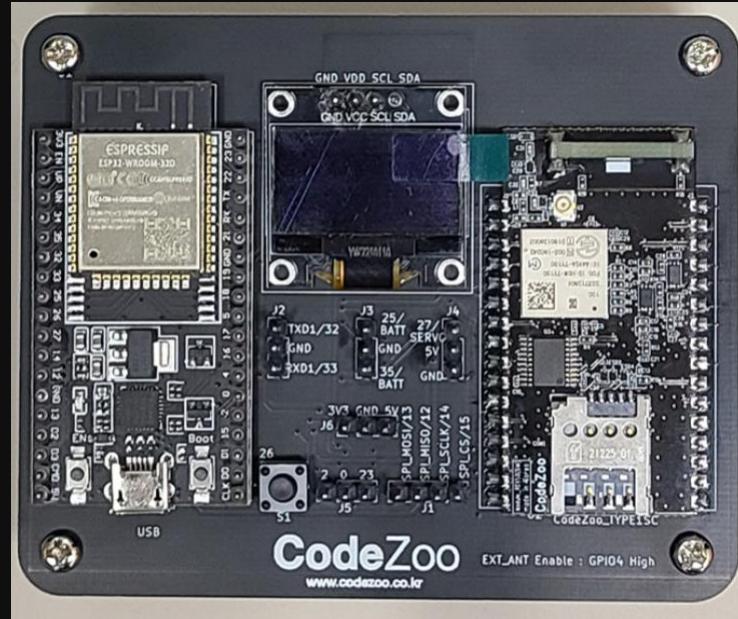


LTE-CAT.M1 with Global Sim ESP32 IoT Board hands-on



muRata
INNOVATOR IN ELECTRONICS

ASI A-SUNG
International CO.,LTD.

CodeZoo

Highlights

1. Introduction to LTE Cat.M1
2. Pay as you go data
3. Introduction to the Vodafone IoT Global SIM
4. Cat.M1 ESP32 IoT Board Hardware Configuration and Design
5. ESP32 IoT Lab (Monitoring)
 - Board Setup
 - Install libraries for LTE Cat.M1 development
 - 1) APN Setup : Setting up access network connection
 - 2) Basic Test : Get Cat.M1 current information
 - 3) Button Interrupt : Using ESP32 IoT board button
 - 4) TCP, UDP Test: Sending and receiving Echo Server data
 - 5) HTTP Test : Uploading and visualizing sensor data to Thingspeak.com
 - 6) MQTT Test : Publishing HiveMQTT sensor data
 - 7) AWS-IoT Setup : Create IoT node, set rules, and get access address
 - 8) Write KEY (Root CA, Client CA, Client Private KEY): Configure AWS-IoT security settings
 - 9) AWSIOT Test : Upload sensor data collected by Cat.M1 to AWS cloud
 - 10) AWSIOT SiteWise_demo: Manage sensor data collected by CAT.M1 with AWS IoT SiteWise Portal
6. ESP32 IoT Lab (Control)
 - 1) MQTT LED remote control: Control device LED with MQTTBox from remote location
 - 2) TCP socket remote control: Programming a TCP socket to wait for server response

1. Introduction to LTE Cat.M1



Bluetooth LE



NFC



RFID



Wi-Fi



Thread



Ethernet

NB-IoT



6LoWPAN Sub-GHz Mesh



LoRa LPWAN



LTE Cat.M1



Among the various IoT connectivity,

- LTE Cat.M1, NB-IoT, and 5G use carrier base stations to connect to use carrier base stations to provide nationwide network service using carrier base stations
- Communication infrastructure for data communication Deployment-free and independent communication Network deployable, TCP/UDP communication
- Cat.M1, NB-IoT, 5G each Communication speeds and plans vary

국내 주요 IoT 기술 비교

	LTE Cat.M1	LTE-M	LoRa	NB-IoT
주파수	LTE 대역 내 0~1.08MHz 가변	LTE 대역 내 0~20MHz 가변	비면허 대역 125kHz*8채널	LTE 대역 내 180kHz 고정
전송 속도	~300Kbps	~10Mbps	~5.4kbps	~27kbps
가능 서비스	데이터, 음성, 사진	데이터, 음성, 영상	센서 측정치 등 소량 데이터	센서 측정치 등 소량 데이터
배터리 수명	수 년 이상	수 개월	수 년 이상	수 년 이상

※전송 속도는 다운로드 기준 최대 속도

자료=SK텔레콤

Pricing Plans

- LoRa Basic Rate : KRW 385, 100KB (SKT)
 - NB-IoT: KRW 330, based on 100KB (LG U+)
 - Cat.M1 basic rate: KRW 1,100, based on 5MB (SKT)
- (*) 5G is still a pilot service for smartphones and high-speed services

Data Transfer Interval (Minimum)

- LoRa, NB-IoT: once every 1 hour
- Cat.M1 once every 1 minute

2. Cat.M1 Data Pay-as-you-go

TCP Packet Transmit Header	230byte
User Packet	xxbyte

TCP Transmit : 230 + xx (byte)

TCP Packet Receive Header	199byte
User Packet	xxbyte

TCP Receive : 199 + xx (byte)

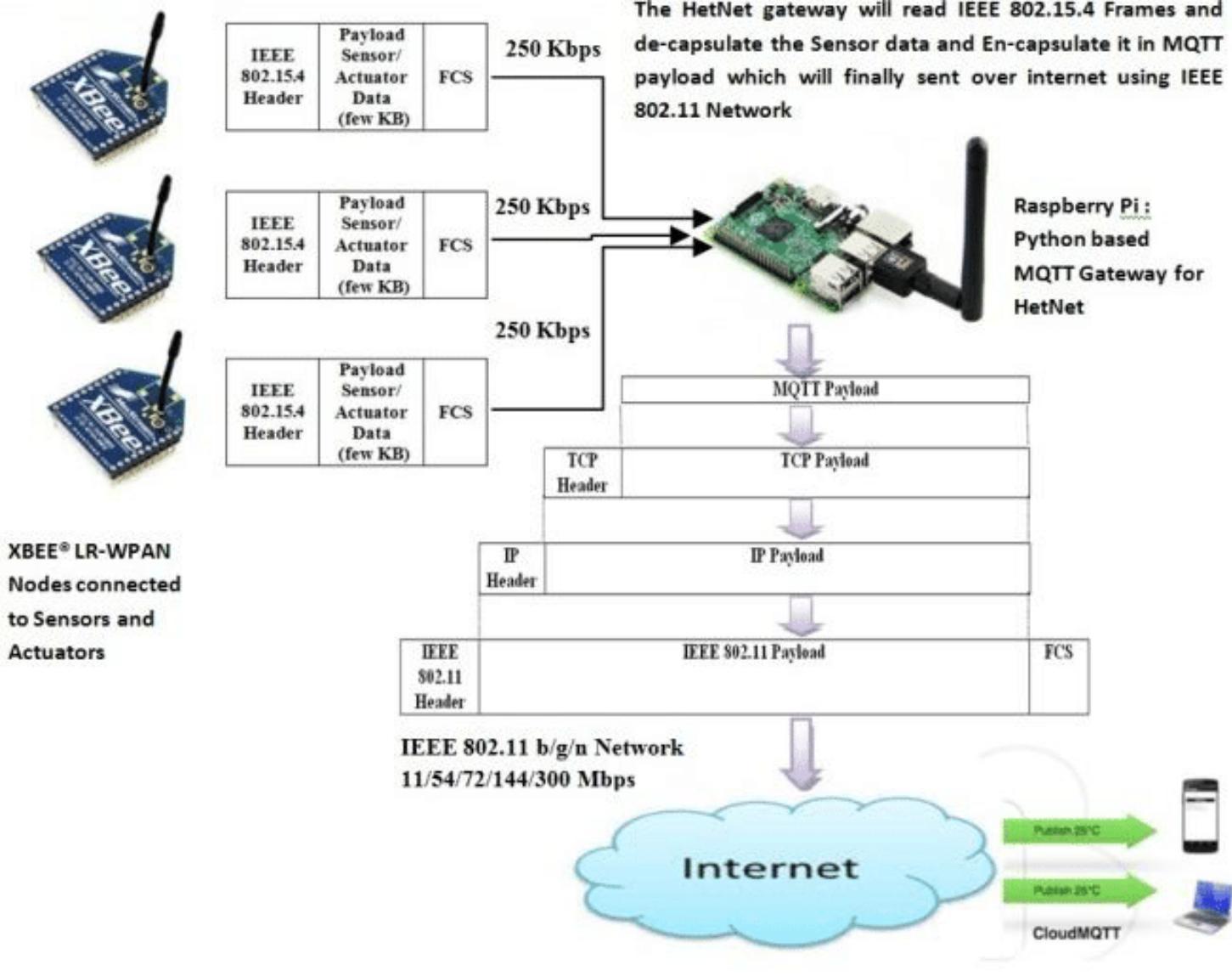
UDP Packet Transmit Header	40byte
User Packet	xxbyte

UDP Transmit : 40 + xx (byte)

UDP Packet Receive Header	40byte
User Packet	xxbyte

UDP Receive : 40 + xx (byte)

AT Command When executing TCP and UDP commands, you can pre-calculate the data usage, including packet headers, for your IoT service. including the number of transmissions and receptions in advance, including packet data usage and the number of transmissions and receptions in advance, including packet headers, and select the appropriate plan for your IoT service (Important!!!)



**MQTT Data Send Size ==
MQTT Payload + MQTT Packet Header +
TCP Packet Header**

Data pay-as-you-go (!)
CATM1 charges based on the amount of data you use based on the amount of data you use, so you need to calculate the monthly data usage and the number of transmissions and receptions in advance and estimate the operating cost of communication service (very important!!!)

source :

https://www.researchgate.net/figure/further-shows-the-MQTT-application-layer-message-is-encapsulated-in-TCP-Segment-TCP_fig3_321341034

3. Vodafone IoT Global SIM Introduction

Global IOT SIM?

1. Easy to apply to overseas projects as it can be used roaming in more than 180 countries with a single glass SIM (no need to negotiate with each country's carrier)
2. Time to marketSupports services from carriers in multiple countries with only Global SIM-based Global Standard Certification (GCF) for complex certification procedures



In over 180 countries worldwide
IoT data roaming support
(including South Korea)

We are the market leader in IoT managed connectivity

Figure 1. Magic Quadrant for Managed IoT Connectivity Services, Worldwide



We're positioned highest for our “ability to execute” and furthest to the right for “completeness of vision”

Gartner defines a “Leader” in the Magic Quadrant as a company that executes well against their current vision and is well positioned for tomorrow.

We've maintained our leadership position in the Magic Quadrant for the last six years.



— Your devices can connect anywhere in the world

Network relationship:

- Vodafone Network
- Roaming Partners
- IoT Partner

Expertise

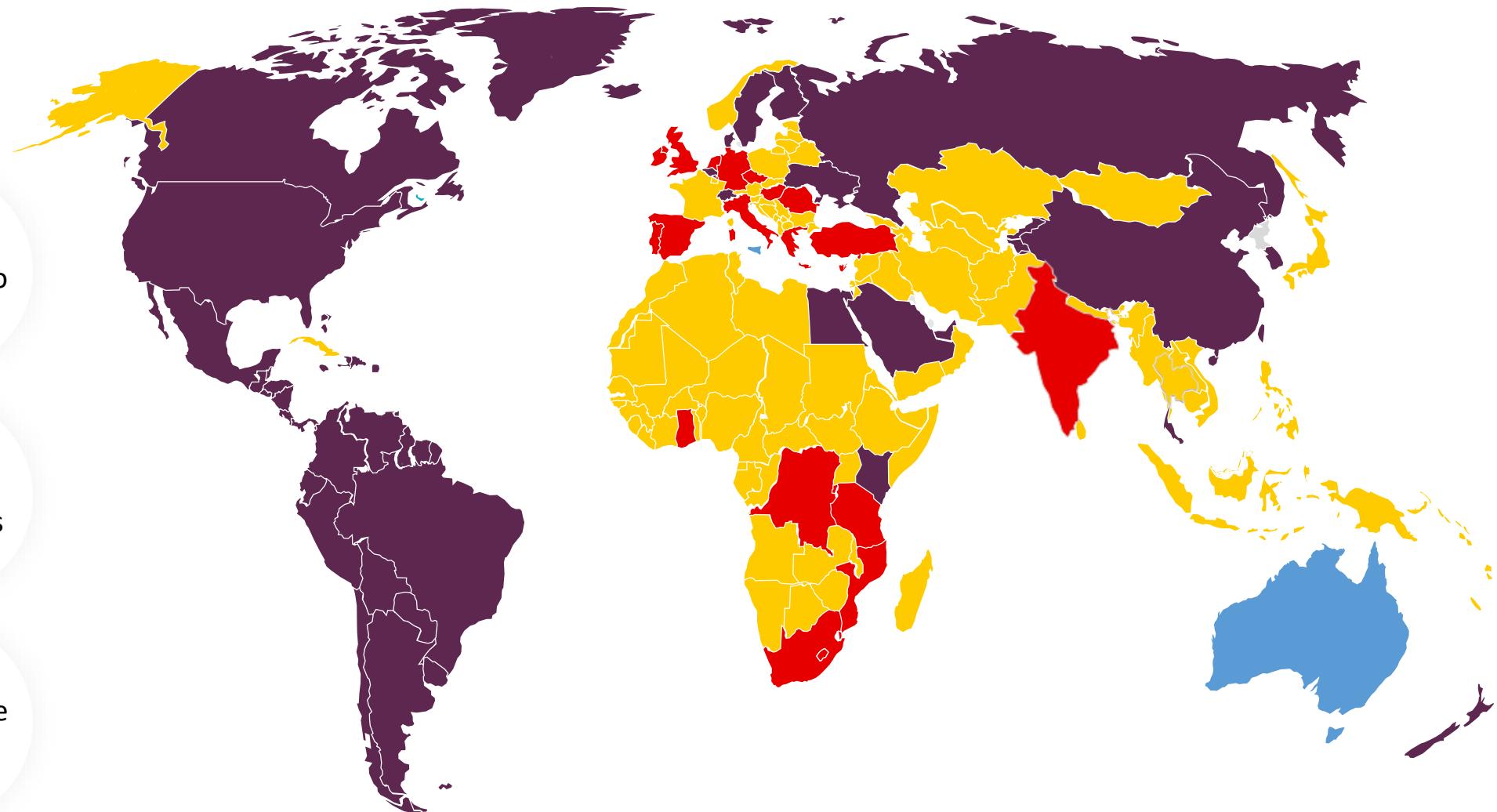
1,400+ IoT experts operating across 5 continents

Scale

182 countries
+570 networks 2/3/4/5G, LPWA
satellite and fixed capability

Capabilities

Specialist Automotive and private network divisions providing an end-to-end solution.



CodeZoo VodaFone Global IoT Sim Service

Global Sim Service



IoT specialist CodeZoo sells Vodafone Global Sim (with plan) and provides technical support service for IoT cellular device integration

Purchase : <https://url.kr/m2yie6> (Naver Smart Store)
Inquiries : rooney.jang@codezoo.co.kr

Vodafone Global Sim

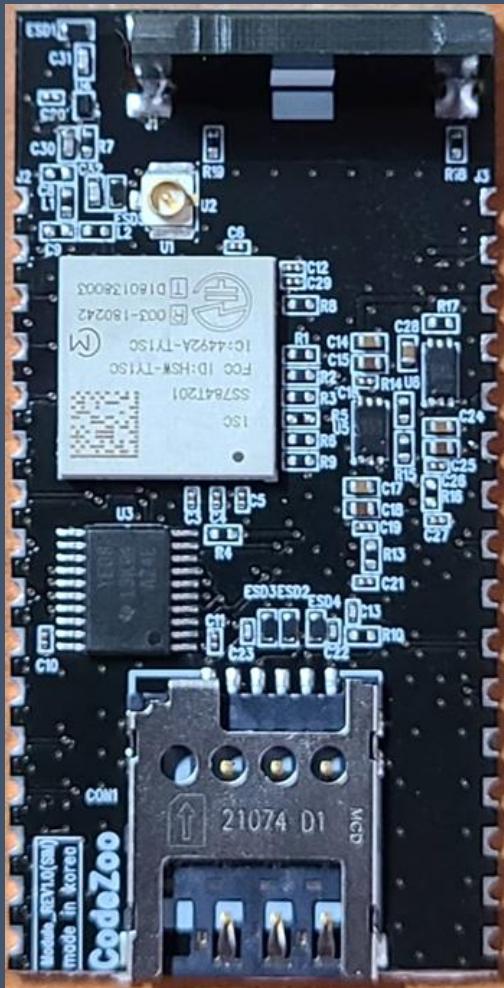
Purchase of SimCard, Data Plan

Vodafone Global Sim

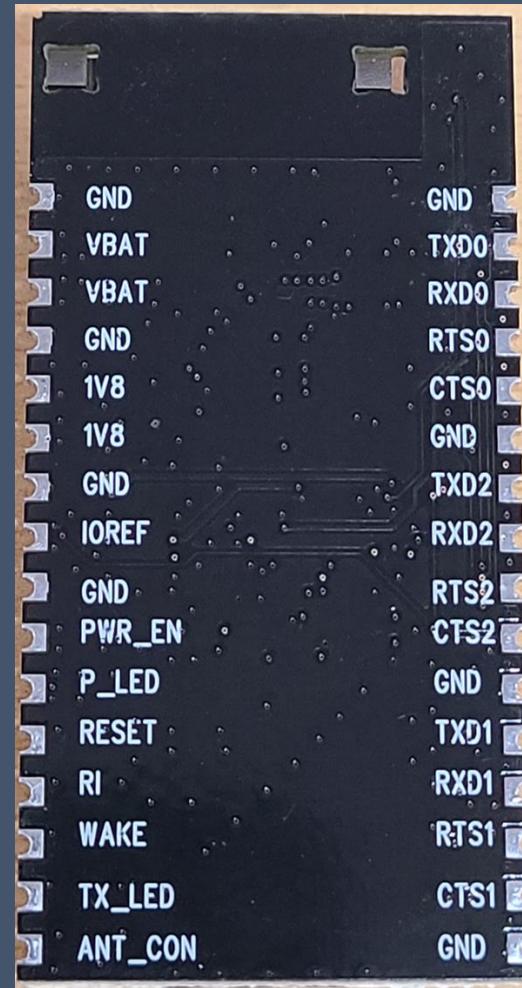
IoT Device Integration
Technical Support Services

4. Cat.M1 ESP32 IoT Board Hardware Configuration and Design

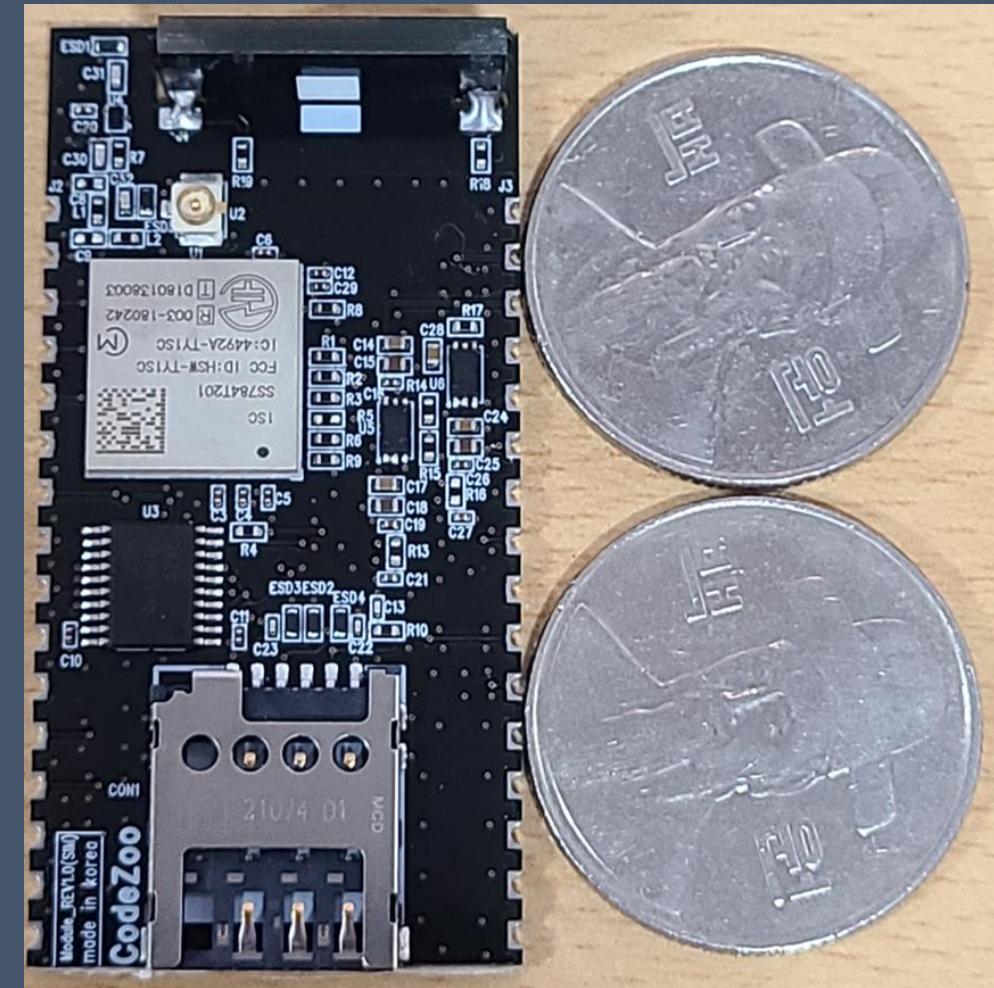
Cat.M1 embedded modem



TOP



BOTTOM

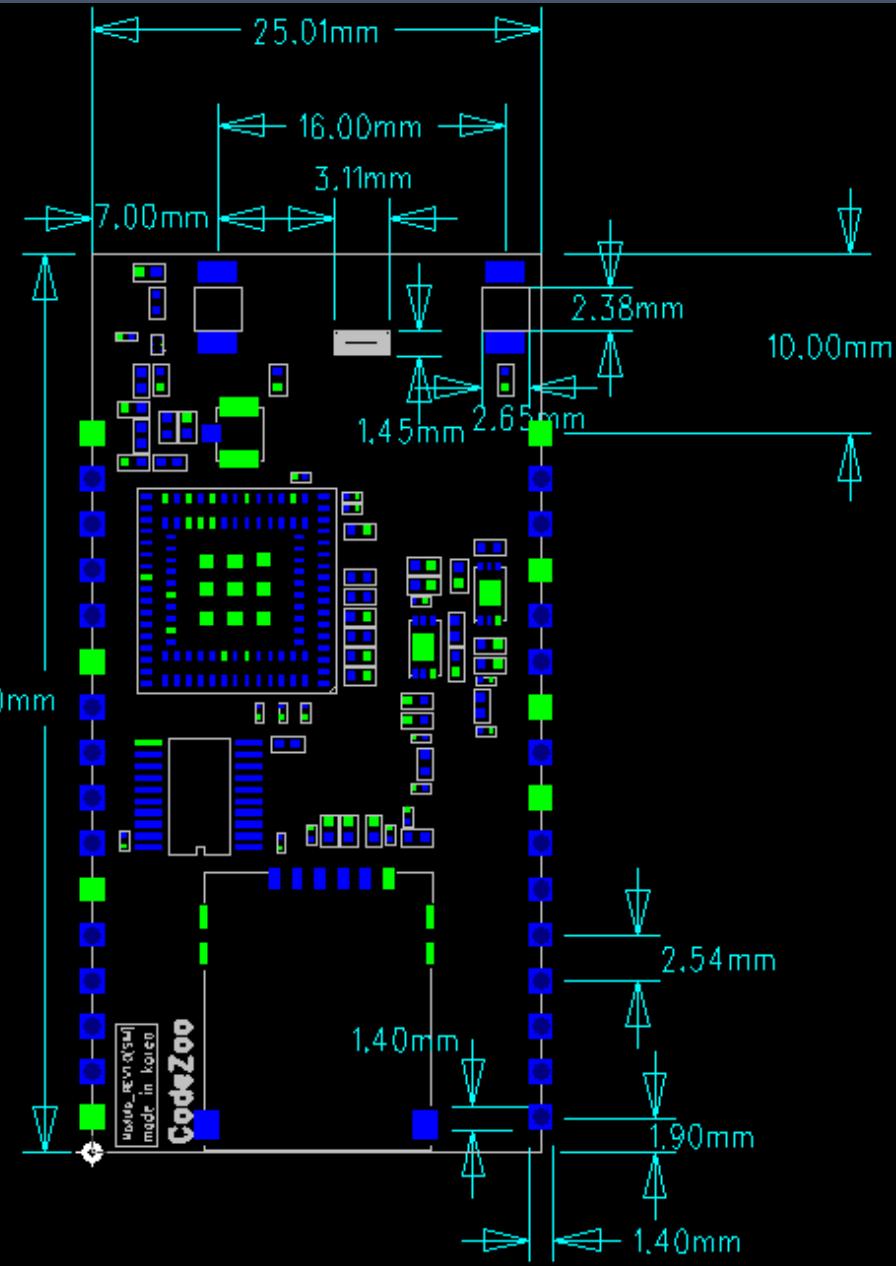


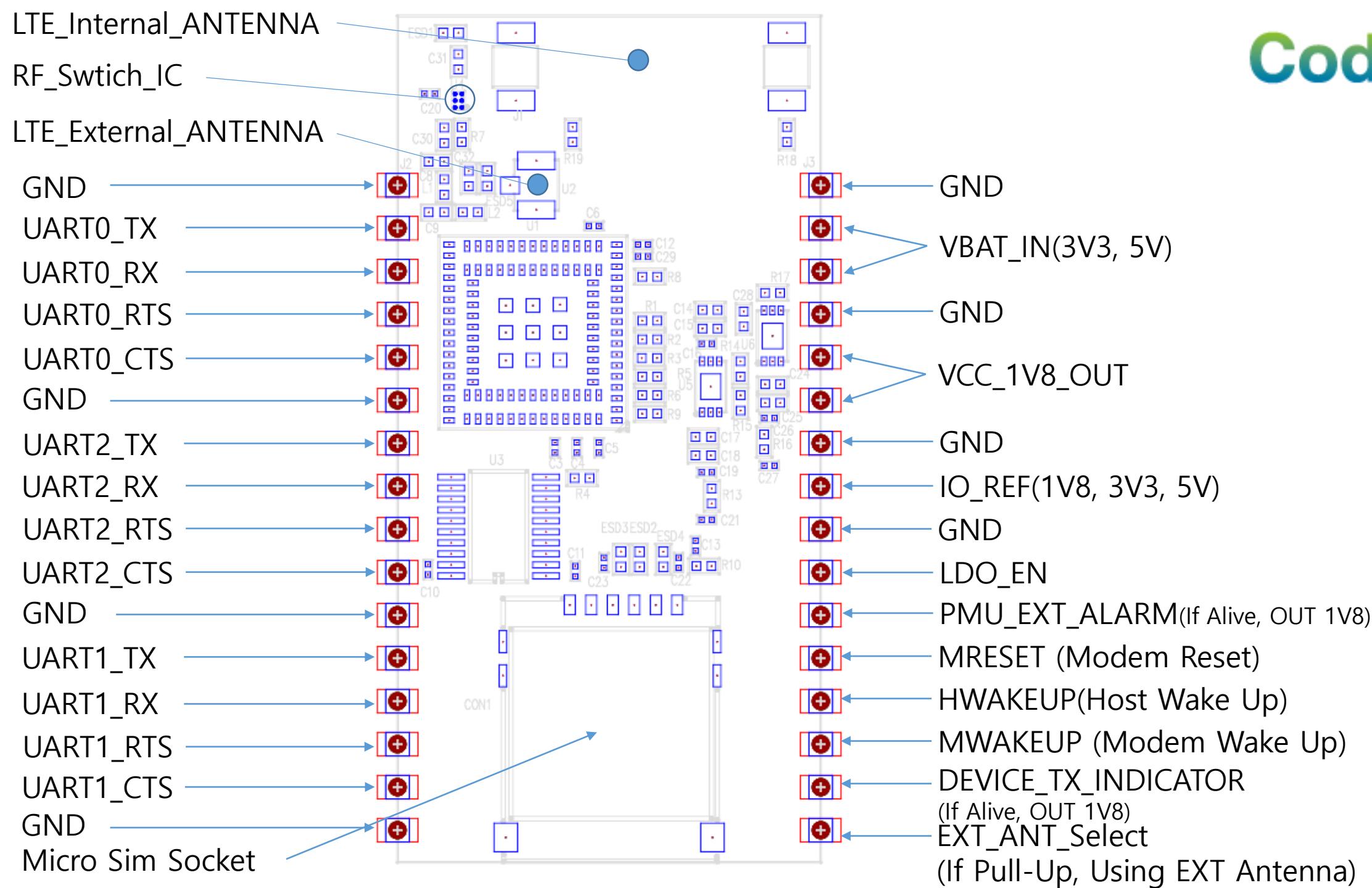
Cat.M1 embedded modem

구분 (classification)	규격 (Standard)
1. 제품명 (Product Name)	Type1SC LTE-CAT.M1 Board
2. 제품 모델명 (Product Model)	CZ-Type1SC
3. 제품 제조사 (Product Manufacturer)	CodeZoo
4. 통신모듈 모델명/제조사 (Module Model/Vendor)	Type1SC / Murata
5. 통신칩셋 모델명/제조사 (Chipset Model/Vendor)	ALT1250 / Sony Altair
6. 외형크기 (Dimension) [unit : mm]	Width(25.2)*Height(50.0)*Depth(1.4)
7. 기능용도 (Function-Use)	LTE communication module
8. 전원 타입 (Power Supply Type)	2.4V ~ 5.5V
9. 안테나 타입 (Antenna Type)	Internal and external (RF Switch IC Select)
10. 지원 통신규격 주파수 (Frequency Band)	Low-band B5/B8/B12/B13/B14/B17/B18/B19/B20/B26/B28 Mid-band B1/B2/B3/B4/B25

TYPE1SC_32_PAD

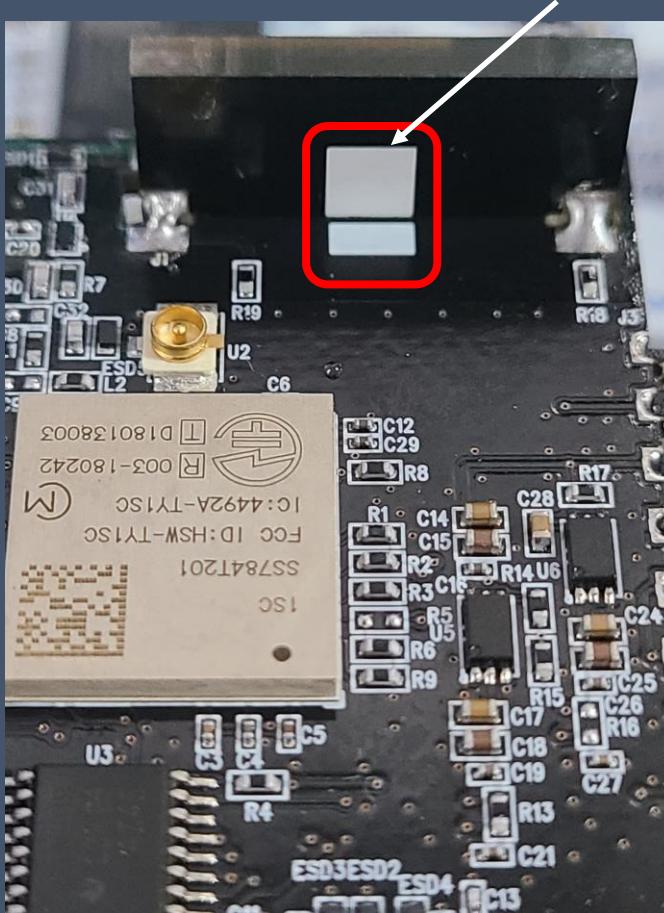
1	GND	17	GND
2	UART0_TX	18	VBAT
3	UART0_RX	19	VBAT
4	UART0_RTS	20	GND
5	UART0_CTS	21	VCC_1V8
6	GND	22	VCC_1V8
7	UART2_TX	23	GND
8	UART2_RX	24	IO_REF
9	UART2_RTS	25	GND
10	UART2_CTS	26	LDO_EN
11	GND	27	PMU_EXT_ALARM
12	UART1_TX	28	MRESET
13	UART1_RX	29	HWAKEUP
14	UART1_RTS	30	MWAKEUP
15	UART1_CTS	31	DEVICE_TX_INDICATOR
16	GND	32	EXT_ANT



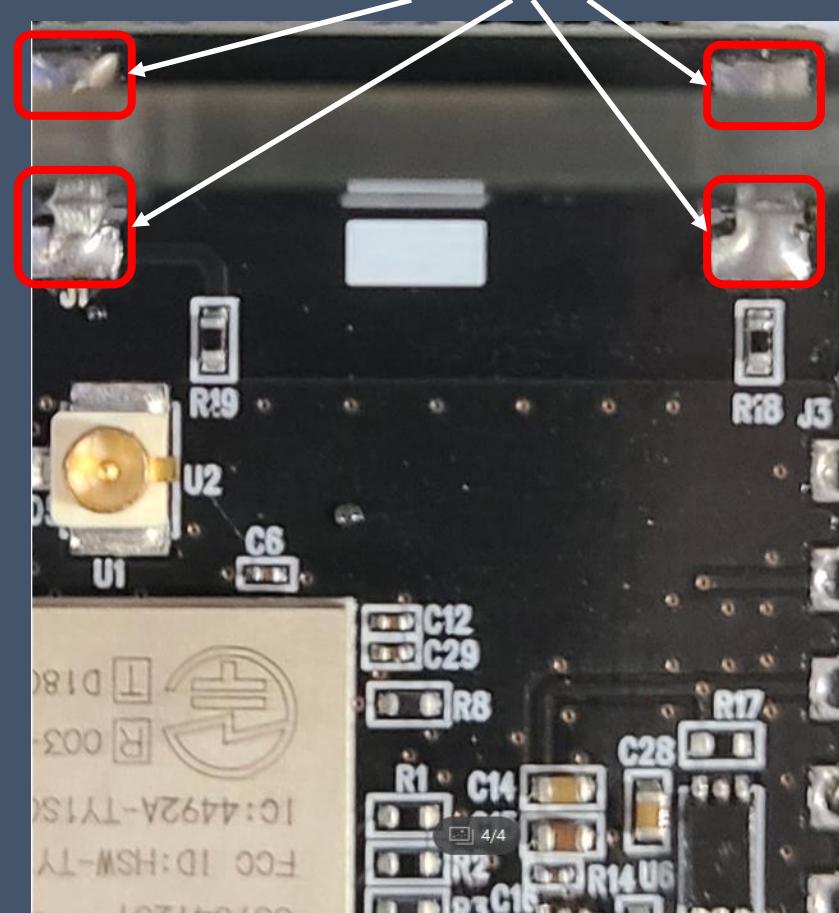


To connect the built-in antenna (included)

