NEW HORIZON COLLEGE OF ENGINEERING

MINI PROJECT REPORT ON

"HOME AUTOMATION USING ESP32"

SUBMITTED BY:

Rohit M (1NH18EC062)

ROHAN S PATIL (1NH18EC097)

RUPENDRAN A M(1NH18EC138)

RAVI N (1NH18EC141)

Under the guidance of

DR.ARAVINDA K

Associate Professor, Dept. of ECE, NHCE, Bengaluru.

NEW HORIZON COLLEGE OF ENGINEERING

(ISO-9001:2000 certified, Accredited by NAAC 'A', Autonomous
college permanently affiliated to VTU)Outer Ring Road, Panathur
Post, Near Marathalli, Bengaluru-560103

NEW HORIZON COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled “**HOME AUTOMATION USING ESP32**” carried out by **ROHIT M(1NH18EC062), ROHAN S PATIL (1NH18EC097), RUPENDRAN A M (1NH18EC138),RAVI N (1NH18EC141)** bonafide students of Electronics and Communication Department, New Horizon College of Engineering, Bangalore.

The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Project Guide	HOD ECE
DR.ARAVINDA K	
-------------------------	----------------------

<u>External Viva</u>

Name of Examiner Signature with Date

1.

2.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **DR. ARAVINDA K** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

Rohit M (1NH18EC062)

ROHAN S PATIL (1NH18EC097)

RUPENDRAN A M(1NH18EC138)

RAVI N (1NH18EC141)

INDEX

ABSTRACT ……………………………………………………………………………………………05

KEYWORDS ……………………………………………………………………………………………06

CHAPTER-01:

Introduction………………………………………………………………………07

CHAPTER-02:

Literature survey ………………………………………………………………10

CHAPTER-03:

Proposed Methodology …………………………………………………………11

CHAPTER-04:

Project Description ……………………………………………………………12

Software/ Hardware Description ………………………………………………12

Features …………………………………………………………………………14

Components ………………………………………………………………………15

CHAPTER-05:

Result ……………………………………………………………………………21

Advantages ………………………………………………………………………22

Limitations ………………………………………………………………………22

Applications ……………………………………………………………………23

CHAPTER-06:

Future scope ……………………………………………………………………25

Conclusion…………………………………………………………………………25

Reference …………………………………………………………………………26

APPENDIX ……………………………………………………………………………27

LIST OF FIGURES

SL No	FIGURE No	FIGURE DESCRIPTION	Page No
1	1.1	Architecture of IOT based home monitoring and device control	8
2	1.2	working of MQTT protocol	9
3	3.1	working protocol of our project	13
4	3.2	circuit diagram of the project	13
5	4.1	IMAGE OF ESP32	15
6	4.2	PIN DIAGRAM OF ESP32	16
7	4.3	FUNCTIONAL BLOCK DIAGRAM	17
8	4.4	5V ADAPTER	19
9	4.5	RESISTOR	19
10	4.6	IMAGE OF 1N4007 DIODE	20
11	4.7	NPN TRANSISTOR	21
12	4.8	5V RELAY	22
13	5.1	FINAL OUTCOME	23
14	5.2	ADAFRUIT.IO	23
15	5.3	adafruit.io when switch is on	23

LIST OF TABLES

SL No	Table No	TABLE DESCRIPTION	Page No
1	1.1	Literature review	11

ABSTRACT

This project presents a design and prototype implementation of new home automation system that uses Wi-Fi technology as a network infrastructure connecting its parts. The proposed system consists of two main components; the first part is the server (web server), which presents system core that manages, controls, and monitors users' home.

Users and system administrator can locally (LAN) or remotely (internet) manages and control system code. Second part is hardware interface module, which provides appropriate interface to sensors and actuator of home automation system.

Unlike most of available home automation system in the market the proposed system is scalable that one server can manage many hardware interface modules as long as it exists on Wi-Fi network coverage. System supports a wide range of home automation devices like power management components, and security components.

The proposed system is better from the scalability and flexibility point of view than the commercially available home automation systems.

In the era of Internet of Things (IOT) and digital technology automation of everything has become more popular. People are very smart, they want to control and monitor everything from working places. This paper explains a cloud based home automation system using MQTT protocol. It enable users to control and monitor home appliances using a mobile app or a web page. Using cloud technologies became cost effective because most cloud developers are offering their services freely.

CHAPTER-1

INTRODUCTION

The day by day innovation improved from automatic machine to customer products. IoT is another pattern advancement that empowers us to screen and control hardware devices through the web. Here we propose to use IoT in order to screen and control home apparatuses, in this way computerizing present day homes through a web. This proposed framework enables a consumer to effortlessly control these home apparatuses through the web. The undertaking proposes a capable usage for IOT utilized for checking and controlling the home apparatuses by means of World Wide Web. Home robotization framework utilizes the reasonable gadgets as a UI. They can likewise speak with home computerization organize through an Internet access, by strategies for low power correspondence traditions like Zigbee, WI-Fi, etc. This endeavor goes for controlling home machines by methods for Smartphone using Wi-Fi as correspondence tradition and raspberry pi as server structure. The IOT based Monitoring and Controlling System for home is a progression which can control and screen gadgets not just for home mechanization but any real life appliances remotely. It provides facility to have control over a wide range of home appliances and ensure securities. In this project we presented implementation of home automation system through the Wi-Fi module, ADAFRUIT.IO web server. And Webpage server using ESP32. This project monitors all home appliances such as light, fan and controlled based on the threshold value programmed in the ESP 32. This system is low cost, allowing additional home appliances. Home monitoring and device controlling without involvement of humans the system can be control the devices, So that we can save the time and energy levels. So now we have introducing this system.

The fig 1.1 shows the architecture of the system here we are using the four sensors and four devices for the home appliances for monitoring and controlling devices for the home respectively. The data can be sending to open source cloud storage for the store the data and use the MQTT app Inventor for display the status of the sensors.

