

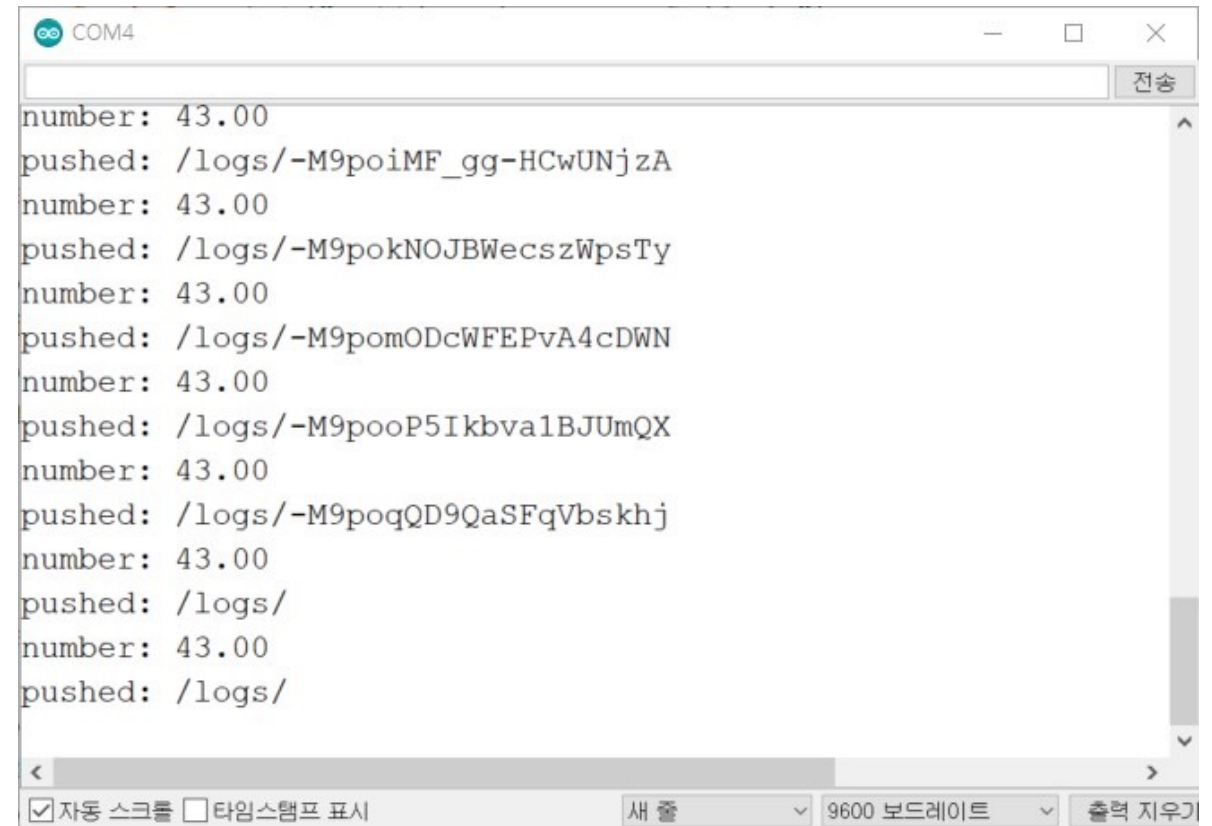
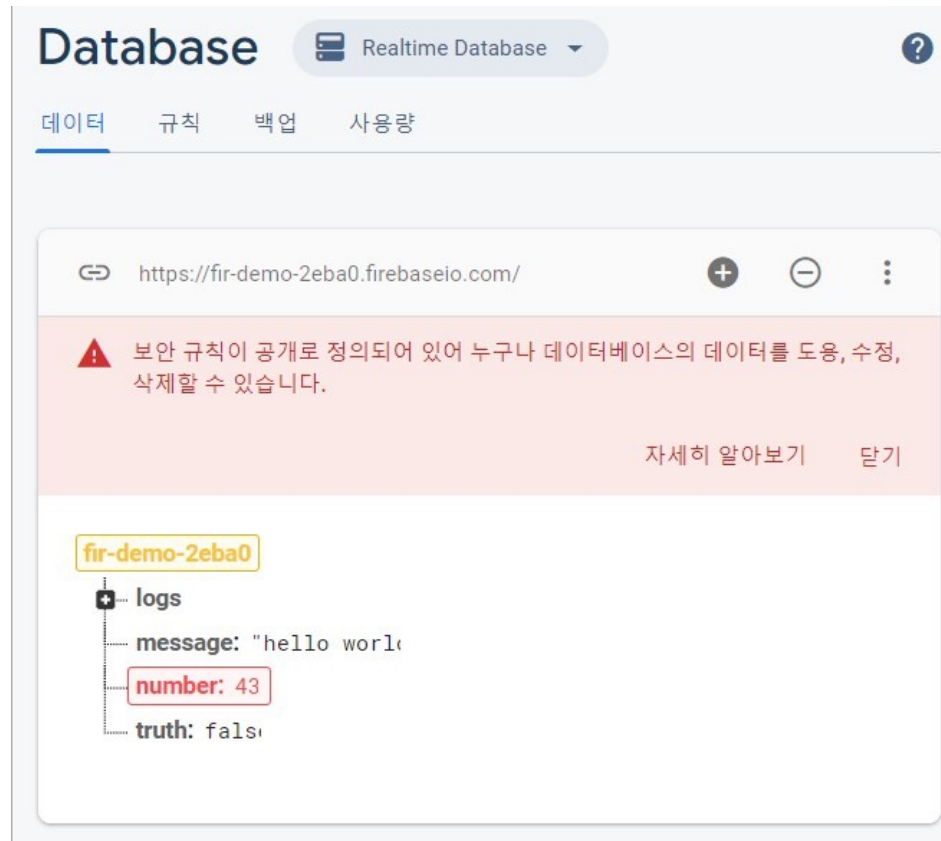
구글 Firebase, NodeMCU를 이용한 IoT 기반 온습도 모니터링