1. Check antenna orientation

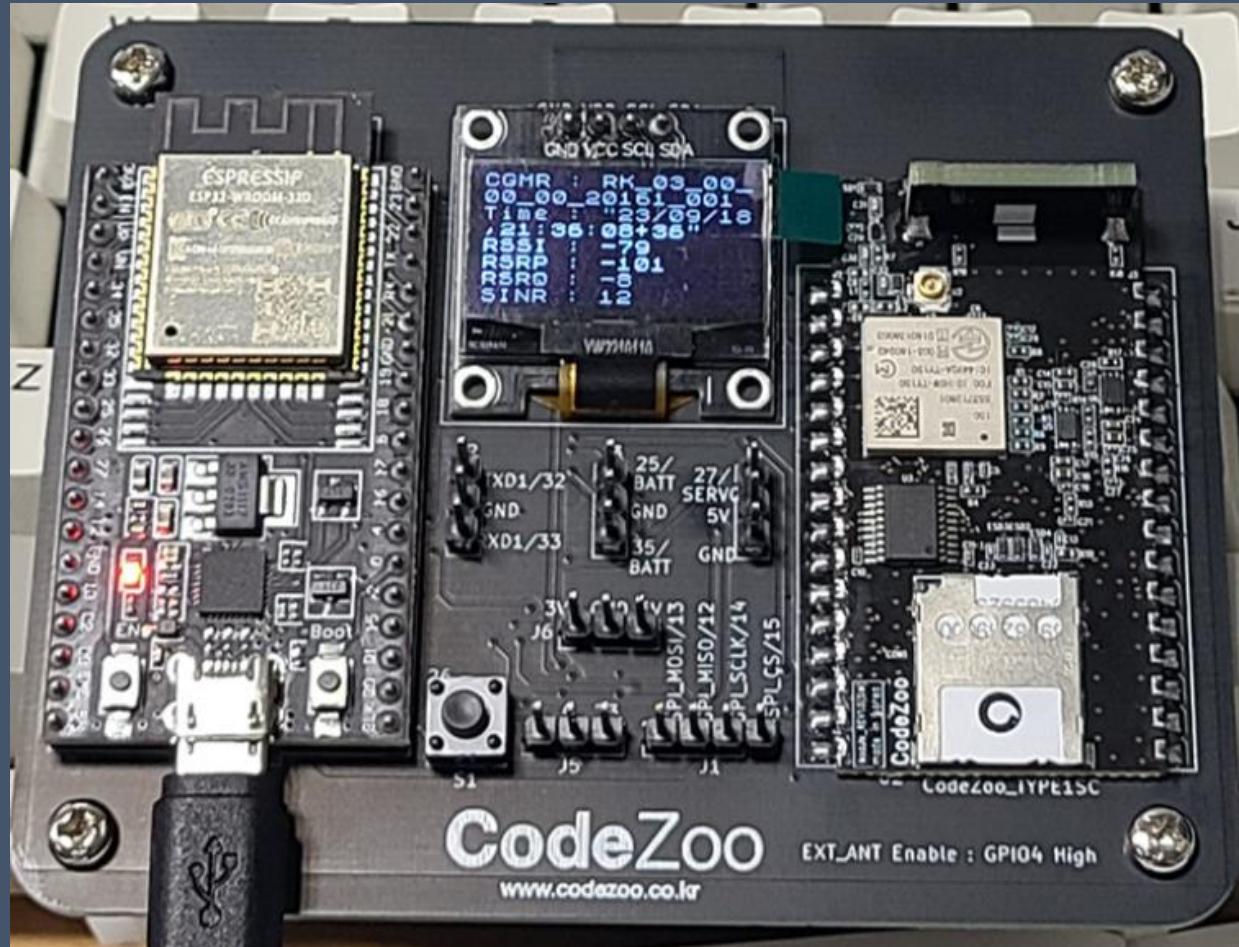
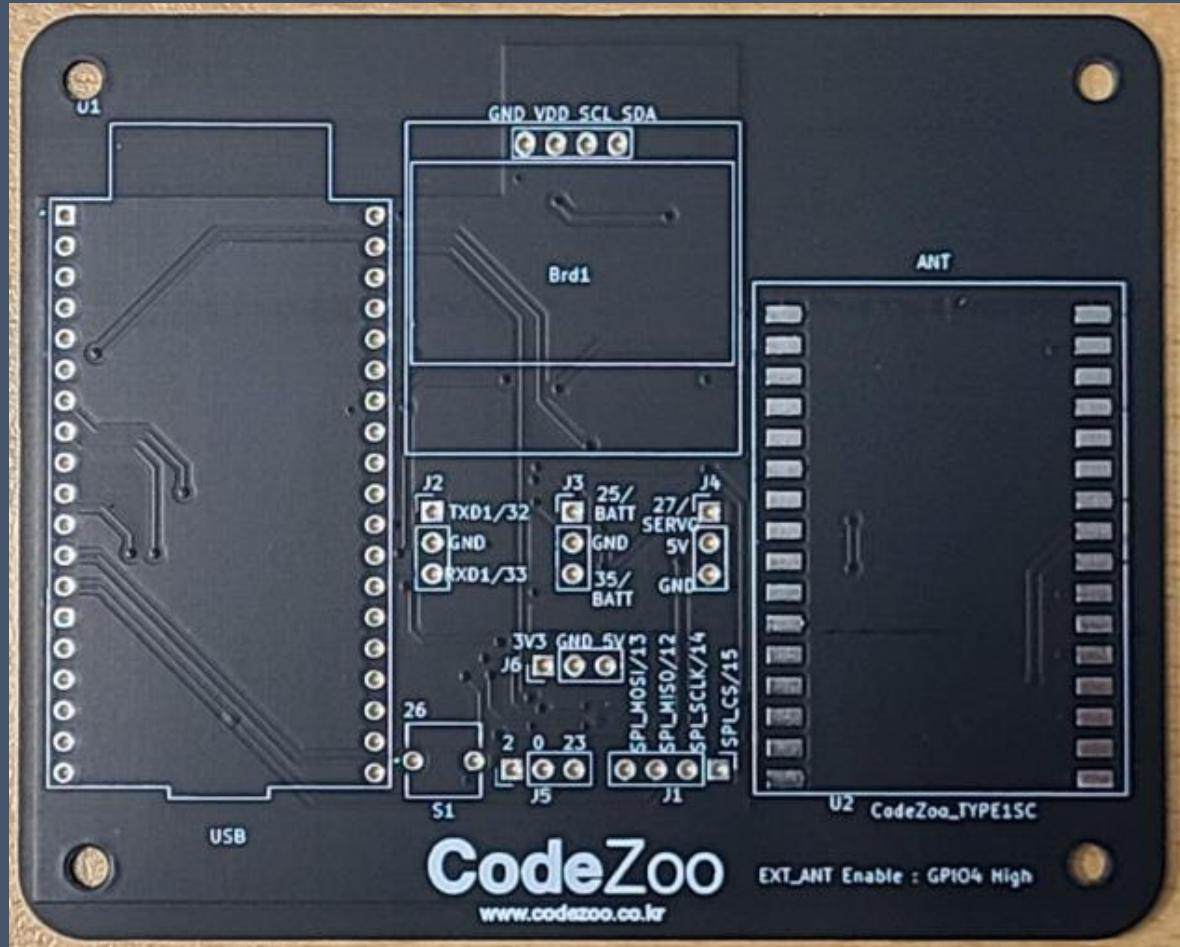


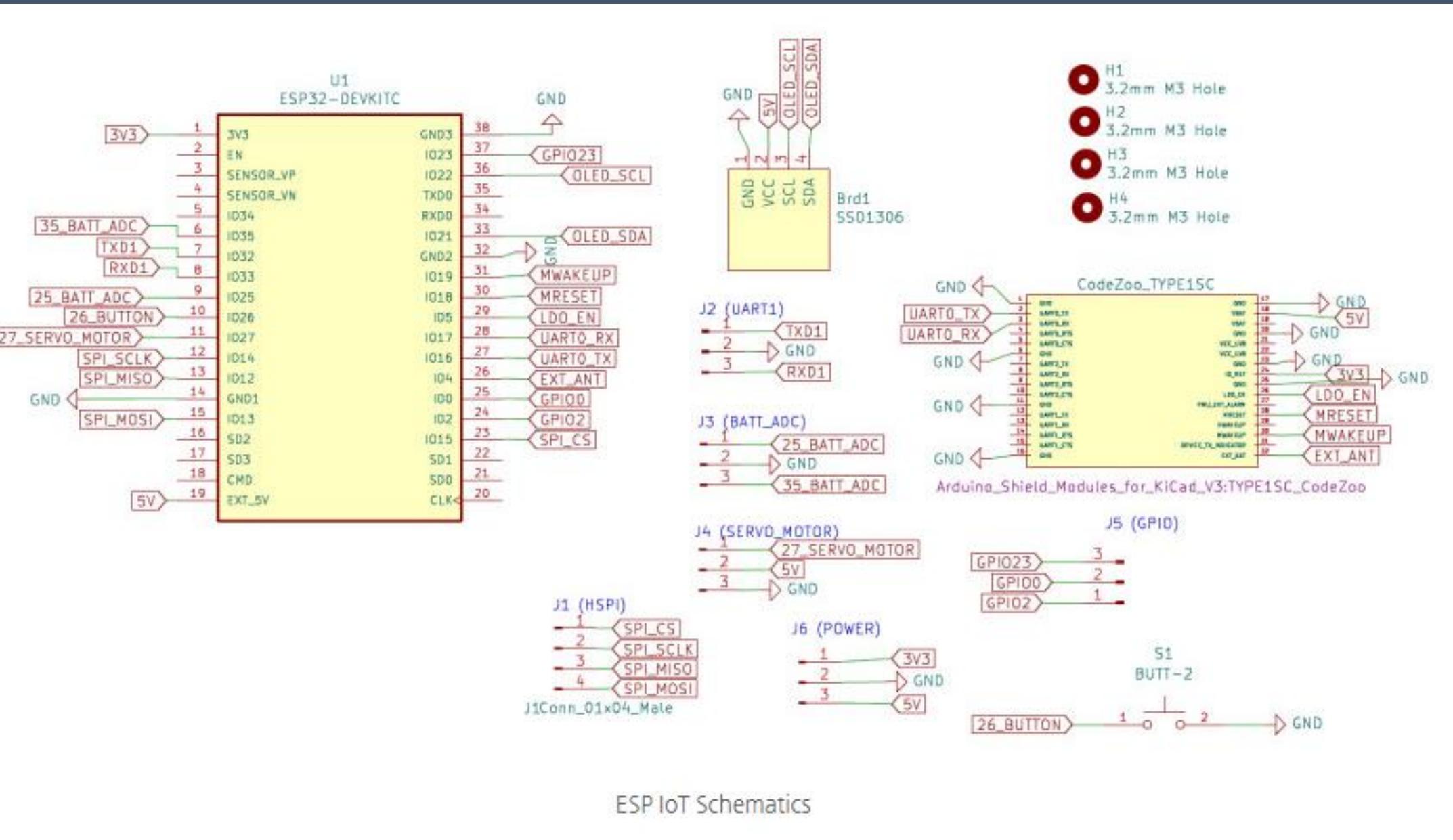
2. Antenna soldering (4 points)



Since the built-in antenna can wobble when soldering, secure it in the center with cellophane tape and work

Cat.M1 ESP32 IoT Board





ESP IoT Schematics

Cat.M1 ESP32 IoT Board Schematics

ESP32 DevKitC_V4 Board ESP32-WROOM-32D Module (OK)



[Espresif ESP32](#)

Release date: 2016 Sep 1

Product details

RAM	520 KB
CPU	Tensilica Xtensa LX6 32 bit Dual-Core @ 160 / 240Mhz
Flash	up to 64 MBytes
Input Voltage	2.2 V - 3.6 V
Operating Current	80 mA average
Temperature Range	-40°C to 125°C
Standards	FCC/CE/TELEC/KCC

Connectivity

Wi-Fi	✓ 802.11b/g/n
Bluetooth®	✓ 4.2 BR/EDR + BLE
UART	3x UART

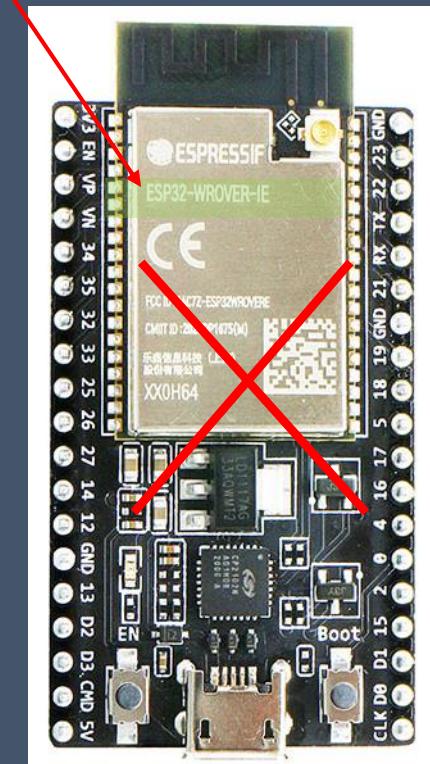
I/O

I2C	✓ up to 2
GPIO	32 (up to)
PCM	✓ up to 1
PWM	✓ up to 8
ADC	✓ 12 bit SAR ADC up to 18 channels
DAC	✓ 8 bit up to 2 channels

Dimensions

Package	QFN-48
Height	0.23622 in (6 mm)
Width	0.23622 in (6 mm)

ESP32 DevKitC_V4 Board
ESP32-WROVER Module
→ Incompatible!!!



See how to assemble the ESP32 IoT board
<https://codezoo.tistory.com/65>



ESP32 IoT 보드 조립 방법

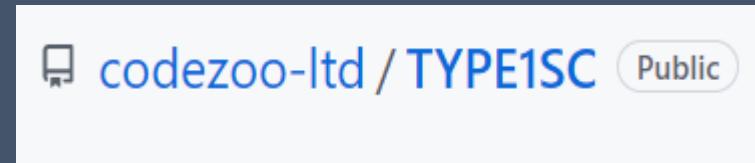
조립 & 하드웨어 · 2023. 9. 19. 09:36

ESP32 IoT 보드가 조립된 모습을 보면 복잡해 보입니다. 조립 방법에 대한 문의가 있어서, 순서대로 작업하면서 기록했습니다. 이 방법이 정답은 아니지만, 그래도...

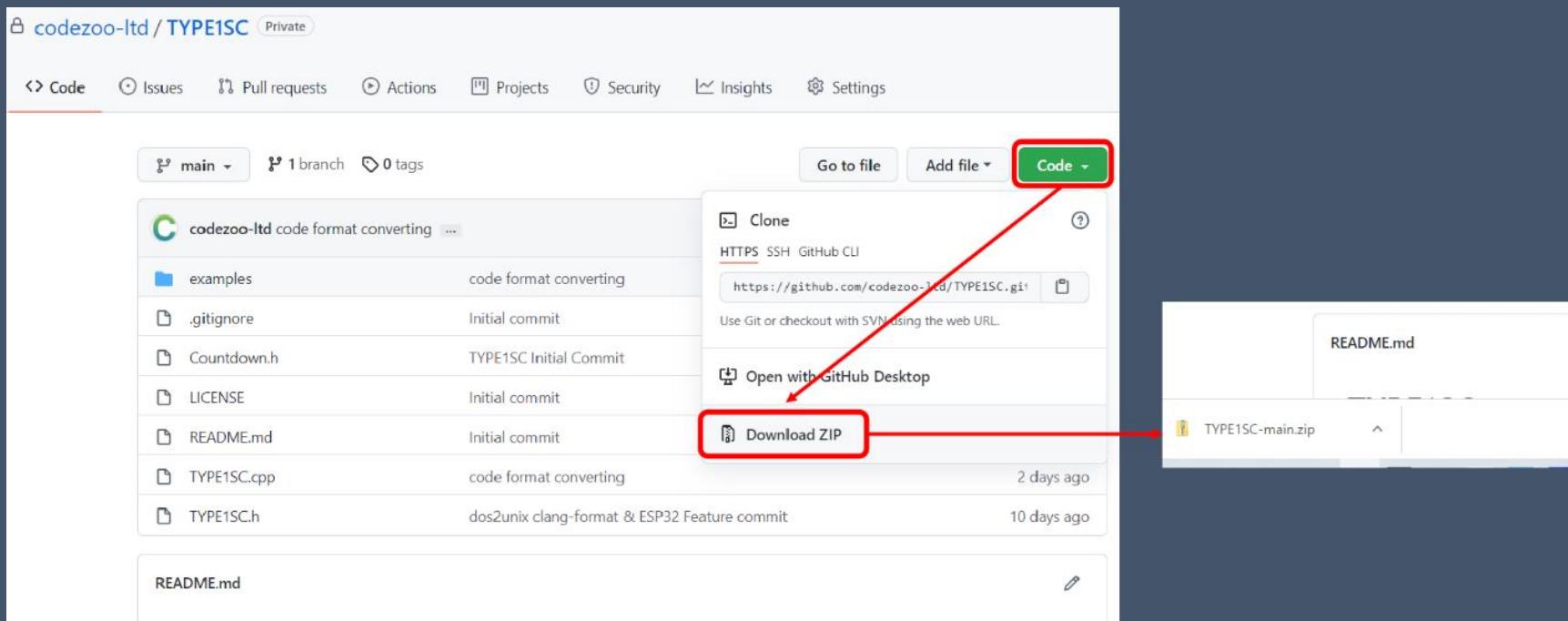
[Read More](#)

5. *ESP32 IoT Hands-on (monitoring)*

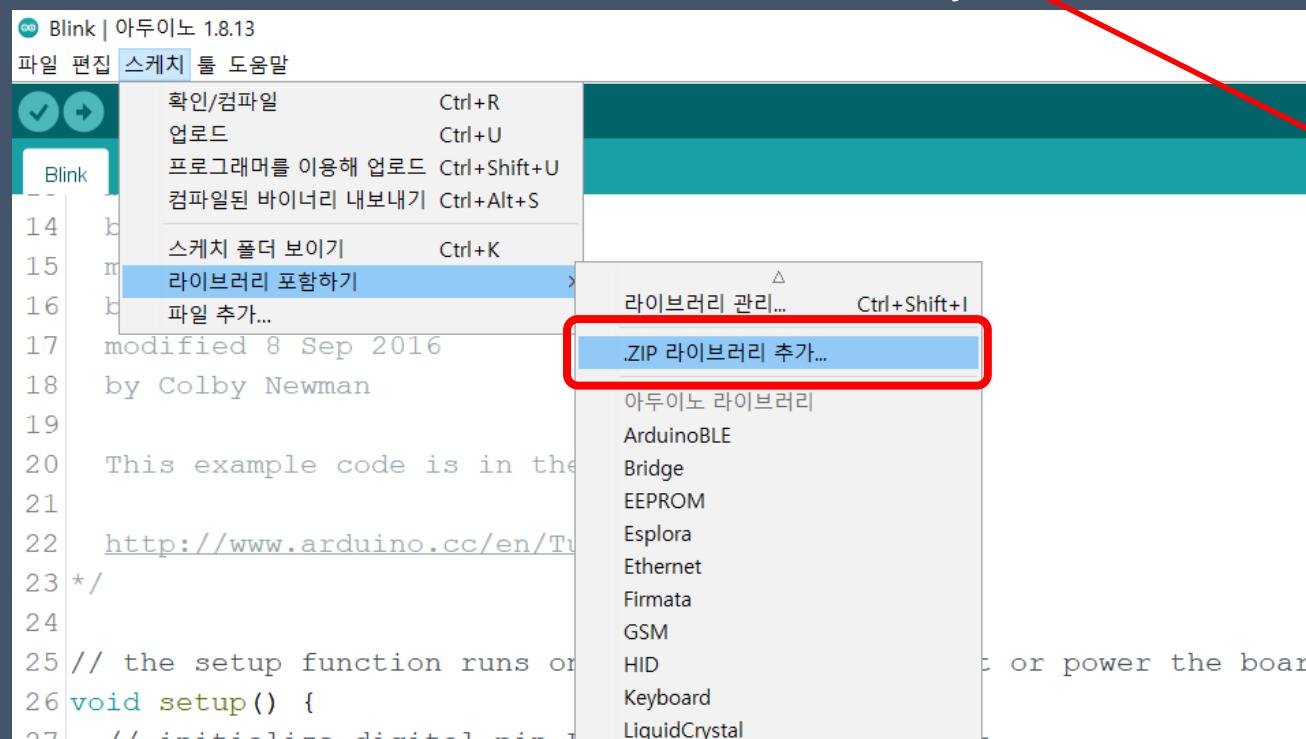
- ✓ CodeZoo CATM1 Library Download



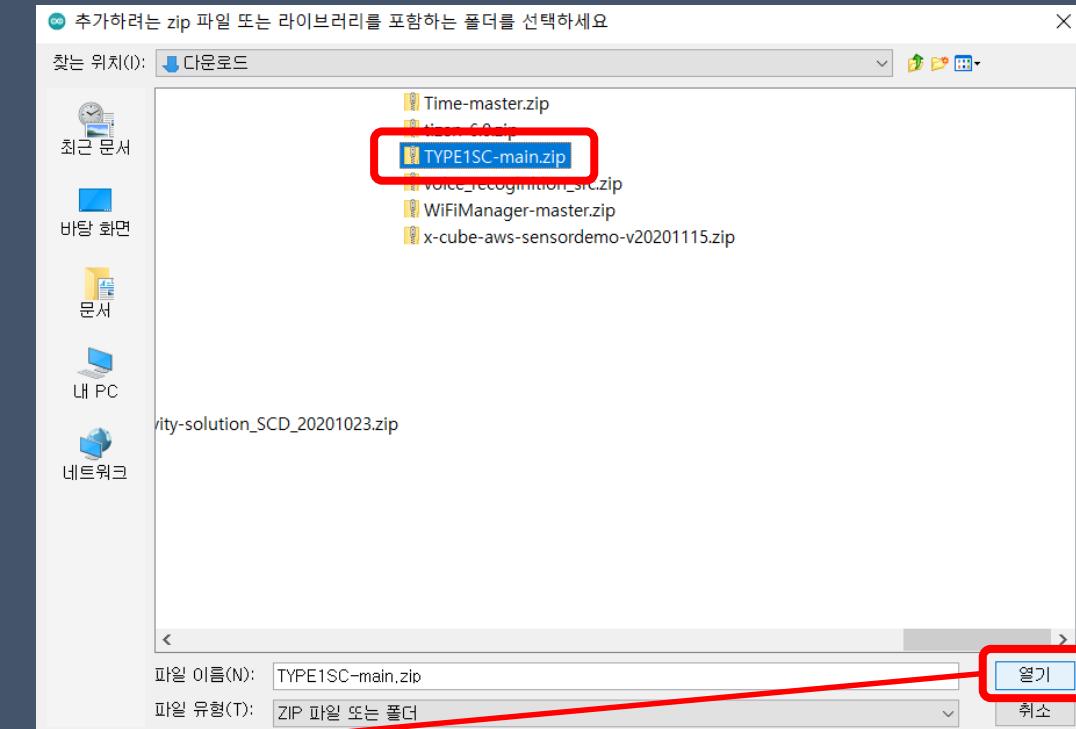
- ✓ Download the Library Install : TYPE1SC, ppposclient for CodeZoo CATM1 development and follow the steps below. After downloading, install it on your Arduino Sketch.



Sketch -> Include Libraries -> Add .ZIP Library...



Downloaded directory -> TYPE1SC-main.zip -> Open

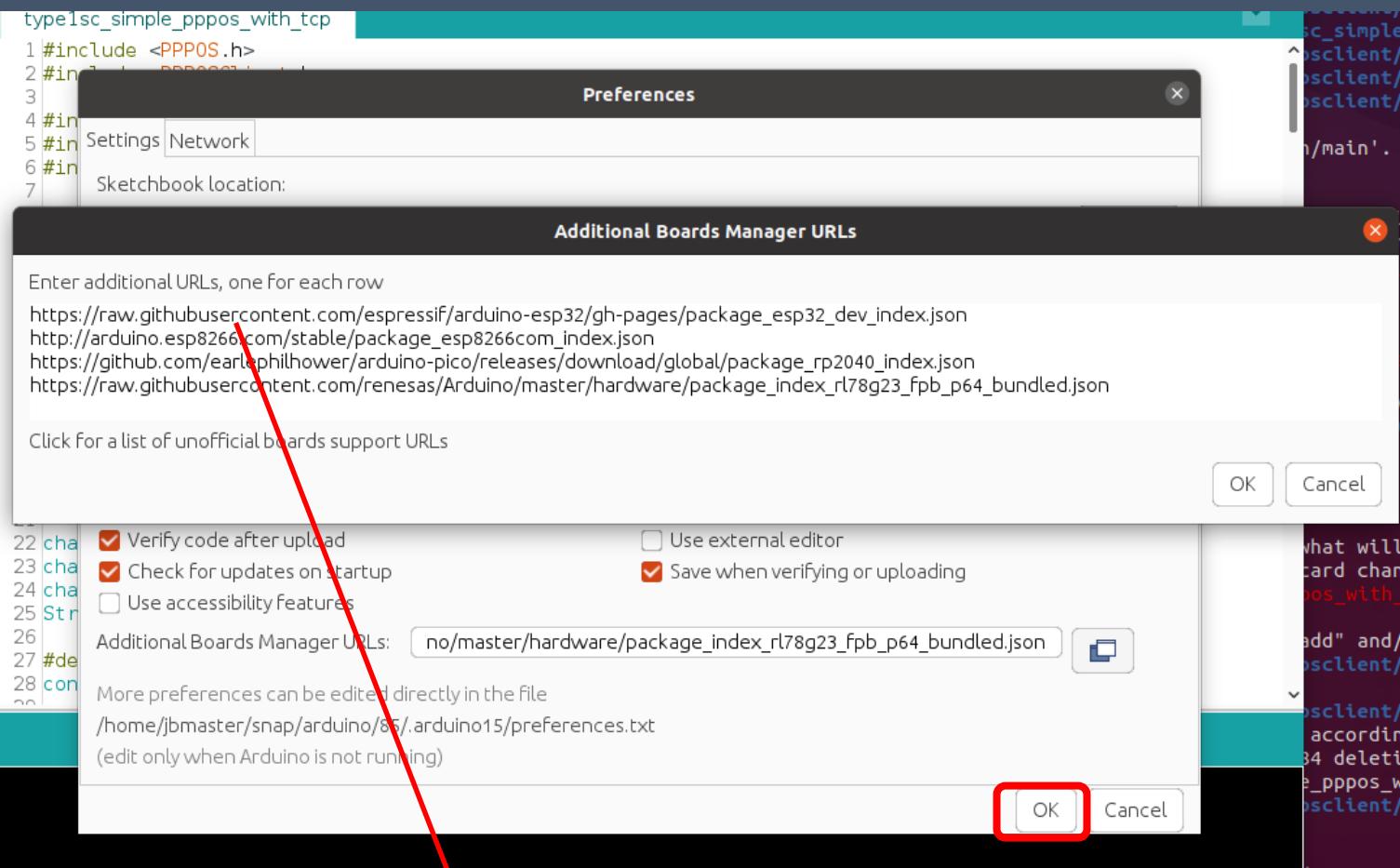


라이브러리가 추가되었습니다. "라이브러리 포함하기" 메뉴를 확인하세요

스케치는 프로그램 저장 공간 1118 바이트(2%)를 사용. 최대 49152 바이트.

전역 범주는 동적 메모리 22바이트(0%)를 사용, 6122바이트의 지역변수가 남음, 최대는 6144 바이트.

ESP32 package install



File -> Preferences -> Additional Board Manager URLs -> Add the URLs below -> OK

https://espressif.github.io/arduino-esp32/package_esp32_index.json

TYPE1SC_AWSIOT_demo | 아두이노 1.8.13

파일 편집 스케치 툴 도움말

자동 포맷 Ctrl+T

스케치 보관하기

인코딩 수정 & 새로 고침

라이브러리 관리... Ctrl+Shift+I

시리얼 모니터 Ctrl+Shift+M

시리얼 플로터 Ctrl+Shift+L

보드: "Arduino Uno"

보드 매니저...

포트

보드 정보 얻기

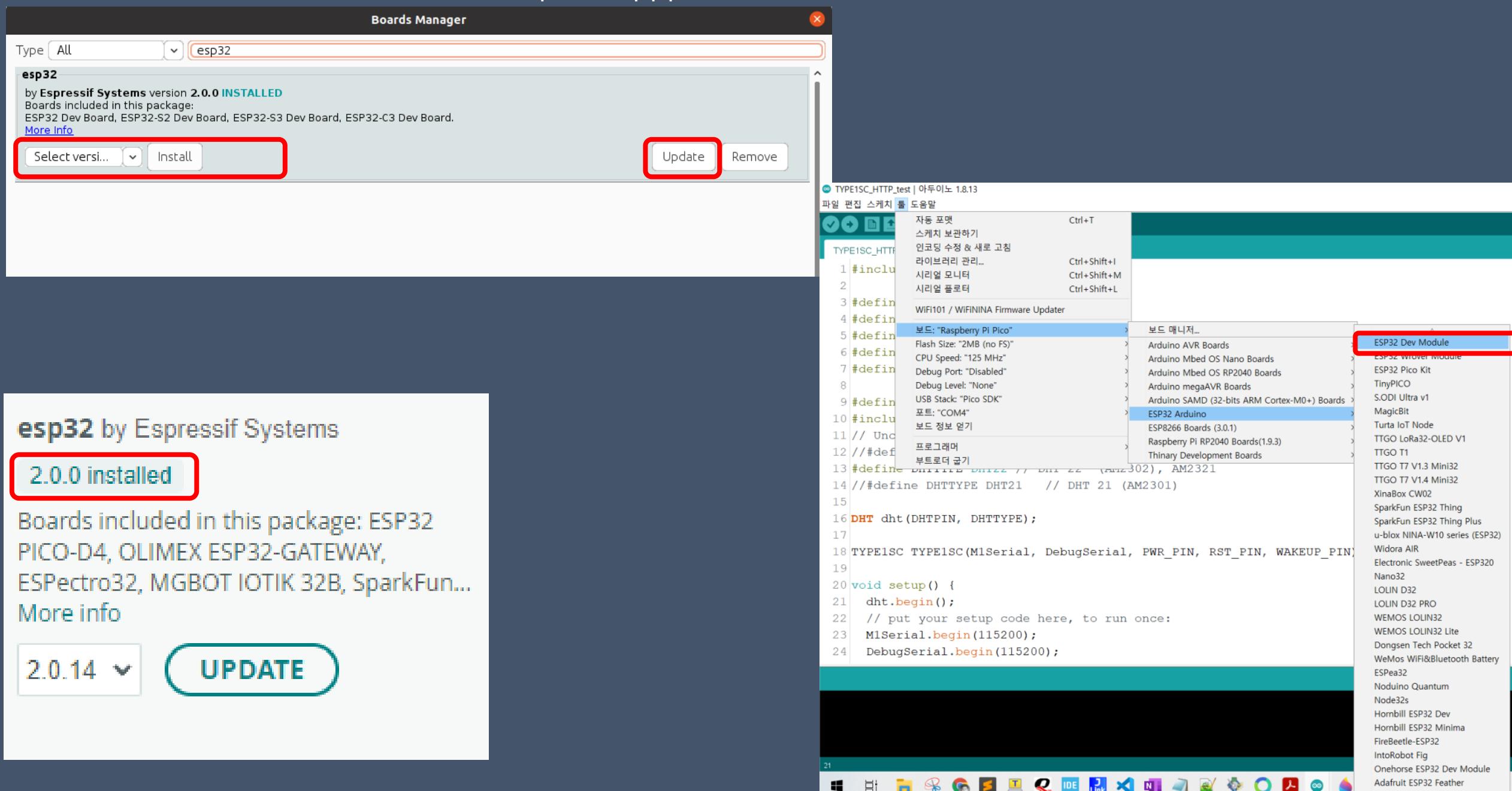
프로그래머: "AVRISP mkII"

부트로더 굽기

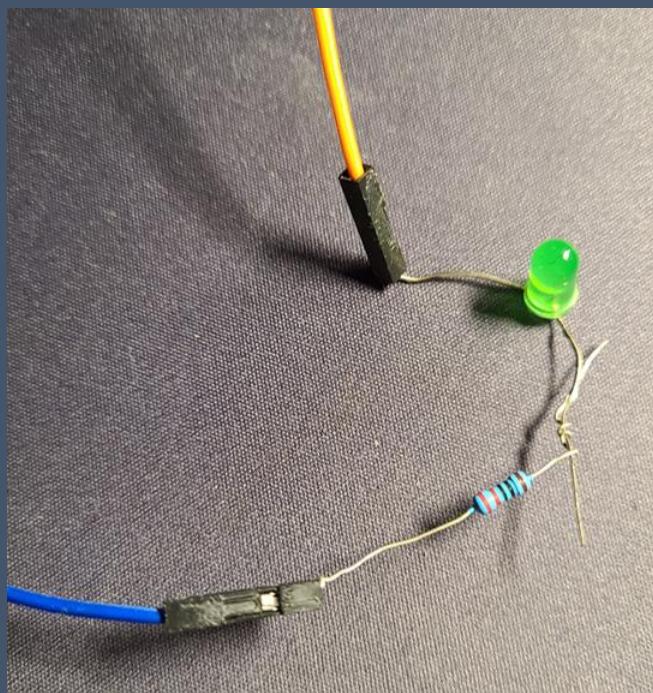
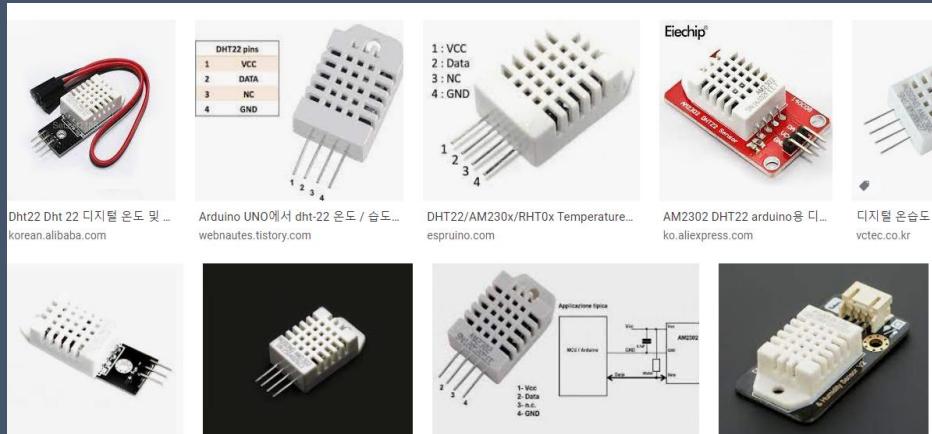
```
1 #include "DHT.h" /* https://github.com/adafruit/DHT-sensor-library */  
2 // Uncomment whatever type you're using:  
3 // #define DHTTYPE DHT11 // DHT 11  
4 #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321  
5 // #define DHTTYPE DHT21 // DHT 21 (AM2301)  
6  
7 DHT dht(DHTPIN, DHTTYPE);  
8  
9 TYPE1SC TYPE1SC(M1Serial, DebugSerial);  
10  
11 void setup() {  
12     // initialize digital pin LED_BUILTIN as an output.  
13 }
```

Board Manager

- ✓ Arduino ESP32 2.0.0 SDK install or Update (pppos available version)



Hardware configuration for example labs



LED + Resistor (1K)

+ Pole (+Resistor1K): Connect to ESP IoT board '23'

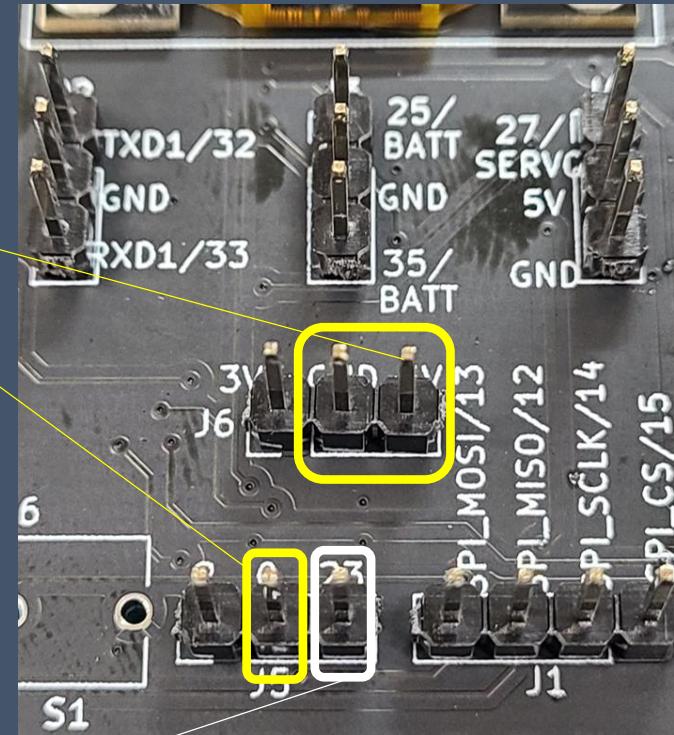
- Pole: ESP IoT board GND connection

DHT22 Temperature and Humidity Sensor

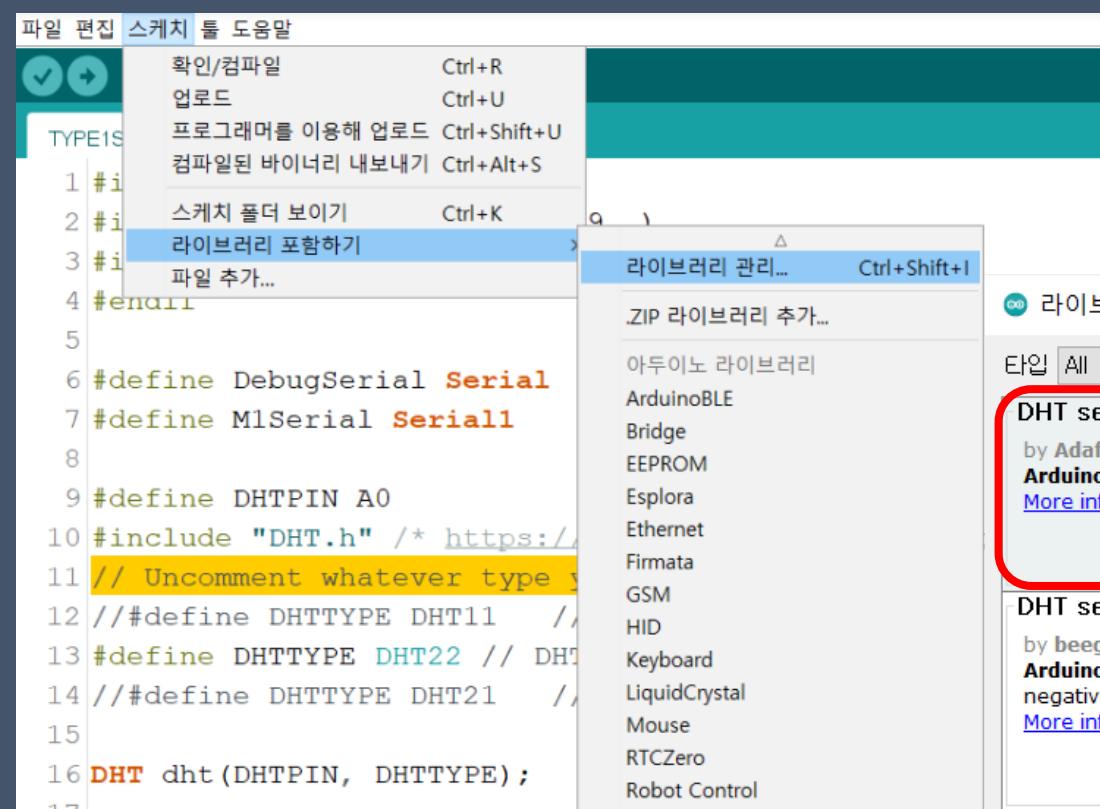
VCC : ESP IoT board 5V connection

DATA : ESP IoT board '0' connection

GND : ESP IoT board GND connection



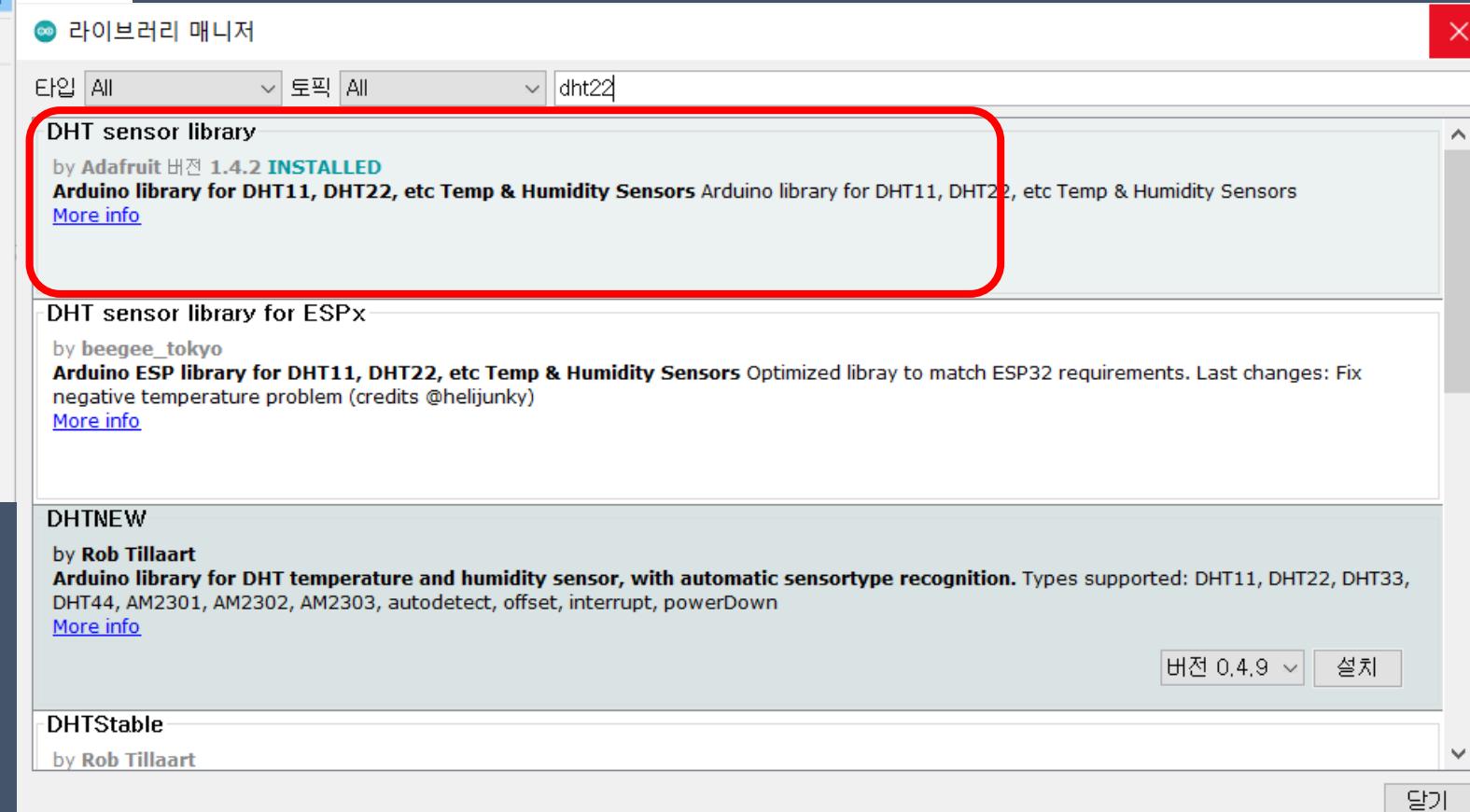
Installing libraries for example labs



```
파일 편집 스케치 툴 도움말
확인/컴파일 Ctrl+R
업로드 Ctrl+U
프로그래머를 이용해 업로드 Ctrl+Shift+U
컴파일된 바이너리 내보내기 Ctrl+Alt+S
스케치 폴더 보기 Ctrl+K
라이브러리 포함하기
파일 추가...
1 #include <SoftwareSerial.h>
2 #include <DHT.h>
3 #define DHTPIN A0
4 #define DHTTYPE DHT22 // DHT
5
6 #define DebugSerial Serial
7 #define M1Serial Serial
8
9 #define DHTPIN A0
10 #include "DHT.h" /* https://github.com/adafruit/DHT-sensor-library */
11 // Uncomment whatever type you're using!
12 // #define DHTTYPE DHT11 // DHT
13 #define DHTTYPE DHT22 // DHT
14 // #define DHTTYPE DHT21 // DHT
15
16 DHT dht(DHTPIN, DHTTYPE);
```

Install the DHT22 Temperature and Humidity Sensor Library

Sketch -> Include Libraries -> Manage Libraries->
Install DHT sensor library



라이브러리 관리... Ctrl+Shift+I
.ZIP 라이브러리 추가...

