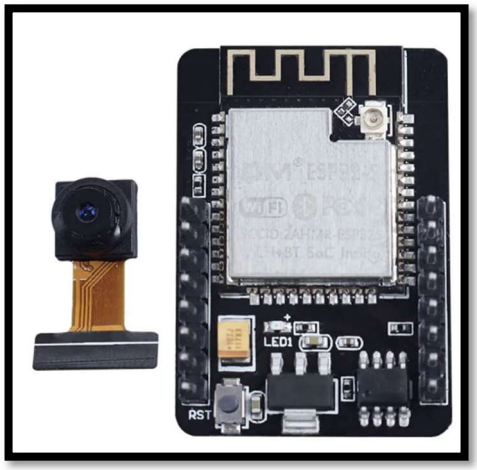
1.2 Project Overview

The project aims to revolutionize the way households function by implementing an advanced automated infrastructure that integrates smart technologies and systems. The project envisions creating an intelligent living space that enhances convenience, energy efficiency, security, and overall quality of life for residents. Through the seamless integration of various smart devices, sensors, and control systems.

The project aims to enhance convenience, efficiency and control through integration of technologies. This includes automated and controlled lighting, temperature, security and more using different sensors. The main highlight of the project is the use of OpenCV for object detection to implement smart refrigerators, automated cleaning systems and face recognition systems for enhanced security and privacy. The project further includes the scope of energy management as per homeowner's requirements

1.3. Hardware Specification

- ESP32 Wi-Fi & ESP32-CAM Module



The ESP32 is a popular microcontroller that integrates Wi-Fi and Bluetooth capabilities. It's widely used in the development of IoT (Internet of Things) projects and is known for its versatility and ease of use. The ESP32-CAM module is a specific variant of the ESP32 that includes a camera module, making it suitable for projects involving image and video processing.

Here are some key features and information about the ESP32 Wi-Fi module and the ESP32-CAM module:

ESP32 Wi-Fi Module:
Microcontroller: The ESP32 is based on the Tensilica Xtensa LX6 microcontroller.

Wireless Connectivity: It features built-in Wi-Fi (802.11 b/g/n) and Bluetooth (Bluetooth Classic and BLE) capabilities, making it suitable for a wide range of IoT applications.

Processing Power: The dual-core processor in the ESP32 allows for efficient multitasking and processing.

Peripheral Interfaces: The ESP32 provides a variety of GPIO (General Purpose Input/Output) pins, SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver/Transmitter), and more, making it versatile for interfacing with various sensors and devices.

Development Environment: Arduino IDE and PlatformIO are commonly used development environments for programming the ESP32.

Open-Source: The ESP32 is supported by an active and open-source community, contributing to its popularity.

ESP32-CAM Module:
Camera Module: The ESP32-CAM module includes a camera that allows for image and video processing.

OV2640 Camera: The most common camera module used with ESP32-CAM is the OV2640, which is capable of capturing still images and video.

SD Card Slot: Many ESP32-CAM modules come with an SD card slot for storing captured images and videos.

Programming and Configuration: The ESP32-CAM can be programmed using the Arduino IDE or other platforms compatible with the ESP32. It often requires specific libraries for camera functions.

Power Requirements: The power requirements may vary based on the camera usage, and it's important to ensure a stable power supply, especially when the camera is actively capturing images or recording video.

Applications: ESP32-CAM is commonly used in projects such as home security cameras, DIY surveillance systems, and other applications requiring image or video capture.

- PIR Motion Sensor

A Passive Infrared (PIR) motion sensor is a type of electronic sensor that detects infrared radiation emitted by objects in its field of view. PIR sensors are commonly used in various applications, including security systems, lighting control, and home automation, to detect the presence of moving objects or people.

Here are some key features and information about PIR motion sensors:

How PIR Motion Sensors Work:
Infrared Sensing: PIR sensors detect changes in infrared radiation within their detection range. Humans and animals emit infrared radiation, and when they move, the sensor detects a change in the amount of infrared radiation reaching its sensor elements.

Pyroelectric Sensor: PIR sensors typically use a pyroelectric sensor, which generates an electric charge in response to changes in temperature (due to infrared radiation). This change in charge is then processed to detect motion.

Key Features:
Detection Range: PIR sensors have a specific detection range and field of view. The range depends on the sensor's design and can vary from a few feet to several meters.

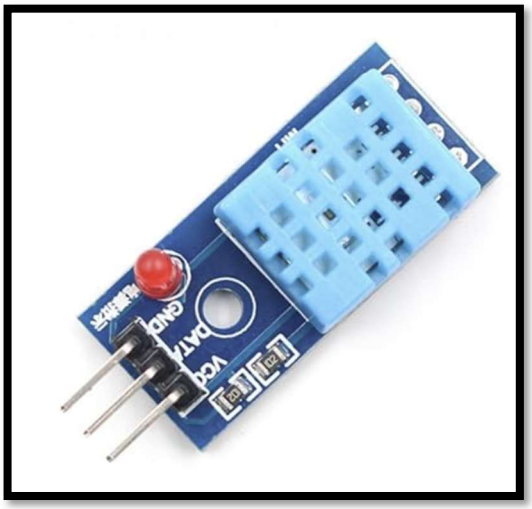
Sensitivity and Time Adjustment: Many PIR sensors allow users to adjust sensitivity and time

settings. Sensitivity adjustment determines the range of motion detection, and time adjustment controls how long the sensor will remain active after detecting motion.

Two-Element vs. Four-Element Sensors: PIR sensors come in two-element and four-element configurations. Four-element sensors are more advanced and can provide better performance by minimizing false positives.

Voltage and Current Ratings: PIR sensors typically operate on low voltage and low current, making them suitable for battery-powered devices.

- DHT11 Humidity/ Temperature Sensor



The DHT11 is a popular and low-cost sensor that is commonly used to measure temperature and humidity. It is widely used in various DIY electronics and IoT projects. Here are some key features and information about the DHT11 humidity/temperature sensor:

Key Features:
Combined Sensor: The DHT11 is a digital sensor that combines both temperature and humidity sensing capabilities in a single module.

Sensing Range:

Temperature: Typically, the DHT11 has a temperature sensing range from 0 to 50 degrees Celsius (32 to 122 degrees Fahrenheit).
Humidity: The humidity sensing range is generally from 20% to 80%.
Accuracy:

Temperature: The temperature accuracy is around ±2 degrees Celsius.
Humidity: The humidity accuracy is around ±5%.
Digital Output: The DHT11 provides a digital signal output that can be easily interfaced with microcontrollers without the need for external components.

Single-Wire Communication: It communicates over a single-wire serial interface, simplifying the connection to microcontrollers.

Low Power Consumption: The DHT11 is designed to be energy-efficient, making it suitable for battery-powered applications.

Calibration: The DHT11 is factory-calibrated, and calibration data is stored in its internal memory.

Wiring and Connection:
The DHT11 typically has four pins:

VCC (Power): Connects to the power supply (3.3V or 5V).
Data: Connects to a digital pin on the microcontroller for data communication.
Not Connected (NC): No connection.
Ground (GND): Connects to the ground.

- Bending Beam Weight Sensors

Bending beam weight sensors, also known as bending beam load cells, are devices designed to measure force or weight applied to them. They are commonly used in industrial and commercial applications for tasks such as weighing scales, industrial automation, and force measurement. Here are some key features and information about bending beam weight sensors:

Key Features:

Load Measurement Principle: Bending beam load cells operate on the principle of bending stress. When a force is applied to the load cell, it causes it to deform, and the resulting stress is measured to determine the applied force.

Material: Bending beam load cells are often made of materials such as alloy steel or aluminum, chosen for their strength and durability.

Capacity Range: These load cells come in various capacity ranges, from a few kilograms to several tons, catering to a wide range of applications.

High Accuracy: Bending beam load cells are known for their high accuracy in measuring weight or force, making them suitable for precise applications.

Compact Design: They typically have a compact and robust design, allowing for easy integration into various systems.

Multiple Load Cells: In some applications, multiple bending beam load cells may be used together to distribute the load evenly and increase the overall capacity.

Environmental Protection: Depending on the model, load cells may have various levels of protection against environmental factors, such as moisture and dust.

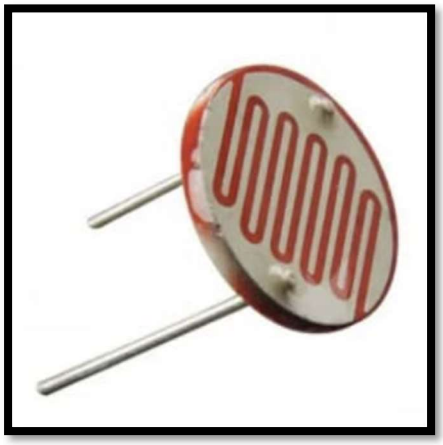
Wiring and Connection:

Bending beam load cells usually have multiple wires for connection, and the most common configuration includes four wires:

Excitation (+): Provides a voltage to the load cell for operation.
Excitation (-): Ground connection for the excitation voltage.
Signal (+): Carries the actual measurement signal.

Signal (-): Ground connection for the measurement signal.

- LDR Sensors



A Light Dependent Resistor (LDR), also known as a photoresistor, is a type of resistor whose resistance changes with the amount of light incident upon it. LDRs are commonly used in electronic circuits as light sensors. Here are some key features and information about LDR sensors:

Key Features:
Resistance Variation: The resistance of an LDR changes in response to the intensity of light. In the presence of light, the resistance decreases, and in darkness, the resistance increases.

Material: LDRs are typically made of semiconductor materials, such as cadmium sulfide (CdS) or cadmium selenide (CdSe).

Dark Resistance: Dark resistance refers to the resistance of the LDR when there is no light falling on it. It is usually measured in megaohms (MΩ) or kiloohms (kΩ).

Illumination Resistance: Illumination resistance refers to the resistance of the LDR when exposed to light. It is typically much lower than the dark resistance.

Spectral Response: LDRs are sensitive to a range of wavelengths in the electromagnetic spectrum, with peak sensitivity in the visible light range.

Wiring and Connection:

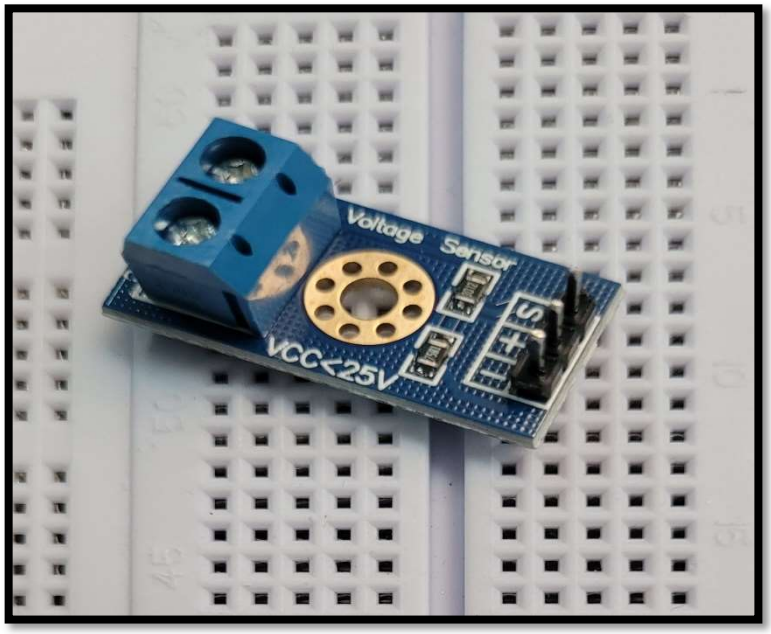
LDRs are simple to use and can be connected to a circuit in a voltage divider configuration. The basic setup includes:

Power Supply (Vcc): Connect one end of the LDR to the positive supply voltage.

