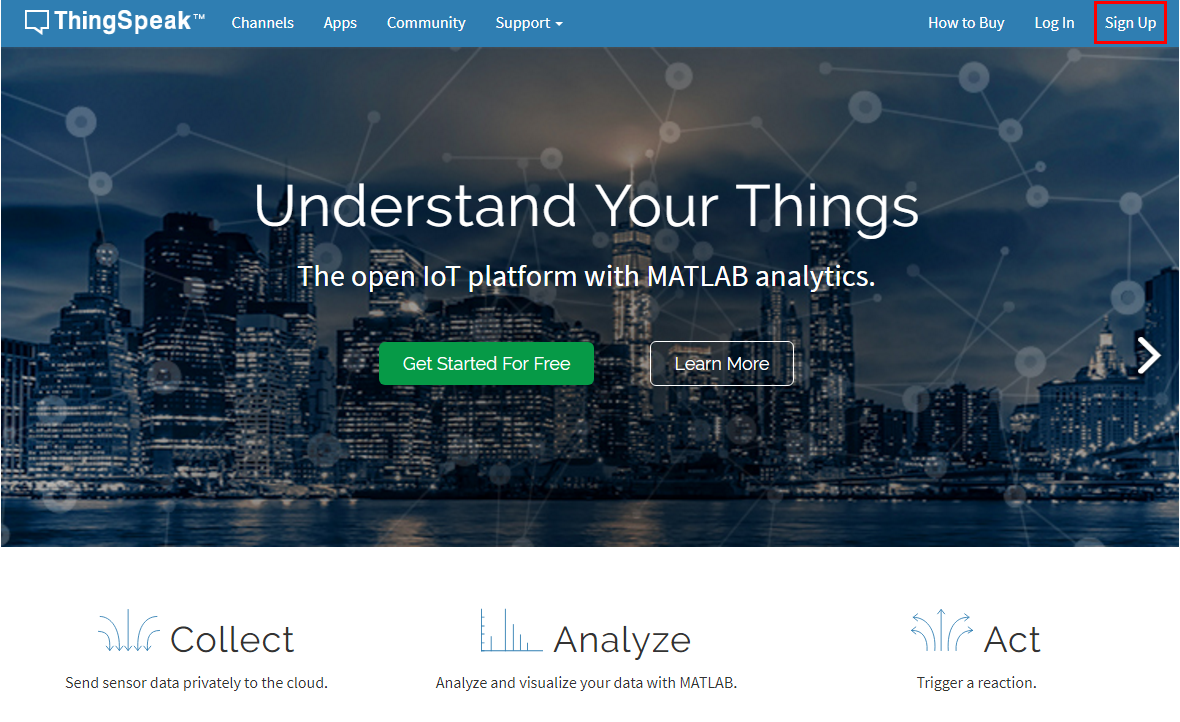
Thing Speak 클라우드 연동

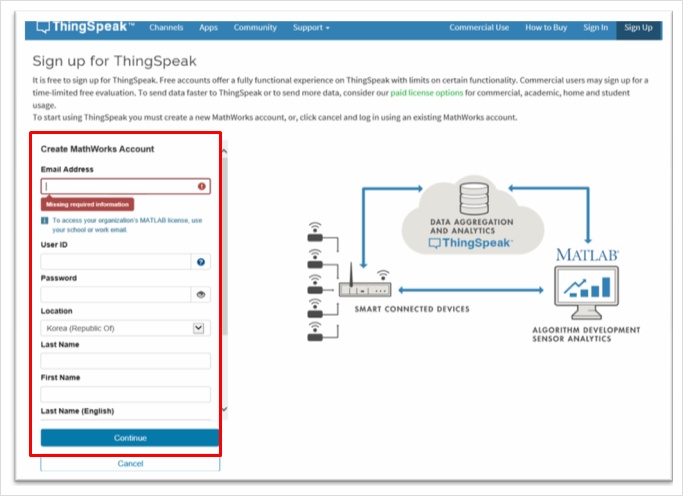
Thing Speak 클라우드는 2010년 미국의 ioBridge에서 만들어 졌습니다

