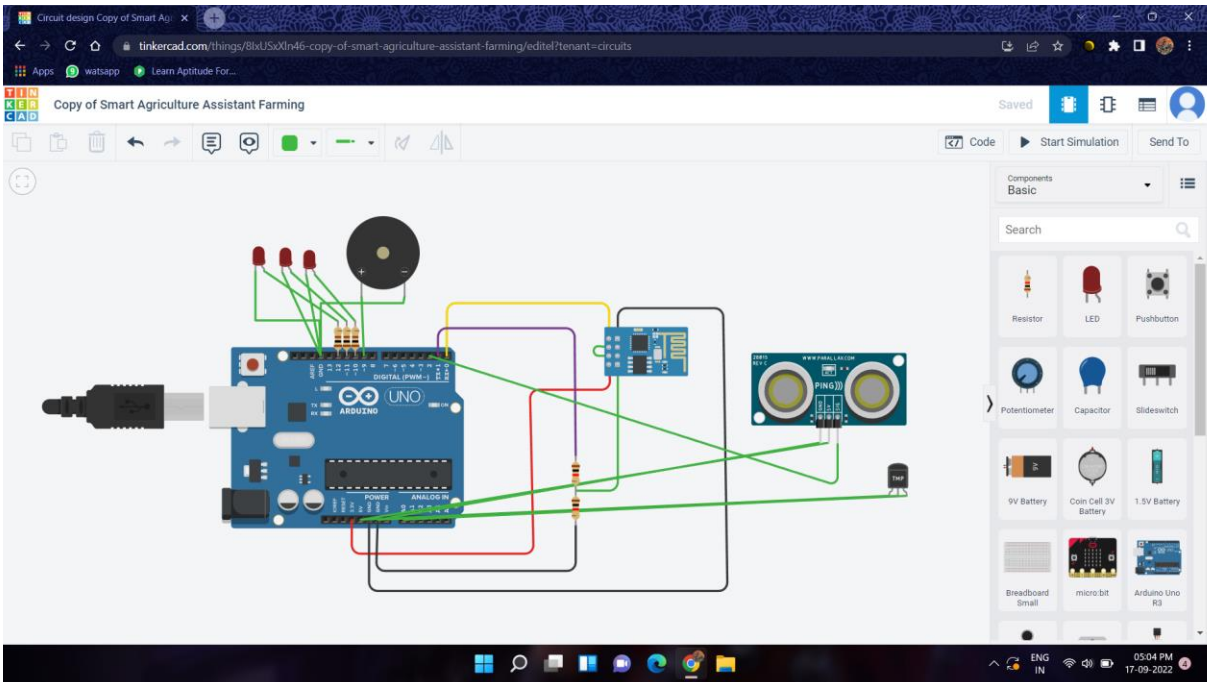

```
String ssid     = "Simulator Wifi";  // SSID to connect to

String password = "";

String host     = "api.thingspeak.com";

const int httpPort   = 80;

String url     = "/update?api_key=05SK07U2QTV09LOF&field1=";


String ssid1     = "Simulator Wifi";  // SSID to connect to

String password1 = ""; //virtual wifi has no password

String host1     = "api.thingspeak.com"; // Open Weather Map API

const int httpPort1   = 80;

String url1     = "/update?api_key=GWV5A71UI12BTLGD&field1=";

//Replace XXXXXXXXXXXXXXXX by your ThingSpeak Channel API Key

float value;

int dist = 0;

int tmp=A0;

long readUltrasonicDistance(int triggerPin, int echoPin)

{

  pinMode(triggerPin, OUTPUT);  // Clear the trigger

  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setupESP8266(void) {

  // Start our ESP8266 Serial Communication
  Serial.begin(115200);   // Serial connection over USB to computer
  Serial.println("AT");   // Serial connection on Tx / Rx port to ESP8266
  delay(5);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
    Serial.println("ESP8266 OK!!!");

  // Connect to Simulator Wifi
  Serial.println("AT+CWJAP=\"" + ssid1 + "\",\"" + password1 + "\"");
  delay(10);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
    Serial.println("Connected to WiFi!!!");

  // Open TCP connection to the host:
  //ESP8266 connects to the server as a TCP client.

  Serial.println("AT+CIPSTART=\"TCP\",\"" + host1 + "\"," + httpPort1);
  delay(50);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
```

```
   Serial.println("ESP8266 Connected to server!!!") ;

  Serial.begin(115200);   // Serial connection over USB to computer
  Serial.println("AT");   // Serial connection on Tx / Rx port to ESP8266
  delay(5);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
    Serial.println("ESP8266 OK!!!");

  // Connect to Simulator Wifi
  Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");
  delay(10);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
    Serial.println("Connected to WiFi!!!");

  // Open TCP connection to the host:
  //ESP8266 connects to the server as a TCP client.

  Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
  delay(50);        // Wait a little for the ESP to respond
  if (Serial.find("OK"))
   Serial.println("ESP8266 Connected to server!!!") ;


}

void anydata(void) {

  value =analogRead(tmp)*0.004882814;
  value = (value - 0.5) * 100.0;
  dist = 0.01520 * readUltrasonicDistance(2, 2);
  if (dist > 200) {
```

```
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);
  digitalWrite(12, HIGH);
} else {
  if (dist > 150 && dist <= 200) {
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(12, HIGH);
  } else {
    if (dist > 100 && dist <= 150) {
      digitalWrite(9, LOW);
      digitalWrite(10, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(12, HIGH);
    } else {
      digitalWrite(9, HIGH);
      digitalWrite(10, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(12, HIGH);
    }
  }
}
// Construct our HTTP call
String httpPacket = "GET " + url1 + String(dist) + " HTTP/1.1\r\nHost: " + host1 + "\r\n\r\n";
int length = httpPacket.length();

// Send our message length
Serial.print("AT+CIPSEND=");
Serial.println(length);
```

```
  delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

  // Send our http request
  Serial.print(httpPacket);
  delay(10); // Wait a little for the ESP to respond
  if (Serial.find("SEND OK\r\n"))
  Serial.println("ESP8266 sends data to the server");



  String httpPacket1 = "GET " + url + String(value) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
  int length1 = httpPacket.length();

  // Send our message length
  Serial.print("AT+CIPSEND=");
  Serial.println(length1);
  delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

  // Send our http request
  Serial.print(httpPacket1);
  delay(10); // Wait a little for the ESP to respond
  if (Serial.find("SEND OK\r\n"))
    Serial.println("ESP8266 sends data to the server");

}




void setup()
{
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);

  pinMode(tmp,INPUT);
  pinMode(dist, INPUT);
  setupESP8266();

}

void loop()
{

  {

 anydata();

  delay(4000); // delay changed for faster analytics
}

  delay(10); // Delay a little bit to improve simulation performance
}
//----------------------------------------------------------------
```