SMART PARKING PHASE -04

DEVELOPMENT PART-II

Connecting the Hardware in Wokwi

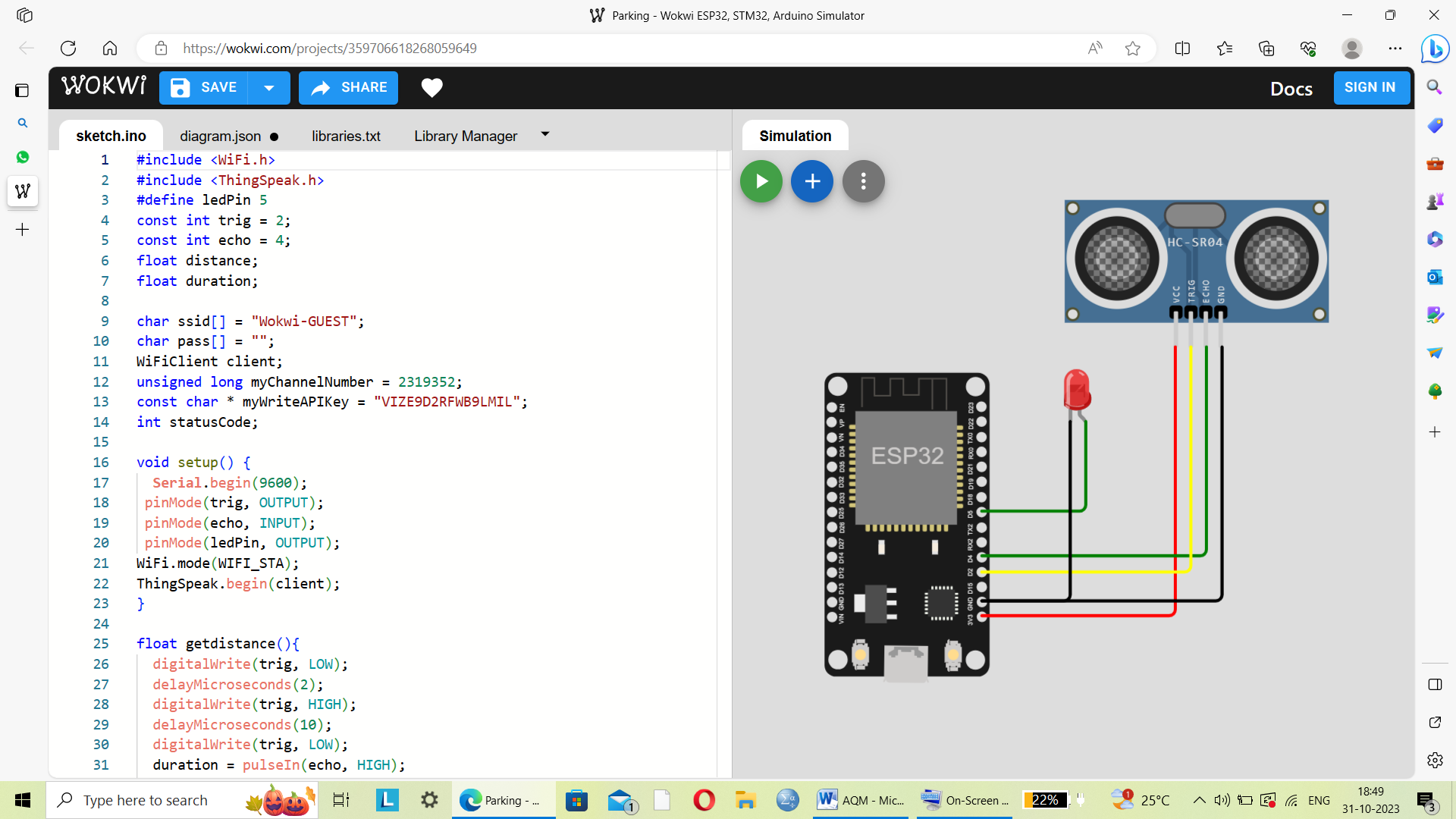
1. In the Wokwi simulator, you can add components like the ESP32,

2. Connect the components using virtual jumper wires.

3. Ultrasonic distance senson connect to the pin on the ESP32.

4.let light connect to the esp32 .

