



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

Lecturer : HEL CHANTHAN

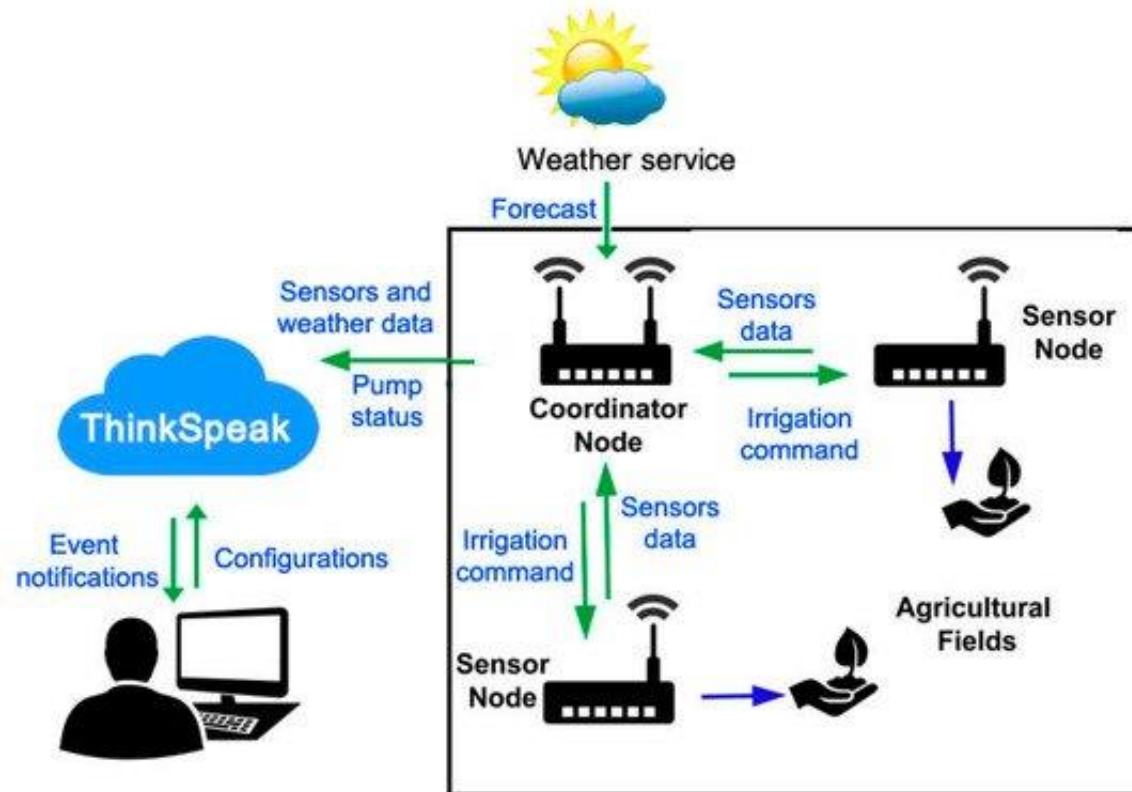
Student : PROEUNG BUNRONG (e20191346)

Department : I4 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 | | | |

❖ Outline

- System Working
- Code Coordinator
- Code NodeMCU
- ThingSpeak and LCD
- Export Value

❖ System Working

Collecting data : Temperature, Humidity, Moisture, Solar Radiation from ITC GREENHOUSE. We have two sensor nodes working in Greenhouse and a Coordinator collected data from those nodes sending data to ThingSpeak using NodeMCU by delay 2minutes.

