NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

Internet of Things Lab Record

Submitted in partial fulfilment of the requirement for the award of Degree of

Bachelor of Engineering

in

Artificial Intelligence and Machine Learning

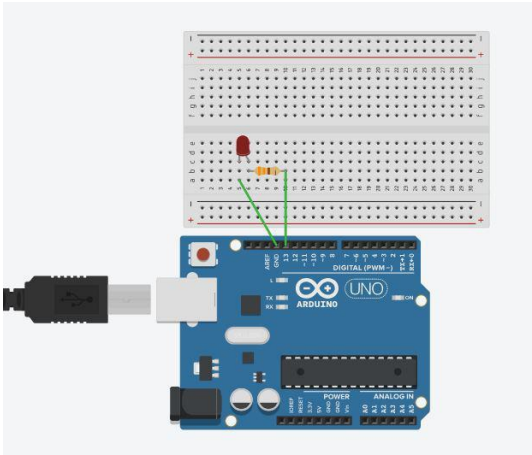
Submitted by:

Swathika Kannan 1NT21AI061

Contents

Lab Experiment 1

Onboard and External LED Blink using Arduino uno



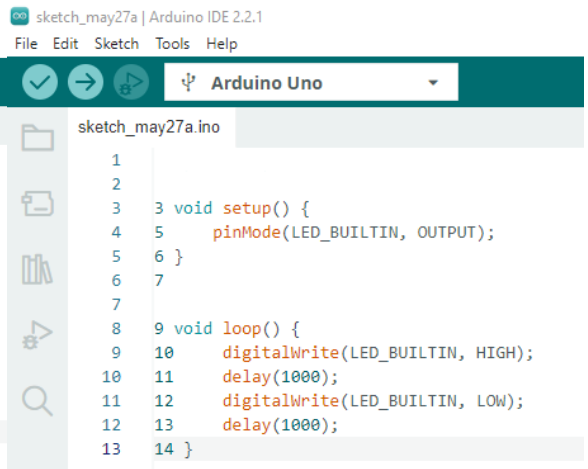
Onboard Code

```
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    delay(1000);
}
```

External LED

```
int LEDpin = 13;
int delayT = 1000;
void setup() {
  // put your setup code here, to run once:
pinMode(LEDpin, OUTPUT);
```

```
}
void loop() {
  // put your main code here, to run repeatedly:
digitalWrite(LEDpin, HIGH);
delay(delayT);
digitalWrite(LEDpin, LOW);
delay(delayT);
}
```



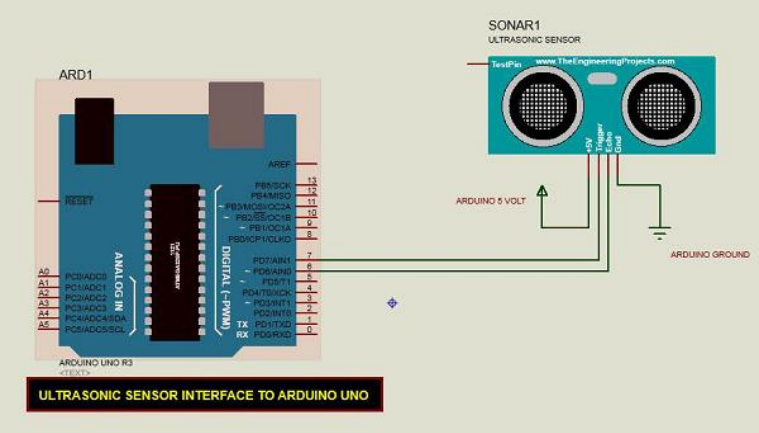
OUTPUT:

For Onboard the led blinking at regular intervals of 1s.

For external LED, the light blinking at regular interval of 1s.

Lab Experiment 2

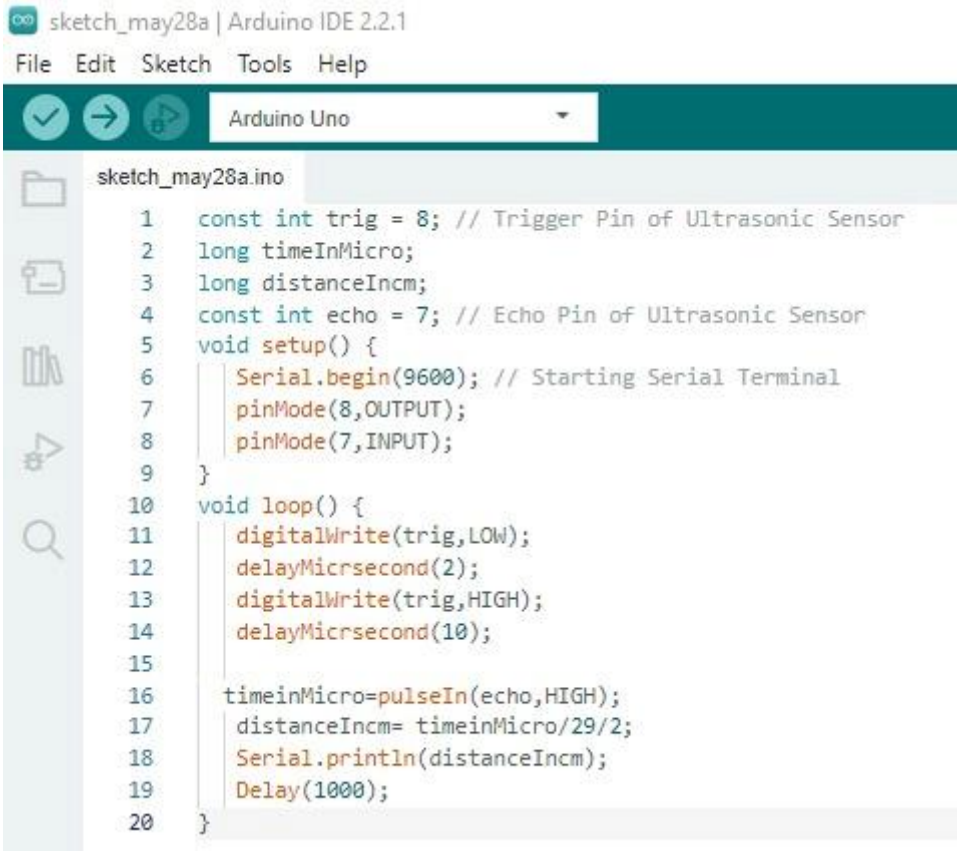
Ultrasonic sensor using Arduino UNO



CODE:

```
const int trig = 8; // Trigger Pin of Ultrasonic Sensor
long timeInMicro;
long distanceIncm;
const int echo = 7; // Echo Pin of Ultrasonic Sensor
void setup() {
  Serial.begin(9600); // Starting Serial Terminal
  pinMode(8,OUTPUT);
  pinMode(7,INPUT);
}
void loop() {
  digitalWrite(trig,LOW);
  delayMicrsecond(2);
  digitalWrite(trig,HIGH);
  delayMicrsecond(10);
```

```
  timeinMicro=pulseIn(echo,HIGH);
   distanceIncm= timeinMicro/29/2;
   Serial.println(distanceIncm);
   Delay(1000);
}
```

