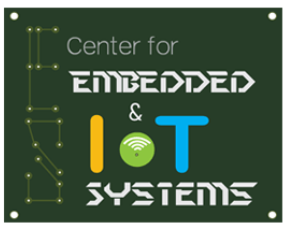
**SMART HOME USING MQTT PROTOCOL**

****

A project report submitted in partial fulfilment of requirement for the course

on

Security in IoT

by

G. HARINI (19K41A04C8)

G. SUJITH (19K41A04C9)

Under the guidance of

**Mr. P. Ramchandar Rao**

Assistant Professor & Member, Center for Embedded Systems and IoT

Department of ECE

****

**S R ENGINEERING COLLEGE**

Ananthasagar, Warangal

****

**CERTIFICATE**

This is to certify that the course project entitled **“SMART HOME USING MQTT PROTOCOL”** is the bonafied work carried out by G. Harini (19K41A04C8), G. Sujith (19K41A04C9) in the partial fulfilment of the requirement for the award of course **Security in IoT** during the academic year 2022-2023 under the guidance and supervision.

**Mr. P. Ramchandar Rao**

Assistant Professor &

Member, Center for Embedded Systems and IoT

Department of ECE

**CONTENTS**

Abstract 4

List of Figures 5

**Chapter no. Title Page.no**

**1 INTRODUCTION** 6

1.1 Literature review 6

1.2 Existing method 8

1.3 Proposed method 9

**2** **METHODOLOGY** 10

2.1 Block diagram 10

2.2 Hardware & Software components 10

2.3 Description of Hardware components 11

2.3.1 ESP32 Microcontroller 11

2.3.2 Gas Sensor 12

2.3.3 Relay Module 13

2.3.4 SEN-18 Water level Sensor 13

2.3.5 Servomotor 14

2.4 Working of proposed project 15

2.5 Limitations 16

**3 RESULTS AND DISCUSSIONS** 17

**4 CONCLUSION AND FUTURE SCOPE** 19

**REFERENCES** 20

**ABSTRACT**

In recent years, there has been a huge growth in the world of intelligent devices for home automation. Such gadgets are designed in order to ease the interaction between people and daily home duties. Although individually they are simple to work with, each appliance has its own configuration interface which adds overhead to the general user experience. This paper presents a solution for connecting more devices into a signal entity which can be easily accessed at any time. The implementation integrates the functionalities of different home automation devices into a single application. The [**concept of Home Automation**](https://smartify.in/knowledgebase/home-automation-details/) aims to bring the control of operating your every-day home electrical appliances to the tip of your finger, thus giving user affordable lighting solutions, better energy conservation with optimum use of energy. Apart from just lighting solutions, the concept also further extends to have an overall control over your home security as well as build a centralised home entertainment system and much more. The **Internet of Things** based Home Automation system, as the name suggests aims to control all the devices of your smart home through internet protocols.

**List of Figures**

**Fig No Title Page no**

2.1 Block diagram of proposed method 10

2.2 ESP32 Micro controller 11

2.3 Gas sensor 12

2.4 Relay Module 13

2.5 SEN-18 Water level sensor 14

2.6 Servo motor 14

3.1 Adafruit graph for MQ2 sensor 17

3.2 Adafruit graph for SEN18 water sensor 17

3.3 Adafruit graph for Servo (window) 18

3.4 Adafruit graph for Servo1 (door) 18

**CHAPTER-1**

**INTRODUCTION**

**1.1 LITERATURE REVIEW**

The Internet of Things (IoT) is a technology that allows connection between devices using the internet to collect and exchange data with each other. Privacy and security have become the most pressing issues in the IoT network, especially in the smart home. Nevertheless, there are still many smart home devices that have not implemented security and privacy policies. Smart home technology is the concept of integrating Information and Communication Technology (ICT) into home appliances, which facilitates the simple tasks to be automated and controlled remotely, thus improves the quality of life of smart home inhabitants.

This study proposes a remote sensor control system built on ESP32 to implement a smart home through the Message Queuing Telemetry Transport (MQTT) protocol by applying the Advanced Encryption Standard (AES) algorithm with a 256-bit key. It addresses security issues in the smart home by encrypting messages sent from users to sensors. The network was analysed using Wireshark, and it showed that the message sent was encrypted. This implementation could prevent brute force attacks, with the result that it could guarantee the confidentiality of a message. Meanwhile, from several experiments conducted in this study, the difference in the average time of sending encrypted and unencrypted messages was not too significant, i.e., 20ms. [1]

This paper aims to investigate the main advantages brought by Message Queuing Telemetry Transport (MQTT) protocol which is used by Internet of Things (IoT) devices. In addition to the advantages brought by the MQTT protocol in the smart grid environment we will also analyze the challenges and limitations that are introduced by it and were it should be adopted in the current smart grid. As a study case for this paper the practical implementation of MQTT in a smart home automation system will be presented.[2]

This paper presents the development of a low-cost smart home-automated system using Android application, which is communicating with MQTT broker for automatic control functionalities. The proposed system assists and supports people with special needs of elderly and physical impairments to increase their independence and to achieve a good quality of life and will enable users to control remotely home appliances using an Android smartphone and a control circuit. The proposed system is tested on controlling a servo motor through the developed android application, which can be used for controlling the door lock to open/close the house doors.[3]

In this paper we propose to implement the security among M2M smart home objects over 6LoWPAN networks, with the help of an enhanced mutual authentication and key establishment scheme. In the enhanced mutual authentication, a hybrid cryptography is used for secure authentication and suitable key establishment, for 6LoWPAN resource constrained devices. A handover ticket is generated for secure authentication, when there is mobility among nodes. In addition to it, we propose to use a communication protocol, MQTT (Message Queueing Telemetry Transport), which is a light weight, publish subscribe based protocol and more compatible with resource constrained devices like smart home objects in 6LoWPAN networks and gives better results in M2M communication.[4]