타입 All 토픽 All dht22

DHT sensor library
by Adafruit 버전 1.4.2 INSTALLED
[Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors](#) Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
[More info](#)

DHT sensor library for ESPx
by beegee_tokyo
[Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors](#) Optimized libray to match ESP32 requirements. Last changes: Fix negative temperature problem (credits @helijunky)
[More info](#)

DHTNEW
by Rob Tillaart
[Arduino library for DHT temperature and humidity sensor, with automatic sensortype recognition.](#) Types supported: DHT11, DHT22, DHT33, DHT44, AM2301, AM2302, AM2303, autodetect, offset, interrupt, powerDown
[More info](#)

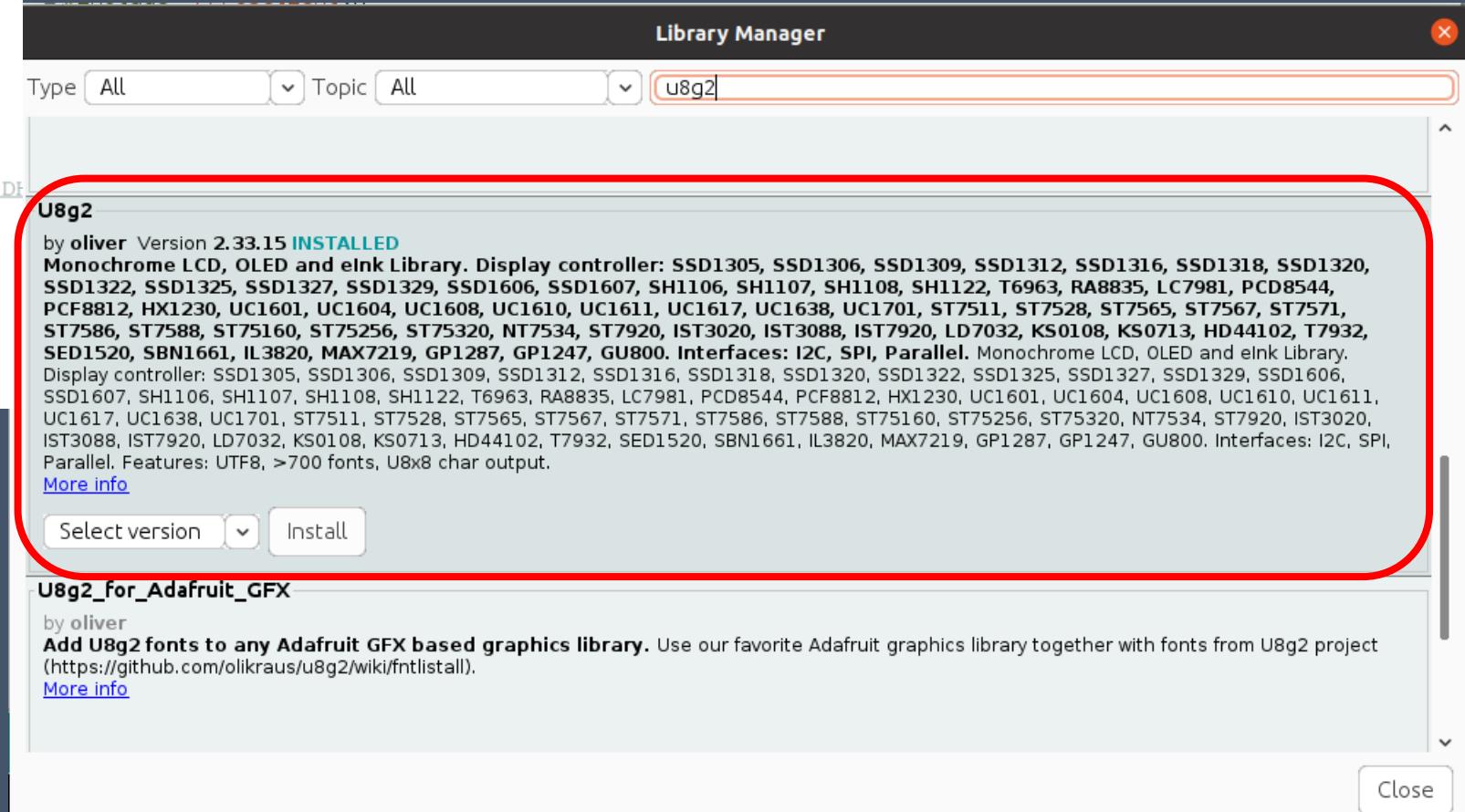
DHTStable
by Rob Tillaart

버전 0.4.9 설치 닫기

```
파일 편집 스케치 툴 도움말
확인/컴파일 Ctrl+R
업로드 Ctrl+U
프로그래머를 이용해 업로드 Ctrl+Shift+U
컴파일된 바이너리 내보내기 Ctrl+Alt+S
1 #include <Adafruit_SSD1306.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5
6 #define DebugSerial Serial
7 #define M1Serial Serial1
8
9 #define DHTPIN A0
10 #include "DHT.h" /* https://github.com/adafruit/DHT-sensor-library */
11 // Uncomment whatever type you're using!
12 // #define DHTTYPE DHT11
13 #define DHTTYPE DHT22 // DHT 22
14 // #define DHTTYPE DHT21
15
16 DHT dht(DHTPIN, DHTTYPE);
```

Installing the U8g2 Library (OLED LCD Output)

Sketch -> Include Library -> Manage Library -> Install U8g2 library



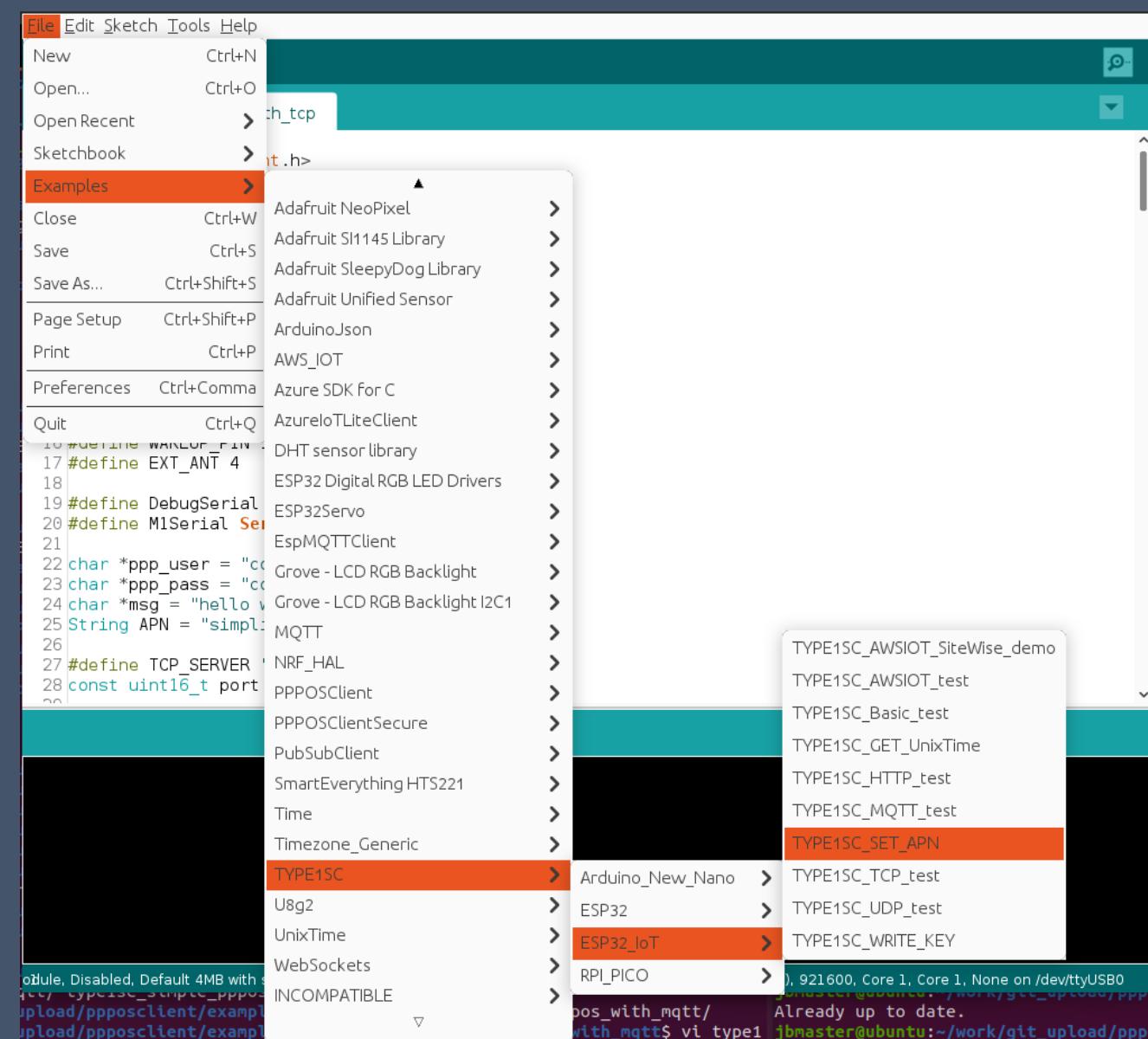
1) APN settings

- APN (Access Point Name)

A gateway between a carrier's cellular network and the Internet. An IP address is generated by the carrier using APN settings. In conclusion, in order to communicate with the network using LTE CAT.M1, it must be set up. After setup, the APN value is stored in the CAT.M1 modem internal filesystem.

Afterwards, it is only used to change the APN

1) APN settings



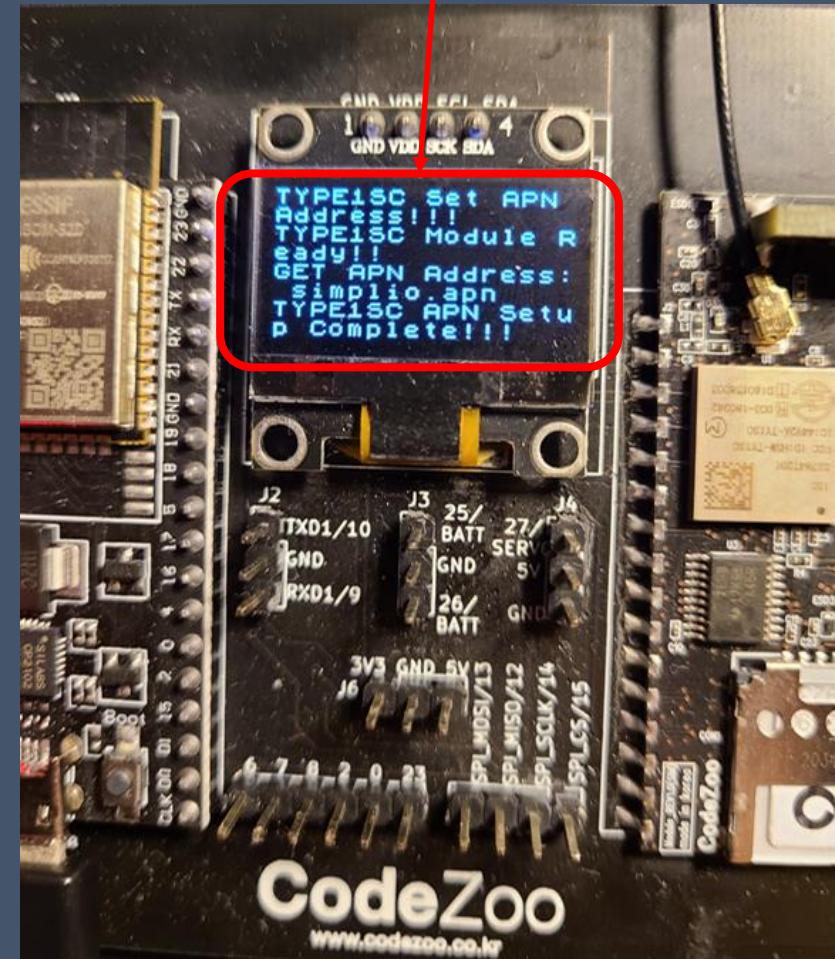
✓ Examples -> TYPE1SC-main -> ESP32_IoT ->
Select TYPE1SC_SET_APN

✓ Vodafone global IoT Sim APN :
simplio.apn
Simplio.ipwa.apn

1) APN settings

Verify APN -> Build -> Run -> Check Results

```
65     char *apnAddr = "simplio.apn"; /* Vodafone Global IoT SIM APN */
66
67     if (TYPE1SC.setAPN(apnAddr) == 0) {
68         DebugSerial.println("TYPE1SC Set APN Address!!!");  
69         u8x8log.print("TYPE1SC Set APN Address!!!\n");
70     }
71
72     /* Board Reset */
73     TYPE1SC.reset();
74     delay(2000);
75
76     /* TYPE1SC Module Initialization */
77     if (TYPE1SC.init()) {
78         DebugSerial.println("TYPE1SC Module Error!!!");  
79         u8x8log.print("TYPE1SC Module Error!!!\n");
80     }
81
```



2) Basic Test

After initializing the CAT.M1 modem and connecting to the carrier network, get the following information.

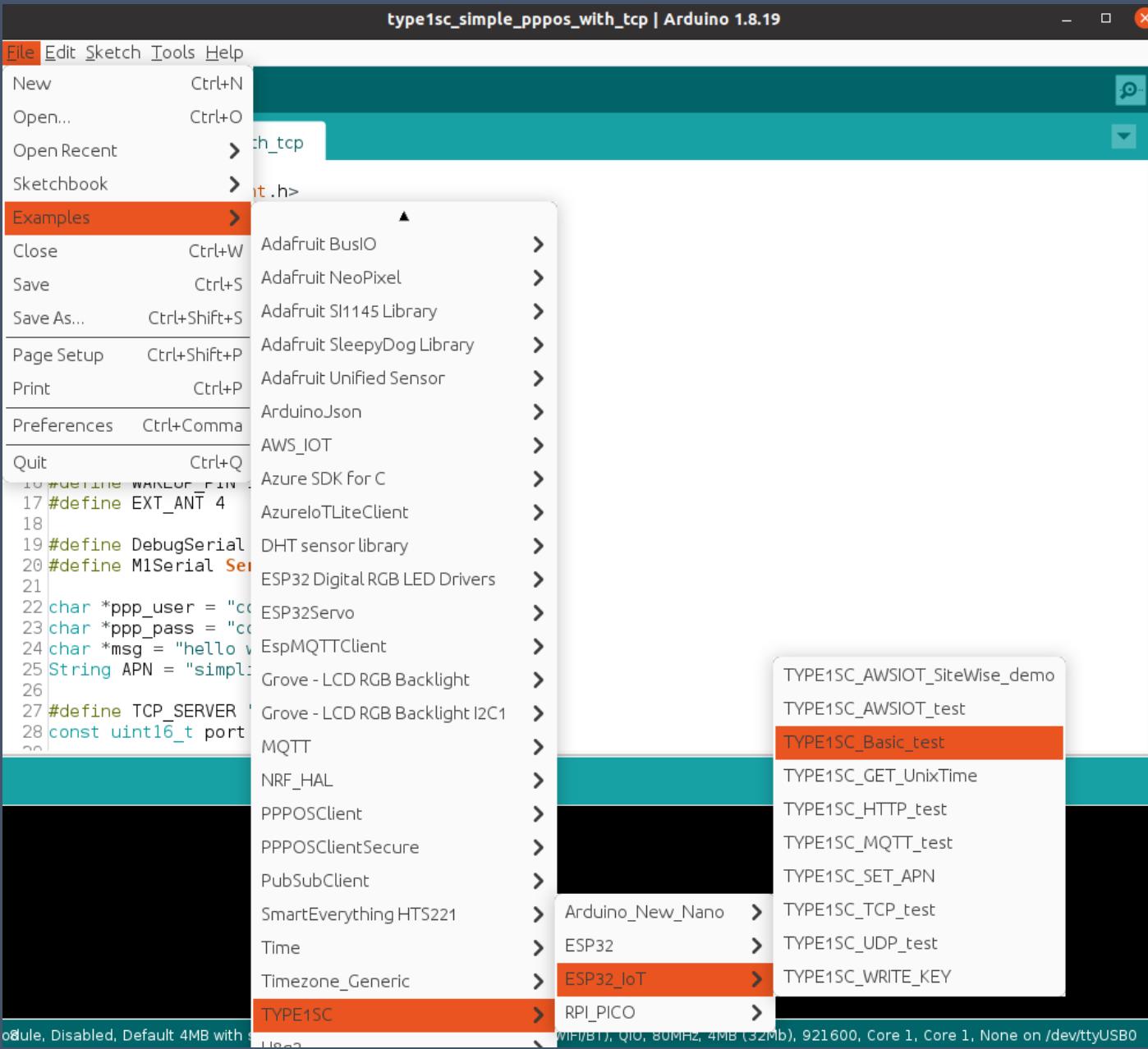
1. CIMI: Gets the value of IMSI (International Mobile Subscriber Identity)
2. IMEI: Gets the unique value (phone's fingerprint) of the International Mobile Equipment Identity modem.
3. ICCID: Integrated Circuit Card Identifier (ICCID) Gets the unique serial number of the SIM card.
4. CGMR: Modem firmware version
5. Time: Get the current time (obtained upon base station registration)
6. RSSI, RSRP, RSRQ, SINR, Cell ID, TX Power
Gets wireless communication sensitivity, signal strength, base station network ID, and transmission strength.

[Check]

Basic Test outputs debug messages to the ESP32 USB port and Basic_test.ino outputs debug messages to the ESP32 Serial1. Basic_test_Serial1.ino, which outputs debug messages to the ESP32 USB port, and Basic_test_Serial1.ino, which outputs debug messages to the ESP32 Serial1 via USB_TTL.

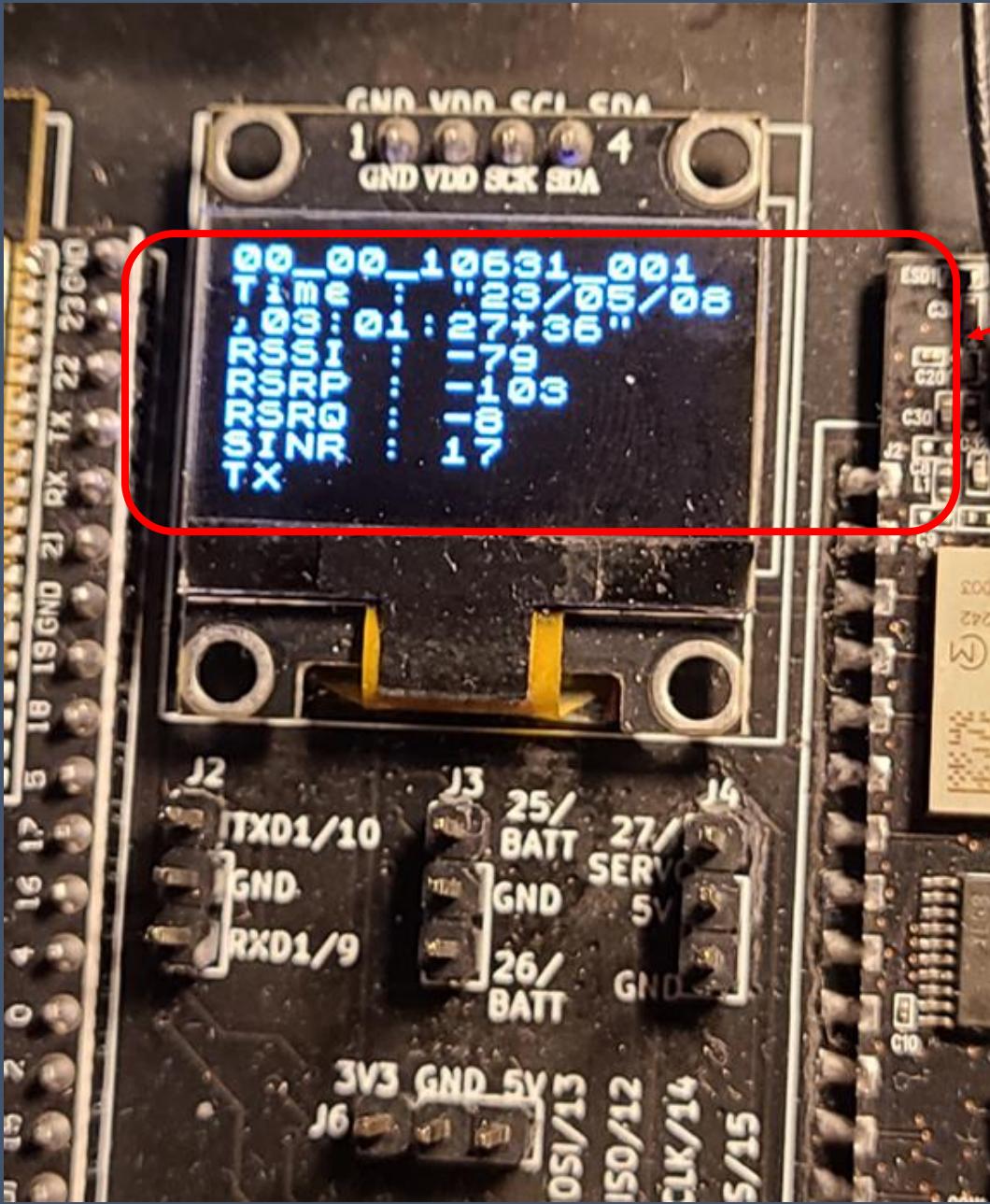
Refer to the Basic_test_Serial1.ino example to connect and use serial devices, including GPS.

2) Basic Test



- ✓ Examples -> TYPE1SC-main -> ESP32_IoT -> Select TYPE1SC_Basic_test
- ✓ Since the wireless network status is changed by the surrounding environment so it is used to measure the communication environment.

2) Basic Test

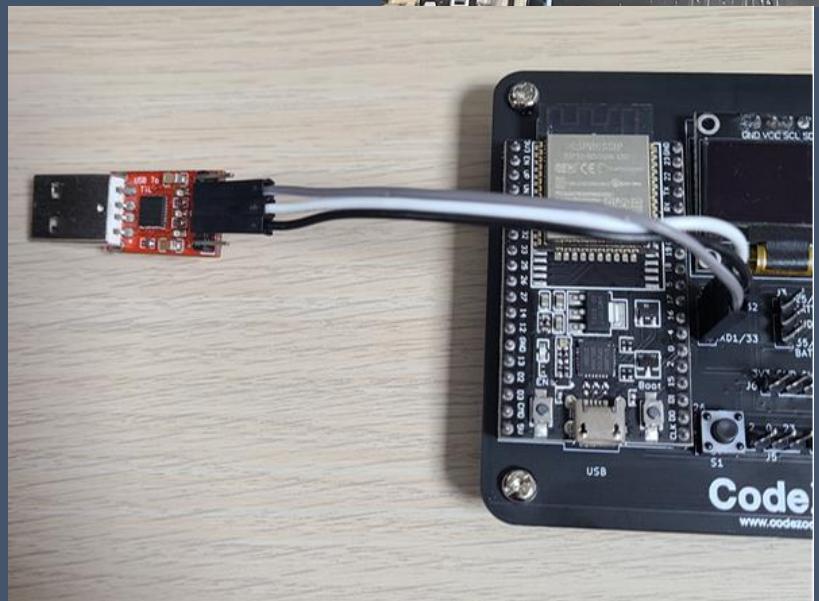
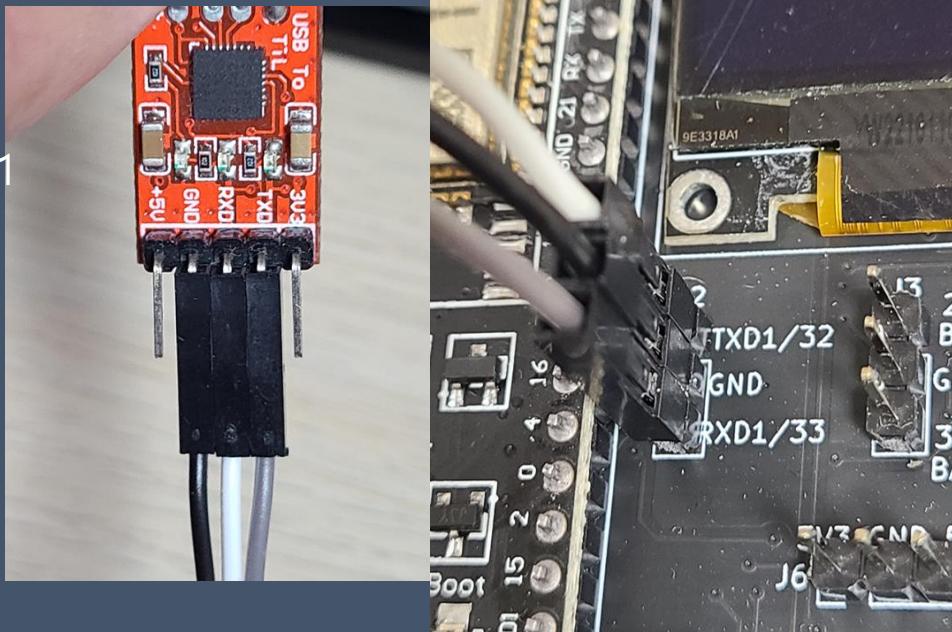
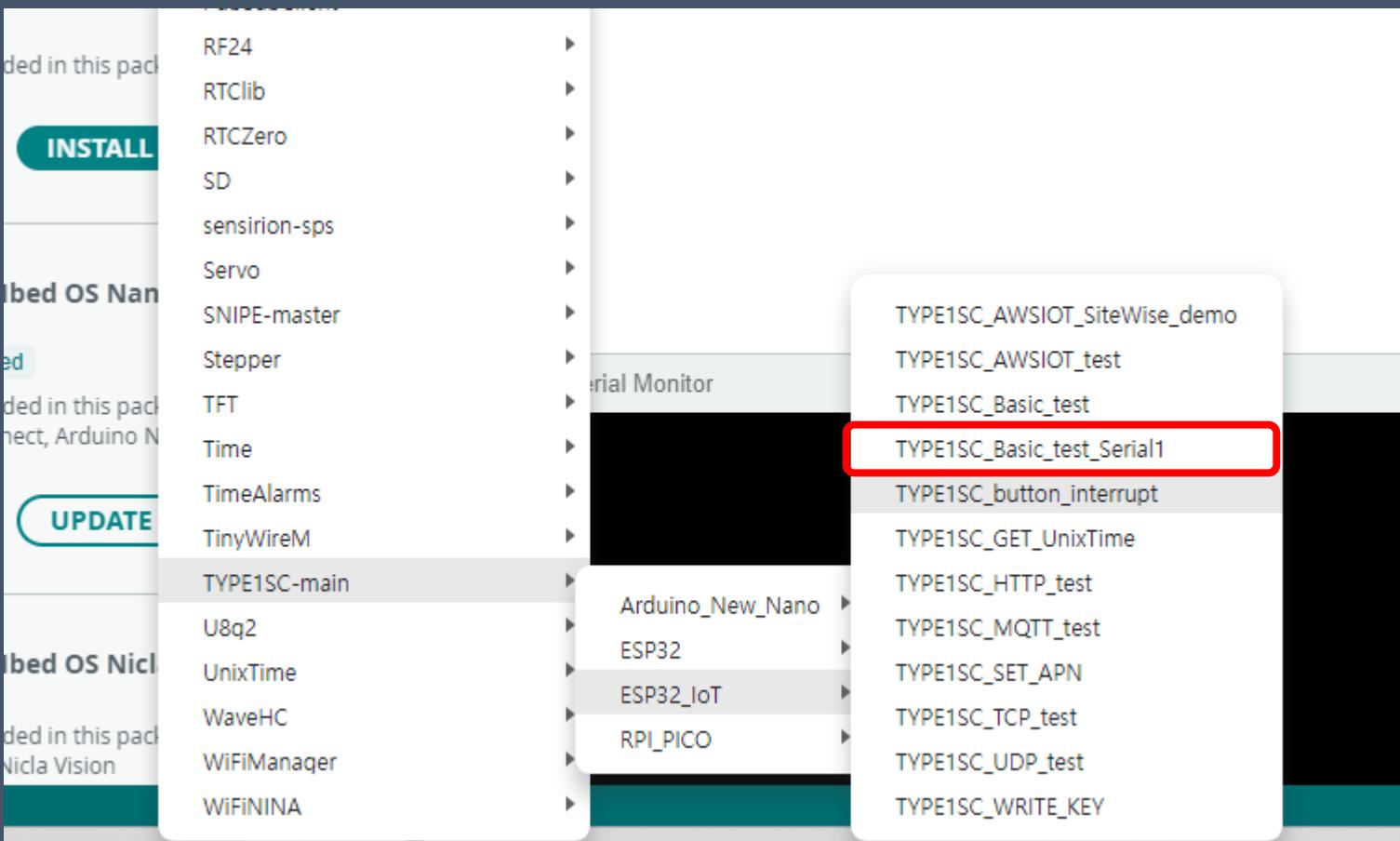


Signal sensitivity measured at the current location, Cell ID, and transmit strength

2) Basic Test Serial1

✓ Examples -> TYPE1SC-main -> ESP32_IoT -> Select TYPE1SC_Basic_test_Serial1

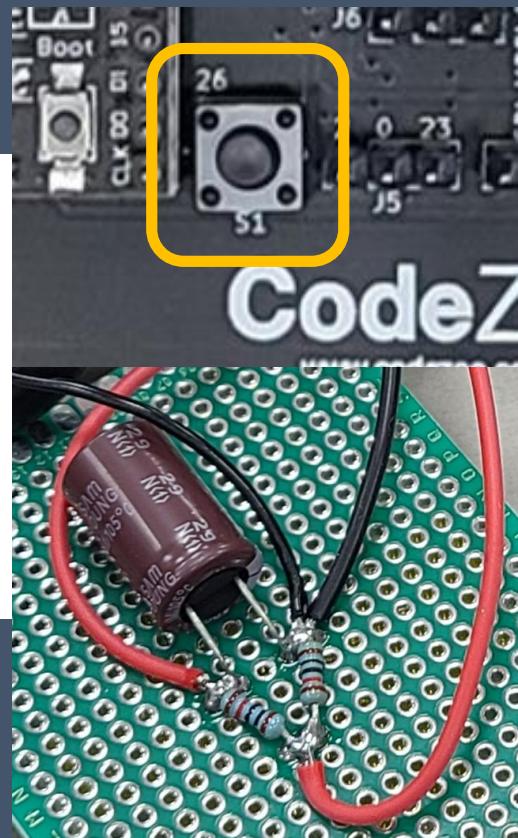
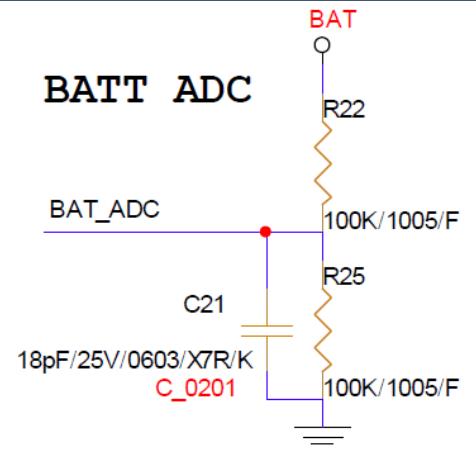
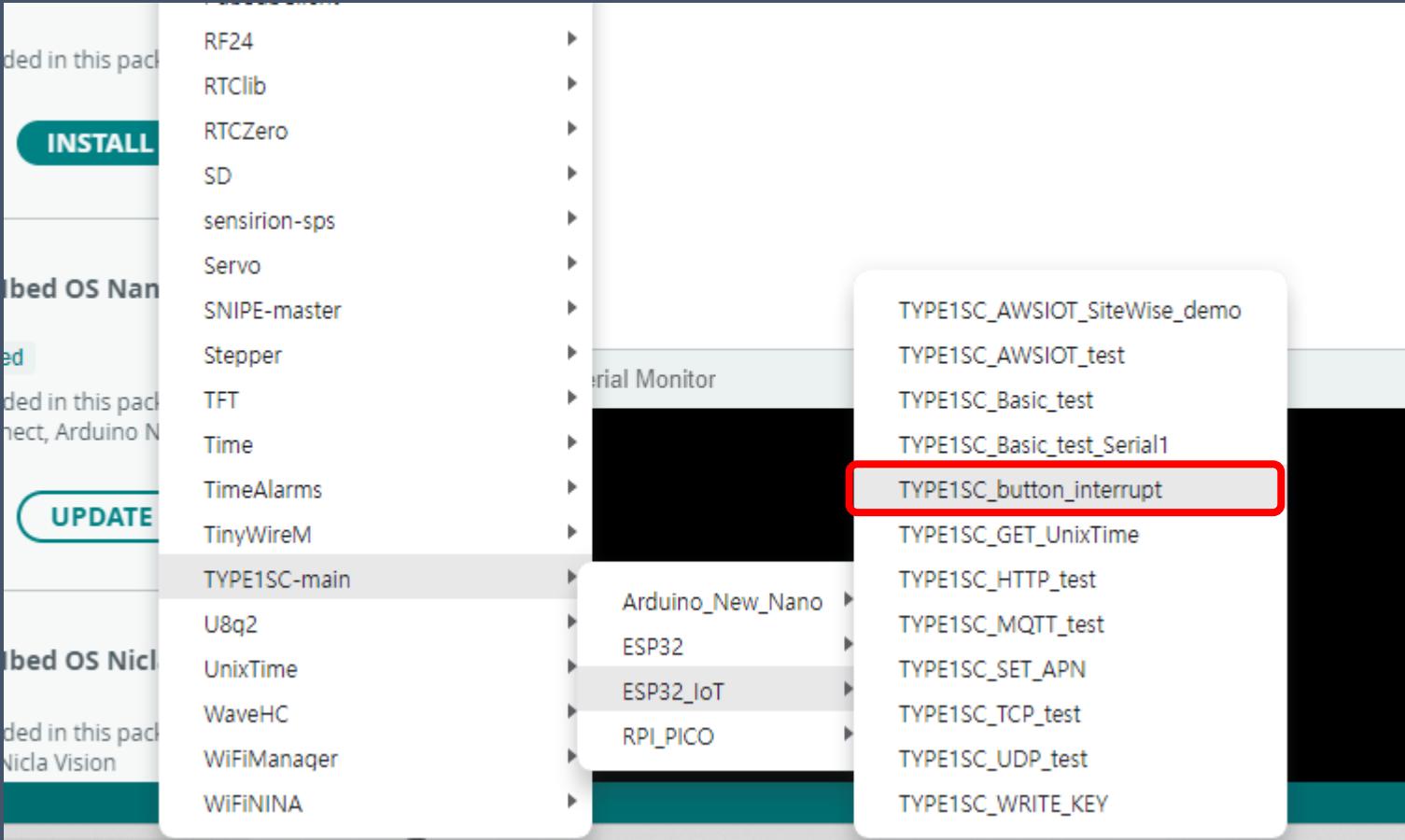
- ✓ USB TTL TXD ---> ESP32_IoT_RXD1
- ✓ USB TTL RXD ---> ESP32_IoT_TXD1
- ✓ USB TTL GND---> ESP32_IoT_GND



3) Button Interrupt

The Button Interrupt example is an implementation of how to use the Button built into the ESP32 IoT board. The example is implemented to measure the battery voltage when the button is pressed. To measure the battery voltage We need the voltage distribution circuit shown below.