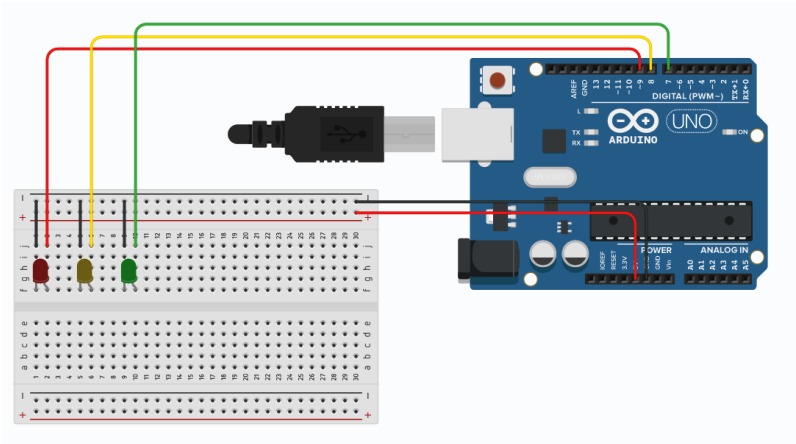


OUTPUT:

The distance measured by sensor in inches and cm on Arduino serial monitor.

Lab Experiment 3

Traffic light using Arduino UNO



CODE:

```
int red = 9;
int yellow = 8;
int green = 7;
void setup(){
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}
void loop(){
digitalWrite(red, HIGH);
 delay(15000);
digitalWrite(red, LOW);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
```

```
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
digitalWrite(green, HIGH);
delay(20000);
digitalWrite(green, LOW);
digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
```

```
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
}
```

sketch_may27a | Arduino IDE 2.2.1

File Edit Sketch Tools Help

Arduino Uno

sketch_may27a.ino

```
int red = 9;
int yellow = 8;
int green = 7;

void setup(){
  pinMode(red, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(green, OUTPUT);
}
void loop(){
digitalWrite(red, HIGH);
delay(15000);
digitalWrite(red, LOW);
digitalWrite(yellow, HIGH);
delay(1000);
digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
```

sketch_may27a | Arduino IDE 2.2.1

File Edit Sketch Tools Help

Arduino Uno

sketch_may27a.ino

```
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
digitalWrite(green, HIGH);
delay(20000);
digitalWrite(green, LOW);
digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
  digitalWrite(yellow, HIGH);
delay(1000);
  digitalWrite(yellow, LOW);
delay(500);
}
```

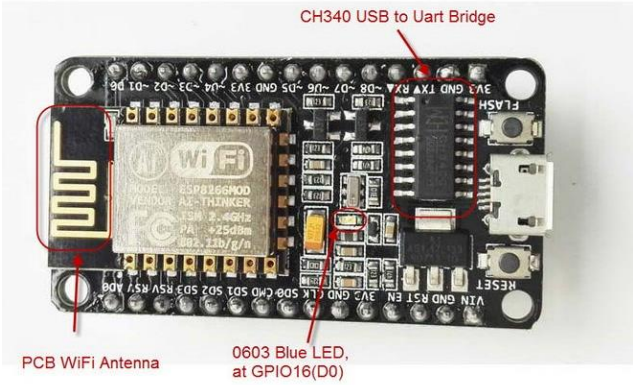
OUTPUT:

The red LED is initially on for 15s then it turns off, followed by the yellow LED which is on for 1s and turns off. The yellow LED is turned on for 0.5s for 5 times and turns off and finally the green LED which glows for 20s. This repeats on loop and variations of time delay can be made.

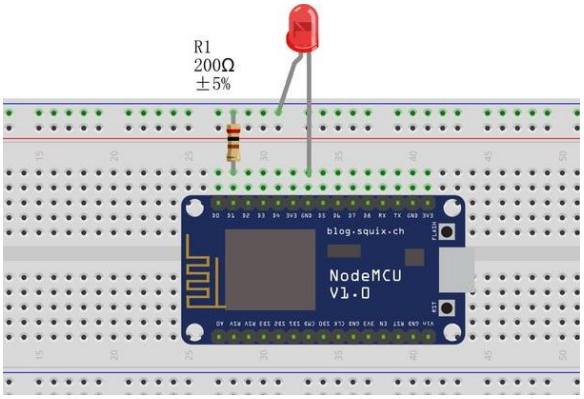
Lab Experiment 4

Onboard and External LED blink using ESP8266

Onboard



External



External:

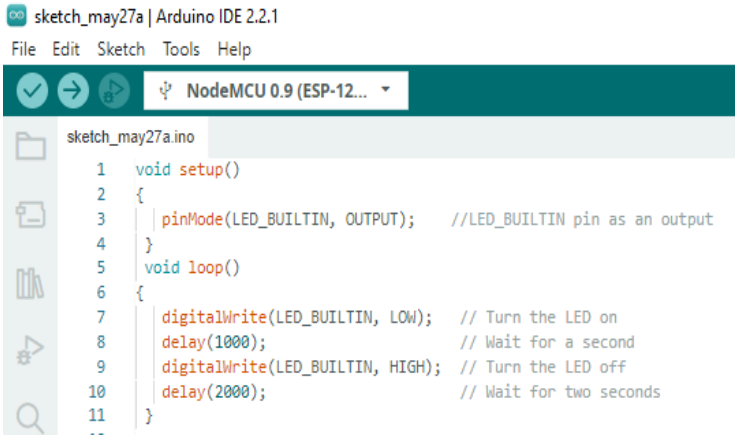
```
#define LED D0
void setup()
{
 pinMode(LED, OUTPUT); //LED pin as output
}
void loop()
{
 digitalWrite(LED, HIGH); //turn the led off
 delay(1000); //wait for 1 sec
 digitalWrite(LED, LOW); //turn the led on
 delay(1000); //wait for 1 sec
}
```

Onboard:

```
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);    //LED_BUILTIN pin as an output
 }
 void loop()
{
  digitalWrite(LED_BUILTIN, LOW);   // Turn the LED on
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off
  delay(2000);                      // Wait for two seconds
 }
```

External

Onboard



OUTPUT:

For Onboard the led blinking at regular intervals of 2s and is switched on for 1s.