Week 10

Firestore-Ardunio API 살펴보기

- FirebaseDemo_ESP8266

https://github.com/IoT-Lab-02/firebase-arduino/tree/master/examples/FirebaseDemo_ESP8266



Firebase-Ardunio 라이브러리 API

함수	설명	비고
setInt()	Firebase DB에 정수형(Int) 값 저장	
setFloat()	Firebase DB에 실수형(Float) 값 저장	
setString()	Firebase DB에 문자형(String) 값 저장	
setBool()	Firebase DB에 Bool 값 저장	Firebase.setBool("truth",false)
getInt()	Firebase DB에 정수형(int) 값 읽어 오기	
getFloat()	Firebase DB에 실수형(Float) 값 읽어 오기	
getString()	Firebase DB에 문자형(String) 값 읽어 오기	
pushInt()	Firebase DB에 정수형(Int) 값 저장	list 값 append
pushFloat()	Firebase DB에 실수형(Float) 값 저장	list 값 append
pushString()	Firebase DB에 문자형(String) 값 저장	list 값 append
Push()	Firebase DB에 객체형(Object) 값 저장	
remove()	Firebase DB에 값 삭제하기	
failed(), error()	Firebase 오류 체크, 메시지 출력	

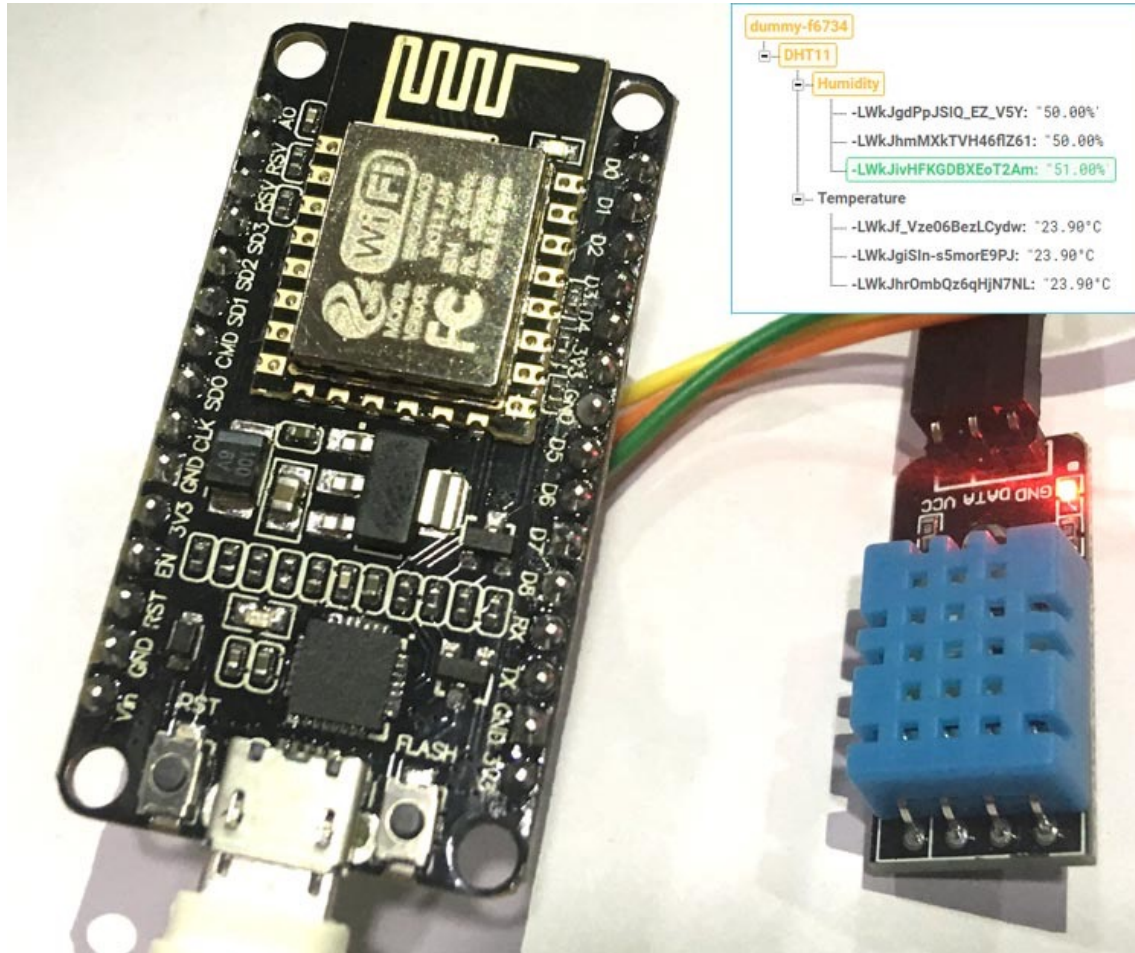
무엇을 개발하나?