Ground (GND): Connect the other end of the LDR to the ground.

Output (Analog or Digital Pin): Connect the junction between the LDR and a resistor to an analog or digital pin of a microcontroller or other measurement device.

- Power Meter

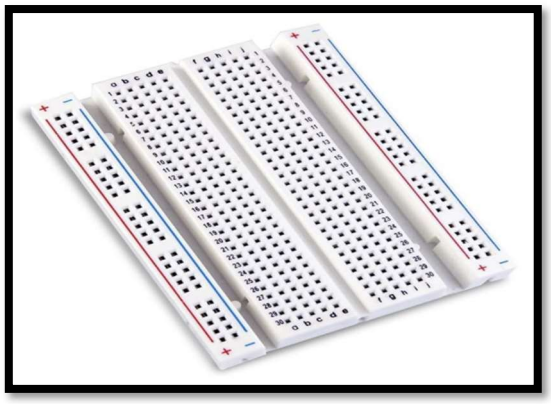


A power meter is a device used to measure the electrical power consumed by an electrical device or system. It typically measures parameters such as voltage, current, and power factor to calculate real power consumption. Power meters are commonly used in residential, commercial, and industrial settings for energy management and monitoring.

Key Features:

Measures voltage (V), current (I), power factor (PF), and real power (Wattage).
Available in different form factors, including handheld devices and panel-mounted meters.

- Breadboard



A breadboard is a crucial tool in electronics prototyping and experimentation. It provides a way to build and test electronic circuits without soldering components permanently. Here are the key features and information about breadboards:

Key Features:
Grid Layout: Breadboards typically have a grid of holes arranged in rows and columns. The columns are often labeled with numbers (usually 1-30) and the rows are lettered (usually A-J).

Connection Strips: The holes in each column are electrically connected internally, allowing components to be easily connected without soldering. The rows are not connected, providing a separation for different parts of the circuit.

Power Rails: Breadboards often have two long rows on the sides, known as power rails. These are typically used for providing power to the components on the breadboard. The top rail is usually connected to the positive power supply (Vcc), and the bottom rail is connected to the ground (GND).

Component Insertion: Components like resistors, capacitors, integrated circuits, and jumper wires can be inserted into the holes on the breadboard.

Jumper Wires: Jumper wires are used to make electrical connections between components on the breadboard. They are essential for building a circuit.

Reusable: The components can be easily inserted and removed, making the breadboard reusable for multiple projects.

Usage:
Component Placement: Insert electronic components into the breadboard by placing their leads or pins into the holes.

Connection: Use jumper wires to create connections between the components. The internal connections on the breadboard allow for easy and temporary circuit building.

Power Supply Connection: Connect the power supply or battery to the power rails to provide power to the circuit.

Prototyping: Breadboards are widely used for prototyping and testing circuit designs before soldering them onto a permanent circuit board.

Types of Breadboards:
Full-Size Breadboard: The standard size often used for general prototyping.

Half-Size Breadboard: A smaller version of the full-size breadboard, suitable for smaller projects.

Mini Breadboard: Compact and portable, ideal for small projects or when space is limited.

Considerations:
Connection Stability: While breadboards are excellent for prototyping, it's important to note that the connections may not be as stable as soldered connections. Vibrations or movement can disrupt

the circuit.

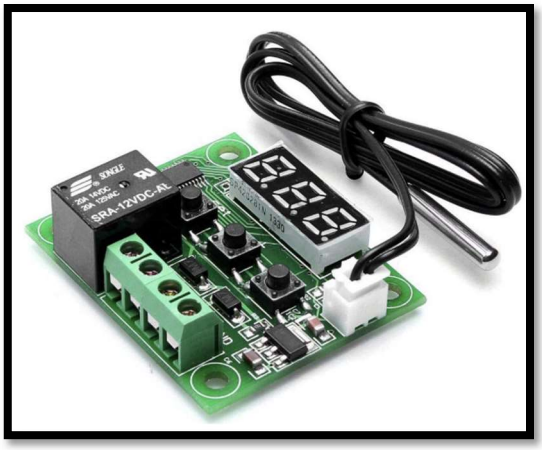
Jumper Wire Length: Keep jumper wires neat and at an appropriate length to avoid clutter on the breadboard.

Component Leg Bending: When inserting components, be mindful of bending the legs excessively, as this may damage the component.

Power Supply Ratings: Ensure that the power supply voltage and current ratings are suitable for the components on the breadboard.

Breadboards are fundamental tools for anyone learning or working in electronics, providing a convenient and flexible platform for quickly testing and iterating on circuit designs.

- Thermostat



A thermostat is a device that is used to regulate temperature, maintaining it at a set point by controlling the heating or cooling system in a space. Thermostats are commonly used in HVAC (Heating, Ventilation, and Air Conditioning) systems, home appliances, and various industrial applications. Here are some key features and information about thermostats:

Key Features:

Temperature Control: The primary function of a thermostat is to control the temperature of a space by activating or deactivating a heating or cooling system.

Set Point: Users can set a desired temperature level (set point) on the thermostat. The thermostat works to maintain the ambient temperature close to this set point.

Sensors: Thermostats are equipped with temperature sensors that measure the current temperature of the space. Some thermostats may also have additional sensors for humidity control.

User Interface: Modern thermostats often come with digital displays and user-friendly interfaces, allowing users to easily adjust settings and view the current temperature.

Modes of Operation:

Heating Mode: Activates the heating system when the temperature falls below the set point.

Cooling Mode: Activates the cooling system when the temperature rises above the set point.

Programmable Thermostats: Some thermostats are programmable, allowing users to schedule temperature changes at different times of the day. This helps optimize energy efficiency.

Smart Thermostats: Smart thermostats can be connected to the internet and controlled remotely through mobile apps. They often include additional features like learning the user's preferences and integrating with home automation systems.

Types of Thermostats:

Mechanical Thermostats: Traditional thermostats with simple mechanical components and a bimetallic strip that bends with temperature changes.

Digital Thermostats: Use digital technology for more accurate temperature control and may have programmable features.

Programmable Thermostats: Allow users to program different temperature settings for various times of the day or week, optimizing energy consumption.

Smart Thermostats: Connected to the internet, these thermostats can be controlled remotely, learn

user habits, and often integrate with smart home ecosystems.

Applications:
Residential HVAC Systems: Thermostats are commonly used in homes to control heating and cooling systems.

Commercial Buildings: Used in larger buildings to regulate temperature and energy consumption.

Industrial Processes: Thermostats are used in various industrial applications to control temperature in manufacturing processes.

Refrigeration: Thermostats are used in refrigerators and freezers to maintain the desired temperature for preserving food.

Considerations:
Compatibility: When replacing a thermostat, it's essential to ensure compatibility with the heating or cooling system in use.

Location: The thermostat's location within a space can affect its accuracy. It should be placed in a representative area and away from direct sunlight, drafts, or other factors that may skew temperature readings.

Energy Efficiency: Programmable and smart thermostats contribute to energy efficiency by allowing users to schedule temperature changes based on occupancy patterns.

Maintenance: Regular maintenance, such as changing batteries in battery-powered thermostats, ensures proper functionality.

When choosing a thermostat, consider the specific needs of the space and the desired features, such as programmability or smart capabilities. Installation and usage instructions provided by the thermostat manufacturer should be followed for optimal performance.

- Transistor

A transistor is a semiconductor device that can be used for signal amplification, switching, and other electronic functions. Transistors are fundamental building blocks in electronic circuits and play a crucial role in the field of electronics. There are two main types of transistors: bipolar junction transistors (BJTs) and field-effect transistors (FETs). Here's an overview of transistors:

Bipolar Junction Transistor (BJT):
Types of BJTs:

NPN (Negative-Positive-Negative): The majority charge carriers are electrons.
PNP (Positive-Negative-Positive): The majority charge carriers are holes.
Structure:

A BJT consists of three layers of semiconductor material: the emitter, base, and collector.
In an NPN transistor, for example, the emitter is N-type, the base is P-type, and the collector is N-type.
Operation:

The transistor operates based on the movement of charge carriers (electrons or holes) from the emitter to the collector, controlled by the voltage applied to the base.
NPN transistors are typically used for amplification, and PNP transistors are often used in complementary configurations.
Field-Effect Transistor (FET):
Types of FETs:

MOSFET (Metal-Oxide-Semiconductor FET): Utilizes a metal gate insulated from the semiconductor by a thin oxide layer.

JFET (Junction Field-Effect Transistor): Utilizes a reverse-biased pn-junction to control the flow of current.

Structure:

A MOSFET consists of a metal gate, an insulating oxide layer, and a semiconductor (usually silicon).

A JFET consists of a channel of semiconductor material.

Operation:

In a MOSFET, the flow of current between the source and drain is controlled by the voltage applied to the gate.

In a JFET, the flow of current between the source and drain is controlled by the voltage applied to the gate, which creates or depletes a channel in the semiconductor.

Common Uses of Transistors:

Amplification: Transistors are widely used for signal amplification in audio amplifiers, radio-frequency amplifiers, and other electronic circuits.

Switching: Transistors act as electronic switches in digital circuits, enabling binary logic operations.

Signal Modulation: Transistors are used in signal modulation circuits, such as frequency modulation (FM) and amplitude modulation (AM) circuits.

Oscillation: Transistors can be used in oscillator circuits to generate periodic waveforms, such as in radio frequency (RF) oscillators.

Voltage Regulation: Transistors are used in voltage regulator circuits to stabilize and regulate voltage levels in power supplies.

Considerations:

Biasing: Proper biasing is crucial to ensure the transistor operates in its desired region (active, saturation, or cutoff).

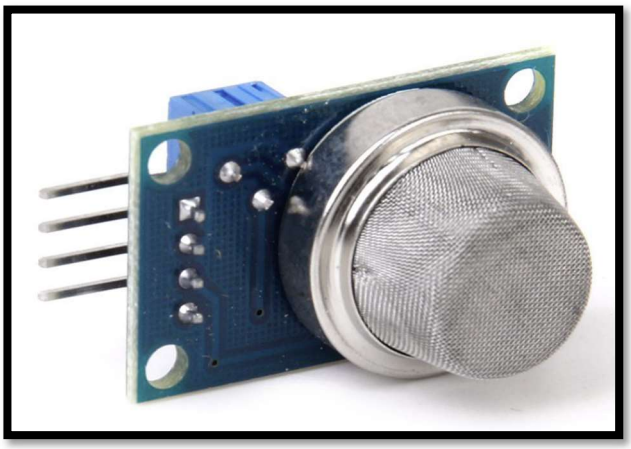
Maximum Ratings: Adhering to the maximum ratings specified by the transistor's datasheet is essential to prevent damage.

Temperature Dependence: Transistor characteristics can be temperature-dependent, and thermal considerations are important, especially in high-power applications.

Transistor Configurations: Different transistor configurations (common emitter, common collector, common base for BJTs; common source, common drain, common gate for FETs) offer different characteristics and are chosen based on the specific application.

Understanding the characteristics and configurations of transistors is fundamental for anyone working with electronic circuits. The specific type of transistor chosen for a particular application depends on factors such as voltage requirements, current, and desired performance characteristics. Always refer to the datasheet provided by the manufacturer for detailed specifications and usage guidelines.

- Smoke Sensors



A smoke sensor, often referred to as a smoke detector, is a device designed to detect the presence of smoke in the air, indicating a potential fire. Smoke sensors are essential components in fire detection and alarm systems, providing early warning in the event of a fire. Here are key features

and information about smoke sensors:

Types of Smoke Sensors:

Photoelectric Smoke Sensor:

Operation: Uses a light source and a photosensitive sensor. When smoke particles enter the chamber, they scatter the light, leading to a reduction in the light reaching the sensor.