This paper describes the design and implementation of a wireless home automation system that can be used for general purposes in our life. The proposed system is designed for easy installation, so it could be use at any location to gather indoor conditions. The proposed system comprises of a wireless sensor and actuator network which is low cost and low power. Each wireless sensor node senses and transmits the variations in the local temperature, humidity and luminosity to the cloud database. The cloud server receives the data and stores it in the table and plotting the variations simultaneously. The user web interface provides graphical data analysis. Also, when the large variations in temperature, humidity and luminosity value occur, system will run actuators like heater, cooler, ventilation fan and lighting equipment’s.[5]

The main objective of this article is to develop a smart home system using SCADA, Raspberry Pi3 Module and Wemos-Dl board. Nowadays people are expecting to control the home appliances through remote control. This expectation is fulfilled by our advanced technology. In this paper, we are using SCADA, Transmission Control Protocol/Internet Protocol (TCP/IP) and Message Queuing Telemetry Transport (MQTT) to develop a smart home system.[6]

**1.2 EXISTING METHOD**

**Bluetooth based home automation system**

Home automation systems using smart phone, Arduino board and Bluetooth technology are secured and low cost. A Bluetooth based home automation system proposed by R.Piyare and M.Tazil. The Bluetooth system uses a PC or smart phone as receiver device. It has a high communication rate, great security and low cost, so it can be implemented as a real time system. Bluetooth network has limited range of 10 meters if the smart phone is out of range, then it will not be able to control the home appliances, this is one of the main disadvantages of Bluetooth based home automation system

**Voice recognition-based home automation**

A voice recognition-based home automation system proposed and implemented by a researcher. The wireless communication between the smart phone and the Arduino UNO is done through Bluetooth technology. This will be more helpful for handicapped and aged people who wants to control appliances by speaking voice command The main drawback of this system is that communication between user and voice recognition tool depends on signal to noise ratio (SNR), if voice signal is noisy then communication can highly effect and the system will fail to show accuracy.

**Zig Bee Based Wireless Home Automation System**

Zig Bee based wireless home automation system has also been studied, Zig Bee is similar to Bluetooth technology. It is one of the broadly used transceiver standard with low data rate and power. It has physical range is between 10 to 20 meters, which can increase up to 150 meters by using direct sequence spread spectrum (DSSS). It is ideal for developing prototypes and research related activities.

**GSM Based Home Automation System**

A smart home automation system implemented by using Global System for Mobile communication (GSM). In GSM based home automation systems, communication between main module and appliances is done through text messages. The main drawback of GSM based home automation system is that, there is no guarantee text message deliver to the system every time so it is not a reliable system.

* These are the drawbacks of existing methods, to overcome that drawbacks we are implementing “IOT Based Smart security and Smart Home Automation”.

**1.3 PROPOSED METHOD**

In order to eradicate the problems previously identified we came up with a concept called home automation using MQTT protocol, which is formed by introducing the concept of IOT to connect all appliances in house. It may include setting complex lighting systems in advance, setting alarms and home security controls, all connected by a central hub and remote-controlled by a mobile application. In this we are using an application for door locking system whereas, Gas sensor is used to detects the gas leakage from the cylinder and opens the windows and hence sends the alert to the user. Water level detector to see the tank overflow problem and helps the user on motor use-age. Relay Module is used to control home appliances by the user’s choice through an interface. Message Queuing Telemetry Transport (MQTT) protocol is used to send the data to the database and to the client.

**CHAPTER-2**

**METHODOLOGY**

**2.1 BLOCK DIAGRAM OF PROPOSED METHOD**

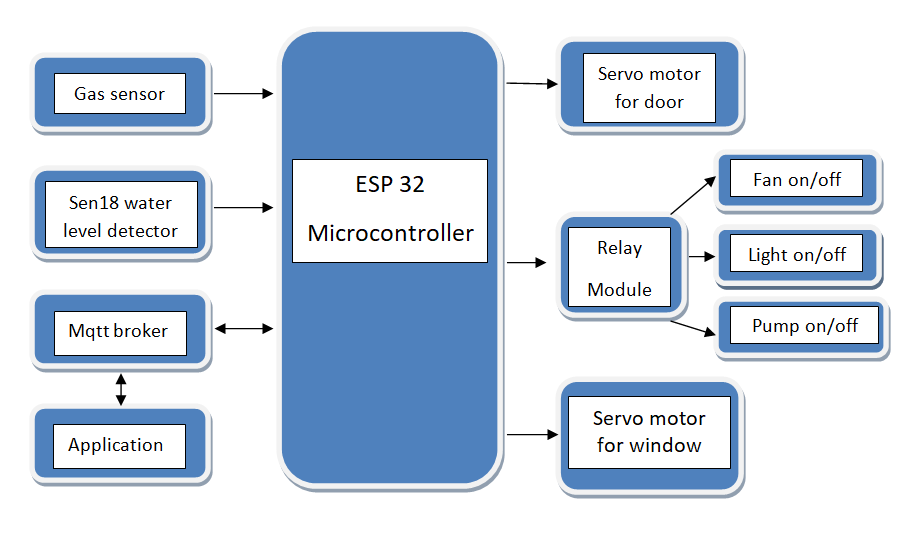


Figure 2.1 Block diagram of proposed method

**2.2 HARDWARE & SOFTWARE COMPONENTS**

**2.2.1. Hardware components:**

We have made the system in which the components used are:

* Esp32
* Gas sensor
* Relay Modules
* SEN-18 Water level sensor
* 2 Servo motors

**2.2.2 Software required:**

* Arduino IDE software
* Adafruit software

**2.3 DESCRIPTION OF HARDWARE COMPONENTS**

**2.3.1 ESP-32 Microcontroller:**

ESP32 is a low-cost, low-power Microcontroller with an integrated Wi-Fi and Bluetooth, which is also a low-cost Wi-Fi microchip albeit with limited vastly limited functionality. It is an integrated antenna and RF balun, power amplifier, low-noise amplifiers, filters, and power management module.