For external LED, the light blinking at regular interval of 1s and is switched on for 1s.

Lab Experiment 5

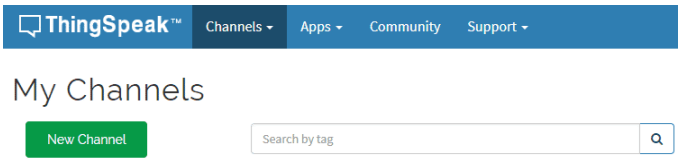
Humidity and Temperature measurement using ThingsSpeak, DHT11 and NodeMCU

ThingsSpeak:

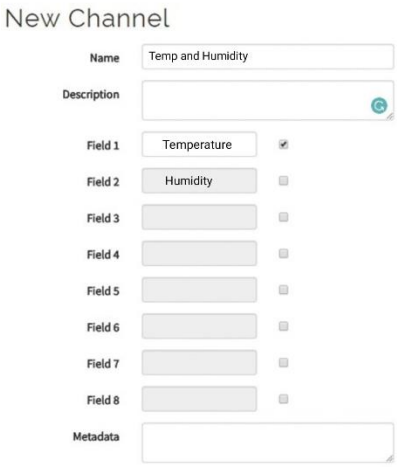
ThingSpeak is used for temperature and humidity IoT projects due to its user-friendly interface, real-time data visualization, and cloud-based data storage. It integrates seamlessly with MATLAB for advanced data analysis. The platform provides robust APIs for flexible data handling and has strong community support and documentation. It offers a free tier and scalable options for various project sizes.

Steps for setting up the ThinkSpeak App:

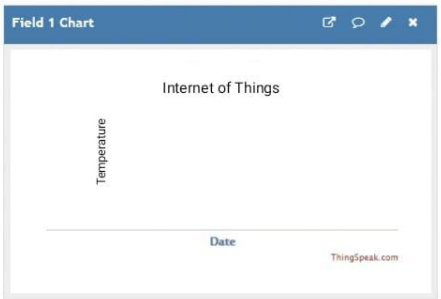
1. Create a ThingSpeak account.
2. Create a new channel:

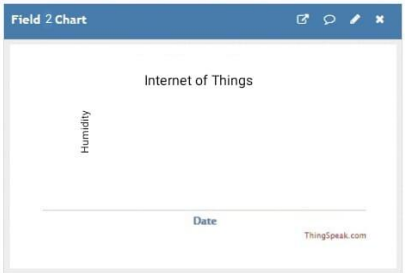


3. You can store up to 8 fields on 1 channel. We will store 1 parameter



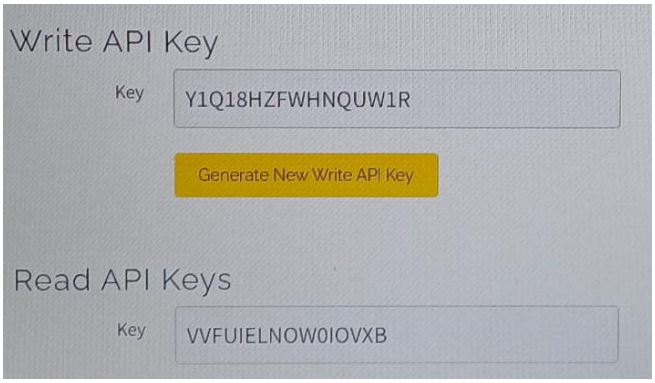
4. When a new channel is created, you can see four graphs for each parameter:



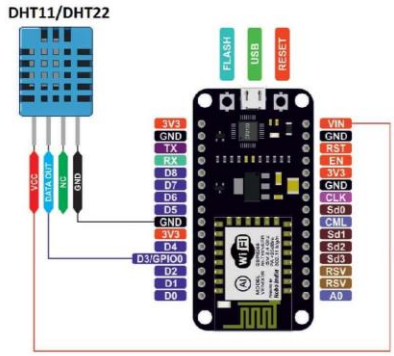


5. Get the Channel ID of your Channel (To be used while connecting to the channel):

6. Get the Read and Write API from API tab:



Circuit Diagram:

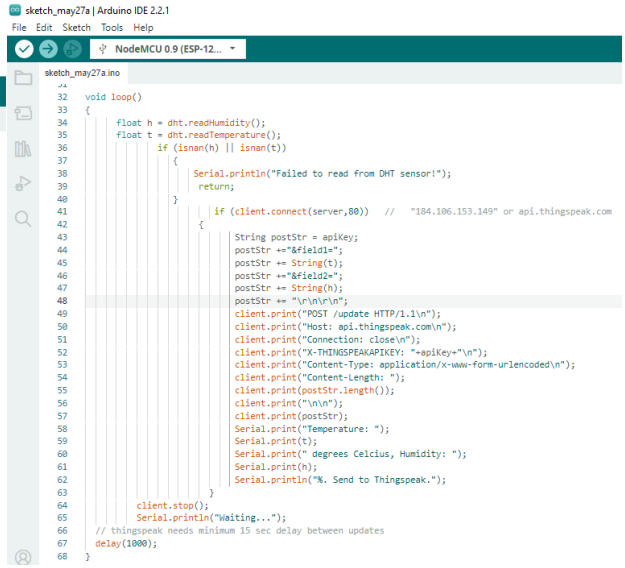


CODE:

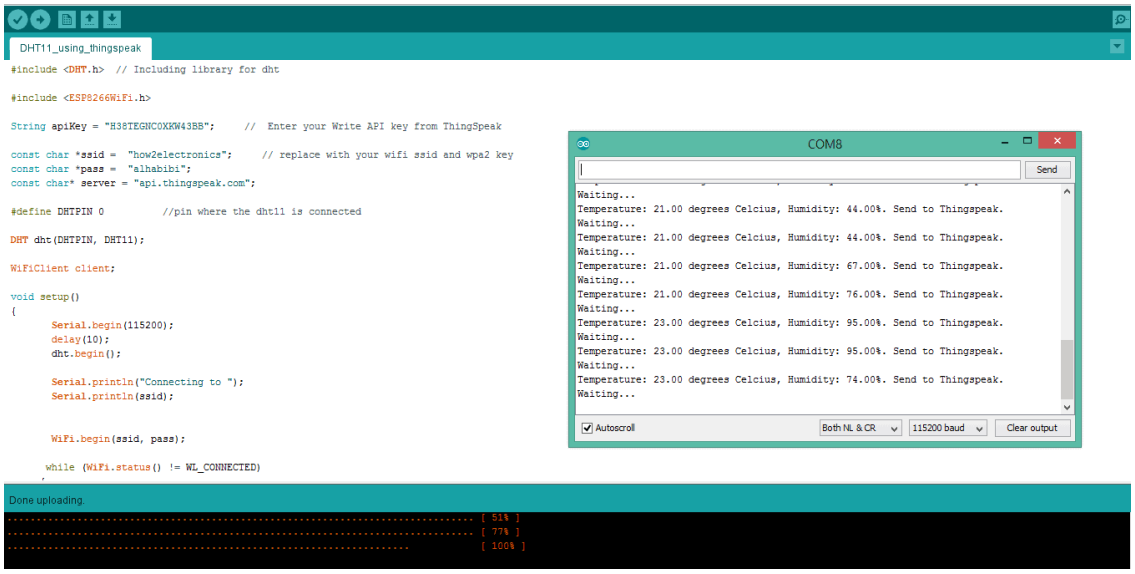
```
#include <DHT.h>  // Including library for dht
#include <ESP8266WiFi.h>
String apiKey = "H38TEGNC0XKW43BB"; //Enter your Write API key from
ThingSpeak
const char *ssid =  "how2electronics";  // replace with your wifi ssid and wpa2 key
const char *pass =  "alhabibi";
const char* server = "api.thingspeak.com";
#define DHTPIN 0          //pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
WiFiClient client;
void setup()
{
      Serial.begin(115200);
      delay(10);
      dht.begin();
      Serial.println("Connecting to ");
      Serial.println(ssid);
      WiFi.begin(ssid, pass);
     while (WiFi.status() != WL_CONNECTED)
    {
          delay(500);
          Serial.print(".");
    }
     Serial.println("");
     Serial.println("WiFi connected");
}
```

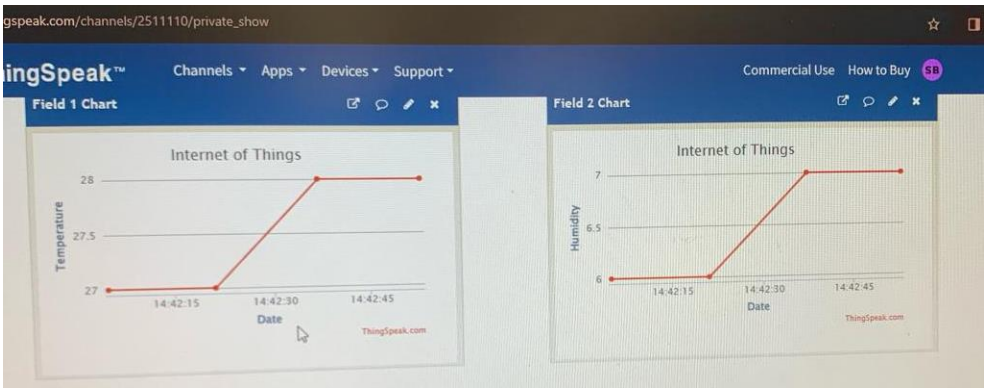
```
void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();
        if (isnan(h) || isnan(t))
          {
            Serial.println("Failed to read from DHT sensor!");
             return;
          }
              if (client.connect(server,80)) // "184.106.153.149" or
api.thingspeak.com
             {
                 String postStr = apiKey;
                 postStr +="&field1=";
                 postStr += String(t);
                 postStr +="&field2=";
                 postStr += String(h);
                 postStr += "\r\n\r\n";
                 client.print("POST /update HTTP/1.1\n");
                 client.print("Host: api.thingspeak.com\n");
                 client.print("Connection: close\n");
                 client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
                 client.print("Content-Type: application/x-www-form-
urlencoded\n");
                 client.print("Content-Length: ");
                 client.print(postStr.length());
                 client.print("\n\n");
```

```
                client.print(postStr);
                Serial.print("Temperature: ");
                Serial.print(t);
                Serial.print(" degrees Celcius, Humidity: ");
                Serial.print(h);
                Serial.println("%. Send to Thingspeak.");
            }
        client.stop();
        Serial.println("Waiting...");
  // thingspeak needs minimum 15 sec delay between updates
  delay(1000);
}
```



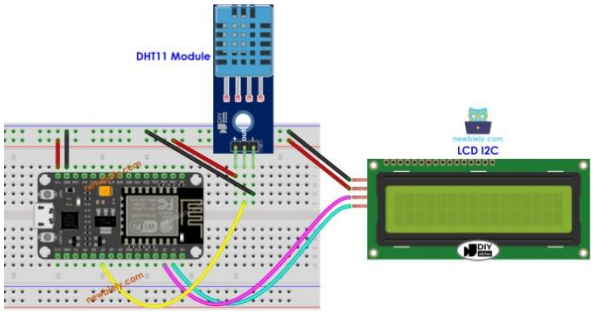
OUTPUT:


DHT11_using_thingspeak
#include <DHT.h> // Including library for dht
#include <ESP8266WiFi.h>
String apiKey = "H38TEGNC0XKW43BB"; // Enter your Write API key from ThingSpeak
const char *ssid = "how2electronics"; // replace with your wifi ssid and wpa2 key
const char *pass = "alhabibi";
const char* server = "api.thingspeak.com";
#define DHTPIN 0 //pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
WiFiClient client;
void setup()
{
Serial.begin(115200);
delay(10);
dht.begin();
Serial.println("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED)
COM8
Send
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 44.00%. Send to Thingspeak.
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 44.00%. Send to Thingspeak.
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 67.00%. Send to Thingspeak.
Waiting...
Temperature: 21.00 degrees Celcius, Humidity: 76.00%. Send to Thingspeak.
Waiting...
Temperature: 23.00 degrees Celcius, Humidity: 95.00%. Send to Thingspeak.
Waiting...
Temperature: 23.00 degrees Celcius, Humidity: 95.00%. Send to Thingspeak.
Waiting...
Temperature: 23.00 degrees Celcius, Humidity: 74.00%. Send to Thingspeak.
Waiting...
Autoscroll
Both NL & CR
115200 baud
Clear output
Done uploading.
[51%]
[77%]
[100%]


gspeak.com/channels/2511110/private_show
ingSpeak™
Channels
Apps
Devices
Support
Commercial Use
How to Buy
Field 1 Chart
Internet of Things
Temperature
28
27.5
27
14:42:15
14:42:30
14:42:45
Date
ThingSpeak.com
Field 2 Chart
Internet of Things
Humidity
7
6.5
6
14:42:15
14:42:30
14:42:45
Date
ThingSpeak.com