Applications: Effective at detecting smoldering fires and are less prone to false alarms from cooking or steam.

Ionization Smoke Sensor:

Operation: Contains a small amount of radioactive material that ionizes the air between two electrically charged plates. When smoke enters the chamber, it disrupts the ionization process, causing a drop in current flow and triggering the alarm.

Applications: Effective at detecting fast-burning, flaming fires. More prone to false alarms from cooking or steam.

Dual Sensor (Photoelectric and Ionization):

Operation: Combines both photoelectric and ionization technologies for a more comprehensive smoke detection capability.

Applications: Offers a balance between detecting both smoldering and fast-burning fires while minimizing false alarms.

Air Sampling Smoke Detector:

Operation: Uses a network of pipes to actively draw air samples into a sensing chamber. It can detect very low concentrations of smoke particles.

Applications: Common in critical environments where early detection is crucial, such as data centers or museums.

Key Features:

Alarm Activation: Smoke sensors are designed to activate an audible and/or visual alarm when smoke is detected, providing an early warning of a potential fire.

Power Source: Smoke sensors are typically powered by batteries or are hardwired into the electrical system of a building.

Interconnection: In multiple sensor systems, detectors are often interconnected so that if one sensor detects smoke, all interconnected sensors will activate their alarms.

Tamper Resistance: Some smoke sensors include tamper-resistant features to prevent unauthorized interference or disabling.

Maintenance Alerts: Some advanced smoke detectors provide alerts or signals when the device requires maintenance, such as battery replacement or sensor cleaning.

Placement and Installation:
Ceiling Mounting: Smoke sensors are typically installed on the ceiling since smoke tends to rise. The ideal placement is in the center of the ceiling.

Avoid Obstructions: Install smoke sensors away from air vents, windows, and other obstructions that may affect their performance.

Multiple Sensors: Install smoke sensors in various locations throughout a building, especially in sleeping areas and near potential fire hazards like kitchens.

Considerations:
Regular Testing: It's essential to regularly test smoke sensors according to the manufacturer's recommendations to ensure proper functionality.

Battery Replacement: For battery-powered smoke detectors, replace batteries at least once a year or as recommended by the manufacturer.

Cleaning: Keep the sensor surfaces clean from dust and other contaminants, as this can affect their sensitivity.

Local Regulations: Adhere to local building codes and regulations regarding the installation and maintenance of smoke detectors.

Smoke sensors are a critical component of fire safety systems, providing early detection that can save lives and minimize property damage in the event of a fire. Always follow the manufacturer's guidelines for installation, testing, and maintenance.

1.4. Software Specification

- Visual Studio

Visual Studio is an integrated development environment (IDE) developed by Microsoft. It is widely used by software developers to create a variety of applications, including desktop, web,

mobile, and cloud-based applications. Visual Studio supports multiple programming languages and offers a rich set of tools for code editing, debugging, testing, and deployment. Here are key features and information about Visual Studio:

Features:
Multi-Language Support:
Visual Studio supports multiple programming languages, including C#, C++, Visual Basic, F#,

Python, JavaScript, and more.
Code Editor:

The code editor provides features like syntax highlighting, IntelliSense (code completion), and code navigation to enhance developer productivity.
Integrated Debugger:

Visual Studio includes a powerful debugger with features like breakpoints, watch windows, and step-through debugging to help identify and fix issues in code.
Visual Designers:

Visual Studio includes visual designers for building user interfaces, databases, and more. These designers allow developers to design applications using a visual interface.

Integrated Development for Web and Cloud:

Support for web development, including ASP.NET for building web applications, and cloud development with integration for Azure services.

Extensions and Add-ons:

Visual Studio can be extended using a wide range of extensions and add-ons available from the Visual Studio Marketplace. These extensions add functionality and support for additional languages and platforms.

Version Control Integration:

Visual Studio integrates with version control systems like Git, providing tools for source code management and collaboration.

Testing Tools:

Integrated testing tools for unit testing, code coverage analysis, and performance profiling.

NuGet Package Manager:

NuGet integration allows developers to manage and consume third-party libraries and packages in their projects.

Team Collaboration:

Visual Studio provides features for team collaboration, including Team Foundation Server (TFS) or Azure DevOps integration for source control, build automation, and project management.

Cross-Platform Development:

Visual Studio supports cross-platform development for platforms like Windows, macOS, and Linux, making it versatile for various application scenarios.

Editions:

Visual Studio Community:

Free version for individual developers, open-source projects, and small teams.
Visual Studio Professional:

Paid version with additional features and capabilities for professional developers and small teams.
Visual Studio Enterprise:

Comprehensive IDE with advanced features, performance profiling, and collaboration tools for larger development teams and enterprises.
Supported Platforms:
Visual Studio primarily runs on the Windows operating system, but developers can use Visual Studio Code, a lightweight code editor, for cross-platform development on Windows, macOS, and Linux.
Integrated Services:
Azure DevOps: Visual Studio integrates with Azure DevOps for application lifecycle management, including version control, build automation, release management, and project tracking.
Visual Studio is a versatile IDE that caters to the needs of developers across a broad spectrum of applications and platforms. It provides a comprehensive set of tools and services to streamline the development process. Developers can choose the edition that best fits their needs based on the scale and requirements of their projects.

- **Arduino IDE**

The project aims to enhance convenience, efficiency and control through integration of technologies. This includes automated and controlled lighting, temperature, security and more using different sensors. The main highlight of the project is the use of OpenCV for object detection to implement smart refrigerators, automated cleaning systems and face recognition systems for enhanced security and privacy. The project further includes the scope of energy management as per homeowner's requirements
The Arduino Integrated Development Environment (Arduino IDE) is an open-source software application used to program and upload code to Arduino and Arduino-compatible microcontroller

boards. Arduino is a popular platform for prototyping and developing electronics projects, especially for those with little or no experience in embedded programming. Here are key features and information about the Arduino IDE:

Features:

Code Editor:

The Arduino IDE provides a simple yet effective code editor with features such as syntax highlighting, auto-indentation, and code completion.

Built-in Examples:

The IDE comes with a variety of built-in examples that serve as starting points for learning and experimenting with different Arduino functionalities.

Library Manager:

Arduino IDE includes a Library Manager for easily adding, managing, and using third-party libraries that provide additional functionalities.

Serial Monitor:

The Serial Monitor tool allows developers to communicate with the Arduino board and receive real-time data from sensors or debug messages from the Arduino program.

Upload to Arduino Board:

The IDE provides a straightforward process for compiling code and uploading it to the Arduino board, making it easy to test and iterate on projects.

Integrated Tools:

Tools for verifying code (compiling), uploading code to the Arduino board, and opening a serial monitor are integrated into the IDE.

Cross-Platform:

The Arduino IDE is compatible with Windows, macOS, and Linux, providing a consistent

development environment across different operating systems.

Board Manager:

Arduino supports a wide range of microcontroller boards. The Board Manager feature allows users to install and manage board support packages for various Arduino-compatible boards.

Usage:

Installation:

Download and install the Arduino IDE from the official Arduino website.

Writing Code:

Write or modify Arduino sketches (programs) using the code editor.

Compiling:

Use the "Verify" button to compile the code and check for errors.

Uploading:

Connect the Arduino board to the computer, select the correct board and port in the IDE, and use the "Upload" button to upload the compiled code to the board.

Serial Monitor:

Open the Serial Monitor to interact with the Arduino board, view sensor data, or debug the program.

Community and Ecosystem:

Community Support:

The Arduino community is active and provides support through forums, tutorials, and online resources.

Extensibility:

The Arduino IDE is extensible, allowing the addition of third-party boards, libraries, and tools to enhance functionality.

Arduino Create:

Arduino offers a cloud-based platform called Arduino Create, which provides an online version of the IDE with additional features and collaboration tools.

Considerations:

Updates and Versions:

It's advisable to keep the Arduino IDE up to date to access the latest features, improvements, and bug fixes.

Board Selection:

Ensure the correct Arduino board and port are selected in the IDE before uploading code.

Documentation:

Arduino provides extensive documentation and tutorials on its official website, making it a valuable resource for beginners and experienced users alike.

The Arduino IDE is designed to be user-friendly and accessible, making it a popular choice for hobbyists, students, and professionals alike who are entering the world of embedded systems and microcontroller programming.

- **IFTTT**

IFTTT stands for "If This Then That." It is a web-based service that allows you to create applets (formerly called recipes) that automate various tasks by connecting different web services, apps, and devices. The basic idea behind IFTTT is to trigger an action (That) when a specific event occurs (If).

Here's how it works:

Applets: An applet is a simple script that consists of a trigger (If) and an action (Then). For example, if you post a photo on Instagram (trigger), you can set up an applet to automatically save that photo to your Dropbox account (action).

Triggers and Actions: Triggers are the "If" part of the applet, and they represent events or conditions that initiate the automation. Actions are the "Then" part, specifying what should happen when the trigger event occurs.

Channels: IFTTT supports a wide range of channels, which are essentially the different services, apps, and devices that you can connect. Examples of channels include social media platforms, smart home devices, productivity tools, and more.

Recipes (Applets): Users can create their own applets or use pre-made ones from the IFTTT community. These applets are shared and can be customized to suit individual needs.

Examples: Some examples of applets include:

If you receive an email with an attachment, save the attachment to Google Drive.
If you update your profile picture on Facebook, update your Twitter profile picture as well.
If you enter a specific location, send a text message to a designated contact.
IFTTT makes it easy for users to automate tasks without the need for complex programming. It's a versatile tool that can streamline various aspects of your digital life by connecting different online services and devices. Keep in mind that the name has been updated from "recipes" to "applets" to better reflect its capabilities.

- **ThingSpeak**

ThingSpeak is an Internet of Things (IoT) platform that allows users to collect, analyze, and visualize data from their IoT devices. It provides a cloud-based infrastructure for storing and managing data generated by sensors or other devices in real-time. ThingSpeak is often used for monitoring and controlling IoT devices, as well as for building applications that involve data analysis and visualization.

Key features of ThingSpeak include:

Channels: In ThingSpeak, data is organized into channels. Each channel can store and manage data from a specific source, such as a sensor. Users can create multiple channels to organize and categorize their data.

Fields: Within each channel, there are fields where the actual data is stored. For example, if you have a temperature sensor, you might have fields for temperature, humidity, and other relevant data.

Write and Read APIs: ThingSpeak provides APIs (Application Programming Interfaces) that allow devices to write data to channels and read data from channels. This makes it easy to integrate IoT devices with the ThingSpeak platform.

Charts and Visualizations: ThingSpeak offers built-in tools for creating charts and visualizations based on the data stored in channels. Users can easily configure and customize these visualizations to gain insights into their IoT data.

React: ThingSpeak supports React, which allows users to define actions or alerts based on certain conditions in the data. For example, you could set up an alert to be triggered if the temperature in a channel exceeds a certain threshold.

MATLAB Integration: ThingSpeak has integration with MATLAB, a popular numerical computing environment. This integration allows users to perform more advanced analysis on their IoT data using MATLAB scripts.

Open Source: ThingSpeak is open-source, which means that users can modify and extend the platform to suit their specific needs. This openness has contributed to its popularity in the IoT community.

Overall, ThingSpeak is a flexible and user-friendly platform that enables individuals and organizations to build IoT applications and solutions without the need for extensive development effort. It is particularly suitable for projects that involve data logging, monitoring, and basic analysis of sensor data.