Figure 2.2 ESP32 Micro controller

**Specifications:**

* Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, running at 160 or 240 MHz
* Memory: 520 KB SRAM
* Wi-Fi: 802.11 b/g/n
* Bluetooth: v4.2 BR/EDR and BLE
* 12-bit × 18 ADC channels
* 2 × 8-bit [DACs](https://en.wikipedia.org/wiki/Digital-to-analog_converter)
* 10 × touch sensors ([capacitive sensing](https://en.wikipedia.org/wiki/Capacitive_sensing) GPIOs)
* 4 × [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus)
* 2 × [I2C](https://en.wikipedia.org/wiki/I%C2%B2C) interfaces
* 3 × [UART](https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter)
* [SD](https://en.wikipedia.org/wiki/Secure_Digital)/[SDIO](https://en.wikipedia.org/wiki/Secure_Digital#SDIO_cards)/[CE-ATA](https://en.wikipedia.org/wiki/CE-ATA)/[MMC](https://en.wikipedia.org/wiki/MultiMediaCard)/[EMMC](https://en.wikipedia.org/wiki/MultiMediaCard#eMMC) host controller
* SDIO/SPI slave controller
* [CAN bus](https://en.wikipedia.org/wiki/CAN_bus) 2.0
* Infrared remote controller (TX/RX, up to 8 channels)
* Motor [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation)
* LED [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) (up to 16 channels)
* Ultra-low-power analog pre-amplifier
* All security feature of IEEE 802.11 standard, like WFA, WPA/WPA2, and [WAPI](https://en.wikipedia.org/wiki/WLAN_Authentication_and_Privacy_Infrastructure), secure boot, Flash encryption
* Cryptographic hardware acceleration method like [AES](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard), [SHA-2](https://en.wikipedia.org/wiki/SHA-2), [RSA](https://en.wikipedia.org/wiki/RSA_(cryptosystem)), [elliptic curve cryptography](https://en.wikipedia.org/wiki/Elliptic_curve_cryptography) (ECC), [random number generator](https://en.wikipedia.org/wiki/Random_number_generator) (RNG)
  + 1. **Gas sensor:**

Gas sensors also known as gas detectors are electronic devices that detect and identify different types of gases. They are commonly used to detect toxic or explosive gases and measure gas concentration.



Figure 2.3 Gas sensor

**Specifications:**

* Operating Voltage is+5V.
* Analog output voltage: 0V to 5V.
* Digital output voltage: 0V or 5V (TTL Logic)
* The sensitivity of Digital pin can be varied using potentiometer
* Can be used as digital or analog sensor.
  + 1. **Relay Module:**

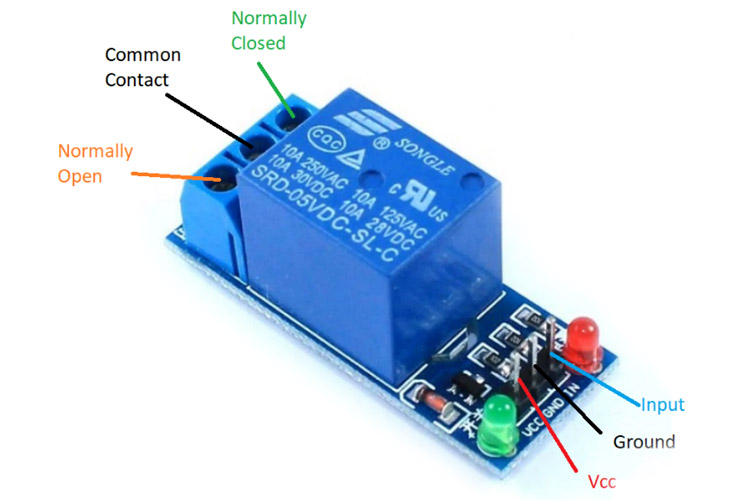
**** A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit

Figure 2.4 Relay Module

**Specifications:**

* Normal Voltage is 5V DC
* Normal Current is 70mA
* AC load current Max is 10A at 250VAC or 125V AC
* DC load current Max is 10A at 30V DC or 28V DC
* It includes 5-pins & designed with plastic material
* Operating time is 10msec
* Release time is 5msec
* Maximum switching is 300 operating per minute
  + 1. **SEN-18 Water level sensor**

Water Level sensors detect the level of liquids and other fluids and fluidized solids, including slurries, granular materials, and powders that exhibit an upper free surface.



Figure 2.5 SEN-18 Water level sensor

**Specifications:**

* Operating voltage: DC3-5V
* Operating current: less than 20mA
* Sensor Type: Analog
* Detection Area: 40mmx16mm
* Operating temperature: 10℃-30℃
* Humidity: 10% -90% non-condensing
  + 1. **Servo motor**

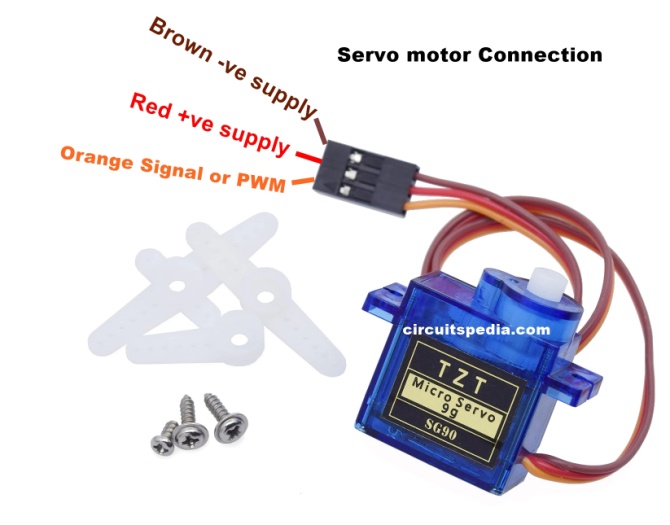
A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback.

