

**A Mini Project Report
On Internet Of Things(IOT)**

DOOR LOCK USING GOOGLE ASSISTANT



By

ROHIT (16951A05M4)

**Department of Computer Science Engineering
Institute of Aeronautical Engineering (Autonomous)
(Affiliated to JNTUH)
Dundigal, Hyderabad**

2018-19

TABLE OF CONTENTS

Title	Page No
Abstract	
1. App title and Description	
1.1 Problem definition	
1.2 Motivation	
1.3 Objectives	
1.4 Scope	
2. Flowchart	
3. Pseudo algorithm/code	

ABSTRACT

Security is the main concern in our everyday life. Each and every individual needs to feel secure. An access control for doors forms an essential pattern in our security pattern. Doors which are locked using conventional locks are not safe now-a-days as they could be easily broken.

Hence there should be an other alternative where we could safe guard our home security pattern. Door Lock using Google assistant is one such trial made where we could safe guard the home lock. It uses password to safeguard the lock.

Password based lock system allows only the approved persons to access restricted areas. This system is fully controlled by aurdino. Password can be entered using keypad. If password entered matches with the stored password in Aurdino the door gets opened.

This programmed password based bolt framework will give client more secure and minimal effort method for locking-opening framework. The security door lock automation promises a bold step to the future where mechanical door locks will be substituted by electronic door locks.

1. Title And Description

Door lock using Google assistant is a very secure and safe way of locking the door. We connect to the nodeMCU through google assistant. A password is first set. To close or open the door we should first connect through the Google assistant by entering the password. If the password matches the door locks opens and similarly closing is also done.

1.1 Problem definition

Node MCU is an open source IOT platform. It consists of inbuilt wifi module associated with it. It is one of the important component used in this Door Lock System. The main purpose of this Door Lock System is to provide smart security to the house as now-a-days these mechanical ie hand opening Door lock Systems became insecure in this busy world. This is secure because only authorized persons can access/ open / close the door.

1.2 Motivation

There are a few smart home devices types that your Google Assistant can control by configuring. Lights, plugs and switches and thermostats were some of the first, but recently we have seen the addition of dryers, washers, dishwashers, vacuums, fridges. Similarly for the safeguarding the house, the smart door lock using Google Assistant.

Now-a-days in this fast running and busy world, we come across many theft cases where thieves just break the door lock with their own tricky techniques. People just lock the door with some mechanical locks which are not at all safe and secure.

Hence there should be some technique where locking of door using a bolt which is inside the house. That door lock should be controlled by some software technique. It can be a wireless connection like Bluetooth ,Google assistant etc. Here is the trial made which can control thr door lock using Google Assistant which is more secure.

1.3 Motivation

1. To develop a system so the in increases the security for the home.
2. It prevents the unauthorized users to open/ close the door.