❖ Code Coordinator

Project Analysis

```
Coordinator_for_GHITC
1 #include <SoftwareSerial.h>
2 #include <LiquidCrystal_I2C.h>
3 #include <XBee.h>
4 #include <Wire.h>
5
6 HardwareSerial &XBee = Serial;
7 //SoftwareSerial XBee (0,1);
8 SoftwareSerial esp(8,9);
9 // Sensor node data transfer testing 2 parameters
10 // |Header|Node ID|Humidity|Temp|moisture|radiation|Checksum|
11 //   1       1       4       4       4       4       2   = 20 byte
12
13 typedef struct {
14     uint8_t header;
15     uint8_t node_id;
16     float humidity;
17     float temperature;
18     float moisture;
19     float radiation;
20     uint16_t checksum;
21 } data_struct_t;
22
23 typedef union {
24     data_struct_t data_struct;
25     uint8_t data_byte[20];
26 } packet_t;
27
28 float hum1 = 0.0;
29 float temp1 = 0.0;
30 float hum2 = 0.0;
31 float temp2 = 0.0;
32 //float mois1 = 0.0;
33 //float mois2 = 0.0;
34 float rad = 0.0;
35
Coordinator_for_GHITC
35
36 //Define LCD
37 LiquidCrystal_I2C lcd (0x3F, 20, 4);
38
39 //Define relay pin
40 #define relayPin 5
41 //Define mois1 and mois2, mois1 and 2 should more than
42 float mois1 = 50.0;
43 float mois2 = 50.0;
44 //unsigned long pre_time;
45 //unsigned long cur_time;
46 void setup() {
47     // put your setup code here, to run once:
48     XBee.begin(9600);
49     Serial.begin(9600);
50     esp.begin(115200);
51     lcd.begin();
52     lcd.backlight();
53
54     //Set relay pin
55     pinMode(relayPin, OUTPUT);
56
57     //Set title
58     lcd.setCursor(4,1);
59     lcd.print("NEVER GIVE UP");
60     delay(5000);
61     lcd.clear();
62 }
63
64
65 void loop() {
66     packet_t coord;
67     if (XBee.available() > 0) {
68         //Set separete display
69         lcd.setCursor(9,0);
68         //Set separete display
69         lcd.setCursor(9,0);
69         lcd.setCursor(9,0);
70         lcd.print("|");
71         lcd.setCursor(9,1);
72         lcd.print("|");
73         lcd.setCursor(9,2);
74         lcd.print("|");
75         lcd.setCursor(9,3);
76         lcd.print("|");
77
78         coord.data_byte[0] = XBee.read();
79         if (coord.data_struct.header == 0x55) {
80             Serial.print(coord.data_byte[0], HEX); Serial.print(",");
81             for (int i = 1; i < 20; i++) {
82                 delay(15);
83                 coord.data_byte[i] = XBee.read();
84
85                 Serial.print(coord.data_byte[i], HEX); Serial.print(",");
86             }
87             //write data to esp
88             esp.write(coord.data_byte, 20);
89
90             //Print data from Node 1 on LCD
91             if (coord.data_struct.node_id == 1)
92             {
93                 lcd.setCursor(0,0);
94                 lcd.print("H1:");
95                 lcd.setCursor(3,0);
96                 lcd.print(coord.data_struct.humidity, 2);
97                 lcd.setCursor(0,1);
98                 lcd.print("T1:");
99                 lcd.setCursor(3,1);
100                 lcd.print(coord.data_struct.temperature, 2);
101                 lcd.setCursor(0,2);
102                 lcd.print("M1:");
```