Lab Experiment 6

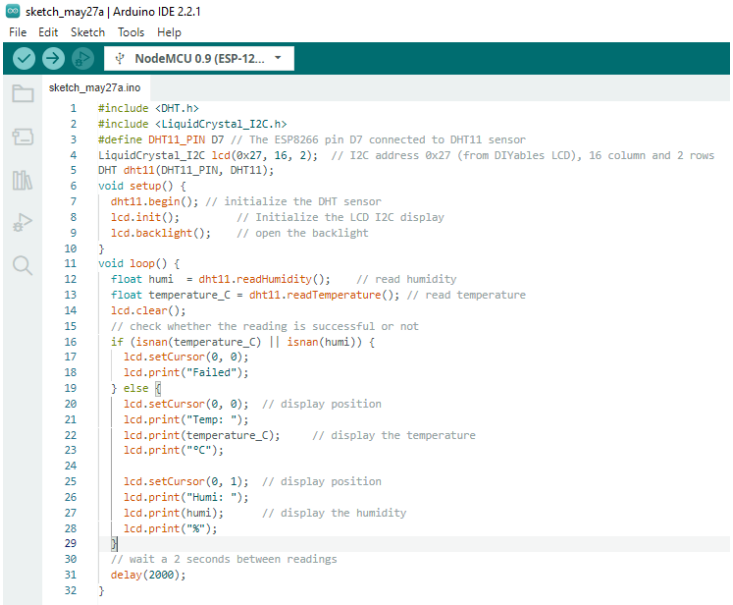
Liquid Crystal LCD with ESP8266



CODE:

```
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
#define DHT11_PIN D7 // The ESP8266 pin D7 connected to DHT11 sensor
LiquidCrystal_I2C lcd(0x27, 16, 2);  // I2C address 0x27 (from DIYables LCD),
16 column and 2 rows
DHT dht11(DHT11_PIN, DHT11);
void setup() {
  dht11.begin(); // initialize the DHT sensor
  lcd.init();        // Initialize the LCD I2C display
  lcd.backlight();    // open the backlight
}
void loop() {
  float humi  = dht11.readHumidity();    // read humidity
  float temperature_C = dht11.readTemperature(); // read temperature
  lcd.clear();
  // check whether the reading is successful or not
  if (isnan(temperature_C) || isnan(humi)) {
    lcd.setCursor(0, 0);
```

```
    lcd.print("Failed");
  } else {
    lcd.setCursor(0, 0);  // display position
    lcd.print("Temp: ");
    lcd.print(temperature_C);     // display the temperature
    lcd.print("°C");
    lcd.setCursor(0, 1);  // display position
    lcd.print("Humi: ");
    lcd.print(humi);      // display the humidity
    lcd.print("%");
  }
  // wait a 2 seconds between readings
  delay(2000);
}
```



OUTPUT:

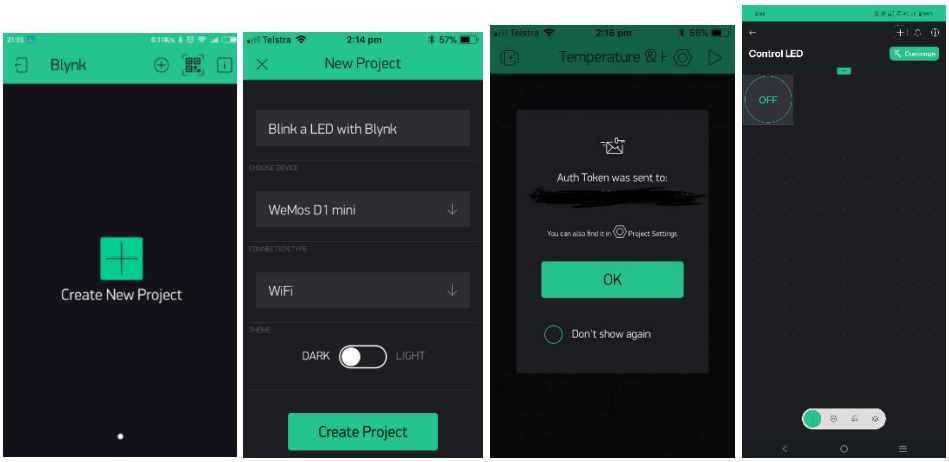
Displays the humidity and temperature on the LCD display and is monitored every 2 seconds.

Lab Experiment 7

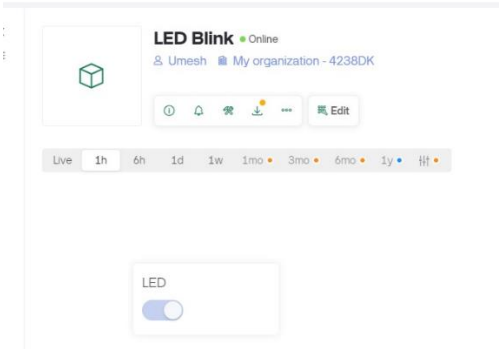
External LED Blink using Blynk App and ESP8266

Configuring Blynk App:

1. Download and install the Blynk app from the Google Play Store, then create an account using your Gmail.
2. Create a new project named "Control LED," select ESP32 Dev Board, choose Wi-Fi as the connection type, and click Create.



3. Check your email for the authentication token sent by Blynk.
4. On the canvas window, tap to open the widget box and add a Button widget.
5. Set the Button widget to control GPIO2 in switch mode, then press Play to interact with the hardware.



CODE:

```
#define BLYNK_TEMPLATE_ID "TMPL3iOcDda6Z"
#define BLYNK_DEVICE_NAME "Control LED"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define BLYNK_AUTH_TOKEN "kYSJcSlRqpHsPMkZgbaHbBnYms2nleaQ"
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Nokia G21";//Enter your WIFI name
char pass[] = "bandi12345";//Enter your WIFI password
int ledpin = D4;
//Get the button value
BLYNK_WRITE(V0) {
  digitalWrite(ledpin, param.asInt());
}
void setup() {
  pinMode(ledpin, OUTPUT);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  WiFi.begin(ssid, pass);
  Blynk.config(BLYNK_AUTH_TOKEN);
}
```

```
void loop() {
  //Run the Blynk library
  Blynk.run();
}
```