Figure 2.6 Servo motor

**Specifications of servo motor:**

* Operating Voltage is +5V typically
* Torque: 2.5kg/cm
* Operating speed is 0.1s/60°
* Gear Type: Plastic
* Rotation: 0°-180°
* Weight of motor: 9gm
* Package includes gear horns and screws

A barcode is an

optical device which shows data on certain

products like unique ID. Purpose of using

barcode scanner is to automatically identify

the product from its unique barcode label

printed on it

**2.4 WORKING OF PROPOSED PROJECT**

The first step is starts with the scanning of

The barcode of the product.

2. Then system need to check for the switch

Condition is there Add SW. Is pressed or the

Sub SW. Is pressed.

3. If any switch is pressed then it need to

Compare for the load cell weight with the

The first step is starts with the scanning of

The barcode of the product.

2. Then system need to check for the switch

Condition is there Add SW. Is pressed or the

Sub SW. Is pressed.

3. If any switch is pressed then it need to

Compare for the load cell weight with the

This paper aims to investigate the main advantages brought by Message Queuing Telemetry Transport (MQTT) protocol which is used by Internet of Things (IoT) devices. As a study case for this paper the practical implementation of MQTT in a smart home automation system will be presented. IOT home automation is the ability to control domestic appliances by electronically controlled, internet-connected systems. In this system, Servo motor is integrated as entrance which can be controlled by application and helps the house to be secured. A Gas sensor is integrated in kitchen for detecting the leakage of gas from cylinder and send alert to the user’s mobile and opens the window to get the air into the house to prevent gas explosion and makes house atmosphere clean. Relay Module is used to operate the home appliances from the users mobile. Water level detector is used to monitor the water level in the tank. When the water overflows, the information is sent to the client and can be monitored using MQTT protocol.

**Smart Lighting-**

* Smart Lighting for home helps in saving energy by adapting the life to the ambient condition and switching on/off or dimming the light when needed.
* Smart Lighting solutions for homes achieve energy saving by sensing the human movements and their environments and controlling light accordingly.

**Smoke/gas detectors-**

* Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire.
* Gas detectors can detect the presence of harmful gases such as CO, LPG etc and raise the raise the alerts.

**2.5 LIMITATIONS**

* Compatibility Considerations
* Internet Outages
* Complex Technology Systems to Learn
* Initial Start-up Costs
* Power Supply

**CHAPTER-3**

**RESULTS AND DISCUSSIONS**

**3.1 RESULTS AND DISCUSSIONS**

The utility of trolley will be first of its

kind for commercial use. This device records

the data of the different products with help of

the suitable sensors like RFID Tags. This

recorded data helps the shop owner with

detailed analysis of shopping by the customer

& their preferences through the computer;

printout of the same can be obtained. In

Automatic trolley, there is no need to pull

heavy trolley, no need to wait in billing queue

and no need of thinking about budget. The

microcontroller based trolley automatically

follows the customer. Also it maintains safe

distance between customer and itself. It gives

number of products in trolley and total cost of

the products on the spot

The utility of trolley will be first of its

kind for commercial use. This device records

the data of the different products with help of

the suitable sensors like RFID Tags. This

recorded data helps the shop owner with

detailed analysis of shopping by the customer

& their preferences through the computer;

printout of the same can be obtained. In

Automatic trolley, there is no need to pull

heavy trolley, no need to wait in billing queue

and no need of thinking about budget. The

microcontroller based trolley automatically

follows the customer. Also it maintains safe

distance between customer and itself. It gives

number of products in trolley and total cost of

the products on the spot

This project records the data of different sensors and stores in the Adafruit Io website. MQ2 Sensor is used to detect the Nitrogen gas that is leaked from the cylinders and opens window to free flow of air. This recorded data helps the house owner to see what is happening in the house. Below is the data of Mq2 sensor taken from website.