Project Analysis

```
Coordinator_for_GHITC
101     lcd.setCursor(0,2);
102     lcd.print("M1:");
103     lcd.setCursor(3,2);
104     lcd.print(coord.data_struct.moisture, 2);
105
106     hum1 = coord.data_struct.humidity;
107     temp1 = coord.data_struct.temperature;
108     mois1 = coord.data_struct.moisture;
109 }
110
111 //Print data from Node 2 on LCD
112 if (coord.data_struct.node_id == 2)
113 {
114     lcd.setCursor(11,0);
115     lcd.print("H2:");
116     lcd.setCursor(14,0);
117     lcd.print(coord.data_struct.humidity, 2);
118     lcd.setCursor(11,1);
119     lcd.print("T2:");
120     lcd.setCursor(14,1);
121     lcd.print(coord.data_struct.temperature, 2);
122     lcd.setCursor(11,2);
123     lcd.print("M2:");
124     lcd.setCursor(14,2);
125     lcd.print(coord.data_struct.moisture, 2);
126     lcd.setCursor(11,3);
127     lcd.print("R:");
128     lcd.setCursor(14,3);
129     lcd.print(coord.data_struct.radiation, 1);
130
131     hum2 = coord.data_struct.humidity;
132     temp2 = coord.data_struct.temperature;
133     mois2 = coord.data_struct.moisture;
134     rad = coord.data_struct.radiation;
135 }
```

```
Coordinator_for_GHITC
125     lcd.print(coord.data_struct.moisture, 2);
126     lcd.setCursor(11,3);
127     lcd.print("R:");
128     lcd.setCursor(14,3);
129     lcd.print(coord.data_struct.radiation, 1);
130
131     hum2 = coord.data_struct.humidity;
132     temp2 = coord.data_struct.temperature;
133     mois2 = coord.data_struct.moisture;
134     rad = coord.data_struct.radiation;
135 }
136
137 //Define when to start the Valve
138 if (mois1 <= 32.00 or mois2 <= 32.00)
139 {
140     digitalWrite(relayPin, HIGH);
141 }
142
143 //Define when to stop the Valve
144 if (mois1 >= 34.00 and mois2 >= 34.00)
145 {
146     digitalWrite(relayPin, LOW);
147 }
148 Serial.println();
149 Serial.print("Temp1: "); Serial.println(temp1);
150 Serial.print("Hum1: "); Serial.println(hum1);
151 Serial.print("Mois1: "); Serial.println(mois1);
152
153 Serial.print("Temp2: "); Serial.println(temp2);
154 Serial.print("Hum2: "); Serial.println(hum2);
155 Serial.print("Mois2: "); Serial.println(mois2);
156 Serial.print("Rad: "); Serial.println(rad);
157 }
158 }
159 }
```

Project Analysis

❖ Code NodeMCU

```
NodeMCU_for_GHITC
1 #include <SoftwareSerial.h>
2 #include <ESP8266WiFi.h> //Library connecting ESP8266 to connect
3 WiFiClient client;
4 String apiKey = "8800FGWCTW33Q0I6"; //Write API key of your Thingspeak channel
5 const char* server = "api.thingspeak.com"; // API for thingspeak
6 const char* ssid = "GTR_LAB"; // Wifi SSID of your Internet connection
7 const char* pass = "@gtrlab@"; // Password
8 unsigned long pre_time;
9 unsigned long cur_time;
10 SoftwareSerial esp(D6,D5); //RX,TX
11 // Sensor node data transfer testing 2 parameters
12 // |Header|Node ID|Humidity|Temp|moisture|radiation|Checksum|
13 //   1      1      4      4      4      4      2      = 20 byte
14
15 typedef struct __attribute__((__packed__)) {
16     uint8_t header;
17     uint8_t node_id;
18     float humidity;
19     float temperature;
20     float moisture;
21     float radiation;
22     uint16_t checksum;
23 } data_struct_t;
24
25 typedef union {
26     data_struct_t data_struct;
27     uint8_t data_byte[20];
28 } packet_t;
29
30 float hum1 = 0.0;
31 float temp1 = 0.0;
32 float hum2 = 0.0;
33 float temp2 = 0.0;
34 float mois1 = 0.0;
35 float mois2 = 0.0;
36
37
38 packet_t coord;
39
40 void setup() {
41     esp.begin(115200);
42     Serial.begin(9600);
43     // delay(1000);
44     Serial.println("Trying to Connect with ");
45     Serial.println(ssid);
46     WiFi.begin(ssid, pass); // Connecting ESP8266 with Internet enabled
47     while (WiFi.status() != WL_CONNECTED)
48     {
49         delay(500);
50         Serial.print("connection was unsuccesfull.");
51     }
52     Serial.println("");
53     Serial.println("WiFi connected.");
54 }
55
56 void loop() {
57     if (esp.available()>0){
58         coord.data_byte[0] = esp.read();
59         if (coord.data_struct.header == 0x55)
60         {
61             // Serial.print("");
62             for (int i=1; i<20; i++)
63             {
64                 coord.data_byte[i] = esp.read();
65             }
66             for (int i = 0; i < 20; i++){
67                 Serial.print(coord.data_byte[i], HEX); Serial.print(",");
68             }
69             // Serial.println();
```