- 클라우드 기반 IoT 실시간 원격 온습도 모니터



필요한 실습 부품

- NodeMCU(ESP8266-E), DHT11, 구글 Firebase + Realtime Database



Weather Monitor using Firebase & NodeMCU

Temperature

27.2 °C

Humidity

72 %

/dev/cu.wchusbserial1410

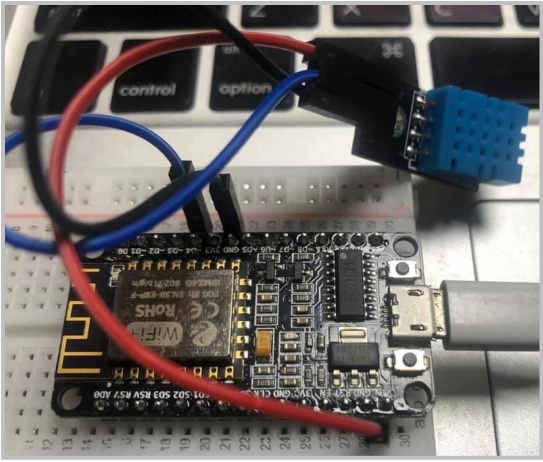
```
pushed: /logDHT/-M9kxtY3IJExghg0y72t
Temperature = 27.10
Humidity = 72.00
pushed: /logDHT/-M9kxv4CkygVSkNhtlml
Temperature = 27.10
Humidity = 72.00
pushing /logs failed:
Temperature = 27.10
Humidity = 72.00
```

DHT11_Firebase_DB_NodeMCU

DHT11_Firebase_DB_NodeMCU

```
1 #include <ESP8266WiFi.h>
2 #include <FirebaseArduino.h>
3
4 #define DHTPIN D4
5 #define DHTTYPE DHT11
6
7 #include <DHT.h>
8 DHT dht(DHTPIN, DHTTYPE);
9
10 // Set these to run example.
11 #define FIREBASE_HOST "dht11-2020-1.firebaseio.com"
12 #define FIREBASE_AUTH "f6cHQJ63t8KQvPiZPI"
13 #define WIFI_SSID "Amadeus"
14 #define WIFI_PASSWORD "deitcs3217"
15
16 void setup() {
17   Serial.begin(9600);
18   // connect to wifi.
19   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
20   dht.begin(9600);
21
22   Serial.print("connecting");
23   while (WiFi.status() != WL_CONNECTED)
24     Serial.print(".");
25     delay(500);
```

Leaving...