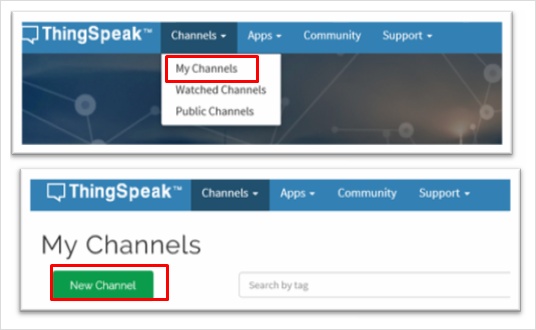
Thing Speak의 통신방식은 TCP/IP방식을 사용하며 기본으로는 무료로 이용이 가능하지만 83000개의 데이터가 넘어가면 유료로 전환이 됩니다 그리고 Thing Speak만의 장점으로 내가만든 라이브러리 뿐만 아니라 다른사람이 만든 라이브러리를 쉽게 가져와서 볼 수 있고 내용을 들여다볼수도 있습니다

또 Thing Speak은 아두이노를 사용하면서 사용하기에 가장 적합한 클라우드로서 아두이노 내에서 라이브러리를 제공할 만큼 아두이노 사용자에게 사용이 편리한 클라우드 서비스 입니다

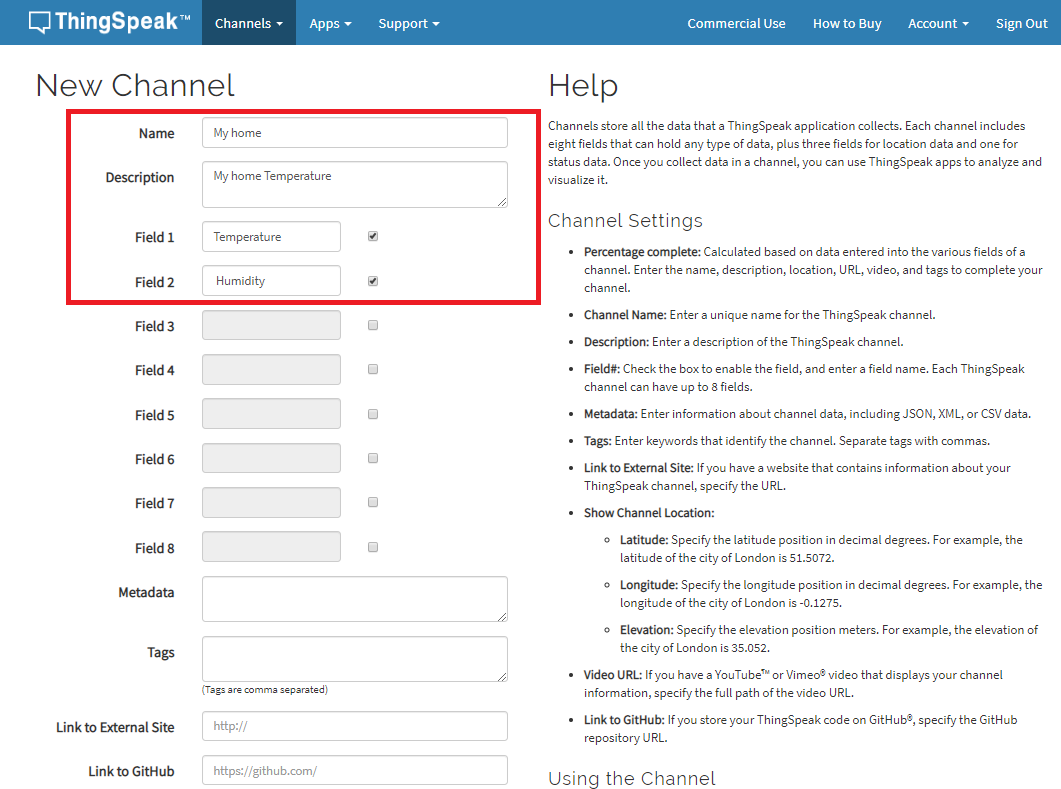
Thing Speak은 회원만이 사용을 할수 있기 떄문에 홈페이지에 들어가면 가장먼저 회원가입부터 해야 합니다

