

ASSIGNMENT-1

Er.Perumal Manimekalai College of

Engineering,Hosur

NAME : RETHESH R

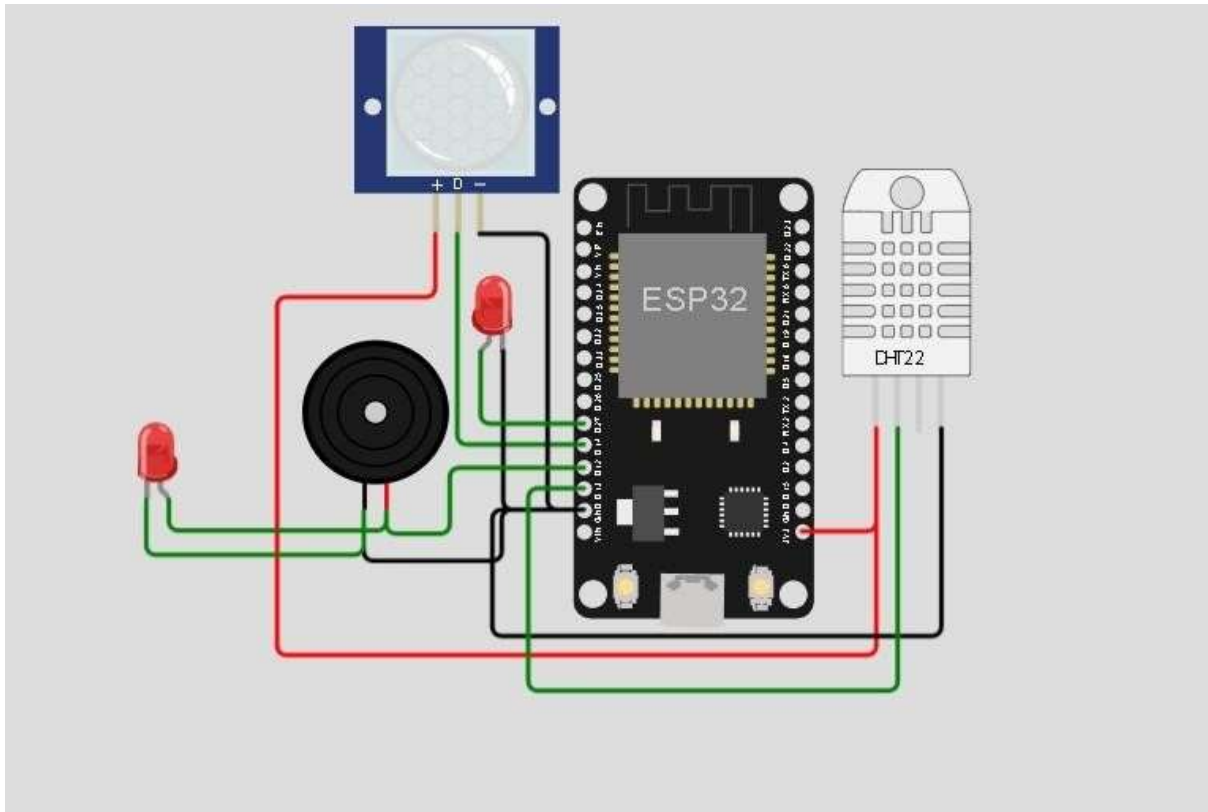
CLASS : 3 YEAR ECE

SUBJECT : IBM

REGISTER NO : 610820106043

Build a smart home in wokwi with minimum 2
sensors, Led, buzzer

DESIGN PART



CODING PART

sketch.ino

```
#include <FirebaseESP32.h> //call library firebase esp32
```

```
#include <WiFi.h> //call library wifi
```

```
#include "DHT.h" //call library dht
```

```
#define FIREBASE_HOST "https://smart-home-7b44f-default-  
rtdb.firebaseio.com" //declare variable as a firebase host
```

```
#define WIFI_SSID "Wokwi-GUEST" //declare variable as a  
wifi_ssid
```

```
#define WIFI_PASSWORD "" //declare variable as a wifi  
password
```

```
#define FIREBASE_Authorization_key  
"oTRPje2my9ROIP2dIhJMzEd66sd3arqmtC4WG6cv"  
//declare variable as a firebase auth key  
FirebaseData firebaseData; //Define the FirebaseData object as  
firebaseData  
FirebaseJson json; //Define the FirebaseJson as json
```

```
int led = 27; //declare variable as a integer String  
message = ""; //declare variable as a string  
String message1 = ""; //declare variable as a string  
String s_pir = ""; //declare variable as a string  
String b = ""; //declare variable as a string  
String c = ""; //declare variable as a string bool  
a = 0; //declare variable as a boolean #define  
pir 14 //declare variable as a pin pir  
#define buzzer 12 //declare variable as a pin buzzer  
#define DHTPIN 13 //declare variable as a pin dht  
#define DHTTYPE DHT22 //declare variable as a dht11  
DHT dht(DHTPIN, DHTTYPE); //connect esp32 to dht
```

```
void setup() {  
  pinMode(led, OUTPUT); //led as an output pinMode(pir,  
  INPUT); //pir as an input  
  pinMode(buzzer, OUTPUT); //buzzer as an output  
  Serial.begin(115200); //open serial connection  
  dht.begin(); //initiate the connection with the dht  
  WiFi.begin (WIFI_SSID, WIFI_PASSWORD); //initiate the  
  connection with ap  
  Serial.println("Connecting..."); //serial print monitor  
  connecting
```

```

//if wifi not connect serial print monitor not connected while
(WiFi.status() != WL_CONNECTED) {
Serial.println("NOT CONNECTED");
  delay(300);
} b = WIFI_SSID; //define variable b like variable
wifi_ssid
Serial.println();
Serial.print("IP Address: "); //serial print monitor ip address:
Serial.println(WiFi.localIP()); //serial print monitor wifi local ip
Serial.println();

```

```

Firebase.begin(FIREBASE_HOST,FIREBASE_Authorization_
key); //initiate the connection with firebase }

```

```

void loop() {
