



IoT Fundamentals – ECE3501

Allen Ben Philipose – 18BIS0043

Lab Task – 2

To: Prof. Suresh Chavhan

TASK - II

Aim

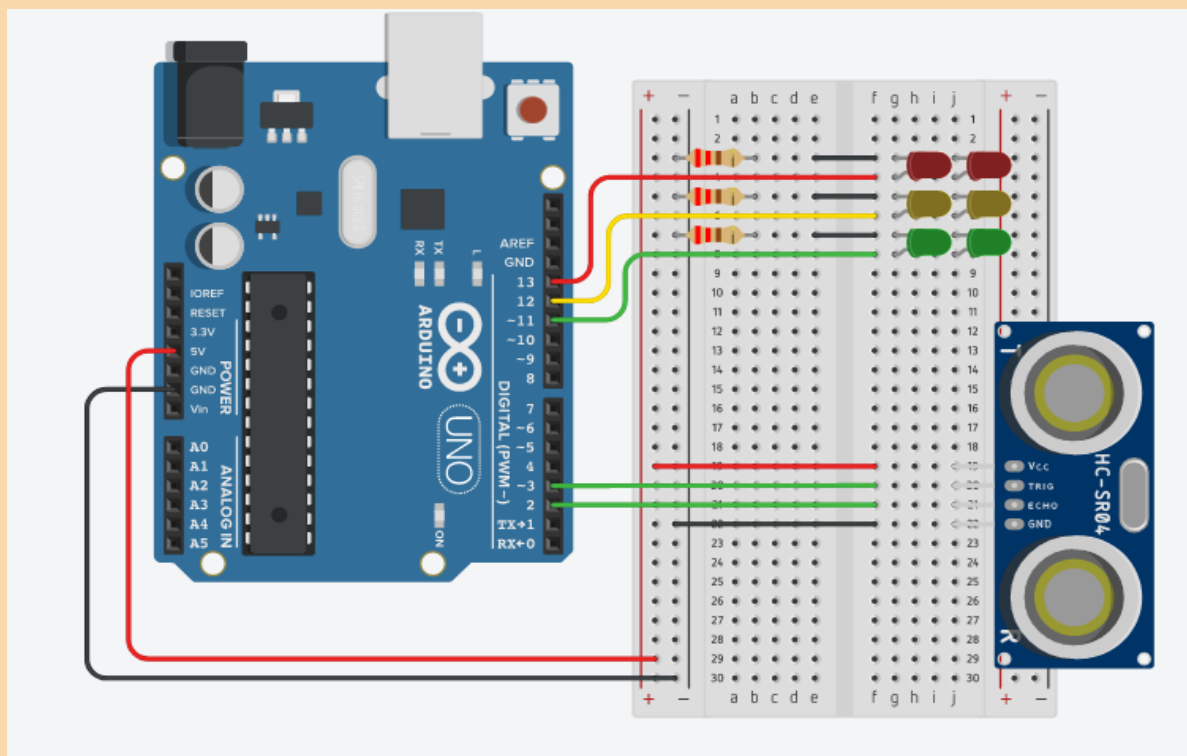
To design a circuit using Arduino for detecting distance between the system and an object using an ultrasonic sensor and plot it with respect to time