HARDWARE CONNECTION



SOURCE CODE:

#include <WiFi.h>

#include <ThingSpeak.h>

#define ledPin 5

const int trig = 2;

const int echo = 4;

float distance;

float duration;

char ssid[] = "Wokwi-GUEST";

char pass[] = "";

WiFiClient client;

unsigned long myChannelNumber = 2319352;

const char \* myWriteAPIKey = "VIZE9D2RFWB9LMIL";

int statusCode;

void setup() {

Serial.begin(9600);

pinMode(trig, OUTPUT);

pinMode(echo, INPUT);

pinMode(ledPin, OUTPUT);

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client);

}

float getdistance(){

digitalWrite(trig, LOW);

delayMicroseconds(2);

digitalWrite(trig, HIGH);

delayMicroseconds(10);

digitalWrite(trig, LOW);

duration = pulseIn(echo, HIGH);

return {duration\*0.034/2};

}

void loop() {

connectToCloud();

writeData();

distance = getdistance();

if (distance<200){

digitalWrite(ledPin, HIGH);

Serial.println("Parking Space Occupied");

}

else{

Serial.println("Parking Space Available");

digitalWrite(ledPin, LOW);

}

delay(5000);

}

void connectToCloud(){

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

Serial.print("Attempting to connect");

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

WiFi.begin(ssid, pass);

for(int i=0;i<5;i++) {

Serial.print(".");

delay(5000);

}

}

Serial.println("\nConnected.");

}

}

void writeData(){

distance = getdistance();

ThingSpeak.setField(1, distance);

statusCode = ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);

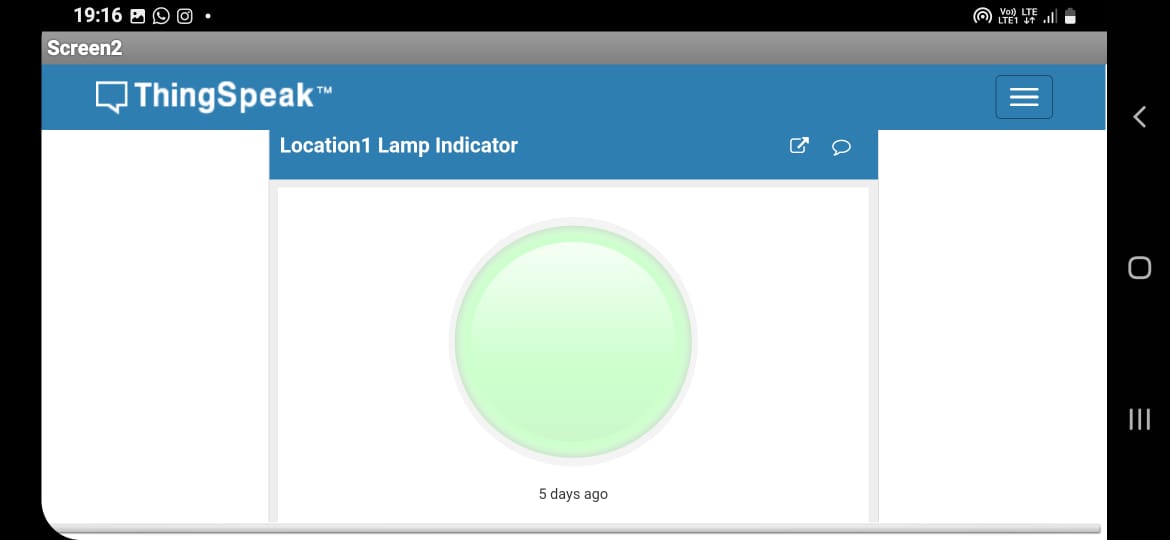
if(statusCode == 200) //successful writing code