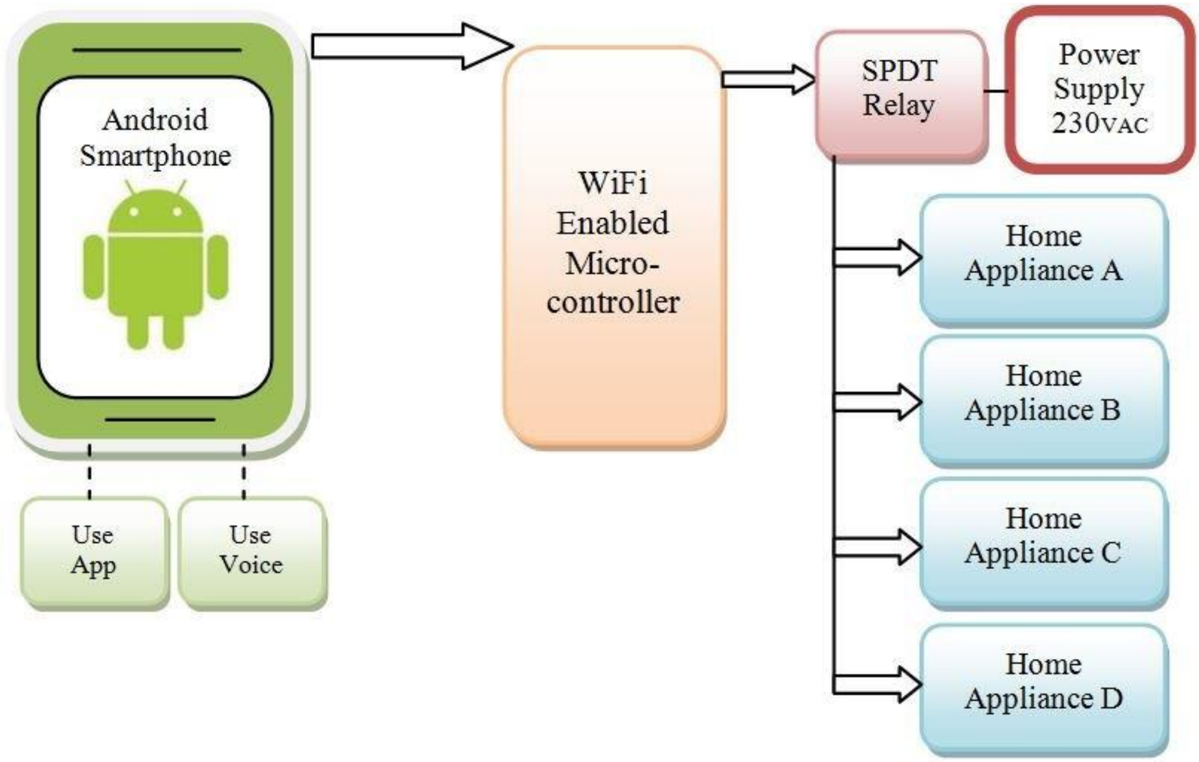


Figure 1.1: Architecture of IOT based home monitoring and device control.

Software

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a word processor for composing code, a message zone, a book reassure, a toolbar with catches for regular capacities and a progression of menus. It associates with the Arduino and Genuine equipment to transfer programs and speak with them. The ESP32 board is associated with a PC through USB, where it interfaces with the Arduino advancement climate (IDE). The client composes the Arduino code in the IDE, and afterward transfers it to the microcontroller which executes the code, cooperating with sources of info and yields, for example, sensors, engines, and lights.

MQTT

In a simple language when you want to communicate to your controller through mobile or a web browser you need a communication protocol and MQTT does that magic. MQTT is a message queuing telemetry transport. It is specifically designed for automation. It's having a very low footprint to send and receive data. So it uses very less amount of data to send and receive and mobile app using MQTT also consumes less battery. So MQTT is the best for this project.

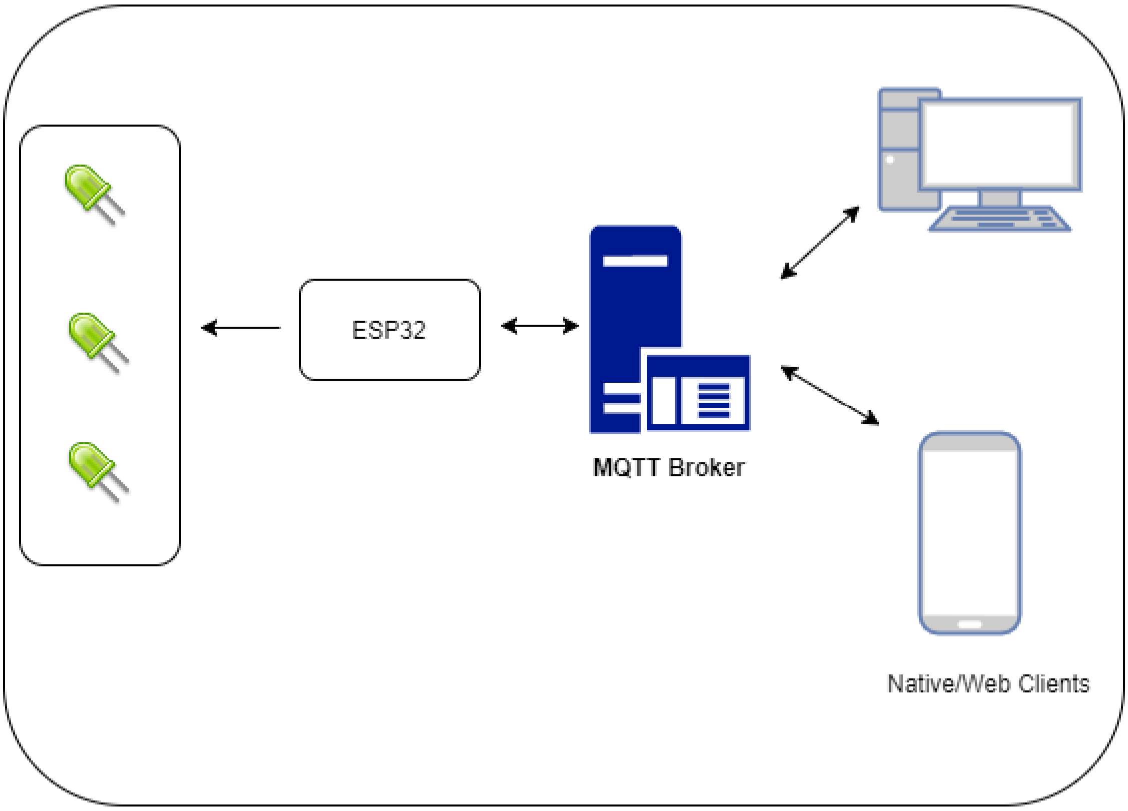


Figure 1.2: working of MQTT protocol.