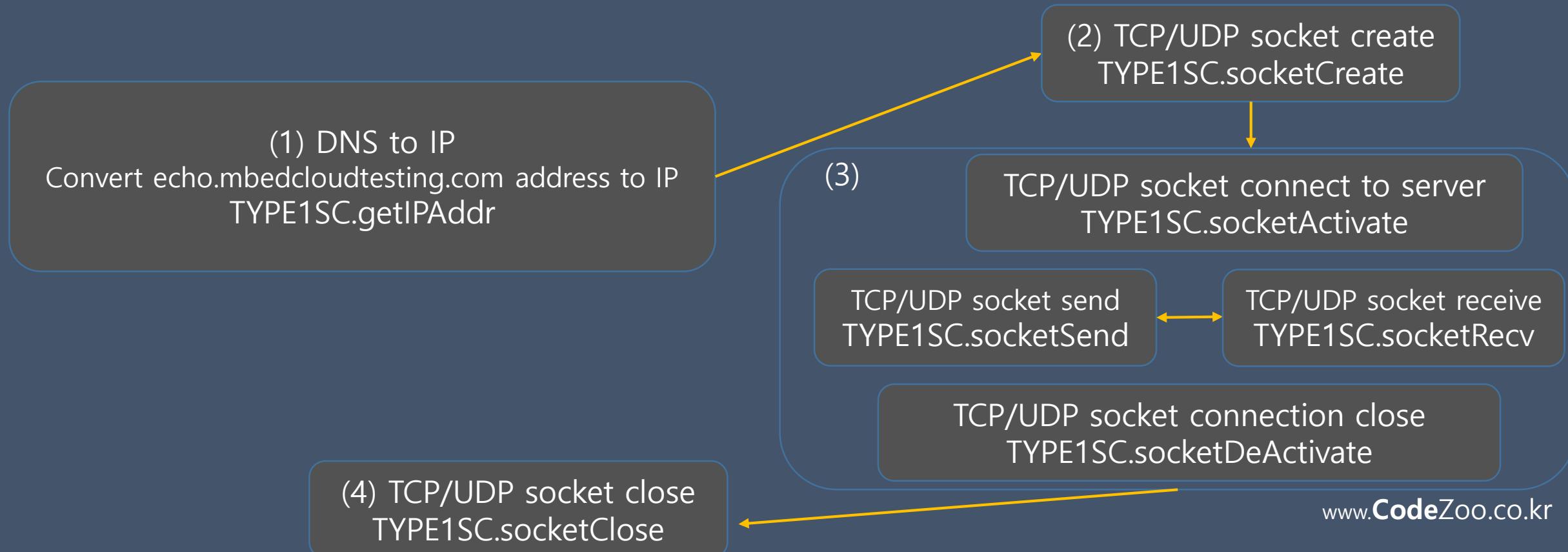
- ✓ Examples -> TYPE1SC-main -> ESP32_IoT -> Select TYPE1SC_button_interrupt



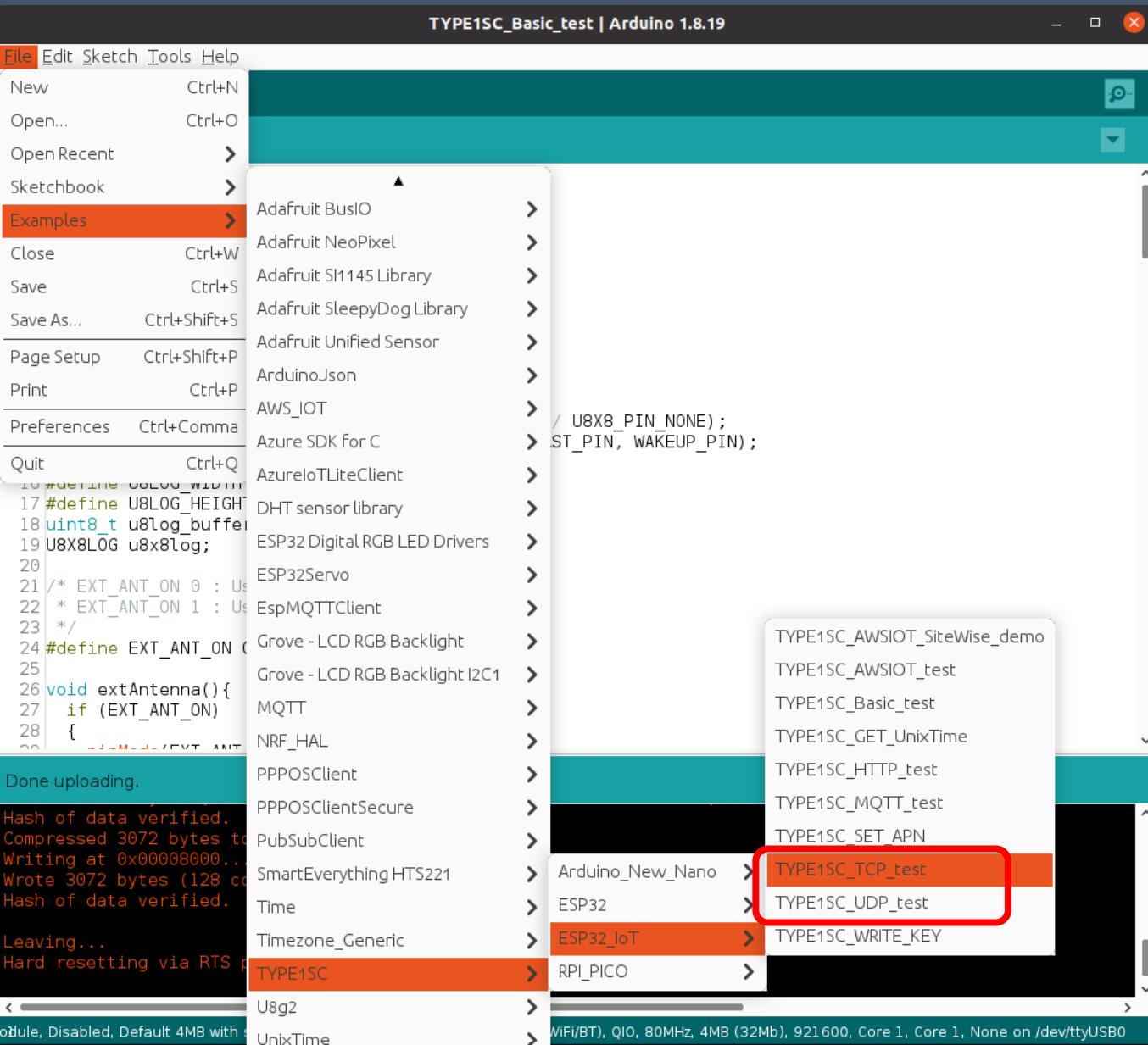
4) TCP, UDP Test

The CAT.M1 modem uses the internal software stack to connect to the echo server and send data, and receives data back.

1. the CAT.M1 modem creates TCP, UDP sockets inside, so the Arduino board only passes commands.
2. echo server is echo.mbedcloudtesting.com, and port is 7. (Both TCP/UDP use port 7)
3. The TCP, UDP socket behavior scenario of the Type1SC module is as follows.

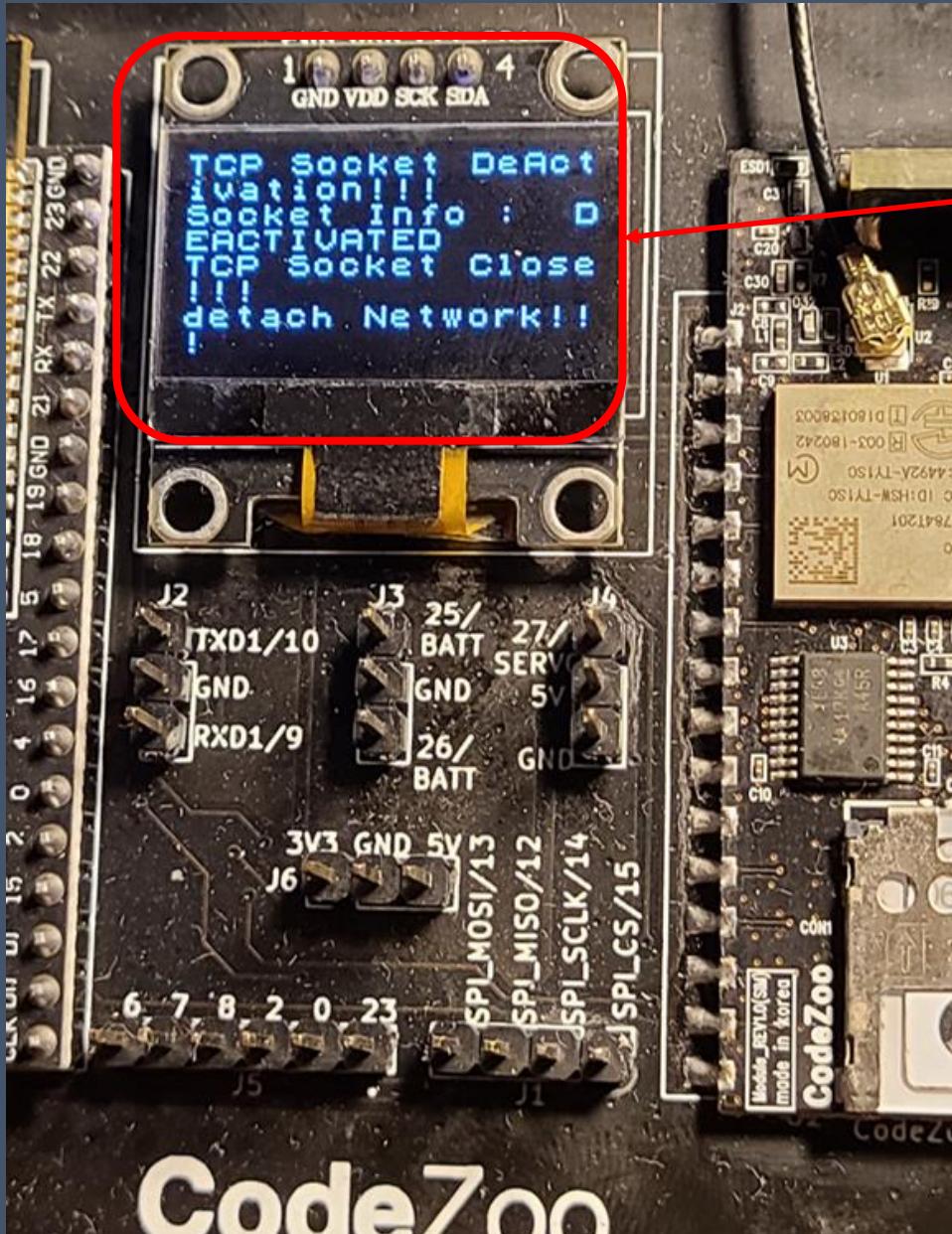


4) TCP, UDP Test



✓ Examples -> TYPE1SC-main -> ESP32_IoT ->
Select TYPE1SC_TCP_test or TYPE1SC_UDP_test

4) TCP, UDP Test



One-time run for TCP/UDP socket communication scenarios.

LTE Cat.M1 communication is a communication method that uses a monthly usage allowance data usage check is required.

5) HTTP Test

The CAT.M1 modem uses the internal communication stack to upload the temperature and humidity data measured by the Arduino board to the Thingspeak.com server via HTTP protocol. HTTP protocol to the Thingspeak.com server for real-time monitoring on a PC or smartphone.

1. Sign up and set up to use Thingspeak.com.
2. Connect to Thingspeak.com with CAT.M1 modem internal TCP socket and measure temperature, humidity data with DHT22 sensor on the Arduino board. After measuring the temperature and humidity data, create an HTTP protocol to upload the data to Thingspeak.com.

What is ThingSpeak?

- One of the IoT Cloud Services Created by MathWorks
- To get started with Thinkspeak, you need to subscribe to MathWorks
- www.thingspeak.com

ThingSpeak™ Channels Apps Support Commercial Use How to Buy HL

ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

Channels Learn More



ThingSpeak for Students and Educators
Implement IoT research projects quickly with built-in MATLAB data analysis tools and real-time sensor data collection



ThingSpeak for Air Quality Monitoring
Build IoT services for remote monitoring of air quality sensors, and create MATLAB models to predict pollution levels



ThingSpeak for Energy Monitoring
Build IoT applications to monitor energy usage, and develop MATLAB models for power signature identification and load forecasting



ThingSpeak for Smart Farming
Build applications for remote monitoring of sensors deployed on farms, and develop MATLAB models to increase crop yields and reduce costs



Create a ThingSpeak account

1. Go to www.thingspeak.com and click "Get Started For Free"



2. Click 'Create one!'

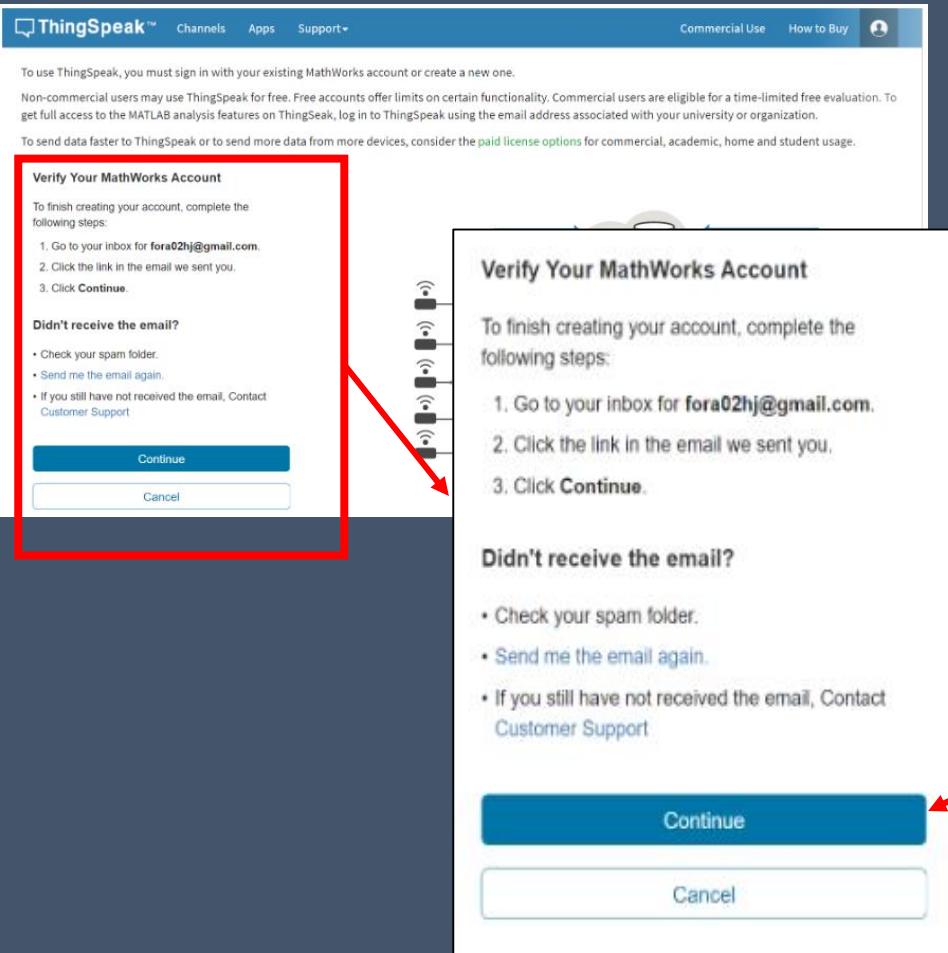


3. Fill in the information and click 'Continue'

The screenshots show the "Create MathWorks Account" process. The first screenshot shows the initial form with a red box around the "Email Address" field. The second screenshot shows the form with the email "fora02h@gmail.com" entered, and a red arrow points to the "Continue" button. The third screenshot shows the final step of the process where all fields (Email, Location, Last Name, First Name, Last Name (English), First Name (English)) are filled, and a red arrow points to the "Continue" button.

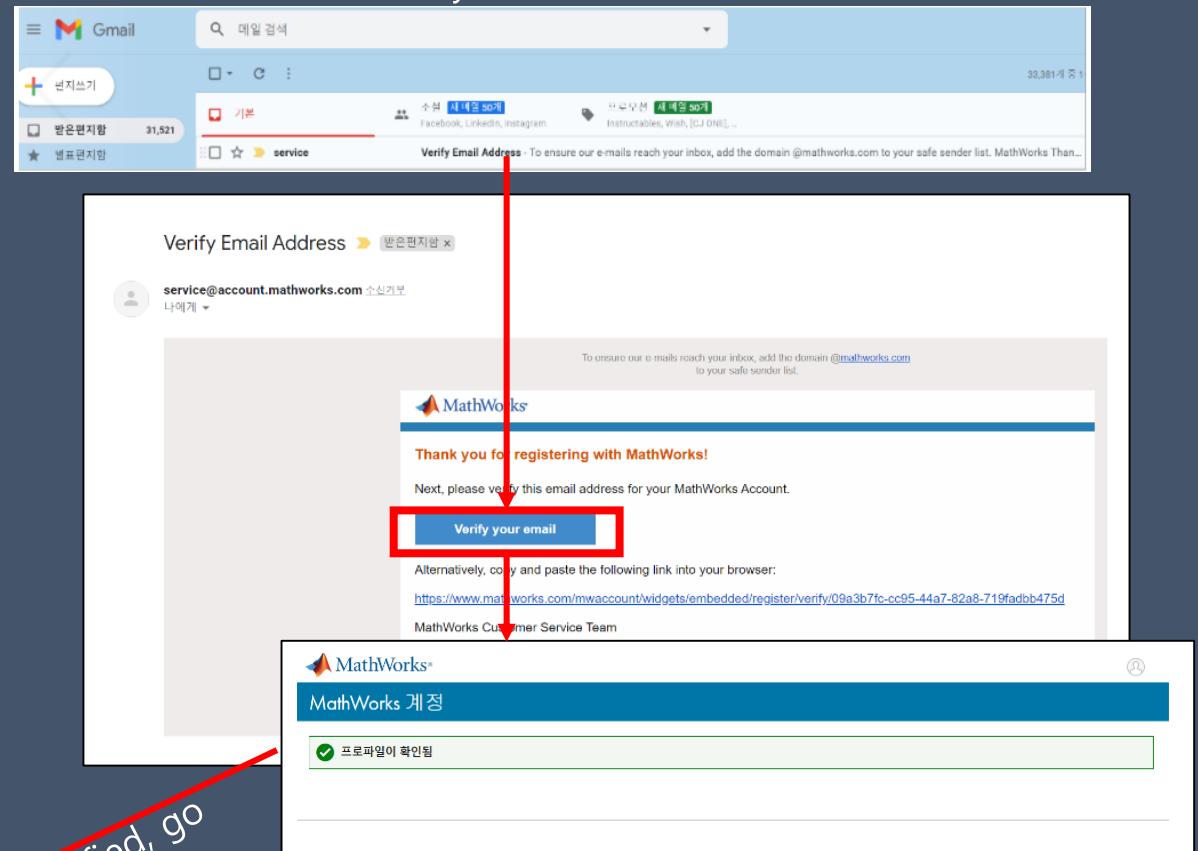
Create a ThingSpeak account

4. See how to authenticate



5. Verify with your email

If you don't see the email, check your spam folder.



Once verified, go back to the previous step and click 'Continue'

Create a ThingSpeak account

6. set your password and click 'Continue'

The screenshot shows the 'Finish your Profile' step of the account creation process. On the left, there's a sidebar with 'Finish your Profile' and two checkboxes: 'I have read and agree to this Privacy Policy' and 'I accept the Online Services Agreement'. Below these are 'Continue' and 'Cancel' buttons. On the right, the main form has a 'Password' field with a strength meter showing 'Strong'. Below it are 'Password Requirements' with three green checkmarks: 'Between 8-50 characters', 'At least 1 upper and lower case letter', and 'At least 1 number'. There are also two checked checkboxes at the bottom: 'I have read and agree to this Privacy Policy' and 'I accept the Online Services Agreement'. The 'Continue' button is highlighted with a large blue rectangle, and a red arrow points from the left towards this button.

7. click 'OK' for successful signup

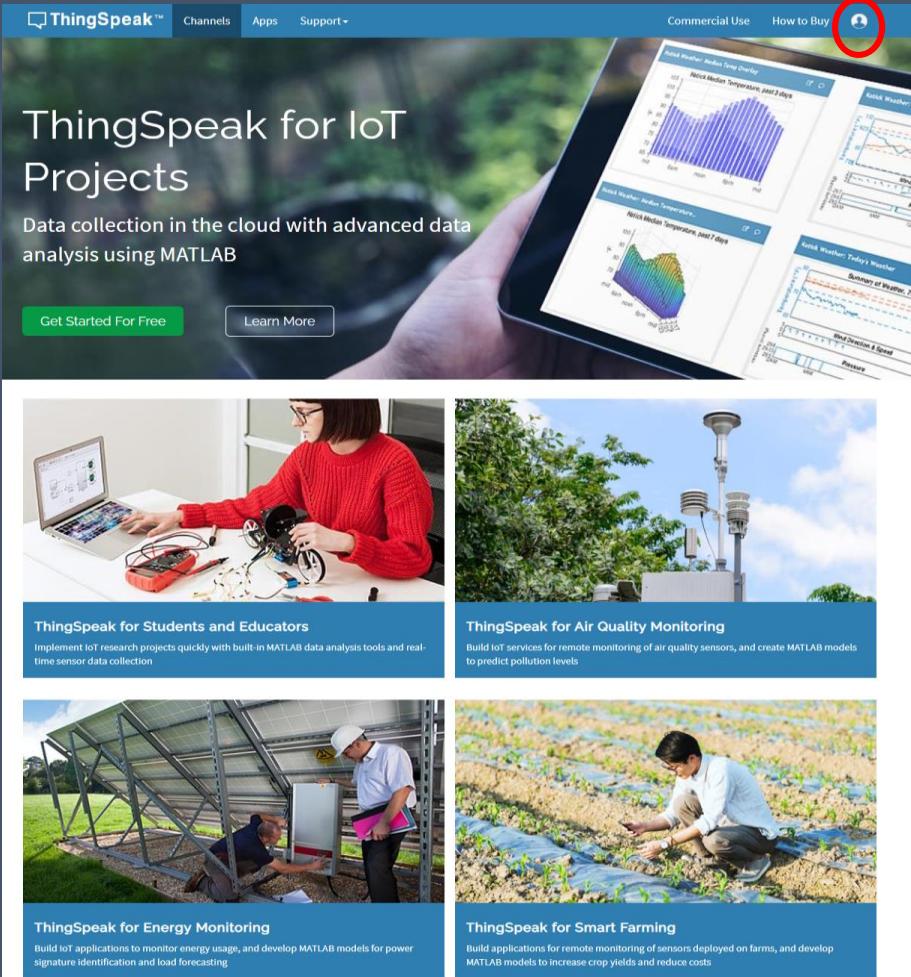


8. Complete the survey

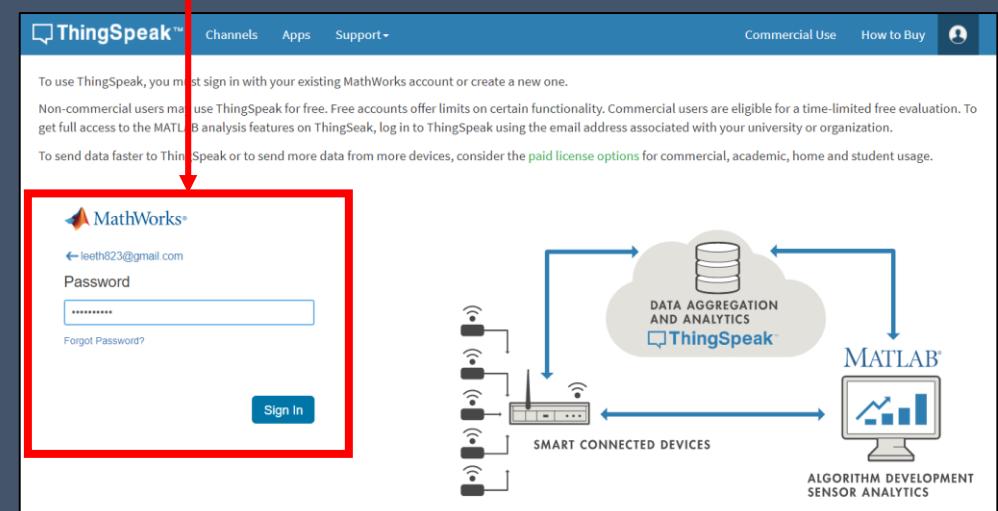
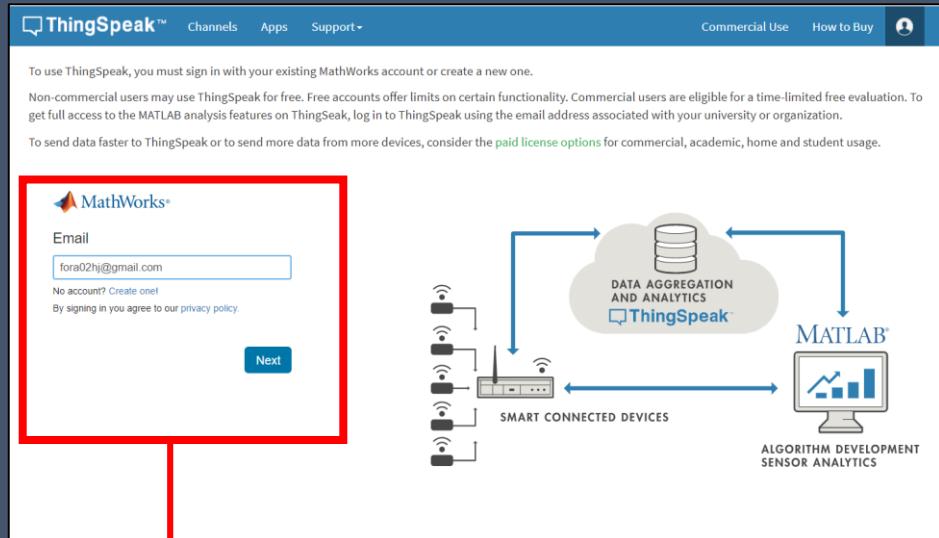
The screenshot shows the 'ThingSpeak Usage Intent' survey. Question 1 asks 'How are you planning to use ThingSpeak?' with options for commercial work, government work, personal projects, teaching/research in school, and student use. 'Teaching or research in school' is selected. Question 2 asks 'Tell us something about your project (optional)' with a text input field containing 'smart farm'. At the bottom right is a green 'OK' button.

ThingSpeak Login

1. Click the top-right person icon

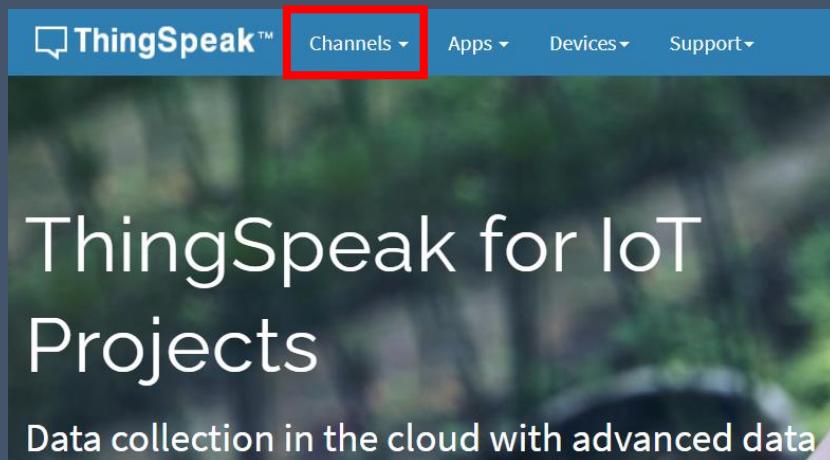


2. Enter your login Email/Password

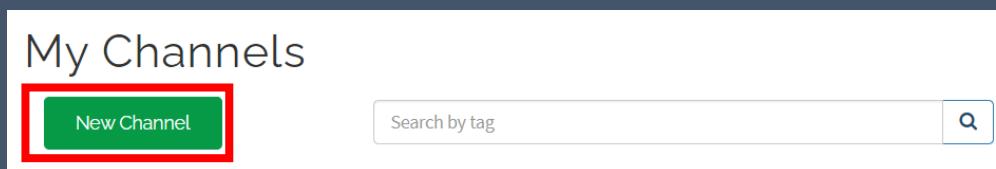


Creating and setting up a ThingSpeak channel

1. Click 'channels' in the top menu



2. 새로운 채널 생성



New Channel

Help

Channels store all the data that a ThingSpeak app collects. A channel has eight fields that can hold any type of data, plus location and status data. Once you collect data in a channel, you can visualize it.

Channel Settings

- Percentage complete: Calculated based on the number of fields in the channel. Enter the name, description, location, and tags for your channel.
- Channel Name: Enter a unique name for your channel.
- Description: Enter a description of the ThingSpeak channel.
- Field#: Check the box to enable the field, up to 8 fields per channel. A channel can have up to 8 fields.
- Metadata: Enter information about channel.
- Tags: Enter keywords that identify the channel.
- Link to External Site: If you have a website associated with your ThingSpeak channel, specify the URL.
- Show Channel Location:
 - Latitude: Specify the latitude position of the channel. The latitude of the city of London is 51.5074.
 - Longitude: Specify the longitude position of the channel. The longitude of the city of London is 0.1278.

Name: LTE Cat.M1 Sensor Test

Description:

Field 1: Temperature

Field 2: Humidity

Field 3:

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

Metadata:

Enter as above and click 'Save Channel'

Save Channel

Creating and setting up a ThingSpeak channel

The screenshot shows two parts of the ThingSpeak interface. The top part is titled 'Channel Sharing Settings' and includes tabs for 'Private View', 'Public View', 'Channel Settings', 'Sharing' (which is highlighted with a red box), 'API Keys', and 'Data Import / Export'. A red arrow points from the text above this section to the 'Sharing' tab. The bottom part is titled 'Write API Key' and includes tabs for 'Private View', 'Public View', 'Channel Settings', 'Sharing', 'API Keys' (which is highlighted with a red box), and 'Data Import / Export'. A red arrow points from the text below this section to the 'API Keys' tab. Both sections contain descriptive text and help links.

Enable Public View to make your data always available on mobile apps and websites.
Check Keep channel view private if you don't want to share data

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

Channel Sharing Settings

Keep channel view private
 Share channel view with everyone
 Share channel view only with the following users:

Email Address Add User

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

Write API Key

Key Generate New Write API Key

Enable Public View to make your data always available on mobile apps and websites.
Check Keep channel view private if you don't want to share data

To upload sensor data to Thingspeak.com, you need a Write API Key is required to upload sensor data to Thingspeak.com.

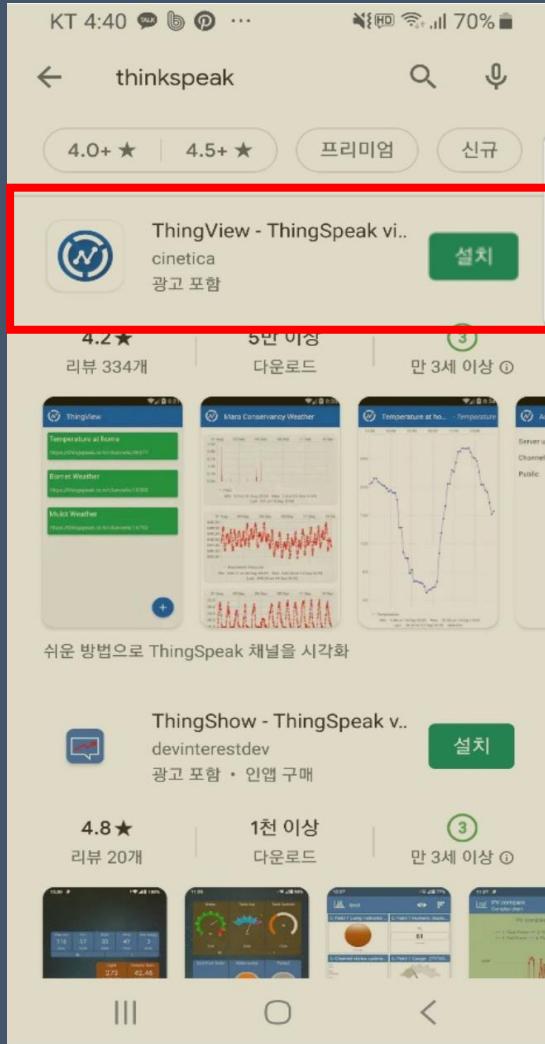
Creating and setting up a ThingSpeak channel

4. Look up Private View

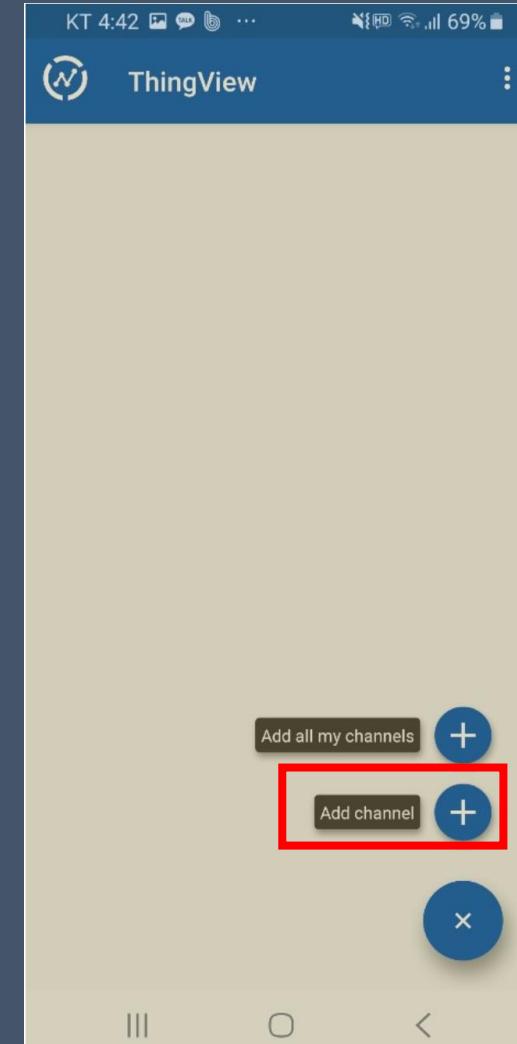
The screenshot shows the ThingSpeak channel interface. At the top, there's a navigation bar with links for 'Channels', 'Apps', 'Devices', 'Support', 'Commercial Use', 'How to Buy', and a user icon. Below the navigation bar are several buttons: 'Add Visualizations', 'Add Widgets', 'Export recent data', 'MATLAB Analysis' (which is highlighted in green), and 'MATLAB Visualization'. A status message 'Channel 4 of 4 < >' is also present. The main area is titled 'Channel Stats' and shows 'Created: 14.minutes.ago' and 'Entries: 0'. There are two chart sections: 'Field 1 Chart' and 'Field 2 Chart', both titled 'LTE Cat.M1 Sensor Test'. The 'Field 1 Chart' has 'Temperature' on the y-axis and 'Date' on the x-axis. The 'Field 2 Chart' has 'Humidity' on the y-axis and 'Date' on the x-axis. Both charts are currently empty.