Tools Required

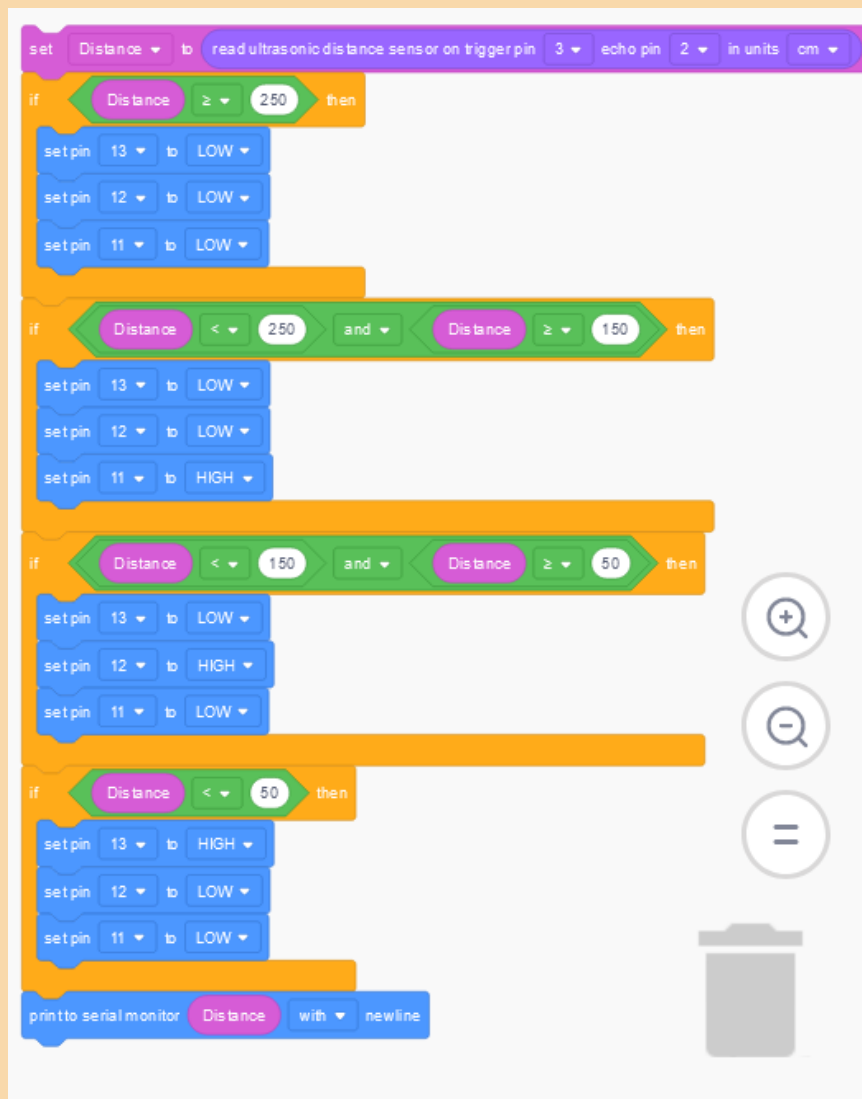
Tinkercad – for simulating the connection and coding of the Arduino circuit

ThingSpeak – for plotting the graph

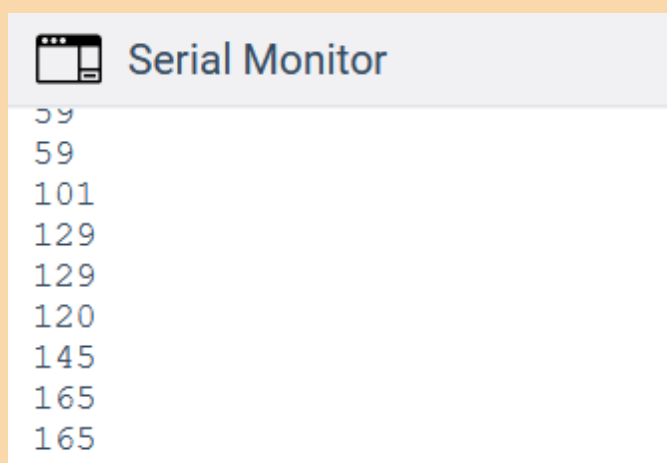
Circuit Diagram



Flow diagram



Output from Tinkercad



Code

```
int Distance = 0;

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 setup()
{
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    Distance = 0.01723 * readUltrasonicDistance(3,
2);

    if (Distance >= 250) {
        digitalWrite(13, LOW);
        digitalWrite(12, LOW);
        digitalWrite(11, LOW);
    }

    if (Distance < 250 && Distance >= 150) {
        digitalWrite(13, LOW);
```

```
    digitalWrite(12, LOW);  
    digitalWrite(11, HIGH);  
}
```

```
if (Distance < 150 && Distance >= 50) {  
    digitalWrite(13, LOW);  
    digitalWrite(12, HIGH);  
    digitalWrite(11, LOW);  
}
```

```
if (Distance < 50) {  
    digitalWrite(13, HIGH);  
    digitalWrite(12, LOW);  
    digitalWrite(11, LOW);  
}
```

