



Automated Irrigation system Applied in Crop Farming (ITC's Green House)

Lecturer : HEL CHANTHON

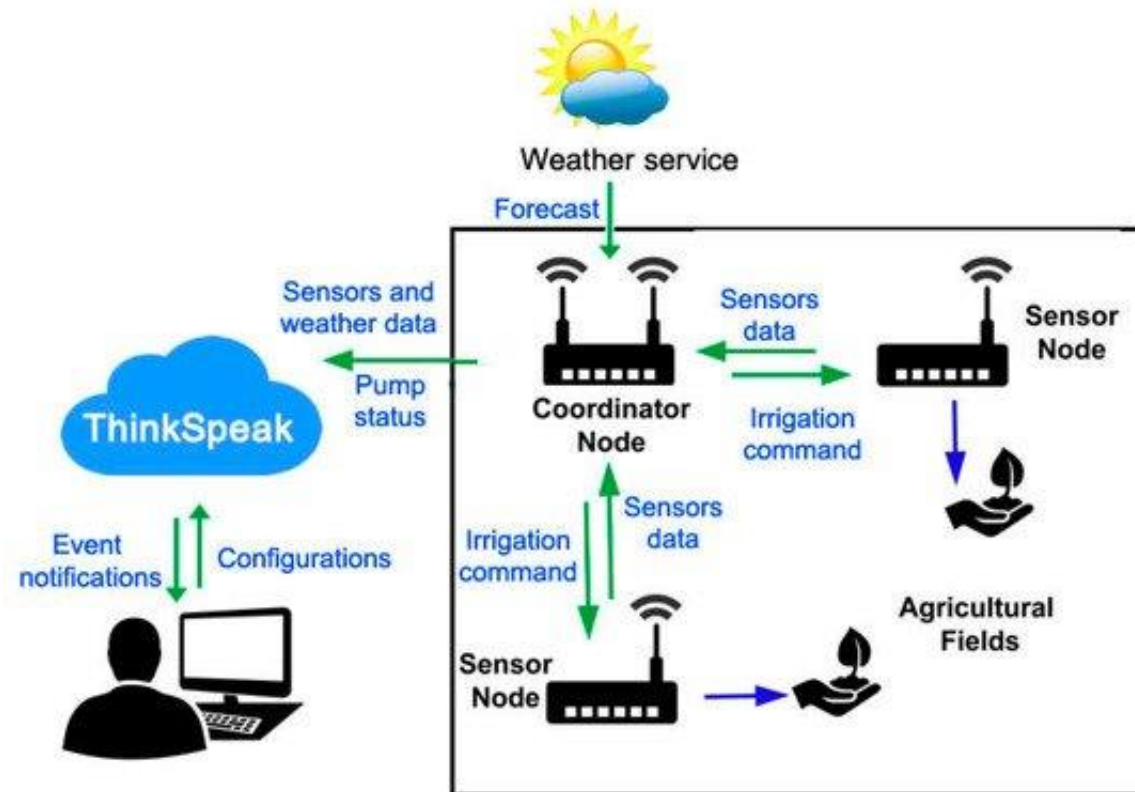
Student : PROEUNG BUNRONG (e20191346)

Department : I3 GTR

❖ The Composition of the system

The system is composed mainly of three parts:

Cloud Platform, Coordinator Node, and Sensor Nodes ...



Project Analysis

❖ Planning for first month

Week 1 6 Aug – 13 Aug	Week2 14 Aug – 20 Aug	Week3 21 Aug – 27 Aug	Week4 28 Aug – 3 Sep
Researching Code and test with dht11 Create charnel Thingspeak and testing.	To understand problem of code, Using Wi-Fi with username Combine code with coordinator and testing then send data to Thingspeak.	Collect data from Greenhouse send to Thingspeak And show all the result of Temperature (T1+T2) Or another Sensor on thingspeak.	Need to Complete planning of first month and continue working for second month to use Camera in greenhouse.
Missing: testing some error, problem to understand something in code			

