```
#include <ThingSpeak.h>
#include <SoftwareSerial.h>
#define RX 2
#define TX 3
String AP = "ESP"; // AP NAME
String PASS = "Password"; // AP PASSWORD
String API = "RRTUEV94KN8SDQD8"; // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
//String field = "field1";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
SoftwareSerial esp8266(RX, TX);
int redled = 10;
int coThres = 50;
int noThres = 80;


void setup() {
  pinMode(redled, OUTPUT);
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT", 5, "OK");
  sendCommand("AT+CWMODE=1", 5, "OK");
  sendCommand("AT+CWJAP=\"" + AP + "\",\"" + PASS + "\"",
20, "OK");
}

void loop() {
  //valSensor0 = getSensorData0();
  //valSensor1 = getSensorData1();
  //valSensor2 = getSensorData2();
  String getData = "GET /update?api_key=" + API +
"&field1=" + getSensorData0() + "&field2=" +
getSensorData1() + "&field3=" + getSensorData2();
  sendCommand("AT+CIPMUX=1", 5, "OK");
  sendCommand("AT+CIPSTART=0,\"TCP\",\"" + HOST + "\","
+ PORT, 15, "OK");
  sendCommand("AT+CIPSEND=0," +
String(getData.length() + 4), 4, ">");
  esp8266.println(getData);
  delay(1500);
  countTrueCommand++;
  sendCommand("AT+CIPCLOSE=0", 5, "OK");
}
String getSensorData0() {
  Serial.print("mq2");
  float sensor_volt = 20.00;
  float RS_air;
  float R0;
  float sensorValue = 20.0; //Define variable for analog readings
  for (int x = 0 ; x < 500 ; x++)
  {
    sensorValue = sensorValue + analogRead(A0);
  }
  Serial.print("Average = ");
  Serial.println(sensorValue);
  sensorValue = sensorValue / 500.0;
  sensor_volt = sensorValue * (5.0 / 1023.0); //Convert average to voltage
  RS_air = ((5.0 * 10.0) / sensor_volt) - 10.0; //Calculate RS in fresh air
  R0 = RS_air / 9.8; //Calculate R0
  if (R0 > 50.00){
    Serial.print("R0= vheical is over limit!!!");
  }
  else{
  Serial.print("R0 = ");
  }
  //Display "R0"
  Serial.println(R0);
  return String(R0); ; // Replace with your own sensor code
}
String getSensorData1() {
  Serial.print("mq7");
  float sensor_volt = 20.00;
  float RS_air;
  float R0;
  float sensorValue = 20.0; //Define variable for analog readings
  for (int x = 0 ; x < 500 ; x++)
  {
    sensorValue = sensorValue + analogRead(A1);
  }
  Serial.print("Average = ");
  Serial.println(sensorValue);
  sensorValue = sensorValue / 500.0;
  sensor_volt = sensorValue * (5.0 / 1023.0); //Convert average to voltage
  RS_air = ((5.0 * 10.0) / sensor_volt) - 10.0; //Calculate RS in fresh air
  R0 = RS_air / 27; //Calculate R0
 if (R0 > 50.00){
    Serial.print("R0= The vheicale is over limit!!!");
  }
  else{
  Serial.print("R0 = ");
  }
  Serial.println(R0);
  return String(R0); ; // Replace with your own sensor code
  if (R0 > coThres)
  { digitalWrite (redled, HIGH);
  }
}
String getSensorData2() {
  Serial.print("mq135");
  float sensor_volt = 20.00;
  float RS_air;
  float R0;
  float sensorValue = 20.0; //Define variable for analog readings
  for (int x = 0 ; x < 500 ; x++)
  {
    sensorValue = sensorValue + analogRead(A2);
  }
  Serial.print("Average = ");
  Serial.println(sensorValue);
  sensorValue = sensorValue / 500.0;
  sensor_volt = sensorValue * (5.0 / 1023.0); //Convert average to voltage
  RS_air = ((5.0 * 10.0) / sensor_volt) - 10.0; //Calculate RS in fresh air
  R0 = RS_air / 3.7; //Calculate R0
 if (R0 > 50.00){
    Serial.print("R0= vheical is over limit!!!");
  }
  else{
  Serial.print("R0 = ");
  }
  Serial.println(R0);
  return String(R0); ; // Replace with your own sensor code
  if (R0 > noThres)
  {
    digitalWrite(redled, HIGH);

  }

}
void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while (countTimeCommand < (maxTime * 1))
  {
    esp8266.println(command);//at+cipsend
    if (esp8266.find(readReplay)) //ok
    {
      found = true;
      break;
    }
    countTimeCommand++;
  }

  if (found == true)
  {
    Serial.println("OYI");
    countTrueCommand++;
    countTimeCommand = 0;
  }

  if (found == false)
  {
    Serial.println("Fail");
    countTrueCommand = 0;
    countTimeCommand = 0;
  }

  found = false;
}
```