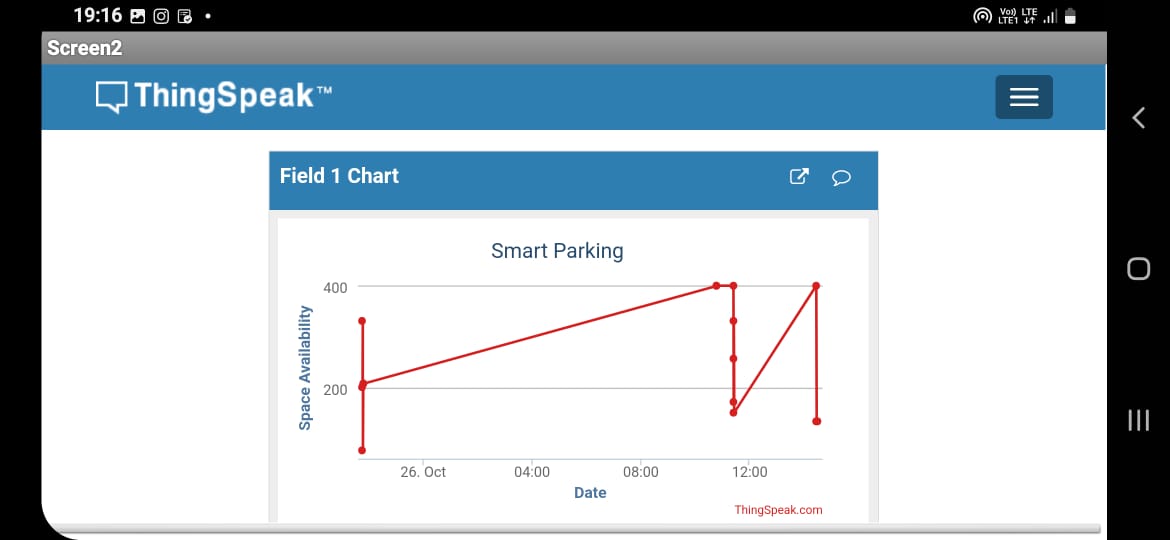
Serial.println("Channel update successful.");

else

Serial.println("Problem Writing data. HTTP error code :" +

String(statusCode));

}



DEVELOP APK USING MIT APP INVENTOR

For every simulation in the wokwi platform the data can be update into personal channel created in the Thingspeak. We can use the data to know the difference level daily update and also live stream the data into the SM Interface using MIT APP INVENTOR