CHAPTER-2

LITERAURE SURVEY

Wi-Fi based home computerization framework fundamentally comprises three modules, the worker, the equipment interface module, and the product bundle. The figure shows the framework model design. Wi-Fi innovation is utilized by worker, and equipment Interface module to speak with one another. A similar innovation uses to login to the worker online application. The worker is associated with the web, so distant clients can get to worker electronic application through the web utilizing viable internet browser. Programming of the most recent home mechanization framework is part to worker application programming, and Microcontroller (Arduino) firmware. The Arduino programming, assembled utilizing C language, utilizing IDE accompanies the microcontroller itself. Arduino programming is at fault for get-together occasions from associated sensors, at that point applies activity to actuators and prearranged in the worker. Another occupation is to report and record the set of experiences in the worker DB. The worker application programming bundle for the proposed home mechanization framework is an online application constructed utilizing asp.net. The worker application programming can be gotten to from inside organization or from web if the worker has genuine IP on the web utilizing any web pilot underpins asp.net innovation. Worker application programming is prepared to do; keep up the entire home robotization framework, arrangement, and setup. Worker use information base to keep log of home computerization framework segments, we decide to utilize XML records to save framework log.

Title of the paper	**Author & Year of Publication**	**Outcome**	**Limitation**
IOT home automation	Kunal Verma July 12,2018	Control home appliances	High expensive
Smart living using Bluetooth	M.Yan and H.shi	Bluetooth based systems.	Not fast enough
Bluetooth home iot using mobile automat	r.piyare	Control home appliances using mobile	Can hacked
Rtos based home automation	S,anwaaruullah	home automation	Expensive

CHAPTER-3

PROPOSED METHODOLOGY

In this advanced period, the idea of home computerization is developing at incredible speed. Keen frameworks are being introduced in pretty much every spot. Keen Home Systems is an idea where all the electrical machines or the gadgets are controlled utilizing a solitary controller. In these frameworks, more often than not, the controller is a portable application. As an android versatile is the most well-known among the individuals, so an android application is the most ideal choice to control every one of these gadgets.

So in this undertaking, we will interface a portion of the home's electrical apparatuses to the Relay module and control it through ESP32 Microcontroller. We will make a firebase continuous information base and interface it from the android application. This android application will send the information to the cloud and afterward it will be shipped off the microcontroller to turn the electrical apparatus on or off. Best of all, you can have full power over the exchanging of your machines from anyplace on the planet. You simply need a web association with work the android application.

In this task, we will control home AC machines with adafruit.io which will be useful in our IOT home robotization projects.

In this article we will control a bulb associated with AC mains through hand-off module with adafruit.io worker and ESP32. Adaruit.io is entirely viable worker with Arduino IDE to make IOT based tasks. This application can without much of a stretch be perused on the site and can undoubtedly be arranged. This site is easy to understand and can be utilized to make great IOT projects.