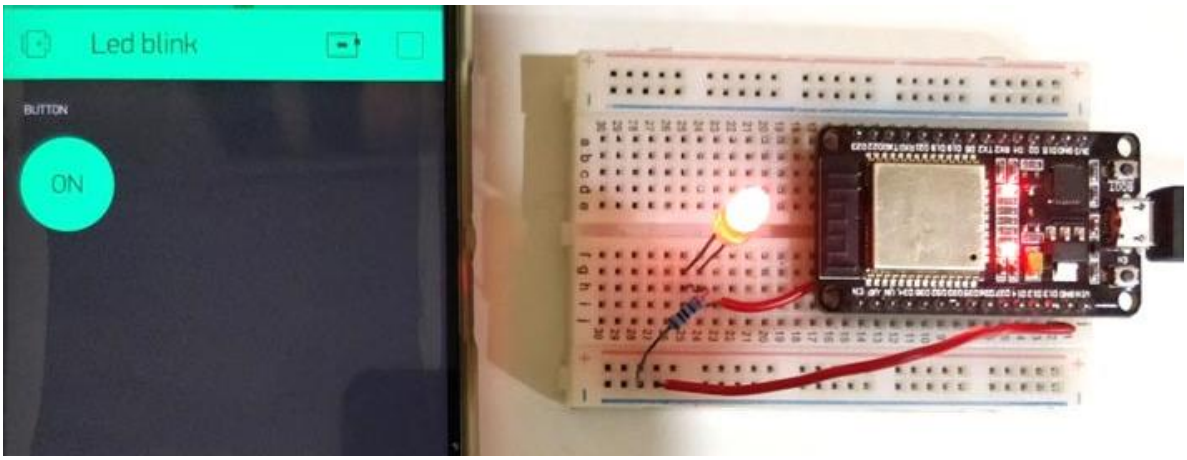
sketch_may27a | Arduino IDE 2.2.1

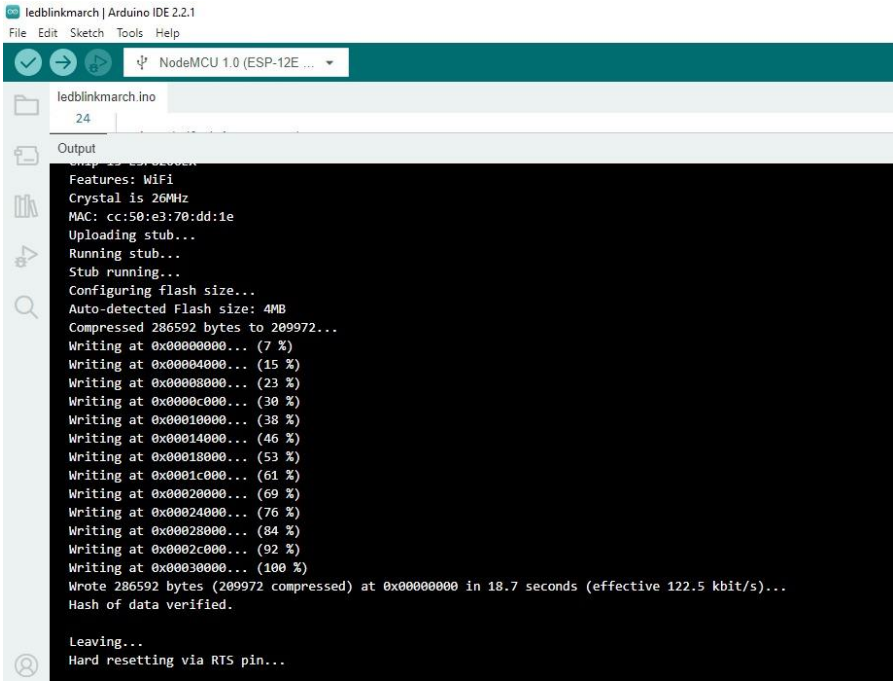
File Edit Sketch Tools Help

NodeMCU 0.9 (ESP-12...

sketch_may27a.ino

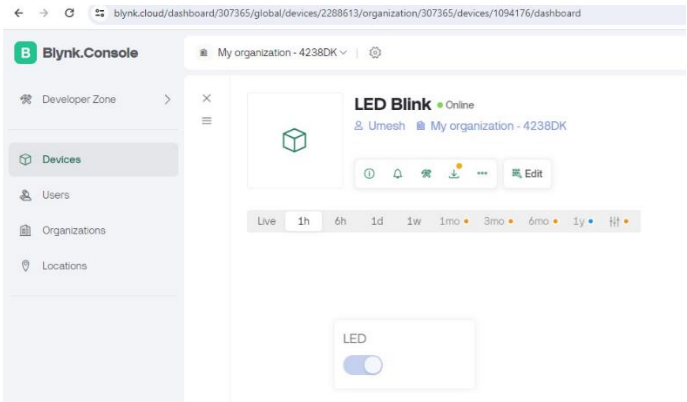
```
#define BLYNK_TEMPLATE_ID "TMPL3iOcDda6Z"
#define BLYNK_DEVICE_NAME "Control LED"
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define BLYNK_AUTH_TOKEN "kYSJcSlRqpHsPMkZgbaHbBnYms2nleaQ" //Enter your blynk auth token
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Nokia G21";//Enter your WIFI name
char pass[] = "bandi12345";//Enter your WIFI password
int ledpin = D4;
//Get the button value
BLYNK_WRITE(V0) {
  digitalWrite(ledpin, param.asInt());
}

void setup() {
  //Set the LED pin as an output pin
  pinMode(ledpin, OUTPUT);
  //Initialize the Blynk library
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  WiFi.begin(ssid, pass);
  Blynk.config(BLYNK_AUTH_TOKEN);

}
void loop() {
  //Run the Blynk library
  Blynk.run();
}
```





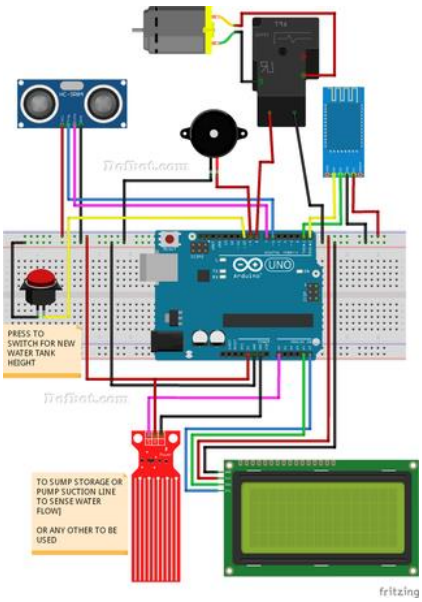
OUTPUT:

The LED turns on and off based on the switch of the blynk app and is also displayed on the interface.