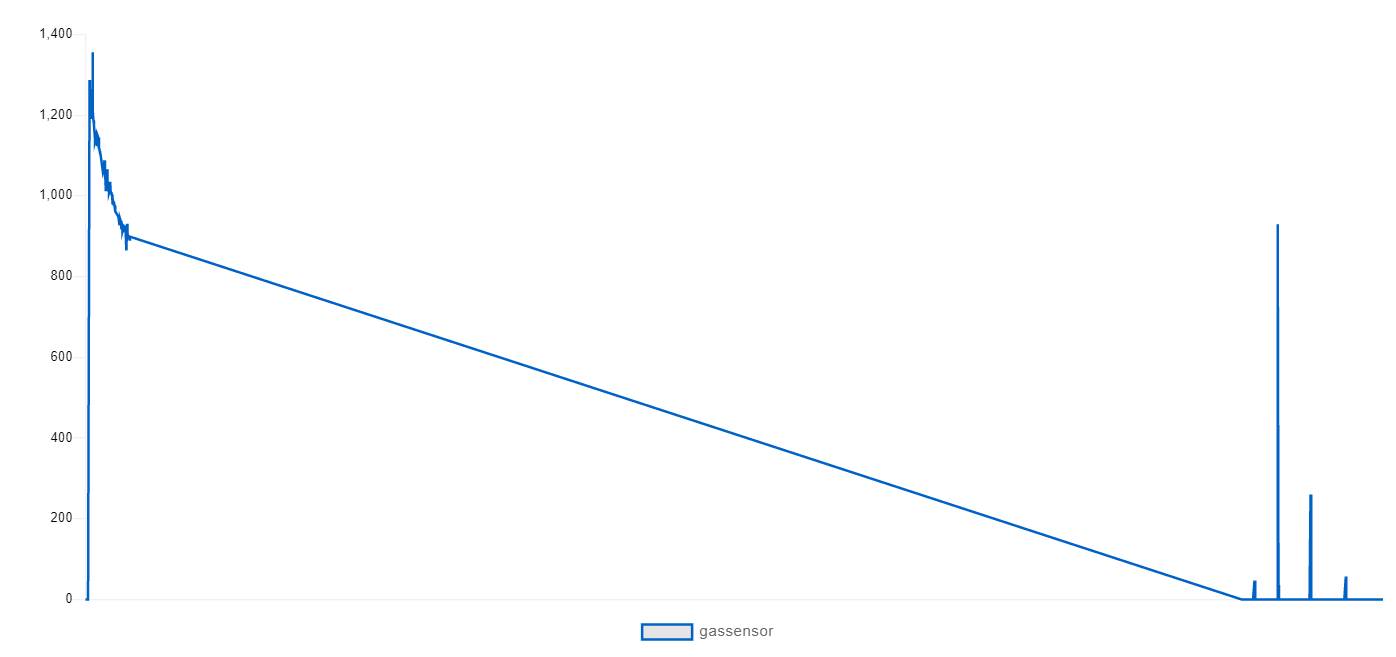


Fig-3.1: Adafruit graph for MQ2 sensor

The below graph represents the data of SEN18 water sensor. If the tank is empty tells the user to ON the motor else tells to OFF the motor.

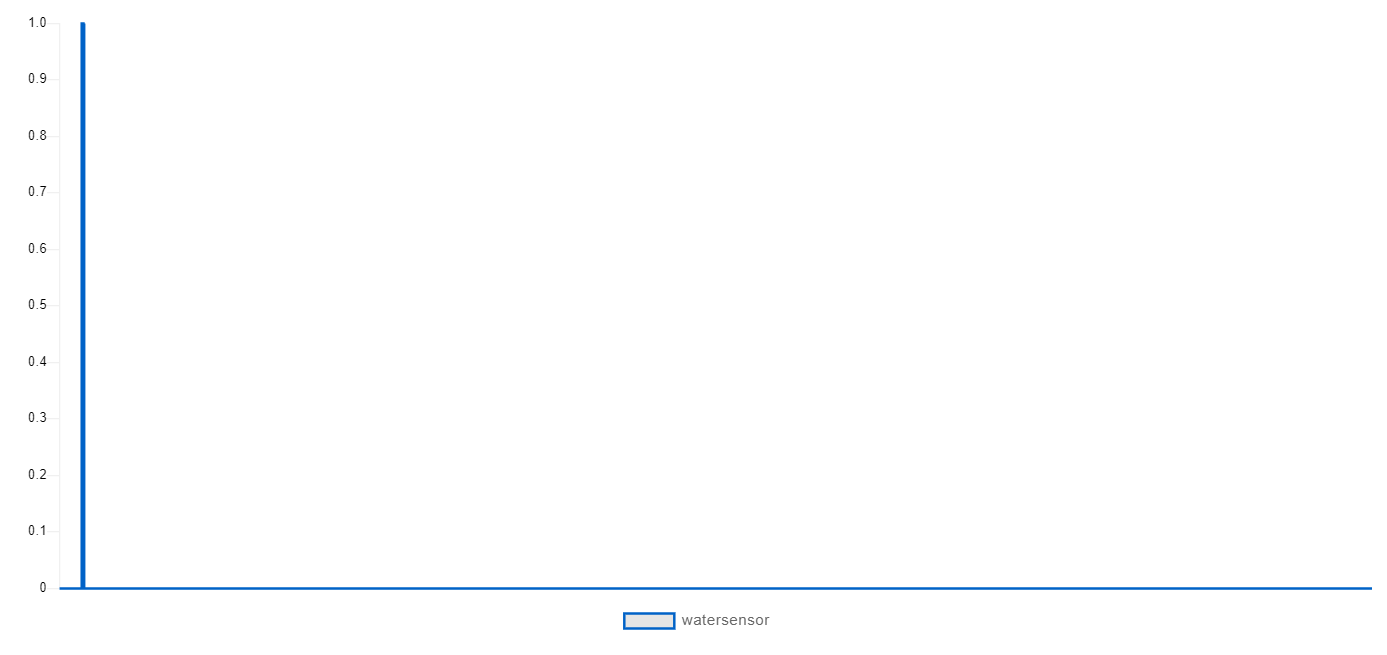


Fig-3.2: Adafruit graph for SEN18 water sensor

The data of two servos represent the door and window is shown below. The door can be controlled by the user. The window can be controlled by both user and Mq2 sensor that detects the gas.