- **Home Assistant**

Home Assistant allows users to control and automate various devices and services within their home, creating a smart home environment. Here are key features and information about Home Assistant:

Key Features:
Open Source:

Home Assistant is open-source software, allowing users to view, modify, and contribute to its source code. It is actively developed by a community of contributors.
Device Integration:

Home Assistant supports a wide range of devices and platforms, including smart lights, thermostats, sensors, cameras, and more. It integrates with popular smart home protocols like Zigbee, Z-Wave, MQTT, and others.
Automation:

Users can create automation rules to trigger actions based on events, conditions, and time schedules. Automation can be used to control devices, send notifications, and perform other tasks.
User Interface:

Home Assistant provides a web-based user interface that allows users to monitor and control their smart home devices. The interface is customizable, and users can create dashboards tailored to their preferences.
Voice Control:

Integration with voice assistants like Amazon Alexa and Google Assistant allows users to control devices using voice commands.
Mobile App:

Home Assistant has a mobile app for iOS and Android, providing remote access and control of smart home devices from smartphones and tablets.

Integrations and Add-ons:

Home Assistant supports a wide variety of integrations and add-ons, extending its functionality. These integrations cover popular smart home devices, cloud services, and more.

Security and Privacy:

Home Assistant puts a strong emphasis on security and privacy. Users have control over their data, and the system can be run locally without the need for cloud services.

Community Support:

The Home Assistant community is active and provides support through forums, documentation, and social media channels. Users can share their experiences, seek advice, and contribute to the development.

Usage:

Installation:

Home Assistant can be installed on various platforms, including Raspberry Pi, Linux servers, and Docker containers. The installation process is well-documented on the official website.

Configuration:

Users configure Home Assistant through YAML files, defining devices, automations, and other settings. The configuration can be done through the web interface or by editing configuration files directly.

Integration:

Devices are integrated by configuring the appropriate components or platforms within Home Assistant. This may involve setting up communication protocols or using add-ons for specific devices.

Automation Setup:

Automation rules are defined to create scenarios where devices interact based on specific conditions or events. Users can create complex automation sequences using the built-in automation editor.

Considerations:

Device Compatibility:

Check the compatibility of your smart home devices with Home Assistant to ensure seamless integration.

Community and Documentation:

Explore the Home Assistant community forums and documentation for support and guidance on specific setups and configurations.

Updates:

Regularly update Home Assistant to benefit from new features, improvements, and security patches.

Backup:

Periodically backup your Home Assistant configuration to safeguard against data loss or system failures.

Home Assistant is a powerful and flexible home automation platform that caters to users looking for a customizable and self-hosted solution for managing their smart home devices. Whether you're a beginner or an advanced user, Home Assistant offers a comprehensive set of tools for creating a personalized and intelligent home environment.

- **Adafruit IO**

Adafruit IO is a cloud service provided by Adafruit Industries that enables the easy integration of Internet of Things (IoT) devices and projects. It offers a platform for collecting, visualizing, and storing data from connected devices. Adafruit IO simplifies the process of building IoT applications by providing an infrastructure for communication between devices and the cloud. Here are key features and information about Adafruit IO:

Features:

Data Feeds:

Adafruit IO allows users to create data feeds, which act as channels for sending and receiving data between devices and the cloud. Each data feed represents a stream of data.

Dashboard:

Users can create dashboards to visualize and monitor the data from their devices in real-time. Dashboards can include charts, gauges, and other visual elements.

Integration with Adafruit Hardware:

Adafruit IO is designed to seamlessly integrate with Adafruit's hardware, making it easy to connect Adafruit devices, sensors, and components to the cloud.

RESTful API:

Adafruit IO provides a RESTful API, allowing developers to interact with the platform programmatically. This enables the creation and management of feeds, as well as the retrieval and posting of data.

Integration with IoT Platforms:

Adafruit IO can be integrated with popular IoT platforms, such as Arduino IoT Cloud and others, providing flexibility and compatibility with a variety of development environments.

User Authentication:

Users can create accounts on Adafruit IO, and the platform supports user authentication to ensure secure access to data feeds and dashboards.

MQTT Support:

Adafruit IO supports the MQTT (Message Queuing Telemetry Transport) protocol, a lightweight and efficient messaging protocol commonly used in IoT applications.

Usage:

Account Setup:

Users need to create an account on the Adafruit IO platform to start using its services.

Creating Data Feeds:

Data feeds are created to represent the different types of data that devices will send or receive. For example, a temperature sensor might have a corresponding temperature data feed.

Connecting Devices:

IoT devices are connected to Adafruit IO by configuring them to publish data to specific data feeds. This involves setting up the devices to communicate with the Adafruit IO API.

Building Dashboards:

Users can create dashboards to visualize and monitor data in real-time. Dashboards can be customized with various widgets to display data in different formats.

Programming and Interacting with the API:

Developers can use the Adafruit IO API to programmatically interact with the platform. This includes retrieving data, posting data, managing feeds, and more.

Considerations:

Security:

Ensure that proper security measures are in place, especially when dealing with sensitive data. Use secure authentication practices and consider using secure communication protocols like HTTPS.

Data Retention:

Understand the data retention policies of Adafruit IO, especially if there are specific requirements for storing historical data.

Platform Compatibility:

Verify that Adafruit IO is compatible with the hardware and development platforms you intend to

use in your IoT project.

Community Support:

Adafruit IO has an active community, and users can seek support, ask questions, and share experiences through forums and other community channels.

Adafruit IO is a valuable resource for IoT enthusiasts and developers looking to build connected projects without the need to set up and manage their own cloud infrastructure. It simplifies the process of collecting and visualizing data from IoT devices, making it accessible to a broad range of users.

2. LITERATURE SURVEY

2.1 Existing System

In the world of technology, new developments can be so fast that predicting what may happen in the future can be redundant as it appears as soon as it is predicted. Although the future of smart home is definitely bright but we need to take a holistic view of these developments, understanding negative effects that such growth creates.

The Home automated systems that are currently in use has certain demerits such as extremely high cost, security issues, neither the concept of Smart Energy Management nor the presence of Smart Refrigerators that can manage its inventory itself. Further the existing system has no automated smart cleaning system.

Disadvantages of Home Automation:

Cost: One of the main drawbacks of home automation is the cost. Smart devices and systems can cost a fortune but with our proposed system we can reduce the cost significantly by using hardware which is inexpensive but still effective.

Privacy Concerns: With so many connected devices in your home, there is also the potential for privacy concerns. Hackers could potentially gain access to your smart devices and collect sensitive information or even control them remotely. But with our system that will be improved because of the used of different protocols for data integrity.

Learning Curve: Finally, home automation can also have a learning curve. Setting up and configuring your smart devices and systems can be complex, and it may take some time to learn how to use them effectively.

High Maintenance: The existing system has complex hardware the requires regular maintenance due to which the whole system isn't much user friendly and convenient to use.

*Lack of Personalization***:** Non-automated homes don't have the capability to personalize settings based on individual preferences and occupancy patterns, leading to less comfortable living conditions.

2.2 Proposed System

The scope of an intelligent home system, often referred to as a smart home system, is quite broad and encompasses various aspects of residential living that can be enhanced through technology and automation.

The primary goal of such a system is to create a more convenient, efficient, secure, an comfortable living environment for the occupants. Here are some key areas within the scope of an intelligent home system:

- ✓ **Home Automation:** The central feature of a smart home system is automation. This involves controlling and managing various devices and systems in the home, such as lighting, heating, cooling, appliances, and more, through centralized control systems.

  I. *Inventory Management***:** The smart refrigerator will detect the availability of items in the fridge using OpenCV and if in case, any of the necessary item is going to get consumed sooner, the user will be notified about the same and the fridge after comparing databases of certain online retail shops will automatically order the item needed at the cheapest price

  II. *Automated Cleaning System*: OpenCV is to be used for the detection of dirt and insects in the room which will be smartly cleaned.

  III. *Path finder*: A pathway that can help outsiders/guests to find their respective room or any other room in the house using LEDs and triggers generated via IFTTT services.

  IV. *Convertible Kitchen* : A kitchen that can adapt to open kitchen and closed kitchen design as per user instructions and convenience.

  V. *Smart Humidifier/ Temperature control system*: An air conditioner or air cooler that can regulate its fan's speed as per humidity or temperature of the external environment. This concept can further be extended to exhausts installed in the kitchen/washrooms in the house.

  VI. *Smart Garage*: Garage that can automatically detect the presence of any vehicle approaching towards it and regulate it's closing and opening mechanism

- ✓ **Energy Management**: Smart home systems can help optimize energy usage by intelligently regulating heating, cooling, and lighting based on occupancy and external conditions. Everything in the house will be working according to the energy consumption limit provided

  by the utilizer to regulate bills and prevent higher energy consumption. This can lead to energy savings and a reduced carbon footprint.

- ✓ **Appliance Control:** Smart appliances can be remotely controlled and monitored, allowing users to manage their usage more effectively.

- ✓ **Security and Surveillance:** Smart home systems often include security features such as motion sensors and door/window sensors. These components can be monitored remotely and integrated with notifications to enhance home security.

- ✓ **Environmental Controls:** Smart thermostats and climate control systems allow homeowners to remotely adjust the temperature and climate settings, ensuring comfort while minimizing energy waste.

- ✓ **Lighting Control:** Intelligent lighting systems can adjust brightness, color, and ambiance according to user preferences or preset schedules. Motion sensors can also automate lighting based on occupancy

- ✓ **Health and Wellness:** Some smart home systems include features that monitor air quality, humidity, and other environmental factors that can impact health and well-being.

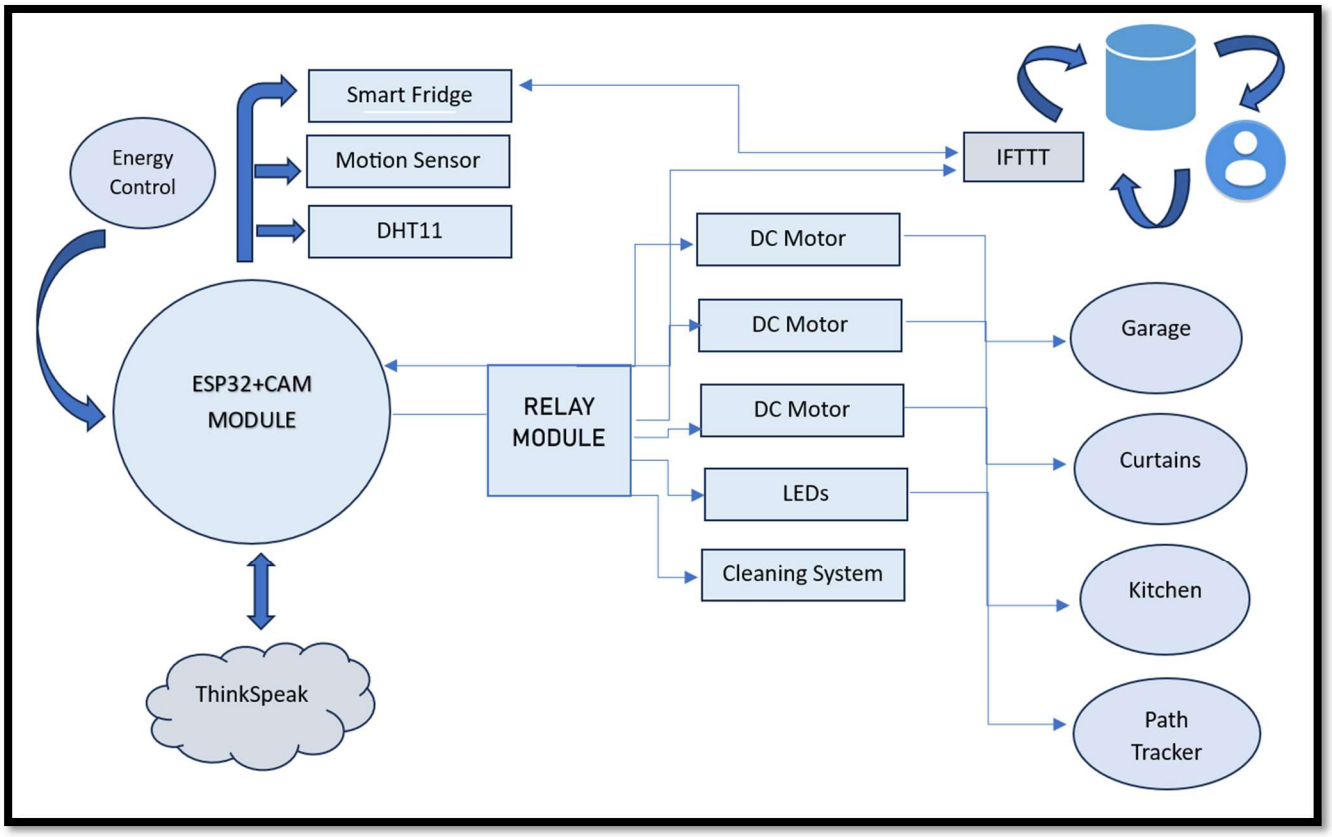
- ✓ **Integration and Interoperability:** One of the challenges in smart home systems is integrating devices from different manufacturers and ensuring they work together seamlessly.