Lab Experiment 8

Water Level monitoring using NodeMCU and Ultrasonic



CODE:

```
int trig=8; int echo=7;
long timeInMicro; long distanceInCm;
void setup(){
Serial.begin(9600); pinMode(8,OUTPUT); pinMode(2,INPUT);
}
void loop(){
digitalWrite(trig,LOW); delayMicroseconds(2);
digitalWrite(trig,HIGH); delayMicroseconds(10);
digitalWrite(trig,LOW);
timeInMicro= pulseIn(echo,HIGH); distanceInCm=timeInMicro/29 /2;
Serial.println(distanceInCm); delay(2000);
}
```

sketch_may30a | Arduino IDE 2.2.1

File Edit Sketch Tools Help

Arduino Uno

sketch_may30a.ino

```
int trig=8; int echo=7;
long timeInMicro; long distanceInCm;
void setup()
{
Serial.begin(9600); pinMode(8,OUTPUT); pinMode(2,INPUT);
}
void loop()
{
digitalWrite(trig,LOW); delayMicroseconds(2);
digitalWrite(trig,HIGH); delayMicroseconds(10);
digitalWrite(trig,LOW);
timeInMicro= pulseIn(echo,HIGH); distanceInCm=timeInMicro/29 /2;
Serial.println(distanceInCm); delay(2000);
}
```

OUTPUT

The distance between the water level and the place of the senor is calculated and displayed.

sketch_jun6a | Arduino IDE 2.2.1

File Edit Sketch Tools Help

Arduino Uno

sketch_jun6a.ino

```
long timeInMicro; long distanceInCm;
```

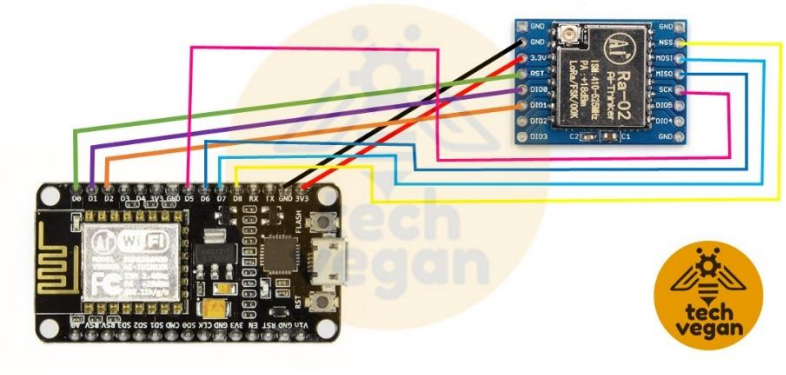
Output Serial Monitor ×

Not connected. Select a board and a port to connect autom

```
□□□□□□□□□□□□□□□□□□□□□□□176
175
10
4
4
3
3
4
2
3
3
4
13
202
5
176
21
21
22
17
17
18
18
```

Lab Experiment 9

LoRa using NodeMCU(ESP8266)

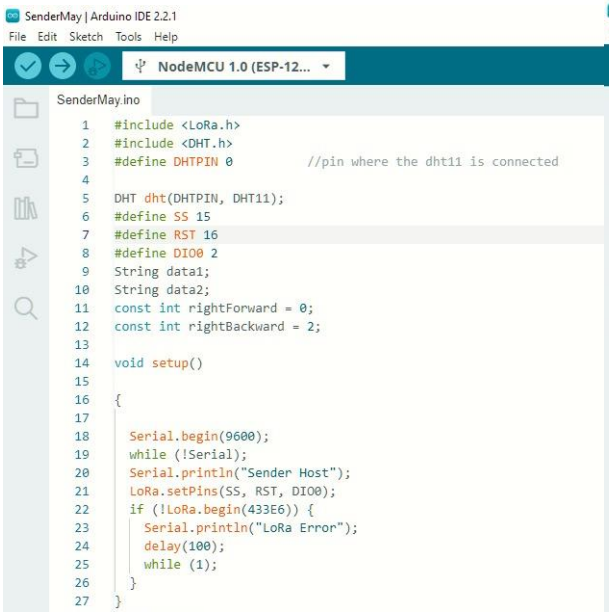


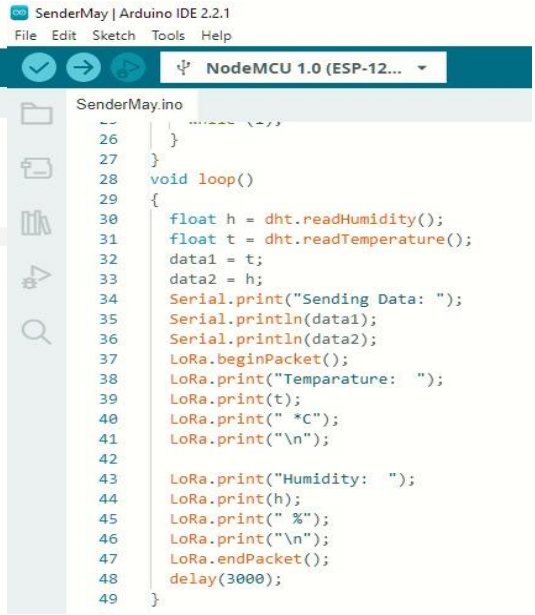
CODE:

Sender

```
#include <LoRa.h>
#include <DHT.h>
#define DHTPIN 0          //pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
#define SS 15
#define RST 16
#define DIO0 2
String data1;
String data2;
const int rightForward = 0;
const int rightBackward = 2;
void setup(){
  Serial.begin(9600);
  while (!Serial);
  Serial.println("Sender Host");
  LoRa.setPins(SS, RST, DIO0);
```

```
  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa Error");
    delay(100);
    while (1);
  }
}
void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  data1 = t;
  data2 = h;
  Serial.print("Sending Data: ");
  Serial.println(data1);
  Serial.println(data2);
  LoRa.beginPacket();
  LoRa.print("Temparature:  ");
  LoRa.print(t);
  LoRa.print(" *C");
  LoRa.print("\n");
  LoRa.print("Humidity:  ");
  LoRa.print(h);
  LoRa.print(" %");
  LoRa.print("\n");
  LoRa.endPacket();
  delay(3000);
}
```