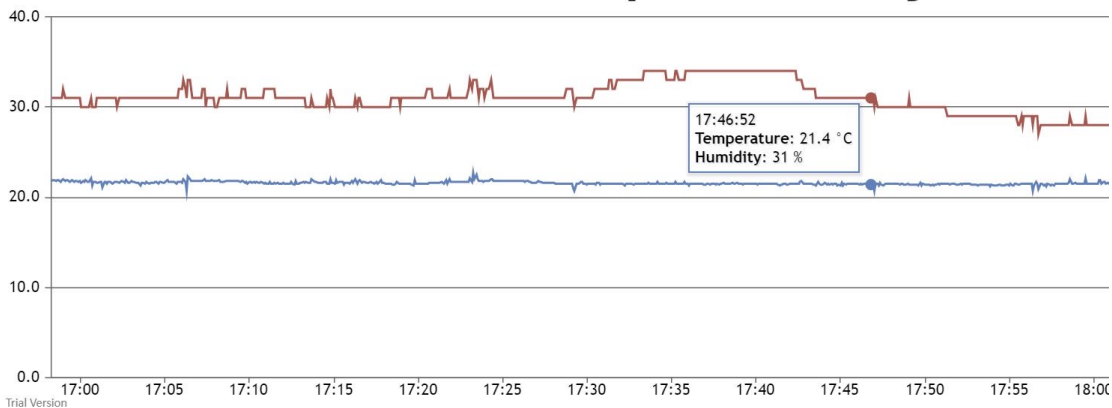
< NodeMCU/ DHT11>



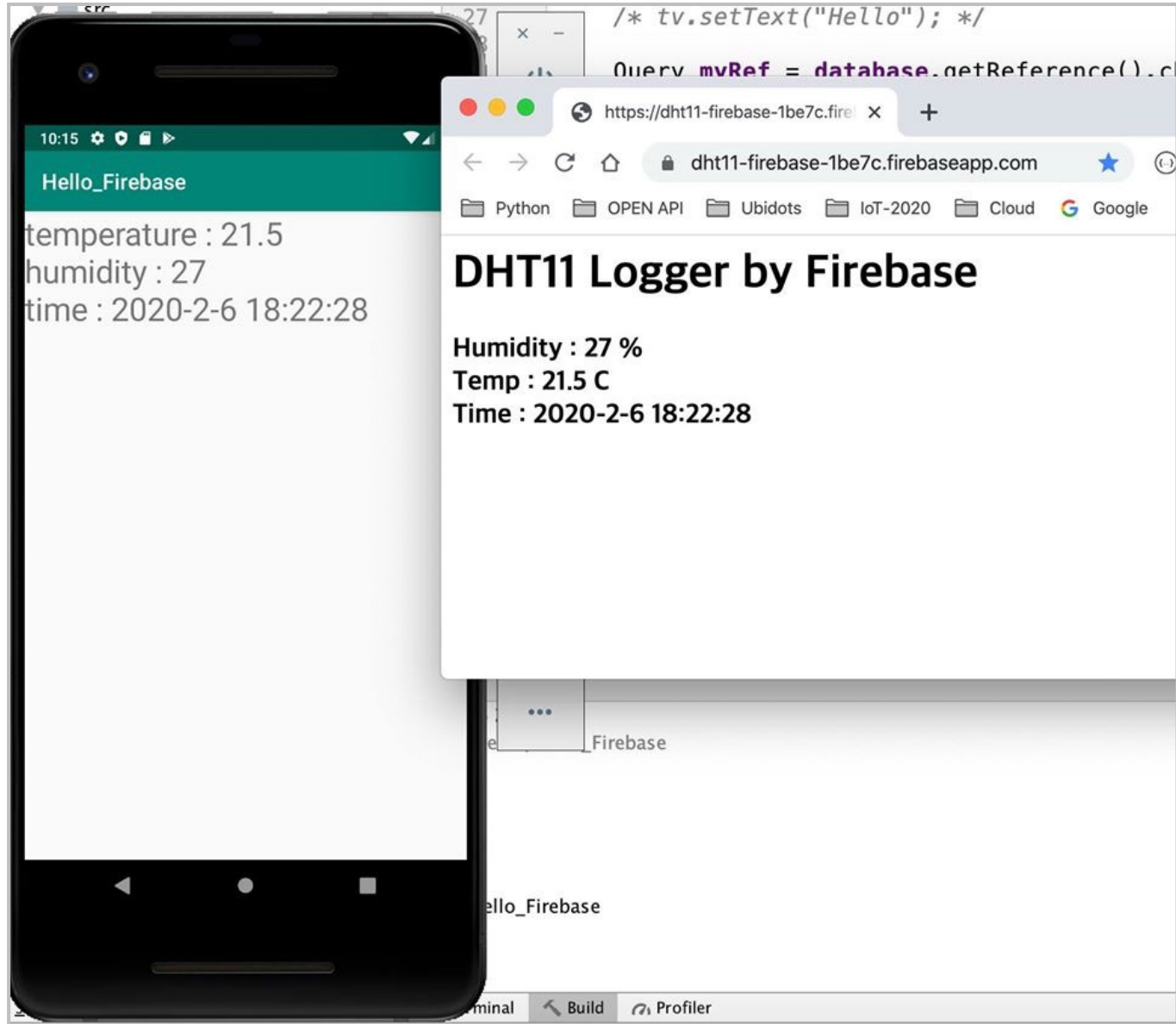
< Realtime Database>

DHT11 Logger by Firebase

Temperature & Humidity

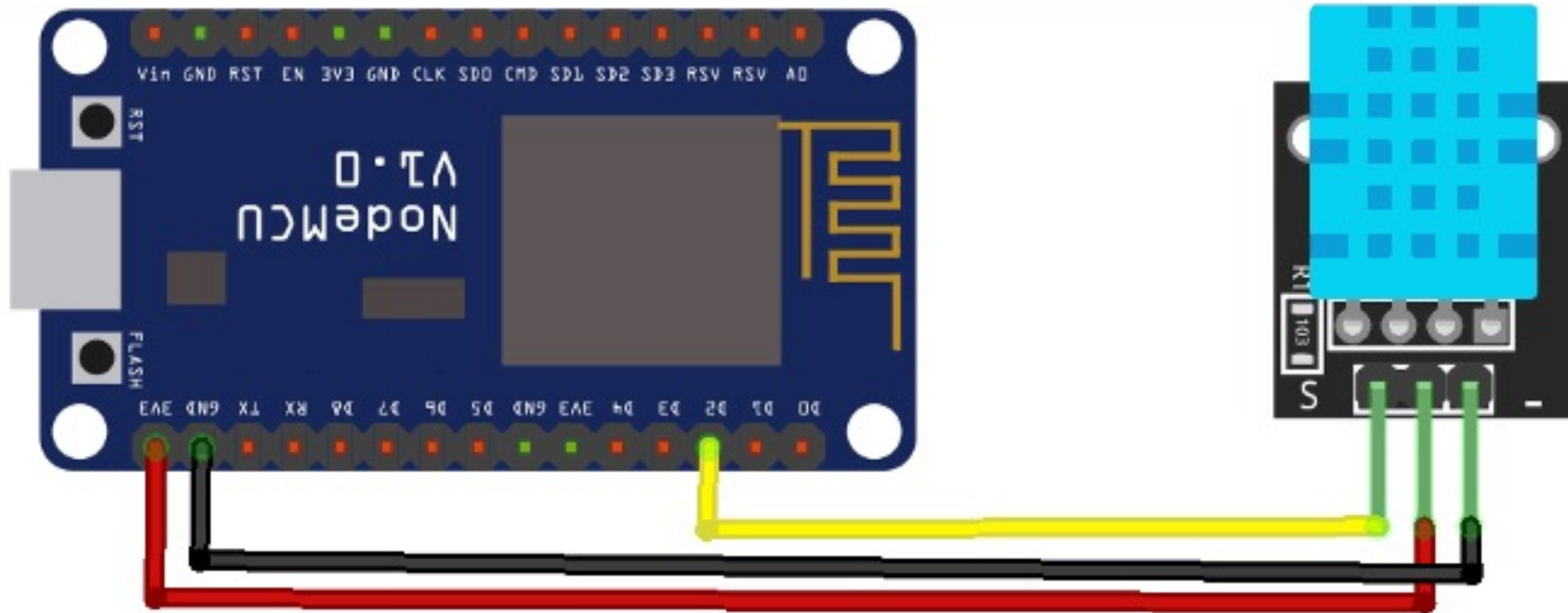


< 웹 대시보드 : CanvasJS>



< 안드로이드 앱/ 웹 앱>

하/하



Database Setup



Database Rule

The screenshot shows the Firebase console interface for a project named "Sensorandroid". The left sidebar contains navigation links for "Project Overview", "Develop" (Authentication, Database, Storage, Hosting, Functions, ML Kit), and "Quality" (Crashlytics, Performance, Test Lab, App Distribution). The main content area is titled "Database" and shows the "Realtime Database" selected. Below the