ThingSpeak channel lookup (app)

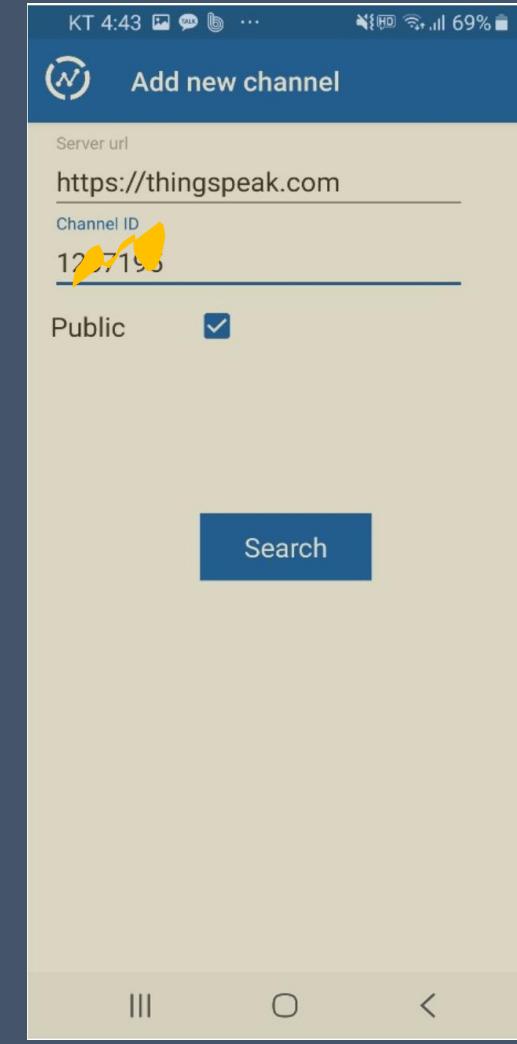
1. Download and install apps



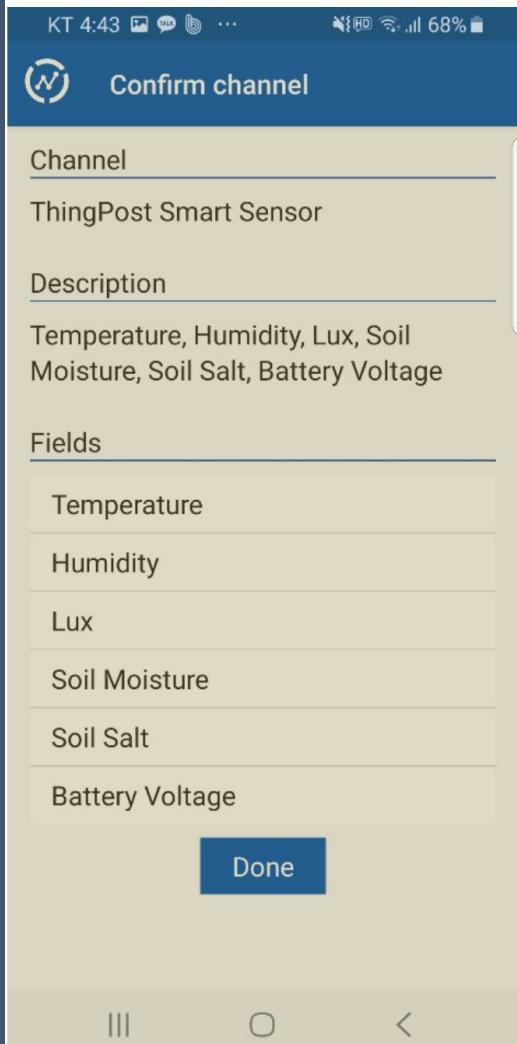
2. Setting up channels



3. Enter a channel ID (yours!!)



4. Check your channel

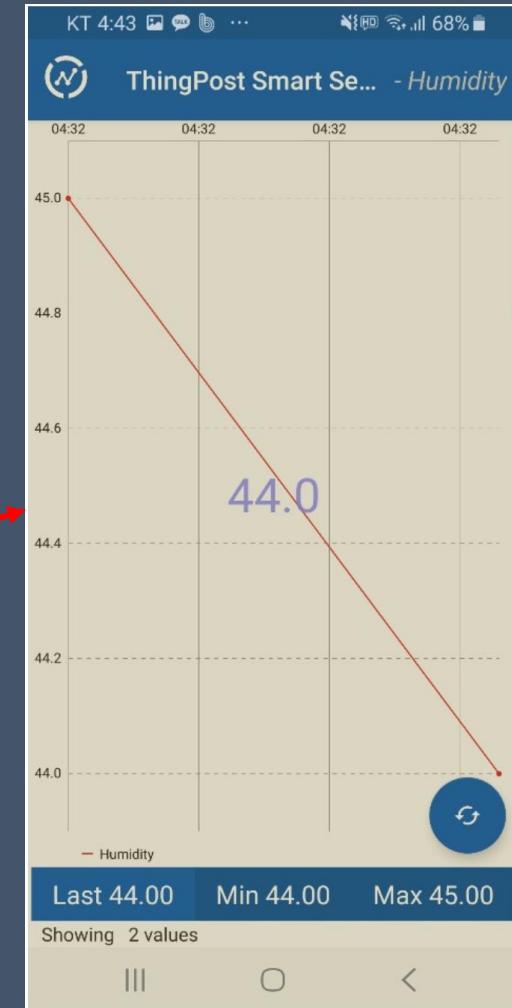
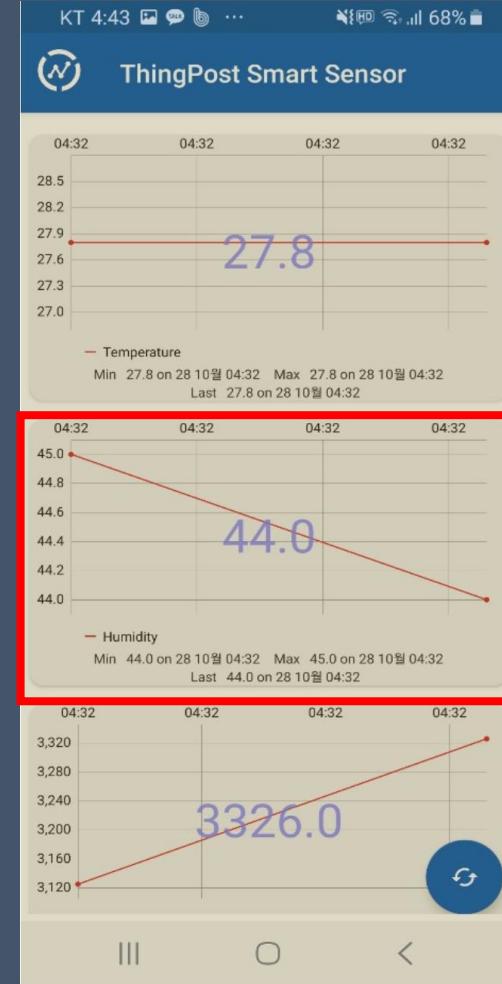


ThingSpeak 채널 조회 (앱)

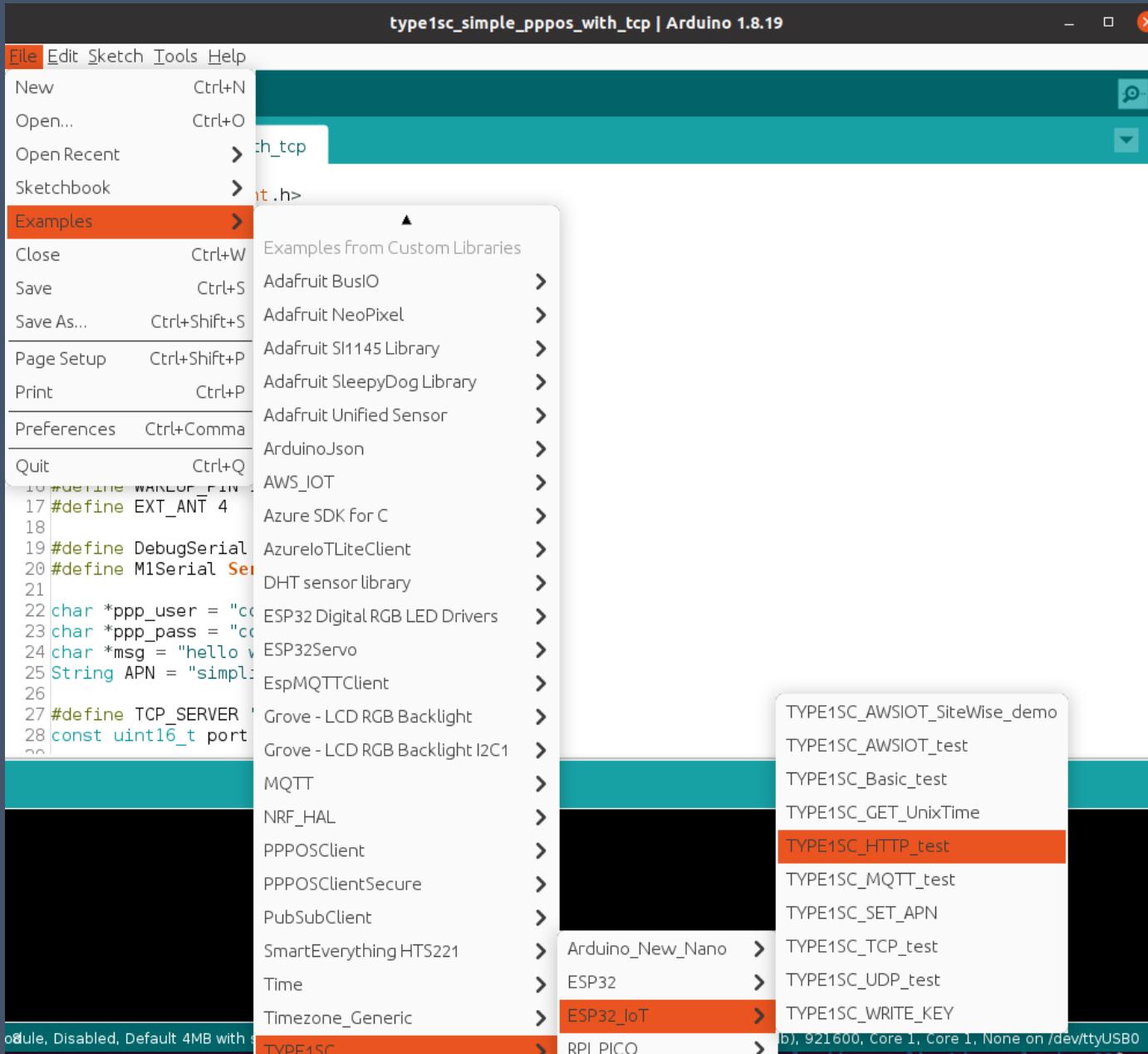
5. Select a channel



6. Channel lookup



5) HTTP Test



✓ Examples -> TYPE1SC-main -> ESP32_IoT ->
Select TYPE1SC_HTTP_test

5) HTTP Test

```
102     String WApiKey = "*****"; // Thing Speak Write API Key 16Character
103     float temp = 0.0;
104     float humi = 0.0;
105     String fieldTemp = "field1"; // Air temperature
106     String fieldHumi = "field2"; // Air humidity
107
108     /*Get Temperature & Humidity */
109     while (1) {
110         /* Get DHT22 Sensor */
111         temp = dht.readTemperature();
112         humi = dht.readHumidity();
113         if (String(temp) != "nan" && String(humi) != "nan")
114             break;
115         else {
116             DebugSerial.println("case nan...");
117             u8x8log.print("case nan...\n");
118             delay(1000);
119         }
120     }
```

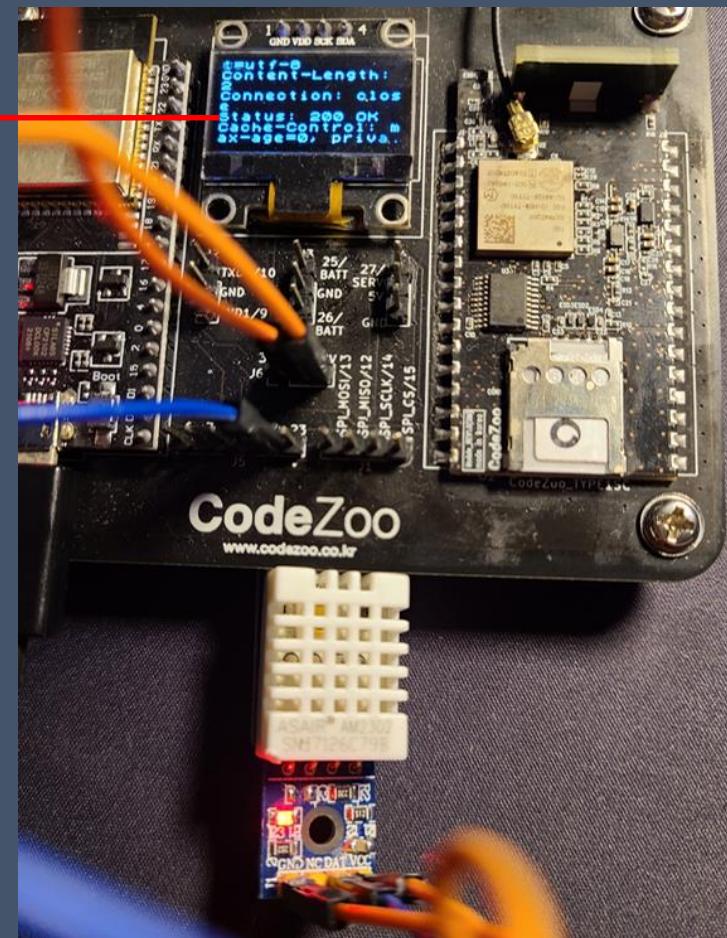
Erase ***** and enter the Write Key you created in the Thingspeak.com channel in the Thingspeak.com channel

5) HTTP Test

```
COM9
04:28:19.700 -> TCP Socket Create...
04:28:19.277 -> TCP Socket Activation!!!
04:28:19.325 -> Socket Info : ACTIVATED
04:28:19.315 -> [HTTP Send] >> GET /update?api_key=XXXXXXXXXXXXXX&field1=26.90&field2=90.90 HTTP/1.1
04:28:19.315 -> Host: api.thingspeak.com
04:28:19.315 -> Connection: close
04:28:19.320 ->
04:28:19.325 ->
04:28:22.414 -> [Recv] >> HTTP/1.1 200 OK
04:28:22.444 -> Date: Tue, 31 Aug 2021 19:28:20 GMT
04:28:22.444 -> Content-Type: text/plain; charset=utf-8
04:28:22.444 -> Content-Length: 1
04:28:22.444 -> Connection: close
04:28:22.444 -> Status: 200 OK
04:28:22.444 -> X-Frame-Options: SAMEORIGIN
04:28:22.491 -> Access-Control-Allow-Origin: *
04:28:22.491 -> Access-Control-Allow-Methods: GET, POST, PUT, OPTIONS, DELETE, PATCH
04:28:22.491 -> Access-Control-Allow-Headers: origin, content-type, X-Requested-With
04:28:22.491 -> Access-Control-Max-Age: 1800
04:28:22.491 -> ETag: W/"6b86b273ff34fce19d6b804eff5a3f57"
04:28:22.491 -> Cache-Control: max-age=0, private, must-revalidate
04:28:22.491 -> X-Request-Id: f97d7f48-a97c-4770-88b4-c24a39285f31
04:28:22.491 -> X-Runtime: 0.038395
04:28:22.491 -> X-Powered-By: Phusion Passenger 4.0.57
04:28:22.491 -> Server: nginx/1.9.3 + Phusion Passenger 4.0.57
04:28:22.491 ->
04:28:22.491 -> 1
04:28:22.491 -> [RecvSize] >> 640
04:28:22.491 -> TCP Socket DeActivation!!!
04:28:22.539 -> Socket Info : DEACTIVATED
04:28:22.539 -> TCP Socket Close!!!
```

Upload temperature and humidity data collected by the DHT22 sensor to Thingspeak.com over HTTP

Return 200 OK if data is successfully sent



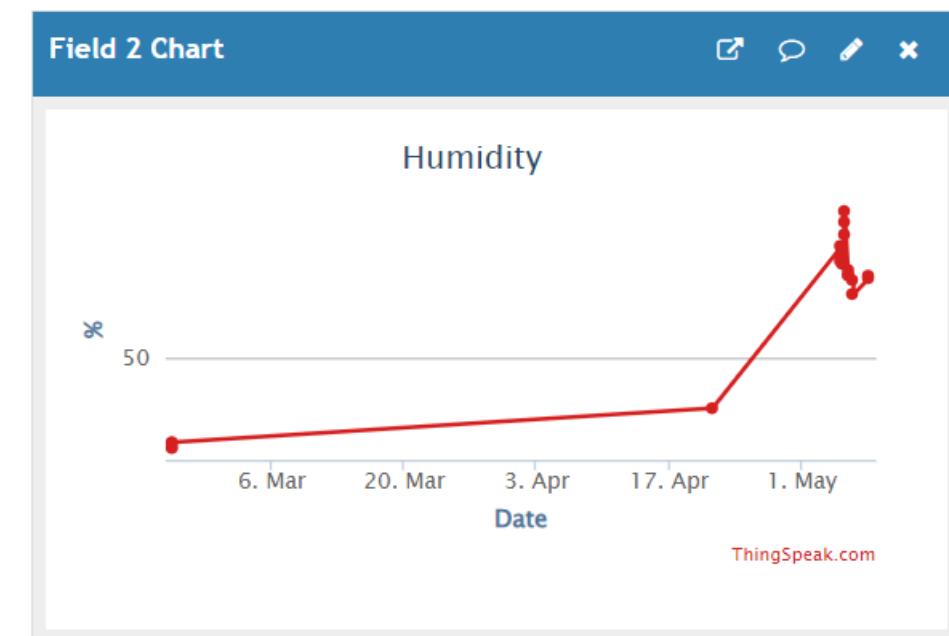
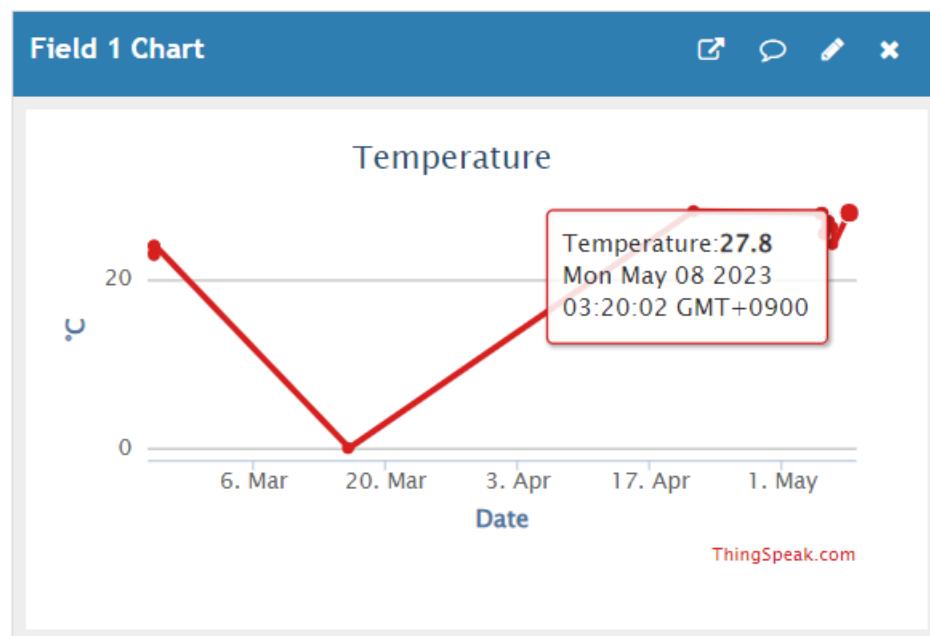
5) HTTP Test

Channel Stats

Created: 11 months ago

Last entry: 2 minutes ago

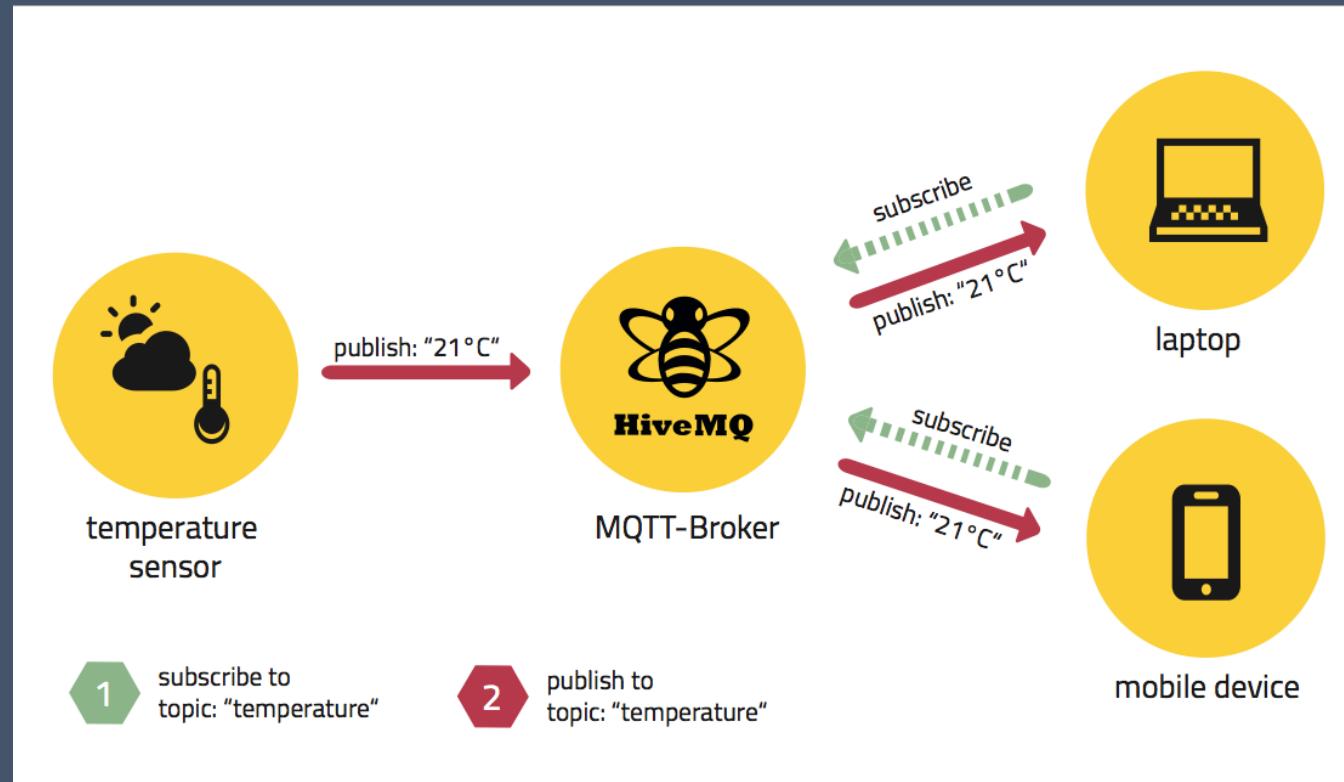
Entries: 20



6) MQTT Test

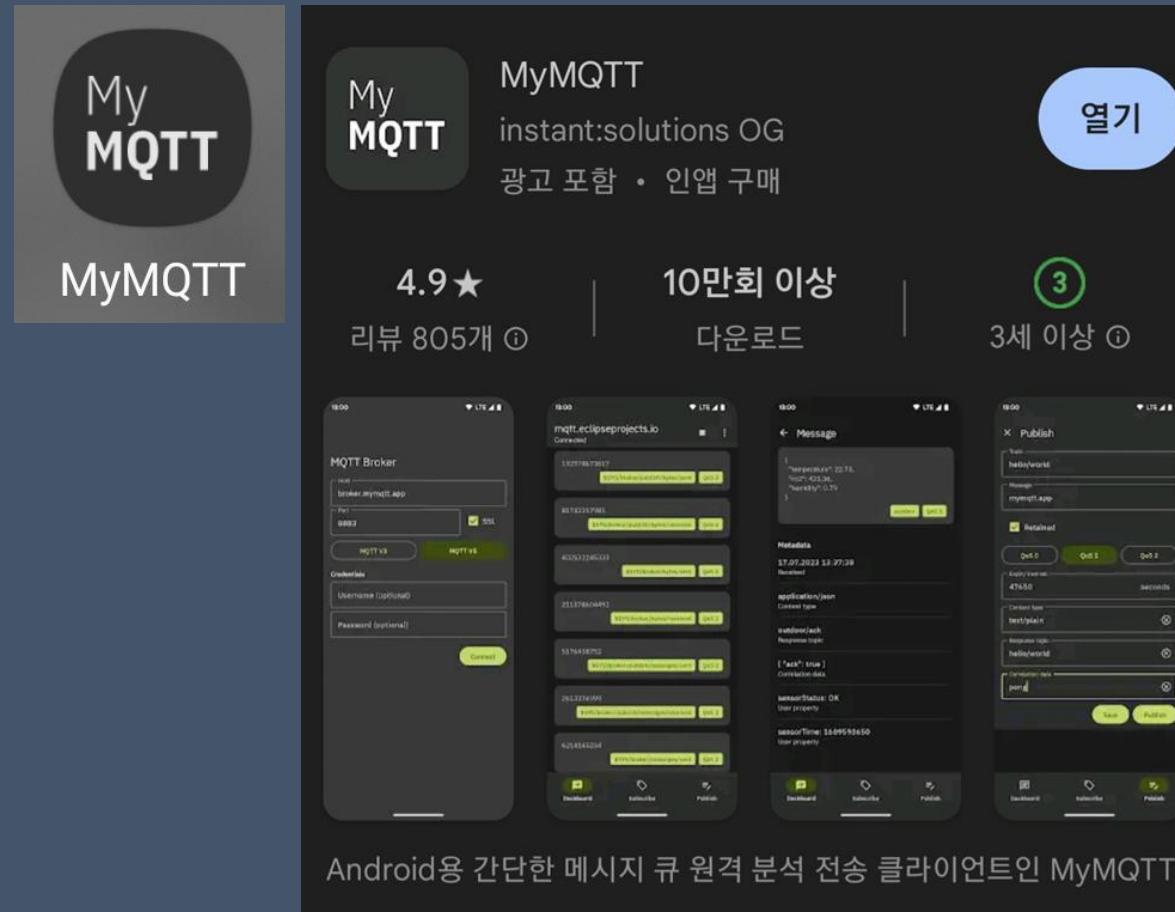
Using the CAT.M1 modem's internal MQTT communication stack, you can publish the temperature and humidity data measured by the Arduino board to HiveMQ and monitor it in real time on your PC. temperature and humidity data measured by the Arduino board to HiveMQ and monitor it in real time on the PC.

1. Set up to monitor MQTT data on PC.
2. Connect to HiveMQ Broker with CAT.M1 modem's internal MQTT communication stack and send temperature and humidity data from Arduino board to DHT22 sensor. temperature and humidity data from the Arduino board and publish the data.



Install the Android MQTT App

- Install MyMQTT from the Android MQTT app.
- Search for MyMQTT in the PlayStore and install it.

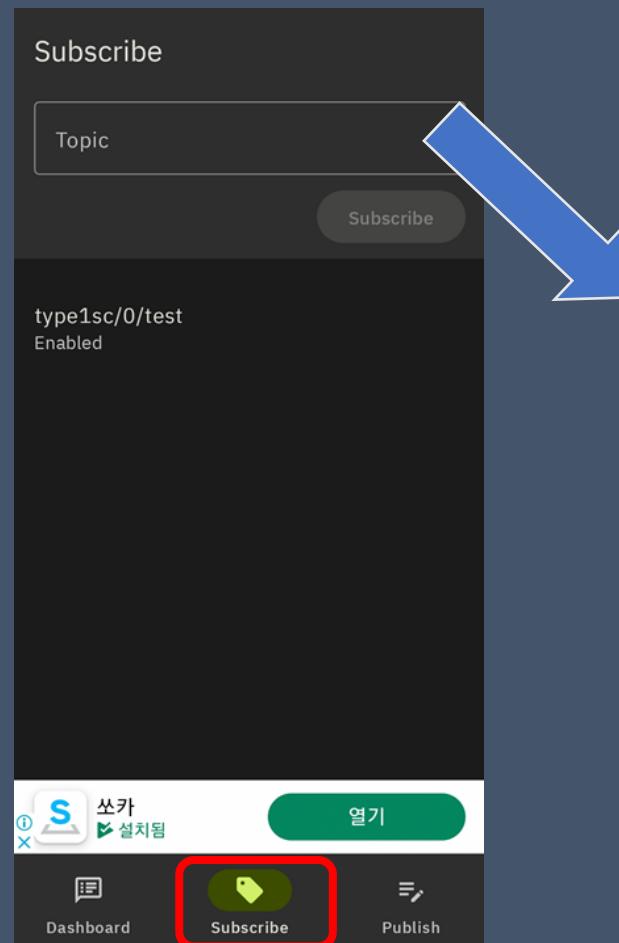
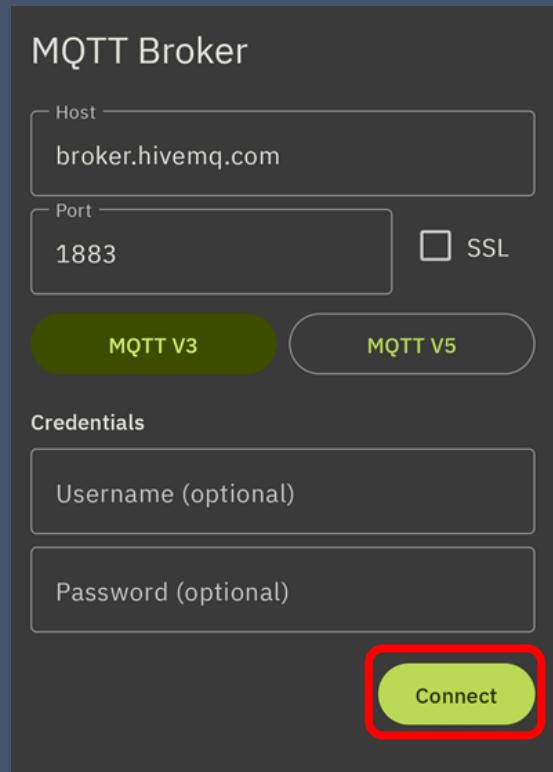


Setting up MyMQTT

1) Host : broker.hivemq.com

Port : 1883

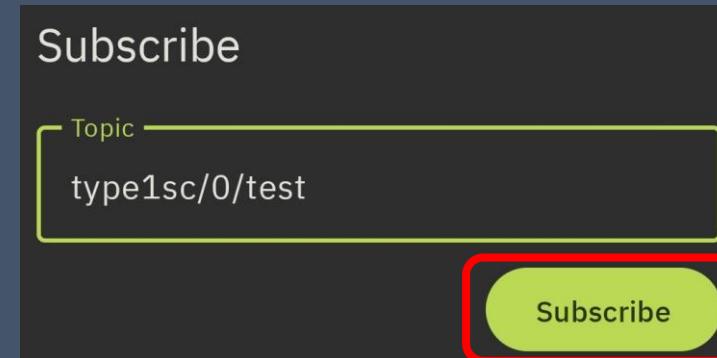
2) Select Connect



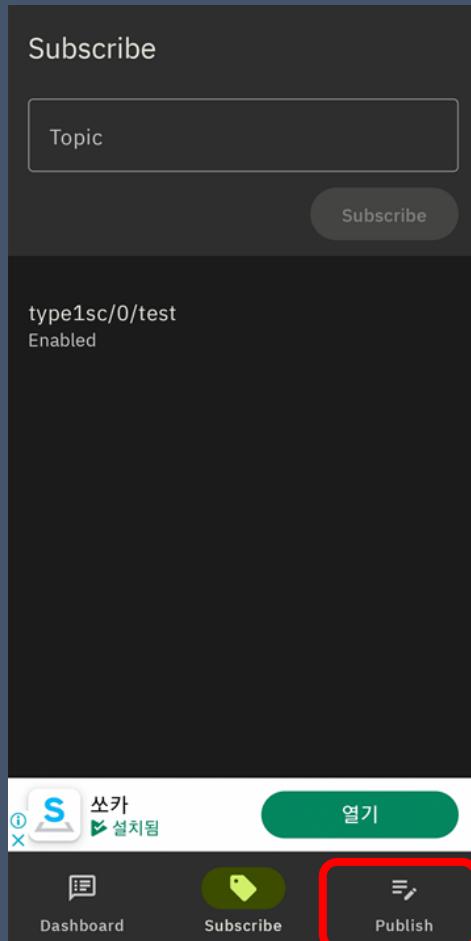
3) Move the Subscribe tab

4) Enter a Topic
type1sc/0/test

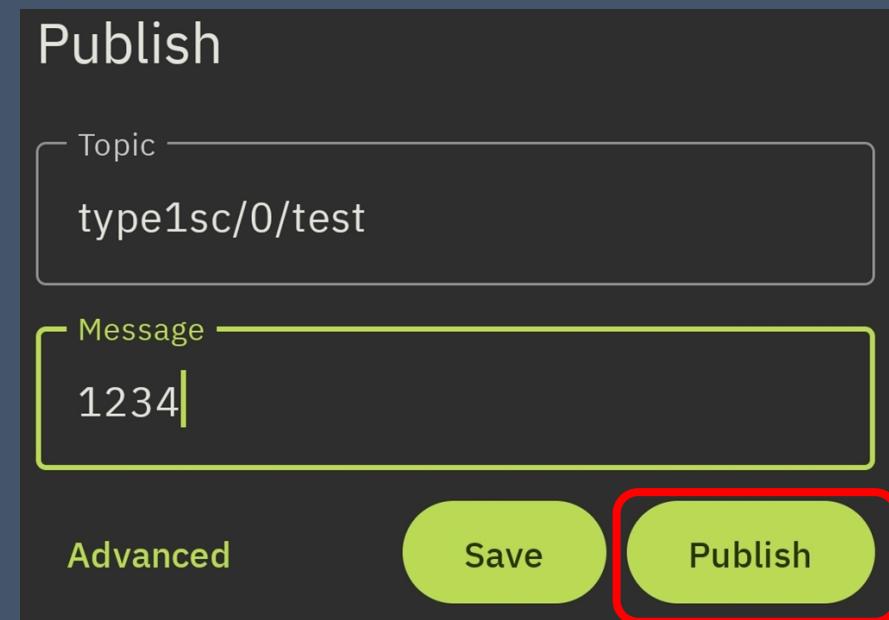
5) Select the Subscribe button under the Topic



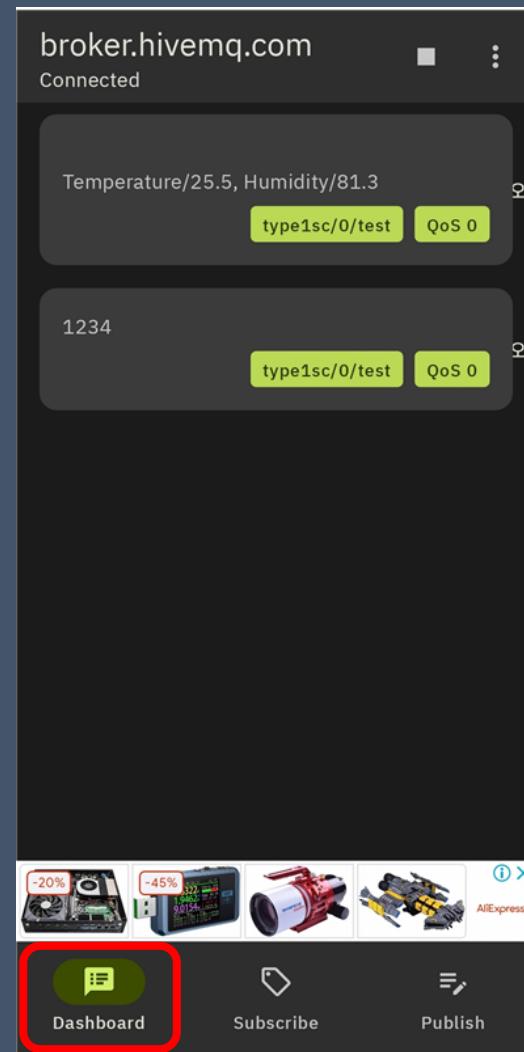
Setting up MyMQTT



6) Go to the Publish tab



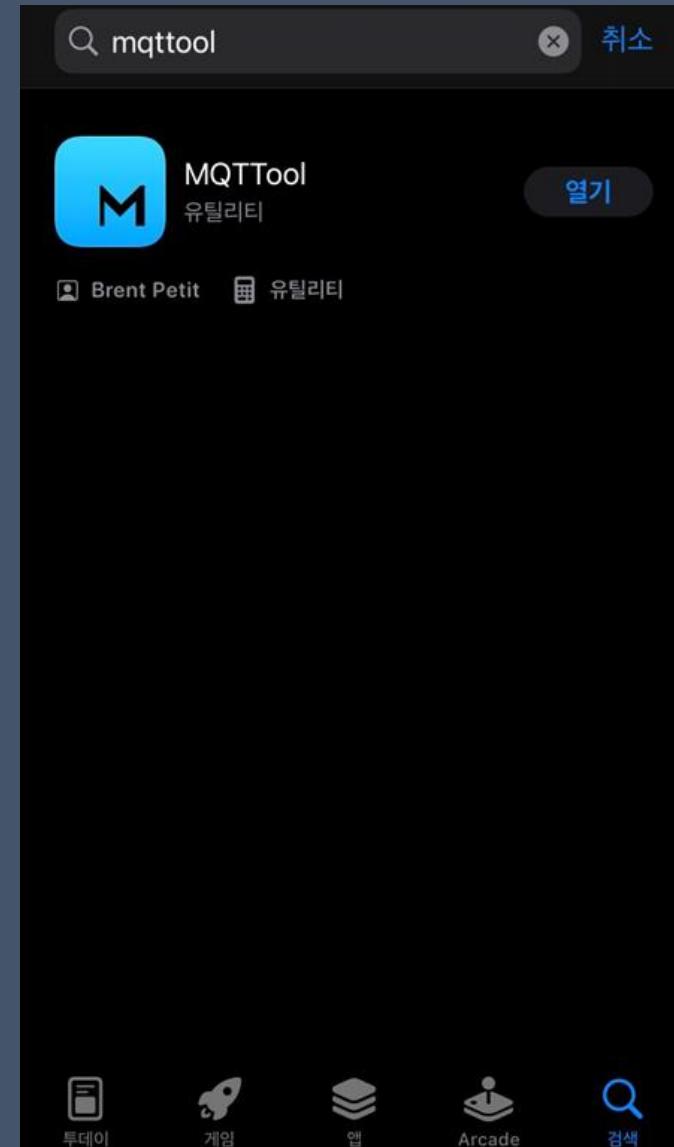
7) In the Publish window, enter a Topic and click Message and select the Publish button.