Project Analysis

❖ Testing code with DHT11

esp8266_and_DHT11_sensor

```
1 #include <SoftwareSerial.h>
2 #include <Adafruit_Sensor.h>
3 #include <DHT.h>
4 #define DHTPIN 8 // what pin we're connected to
5 #define DHTTYPE DHT22 // DHT 22 (AM2302)
6 #define RX 11 // set rx pin
7 #define TX 12 // set tx pin
8 String AP = "GTR Lab"; // AP NAME
9 String PASS = "@gtrlab@"; // AP PASSWORD
10 String API = "6S4RZP5SRTV0RGX0"; // Write API KEY
11 String HOST = "api.thingspeak.com";
12 String PORT = "80";
13 String field1 = "field1";
14 String field2 = "field2";
15 int countTrueCommand;
16 int countTimeCommand;
17 boolean found = false;
18 DHT dht(DHTPIN, DHTTYPE);
19 SoftwareSerial esp8266(RX, TX);
20
21 void setup() {
22   Serial.begin(9600);
23   esp8266.begin(115200);
24   sendCommand("AT", 5, "OK"); // send AT command
25   sendCommand("AT+CWMODE=3", 5, "OK"); // set MODE 1
26   sendCommand("AT+CWJAP=\"" + AP + "\",\"" + PASS + "\"", 20, "OK"); // configure esp8266 to WiFi
27   dht.begin();
28 }
29
30 void loop() {
31   // Wait a few seconds between measurements.
32   delay(2000);
33
34   // Reading temperature or humidity takes about 250 milliseconds!
35   // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
```

esp8266_and_DHT11_sensor

```
35 // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
36 float h = dht.readHumidity();
37 // Read temperature as Celsius
38 float t = dht.readTemperature();
39
40 // Check if any reads failed and exit early (to try again).
41 if (isnan(h) || isnan(t)) {
42   Serial.println("Failed to read from DHT sensor!");
43   return;
44 }
45 String getData1 = "GET /update?api_key="+ API +"&"+ field1 +"="+String(t);// set string for send data to thing speak
46 String getData2 = "GET /update?api_key="+ API +"&"+ field2 +"="+String(h);
47 sendCommand("AT+CIPMUX=1", 5, "OK"); // set to single connection
48 sendCommand("AT+CIPSTART=0,\"TCP\", \"" + HOST + "\", " + PORT, 15, "OK"); // to connect to the Thingspeak API using TCP protocol
49 sendCommand("AT+CIPSEND=0," +String(getData1.length()+4), 4, ">");// for read data and start sending data
50 esp8266.println(getData1); // send data to Thingspeak
51 delay(1500);
52 countTrueCommand++;
53 sendCommand("AT+CIPCLOSE=0", 5, "OK");// for end and close transmission
54 delay(1000);
55 sendCommand("AT+CIPMUX=1", 5, "OK"); // set to single connection
56 sendCommand("AT+CIPSTART=0,\"TCP\", \"" + HOST + "\", " + PORT, 15, "OK"); // to connect to the Thingspeak API using TCP protocol
57 sendCommand("AT+CIPSEND=0," +String(getData2.length()+4), 4, ">");// for read data and start sending data
58 esp8266.println(getData2);
59 delay(1500);
60 countTrueCommand++;
61 sendCommand("AT+CIPCLOSE=0", 5, "OK");// for end and close transmission
62
63 // Serial.print("Humidity: ");
64 // Serial.print(h);
65 // Serial.print(" %\t");
66 // Serial.print("Temperature: ");
67 // Serial.print(t);
68 // Serial.println(" .,*C ");
69 }
```

Full-screen Snip

Project Analysis

esp8266_and_DHT11_sensor

```
67 // Serial.print(t);
68 // Serial.println(" .,*C ");
69 }
70
71 void sendCommand(String command, int maxTime, char readReplay[]) {
72     Serial.print(countTrueCommand);
73     Serial.print(" . at command => ");
74     Serial.print(command);
75     Serial.print(" ");
76     while(countTimeCommand < (maxTime*1))
77     {
78         esp8266.println(command);//at+cipstart
79         if(esp8266.find(readReplay))//ok
80         {
81             found = true;
82             break;
83         }
84
85         countTimeCommand++;
86     }
87
88     if(found == true)
89     {
90         Serial.println("Yes");
91         countTrueCommand++;
92         countTimeCommand = 0;
93     }
94     if(found == false)
95     {
96         Serial.println("Failed");
97         countTrueCommand = 0;
98         countTimeCommand = 0;
99     }
100     found = false;
101 }
```