위에서 보이듯이 홈페이지 우측 상단에 있는 Sign Up버튼을 클릭해서 아이디 생성에 들어가 줍니다

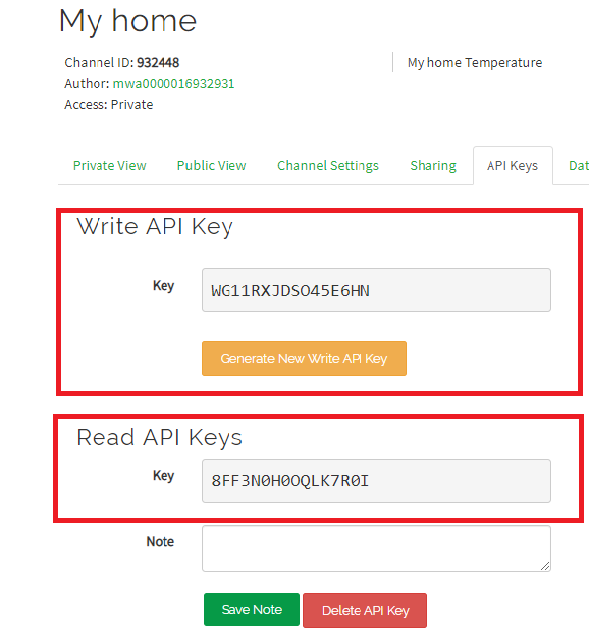
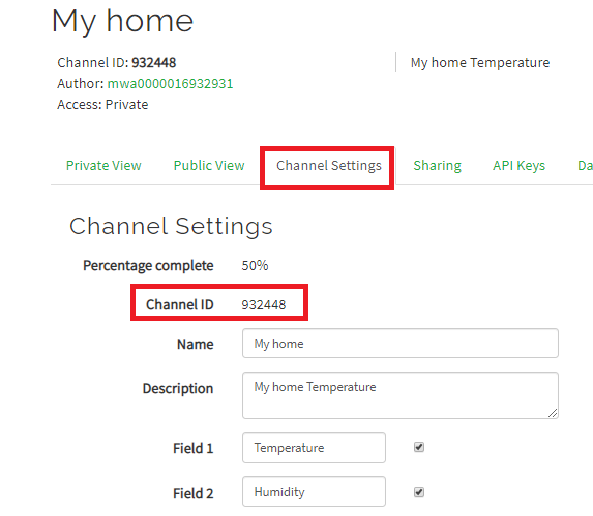


들어와서 자신이 사용할 이메일 주소와 아이디, 비밀번호 자신이 거주하고 있는 나라를 설정하고 자신의 성 과 이름을 입력하면 간단하게 회원가입을 하실수 있습니다

회원가입이 완료 되었으면 채널을 만들어서 사용할수 있습니다 채널은 총9개 생성 가능하며 하나의 채널에는 8개의 속성들을 추가해서 사용할수 있습니다



채널 생성 방법은 먼저 채널의 이름을 설정해 주어야 합니다 이름은 자신이 원하는 것을 사용하면 됩니다 그후 Description은 이름의 세부사항이라고 생각하시면 됩니다 예를들어 집 온도와 습도를 측정해주는 클라우드면 위에 우리집 이라고 적으시고 아래쪽에 우리집 온도측정 이런식으로 적으시면 됩니다

그리고 3번쨰 칸에있는 Field칸에 자신이 원하는 속성들을 추가하실수 있습니다 저희가 만든 것은 온습도 측정이라 1번칸에는 온도를, 2번칸에는 습도를 추가해 주었습니다

채널을 만든후 채널의 ID와 채널의 API Key를 확인하실수 있는데 이것은 후에 아두이노와 ThingSpeak를 연동하는데 있어 반드시 있어야 하는것이므로 숙지해두어야 합니다

#include <WiFiClient.h>

#include <ThingSpeak.h>

#define DHTPIN D5 // GPIO 14

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// wifi

const char\* ssid = "Simsunghwan";

const char\* password = "!Asd1375692";