title are tabs for "Data", "Rules", "Backups", and "Usage", with "Rules" being the active tab. There are two buttons: "Edit rules" and "Monitor rules". A "Rules Playground" button is also visible. The rules editor shows a JSON configuration for the database rules, with line numbers 1 through 7 on the left. The rules are set to allow both read and write operations.

Firebase

Sensorandroid

Go to docs

Database

Realtime Database

Data Rules Backups Usage

Edit rules Monitor rules

Rules Playground

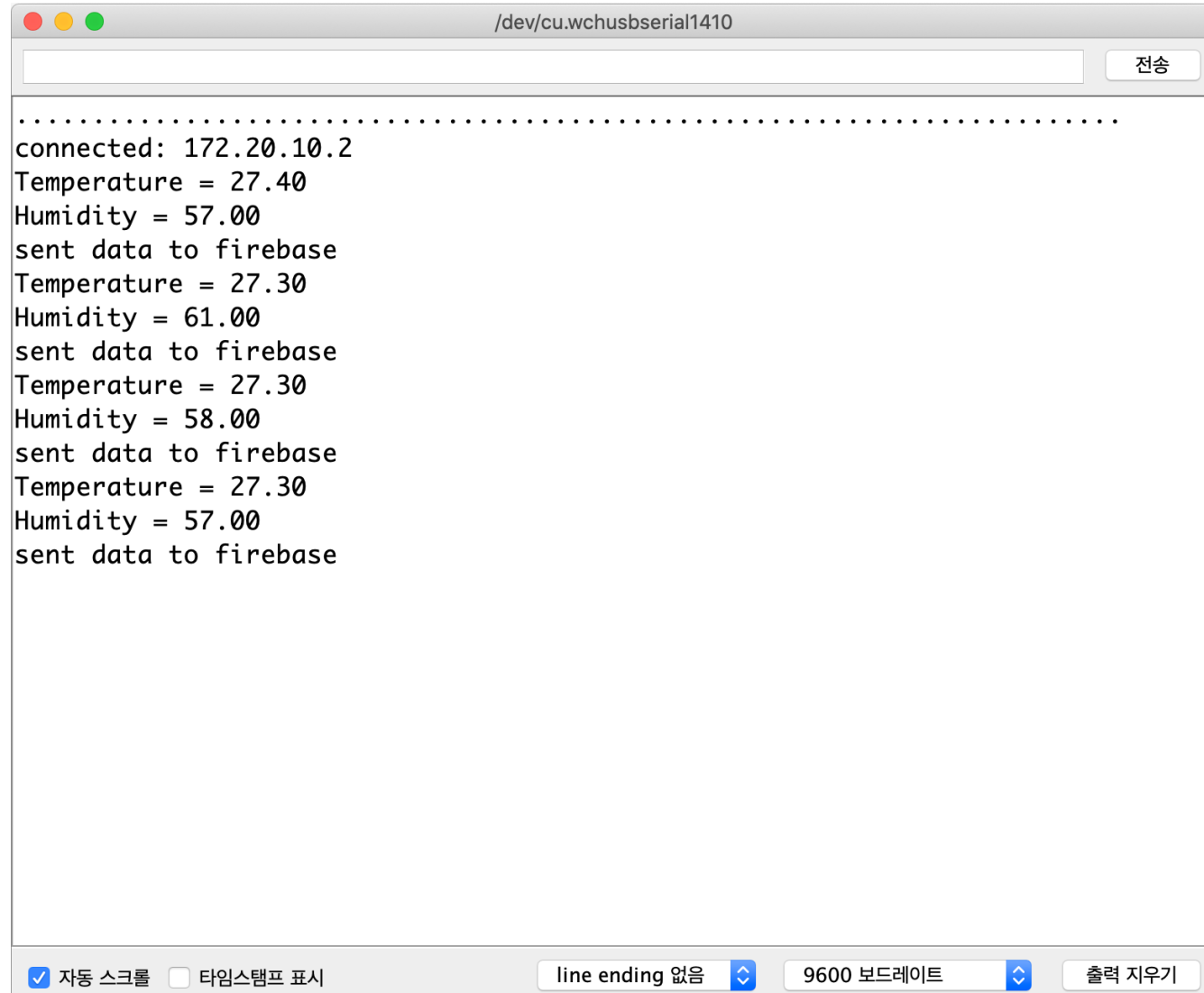
```
1 {
2   /* Visit https://firebase.google.com/docs/database/security to learn
3   "rules": {
4     ".read": true,
5     ".write": true
6   }
7 }
```