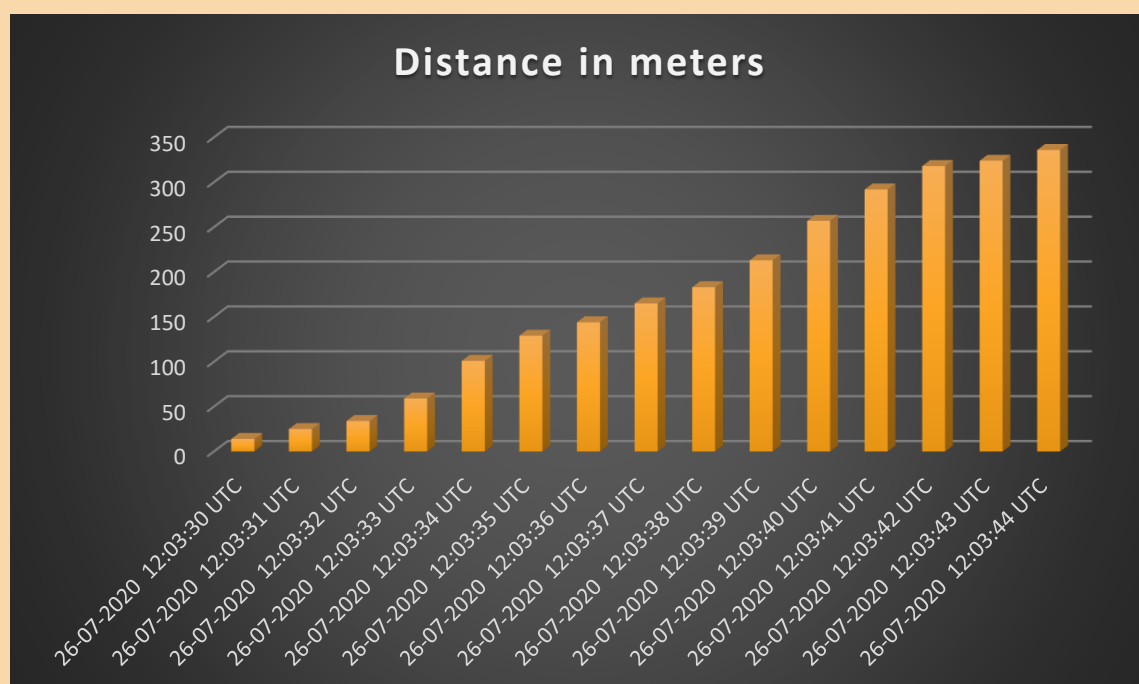
```
Serial.println(Distance);  
delay(10); // Delay a little bit to improve  
simulation performance  
}
```