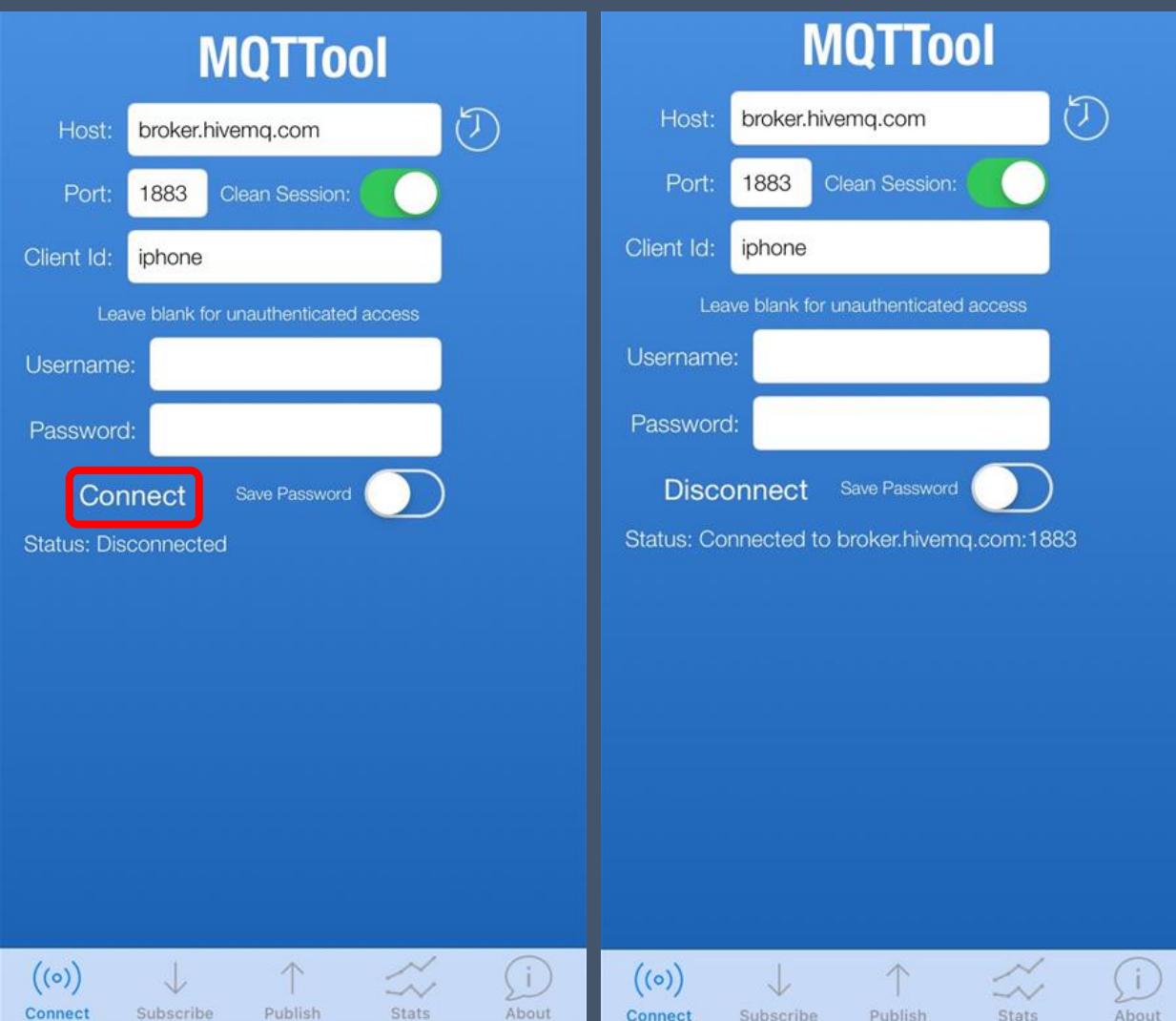
8) Dashboard tab and make sure the message is coming through messages are coming through.

Install the iPhone MQTT app

- Install MQTTTool from the iPhone MQTT app.
- Search for mqttool in the AppStore and install it.



Setting up MQTTTool



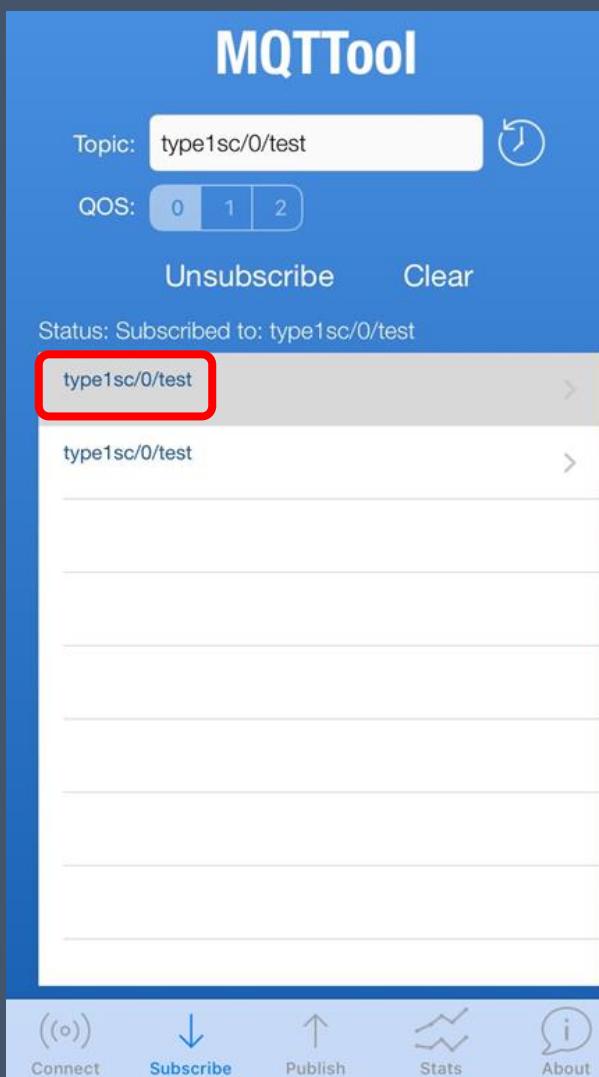
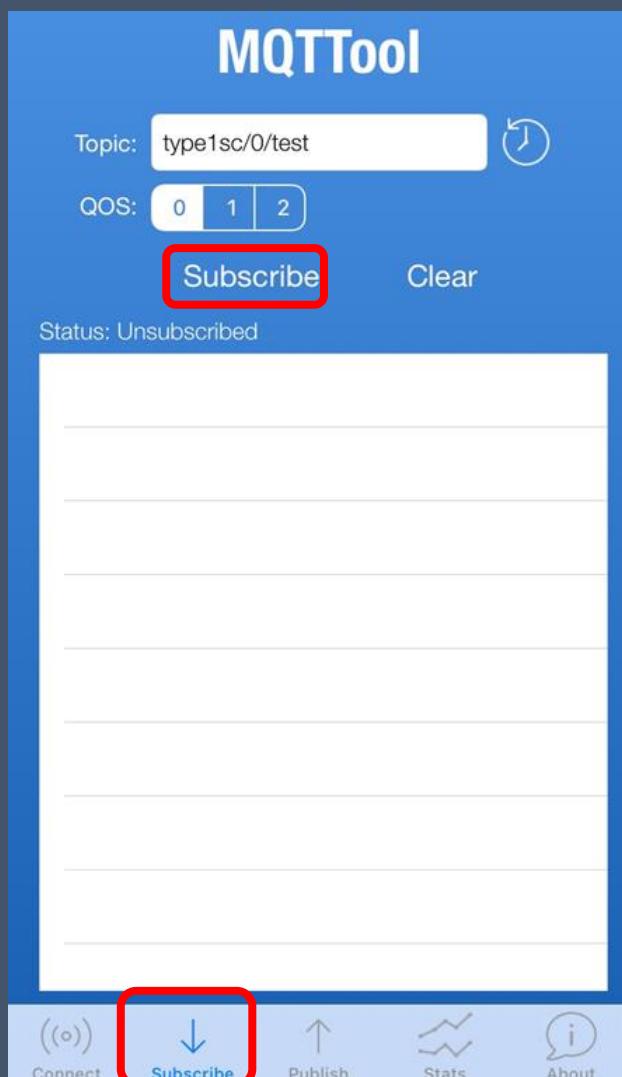
1) Host : broker.hivemq.com

Port : 1883

Client Id : iphone

2) Select Connect

Setting up MQTTTool



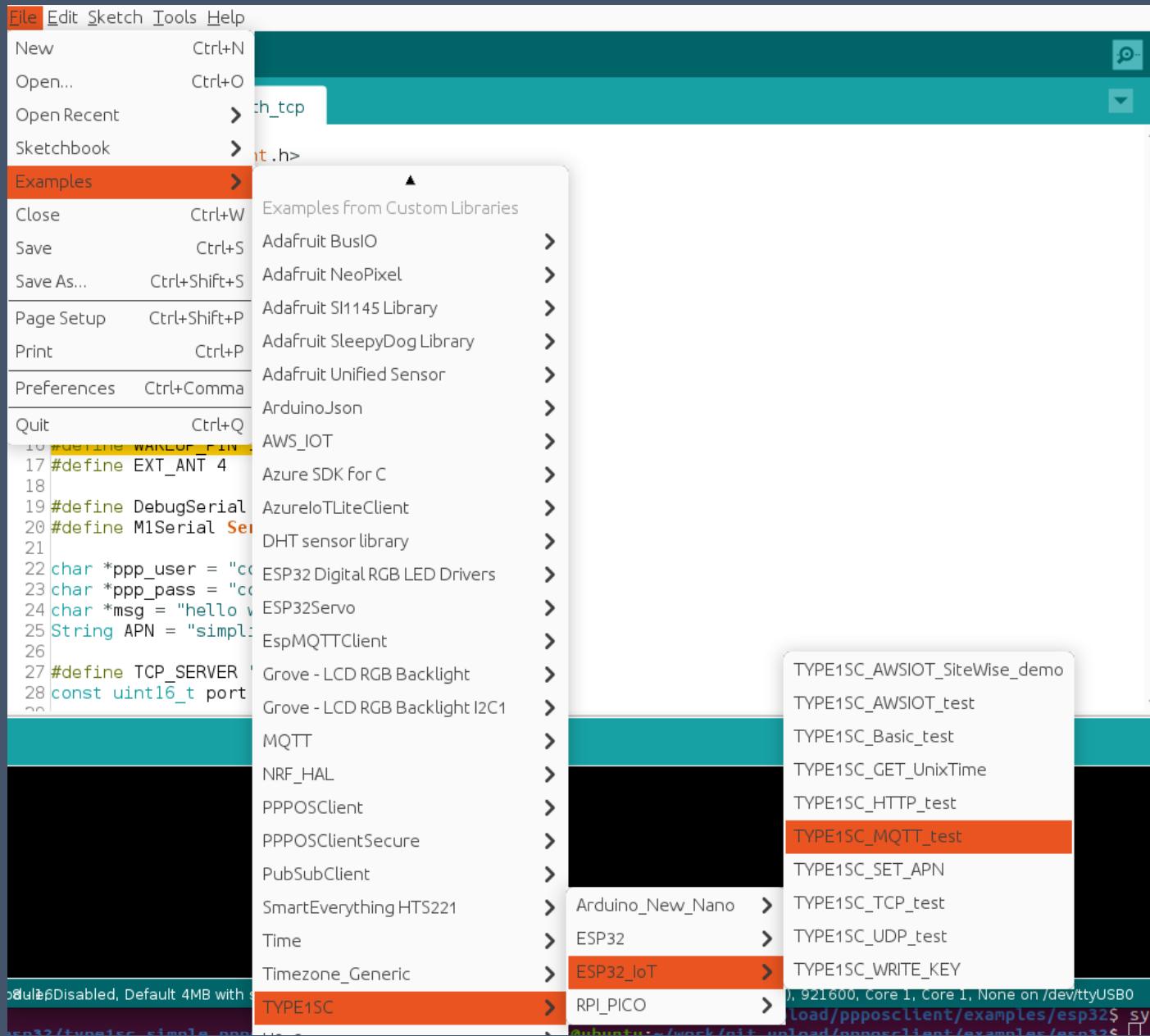
3) Move the Subscribe tab

4) Topic : type1sc/0/test

5) Select the Subscribe button

6) Subscribe to the
type1sc/0/test topic Confirm
message received

6) MQTT Test



✓ Examples -> TYPE1SC-main -> ESP32_IoT ->
Select TYPE1SC_MQTT_test

6) MQTT Test

broker.hivemq.com
Connected

Temperature/26.2, Humidity/74.0
type1sc/0/test QoS 0

Temperature/26.0, Humidity/74.7
type1sc/0/test QoS 0

Temperature/26.0, Humidity/74.1
type1sc/0/test QoS 0

Temperature/25.7, Humidity/78.0
type1sc/0/test QoS 0

-20% -45% 322 0.154 AllExpress

Dashboard Subscribe Publish

MQTTTool

Topic: type1sc/0/test

QoS: 0 1 2

Unsubscribe Clear

Status: Subscribed to: type1sc/0/test

- type1sc/0/test >
- type1sc/0/test >

Connect Subscribe Publish Stats About

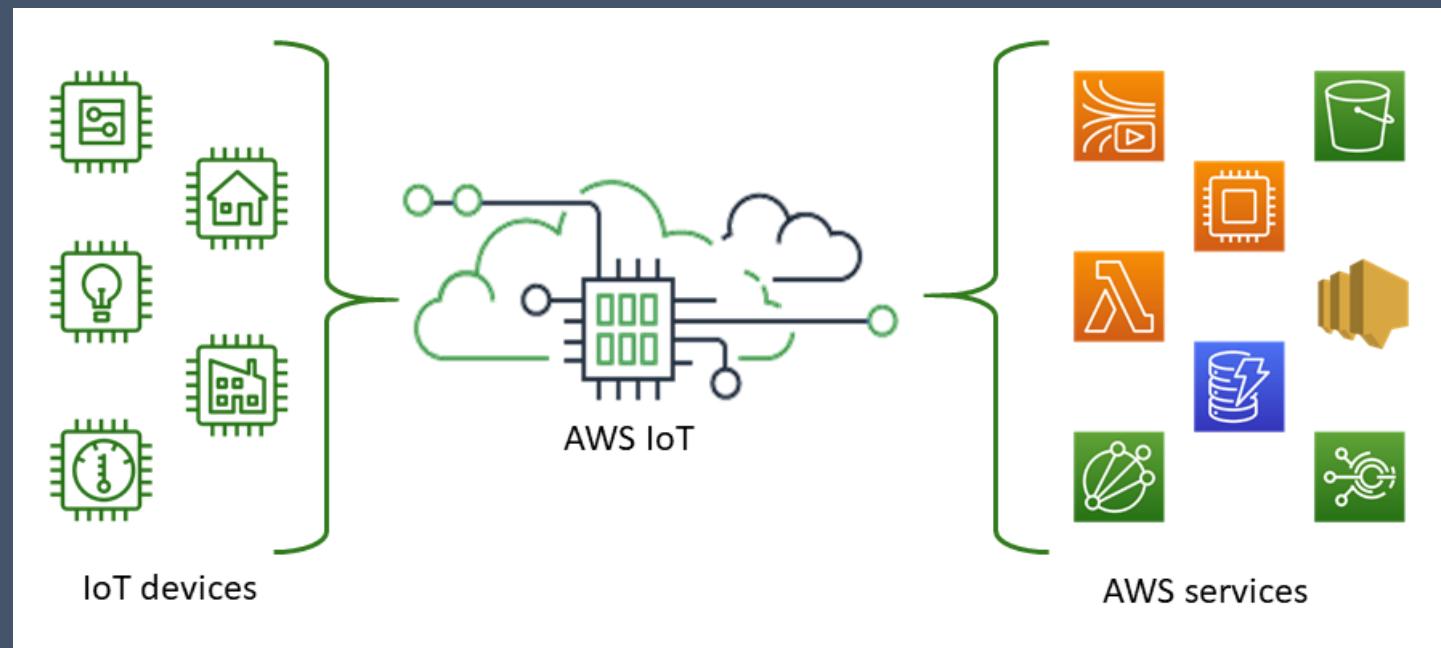
```
05:21:08.304 -> TYPE1SC
05:21:10.371 -> Network
05:21:13.184 -> Network
05:21:15.196 -> Network
05:21:17.165 -> TYPE1SC
05:21:17.165 -> 1.Enable
05:21:17.213 -> 2.Config
05:21:17.213 -> 3.Configure node parameter.Timeout
05:21:19.603 -> 4.Establish connection
05:21:22.040 -> 5.Subscribe to the topic on the endpoint
05:21:24.570 -> 6.Publish data to broker
05:21:26.539 -> 7.UnSubscribe to the topic on the endpoint
05:21:28.558 -> 8.Disconnect MQTT Server
05:21:28.558 -> 9.Disable MQTT events
```



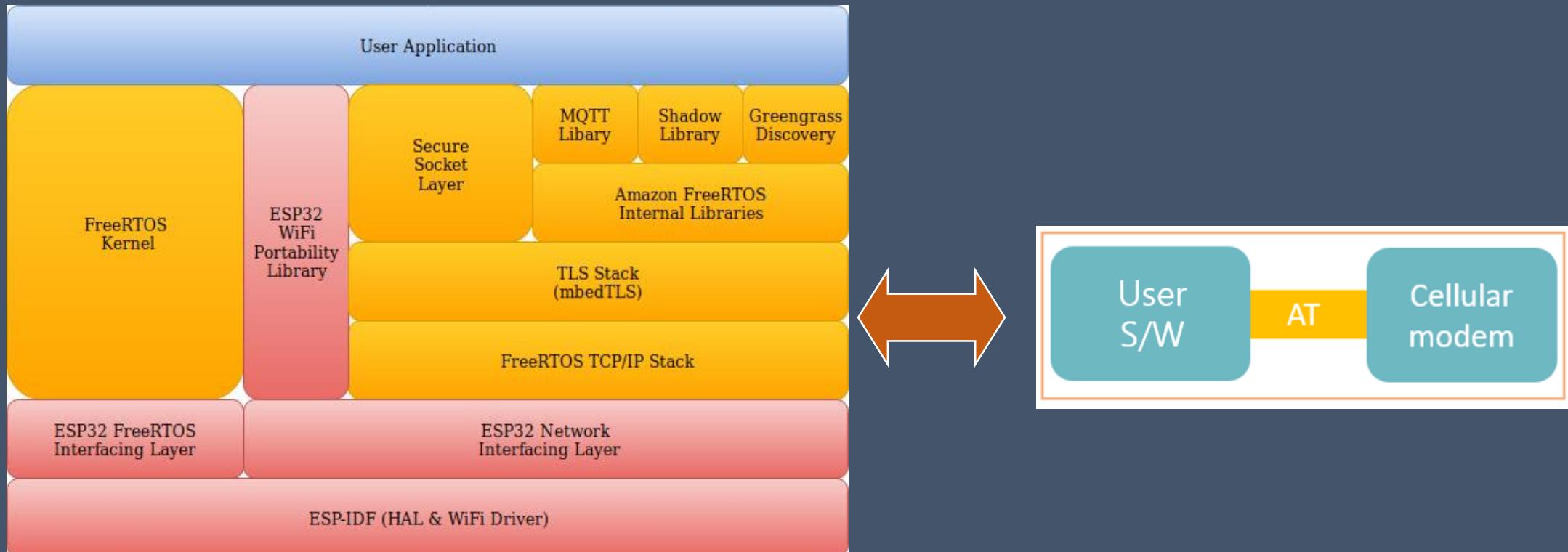
7) AWS-IoT Setup

The CAT.M1 modem uses the AWS-IoT communication stack inside the CAT.M1 modem to send temperature and humidity data measured by the Arduino board to the AWS Cloud using the AWS-IoT communication stack.

1. It is much easier and simpler to use AWS-IoT than dealing with the MCU directly.
2. All the necessary software stacks are built inside the Cat.M1 modem, so you can easily use it by calling commands. with a command line interface.



7) AWS-IoT Setup



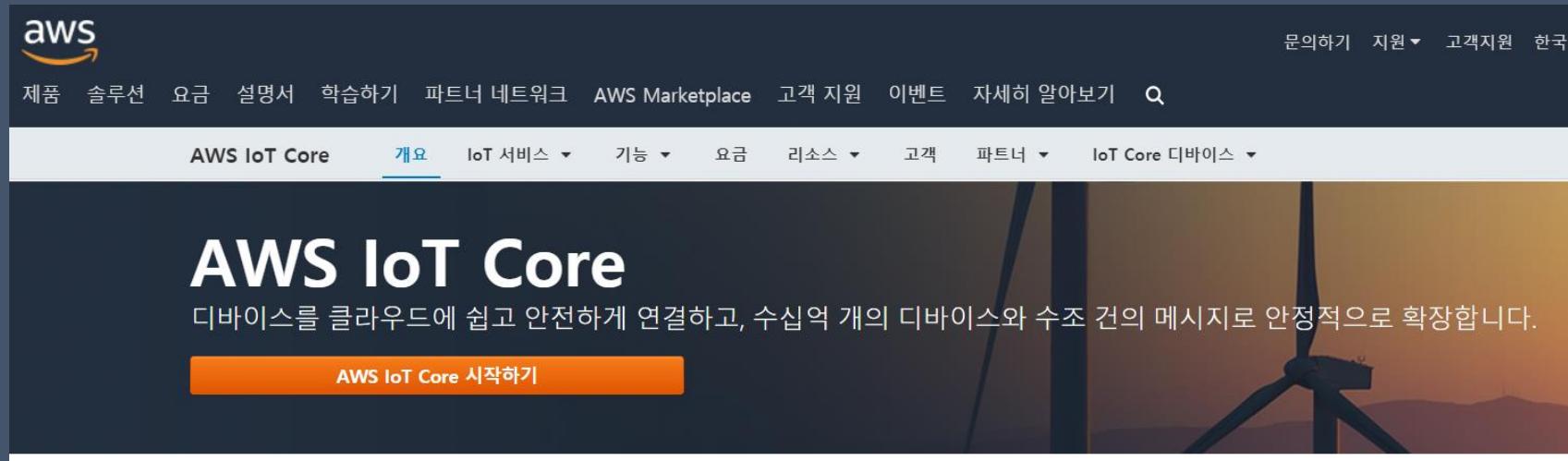
ESP32 AWS-IoT
S/W Architecture



7) AWS-IoT Setup

Create an AWS account

<https://aws.amazon.com/iot-core>



7) AWS-IoT Setup

Access IoT Core

The screenshot shows the AWS Management Console interface. At the top, there is a search bar with the placeholder "서비스, 기능, 마켓플레이스 제품, 설명서 검색" and a keyboard shortcut "[Alt+S]". To the right of the search bar are user information ("codezoo_jang") and location ("서울"). On the far left, there's a sidebar with sections for "즐겨찾기" (Favorites) containing "Resource Groups & Tag Ed..." and "최근 방문" (Recent Visits) listing "IoT Core", "콘솔 홈", and "IoT Analytics". The main content area is titled "모든 서비스" (All Services). The services are organized into several groups: "보안, 자격 증명 및 규정 준수" (Security, Identity, and Compliance), "네트워킹 및 콘텐츠 전송" (Networking and Content Delivery), "미디어 서비스" (Media Services), and "사물인터넷" (IoT). The "IoT Core" service is highlighted with a yellow box. Other visible services include Service Catalog, MSK, AppStream 2.0, WorkLink, AWS Glue DataBrew, Amazon FinSpace, Trusted Advisor, AWS AppConfig, AWS Transfer Family, AWS Snow Family, AWS License Manager, AWS Well-Architected Tool, Personal Health Dashboard, AWS Chatbot, Launch Wizard, AWS Compute Optimizer, AWS Proton, Incident Manager, CloudFront, Route 53, API Gateway, Direct Connect, AWS App Mesh, AWS Cloud Map, Global Accelerator, Cognito, Secrets Manager, GuardDuty, Inspector, Amazon Macie, Amazon Grafana, Amazon Prometheus, AWS Proton, Incident Manager, Key Management Service, CloudHSM, Directory Service, WAF & Shield, Kinesis Video Streams, AWS Firewall Manager, and Amazon GameLift.

7) AWS-IoT Setup

Create a policy

Create

생성

The screenshot shows the AWS IoT console with the 'Policy' section selected. A red box highlights the '정책' (Policy) link in the left sidebar. Another red box highlights the 'Create' button (생성) in the top right corner of the main content area. The interface includes a search bar labeled '정책 검색' and a magnifying glass icon.

7) AWS-IoT Setup

Create a policy2

정책 생성

정책을 생성하여 권한이 있는 작업 세트를 정의합니다. 하나 이상의 리소스(사물, 주제, 주제 필터)에 대한 작업 권한을 부여할 수 있습니다. IoT 정책에 대해 자세히 알아보려면 [AWS IoT 정책 설명서 페이지](#)를 참조하십시오.

이름

Arduino_AWS_Policy

설명문 추가

정책 설명문에서는 리소스를 사용하여 수행 가능한 작업의 유형을 정의합니다.

고급 모드

작업

쉼표를 사용하여 작업을 분리하십시오(예: iot:Publish, iot:Subscribe).

리소스 ARN

특정 리소스는 클라이언트 ID ARN, 주제 ARN 또는 주제 필터 ARN을 포함할 수 있습니다.