NodeMCU_for_GHITC

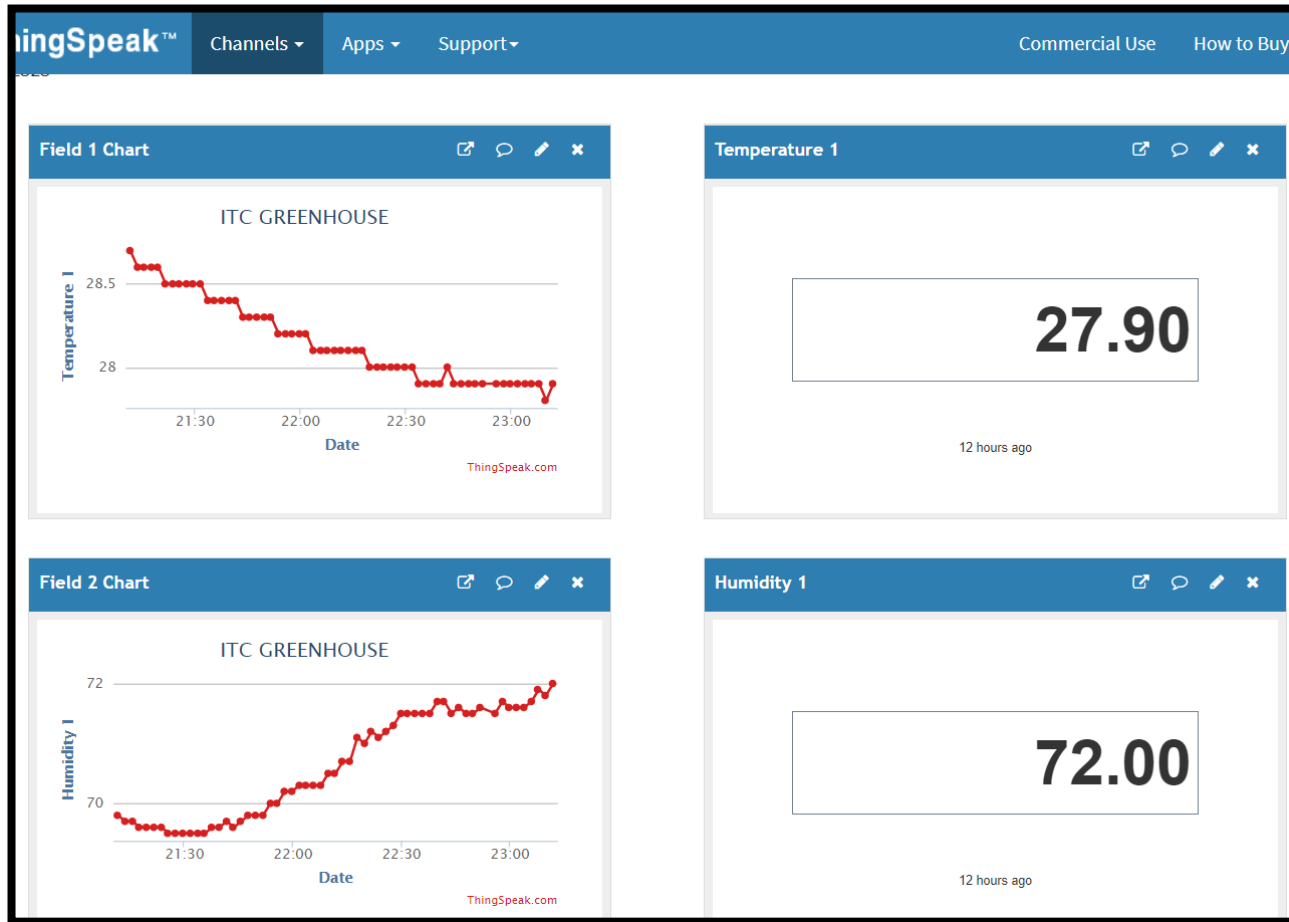
```
68 }  
69 // Serial.println();  
70 Serial.println(coord.data_struct.node_id);  
71 if (coord.data_struct.node_id == 1){  
72     hum1 = coord.data_struct.humidity;  
73     temp1 = coord.data_struct.temperature;  
74     mois1= coord.data_struct.moisture;  
75 }  
76 if (coord.data_struct.node_id == 2)  
77 {  
78     hum2 = coord.data_struct.humidity;  
79     temp2 = coord.data_struct.temperature;  
80     mois2 = coord.data_struct.moisture;  
81     rad = coord.data_struct.radiation;  
82 }  
83 // Serial.println("node1");  
84 // Serial.print("Temp1: "); Serial.println(temp1);  
85 // Serial.print("Hum1: "); Serial.println(hum1);  
86 // Serial.print("Mois1: "); Serial.println(mois1);  
87 // Serial.println("node2");  
88 // Serial.print("Temp2: "); Serial.println(temp2);  
89 // Serial.print("Hum2: "); Serial.println(hum2);  
90 // Serial.print("Mois2: "); Serial.println(mois2);  
91 // Serial.print("Rad: "); Serial.println(rad);  
92 }  
93 cur_time = millis();  
94 if(cur_time - pre_time >=120000){  
95     if (client.connect(server,80)) // "184.106.153.149" or api.thingspeak.com  
96     {  
97         // Format of DATA Packet "Write API Key&field1=Temperature data&field2=  
98         String postStr = apiKey;{  
99         //node1  
100         postStr += "&field1=";  
101         postStr += String(temp1);  
102         postStr += "&field2=";
```