Observations

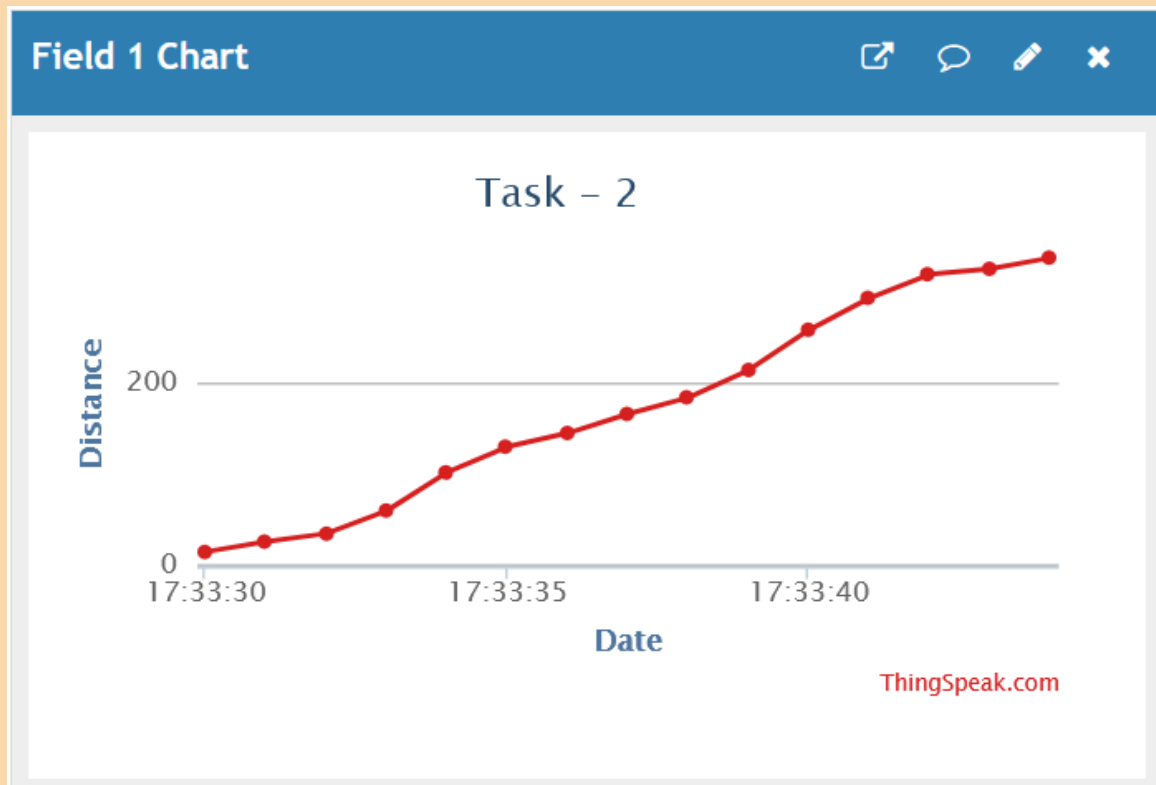
datetime	field1	latitude	longitude	elevation	status
26-07-2020 12:03:30 UTC	14	-89	155.6	22	Awake
26-07-2020 12:03:31 UTC	25	0.5	-170	23	Awake
26-07-2020 12:03:32 UTC	34	90	0	24	Awake
26-07-2020 12:03:33 UTC	59	70	10	25	Awake
26-07-2020 12:03:34 UTC	101	80	20	26	Awake
26-07-2020 12:03:35 UTC	129	40	30	27	Awake
26-07-2020 12:03:36 UTC	144	50	40	28	Awake
26-07-2020 12:03:37 UTC	165	60	50	29	Awake
26-07-2020 12:03:38 UTC	183	20	60	30	Awake
26-07-2020 12:03:39 UTC	213	30	70	31	Awake
26-07-2020 12:03:40 UTC	257	10	80	32	Awake
26-07-2020 12:03:41 UTC	292	30	90	33	Awake
26-07-2020 12:03:42 UTC	318	50	100	34	Awake
26-07-2020 12:03:43 UTC	324	60	110	35	Awake
26-07-2020 12:03:44 UTC	336	70	120	36	Awake

*Only field1 contains observed values, other values are random and do not affect the graph