효과

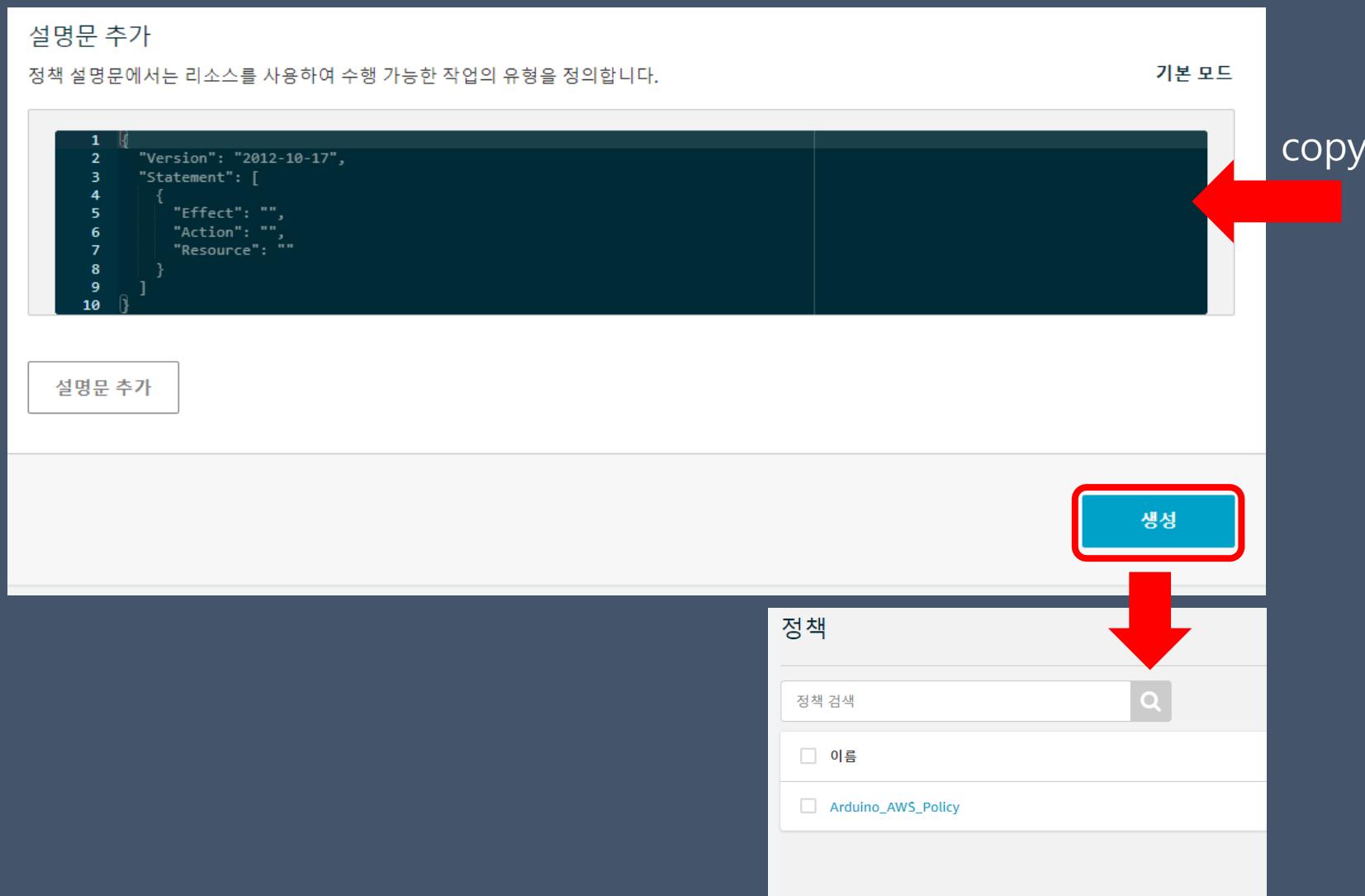
허용 거부

제거

Advanced mode

7) AWS-IoT Setup

Create a policy3 : Enter resources and create



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "*"
    }
  ]
}
```

7) AWS-IoT Setup

Create things

The screenshot shows the AWS IoT Things Management interface. On the left, a sidebar menu is open under the 'AWS IoT' tab, with '관리' (Management) expanded. The '사물' (Things) option is highlighted with a red box. The main content area shows a confirmation message: 'Murata_Node01을(를) 삭제했습니다.' (Deleted Murata_Node01). Below this, the breadcrumb navigation shows 'AWS IoT > 관리 > 사물'. The central part of the screen displays a table for managing things, with a single row for 'Murata_Node01'. A red arrow points from the text 'Create things' to the '사물 생성' (Create Thing) button at the bottom right of the table.

Create things

7) AWS-IoT Setup

Create a Thing2: Create a single thing

사물 생성 정보

사물 리소스란 AWS IoT의 물리적 디바이스 또는 논리적 엔터티를 디지털로 표현한 것입니다. 디바이스 샘플, 이벤트, 작업 및 디바이스 관리 기능과 같은 AWS IoT 기능을 사용하려면 디바이스 또는 엔터티의 레지스트리에 사물 객체가 필요합니다.

등록할 사물 수

단일 사물 생성
사물 리소스를 생성하여 디바이스를 등록합니다. 디바이스가 AWS IoT에 연결하도록 허용하는 데 필요한 인증서 및 정책을 프로비저닝합니다.

많은 사물 생성
여러 사물 리소스를 생성하여 디바이스를 등록하고 해당 디바이스가 AWS IoT에 연결하는 데 필요한 리소스를 프로비저닝하는 작업을 생성합니다.

[취소](#) [다음](#)

7) AWS-IoT Setup

Create Thing3: Thing Name

사물 속성

사물 이름
Murata_Node01

문자, 숫자, 언어 또는 밀줄만 포함하는 고유한 이름을 입력합니다. 사물 이름에는 공백을 포함할 수 없습니다.

추가 구성

이러한 구성은 사용하여 사물을 구성, 관리 및 검색할 수 있는 세부 정보를 추가할 수 있습니다.

- ▶ 사물 유형 - 선택 사항
- ▶ 검색 가능한 사물 속성 - 선택 사항
- ▶ 사물 그룹 - 선택 사항
- ▶ 결제 그룹 - 선택 사항

디바이스 샘플링

디바이스 샘플링을 사용하면 커넥티드 디바이스의 상태를 AWS와 동기화할 수 있습니다. HTTP 또는 MQTT 주제를 사용하여 이 사물의 샘플링에 대한 상태 정보를 가져오거나 업데이트 또는 삭제할 수도 있습니다.

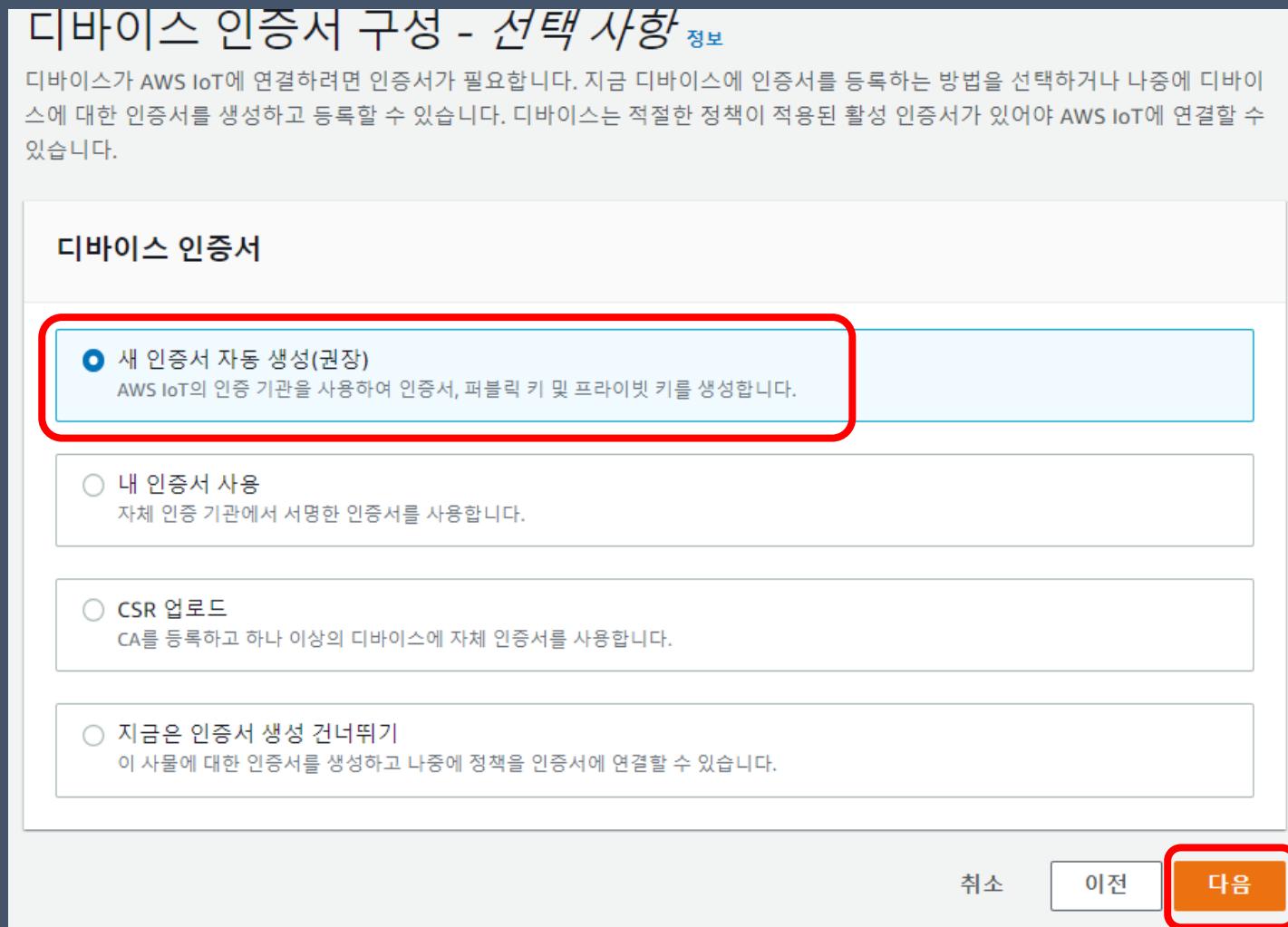
- 샘플링 없음
- 명명된 샘플링
속성에 대한 액세스를 관리하고 디바이스 속성을 논리적으로 그룹화하기 위해 이름이 서로 다른 여러 샘플링을 생성합니다.
- 이름 없는 샘플링(클래식)
사물에는 이름 없는 샘플링이 하나만 있을 수 있습니다.

취소

다음

7) AWS-IoT Setup

Generate things4: Select Automatically generate new certificates (recommended)



7) AWS-IoT Setup

Generate things4: Select Automatically generate new certificates (recommended)

CREATE A THING
Add a certificate for your thing

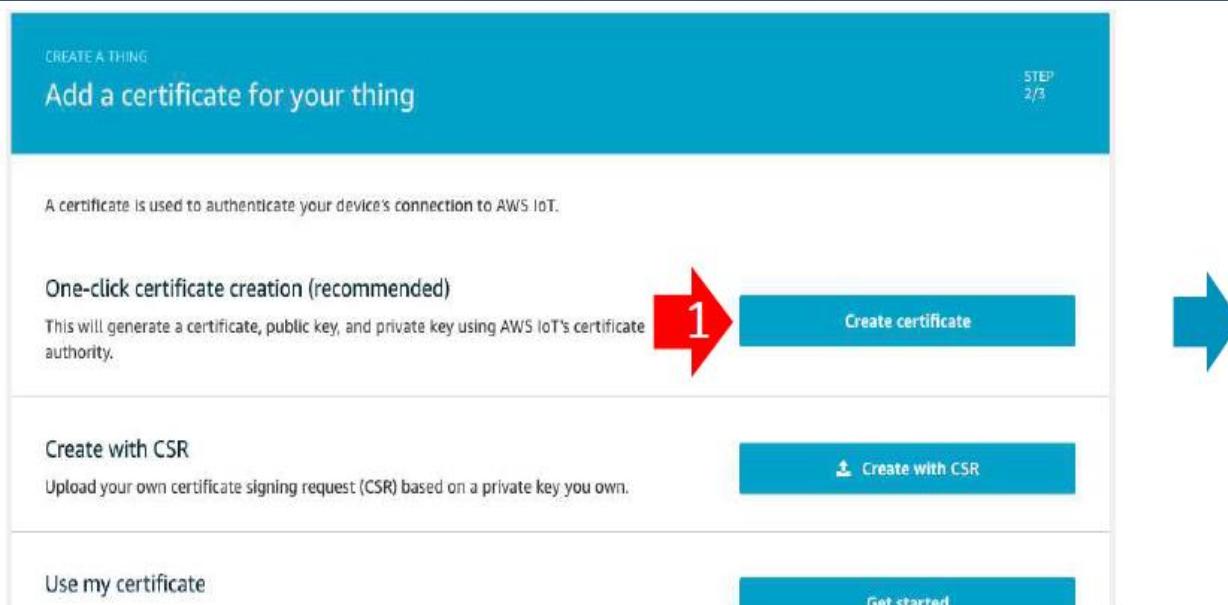
STEP
2/3

A certificate is used to authenticate your device's connection to AWS IoT.

One-click certificate creation (recommended)
This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

Create with CSR
Upload your own certificate signing request (CSR) based on a private key you own.

Use my certificate
Get started



Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	5.cert.pem	2 Download
A public key	5.public.key	Download
A private key	5.private.key	3 Download

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT Download

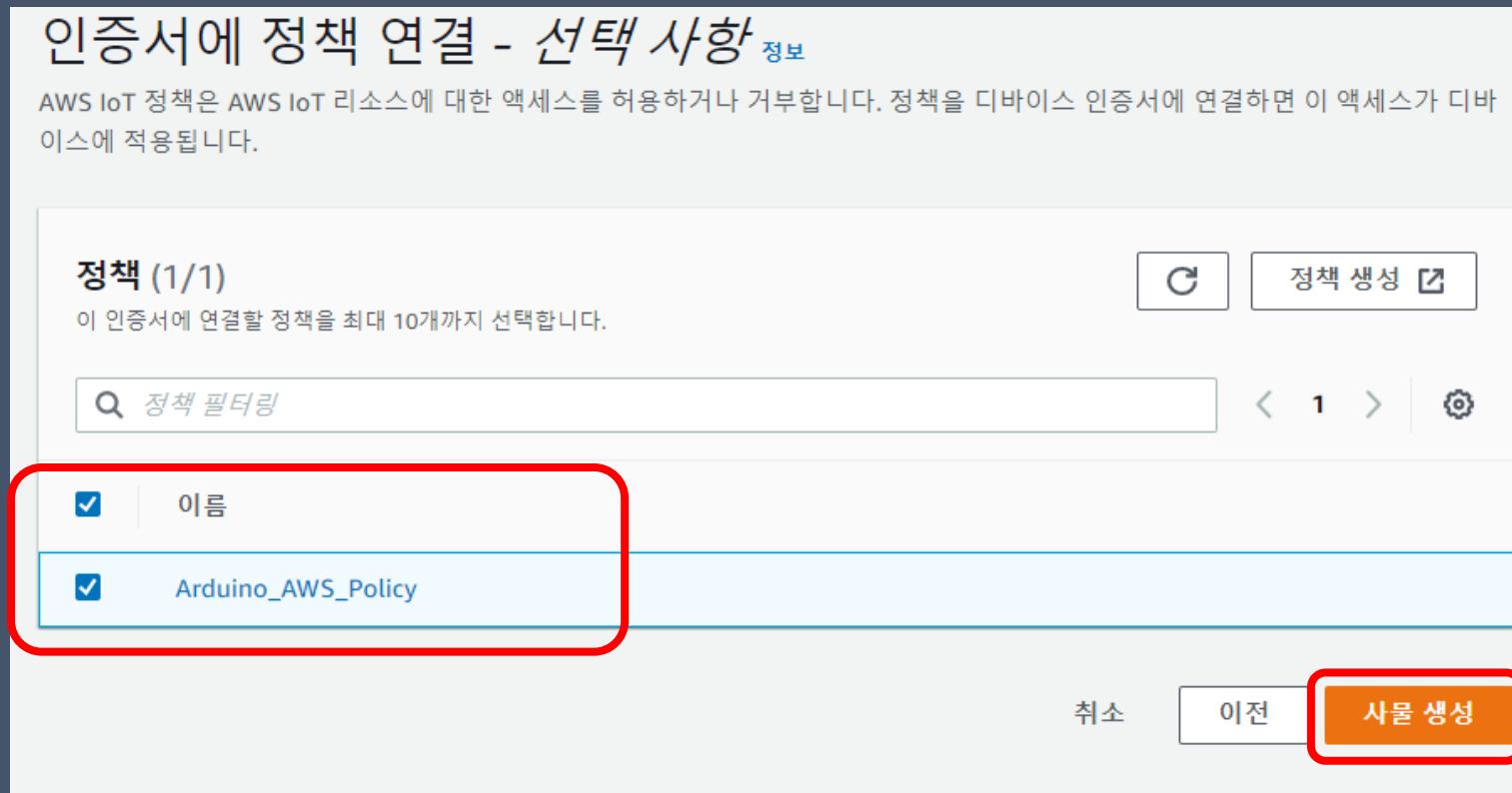
5 Activate

RSA 2048 bit key : Amazon Root CA 1
를 선택후(다운로드) rootCA.pem 로 저장



7) AWS-IoT Setup

Create a Thing5: Link a policy to a certificate



7) AWS-IoT 설정

Check the connection address: Settings -> Device data endpoint -> Copy

The screenshot shows the AWS IoT Settings page. On the left sidebar, under the '설정' (Settings) section, the 'Device data endpoint' link is highlighted with a red box. The main content area displays the 'Device Data Endpoint' information, which includes a URL starting with 'https://XXXXXXXXXX.iot.us-east-2.amazonaws.com'. This URL is also highlighted with a red box.

AWS IoT > 설정

설정 정보

디바이스 데이터 엔드포인트 정보

디바이스는 계정의 디바이스 데이터 엔드포인트를 사용하여 AWS에 연결할 수 있습니다.

각 사물에는 이 엔드포인트에서 사용할 수 있는 REST API가 있습니다. MQTT 클라이언트와 AWS IoT 디바이스 SDK 도 이 엔드포인트를 사용합니다.

엔드포인트
https://XXXXXXXXXX.iot.us-east-2.amazonaws.com

도메인 구성

도메인을 생성하여 디바이스를 AWS IoT Core로 마이그레이션하고, 애플리케이션 인프라를 AWS IoT Core로 마이그레이션하고, 브랜드 자격 증명을 유지하는 등의 태스크를 간소화할 수 있습니다.

작업 ▾ 도메인 구성 생성

작업	도메인 이름	상태	서비스 유형	업데이트된 날짜
도메인 구성 없음	도메인 구성이 없습니다.			

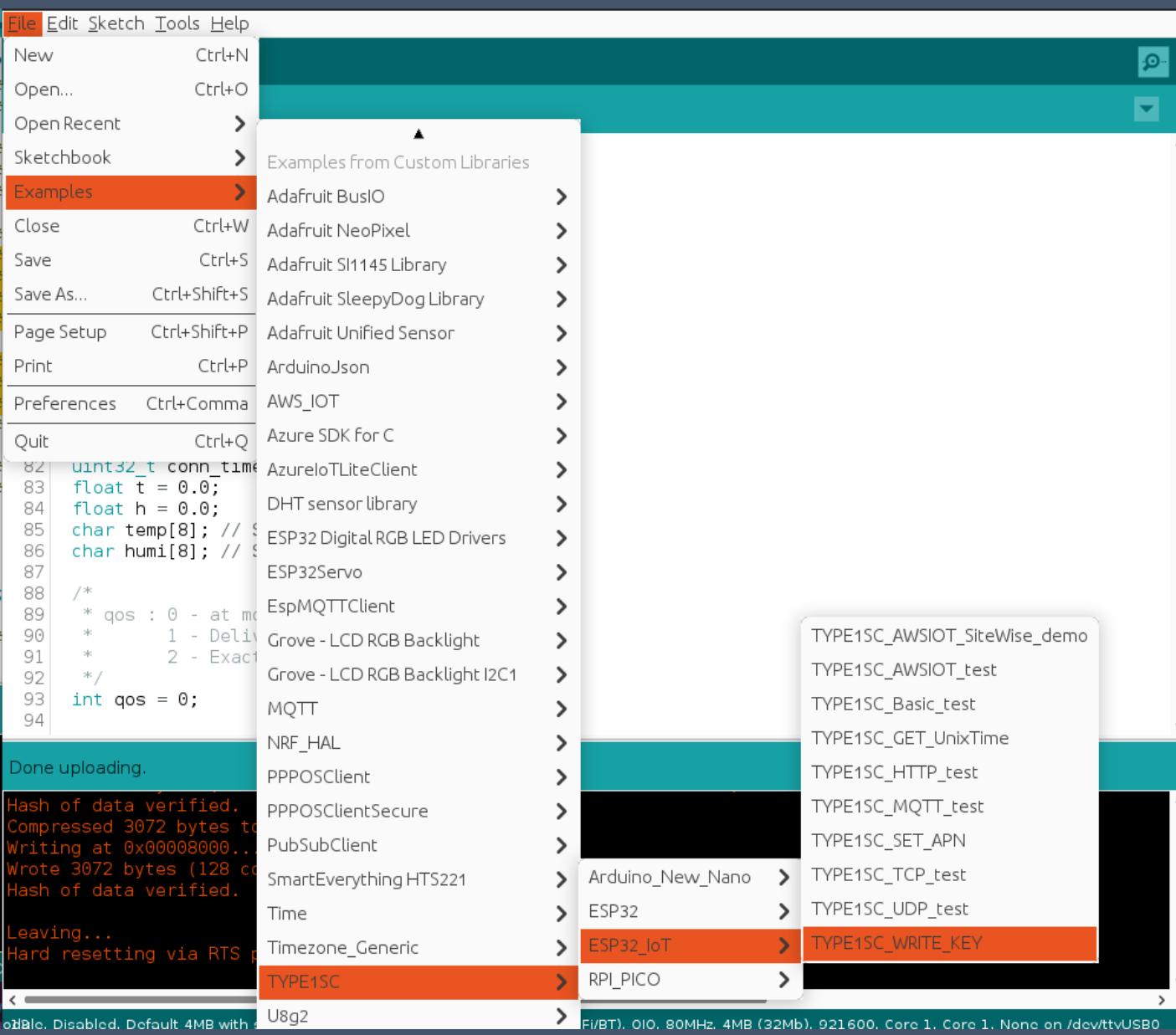
도메인 구성 생성

8) Write KEY (AWS-IoT Security Settings)

Store the security keys for accessing IoT services with AWS-IoT in the CAT.M1 modem internal file system.

1. 3 keys are used: AmazonRootCA file, certificate.pem.crt file, and private.pem.key file.
2. Open these keys with an editor program and add them to the example.

8) Write KEY (AWS-IoT Security Settings)



✓ Examples -> TYPE1SC-main -> ESP32_IoT ->
Select TYPE1SC_WRITE_KEY

8) Write KEY (AWS-IoT Security Settings)

```
TYPE1SC_WRITE_KEY - aws_credentials.h | 아두이노 1.8.13
파일 편집 스케치 툴 도움말

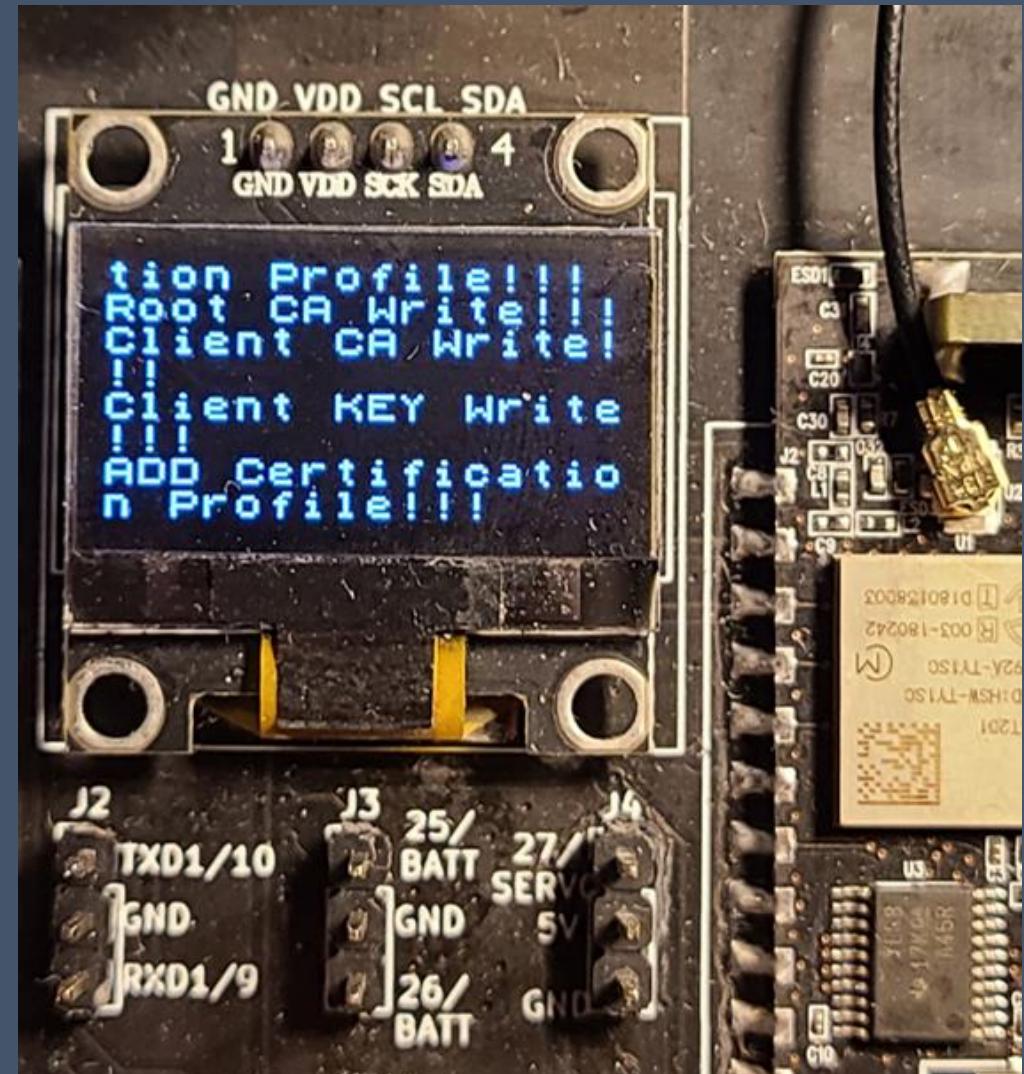
TYPE1SC_WRITE_KEY aws_credentials.h

1 #ifndef AWS_CREDENTIALS_H
2 #define AWS_CREDENTIALS_H
3 const char rootCA[] = "-----BEGIN CERTIFICATE-----\n"
4 "MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF\n"
5 "ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBbWF6\n"
6 "b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL\n"
7 "MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEZMBcGA1UEAxMQQW1hem9uIFJv\n"
8 "b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj\n"
9 "ca9HgFB0fW7Y14h29Jlo91ghYPl0hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzzM\n"
10 "9O6II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw\n"
11 "IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMjb1Hheb4mjUcAwhmahRWa6\n"
12 "VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kD1LL\n"
13 "93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm\n"
14 "jgSubJrIqg0CAwEAAsNCMEAwdwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC\n"
15 "AYYwHQYDVR0OBByEFIGYzIU07LwM1JQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA\n"
16 "A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r41K5IpDB/G/wkjUu0yKGX9rbxenDI\n"
17 "U5PMCCjjmCXPI6T53iHTFIUJrU6adTrCC2qJeHZERxh1bI1Bjjt/msv0tadq1wUs\n"
18 "N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKW1JbYK8U90vv\n"
19 "o/ufQJvtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1b1dZwgJcJmApzyMZFo6IQ6XU\n"
20 "5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\n"
21 "rqXRfboQnoZsG4q5WTP468SQvvG5\n"
22 "-----END CERTIFICATE-----";
23
24 const char clientCrt[] = "-----BEGIN CERTIFICATE-----\n"
```

- ✓ In aws_credentials.h, put the contents of the AmazonRootCA file, certificate.pem.crt file, and private.pem.key file line by line. The names below, which are declared as const char, are used by the internal library and should not be changed arbitrarily.
 - rootCA: AmazonRootCA file
 - clientCrt: certificate.pem.crt file
 - clientKey: private.pem.key
- ✓ The AWS-IoT security key files are stored securely in the Cat.M1 modem's internal filesystem, so you don't need to put them back each time. However, if the key changes, put it back into the Write Key example and run it again.

8) Write KEY (AWS-IoT Security Settings)

```
COM9  
06:45:45.026 -> TYPE1SC Module Start!!!  
06:45:47.294 -> Network not Ready !!!  
06:45:49.276 -> Network not Ready !!!  
06:45:51.268 -> TYPE1SC Module Ready!!!  
06:45:51.268 -> Delete Certification Profile!!!  
06:45:53.384 -> Root CA Write!!!  
06:45:58.494 -> Client CA Write!!!  
06:46:03.830 -> Client KEY Write!!!  
06:46:09.125 -> ADD Certification Profile!!!
```



9) AWS-IoT Test

Use the AWS-IoT communication stack inside the CAT.M1 modem to upload the temperature and humidity data measured by the Arduino to the AWS IoT Core using the AWS-IoT communication stack.

1. You must be ready to use AWS-IoT through steps 6) and 7) and have a security key written inside the CAT.M1 modem to access AWS IoT Core. The security key to access AWS IoT Core must be written inside the CAT.M1 modem. The access address uses the device endpoint.
6) Refer to the last part
2. Set up the AWS Console to connect with AWS IoT Core.

9) AWS-IoT Test

aws 서비스 ▾

플랫 지표 새 기능

▶ 플랫 허브

▶ Greengrass

▼ 보안

- 인증서
- 정책
- CA
- 역할 별칭
- 권한 부여자

▶ 보호

▶ 동작

테스트

소프트웨어
설정
알아보기
기능 정보
설명서

▶ 새로운 콘솔 환경
의견을 알려주십시오

주제 구독 주제 게시

주제 필터 정보

주제 필터는 구독할 주제를 설명합니다. 주제 필터에는 MQTT 와일드카드 문자가 포함될 수 있습니다.

sdkTest/sub

▼ 추가 구성

유지할 메시지 수

MQTT 테스트 클라이언트는 이 주제 필터와 일치하는 주제에 게시된 최신 메시지의 대부분을 유지합니다.

10

서비스 품질

주제를 구독할 때 서비스 품질 0이 기본적으로 선택됩니다.

서비스 품질 0 – 메시지는 최대 한 번 전달됩니다.
 서비스 품질 1 – 메시지가 한 번 이상 전달됩니다.

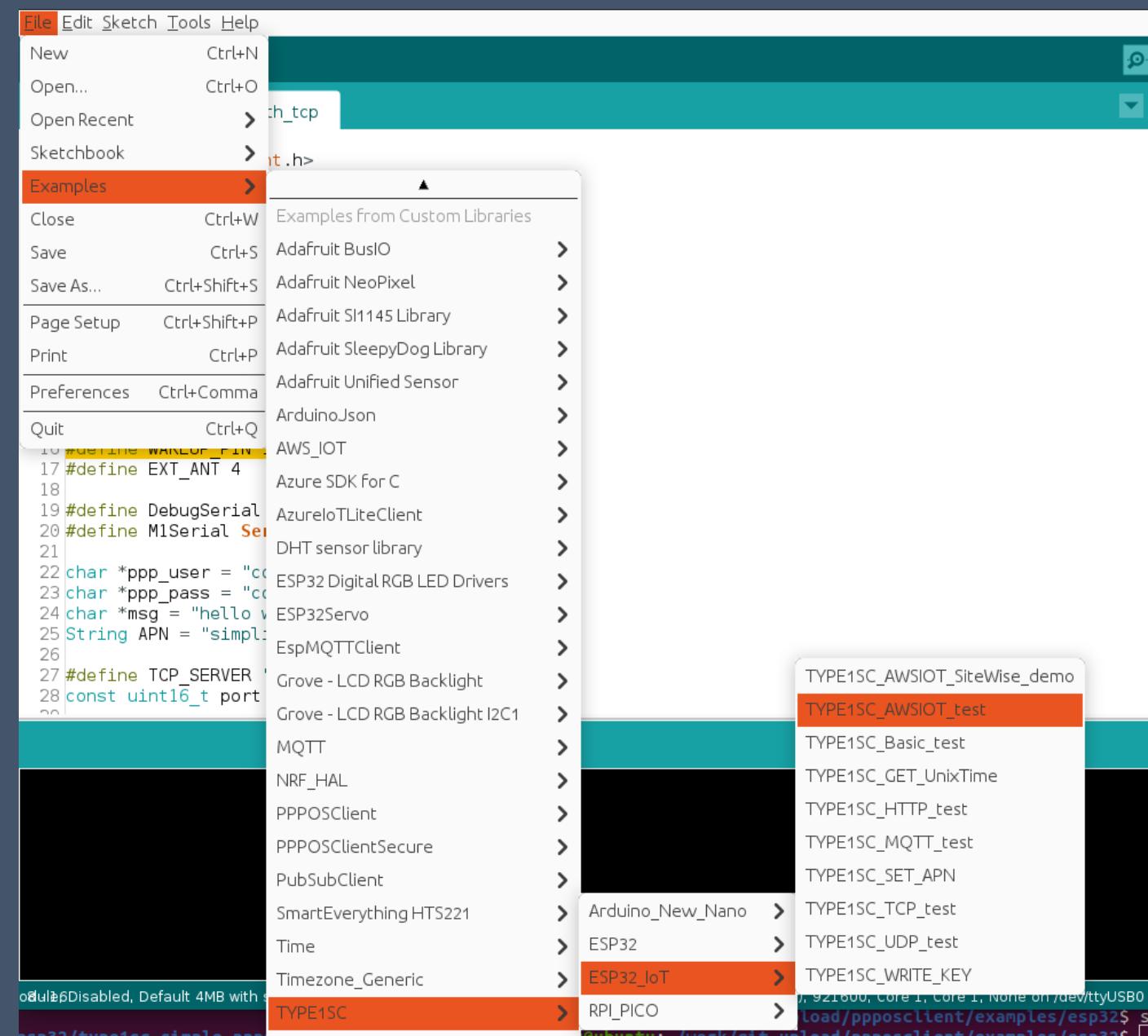
MQTT 페이로드 디스플레이

자동 형식 JSON 페이로드(가독성 향상)
 페이로드를 문자열로 표시(정확도 향상)
 원자 페이로드 표시(이전 데이터를 16진수 값으로 표시)

구독

구독	주제

9) AWS-IoT Test



✓ Examples -> TYPE1SC-main -> ESP32_IoT ->
Select TYPE1SC_AWSIOT_test

9) AWS-IoT Test