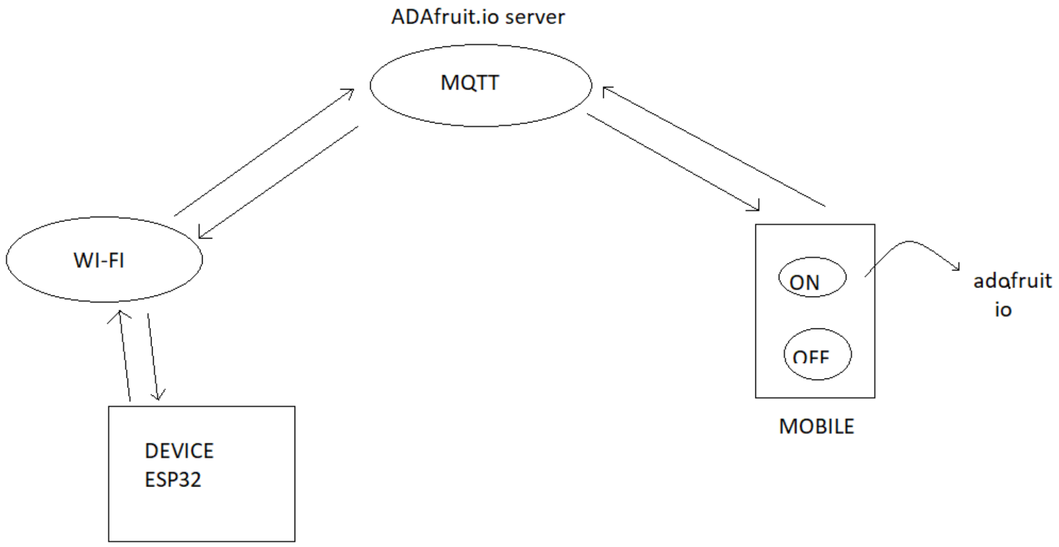


Figure 3.1: working protocol of our project

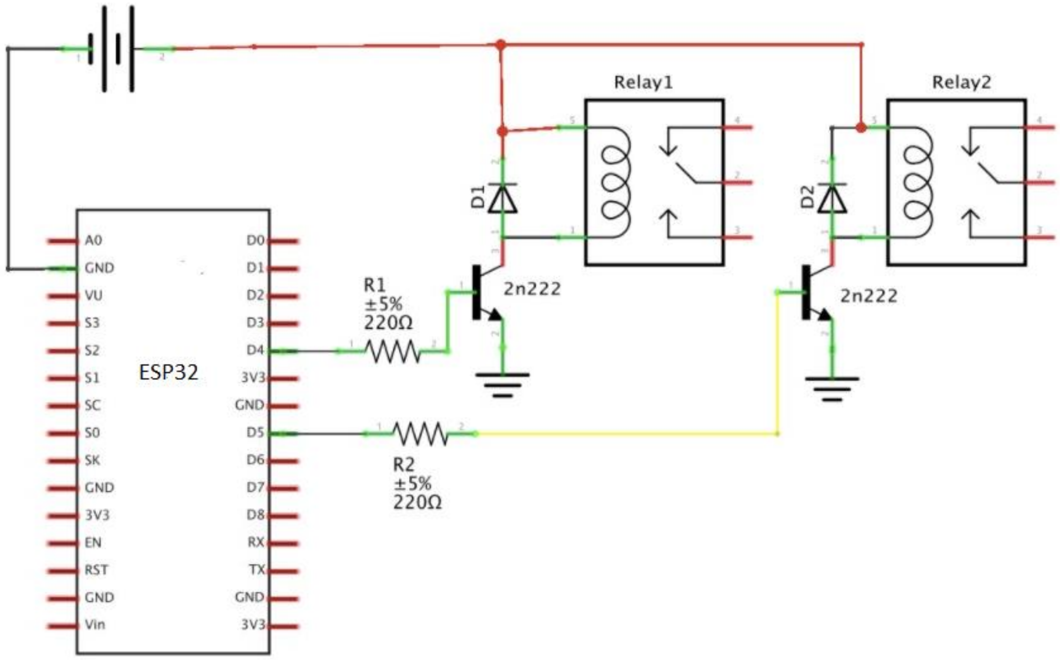


Figure 3.2: circuit diagram of the project

CHAPTER-4

PROJECT DESCRIPTION

Software description

The Arduino Integrated Development Environment (IDE) is a cross stage application (for Windows, macOS, Linux) that is written in capacities from C and C++. It is utilized to compose and transfer projects to Arduino viable sheets, yet in addition, with the assistance of outsider centers, other seller improvement sheets.

The source code for the IDE is delivered under the GNU General Public License, variant 2. The Arduino IDE upholds the dialects C and C++ utilizing exceptional principles of code organizing. The Arduino IDE supplies a product library from the Wiring project, which gives numerous regular information and yield techniques. Client composed code just requires two essential capacities, for beginning the sketch and the primary program circle, that are assembled and connected with a program stub fundamental() into an executable cyclic chief program with the GNU instrument chain, likewise included with the IDE circulation. The Arduino IDE utilizes the program avrdude to change over the executable code into a book document in hexadecimal encoding that is stacked into the Arduino board by a loader program in the board's firmware. Naturally, avrdude is utilized as the transferring apparatus to streak the client code onto official Arduino sheets.

With the rising ubiquity of Arduino as a product stage, different sellers began to actualize custom open source compilers and devices (centers) that can fabricate and transfer portrayals to other microcontrollers that are not upheld by Arduino's true line of microcontrollers.

In October 2019 the Arduino association started giving early admittance to another Arduino Pro IDE with troubleshooting and other progressed highlights.

Hardware description

ESP32

ESP32 is a system on chip microcontroller with inbuilt Bluetooth and Wi-Fi. It is a low cost and flexible to develop application for low powered systems. In this application ESP32 microcontroller accept messages from clients sends to respective relays to control the appliances. It has a CPU of Xtensa dual-core (or single-core) 32-bit LX6 microprocessor; operating at 160 or 240 MHz and performing at up to 600 DMIPS with 520 KB RAM.

Cloud MQTT is a universal known MQTT broker, used to implement “Internet of Things” (IOT) applications. To configure it, we require providing server name, user name, and password and port number.

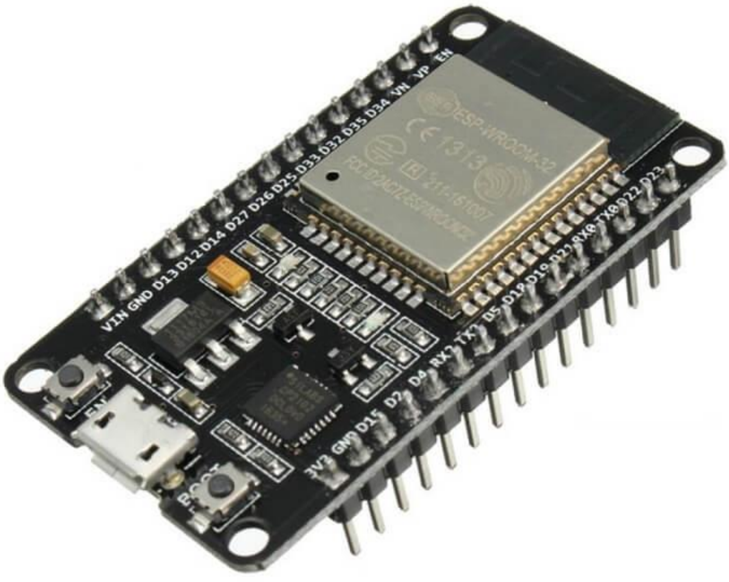

VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
ESP-WROOM-32
EN
BOOT

Figure 4.1: IMAGE OF ESP32

Features

- 11b/g/n （802.11n， speed up to 150Mbps）
- WIFI Frequency Range 2.4GHz ~ 2.5GHz
- Clock frequency adjustment range from 80 MHz to 240 MHz, support for RTOS
- Built-in 2-channel 12-bit high-precision ADC with up to 18 channels
- Support UART/GPIO/ADC/DAC/SDIO/SD card/PWM/I2C/I2S interface
- Support multiple sleep modes, ESP32 chip sleep current is less than 5 µA
- Embedded Lwip protocol stack
- Supports STA/AP/STA + AP operation mode
- Supports remote firmware upgrade (FOTA)
- General AT commands can be used quickly
- Support secondary development, integrated Windows, Linux development environment