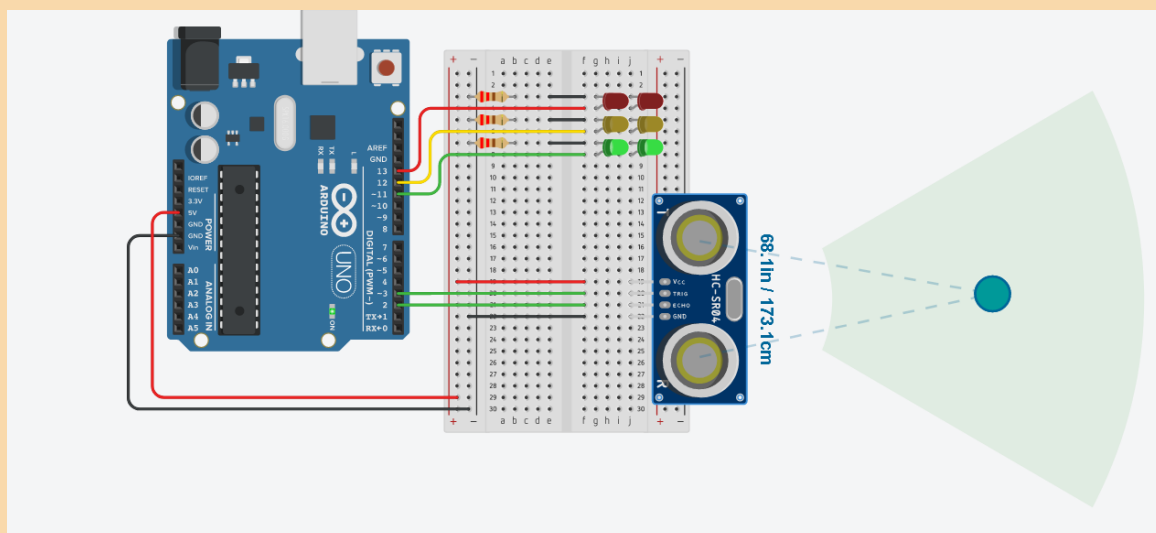
Output from Excel



Output from ThingSpeak



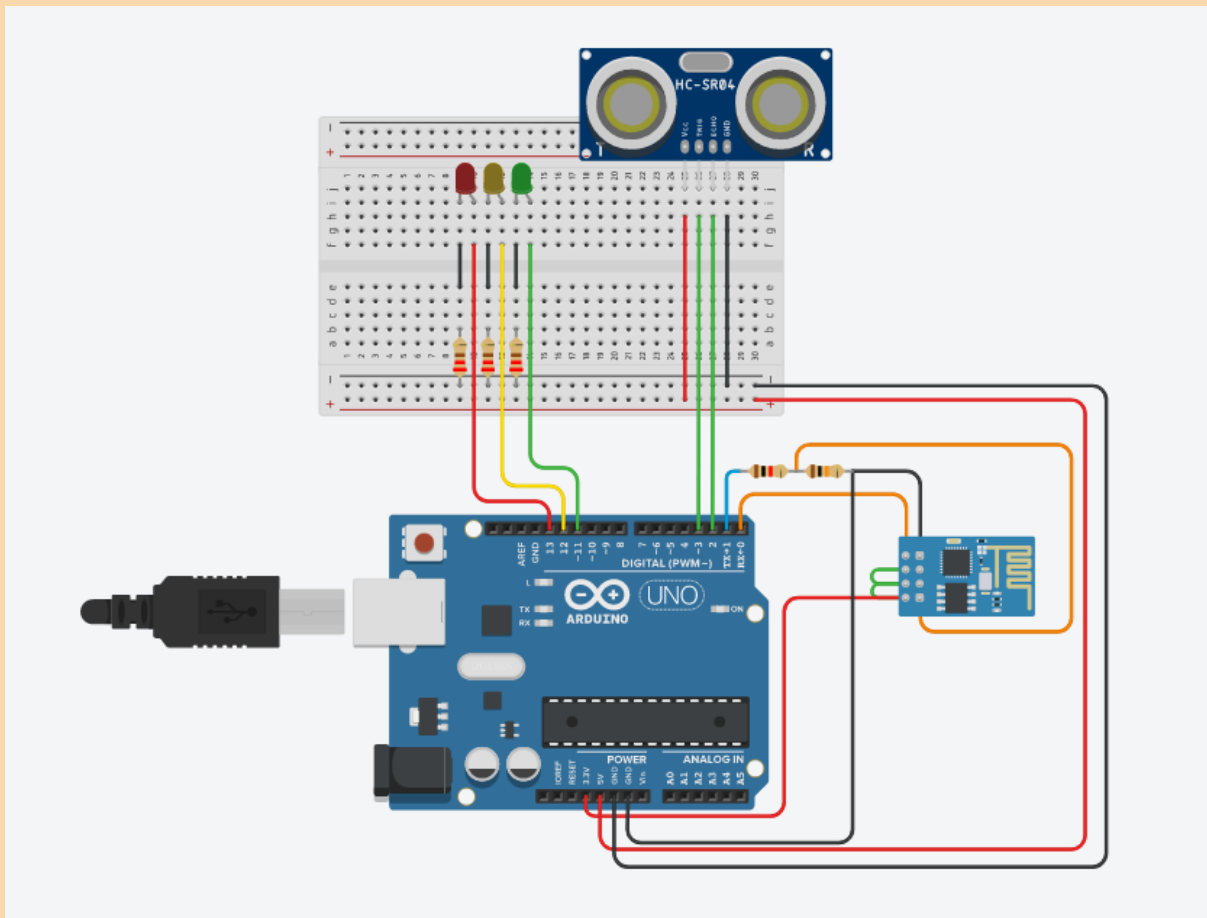
Conclusion



Therefore, by using Tinkercad, we simulated a circuit for measuring the distance between the object and the system, and by recording the output in a csv file, we can plot it using ThingSpeak.

TASK – II: WITH WIFI MODULE

Modified Circuit Diagram



Code

```
String ssid      = "Simulator Wifi";  
String password = "";  
String host      = "api.thingspeak.com";
```

```
const int httpPort    = 80;

String uri            =
"/update?api_key=CWA3UYZBMR330LSM&field1=";

int Distance = 0;

long readUltrasonicDistance(int triggerPin, int
echoPin)
{
    pinMode(triggerPin, OUTPUT);
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
    return pulseIn(echoPin, HIGH);
}

int setupESP8266(void) {
    // Start our ESP8266 Serial Communication
    Serial.begin(115200);
    Serial.println("AT");
    delay(10);
}
```

```
    if (!Serial.find("OK")) return 1;

    Serial.println("AT+CWJAP=\"" + ssid + "\",\""
+ password + "\"");
    delay(10);

    if (!Serial.find("OK")) return 2;

    // Open TCP connection to the host:

    Serial.println("AT+CIPSTART=\"TCP\",\"" +