  bool state_pir = digitalRead(pir); //declare variable state_pir as
a digital read pir sensor (boolean)

```

```

//get string firebase data led
if (Firebase.getString(firebaseData,
"/ESP32APP_LED/LED"))
{
  String message_fb = firebaseData.stringData();
  if (message_fb != message) {
    message = message_fb;
    if(message.indexOf("LED ON") != -1){digitalWrite(led,
HIGH);} if(message.indexOf("LED OFF") != -
1){digitalWrite(led,
LOW);}
  }
}

```

```
}
```

```
//get string firebase data set status motion sensor if  
(Firebase.getString(firebaseData,  
"/ESP32APP_MOTION/ALARM_SET")) {  
  String message1_fb = firebaseData.stringData();  
  if (message1_fb != message1) {  
    message1 = message1_fb;  
    if(message1.indexOf("ALARM SET ON") != -  
1){ a=1; s_pir = "READY";  
    } if(message1.indexOf("ALARM SET OFF") != -  
1){  
    digitalWrite(buzzer, LOW);  
    s_pir = "OFF"; a=0;  
    }  
  }  
}
```

```
//condition pir sensor if detect object if  
(state_pir==1 && a==1){  
digitalWrite(buzzer, HIGH);  
s_pir = "THIEFFFF";  
}
```

```
//declare variable hum as a dht read humidity(%) and  
temperature (c)  
float hum = dht.readHumidity(); float  
temp = dht.readTemperature();
```

```
//condition dht sensor if not connect with esp32 if
(isnan(hum) || isnan(temp) ){
  Serial.println(F("Failed to read from DHT sensor!"));
  c = "FAILED";
} else
{
  c= "READY";
}
```

```
Serial.print("Temperature: "); //serial print monitor
temperature:
Serial.print(temp); //serial print monitor value temperature
Serial.print("°C"); //serial print monitor °C
Serial.print(" Humidity: "); //serial print monitor humidity:
Serial.print(hum); //serial print monitor value humidity
Serial.print("%"); //serial print monitor %
Serial.print(" ");
Serial.print(message); //serial print monitor value message
Serial.print(" ");
Serial.print(message1); //serial print monitor value message1
Serial.print(" ");
Serial.print(s_pir); //serial print monitor value s_pir
Serial.println();
Firebase.setFloat(firebaseData,
"/ESP32APP_DHT11/TEMPERATURE", temp); //firebase
data set float value temperature
Firebase.setFloat(firebaseData,
"/ESP32APP_DHT11/HUMIDITY", hum); //firebase data set
float value humidity
Firebase.setString(firebaseData,
```

```

"/ESP32APP_DHT11/STATUS", c); //firebase data set string
value c
  Firebase.setString(firebaseData,
"/ESP32APP_MOTION/BUZZER", s_pir); //firebase data set
string value s_pir
  Firebase.setString(firebaseData,
"/ESP32APP_WIFI/STATUS", b); //firebase data set string
value b
  delay(200); //delay 200 ms
}