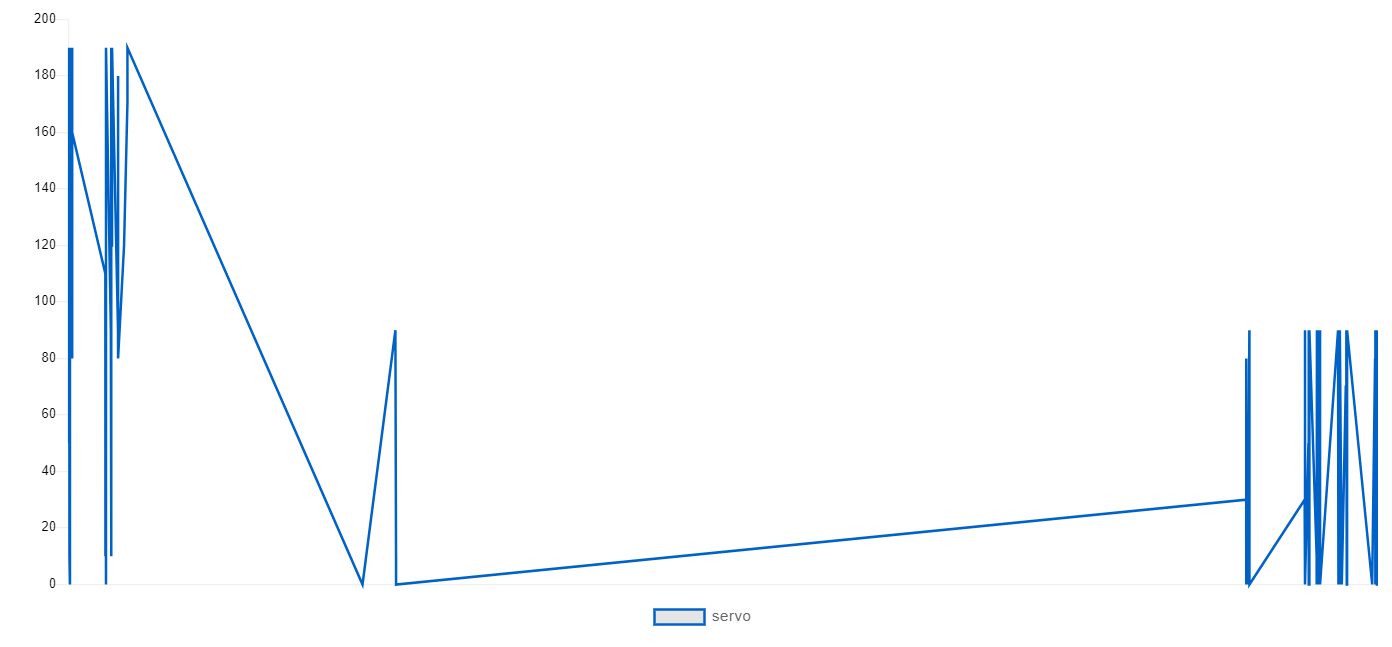


Fig-3.3: Adafruit graph for Servo (window)

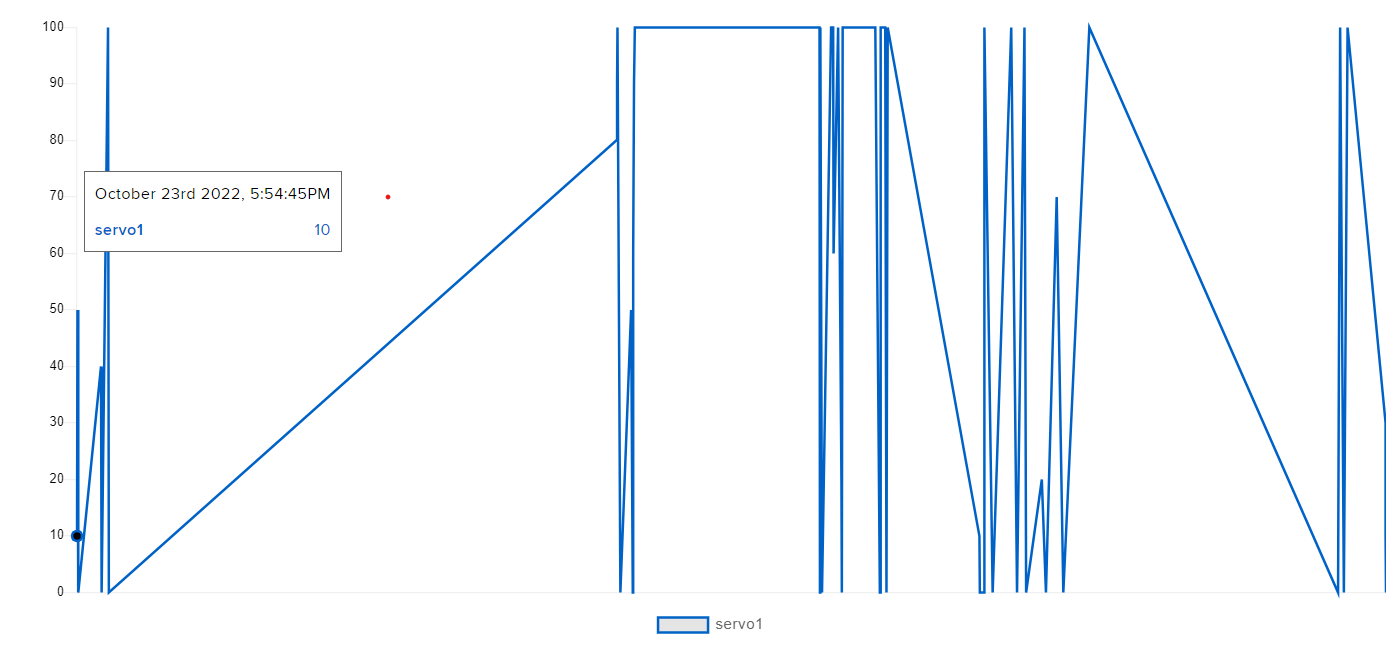
****

Fig 3.4: Adafruit graph for Servo1 (door)

**CHAPTER-4**

**CONCLUSION AND FUTURE SCOPE**

**4.1 CONCLUSION**

The developed product is easy to use, economical and does not require any special training. This project simplifies the home automation, makes it swift. This will take the home space to a different level. This technology is one of the most promising technologies for deployment in enterprise applications like retail stores and other commercial houses etc. Moreover, using a cam module for door locking is very secured. And by water level sensor we can save water without overflow from tank and the gas leakage can be also detected and helps the user a safe environment in the house.