host + "\",\" + httpPort);

    delay(50);          // Wait a little for the
ESP to respond

    if (!Serial.find("OK")) return 3;

    return 0;
}

void anydata(int t) {

    int temp = map(t,0,1000,0,1000);

    // Construct our HTTP call

    String httpPacket = "GET " + uri +
String(temp) + " HTTP/1.1\r\nHost: " + host +
"\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")) return;  
}
```

```
void setup() {  
    pinMode(13, OUTPUT);  
    pinMode(12, OUTPUT);  
    pinMode(11, OUTPUT);  
    setupESP8266();  
}  
void loop() {  
    Distance = 0.01723 *  
readUltrasonicDistance(3, 2);  
    if (Distance >= 250) {  
        digitalWrite(13, LOW);  
        digitalWrite(12, LOW);  
    }  
}
```

```
    digitalWrite(11, LOW);  
}  
if (Distance < 250 && Distance >= 150) {  
    digitalWrite(13, LOW);  
    digitalWrite(12, LOW);  
    digitalWrite(11, HIGH);  
}  
if (Distance < 150 && Distance >= 50) {  
    digitalWrite(13, LOW);  
    digitalWrite(12, HIGH);  
    digitalWrite(11, LOW);  
}  
if (Distance < 50) {  
    digitalWrite(13, HIGH);  
    digitalWrite(12, LOW);  
    digitalWrite(11, LOW);  
}  
anydata(Distance);  
delay(1000);  
}
```