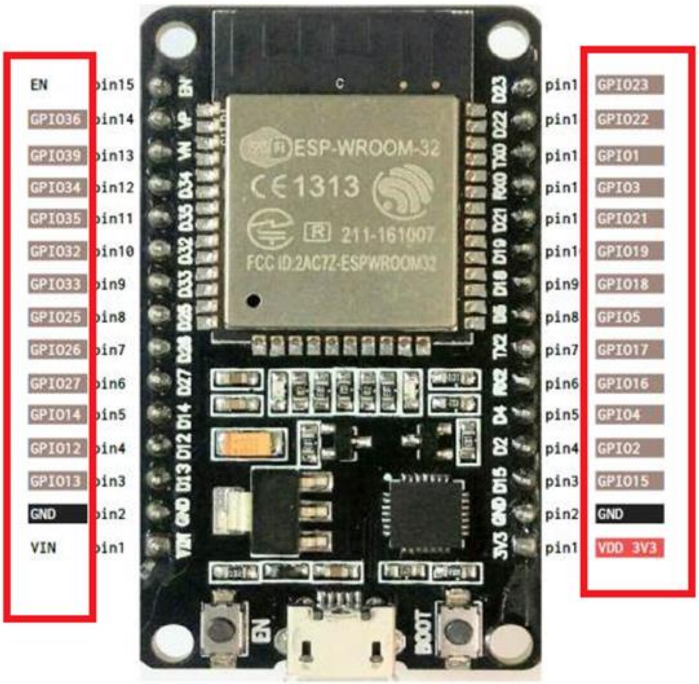


Figure 4.2: PIN DIAGRAM OF ESP32

ESP32 Functional Block Diagram

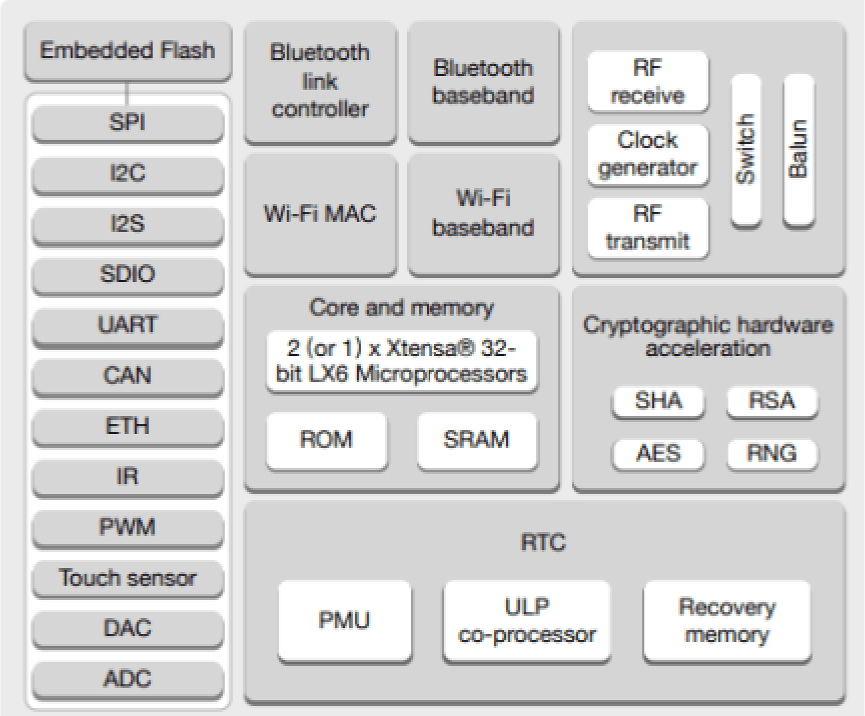


Figure 4.3: FUNCTIONAL BLOCK DIAGRAM

There are absolutely 39 computerized Pins on the ESP32 out of which 34 can be utilized as GPIO and the excess are input just pins. The gadget underpins 18-channels for 12-digit ADC and 2-channel for 8-bit DAC. IT likewise has 16 channels for PWM signal age and 10 GPIO pins upholds capacitive touch highlights. The ESP32 has multiplexing highlight, this empowers the developer to arrange any GPIO pin for PWM or other sequential correspondence however program. The ESP32 upholds 3 SPI Interface, 3 UART interface, 2 I2C interface, 2 I2S interface and furthermore supports CAN convention.

- 3 UART interface: The ESP32 underpins 3 UART interface for TTL correspondence. This would require 3 arrangements of Rx and Tx pins. All the 6 pins are programming configurable and thus any GPIO pin can be modified to be utilized for UART.

- External Interrupt: Again since the ESP32 underpins multiplexing any GPIO pin can be modified to be utilized as an intrude on pin.

- GPIO23 (MOSI), GPIO19 (MISO), GPIO18 (CLK) and GPIO5 (CS): These pins are utilized for SPI correspondence. ESP32 underpins two SPI, this is the primary set.

- GPIO13 (MOSI), GPIO12 (MISO), GPIO14 (CLK) and GPIO15 (CS): These pins are utilized for SPI correspondence. ESP32 upholds two SPI, this is the subsequent set.

- GPIO21 (SDA), GPIO22 (SCL): Used for IIC correspondence utilizing Wire library.

- Reset Pin: The reset pin for ESP32 is the Enable (EN) pin. Making this pin LOW, resets the microcontroller

Applications

1. Generic Low-power IOT Sensor Hub
2. Generic Low-power IOT Data Loggers
3. Cameras for Video Streaming
4. Over-the-top (OTT) Devices
5. Speech Recognition
6. Image Recognition
7. Mesh Network
8. Home Automation – Light control – Smart plugs – Smart door locks
9. Smart Building – Smart lighting – Energy monitoring
10. Industrial Automation – Industrial wireless control – Industrial robotics
11. Smart Agriculture – Smart greenhouses – Smart irrigation – Agriculture robotics
12. Audio Applications – Internet music players – Live streaming devices – Internet radio players – Audio headsets
13. Health Care Applications – Health monitoring – Baby monitors
14. Wi-Fi-enabled Toys – Remote control toys – Proximity sensing toys – Educational toys
15. Wearable Electronics – Smart watches – Smart bracelets • Retail & Catering Applications – POS machines – Service robots

5V ADAPTER

This AC to DC power supply will do 5V at 1A! They're switch mode power supplies which means the output is regulated to 5V . These have a standard USB 'A' connector for the output so you can power your Arduino, Raspberry Pi, etc. through a USB cable.