WiFiClient client;

// thingSpeak

unsigned long ChannelID = 932448;

const char\* WriteAPIKey = "WG11RXJDSO45E6HN";

//DHT

uint8\_t temperature, humidity;

unsigned long lastCheck;

void setup() {

Serial.begin(115200);

delay(10);

initWiFi();

ThingSpeak.begin(client);

lastCheck = 0;

}

void loop() {

float c\_time = 60000 \* 0.1;

if(millis() - lastCheck > c\_time) {

static boolean data\_state = false;

temperature = dht.readTemperature();

humidity = dht.readHumidity();

Serial.print("Temperature : ");

Serial.print(temperature);

Serial.println("C");

Serial.print("Humidity : ");

Serial.print(humidity);

Serial.println("%");

if(data\_state){

ThingSpeak.writeField(ChannelID, 1, temperature, WriteAPIKey);

data\_state = true;

} else {

ThingSpeak.writeField(ChannelID, 2, humidity, WriteAPIKey);

data\_state = true;

}

lastCheck = millis();

}

delay(100);

}

void initWiFi(){

Serial.println();

Serial.println();

Serial.println("Connectiong to ssid ...");

// attempt to connect to WiFi network

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED){

delay(500);

Serial.println(".");

}

// 접속성공!

Serial.println();

Serial.println("Connected WiFi");

//사용하는 ip출력

Serial.println(WiFi.localIP());

Serial.println();

}

이것은 제가 클라우드를 생성하고 아두이노와 연동할떄 사용한 소스입니다 가장위를 보시면

unsigned long ChannelID = 932448;

const char\* WriteAPIKey = "WG11RXJDSO45E6HN";

이부분에 조금전에 말씀드렸던 클라우드의 채널 ID 와 채널 API Key가 사용된 것을 확인할 수 있는데 반 듯이 있어야 하는 부분입니다 또 이부분은 틀리기 쉽기 떄문에 복사해서 사용하는것을 권잡합니다 그리고

const char\* ssid = "Simsunghwan";

const char\* password = "!Asd1375692";

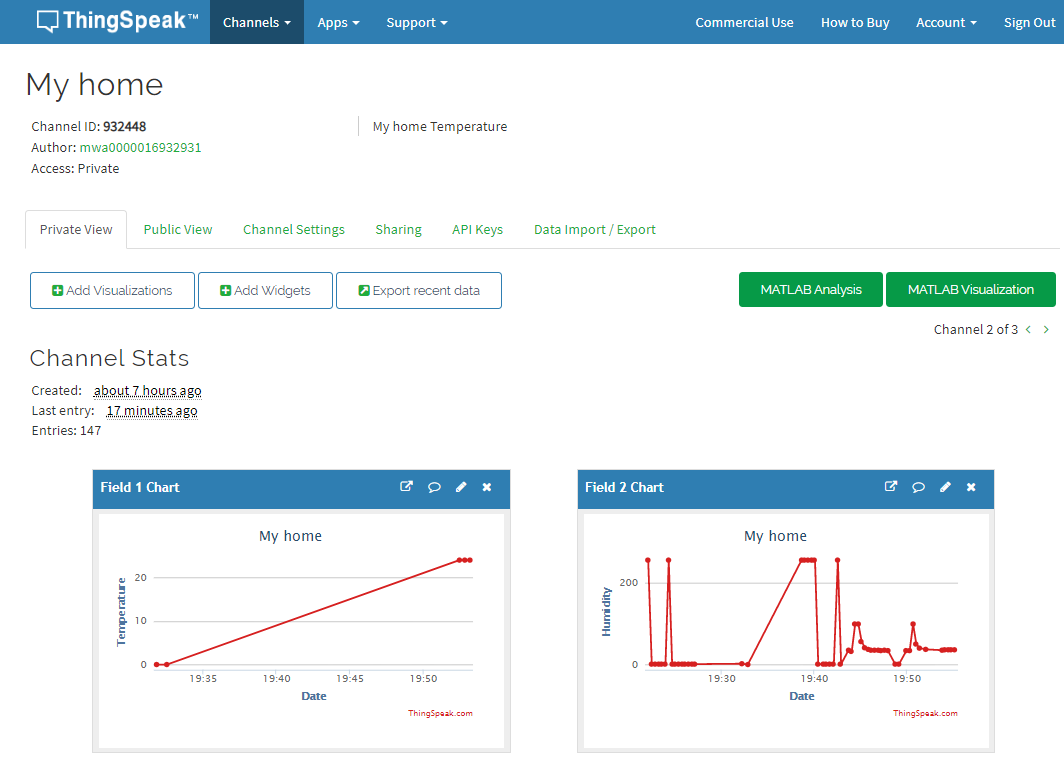
WiFiClient client;