- ✓ **Remote Monitoring and Control**: With the help of mobile apps, homeowners can monitor and control various aspects of their home remotely, providing convenience and peace of

mind, especially when traveling.

- ✓ **Data Privacy and Security:** As smart homes gather and transmit sensitive data, it's important to consider security measures to protect this information from unauthorized access.

2.3 Design flow/Process



2.4 PROBLEM FORMULATION

The problem at hand is to design and implement an efficient and user-friendly home automation system that enhances the convenience, energy efficiency, security, and overall living experience of homeowners.

Challenges:

- **Interoperability:** Ensuring that devices from various manufacturers can communicate and work together within the system can be a challenge due to different communication protocols

and standards.

- **Data Privacy and Security**: Protecting user data and ensuring that the system is not vulnerable to hacking or unauthorized access is critical.
- **Usability:** Striking the right balance between advanced features and simplicity is essential to cater to a wide range of users.
- **Integration Complexity:** Integrating different types of devices, each with its own intricacies, can lead to technical complexities during development and maintenance.
- **Energy efficient**: Ensure that devices that are integrated in home automated system have to use low power and manage the consumption of energy efficiently.

In the absence of automation, controlling household devices remotely isn't possible. This lack of remote control can cause inconvenience, especially when homeowners are away. Non-automated homes don't have the capability to personalize settings based on individual preferences and occupancy patterns, leading to less comfortable living conditions. Without automated diagnostics, identifying issues with appliances or systems becomes challenging, potentially leading to delayed maintenance or unexpected breakdowns. Manual resource management can lead to wastage, such as over-irrigating plants, overusing water, or not optimizing energy consumption.

Houses lacking automation might pose challenges for elderly or disabled individuals who struggle with physical tasks, such as opening doors or adjusting lighting. Without automation, homeowners miss out on valuable data regarding energy consumption, occupancy patterns, and usage trends, which could inform smarter decisions.

2.5. RESEARCH OBJECTIVES

The primary goal of such a system is to comprehensively investigate the design, implementation, and impact of a home automation system with a focus on enhancing energy efficiency, user experience and security. This research aims to contribute to the advancement of smart home technologies by addressing key challenges in interoperability, user interface design, and data privacy. By conducting a detailed analysis of various home automation components and their integration, this

study seeks to propose novel solutions for optimizing energy consumption through intelligent device control and automation.

- **Optimization of Energy Efficiency**: Investigate how home automation systems can effectively optimize energy usage by intelligently controlling devices based on occupancy patterns, user behavior, and environmental conditions.
- **Fault Tolerance and Reliability**: Techniques to enhance the reliability and fault tolerance of home automation systems, ensuring continued functionality in the face of device failures or network disruptions.
- **Sustainable Living and Environmental Impact**: Assess the role of home automation systems in promoting sustainable living practices and quantify their impact on reducing energy consumption, carbon emissions, and overall environmental footprint.
- **Interoperability and Standardization**: Explore methods to enhance interoperability among devices from different manufacturers and propose strategies for standardizing communication protocols in home automation systems.
- **Integration of Emerging Technologies**: Explore the integration of emerging technologies such as Internet of Things (IoT), artificial intelligence (AI) within home automation systems, assessing their impact and potential benefits.
- **Security and Privacy Enhancement**: Research security vulnerabilities in home automation systems and develop novel approaches to ensure data privacy, secure communication, and protection against potential cyber threats.

3. DESIGN FLOW/ PROCESS

3.1 Methodology

I. **Planning and Design**: Designing and planning a system that integrates the different hardware to create a proper ecosystem of automated households.

II. **Integration and Development**: Integrate smart devices, sensors, and systems according to the design plan, and develop the central control platform and mobile application.

III. **Smart Inventory Management Implementation:** The ESP32 and CAM Module will be used with OpenCV for detection of objects. Then the microcontroller will be used to compare the prices online and send the notification to the user to order according to the best price.

IV. **Automated Cleaning system Implementation:** To implement a system using OpenCV that will automatically detect the dirt in the house and deploy the required device for cleaning.

V. **Smart Path Finder Implementation:** To implement path finder, LED lights and triggers generated by user will be integrated via IFTTT services. ESP32 Microcontroller will be used to control the LEDs

VI. **Convertible Kitchen:** In order to implement convertible kitchen, DC motors will be deployed along with the microcontroller that can slide the wall of the kitchen as per instructions provided by the user.

VII. **Smart Humidifier Implementation**: This system can be implemented by deploying DHT11 humidity/Temperature Sensors to detect the Outdoor temperature and humidity levels and set the Indoor temperature relatively. For example, For humidity level of 35 % outside, a temperature of +20 Degree Fahrenheit will be maintained indoor.

VIII. **Energy Management Implementation:** Design and implement a power management system that will according to user need will monitor and manage the energy of the household.

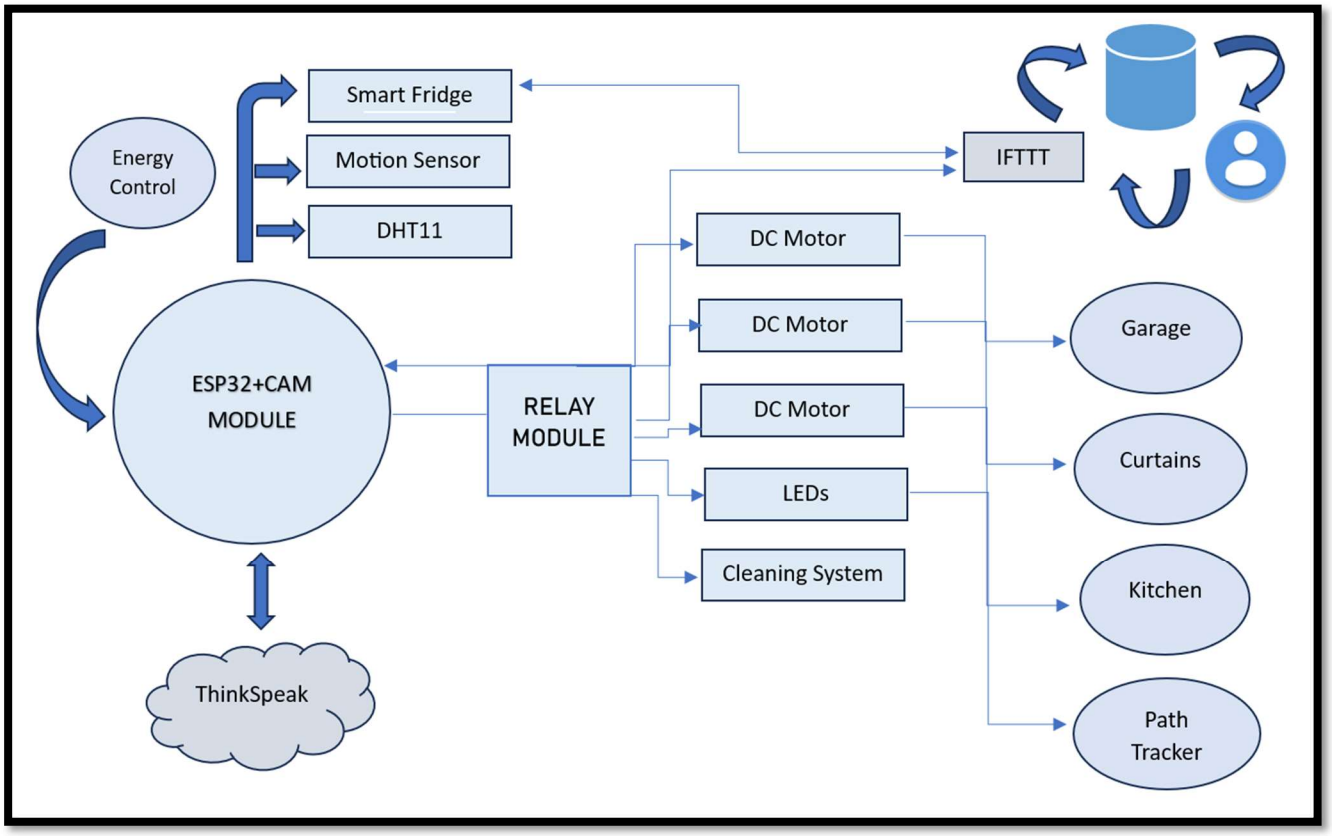
IX. **Security and Safety Setup:** Install security devices such as cameras, motion sensors, and smart locks, and develop the security management system.

X. **User Interface Development:** Design and develop an intuitive user interface for the mobile application, enabling residents to control and monitor the automated systems.

XI. **Data Collection and Insights:** Set up data collection mechanisms and implement data analytics to provide residents with insights into their energy usage and occupancy patterns.

XII. **Deployment and Launch:** Deploy the fully functional automated infrastructure within the household and launch the project to enhance the resident's living experience