Figure 4.4: 5V ADAPTER

RESISTOR

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

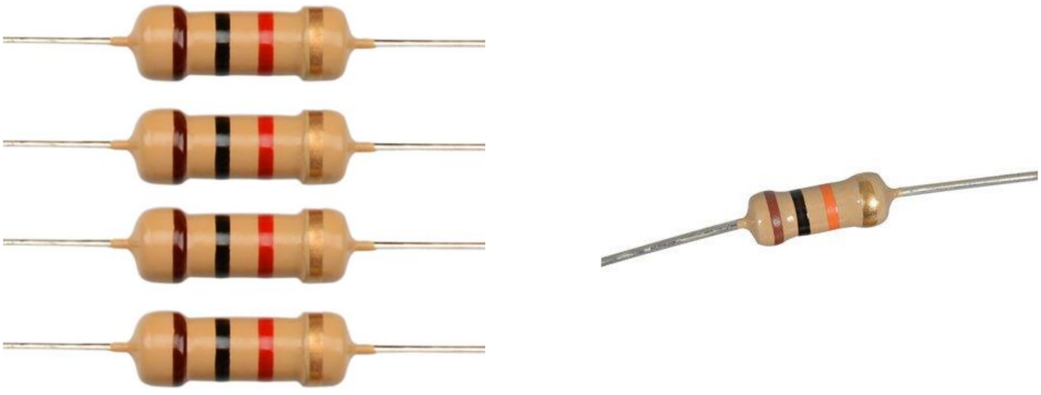


Figure 4.5: RESISTOR

DIODE (1N4007)

The most common function of a diode is to allow an electric current to pass in one direction (called the diode's forward direction), while blocking it in the opposite direction (the reverse direction). As such, the diode can be viewed as an electronic version of a check valve.

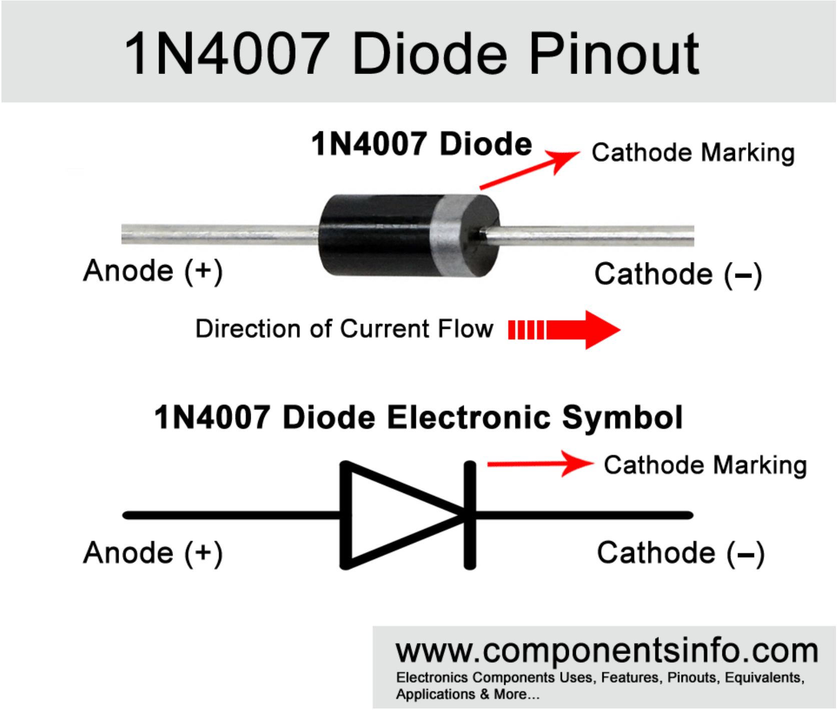

1N4007 Diode Pinout
1N4007 Diode
Cathode Marking
Anode (+)
Cathode (–)
Direction of Current Flow
1N4007 Diode Electronic Symbol
Cathode Marking
Anode (+)
Cathode (–)
www.componentsinfo.com
Electronics Components Uses, Features, Pinouts, Equivalents, Applications & More...

Figure 4.6: IMAGE OF 1N4007 DIODE

TRANSISTOR (BC547)

A semiconductor is fundamentally an electrically controlled switch. There is an information, yield, and a control line alluded to as the producer, authority, and base. At the point when the control line (base) is set off it will associate the producer and the gatherer simply like exchanging a switch. Since the force between the producer and authority can be higher than the base, semiconductors are regularly utilized as speakers.

The BC547 is a NPN semiconductor meaning when force is applied to the base (control pin) it will move from the gatherer to the producer. Normally NPN semiconductors are utilized to "switch ground" on a gadget, which means, they are put after the heap in a circuit.