Output from TinkerCad

```
Serial Monitor

GET /update?api_key=CWA3UYZBMR33OLSM&field1=266 HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=86
GET /update?api_key=CWA3UYZBMR33OLSM&field1=266 HTTP/1.1
Host: api.thingspeak.com

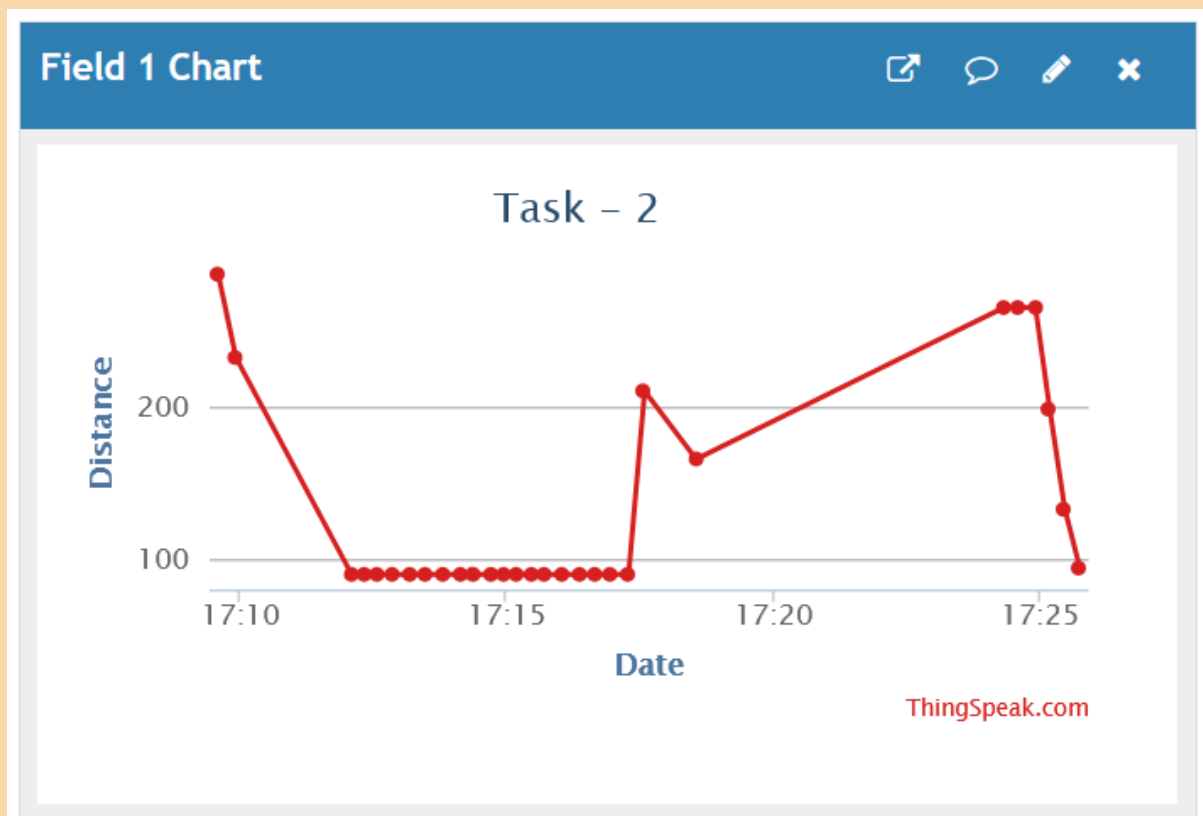
AT+CIPSEND=86
GET /update?api_key=CWA3UYZBMR33OLSM&field1=199 HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=86