Using MIT app inventor we have to create app that can be named as SMART PARKING

It can be used to monitor and regular update from the cloud system

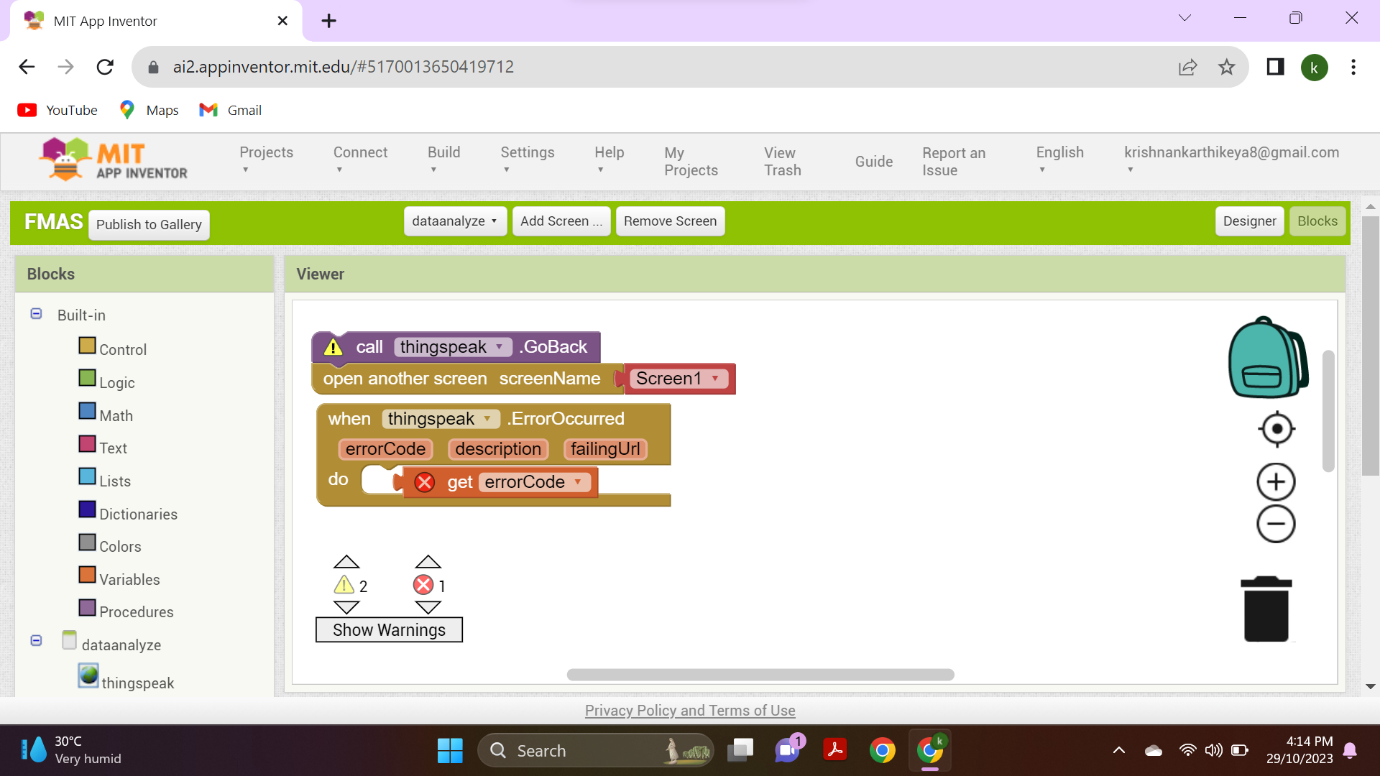
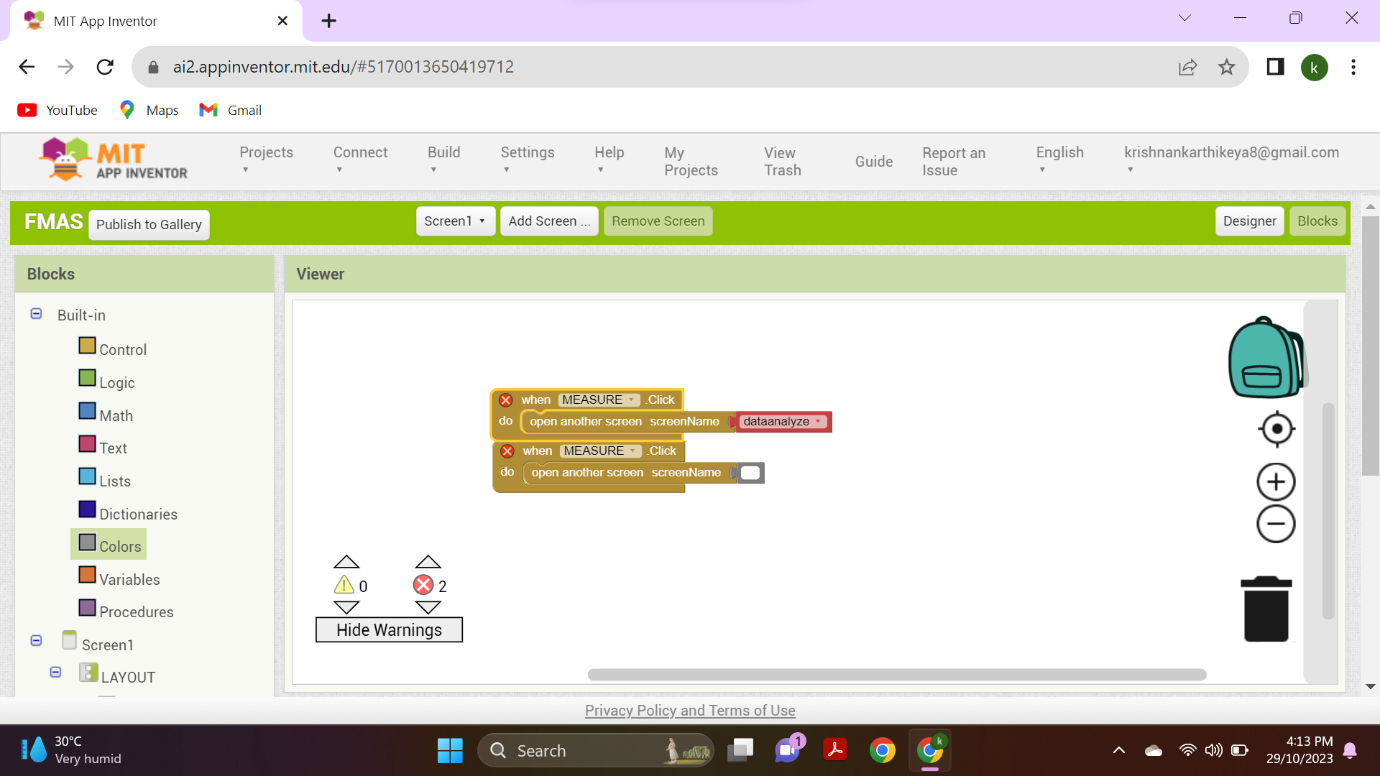
AQM contains two screen