**4.2 FUTURE SCOPE**

A further scope in this work can be viewed in taking this further ahead. A cloud platform can be used to aggregate, analyse and visualize data. Customized GUI can be developed to remotely access the devices to monitor and control them. Further research can help in development of Occupancy -aware control systems, where it is possible to sense the occupancy of the home using smart meter-sand environmental sensors like CO2 sensors, which can be integrated into the building automation system to trigger automatic responses for energy efficiency and building comfort applications. Appliance control and integration with the smart grid and smart meter can be made a reality, taking advantage, instance, of high solar panel output in the middle of the *dayrun* washing machines. Indoor positioning systems can be improved by home.

**REFERENCES**

[1] F. B. Setiawan and Magfirawaty, "Securing Data Communication Through MQTT Protocol with AES-256 Encryption Algorithm CBC Mode on ESP32-Based Smart Homes," 2021 International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE), 2021, pp. 166-170, doi: 10.1109/COSITE52651.2021.9649577.

[2] A. Cornel - Cristian, T. Gabriel, M. Arhip-Calin and A. Zamfirescu, "Smart home automation with MQTT," 2019 54th International Universities Power Engineering Conference (UPEC), 2019, pp. 1-5, doi: 10.1109/UPEC.2019.8893617.

[3] A. Eleyan and J. Fallon, "IoT-based Home Automation Using Android Application," 2020 International Symposium on Networks, Computers and Communications (ISNCC), 2020, pp. 1-4, doi: 10.1109/ISNCC49221.2020.9297320.

[4] S. Biju and N. M. Shekokar, "Security approach on MQTT based smart home," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 1106-1114, doi: 10.1109/ICPCSI.2017.8391883.

[5] S. A. Celtek, M. Durgun and H. Soy, "Internet of Things based smart home system design through wireless sensor/actuator networks," 2017 2nd International Conference on Advanced Information and Communication Technologies (AICT), 2017, pp. 15-18, doi: 10.1109/AIACT.2017.8020054.

[6] B. N. Alhasnawi and B. H. Jasim, "SCADA controlled smart home using Raspberry Pi3," 2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA), 2018, pp. 1-6, doi: 10.1109/ICASEA.2018.8370946.

[7] A. Arunachalam, R. Raghuraman, P. Obed Paul and J. Vishnupriyan, "A System for Energy Management and Home Automation," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), 2021, pp. 1-3, doi: 10.1109/ICSCAN53069.2021.9526526.

[8] R. Kishore Kodali, S. C. Rajanarayanan, L. Boppana, S. Sharma and A. Kumar, "Low Cost Smart Home Automation System using Smart Phone," 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), 2019, pp. 120-125, doi: 10.1109/R10-HTC47129.2019.9042467.

[9] D. Minchev and A. Dimitrov, "Home automation system based on ESP8266," 2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA), 2018, pp. 1-4, doi: 10.1109/SIELA.2018.8447172.

[10] R. Piyare and M. Tazil, “Bluetooth based home automation system using cell phone,” in 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), 2011, pp. 192–195.

[11] Kodali, Ravi & Soratkal, Sreeramya. (2016). MQTT based home automation system using ESP8266. 1-5. 10.1109/R10-HTC.2016.7906845.

[12] Yalcinkaya, Fikret & AYDİLEK, Hüseyin & Erten, Mustafa & İNANÇ, Nihat. (2020). IoT based Smart Home Testbed using MQTT Communication Protocol. Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi. 317. 10.29137/umagd.654056.

[13] K. Gill, S. Yang, F. Yao, and X. Lu, “A ZigBee-Based Home Automation System,” IEEE Trans. Consum. Electron., vol. 55, no. 2, pp. 422–430, 2009.

[14] Vadluri, Thirupathi. (2018). Implementation of Home Automation System using MQTT Protocol and ESP32.

[15] Jasmeet Chhabra, Punit Gupta “IOT BASED SMART HOME DESIGN USING POWER AND SECURITY MANAGEMENT”, Jaypee University of Information Himachal Pradesh, India on 2016 1st International Conference on Innovation and Challenges in Cyber Security (ICICCS 2016).

[16] Ayush Panwar, Anandita Singh, Renu Kumawat, Siddharth Jaidka, Kumkum Garg “EYRIE SMART HOME AUTOMATION USING INTERNET OF THINGS”, Manipal University Jaipur, India on Computing Conference 2017, 18-20 July 2017 at London, UK.

[17] Vamsikrishna Patchava, Hari Babu Kandala, P Ravi Babu “ A SMART HOME AUTOMATION TECHNIQUE WITH RASPBERRY PI USING IOT” on 2015 International Conference on Smart Sensors and Systems (IC-SSS).

**PROJECT CODE**

**Main code:**

#include<ESP32Servo.h>

#include<WiFi.h>

#include "config.h"

#if defined(ARDUINO\_ARCH\_ESP32)

#include <ESP32Servo.h>

#else

#include <Servo.h>

#endif

#include "Adafruit\_MQTT.h"

#include "Adafruit\_MQTT\_Client.h"

// WiFi parameters

#define WLAN\_SSID "Gurrapu-2.4"

#define WLAN\_PASS "Sukram#2"

// Adafruit IO

#define AIO\_SERVER "io.adafruit.com"

#define AIO\_SERVERPORT 1883

#define AIO\_USERNAME "sujio"

#define AIO\_KEY "aio\_SYmx61azm8haZywPaxmg17eIbpWJ"

WiFiClient client;

// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.