```

```

#include <FirebaseESP32.h> //call library firebase esp32
#include <WiFi.h> //call library wifi
#include "DHT.h" //call library dht

```

```

#define FIREBASE_HOST "https://smart-home-7b44f-default-
rtadb.firebaseio.com" //declare variable as a firebase host
#define WIFI_SSID "Wokwi-GUEST" //declare variable as a
wifi_ssid
#define WIFI_PASSWORD "" //declare variable as a wifi
password
#define FIREBASE_Authorization_key
"oTRPje2my9ROlP2dIhJMzEd66sd3arqmtC4WG6cv"
//declare variable as a firebase auth key
FirebaseData firebaseData; //Define the FirebaseData object as
firebaseData
FirebaseJson json; //Define the FirebaseJson as json

int led = 27; //declare variable as a integer String
message = ""; //declare variable as a string

```

```

String message1 = ""; //declare variable as a string
String s_pir = ""; //declare variable as a string
String b = ""; //declare variable as a string
String c = ""; //declare variable as a string bool
a = 0; //declare variable as a boolean #define
pir 14 //declare variable as a pin pir
#define buzzer 12 //declare variable as a pin buzzer
#define DHTPIN 13 //declare variable as a pin dht
#define DHTTYPE DHT22 //declare variable as a dht11
DHT dht(DHTPIN, DHTTYPE); //connect esp32 to dht

void setup() {
  pinMode(led, OUTPUT); //led as an output pinMode(pir,
  INPUT); //pir as an input
  pinMode(buzzer, OUTPUT); //buzzer as an output
  Serial.begin(115200); //open serial connection
  dht.begin(); //initiate the connection with the dht
  WiFi.begin (WIFI_SSID, WIFI_PASSWORD); //initiate the
  connection with ap
  Serial.println("Connecting..."); //serial print monitor
  connecting
  //if wifi not connect serial print monitor not connected
  while (WiFi.status() != WL_CONNECTED) {
    Serial.println("NOT CONNECTED");
    delay(300);
  } b = WIFI_SSID; //define variable b like variable
  wifi_ssid Serial.println();
  Serial.print("IP Address: "); //serial print monitor ip address:
  Serial.println(WiFi.localIP()); //serial print monitor wifi local ip
  Serial.println();

```



```
Firebase.begin(FIREBASE_HOST,FIREBASE_Authorization_
key); //initate the connection with firebase }
```

```
void loop() {
  bool state_pir = digitalRead(pir); //declare variable state_pir as
a digital read pir sensor (boolean)
```

```
  //get string firebase data led if
  (Firebase.getString(firebaseData, "/ESP32APP_LED/LED"))
  {
    String message_fb = firebaseData.stringData();
    if (message_fb != message) {
      message = message_fb;
      if(message.indexOf("LED ON") != -1){digitalWrite(led,
HIGH);} if(message.indexOf("LED OFF") != -
1){digitalWrite(led,
LOW);}
    }
  }
}
```

```
  //get string firebase data set status motion sensor if
  (Firebase.getString(firebaseData,
"/ESP32APP_MOTION/ALARM_SET")) {
    String message1_fb = firebaseData.stringData(); if
    (message1_fb != message1) {
      message1 = message1_fb;
      if(message1.indexOf("ALARM SET ON") != -1){
        a=1;
        s_pir = "READY";
```

```

    } if(message1.indexOf("ALARM SET OFF") != -
1){
    digitalWrite(buzzer, LOW);
    s_pir = "OFF"; a=0;
    }
}
}