NodeMCU_for_GHITC

```
102     postStr += "&field2=";  
103     postStr += String(hum1);  
104     postStr += "&field3=";  
105     postStr += String(mois1);  
106  
107     //node2  
108     postStr += "&field4=";  
109     postStr += String(temp2);  
110     postStr += "&field5=";  
111     postStr += String(hum2);  
112     postStr += "&field6=";  
113     postStr += String(mois2);  
114     postStr += "&field7=";  
115     postStr += String(rad);  
116     postStr += "\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n";  
117  
118     client.print("POST /update HTTP/1.1\n");  
119     client.print("Host: api.thingspeak.com\n");  
120     client.print("Connection: close\n");  
121     client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");  
122     client.print("Content-Type: application/x-www-form-urlencoded\n");  
123     client.print("Content-Length: ");  
124     client.print(postStr.length());  
125     client.print("\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n");  
126     client.print(postStr);  
127     client.flush();  
128 // Serial.println(postStr);  
129 // delay(10000);  
130     client.stop();  
131 // Serial.println("Waiting to initiate next data packet...");  
132 // delay(10000);  
133 }  
134 }  
135 pre_time = cur_time;  
136 }
```

Project Analysis

❖ ThingSpeak and LCD



Project Analysis

❖ Export Value

28.oct.2020 - 31.oct.2020 - Excel (Product Activation Failed)