Adafruit\_MQTT\_Client mqtt(&client, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);

Adafruit\_MQTT\_Publish gassensor = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/gassensor");

Adafruit\_MQTT\_Publish watersensor = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/watersensor");

#define SERVO\_PIN 13

#define SERVO\_PIN1 21

Servo servo;

Servo servo1;

AdafruitIO\_Feed \*servo\_feed = io.feed("servo");

AdafruitIO\_Feed \*servo1\_feed = io.feed("servo1");

const int waterPin = 23;

int pos;

int gassensorPin = 34;

int sensorValue;

int limit = 700;

void setup() {

Serial.begin(9600);

ESP32PWM::allocateTimer(0);

ESP32PWM::allocateTimer(1);

ESP32PWM::allocateTimer(2);

ESP32PWM::allocateTimer(3);

// Debug console

servo.setPeriodHertz(50); //Temperature T0 from datasheet, conversion from Celsius to kelvin

pinMode(waterPin, INPUT);

Serial.println(F("Adafruit IO Example"));

// Connect to WiFi access point.

Serial.println(); Serial.println();

delay(10);

Serial.print(F("Connecting to "));

Serial.println(WLAN\_SSID);

WiFi.begin(WLAN\_SSID, WLAN\_PASS);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(F("."));

}

Serial.println();

Serial.println(F("WiFi connected"));

Serial.println(F("IP address: "));

Serial.println(WiFi.localIP());

// connect to adafruit io

connect();

while(! Serial);

// tell the servo class which pin we are using

servo.attach(SERVO\_PIN);

servo1.attach(SERVO\_PIN1);

// connect to io.adafruit.com

Serial.print("Connecting to Adafruit IO");

io.connect();

servo\_feed->onMessage(handleMessage);

servo1\_feed->onMessage(handleMessage1);

// wait for a connection

while(io.status() < AIO\_CONNECTED) {

Serial.print(".");

delay(500);

}

Serial.println();

Serial.println(io.statusText());

servo\_feed->get();

servo1\_feed->get();

}

void connect() {

Serial.print(F("Connecting to Adafruit IO... "));

int8\_t ret;

while ((ret = mqtt.connect()) != 0) {

switch (ret) {

case 1: Serial.println(F("Wrong protocol")); break;

case 2: Serial.println(F("ID rejected")); break;

case 3: Serial.println(F("Server unavail")); break;

case 4: Serial.println(F("Bad user/pass")); break;

case 5: Serial.println(F("Not authed")); break;

case 6: Serial.println(F("Failed to subscribe")); break;

default: Serial.println(F("Connection failed")); break;

}

if(ret >= 0)

mqtt.disconnect();

Serial.println(F("Retrying connection..."));

delay(10000);

}

Serial.println(F("Adafruit IO Connected!"));

}

void loop()

{

io.run();

// ping adafruit io a few times to make sure we remain connected

if(! mqtt.ping(3)) {

// reconnect to adafruit io

if(! mqtt.connected())

connect();

}

servo.attach(SERVO\_PIN);

servo1.attach(SERVO\_PIN1);

sensorValue = analogRead(gassensorPin);

Serial.println("gas sensor Value : ");

Serial.println(sensorValue);

delay(2000);

int waterStatus = digitalRead(waterPin);

delay(1000);

if (waterStatus == 0)

{

Serial.print("tank empty ON motor");

Serial.println(waterStatus);

delay(1000);

}

else

{

Serial.print("Tank Full Off Motor");

Serial.println(waterStatus);

delay(1000);

}

delay(5000);

if (! gassensor.publish(sensorValue)) { //Publish to Adafruit

Serial.println(F("Failed"));

}

if (! watersensor.publish(waterStatus)) { //Publish to Adafruit

Serial.println(F("Failed"));

}

else {

Serial.println(F("Sent gas and water!"));

}

}

void handleMessage(AdafruitIO\_Data \*data) {

// convert the data to integer

int angle = data->toInt();

if(angle <= 20)

angle = 0;

else if(angle >= 80||sensorValue>=limit)

angle = 90;

servo.write(angle);

}

//door control

void handleMessage1(AdafruitIO\_Data \*data) {

// convert the data to integer

int angle = data->toInt();

if(angle <= 0)

{

angle = 0;

Serial.print("off");

}

else if(angle >= 100)

{

angle = 100;

Serial.print("on");

}

servo1.write(angle);

}

**Config.h code:**

#define IO\_USERNAME "sujio"

#define IO\_KEY "aio\_SYmx61azm8haZywPaxmg17eIbpWJ"

#define WIFI\_SSID "Gurrapu-2.4"

#define WIFI\_PASS "Sukram#2"

// uncomment the following line if you are using airlift

// #define USE\_AIRLIFT

#include "AdafruitIO\_WiFi.h"

#if defined(USE\_AIRLIFT) || defined(ADAFRUIT\_METRO\_M4\_AIRLIFT\_LITE) || \

defined(ADAFRUIT\_PYPORTAL)

// Configure the pins used for the ESP32 connection

#if !defined(SPIWIFI\_SS) // if the wifi definition isnt in the board variant

// Don't change the names of these #define's! they match the variant ones

#define SPIWIFI SPI

#define SPIWIFI\_SS 10 // Chip select pin

#define NINA\_ACK 9 // a.k.a BUSY or READY pin

#define NINA\_RESETN 6 // Reset pin

#define NINA\_GPIO0 -1 // Not connected

#endif

AdafruitIO\_WiFi io(IO\_USERNAME, IO\_KEY, WIFI\_SSID, WIFI\_PASS, SPIWIFI\_SS,

NINA\_ACK, NINA\_RESETN, NINA\_GPIO0, &SPIWIFI);

#else

AdafruitIO\_WiFi io(IO\_USERNAME, IO\_KEY, WIFI\_SSID, WIFI\_PASS);

#endif