GET /update?api_key=CWA3UYZBMR33OLSM&field1=188 HTTP/1.1
Host: api.thingspeak.com

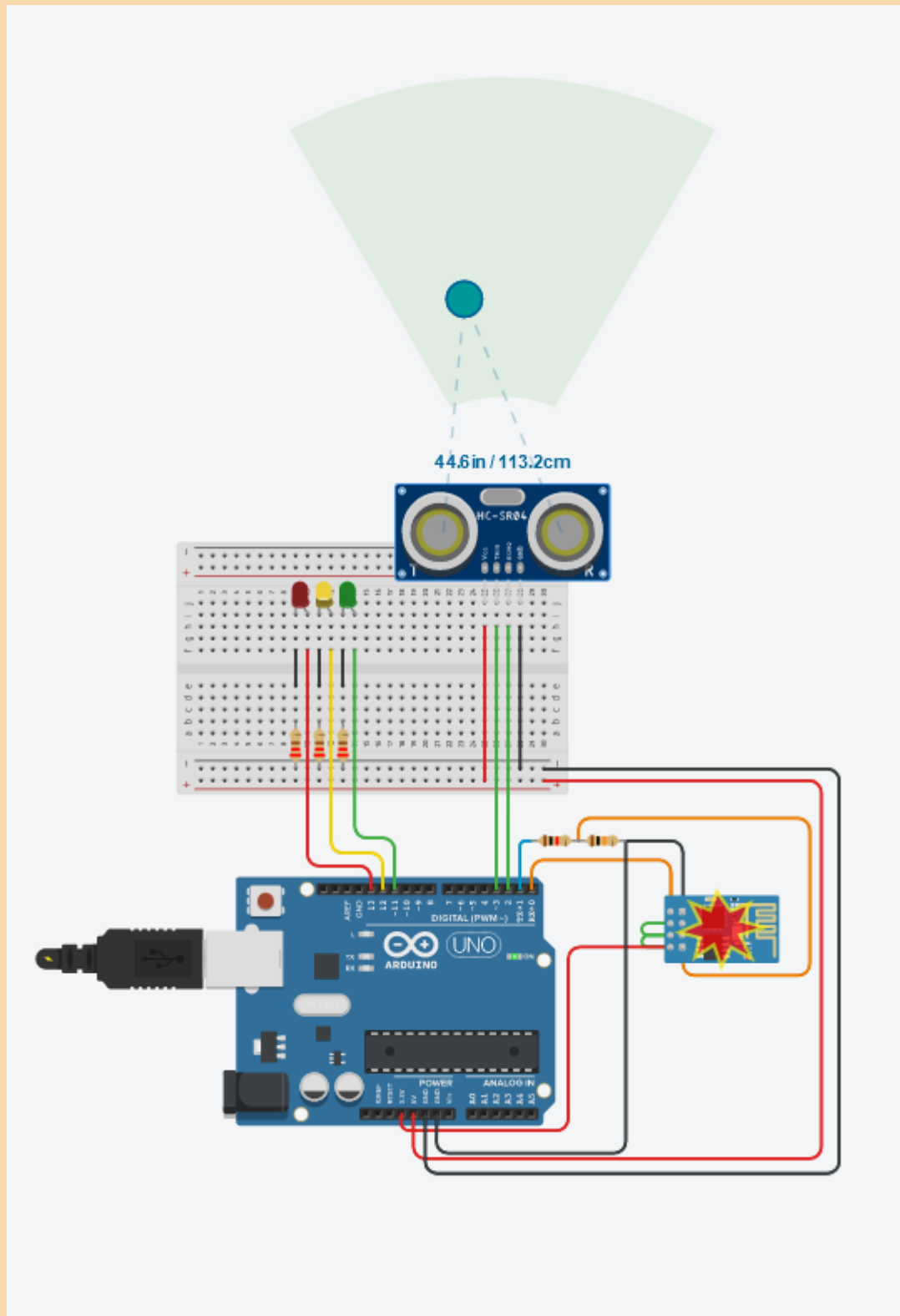
AT+CIPSEND=86
GET /update?api_key=CWA3UYZBMR33OLSM&field1=133 HTTP/1.1
Host: api.thingspeak.com

AT+CIPSEND=86
GET /update?api_key=CWA3UYZBMR33OLSM&field1=133 HTTP/1.1
Host: api.thingspeak.com
```

Output from ThingSpeak



Working circuit



The yellow light is glowing according to the given instructions.