SCREEN 1

It is the open Desktop for the SMART PARKING

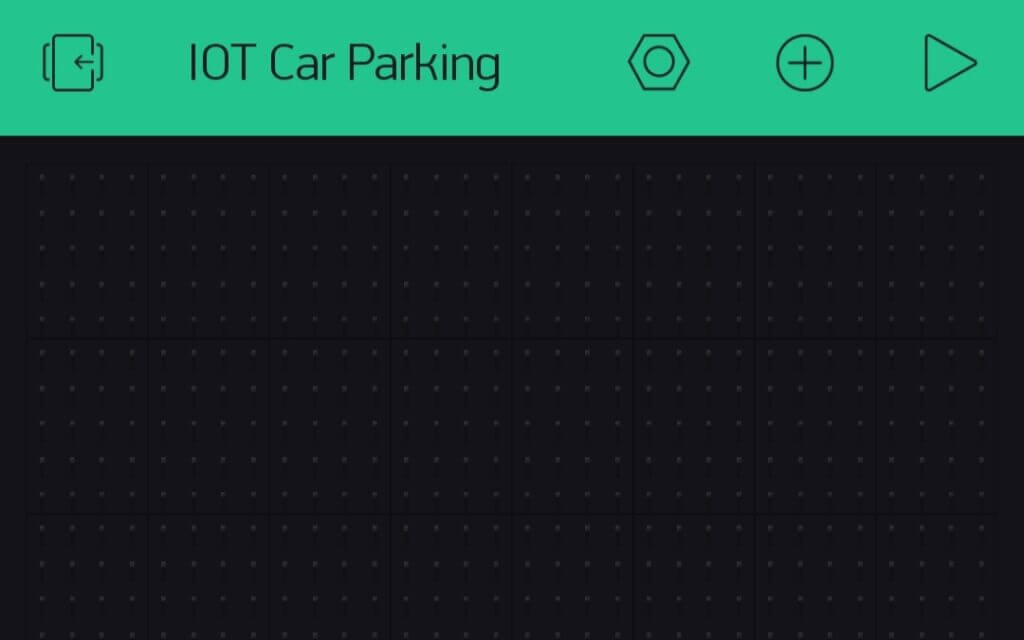
DATA ANALYST

It is used to collect the data from the cloud and provide alert message to the device



INSTALL THE APK MODE INTO THE MOBILE

1. USER INTERFACE



2.DATA ARRIVED FROM THINGSPEAK CONNECTED THROUGH THE MIT APP INVENTOR .