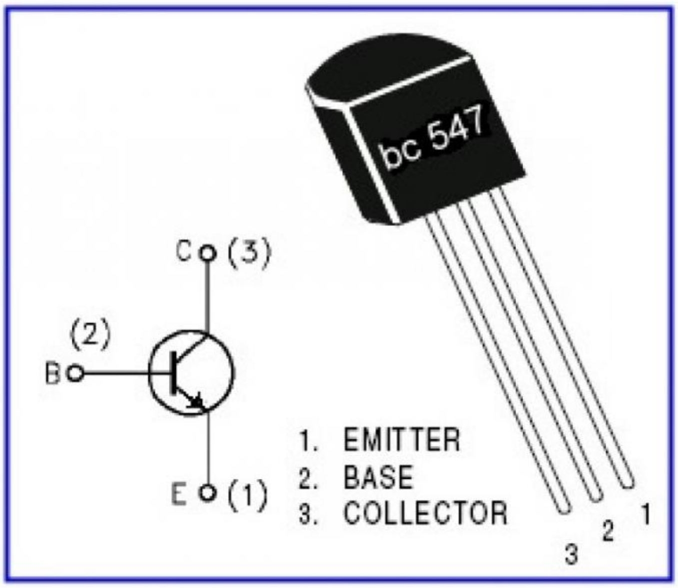


Figure 4.7: NPN TRANSISTOR

RELAY MODULE

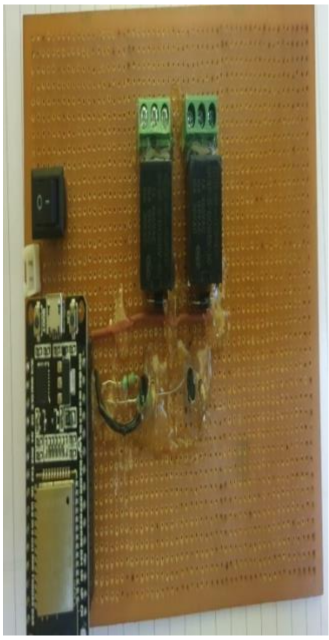
In this task transfer is a significant segment and which is utilized to control high voltage home apparatuses. We utilized 5V four channel transfer to control four machines. This is having four controlling information sources, Vcc, GND and four yields. The yields of transfer are associated with four apparatuses and these are constrained by utilizing control inputs which are associated with four computerized yield pins of ESP 32.

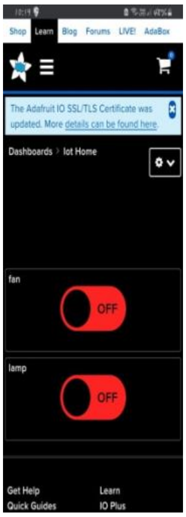
Figure 4.8: 5V RELAY

CHAPTER-5

RESULT:-

Connections of the system are made and the whole setup is embedded into a PCB to be resistant of all conditions .When we on or off the digital switch the bulb on and off with the help of the esp32 and the adafruit server .we can control these lights from anywhere around the world if the esp32 is connected to a wifi.

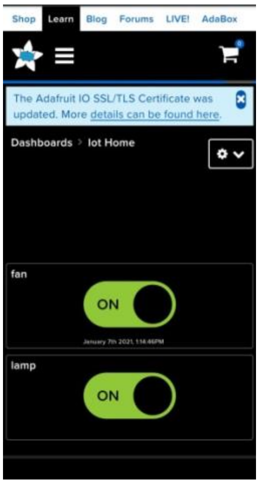




The Adafruit IO SSL/TLS Certificate was updated. More details can be found here.

Figure 5.1: FINAL OUTCOME

Figure 5.2: ADAFRUIT.IO



The Adafruit IO SSL/TLS Certificate was updated. More details can be found here.

Figure 5.2:adafruit.io when switch is on

Advantages

- Easier to Lock and Unlock the Doors.
- Save Energy.
- Know About Maintenance and Service.
- Customize as Per Your Convenience.
- Increase Property Value.
- Security.
- Can be accessible from anywhere around the world.

LIMITATIONS

There are not many limitations for this IOT based domain. Some of them are,

- If this system has to implanted on a larger scale it will be expensive.
- There are possibilities that the hackers might get into the system and the system will get corrupted.

Applications

1. Managing all of your home devices from one place. The accommodation factor here is colossal. Having the option to keep the entirety of the innovation in your home associated through one interface is an enormous advance forward for innovation and home administration. Hypothetically, you'll should simply figure out how to utilize one application on your Smartphone and tablet, and you'll have the option to take advantage of incalculable capacities and gadgets all through your home. This scales route back the expectation to absorb information for new clients, makes it simpler to get to the usefulness you genuinely need for your home.

2. Flexibility for new devices and appliances. Smart home frameworks will in general be brilliantly adaptable with regards to the convenience of new gadgets and apparatuses and other innovation. Regardless of how cutting edge your machines appear today, there will be more up to date, more noteworthy models created over the long haul. Past that, you'll likely add to your set-up of gadgets as you supplant the more seasoned ones or find new innovation to go with your indoor and outside spaces. Having the option to incorporate these newcomers flawlessly will make your occupation as a mortgage holder a lot simpler, and permit you to continue moving up to the most recent way of life innovation.

3. Maximizing home security. At the point when you consolidate security and observation highlights in your shrewd home organization, your home security can soar. There are huge loads of alternatives here a couple dozen of which are as of now being investigated. For instance, home mechanization frameworks can associate movement identifiers, reconnaissance cameras, computerized entryway locks, and other substantial safety efforts all through your home so you can initiate them from one cell phone prior to going to bed. You can likewise decide to get security alarms on your different gadgets relying upon the hour of day an alarm goes off, and screen exercises progressively whether you're in the house or most of the way around the world.

4. Remote control of home functions. Don't underestimate the power of being able to control your home's functions from a distance. On an exceptionally hot day, you can order your house to become cooler in just enough time before you get home from work. If you're in a hurry to get dinner started but you're still at the store, you can have your oven start to preheat while you're still on your way home. You can even check to see if you left the lights on, who is at your front door, or make sure you turned off all your media while you're away.

5. Increased energy efficiency. Contingent upon how you utilize your brilliant home innovation, it's conceivable to make your space more energy-effective. For instance, you can have more exact authority over the warming and cooling of your home with a programmable savvy indoor regulator that learns your timetable and temperature inclinations, and afterward proposes the best energy effective settings for the duration of the day. Lights and mechanized shades can be programmed to change to a night mode as the sun sets, or lights can turn on and off naturally when you go into or leave the room, so you never need to stress over squandering energy.

6. Improved appliance functionality. Brilliant homes can likewise help you run your machines better. A shrewd TV will help you find better applications and channels to find your #1 programming. A shrewd stove will help you with cooking your chicken flawlessly - while never agonizing over overcooking or half-cooking it. A wisely planned home theater and sound framework can make dealing with your film and music assortment easy while engaging visitors. At last, associating your apparatuses and different frameworks with computerization innovation will improve your machine adequacy and generally speaking make your home life substantially simpler and agreeable!