```

```

//condition pir sensor if detect object if
(state_pir==1 && a==1){
digitalWrite(buzzer, HIGH);
s_pir = "THIEFFFF";
}

```

```

//declare variable hum as a dht read humidity(%) and
temperature (c)
float hum = dht.readHumidity(); float
temp = dht.readTemperature();

```

```

//condition dht sensor if not connect with esp32 if
(isnan(hum) || isnan(temp) ){
Serial.println(F("Failed to read from DHT sensor!"));
c = "FAILED";
} else
{
c= "READY";
}
Serial.print("Temperature: "); //serial print monitor
temperature:
Serial.print(temp); //serial print monitor value temperature

```

```
Serial.print("°C"); //serial print monitor °C
Serial.print(" Humidity: "); //serial print monitor humidity:
Serial.print(hum); //serial print monitor value humidity
Serial.print("%"); //serial print monitor %
Serial.print(" ");
Serial.print(message); //serial print monitor value message
Serial.print(" ");
Serial.print(message1); //serial print monitor value message1
Serial.print(" ");
Serial.print(s_pir); //serial print monitor value s_pir
Serial.println();
```

```
    Firebase.setFloat(firebaseData,
"/ESP32APP_DHT11/TEMPERATURE", temp); //firebase
data set float value temperature
    Firebase.setFloat(firebaseData,
"/ESP32APP_DHT11/HUMIDITY", hum); //firebase data set
float value humidity
    Firebase.setString(firebaseData,
"/ESP32APP_DHT11/STATUS", c); //firebase data set string
value c
    Firebase.setString(firebaseData,
"/ESP32APP_MOTION/BUZZER", s_pir); //firebase data set
string value s_pir
    Firebase.setString(firebaseData,
"/ESP32APP_WIFI/STATUS", b); //firebase data set string
value b
    delay(200); //delay 200 ms
}
```

diagram.json

```
{
  "version": 1,
  "author": "chandra kirana",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 0,
      "left": 0, "attrs": {} },
    {
      "type": "wokwi-dht22",
      "id": "dht1",
      "top": -2.08,
      "left": 119.41,
      "attrs": { "temperature": "30.1", "humidity": "29" }
    },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 36.62,
      "left": -55.84,
      "attrs": { "color": "red", "flip": "1" }
    },
    {
      "type": "wokwi-pir-motion-sensor",
      "id": "pir1",
```

```
"top": -65.32,
"left": -97.06,
"attrs": {} },
{
  "type": "wokwi-buzzer",
  "id": "bz1",
  "top": 65.51,
  "left": -119.56,
  "attrs": { "volume": "0.1" }
},
{
  "type": "wokwi-led",
  "id": "led2",
  "top": 102.36,
  "left": -203.89,
  "attrs": { "color": "red", "flip": "" }
}
],
"connections": [
  [ "esp:TX0", "$serialMonitor:RX", "", [] ],
  [ "esp:RX0", "$serialMonitor:TX", "", [] ],
  [ "dht1:VCC", "esp:3V3", "red", [ "v0" ] ],
  [ "pir1:OUT", "esp:D14", "green", [ "v0" ] ],
  [ "pir1:GND", "esp:GND.2", "black", [ "h30.28", "v122.32" ]
],
  [ "esp:GND.2", "led1:C", "black", [ "h0" ] ],
  [ "pir1:VCC", "dht1:VCC", "red", [ "v27.99", "h-70.08",
"v158.75", "h261.03" ] ],
  [ "bz1:2", "esp:D12", "green", [ "v10.06", "h28.07", "v-
29.68" ] ],
```

```
[ "esp:GND.2", "bz1:1", "black", [ "h-36.45", "v22.63", "h-58.84" ] ],  
[ "dht1:GND", "esp:GND.2", "black", [ "v92.14", "h-198.86", "v-55.96" ] ],  
[ "led1:A", "esp:D27", "green", [ "v0" ] ],  
[ "dht1:SDA", "esp:D13", "green", [ "v116.37", "h-163.6", "v-91.7" ] ],  
[ "bz1:1", "led2:C", "green", [ "v19.44", "h-99.3" ] ],  
[ "led2:A", "bz1:2", "green", [ "v13.82", "h100.85" ] ]  
],  
"dependencies": { } }
```