이부분에서 확인할 수 있는 것은 자신이 현재 와이파이를 이용해서 클라우드와 연동하고 있을 때 자신이 사용하고있는 와이파이의 이름과 와이파이의 비밀번호를 한글자도 틀리지 않게 잘 적으셔야 합니다 여기서 한글자라도 틀리면 무조건 에러가 발생하기 때문입니다

그리고 이것이 제가 클라우드 서버와 와이파이로 연결했을떄 받아온 데이터 값입니다

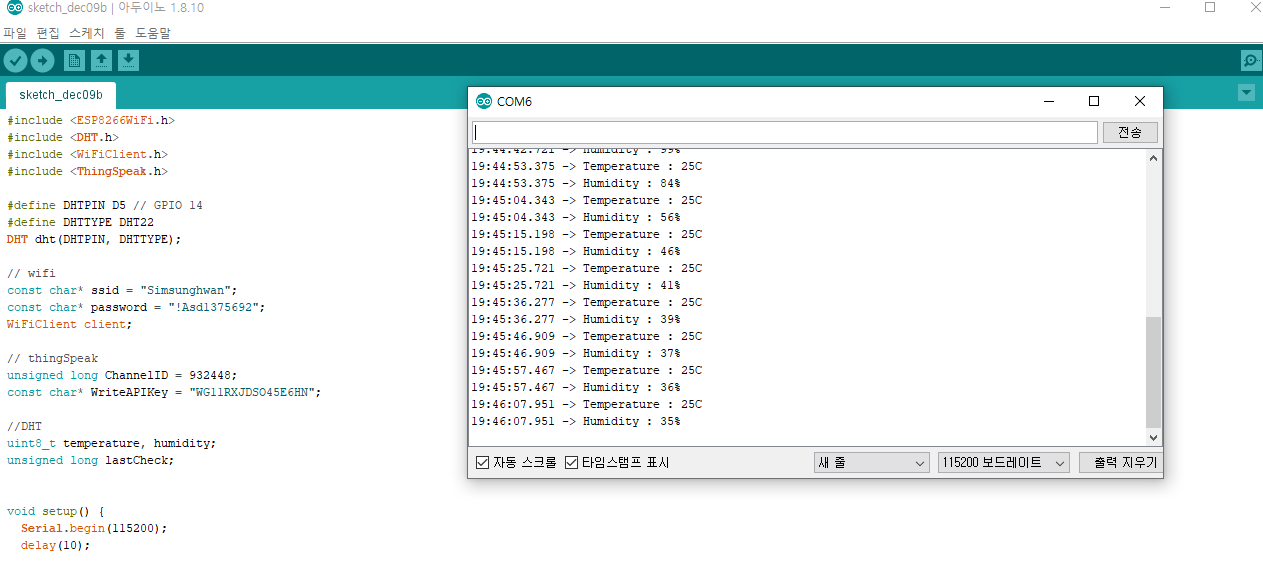
Thing Speak 클라우드의 장점중 하나인데 바로 시각화가 잘 되있다는 것입니다