Receiver:

```
#include <ESP8266WiFi.h>
#include <LoRa.h>
String apiKey = "MXKN78RD40818DX2";// Enter your Write API key from ThingSpeak
const char *ssid =  "vivo 1904";     // replace with your wifi ssid and wpa2 key
const char *pass =  "1e544b2103a4";
const char* server = "api.thingspeak.com";
#define SS D8
#define RST D0
#define DIO0 D1
WiFiClient client;
void setup() {
  //digitalWrite(rightForward,HIGH);
  //digitalWrite(rightBackward,LOW);
  Serial.begin(9600);
```

```
  while (!Serial);
  Serial.println("Receiver Host");
  LoRa.setPins(SS, RST, DIO0);
  delay(1000);
  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa Error");
    delay(100);
    while (1);
  }
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
          delay(500);
          Serial.print(".");
     }
      Serial.println("");
      Serial.println("WiFi connected");
}
void loop() {
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    Serial.println("Receiving Data: ");
    while (LoRa.available()) {
      String data = LoRa.readString();
      Serial.println(data);
      String temp = data.substring(18, 14);
```

```
  Serial.println(temp);
  client.connect(server,80);   //   "184.106.153.149" or api.thingspeak.com
  String postStr = apiKey;
  postStr +="&field1=";
  postStr += String(temp);
  postStr += "\r\n\r\n";
  client.print("POST /update HTTP/1.1\n");
  client.print("Host: api.thingspeak.com\n");
  client.print("Connection: close\n");
  client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
  client.print("Content-Type: application/x-www-form-urlencoded\n");
  client.print("Content-Length: ");
  client.print(postStr.length());
  client.print("\n\n");
  client.print(postStr);
     client.stop();
}}}
```

Receiver | Arduino 1.8.10

File Edit Sketch Tools Help

Receiver §

```
#include <ESP8266WiFi.h>
#include <LoRa.h>


String apiKey = "MXXN78RD40818DX2";     //  Enter your Write API key from ThingSpeak

const char *ssid =  "vivo 1904";     // replace with your wifi ssid and wpa2 key
const char *pass =  "1e544b2103a4";
const char* server = "api.thingspeak.com";

#define SS D8
#define RST D0
#define DIO0 D1

WiFiClient client;

void setup() {

  //digitalWrite(rightForward,HIGH);
  //digitalWrite(rightBackward,LOW);

  Serial.begin(9600);
  while (!Serial);
  Serial.println("Receiver Host");
  LoRa.setPins(SS, RST, DIO0);

  delay(1000);
  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa Error");
    delay(100);
    while (1);
  }

  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED)
     {
          delay(500);
```

Receiver | Arduino 1.8.10

File Edit Sketch Tools Help

Receiver §

```
  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED)
     {
          delay(500);
          Serial.print(".");
     }
      Serial.println("");
      Serial.println("WiFi connected");


}
void loop() {


  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    Serial.println("Receiving Data: ");
    while (LoRa.available()) {

      String data = LoRa.readString();
      Serial.println(data);

      String temp = data.substring(18, 14);
      Serial.println(temp);

      client.connect(server,80);   //   "184.106.153.149" or api.thingspeak.com
      String postStr = apiKey;
      postStr +="&field1=";
      postStr += String(temp);
      postStr += "\r\n\r\n";

      client.print("POST /update HTTP/1.1\n");
      client.print("Host: api.thingspeak.com\n");
      client.print("Connection: close\n");
      client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
      client.print("Content-Type: application/x-www-form-urlencoded\n");
      client.print("Content-Length: ");
      client.print(postStr.length());
      client.print("\n\n");
      client.print(postStr);
```

Receiver | Arduino 1.8.10

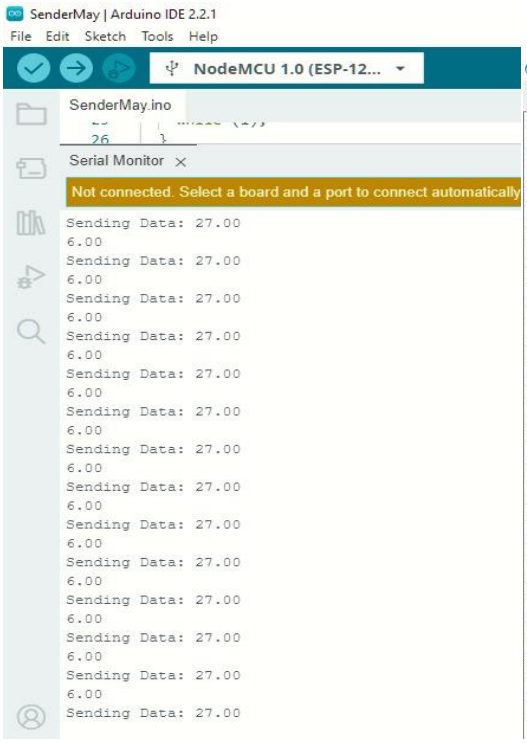
File Edit Sketch Tools Help

Receiver §

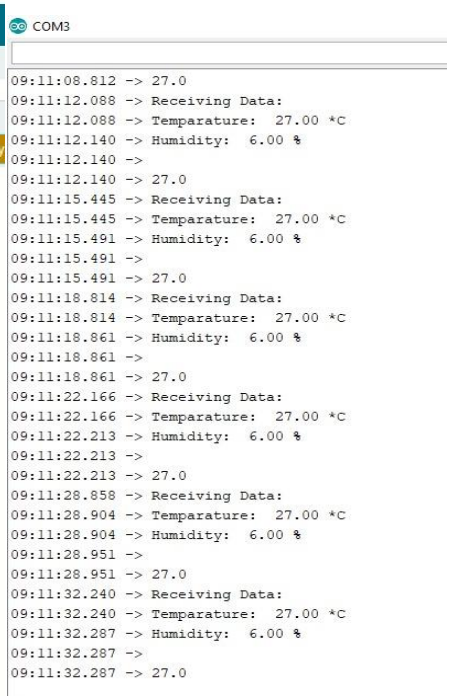
```
}
void loop() {


  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    Serial.println("Receiving Data: ");
    while (LoRa.available()) {

      String data = LoRa.readString();
      Serial.println(data);

      String temp = data.substring(18, 14);
      Serial.println(temp);

      client.connect(server,80);   //   "184.106.153.149" or api.thingspeak.com
      String postStr = apiKey;
      postStr +="&field1=";
      postStr += String(temp);
      postStr += "\r\n\r\n";

      client.print("POST /update HTTP/1.1\n");
      client.print("Host: api.thingspeak.com\n");
      client.print("Connection: close\n");
      client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
      client.print("Content-Type: application/x-www-form-urlencoded\n");
      client.print("Content-Length: ");
      client.print(postStr.length());
      client.print("\n\n");
      client.print(postStr);


          client.stop();


    }
  }

}
```

OUTPUT:

Sender:



Receiver:



On the ThinkSpeak