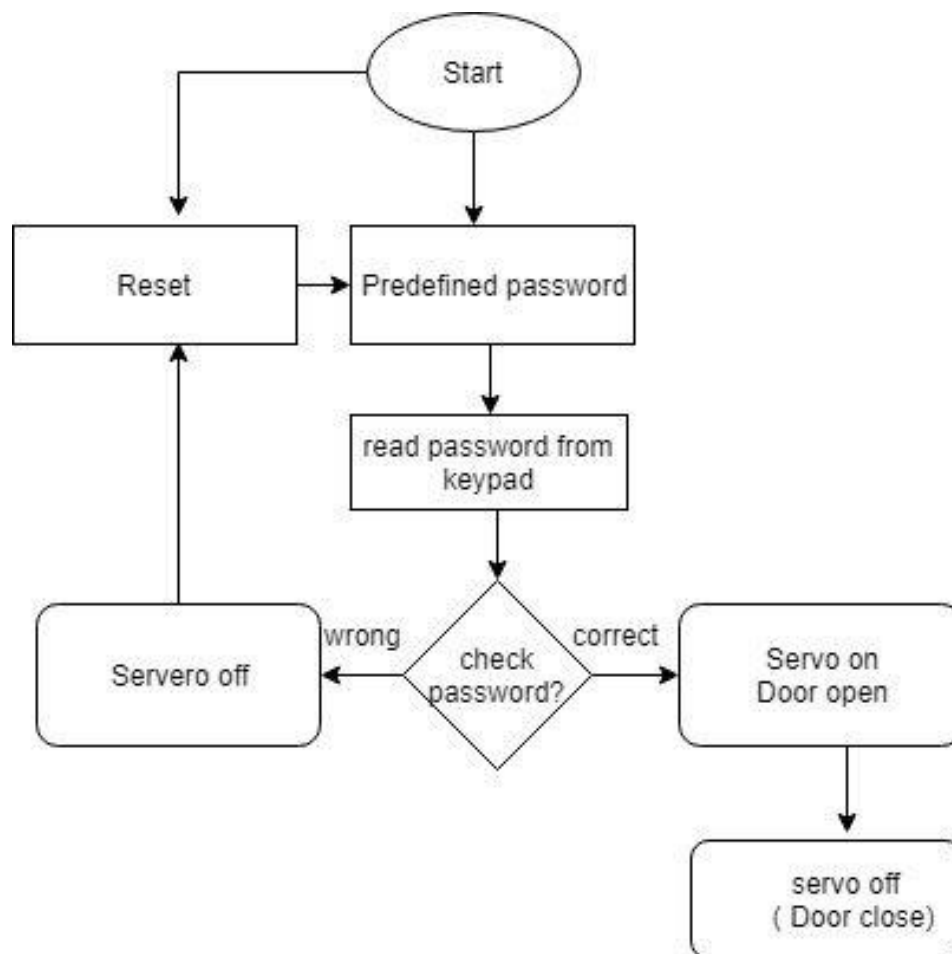
1.4 Scope

Only the authorized users can access the door. Only with a single password we can open/ close the door. If we forgot the password the password can be reset only after the verification of some security checks. We can access the door lock only through certain email ids and password which may sometimes leads to failure.

This works well only when the phone WiFi and node MCU WiFi module connected to the same IP address.

2. FLOWCHART

This flowchart gives the detailed flow of how this Door Lock System works.



3. PSEUDO ALGORITHM/IMPLEMENTATION/CODE

```
#include <ESP8266WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"


#define Relay1      D1

#define Relay2      D2

#define Relay3      D3

#define Relay4      D4


#define WLAN_SSID   " ---- "           // Your SSID
#define WLAN_PASS   " ---- "           // Your password


/***** Adafruit.io Setup *****/

#define AIO_SERVER   "io.adafruit.com" //Adafruit Server

#define AIO_SERVERPORT 1883

#define AIO_USERNAME " ---- "           // Username

#define AIO_KEY      " ---- "           // Auth Key


//WIFI CLIENT
```


WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Subscribe Light1 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME"/feeds/Relay1");

Adafruit_MQTT_Subscribe Light2 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Relay2");

Adafruit_MQTT_Subscribe Light3 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Relay3");

Adafruit_MQTT_Subscribe Light4 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/Relay4");

void MQTT_connect();

void setup() {

 Serial.begin(115200);

 pinMode(Relay1, OUTPUT);

 pinMode(Relay2, OUTPUT);

 pinMode(Relay3, OUTPUT);

 pinMode(Relay4, OUTPUT);

 // 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());
```

```
mqtt.subscribe(&Light1);  
mqtt.subscribe(&Light3);  
mqtt.subscribe(&Light2);  
mqtt.subscribe(&Light4);  
}
```

```
void loop() {
```

```
    MQTT_connect();
```

```
    Adafruit_MQTT_Subscribe *subscription;  
    while ((subscription = mqtt.readSubscription(20000))) {  
        if (subscription == &Light1) {  
            Serial.print(F("Got: "));  
            Serial.println((char *)Light1.lastread);
```

```
int Light1_State = atoi((char *)Light1.lastread);  
digitalWrite(Relay1, Light1_State);  
  
}  
  
if (subscription == &Light2) {  
    Serial.print(F("Got: "));  
    Serial.println((char *)Light2.lastread);  
    int Light2_State = atoi((char *)Light2.lastread);  
    digitalWrite(Relay2, Light2_State);  
}  
  
if (subscription == &Light3) {  
    Serial.print(F("Got: "));  
    Serial.println((char *)Light3.lastread);  
    int Light3_State = atoi((char *)Light3.lastread);  
    digitalWrite(Relay3, Light3_State);  
}  
  
if (subscription == &Light4) {  
    Serial.print(F("Got: "));  
    Serial.println((char *)Light4.lastread);  
    int Light4_State = atoi((char *)Light4.lastread);  
    digitalWrite(Relay4, Light4_State);  
  
}  
}
```

```
}
```

```
void MQTT_connect() {
```

```
    int8_t ret;
```

```
    if (mqtt.connected()) {
```

```
        return;
```

```
    }
```

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

```
    uint8_t retries = 3;
```

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

```
        Serial.println(mqtt.connectErrorString(ret));
```

```
        Serial.println("Retrying MQTT connection in 5 seconds...");
```

```
        mqtt.disconnect();
```

```
        delay(5000);
```

```
        retries--;
```

```
        if (retries == 0) {
```

```
            while (1);
```

```
        }
```

```
    }
```

```
    Serial.println("MQTT Connected!");
```

}



ReplyForward