소스 코드

```
1 #include <ESP8266WiFi.h>
2 #include <FirebaseArduino.h>
3 #define DHTPIN D4
4 #define DHTTYPE DHT11
5 #include <DHT.h>
6
7 DHT dht(DHTPIN, DHTTYPE);
8
9 #define FIREBASE_HOST "dht11-2020-1.firebaseio.com"
10 #define FIREBASE_AUTH "f6cHQJ63t8KQ...N7...3GVB36H"
11 #define WIFI_SSID "Amadeus"
12 #define WIFI_PASSWORD "dht11-2020-1"
13
14 void setup() {
15     Serial.begin(9600);
16     // connect to wifi.
17     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
18     dht.begin(9600);
19
20     Serial.print("connecting");
21     while (WiFi.status() != WL_CONNECTED) {
22         Serial.print(".");
23         delay(500);
24     }
25     Serial.println();
26     Serial.print("connected: ");
27     Serial.println(WiFi.localIP());
28
29     Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
30 }
```

```
32 void loop() {
33     float temp = dht.readTemperature();
34     float humi = dht.readHumidity();
35     Serial.print("Temperature = ");
36     Serial.println(temp);
37     Serial.print("Humidity = ");
38     Serial.println(humi);
39
40     Firebase.setFloat("temperature", temp);
41     Firebase.setFloat("humidity", humi);
42
43     // handle error
44     if (Firebase.failed()) {
45         Serial.print("pushing /logs failed:");
46         Serial.println(Firebase.error());
47         return;
48     }
49     Serial.println("sent data to firebase");
50     delay(5000);
51 }
```

Serial Monitor



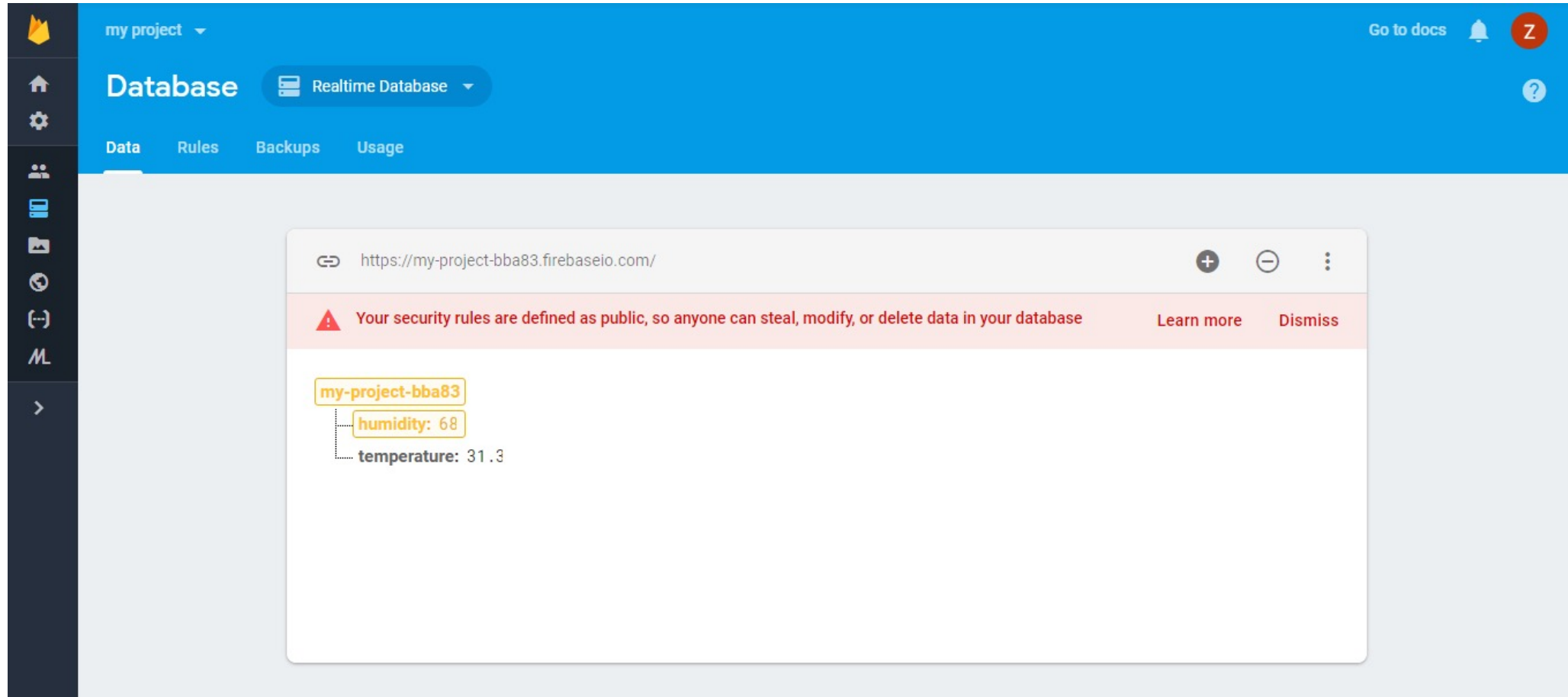
The screenshot shows a Serial Monitor window with a title bar indicating the device path `/dev/cu.wchusbserial1410`. The main text area displays a series of log messages: a connection status, followed by three cycles of temperature and humidity readings, each followed by a confirmation that the data was sent to Firebase. The bottom of the window features a control bar with several options: '자동 스크롤' (checked), '타임스탬프 표시' (unchecked), 'line ending 없음' (selected), '9600 보드레이트' (selected), and a '출력 지우기' button.

```
.....  
connected: 172.20.10.2  
Temperature = 27.40  
Humidity = 57.00  
sent data to firebase  
Temperature = 27.30  
Humidity = 61.00  
sent data to firebase  
Temperature = 27.30  
Humidity = 58.00  
sent data to firebase  
Temperature = 27.30  
Humidity = 57.00  
sent data to firebase
```