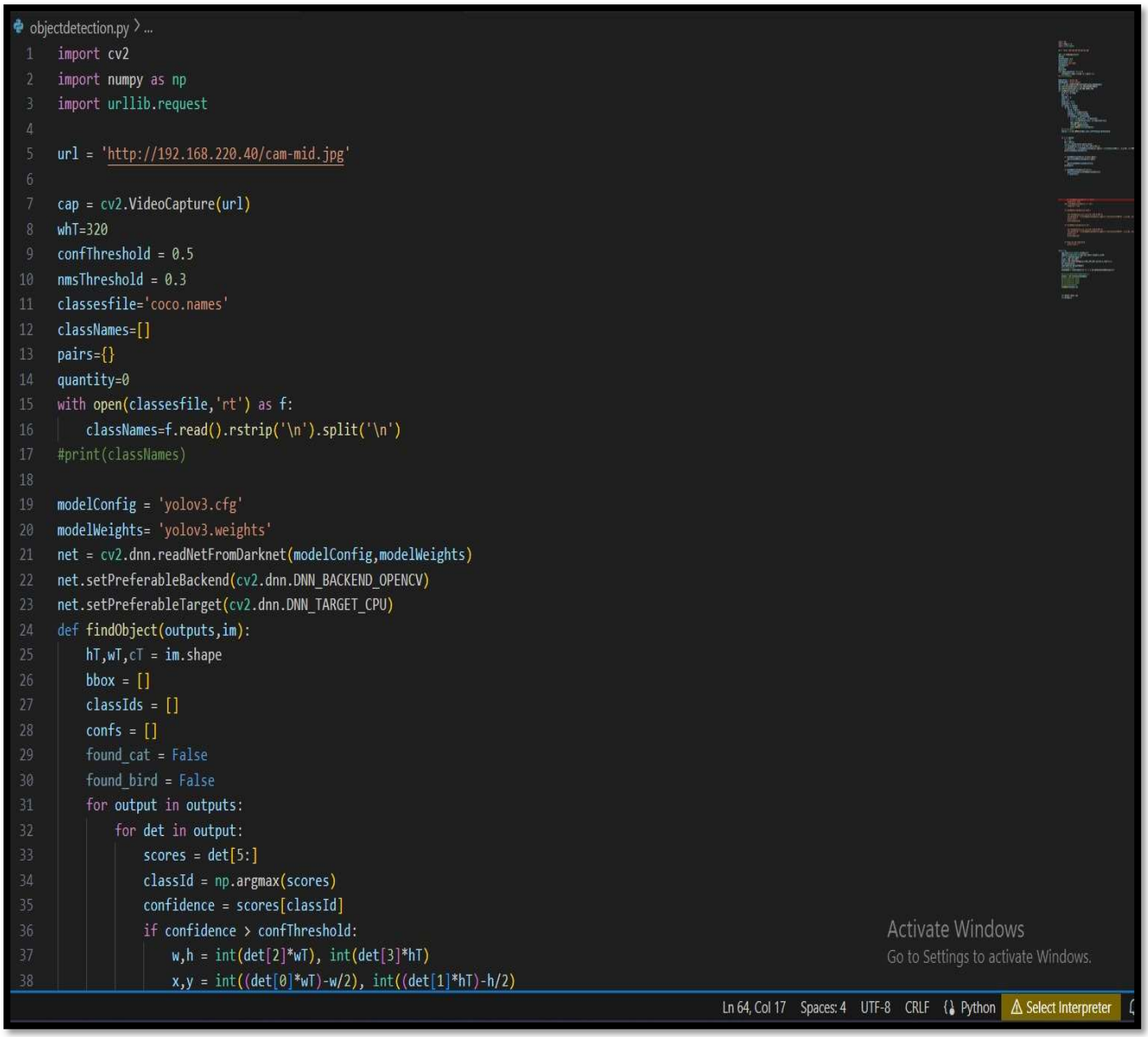
3.2 Design Selection



4. RESULTS AND VALIDATION

Smart inventory management

1) Object Detection using YOLO on Visual Studio



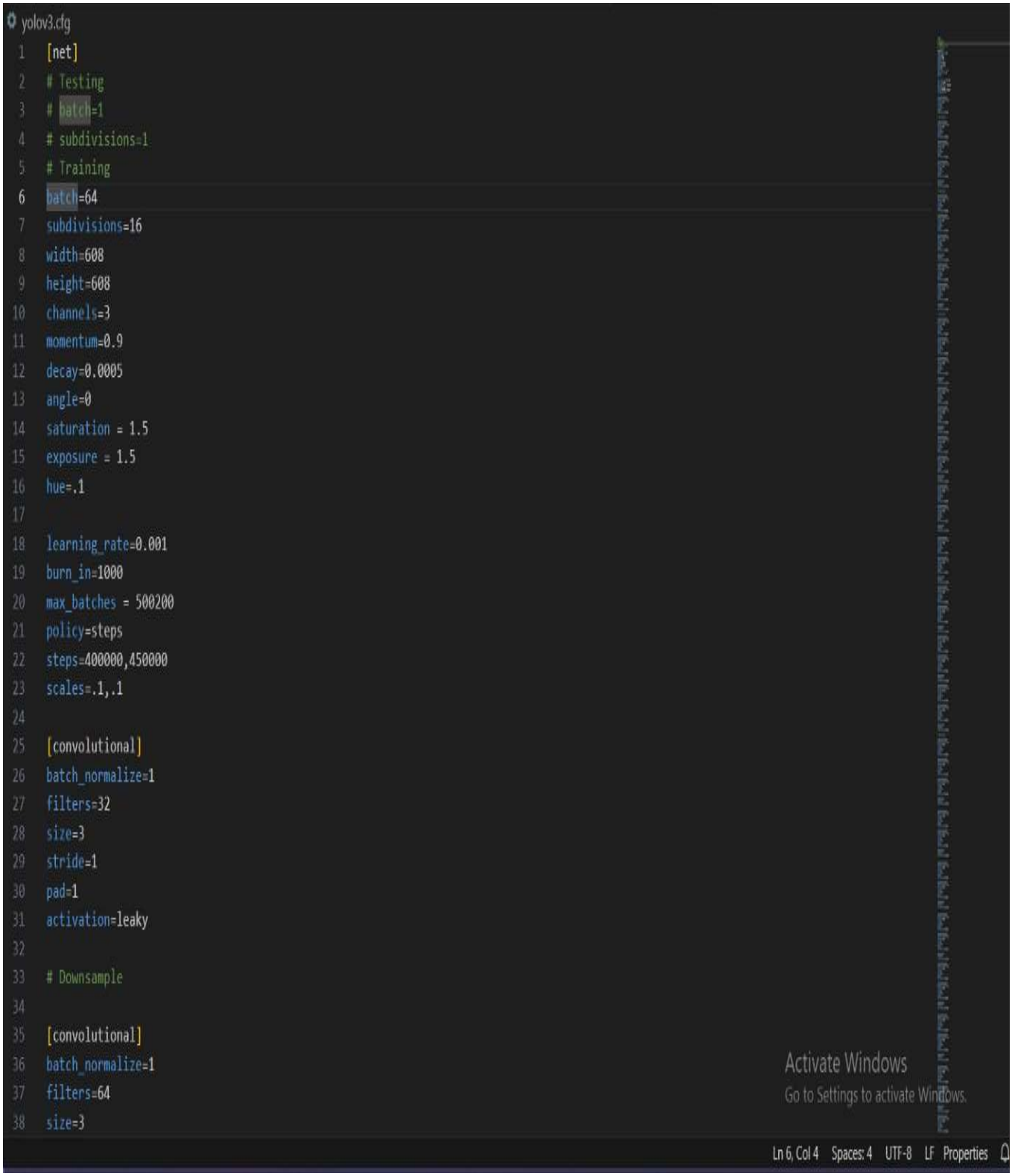
objectdetection.py > ...

```
import cv2
import numpy as np
import urllib.request

url = 'http://192.168.220.40/cam-mid.jpg'

cap = cv2.VideoCapture(url)
whT=320
confThreshold = 0.5
nmsThreshold = 0.3
classesfile='coco.names'
classNames=[]
pairs={}
quantity=0
with open(classesfile,'rt') as f:
    classNames=f.read().rstrip('\n').split('\n')
#print(classNames)

modelConfig = 'yolov3.cfg'
modelWeights= 'yolov3.weights'
net = cv2.dnn.readNetFromDarknet(modelConfig,modelWeights)
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)
def findObject(outputs,im):
    hT,wT,cT = im.shape
    bbox = []
    classIds = []
    confs = []
    found_cat = False
    found_bird = False
    for output in outputs:
        for det in output:
            scores = det[5:]
            classId = np.argmax(scores)
            confidence = scores[classId]
            if confidence > confThreshold:
                w,h = int(det[2]*wT), int(det[3]*hT)
                x,y = int((det[0]*wT)-w/2), int((det[1]*hT)-h/2)
```

Activate Windows
Go to Settings to activate Windows.

Ln 64, Col 17 Spaces: 4 UTF-8 CRLF Python Select Interpreter

2) Yolo Configuration for Object Detection



```
yolov3.cfg
[net]
# Testing
# batch=1
# subdivisions=1
# Training
batch=64
subdivisions=16
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 500200
policy=steps
steps=400000,450000
scales=.1,.1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

# Downsample

[convolutional]
batch_normalize=1
filters=64
size=3
```

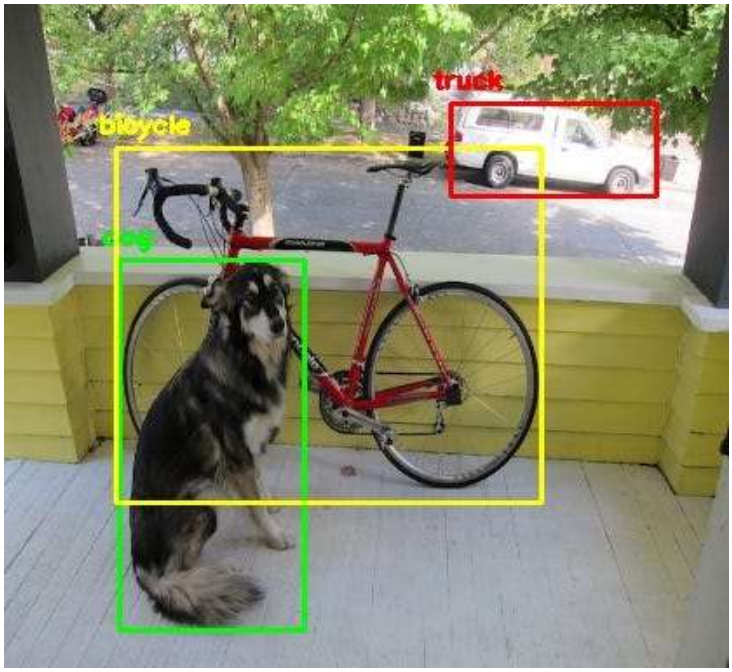
3) Arduino code for ESP8266 Module for Live Video feed

Smart_Inventory.ino

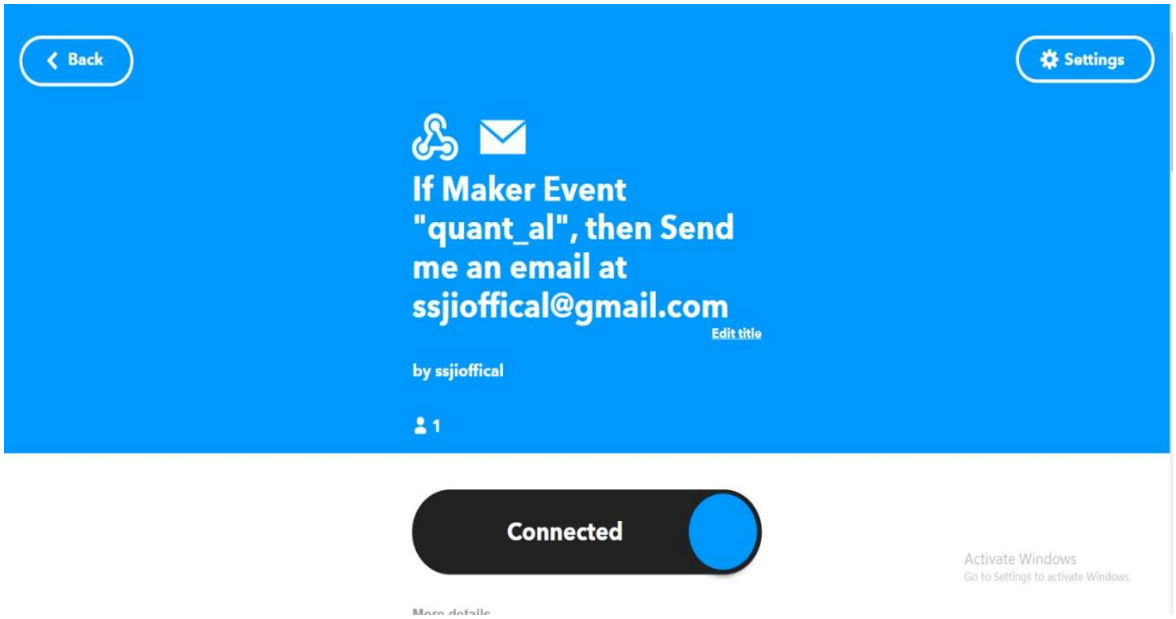
```
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

const char* WIFI_SSID = "moto";
const char* WIFI_PASS = "nnettt2@@@17";
const char* host = "maker.ifttt.com";
const char* apiKey = "UJdPMA_A8e5RoY-aEnY6p";
const int event=9;

WebServer server(80);


static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
void serveJpg()
{
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
                static_cast<int>(frame->size()));

  server.setContentLength(frame->size());
  server.send(200, "image/jpeg");
  WiFiClient client = server.client();
  frame->writeTo(client);
}

void handleJpgLo()
{
  if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
```