| | A | B | C | D | E | F | G | H | I | J |
|----|-------------------------|----------|--------------|-----------|-----------|--------------|-----------|-----------|-----------------|--------|
| 1 | created_at | entry_id | Temperature1 | Humidity1 | Moisture1 | Temperature2 | Humidity2 | Moisture2 | Solar Radiation | field8 |
| 2 | 2020-10-28 15:43:14 +07 | 1 | 0 | 0 | 0 | 29.6 | 67.5 | -15 | 23 | |
| 3 | 2020-10-28 15:44:06 +07 | 2 | 0 | 0 | 0 | 29.6 | 67.5 | -15 | 23 | |
| 4 | 2020-10-28 15:45:39 +07 | 3 | 0 | 0 | 0 | 29.6 | 67.2 | -18 | 0 | |
| 5 | 2020-10-28 15:46:00 +07 | 4 | 0 | 0 | 0 | 29.6 | 67.2 | -18 | 0 | |
| 6 | 2020-10-28 15:46:24 +07 | 5 | 0 | 0 | 0 | 29.6 | 67.2 | -18 | 0 | |
| 7 | 2020-10-28 16:06:07 +07 | 6 | 30.3 | 64.9 | 3 | 28.9 | 68.7 | -18 | 0 | |
| 8 | 2020-10-28 16:06:28 +07 | 7 | 30.2 | 65.7 | 3 | 28.9 | 68.7 | -18 | 0 | |
| 9 | 2020-10-28 16:07:10 +07 | 8 | 30.2 | 65.7 | 3 | 28.8 | 68.1 | -17 | 2 | |
| 10 | 2020-10-28 16:42:01 +07 | 9 | 30.2 | 63.6 | 39 | 28.3 | 70.8 | 8 | 17 | |
| 11 | 2020-10-28 16:51:50 +07 | 10 | 30.1 | 63.8 | 38 | 28.2 | 72.8 | 39 | 14 | |
| 12 | 2020-10-28 17:38:21 +07 | 11 | 29.1 | 66.7 | 38 | 0 | 0 | 0 | 0.00 | |
| 13 | 2020-10-28 17:40:21 +07 | 12 | 29.1 | 66.9 | 39 | 0 | 0 | 0 | 0.00 | |
| 14 | 2020-10-28 17:42:21 +07 | 13 | 29.1 | 66.9 | 38 | 27.3 | 75.6 | 40 | 0.00 | |
| 15 | 2020-10-28 17:49:12 +07 | 14 | 29 | 67.2 | 38 | 27.3 | 78.3 | 40 | 0.00 | |
| 16 | 2020-10-28 17:51:13 +07 | 15 | 28.9 | 67.2 | 39 | 27.3 | 78.6 | 40 | 0.00 | |
| 17 | 2020-10-28 17:53:13 +07 | 16 | 28.9 | 67.3 | 38 | 27.4 | 78.6 | 40 | 0.00 | |
| 18 | 2020-10-28 17:55:13 +07 | 17 | 28.9 | 67.5 | 38 | 27.4 | 78.5 | 40 | 0.00 | |
| 19 | 2020-10-28 17:57:13 +07 | 18 | 28.8 | 67.5 | 39 | 27.5 | 78.1 | 40 | 0.00 | |
| 20 | 2020-10-28 17:59:14 +07 | 19 | 28.8 | 67.7 | 38 | 27.5 | 77.5 | 40 | 0.00 | |
| 21 | 2020-10-28 18:01:14 +07 | 20 | 28.8 | 67.8 | 38 | 27.6 | 77.3 | 40 | 0.00 | |
| 22 | 2020-10-28 18:03:14 +07 | 21 | 28.7 | 67.8 | 39 | 27.5 | 77.1 | 40 | 0.00 | |
| 23 | 2020-10-28 18:05:14 +07 | 22 | 28.8 | 68 | 39 | 27.5 | 76.6 | 40 | 0.00 | |
| 24 | 2020-10-28 18:07:15 +07 | 23 | 28.7 | 68 | 38 | 27.5 | 76.3 | 40 | 0.00 | |
| 25 | 2020-10-28 18:09:15 +07 | 24 | 28.7 | 68 | 38 | 27.5 | 75.9 | 40 | 0.00 | |
| 26 | 2020-10-28 18:11:15 +07 | 25 | 28.7 | 68 | 38 | 27.5 | 75.4 | 40 | 0.00 | |
| 27 | 2020-10-28 18:13:16 +07 | 26 | 28.6 | 67.9 | 38 | 27.5 | 75.4 | 40 | 0.00 | |
| 28 | 2020-10-28 18:15:16 +07 | 27 | 28.7 | 68 | 38 | 27.5 | 75.5 | 40 | 0.00 | |
| 29 | 2020-10-28 18:17:16 +07 | 28 | 28.6 | 68 | 38 | 27.6 | 75.7 | 40 | 0.00 | |
| 30 | 2020-10-28 18:19:16 +07 | 29 | 28.6 | 68 | 38 | 27.5 | 75.9 | 40 | 0.00 | |

28.oct.2020 - 31.oct.2020

Private View Public View Channel Settings Sharing API Keys

Import

Upload a CSV file to import data into this channel.

File No file chosen

Time Zone (GMT-05:00) Eastern Time (US & Canada)

Export

Download all of this Channel's feeds in CSV format.

Time Zone (GMT-05:00) Eastern Time (US & Canada)

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

Channel Stats

Created: 3 months ago

Last entry: less than a minute ago

Entries: 2333



Thank You