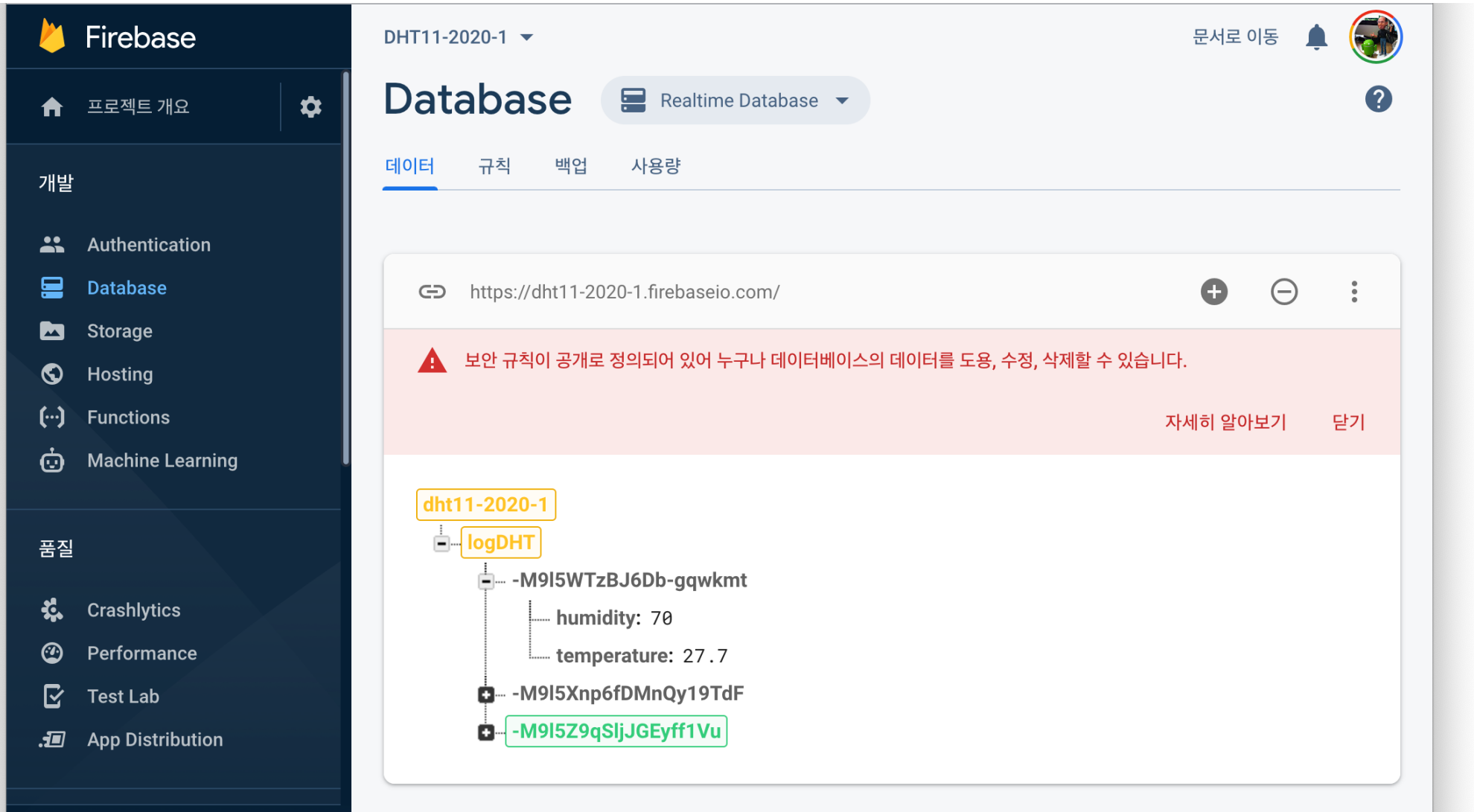
전송

☒ 자동 스크롤 ☐ 타임스탬프 표시 line ending 없음 9600 보드레이트 출력 지우기

Firestore Database



실습 과제 01 : DB에 list 값 append



The screenshot displays the Firebase Realtime Database interface for project 'DHT11-2020-1'. The left sidebar shows the project overview and development tools. The main panel shows the 'Database' section with the 'Realtime Database' selected. A warning message is displayed, indicating that security rules are public, allowing anyone to read, update, or delete data. Below the warning, the database structure is shown, with a node 'logDHT' containing a list of data points. The data points are represented as objects with 'humidity' and 'temperature' values. The last data point is highlighted with a green border.