Ln 97, Col 34

4) Object Detection Using Yolo



5) IFTTT Webhooks Services Interface



To trigger an Event with 3 JSON values

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/ quant_al /with/key/UJdPMA_A8e5RoY-aEnY6p
```

With an optional JSON body of:

```
{ "value1" : " 0 ", "value2" : " 0 ", "value3" : " 0 " }
```

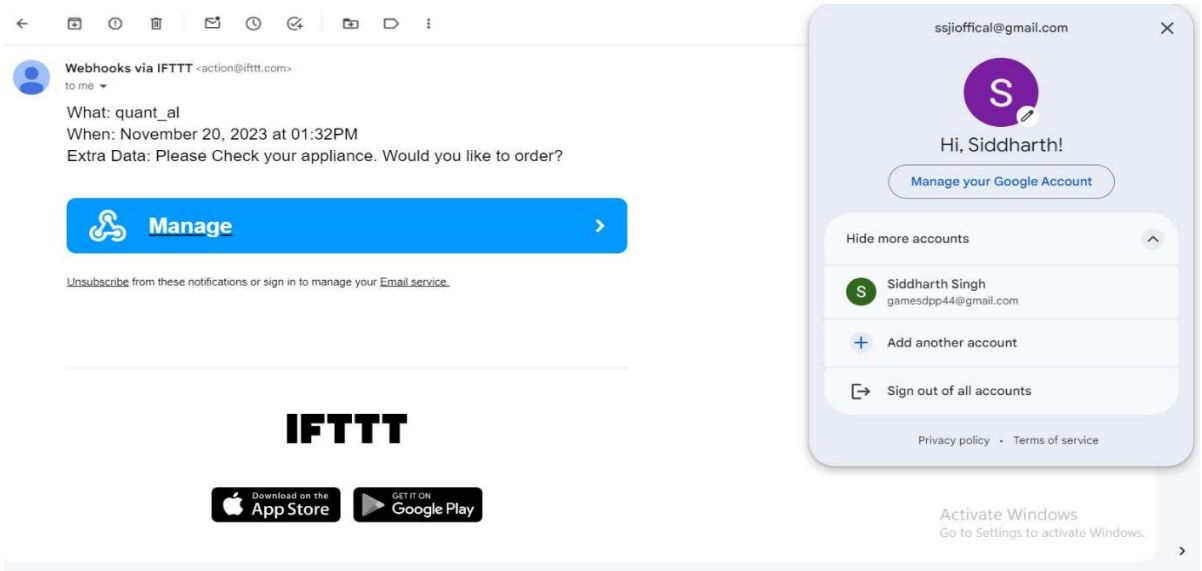
The data is completely optional, and you can also pass `value1`, `value2`, and `value3` as query parameters or form variables. This content will be passed on to the action in your Applet.

You can also try it with `curl` from a command line.

```
curl -X POST -H "Content-Type: application/json" -d '{"value1":"0","value2":"0","value3":"0"}'
https://maker.ifttt.com/trigger/quant_al/with/key/UJdPMA_A8e5RoY-aEnY6p
```

Please read our FAQ on using Webhooks for more info.

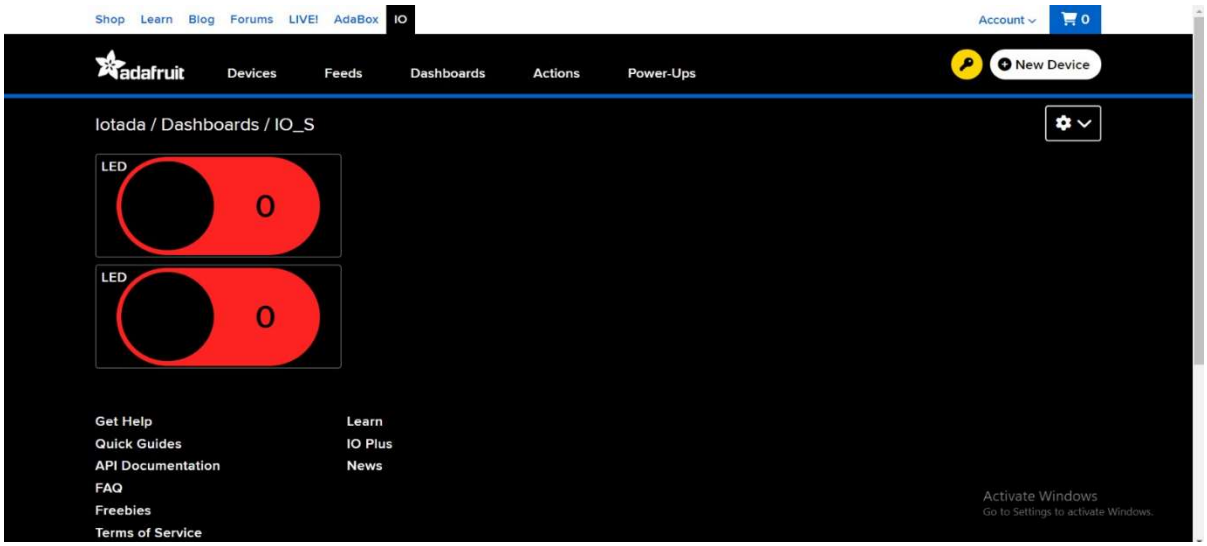
6) Mails Generated via IFTTT Webhooks services using Triggers



Smart Home Functionality Implementation

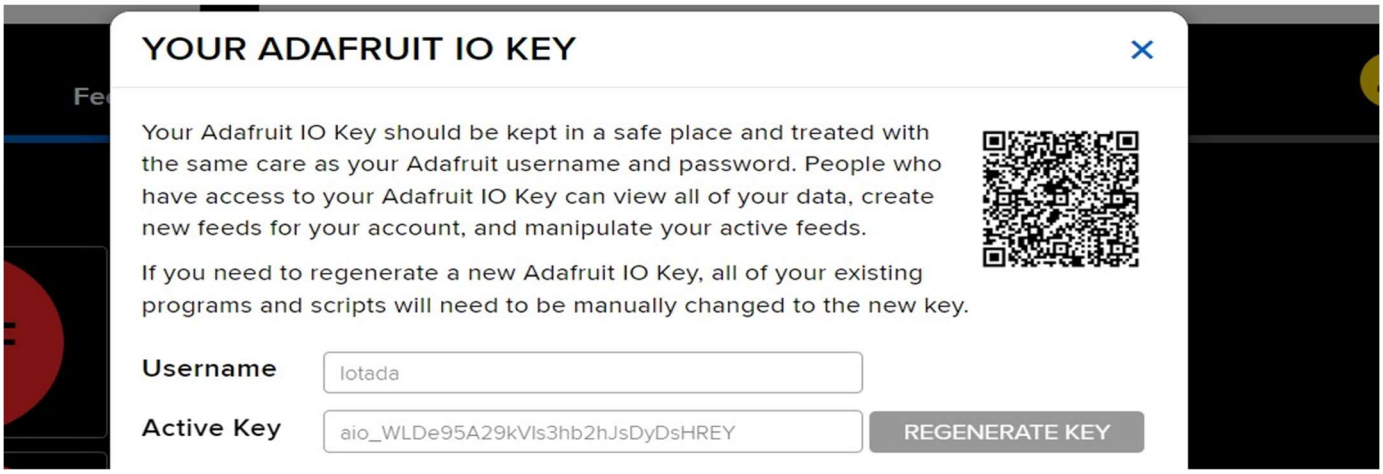
1) IoT-Enabled Smart Lighting Code Snippet

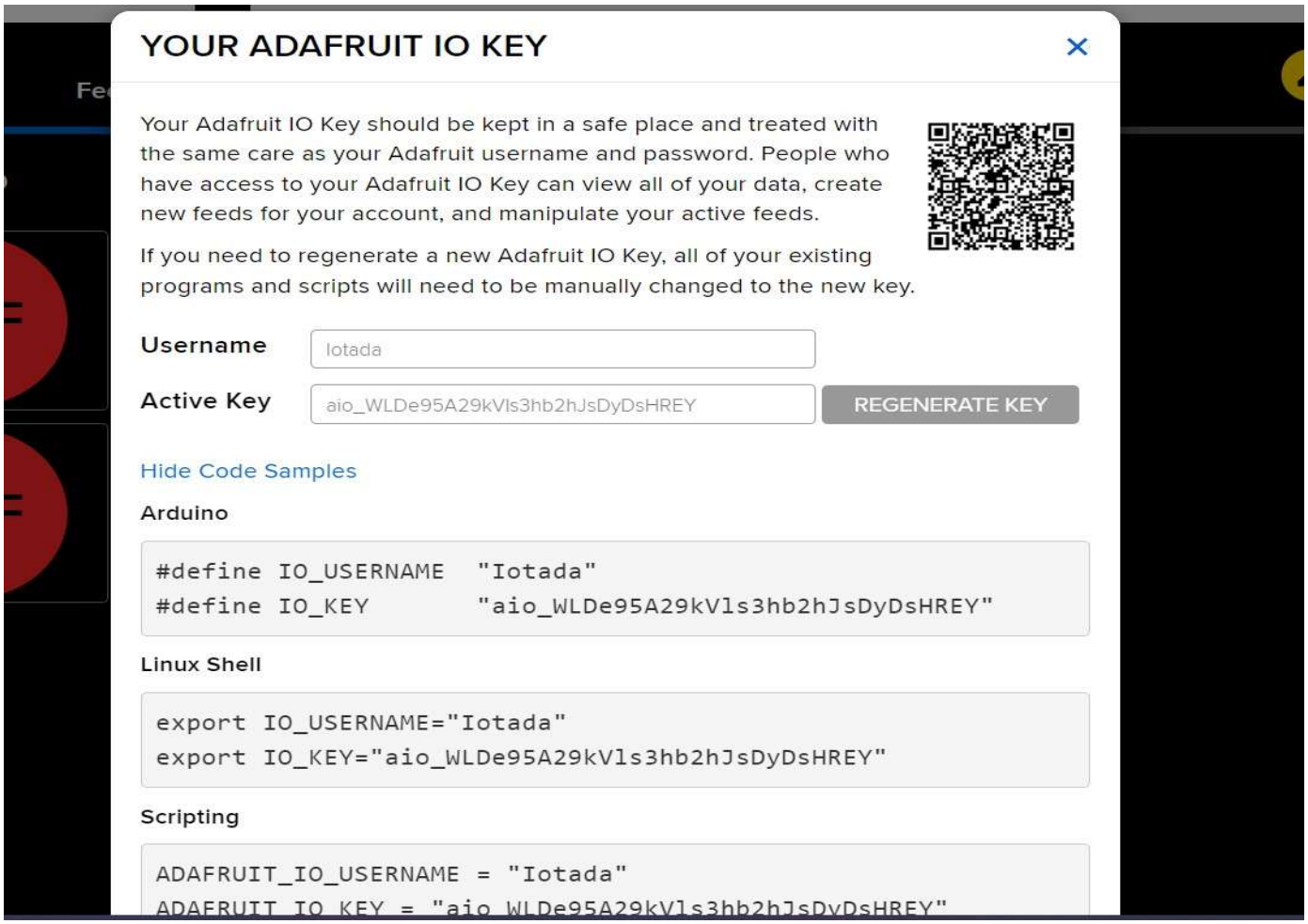
2) Adafruit Dashboard for Smart lighting



3) Arduino Code for Adafruit Connectivity

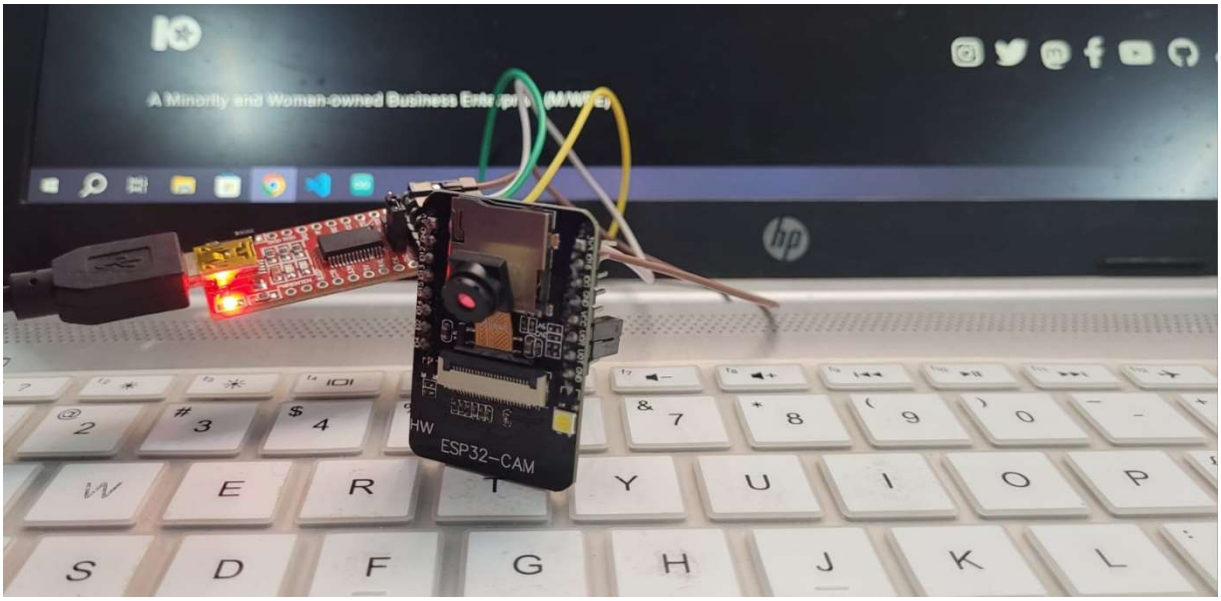
```
/************************ Adafruit IO Config *******************************/