COM14

14:00:58.161 -> 1. at command => AT+CWMODE=3 Yes
14:00:59.176 -> 2. at command => AT+CWJAP="GTR Lab","@gtrlab@" Yes
14:01:04.172 -> 3. at command => AT+CIPMUX=1 Yes
14:01:05.183 -> 4. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Yes
14:01:05.754 -> 5. at command => AT+CIPSEND=0,53 Failed
14:01:11.287 -> 1. at command => AT+CIPCLOSE=0 Failed
14:01:17.293 -> 0. at command => AT+CIPMUX=1 Yes
14:01:17.328 -> 1. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Failed
14:01:33.630 -> 0. at command => AT+CIPSEND=0,53 Failed
14:01:39.127 -> 1. at command => AT+CIPCLOSE=0 Failed
14:01:46.153 -> 0. at command => AT+CIPMUX=1 Yes
14:01:46.187 -> 1. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Yes
14:01:49.902 -> 2. at command => AT+CIPSEND=0,53 Failed
14:01:55.411 -> 1. at command => AT+CIPCLOSE=0 Failed
14:02:01.418 -> 0. at command => AT+CIPMUX=1 Yes
14:02:02.431 -> 1. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80

☒ Autoscroll ☒ Show timestamp Newline 9600 baud Clear output

10:01:12.578 -> 2. at command => AT+CWJAP="GTR Lab","@gtrlab@" Yes
10:01:19.606 -> 3. at command => AT+CIPMUX=1 Yes
10:01:19.640 -> 4. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Yes
10:01:20.146 -> 5. at command => AT+CIPSEND=0,53 Failed
10:01:25.646 -> 1. at command => AT+CIPCLOSE=0 Failed
10:01:31.698 -> 0. at command => AT+CIPMUX=1 Yes
10:01:31.732 -> 1. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Yes
10:01:32.037 -> 2. at command => AT+CIPSEND=0,53 Yes
10:01:33.532 -> 4. at command => AT+CIPCLOSE=0 Yes
10:01:35.801 -> 5. at command => AT+CIPMUX=1 Yes
10:01:35.835 -> 6. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Yes
10:01:36.140 -> 7. at command => AT+CIPSEND=0,53 Yes
10:01:37.624 -> 9. at command => AT+CIPCLOSE=0 Yes
10:01:38.913 -> 10. at command => AT+CIPMUX=1 Yes
10:01:38.946 -> 11. at command => AT+CIPSTART=0,"TCP","api.thingspeak.com",80 Failed
10:01:55.669 -> 0. at command => AT+CIPSEND=0,53

☒ Autoscroll ☒ Show timestamp Newline 9600 baud Clear output

Done compiling.

Project Analysis

Thingspeak platform

API Keys - ThingSpeak IoT

thingspeak.com/channels/1111041/api_keys

ITC 'S GREEN HOUSE

Channel ID: **1111041**
Author: **mwa0000019124963**
Access: Private

Temperature, humidity, moisture, solar radiation,
dht22, moisture sensor, humidity, temperature,
solar radiation, itc's greenhouse

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Write API Key

Key **654RZP5SRTV0RGX0**

Generate New Write API Key

Read API Keys

Key **W5143W70DUL05G6V**

Note

Channel Settings - ThingSpeak

thingspeak.com/channels/1111041/edit

ITC 'S GREEN HOUSE

Channel ID: **1111041**
Author: **mwa0000019124963**
Access: Private

Temperature, humidity, moisture, solar radiation,
dht22, moisture sensor, humidity, temperature,
solar radiation, itc's greenhouse

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Channel Settings

Percentage complete 70%

Channel ID **1111041**

Name **ITC 'S GREEN HOUSE**

Description **Temperature, humidity, moisture, solar radiation,**

Field 1 **Temperature 1** ☒

Field 2 **Humidity 1** ☒

Field 3 **Moisture 1** ☒

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.

ITC 'S GREEN HOUSE - ThingSpeak

thingspeak.com/channels/1111041/private_show

Field 1 Chart

ITC 'S GREEN HOUSE

Temperature 1

34

32

18:00 13. Aug 06:00

Temperature 1: 31.6
Thu Aug 13 20:20
11:20:42 GMT+0700

ThingSpeak.com

Temperature 1

32

42 minutes ago

Field 2 Chart

ITC 'S GREEN HOUSE

Humidity 1

50

25

18:00 13. Aug 06:00

ThingSpeak.com

Humidity 1

62

40 minutes ago



Thank You