Firebase

프로젝트 개요

개발

- Authentication
- Database
- Storage
- Hosting
- Functions
- Machine Learning

품질

- Crashlytics
- Performance
- Test Lab
- App Distribution

DHT11-2020-1

Database

Realtime Database

데이터 규칙 백업 사용량

https://dht11-2020-1.firebaseio.com/

⚠ 보안 규칙이 공개로 정의되어 있어 누구나 데이터베이스의 데이터를 도용, 수정, 삭제할 수 있습니다.

자세히 알아보기 닫기

dht11-2020-1

logDHT

- M9I5WTzBJ6Db-gqwkmtdetected humidity: 70 temperature: 27.7
- M9I5Xnp6fDMnQy19TdF
- M9I5Z9qSljJGEyff1Vu

dht11-2020-1

DHT11

Humidity

-M9Ggyfm3uuTdzQGwL6_: "34.00%"

-M9Gh--XzTWbMKL38Bla: "34.00%"

-M9Gh0KPjm9WW9oX2nZw: "34.00%"

-M9Gh1fkzS88uBRNVObV: "34.00%"

-M9Gh3-hlHz6btZBtpPJ: "34.00%"

-M9Gh4LztUbFqQC2UVtL: "34.00%"

-M9Gh5fo9o3jUMwhFpJG: "34.00%"

-M9Gh7-cGassZYe8vic6: "34.00%"

-M9Gh8LztZywGZpehZtw: "34.00%"

-M9Gh9fpL8nWTjxX07Fg: "34.00%"

-M9GhB-aLcNG6gqyTeXk: "34.00%"

-M9GhCKSi_AYMWUIWgWC: "34.00%"

-M9GhDdoaY6ZjW1G1Yp7: "34.00%"

Temperature

-M9GgyinwtaGmxWvj5xZ: "26.90°C"

-M9Gh-2bCKtLcukBIEZK: "26.90°C"

-M9Gh0P0emmO_kiPmaNZ: "26.90°C"

-M9Gh1inVBCf3q5MLXSk: "26.90°C"

-M9Gh34D1LRd0cxen3ML: "26.90°C"

-M9Gh4P0Yf2_bl-cf_OM: "26.90°C"

-M9Gh5ip6qjAkUdvPGaZ: "26.90°C"

실습 과제 02 : DB에 Object 값 append

The screenshot displays the Firebase Realtime Database interface. On the left is a dark sidebar with the 'Firebase' logo and a navigation menu. The main area shows the 'Database' section for project 'DHT11-2020-1', with a 'Realtime Database' dropdown. A security warning banner is present, stating that public security rules allow anyone to read, update, or delete data. Below the warning, a tree view shows the database structure: a root node 'dht11-2020-1' containing a 'logDHT' node, which in turn contains three child nodes. The first child node is expanded, showing 'humidity: 70' and 'temperature: 27.7'. The third child node is highlighted with a green border.

Firebase

프로젝트 개요

개발

- Authentication
- Database**
- Storage
- Hosting
- Functions
- Machine Learning

품질

- Crashlytics
- Performance
- Test Lab
- App Distribution

DHT11-2020-1

문서로 이동

Database

Realtime Database

데이터 규칙 백업 사용량

데이터

<https://dht11-2020-1.firebaseio.com/>

⚠ 보안 규칙이 공개로 정의되어 있어 누구나 데이터베이스의 데이터를 도용, 수정, 삭제할 수 있습니다.

[자세히 알아보기](#) [닫기](#)

```
graph TD
    Root[dht11-2020-1] --> logDHT[logDHT]
    logDHT --> Node1["-M9I5WTzBJ6Db-gqwkm"]
    logDHT --> Node2["-M9I5Xnp6fDMnQy19TdF"]
    logDHT --> Node3["-M9I5Z9qSljJGEyff1Vu"]
    Node1 --> humidity["humidity: 70"]
    Node1 --> temperature["temperature: 27.7"]
```

dht11-2020-1



logDHT



-M9I5WTzBJ6Db-gqwkm

humidity: 70

temperature: 27.7



-M9I5Xnp6fDMnQy19TdF



-M9I5Z9qSljJGEyff1Vu



-M9I5_ma8_F7LATwouE0



-M9I5b6NDniltUBeMpny



-M9I5ckvnWVexs7p6R3V



-M9I5e4z7czzTGO_rarD

humidity: 70

temperature: 27.7



-M9I5fq1NptghuEnx3rl



-M9I5hC1vctV_GwTajRO



-M9I5imOWXrNni9dSnI2



-M9I5kAbnm2HASp5S23c + ×

humidity: 70

temperature: 27.7



-M9I5lUCbc9dt9DpFtix



-M9I5moWg_1TIFAAIzxE

학습 참고

- 참고 소스
 - <https://github.com/loT-Lab-02/Week10-Lab>
- Sending Sensor Data to Google Firebase
 - <http://bitly.kr/Q00AhwggwU6>