// visit io.adafruit.com if you need to create an account,
// or if you need your Adafruit IO key.
#define IO_USERNAME "Iotada"
#define IO_KEY "aio_WLDe95A29kVls3hb2hJsDyDsHREY"

/******************************* WIFI **************************************/

// the AdafruitIO_WiFi client will work with the following boards:
//   - HUZZAH ESP8266 Breakout -> https://www.adafruit.com/products/2471
//   - Feather HUZZAH ESP8266 -> https://www.adafruit.com/products/2821
//   - Feather HUZZAH ESP32 -> https://www.adafruit.com/product/3405
//   - Feather M0 WiFi -> https://www.adafruit.com/products/3010
//   - Feather WICED -> https://www.adafruit.com/products/3056
//   - Adafruit PyPortal -> https://www.adafruit.com/product/4116
//   - Adafruit Metro M4 Express AirLift Lite ->
//  https://www.adafruit.com/product/4000
//   - Adafruit AirLift Breakout -> https://www.adafruit.com/product/4201
//   - Adafruit AirLift Shield -> https://www.adafruit.com/product/4285
//   - Adafruit AirLift FeatherWing -> https://www.adafruit.com/product/4264

#define WIFI_SSID "Redmi 9A"
#define WIFI_PASS "scientist@29"

// uncomment the following line if you are using airlift
// #define USE_AIRLIFT

// uncomment the following line if you are using winc1500
// #define USE_WINC1500

// uncomment the following line if you are using mrk1010 or nano 33 iot
//#define ARDUINO_SAMD_MKR1010
```


YOUR ADAFRUIT IO KEY
Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.
If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.
Username
Iotada
Active Key
aio_WLDe95A29kVls3hb2hJsDyDsHREY
REGENERATE KEY

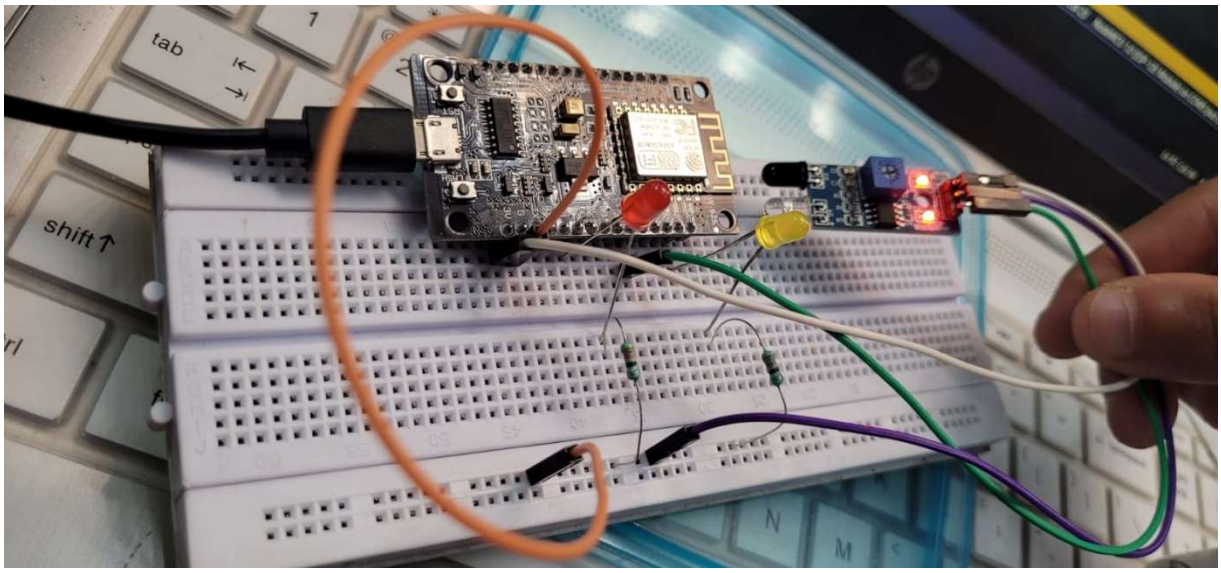

YOUR ADAFRUIT IO KEY
Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.
If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.
Username
Iotada
Active Key
aio_WLDe95A29kVls3hb2hJsDyDsHREY
REGENERATE KEY
Hide Code Samples
Arduino
#define IO_USERNAME "Iotada"
#define IO_KEY "aio_WLDe95A29kVls3hb2hJsDyDsHREY"
Linux Shell
export IO_USERNAME="Iotada"
export IO_KEY="aio_WLDe95A29kVls3hb2hJsDyDsHREY"
Scripting
ADAFRUIT_IO_USERNAME = "Iotada"

Hardware Demonstration

1) Object Detection Via ESP32 CAM Module



2) IoT-Enabled Smart lighting



6. CONCLUSION AND FUTURE SCOPE

In conclusion, the advent of smart home automation systems has brought about a remarkable transformation in the way we interact with and manage our living environments. These systems, which seamlessly integrate devices and technologies, have demonstrated their potential to enhance various aspects of our daily lives. The convenience, energy efficiency, security, and personalization offered by smart home automation have redefined modern living.

The project of Home automation has endless possibilities in future as we know how rapidly technology is transforming and with that there is a need for new ways to implement the previous existing system. Some of the future scope are mentioned as follows:

- ✓ Voice and Gesture Control: The refinement of voice recognition and gesture control technologies will enable users to interact with their smart homes using natural language and physical gestures, making the experience even more intuitive and user-friendly.

- ✓ Health and Wellness Monitoring: Home automation systems could evolve to include health and wellness monitoring features. This might involve integrating wearable devices, smart mirrors, or sensors to track vital signs, sleep patterns, and overall well-being.

- ✓ Smart Grid Integration: Home automation systems could play a pivotal role in smart grid integration, allowing homeowners to optimize their energy usage based on real-time energy pricing and grid demand.

- ✓ Integration of Emerging Technologies: Home automation systems will increasingly integrate emerging technologies like artificial intelligence (AI), machine learning, and edge computing. This will enable systems to learn user behavior patterns, adapt to changing preferences, and make intelligent decisions to optimize energy usage and enhance user experience

- ✓ Predictive Analytics and Proactive Automation: Systems will become more anticipatory, using predictive analytics to anticipate user needs and automate actions accordingly.

7. REFERENCES

[1] Yar H, Imran AS, Khan ZA, Sajjad M, Kastrati Z. Towards smart home automation using IoT-enabled edge-computing paradigm. Sensors. 2021 Jul 20.

[2] P. Suesaowaluk, "Home Automation System Based Mobile Application," 2020 2nd World Symposium on Artificial Intelligence (WSAI), Guangzhou, China, 2020

[3] Garg S, Yadav A, Jamloki S, Sadana A, Tharani K. IoT based home automation. Journal of Information and Optimization Sciences. 2020 Jan 2.

[4] Malagi M. Voice control personal assistant using Raspberry PI. 2019.

[5] Majeed R, Abdullah NA, Ashraf I, Zikria YB, Mushtaq MF, Umer M. An intelligent, secure, and smart home automation system. Scientific Programming. 2020 Oct 29.

[6] Alam T, Salem AA, Alsharif AO, Alhejaili AM. Smart home automation towards the development of smart cities. APTIKOM Journal on Computer Science and Information Technologies. 2020 Mar,5.

[7] Agarwal K, Agarwal A, Misra G. Review and performance analysis on wireless smart home and home automation using iot. In2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) 2019 Dec 12 (pp. 629-633) . IEEE.

[8] Chaudhary C, Kshetri A, Shrestha SL, Chapagain K. Design of Home Automation System using Dual-Tone Multi-Frequency Technique. Himalayan Journal of Applied Science and Engineering. 2021 Nov 24.

[9] Tamakloe E, Kommey B. A Smart GSM-Based Home Electrical Appliances Remote Control System. IPTEK The Journal for Technology and Science. 2022 May 18.

[10] Naing M, Hlaing NN. Arduino based smart home automation system. Int. J. Trend Sci. Res. Dev. 2019 Jun,3.

[11] "A Survey of Smart Home Architecture and Technology" by Elankavi Subramanian and R. Kalpana – This paper provides an overview of smart home architecture, technologies, and protocols used in various smart home automation systems.

[12] "User Experience in Smart Homes: A Systematic Literature Review" by Suleman Shahid and Sarah Faisal - This research paper focuses on user experience aspects in smart home automation systems, including usability, interaction design, and user satisfaction.

[13] H. K. Singh, S. Verma, S. Pal and K. Pandey, "A step towards Home Automation using IOT," 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 2019.

[14] A. Verma, S. Prakash, V. Srivastava, A. Kumar and S. C. Mukhopadhyay, "Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review," in IEEE Sensors Journal, vol. 19, no. 0, pp. 9036-9046, 15 Oct.15, 2019.

[15] Yosra Hajjaji, Wadii Boulila, Imed Riadh Farah, Imed Romdhani, Amir Hussain,"Big data and IoT based applications in smart environments: A systematic review", Computer Science Review, Volume 39,2021. S. Kayastha and P. Upadhyaya, "Design and Implementation of a Cost-Efficient Smart Home System with Raspberry Pi and Cloud Services," 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 2019.

[16] Helo MO, Shaker A, Abdul-Rahaim LA. Design nd Implementation of a Cloud Computing System for Smart Home Automation. Technology. 2021 Oct.