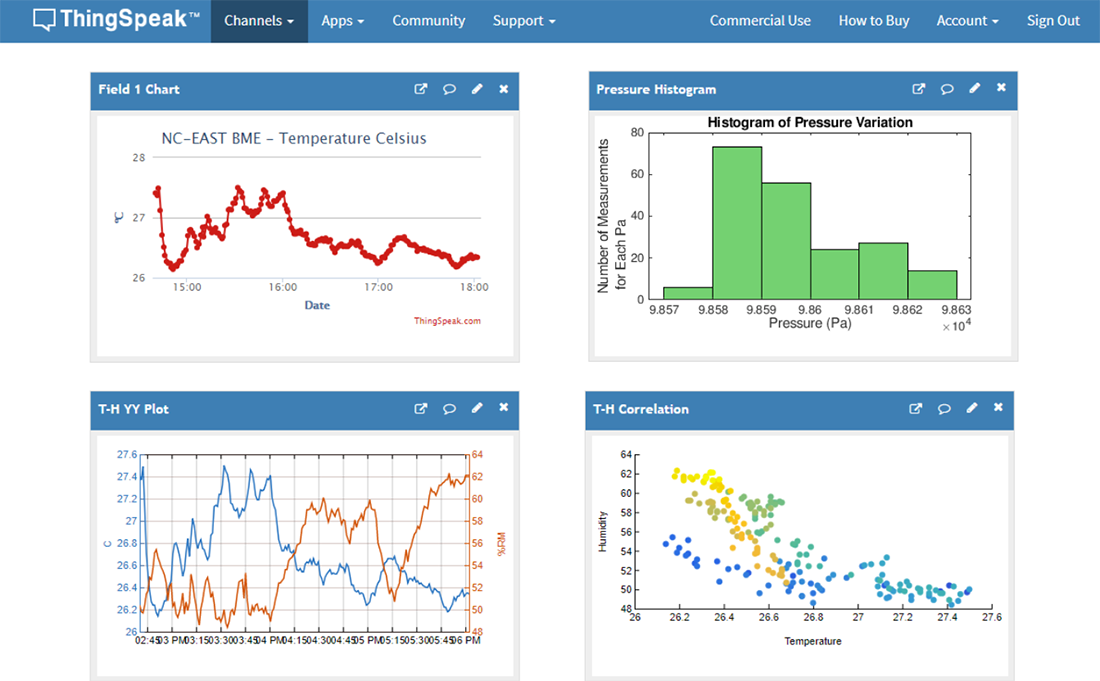
지금부터는 제가 만든 클라우드 서버의 문제점에 대해서 설명드리고 제가 구현하고자 했던 마지막을 보여드리겠습니다

보기에도 오른쪽과 왼쪽이 다르다는게 한눈에 들어오는데 오른쪽은 데이터가 변동이 있으며 초반에 들어온 오류 데이터를 제외하면 데이터가 잘 들어오고 있다는 것이 보이는데 왼쪽의 그래프를 보시면 오류 데이터만 들어오고 그뒤에 제대로된 데이터 값이 들어오지 않는 것을 볼 수 있습니다 처음 아두이노 설정을 잘못했는거 아닌가 라고 생각하실수도 있으신데 아두이노에서는 정상으로 센서의 데이터 값을 받아왔습니다 그 후 소스 코딩도 확인을 해봤지만 특별히 틀린점을 찾지는 못했습니다

밑에 사진은 제가 아두이노에서 받아온 센서의 데이터 값입니다

보다싶이 아두이노 에서의 센서는 정상적으로 온도와 습도를 표현하고 있지만 제가 클라우드 서버로 옮겨왔을 떄에는 습도만 정상적으로 들어오고 온도가 정상적으로 표현되고 있지 않다는 것을 확인했습니다

이 데이터 값이 정상적으로 들어오지 않아서 제가 원래 목표로 했던 MATLAB 이 표현되지 않았습니다 그래서 원래 표현할려고 했던것보다 그래프다 단순해졌고 더 다양한 관측에서 볼 수 있는 방법이 쉽게 제공되어 있었지만 사용하지 못했습니다



정상적으로 MATLAB 까지 연동을 했다면 이런식으로 그래프가 나오게 됩니다

결론적으로 제가 만든 클라우드 시스템은 완성하지 못했습니다 소스코딩이 잘못된 것인가 싶어서 구글링도 해보고 다른사람 코딩 소스도 확인해보고 했지만 틀린점을 발견하지는 못했습니다

감사합니다

클라우드 학기말 보고서 201595004 김광현