7. Home management insights. There's also something to be said for your ability to tap into insights on how your home operates. You can monitor how often you watch TV (and what you watch), what kind of meals you cook in your oven, the type of foods you keep in your refrigerator, and your energy consumption habits over time.

CHAPTER-6

FUTURE SCOPE

In our project we have made a home automation system using MQTT and ESP32. To test the system Android MQTT client pre available application has used. But it is not user friendly. As a future work one can develop an android or ios application with good user interface. It helps in controlling home appliances very easily.

IOT home automation is slowly but steadily becoming a part of daily lives around the world. In fact, it is believed that the global market for smart home automation will reach $40 billion by 2020. This shouldn't be surprising when you consider the convenience and ease that smart home devices offer.

CONCLUSION

We presented a system which can monitor and control multi devices at home using IOT. This proposed system improves the performance of various devices at home by controlling automatically and remotely. The system works in three phases. In first phase the system monitors if the lights are on or off and upload the data to cloud (things speak) and mobile server adafruit. In second phase the system we can control the lights using the adafruit server. In third phase user can control home appliances like fan, light etc using mobile app. The system in build using low cost embedded microcontroller with Wifi module ESP32.The developed system cost is low, simple to operate and is easily embedded with home appliances.

REFERENCES

- S. Das and D. J. Cook, Smart Home Environments: A Paradigm Based on Learning and Prediction, Wireless Mobile and Sensor Networks, Wiley, 2004.
- "Best Home Automation System - Consumer Reports". www.consumerreports.org. Recovered 2016-02-14.
- https://arduino-info.wikispaces.com/BlueTooth-HC05-HC06-Modules-How-To.html
- https://www.electronicsforu.com/electronics-projects/iot-automation-home-appliances-esp-32

APPENDIX

PROGRAM CODE

```
/***************************************************
  Adafruit MQTT Library ESP8266 Example


  Must use ESP8266 Arduino from:
    https://github.com/esp8266/Arduino


  Works great with Adafruit's Huzzah ESP board & Feather
  ----> https://www.adafruit.com/product/2471
  ----> https://www.adafruit.com/products/2821


  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!


  Written by Tony DiCola for Adafruit Industries.

  MIT license, all text above must be included in any redistribution

 ****************************************************/

#include <WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"


#define relay1 13

#define relay2 12


/************************* WiFi Access Point *********************************/


#define WLAN_SSID       "rupee"  //wifi name

#define WLAN_PASS       "ammre9686"  //wifi password


/************************* Adafruit.io Set aup *********************************/
```

```
#define AIO_SERVER      "io.adafruit.com"

#define AIO_SERVERPORT  1883                   // use 8883 for
SSL

#define AIO_USERNAME    "Rupendran"            //adafruit
username

#define AIO_KEY
"aio_EOTs86b7GIRMWuaIHvrWbYIWtLvK"  //adafruit key


/************ Global State (you don't need to change this!)
******************/


// Create an ESP8266 WiFiClient class to connect to the
MQTT server.

WiFiClient client;

// or... use WiFiFlientSecure for SSL

//WiFiClientSecure 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);
```

```
/****************************** Feeds ***************************************/

// Setup a feed called 'photocell' for publishing.
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
//Adafruit_MQTT_Publish photocell = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/photocell");

// Setup a feed called 'onoff' for subscribing to changes.
Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/fan");  //feeds name
Adafruit_MQTT_Subscribe Light2 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/lamp");   // feeds name

/*************************** Sketch Code ************************************/

// Bug workaround for Arduino 1.6.6, it seems to need a function declaration
// for some reason (only affects ESP8266, likely an
arduino-builder bug).
void MQTT_connect();


void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
    digitalWrite(relay1,HIGH);
   digitalWrite(relay2,HIGH);




  Serial.println(F("Adafruit MQTT demo"));


  // Connect to WiFi access point.
  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);
  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.println("WiFi connected");
  Serial.println("IP address: ");
Serial.println(WiFi.localIP());

  // Setup MQTT subscription for onoff feed.
  mqtt.subscribe(&Light1);
  mqtt.subscribe(&Light2);

}

uint32_t x=0;
```

```
void loop() {
  // Ensure the connection to the MQTT server is alive
(this will make the first
  // connection and automatically reconnect when
disconnected).  See the MQTT_connect
  // function definition further below.
  MQTT_connect();


  // this is our 'wait for incoming subscription packets'
busy subloop
  // try to spend your time here


  Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(5000))) {
    if (subscription == &Light1)
    {
      Serial.println((char*) Light1.lastread);
    // uint16_t Light1_State = atoi((char *)Light1.lastread);
    String Light1_State =(char *)Light1.lastread;
      if (Light1_State=="OFF")
      {
        digitalWrite(relay1, HIGH);
      }
      if(Light1_State=="ON")
      {
        digitalWrite(relay1, LOW);
       }
      }
    if (subscription == &Light2)
    {
      Serial.print(F("Got: "));
      Serial.println((char *)Light2.lastread);
      //uint16_t Light2_State = atoi((char *)Light2.lastread);
      String Light2_State =(char *)Light2.lastread;


    if (Light2_State=="OFF")
      {
        digitalWrite(relay2, HIGH);
      }
      if(Light2_State=="ON")
    {
      digitalWrite(relay2, LOW);
    }
    }



  }



  // ping the server to keep the mqtt connection alive
  // NOT required if you are publishing once every KEEPALIVE seconds
  /*
  if(! mqtt.ping()) {
    mqtt.disconnect();
  }
  */
}
```

```
// Function to connect and reconnect as necessary to the MQTT server.
// Should be called in the loop function and it will take care if connecting.
void MQTT_connect() {
  int8_t ret;

  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }

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

  uint8_t retries = 3;


  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
       Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);  // wait 5 seconds
    retries--;
    if (retries == 0) {
      // basically die and wait for WDT to reset me
      while (1);
    }
  }
  Serial.println("MQTT Connected!");

}
```