```
47 DebugSerial.println("TYPE1SC Module Ready!!!!");  
48  
49 /* Device Data EndPoint Address AWS IoT > Settings > Device data endpoint >  
50 * Copy&Paste */  
51 char _IP[] = "alpff!                .2.amazonaws.com";  
52  
53 char _NodeID[] = "Murata_Node01";  
54 char _Topic[] = "sdkTest/sub";  
55 char _message[64];  
56 int tlsProfile = 9;  
57 int conn_timeout = 1200;  
58 float t = 0.0; // Stores temperature value  
59 float h = 0.0; // Stores humidity value  
60 char temp[8];  
61 char humi[8];  
62
```

✓ IoT Core settings -> Enter device endpoint address

9) AWS-IoT Test

MQTT 페이로드 디스플레이

- 자동 형식 JSON 페이로드(가독성 향상)
- 페이로드를 문자열로 표시(정확도 향상)
- 원시 페이로드 표시(이진 데이터를 16진수 값으로 표시)

구독

구독	sdkTest/sub	일시 증지	지우기	내보내기	편집
sdkTest/sub	<p>▼ sdkTest/sub</p> <p>Temperature/26.0, Humidity/79.3</p>	September 01, 2021, 07:04:55 (UTC+0900)			

sdkTest/sub

▼ sdkTest/sub

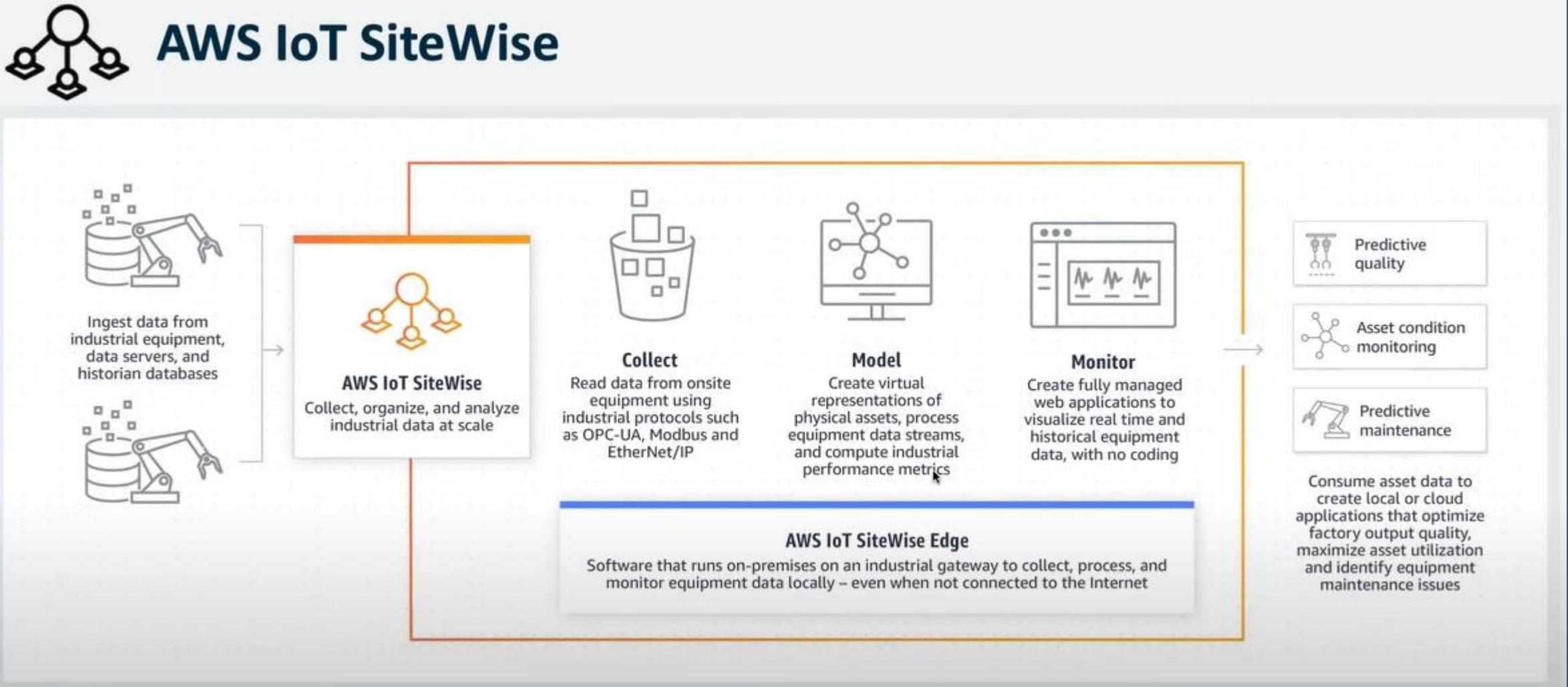
Temperature/26.0, Humidity/79.3

September 01, 2021, 07:04:55 (UTC+0900)

COM9

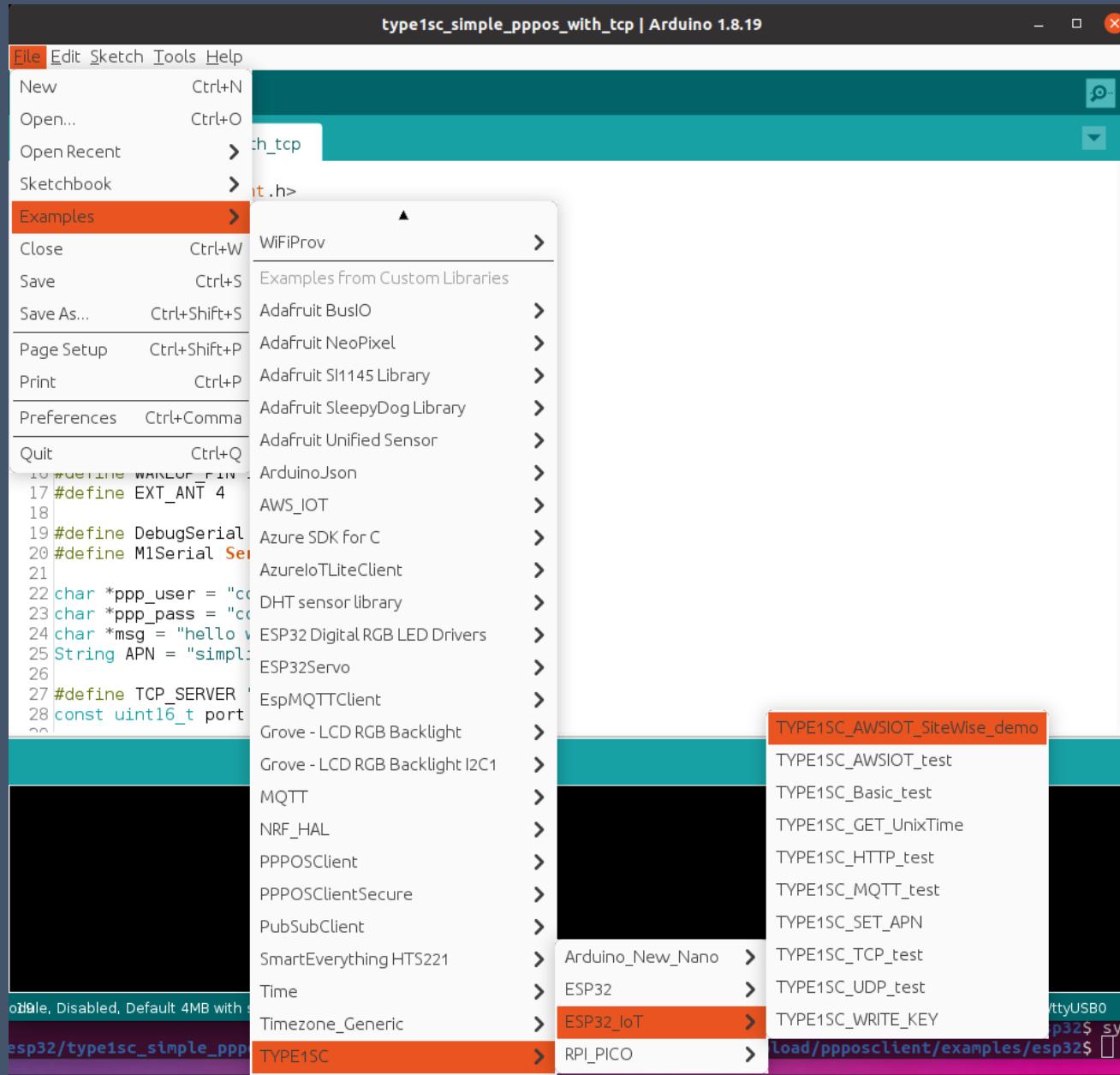
```
07:04:36.039 -> TYPE1SC Module Start!!!
07:04:38.224 -> Network not Ready !!!
07:04:40.255 -> Network not Ready !!!
07:04:42.231 -> Network not Ready !!!
07:04:44.259 -> TYPE1SC Module Ready!!!
07:04:44.259 -> 1.Configure AWS_IOT parameter:ID, Address, tls Profile
07:04:44.259 -> 2.Configure AWS_IOT parameter:Timeout
07:04:44.259 -> 3.Enable AWS_IOT events
07:04:50.006 -> 4.Establish connection
07:04:52.798 -> 5.Subscribe to the topic on the endpoint
07:04:55.568 -> 6.Publish data to broker
07:04:57.585 -> 7.UnSubscribe to the topic on the endpoint
07:04:59.555 -> 8.Disconnect AWS_IOT Service
07:04:59.555 -> 9.Disable AWS_IOT events
```

10) AWS-IoT SiteWise Demo



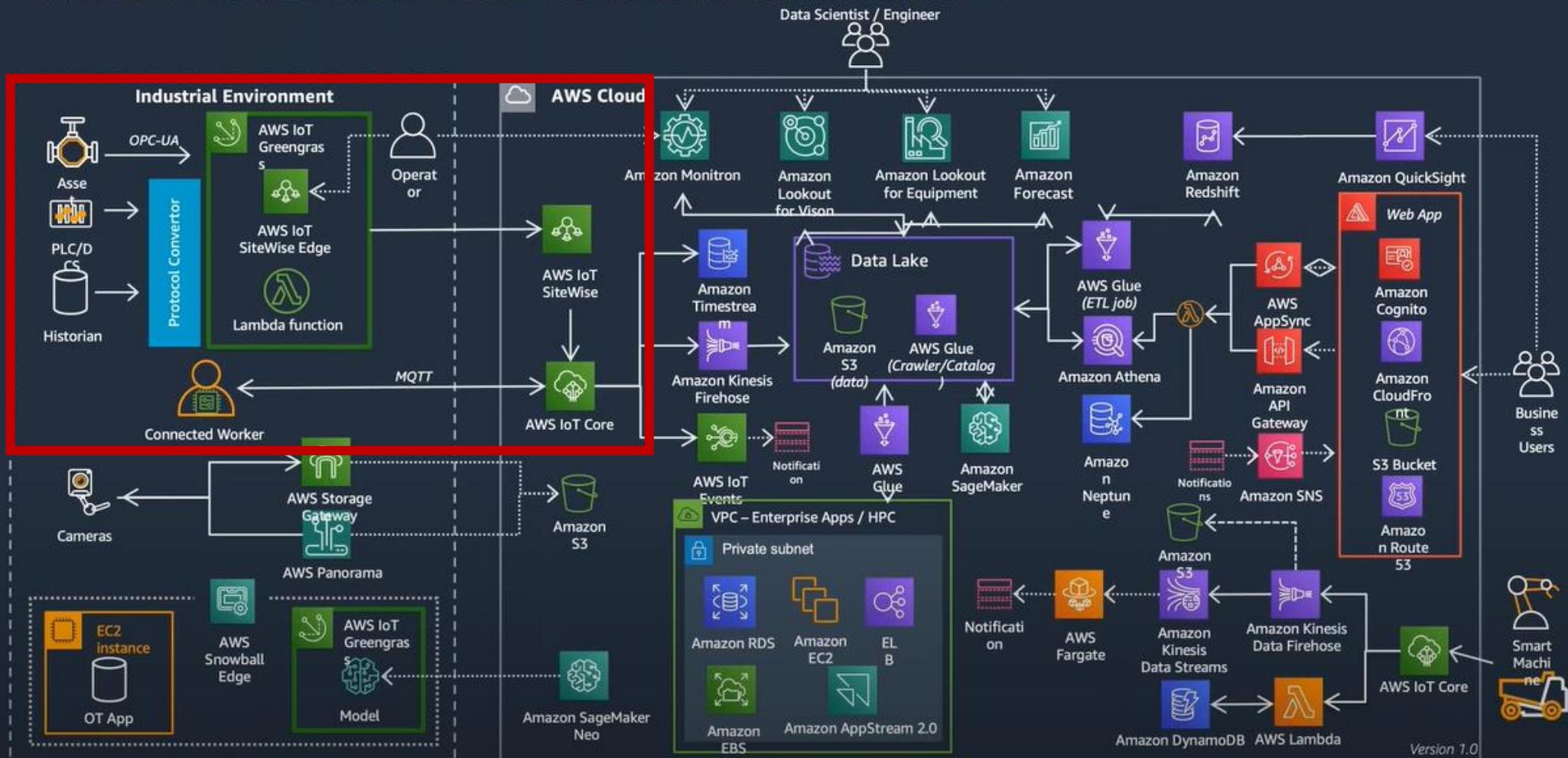
source : <https://aws.amazon.com/ko/iot-sitewise/>

10) AWS-IoT SiteWise Demo

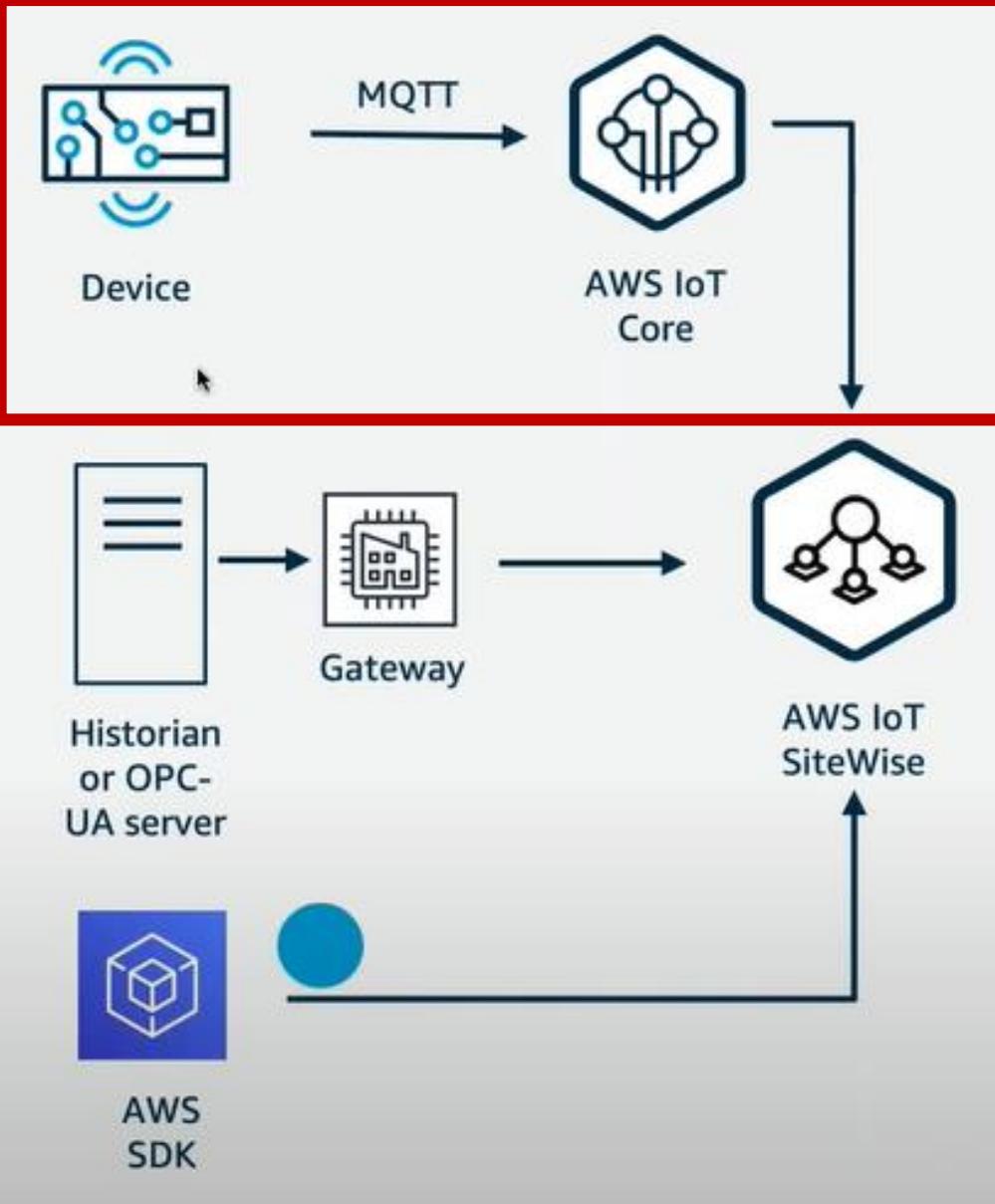


✓ Examples -> TYPE1SC-main -> ESP32_IoT -> TYPE1SC_AWSIOT_SiteWise_demo

AWS Industrial Reference Architecture



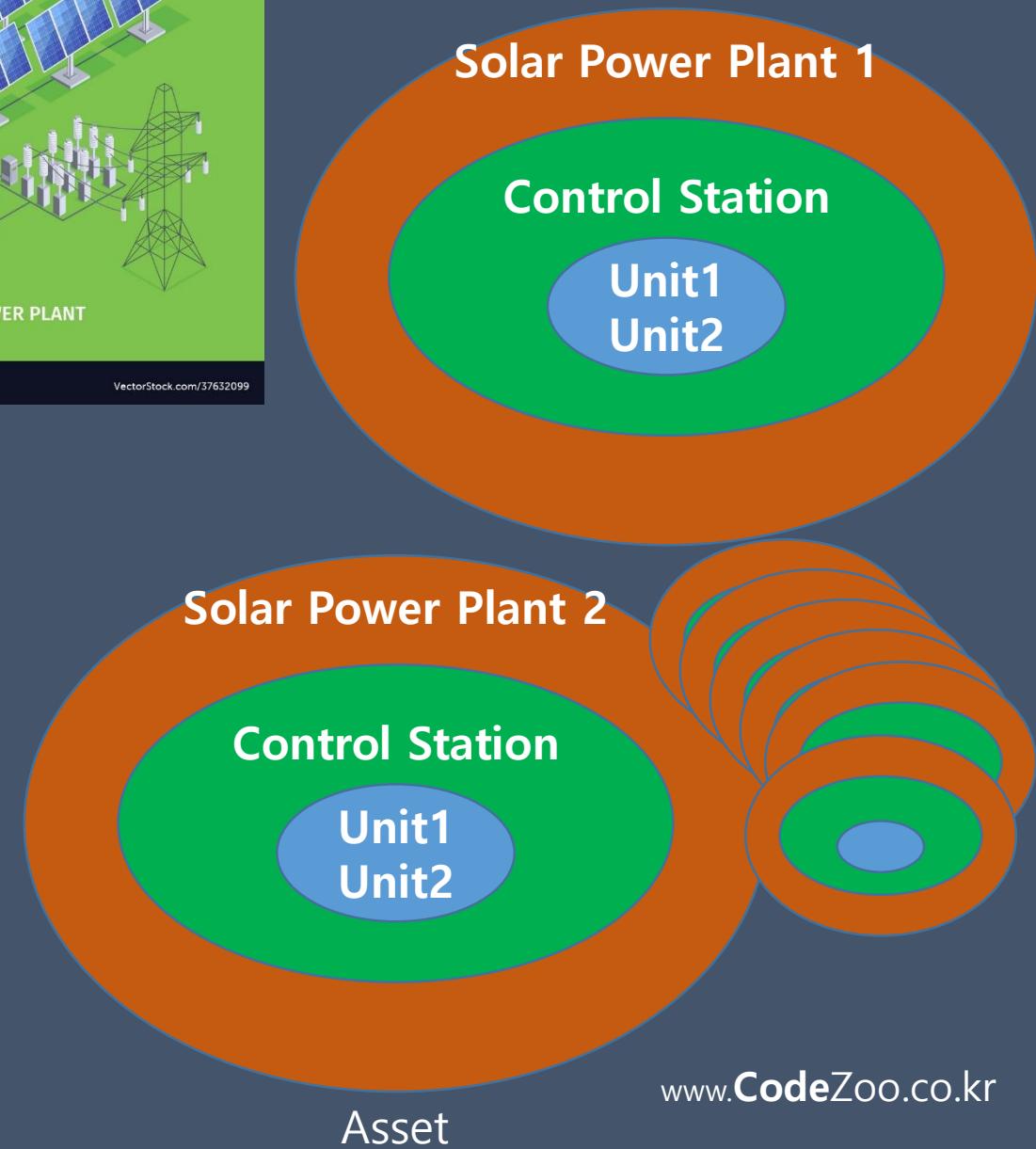
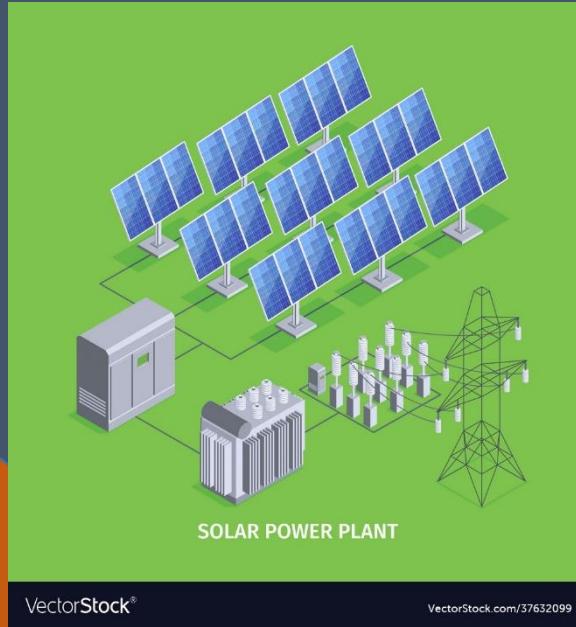
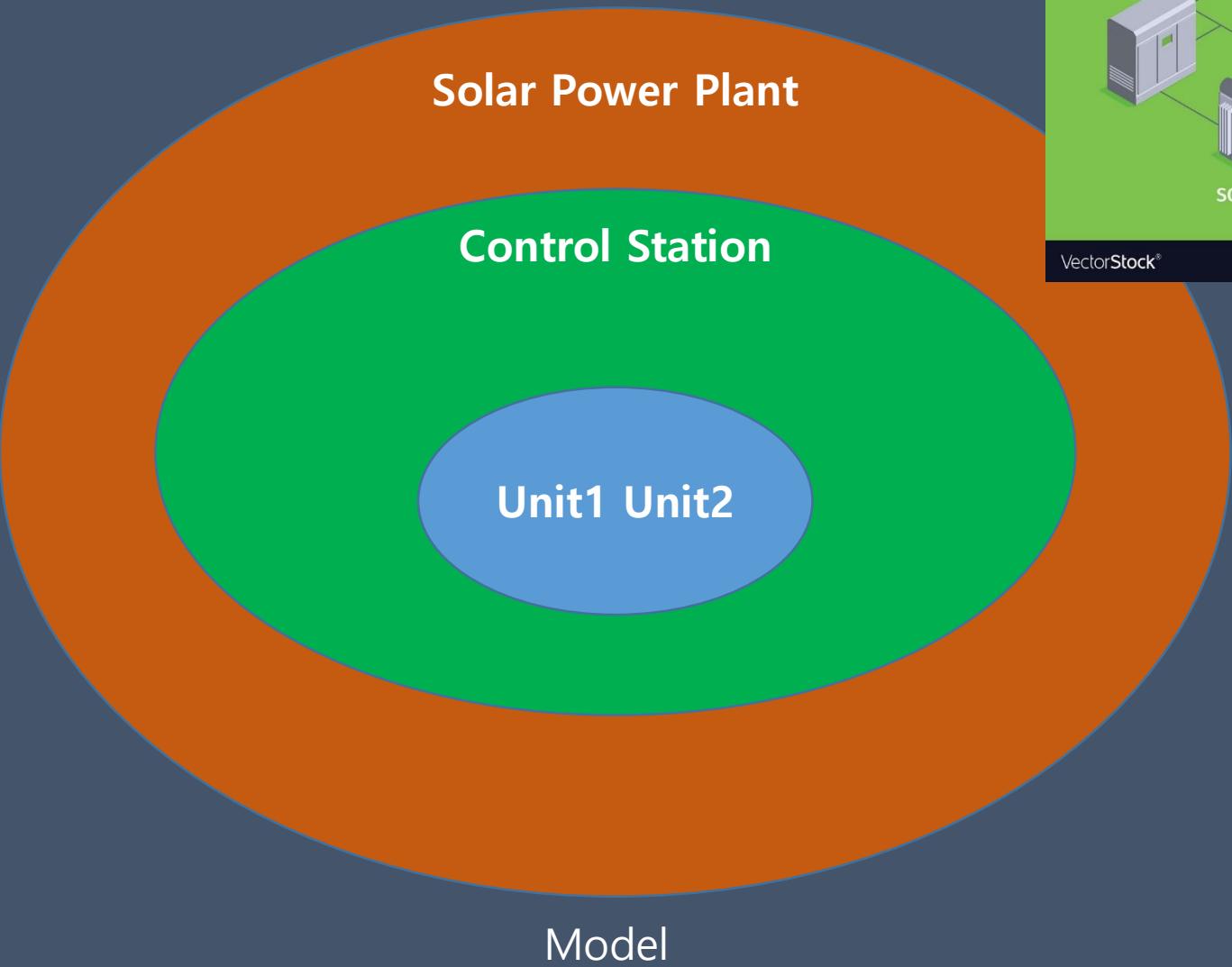
source : AWS IoT SiteWise のご紹介
<https://youtu.be/2sQEDuCQLks>



- ✓ AWS IoT SiteWise related information is Mostly publicly available as OPC-UA server + AWS GreenGlass
- ✓ For devices that are not bound to OPC-UA Protocol Send data to AWS IoT Core via MQTT to integrate with AWS IoT SiteWise service
- ✓ Post messages to MQTT topics in AWS IoT Core and send data with AWS IoT rules and pass the data to AWS IoT SiteWise.

source: AWS IoT SiteWise のご紹介
<https://youtu.be/2sQEDuCQLks>

WHAT TO BUILD? FROM MODELS, ASSET CONCEPTS



AWS IoT SiteWise X

▼ 엣지

게이트웨이

▼ 빌드

모델

자산

데이터 스트림

▼ 모니터링

시작하기

포털

▼ 설정

로깅 옵션

암호화

스토리지

모델 (4)

자산은 데이터 스트림을 SiteWise로 전송하는 산업 디바이스 및 프로세스를 나타냅니다. 모델은 각 자산의 모든 인스턴스에 대해 특정 속성 및 계층 구조 모델을 적용하는 구조입니다. 모델에서

 인스턴스 필터링

이름	상태	생성된 날짜
Demo Solar Power Plant Model	<input checked="" type="checkbox"/> ACTIVE	2022년 2월 07일 17:47:03 (UTC+9:00)
Control Station Model	<input checked="" type="checkbox"/> ACTIVE	2022년 2월 07일 17:44:22 (UTC+9:00)
Unit 2 Model	<input checked="" type="checkbox"/> ACTIVE	2022년 2월 07일 17:40:52 (UTC+9:00)
Unit 1 Model	<input checked="" type="checkbox"/> ACTIVE	2022년 2월 07일 17:38:00 (UTC+9:00)

새로운 기능 4

Define a model

1. Demo Solar Power Plant Model
2. Control Station Model
3. Unit 2 Model
4. Unit 1 Model

자산 및 모델

자산은 데이터 스트림을 SiteWise로 전송하는 산업 디바이스 및 프로세스를 나타냅니다. 모델은 각 자산의 모든 인스턴스에 대해 특정 속성 및 계층 구조 모델을 적용하는 구조입니다. 모델에서 모든 자산을 생성해야 합니다.

이 페이지에서 속성, 측정, 변환, 지표 및 계층 구조를 사용하여 모델을 생성할 수 있습니다. 그런 다음 모델에서 자산을 생성하고 각 자산을 자산 관련 정보로 채울 수 있습니다. [자세히 알아보기](#)

모델 세부 정보

이름
기억하기 쉬운 이름을 사용하면 사용자와 다른 사용자가 모델을 쉽게 찾고 이해할 수 있습니다.

Demo Solar Power Plant Model

이 이름은 고유하고 256자 미만이어야 합니다.

설명 - 선택 사항
자산 모델에 대한 설명입니다.

Demo Solar Power Plant Model Template

2,048자 미만이어야 합니다.

속성 정의

속성은 거의 변경되지 않는 자산의 값(예: 디바이스 일련 번호 또는 부품 번호)입니다. [자세히 알아보기](#)

Transforming definitions: adding a formula to the numbers you get from Things to generate the data values you need

Metric definitions: formulas that use aggregation functions to process data points over a specified time interval. Where you enter something like the average temperature over a specific period of time.

AWS IoT SiteWise ×

▼ 엣지

게이트웨이

▼ 빌드

모델

자산

자산 (1)



자산 생성

자산은 데이터 스트림을 SiteWise로 전송하는 산업 디바이스 및 프로세스를 나타냅니다. 모델은 각 자산의 모든 인스턴스에 대해 특정 속성 및 계층 구조 모델을 적용하는 구조입니다. 모델에서 모든 자산을 생성해야 합니다.

 최상위 수준 자산 필터링

< 1 >

이름



상태



생성된 날짜



수정한 날짜



Demo Solar Power Plant 1

ACTIVE

2022년 2월 07일 18:03:38 (UTC+9:00)

2022년 2월 08일 16:18:09 (UTC+9:00)

Control Station 1

ACTIVE

2022년 2월 07일 18:00:11 (UTC+9:00)

2022년 2월 08일 16:17:35 (UTC+9:00)

Control Staion 1 - Unit 1

ACTIVE

2022년 2월 07일 17:49:50 (UTC+9:00)

2022년 2월 08일 20:33:35 (UTC+9:00)

Control Station 1 - Unit 2

ACTIVE

2022년 2월 07일 17:51:15 (UTC+9:00)

2022년 2월 08일 23:03:32 (UTC+9:00)

새로운 기능 4

Defining assets

1. Demo Solar Power Plant 1
2. Control Station 1
3. Control Station 1 - Unit 1
4. Control Station 1 - Unit 2

모델 정보

모델
자산은 모델에 정의된 속성을 상속합니다.

Demo Solar Power Plant Model

모델이 필요합니다.

자산 정보

이름
기억하기 쉬운 이름을 사용하면 사용자와 다른 사용자가 자산을 쉽게 찾고 이해할 수 있습니다.

Demo Solar Power Plant 1

이 이름은 고유하고 256자 미만이어야 합니다.

이 자산에 연결된 자산

계층 구조	자산	연결 해제
Control Staion Model	Control Station 1	▼

AWS IoT SiteWise



AWS IoT SiteWise > 모니터링 > 포털 > SiteWise demo Solar Power Plant

SiteWise demo Solar Power Plant

▼ 엣지

게이트웨이

▼ 빌드

모델

자산

데이터 스트림

▼ 모니터링

시작하기

포털

▼ 설정

로깅 옵션

암호화

스토리지

새로운 기능

4

포털 세부 정보

이름	SiteWise demo Solar Power Plant	상태	ACTIVE	URL
설명	-	로고 이미지	-	https://p-brzzb8b8.app.iotsitewise.aws

권한

서비스 역할	AWSIoTSiteWiseMonitorServiceRole_28uMKybvN	엔터프라이즈 자격 증명 소스
		AWS SSO

추가 기능

추가 기능이 활성화됨	AWS IoT SiteWise 액세스 역할	발신자
-	-	-



Demo Solar Power Plant 1



Last day

Live

2022년 2월 15일 오전
2:47:432022년 2월 16일 오전
2:47:43

GMT+9

Edit

Dashboards



Assets



Models



Projects



Users



BYOUNG NAM JANG

English (US) >

Help

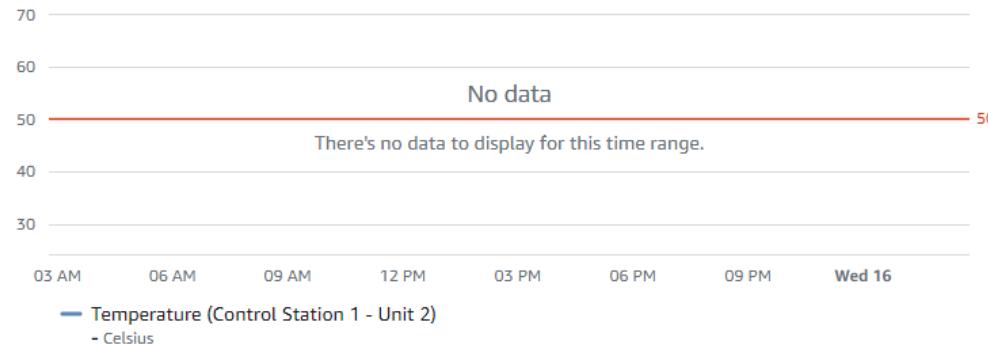


Log Out



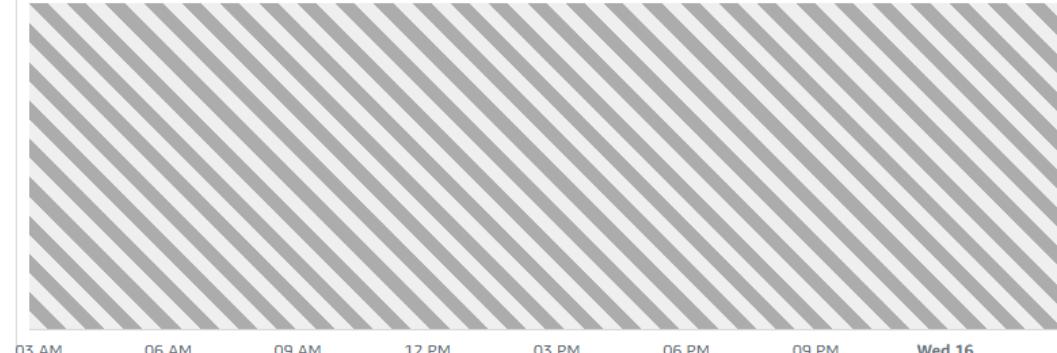
Temperature

Celsius



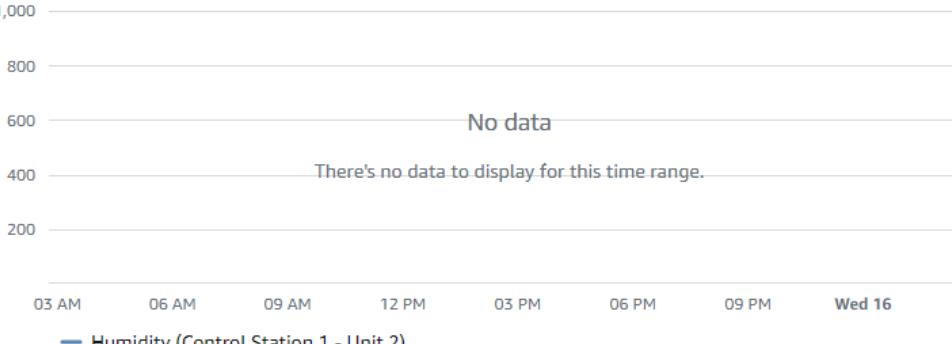
Person existence

Person existence (Control Staion 1 - Unit 1)



Humidity

%



AWS IOT RULES

The screenshot shows the AWS IoT Rules interface in Korean. On the left, there is a sidebar with various navigation options. The 'Rules' option is highlighted with an orange circle. The main content area shows a search bar and two filter options: '이름' and 'SiteWiseDeviceRule'.

AWS IoT

모니터링

활동

▶ Connect

▶ 관리

▶ 플릿 허브

▶ Greengrass

▶ 보안

▶ 보호

▼ 동작

개요

규칙

대상

테스트

AWS IoT > 규칙

규칙

검색 범위

이름

SiteWiseDeviceRule

SiteWiseDeviceRule

규칙 쿼리 설명문

편집

이 규칙을 사용하여 처리하고자 하는 메시지의 소스입니다.

```
SELECT * FROM 'iot/topic'
```

SQL 버전 사용 2016-03-23

작업

작업은 규칙이 트리거되면 이루어지는 것입니다. 자세히 알아보기



AWS IoT SiteWise의 자산 속성에 메시지 데이터 ...
iotSiteWise-resource

제거

편집



작업 추가

오류 작업

규칙 처리 도중 문제가 발생할 때 실행될 작업을 선택적으로 설정합니다.

작업 추가

이 작업은 메시지를 AWS IoT SiteWise의 자산 속성으로 전송합니다.

속성 별칭 기준

속성 별칭으로 속성을 식별합니다.

자산 ID 및 속성 ID 기준

자산 ID 및 속성 ID로 속성을 식별합니다.

AWS IoT SiteWise의 자산 모델에서 속성 ID 선택

선택한 자산 모델이 없음 선택

*자산 ID 정보 선택
21ae2c1c-5796-49fa-a00b-0784eb46970

*속성 ID 선택
87cad420-54fd-4cd2-b0b9-28af56757ca

항목 ID 선택
예: \${entryId}

*시간(초) 선택
\${timeInSeconds}

오프셋(나노초) 선택
예: \${offsetInNanos}

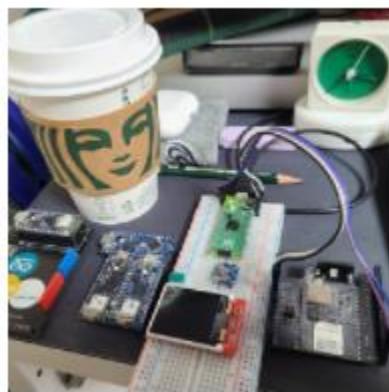
*값 선택
\${Temperature}

*데이터 유형 선택
Double

*시간(초) ?

`${timeInSeconds}`

<https://codezoo.tistory.com/22>



LTE-CATM1 내장형 모뎀으로 Unix Timestamp 처리하기

카테고리 없음 · 2022. 2. 13. 16:53

IoT, AIoT에서 수집한 정보를 서버에 전송할 때 서버에서 요구하는 사항 중 하나가 Unix Timestamp 값 (데이터를 전송하는 현재시간)입니다. 필요한 이유는 단말기에서...

[Read More](#)

Unix Timestamp open source

*자산 ID 정보

21ae2c1c-5796-49fa-a00b-0784eb46970

*속성 ID ?

87cad420-54fd-4cd2-b0b9-28af5675a5

속성 측정값 변환 지표 연결된 자산 경보 태그

측정값 (2)

이름	ID	Alias	MQTT 알림 상태
Humidity	f35a464a-acdd-42c2-8d9e-e803a0bc2b52	/ControlStation1/ControlStation1-Unit2/ControlStationHumidity	비활성화됨
Temperature	87cad420-54fd-4cd2-b0b9-28af567575a5	/ControlStation1/ControlStation1-Unit2/ControlStationTemperature	비활성화됨

Control Station 1 - Unit 2

자산 세부 정보

자산 ID <code>21ae2c1c-5796-49fa-a00b-0784eb469703</code>	마지막으로 수정한 날짜 2022년 2월 08일 23:03:32 (UTC+9:00)	상태 ACTIVE
모델 Unit 2 Model	생성된 날짜 2022년 2월 07일 17:51:15 (UTC+9:00)	

속성 측정값 변환 지표 연결된 자산 경보 태그

www.CodeZoo.co.kr

AWS IoT

구독

주제 구독

주제 게시

모니터링
활동
▶ Connect
▶ 관리
▶ 플랫 허브
▶ Greengrass
▶ 보안
▶ 보호
▼ 동작
개요
규칙
대상
테스트

구독
디바이스는 주제에 MQTT 메시지를 게시합니다. 이 클라이언트를 사용해 주제를 구독한 다음 이러한 메시지를 수신할 수 있습니다.

구독 주제
iot/topic

주제 구독

최대 메시지 캡처 정보
100

서비스 품질 정보
 0 - 이 클라이언트는 메시지가 수신되는 디바이스 게이트웨이를 인지하지 못합니다.
 1 - 이 클라이언트는 메시지가 수신되는 디바이스 게이트웨이를 인지합니다.

MQTT 페이로드 디스플레이
 Auto-format JSON 페이로드(가독성 향상)
 페이로드를 문자열로 표시(정확도 향상)
 원시 페이로드 표시(16진수)

게시
QoS of 0으로 게시할 주제와 메시지를 지정합니다.

iot/topic

1
2 "timeInSeconds": "1645093624",
3 "Temperature": "34.5",
4 "Humidity": "48.23"
5 }

주제 게시

{
 "timeInSeconds": "1645093624",
 "Temperature": "34.5",
 "Humidity": "48.23"
}

AWS IoT SiteWise

Unit 2 Model
2022년 2월 07일 17:51:15
(UTC+9:00)

净资产

측정값 (2)

MQTT 알림 상태	알림 주제	최신 값	최신 값 타임 스탬프
비활성화됨	-	48.23	2022년 2월 17일 19:27:04 (UTC+9:00)
비활성화됨	-	34.5	2022년 2월 17일 19:27:04 (UTC+9:00)

www.CodeZoo.co.kr

현재 유닉스 시간 구하기 : <https://www.epochconverter.com/>

DEMO

SiteWise demo Solar Power Plant > 프로젝트 > SiteWise Demo Project > Demo Solar Power Plant 1

Demo Solar Power Plant 1

Last 10 minutes Live ▾

Temperature

Celsius

50
45
40
35
30
25

03:10 03:11 03:12 03:13 03:14 03:15 03:16 03:17 03:18 03:19

Temperature (Control Station 1 - Unit 2)
27.4 Celsius

Humidity

%

50.0
49.5
49.0
48.5
48.0
47.5
47.0
46.5
46.0

03:10 03:11 03:12 03:13 03:14 03:15 03:16 03:17 03:18 03:19

Humidity (Control Station 1 - Unit 2)
48 %

Person existance

Person existance (Control Station 1 - Unit 2)

03:10 03:11 03:12 03:13 03:14 03:15 03:16 03:17 03:18 03:19

03:19:43.690 -> /=====/
03:19:43.690 ->
03:19:43.690 -> 1.Configure AWS_IOT parameter:ID, Address, tls Profile
03:19:43.690 -> 2.Configure AWS_IOT parameter:Timeout
03:19:43.690 -> 3.Enable AWS_IOT events
03:19:52.495 -> 4.Establish connection
03:19:55.040 -> 5.Subscribe to the topic on the endpoint
03:19:55.040 -> 27.40
03:19:55.040 -> 48.30
03:19:57.792 -> 6.Publish data to broker
03:19:59.748 -> 7.UnSubscribe to the topic on the endpoint
03:20:01.757 -> 8.Disconnect AWS_IOT Service
03:20:01.757 -> 9.Disable AWS_IOT events
03:20:01.757 -> /=====/
03:20:01.757 ->
03:20:01.803 -> 1.Configure AWS_IOT parameter:ID, Address, tls Profile
03:20:01.803 -> 2.Configure AWS_IOT parameter:Timeout
03:20:01.803 -> 3.Enable AWS_IOT events
03:20:08.009 -> 4.Establish connection
03:20:10.710 -> 5.Subscribe to the topic on the endpoint
03:20:10.710 -> 27.40
03:20:10.710 -> 47.80
03:20:13.365 -> 6.Publish data to broker

6. ESP32 IoT Lab (Control)

- ✓ This is experimental work. Codeweek is not responsible for device control and hacking. Publicly available development resources cannot be guaranteed to be complete for device behavior, and security features should be added and thoroughly reviewed before using them for commercial services. - CodeZoo -
1. Tackle concurrency issues: You should use an RTOS or GPOS that can handle concurrency. In layman's terms, this is because you can call it externally at any time, even when the system is doing something else.
 2. Program structure change: Since we are using the OS scheduler, we need to have a separate task structure, rather than the traditional single branch structure.
 3. Change the communication method: For various reasons, it is difficult to guarantee instant responsiveness with AT command, so it is necessary to use PPP (Point to Point Protocol) communication method, which is another way of modem operation. In this case, the data is streamed, so you need a Network Stack to handle it.

Platform : ESP32 (ESP32-IDF)

RTOS : FreeRTOS

IP : LWIP(LightWeight IP)

LwIP Distribution protocols

2

Application protocols

- SNMP,
- DNS client,
- DHCP client,

Transport protocols

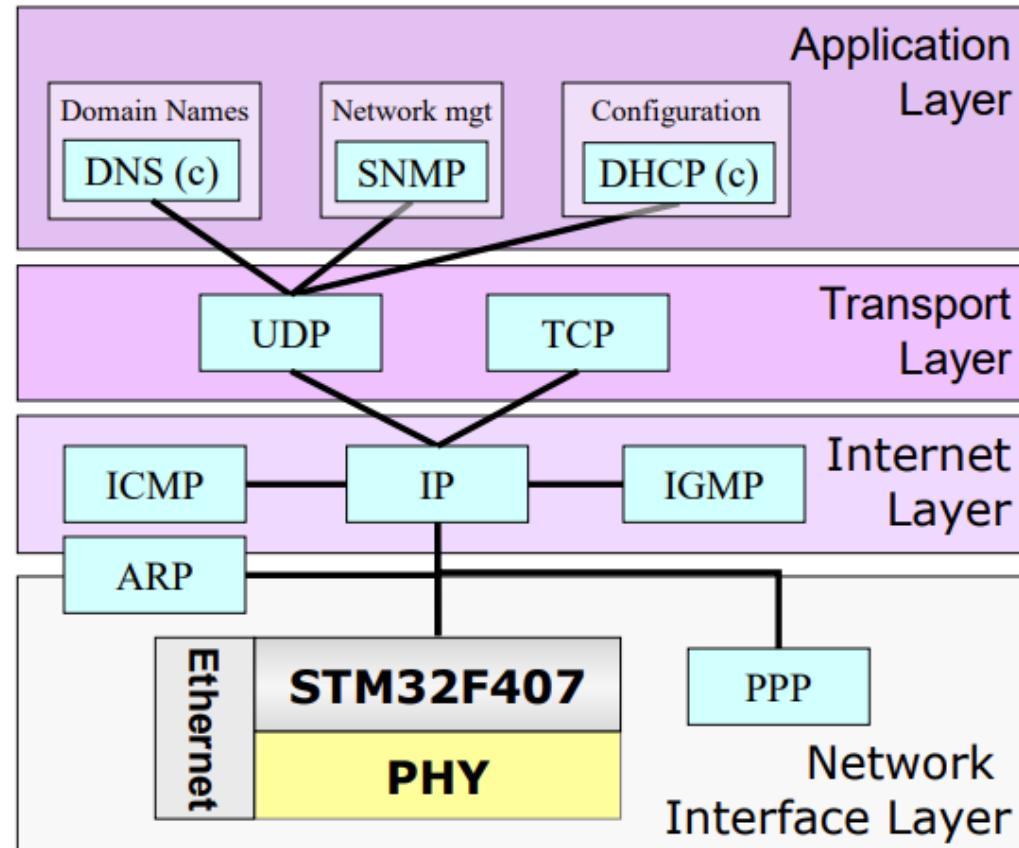
- UDP,
- TCP,

Internet Protocols

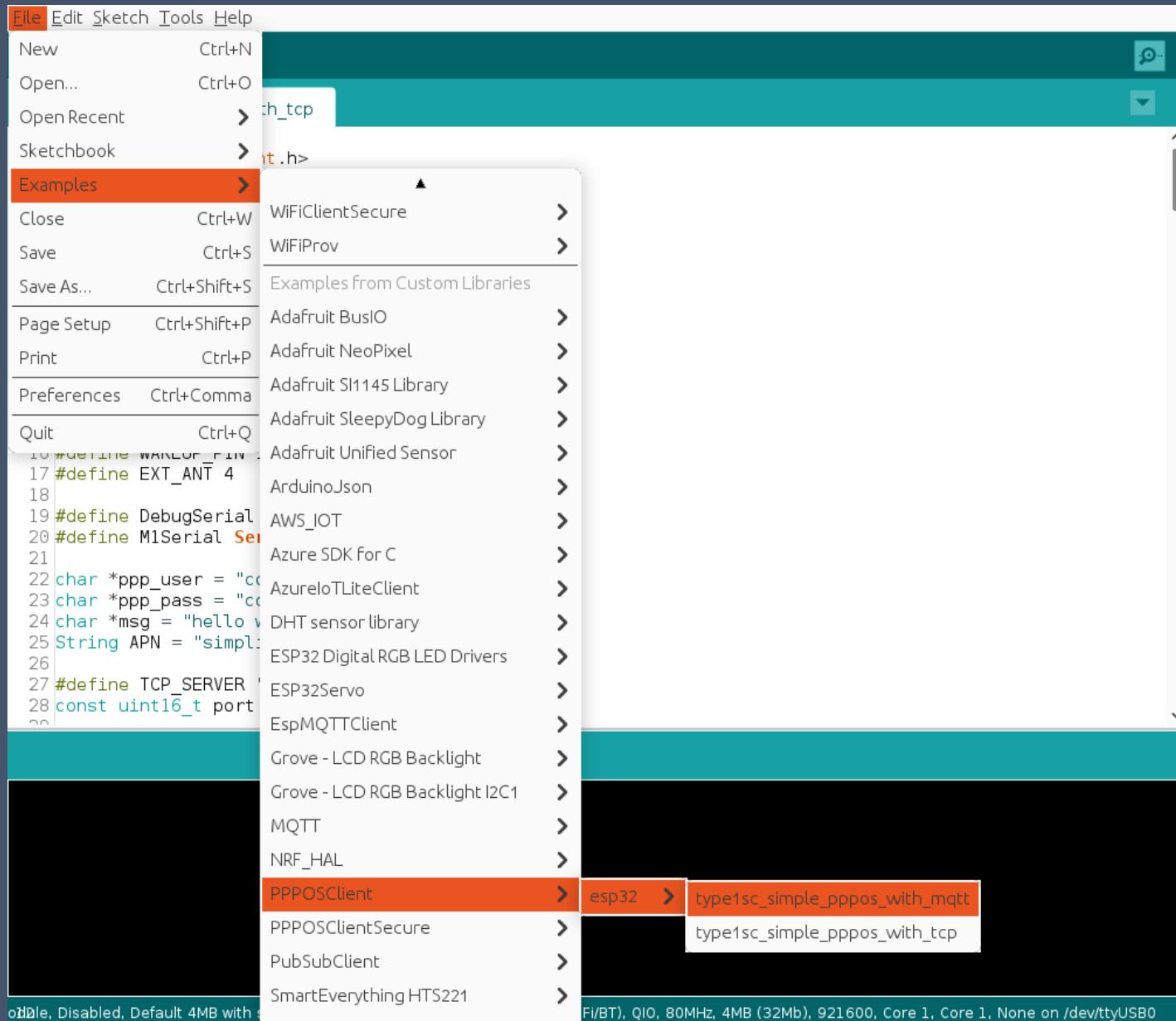
- ICMP,
- IGMP,

Datalink Protocols

- ARP,
- PPP



1) MQTT LED Remote Control



Examples -> PPPOSClient ->
type1sc_simple_pppos_with_mqtt

You can receive incoming control commands from the outside in a callback function and process them.

The screenshot shows the Arduino IDE interface. On the left, a code editor displays the following C++ code:

```
3 #include <PubSubClient.h>
4
5 #define SERIAL_BR 115200
6
7 // 라이브러리 매니저
8 타입 All 토픽 All pubsubclient
9
10
11 #include <PubSubClient.h>
12 // by Nick O'Leary
13 // A client library for MQTT messaging. MQTT is a lightweight messaging protocol ideal for small devices. This library allows you to send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol and can be configured to use the older MQTT 3.1 if needed. It supports all Arduino Ethernet Client compatible hardware, including the Intel Galileo/Edison, ESP8266 and TI CC3000.
14
15 // More info
16
17
18
19 // PubSubClientTools
20 char* PUB_TOPIC = "type1sc/100/status";
21
22 #define MQTT_SERVER "broker.hivemq.com"
23 String buffer = "";
24 char *data = (char *) malloc(1024);
25
```

On the right, a browser window displays the [PubSubClient library page](#). The page includes a brief description, a note about version 2.8.0, and a "설치" (Install) button.

At the bottom of the screen, a terminal window shows the following compilation output:

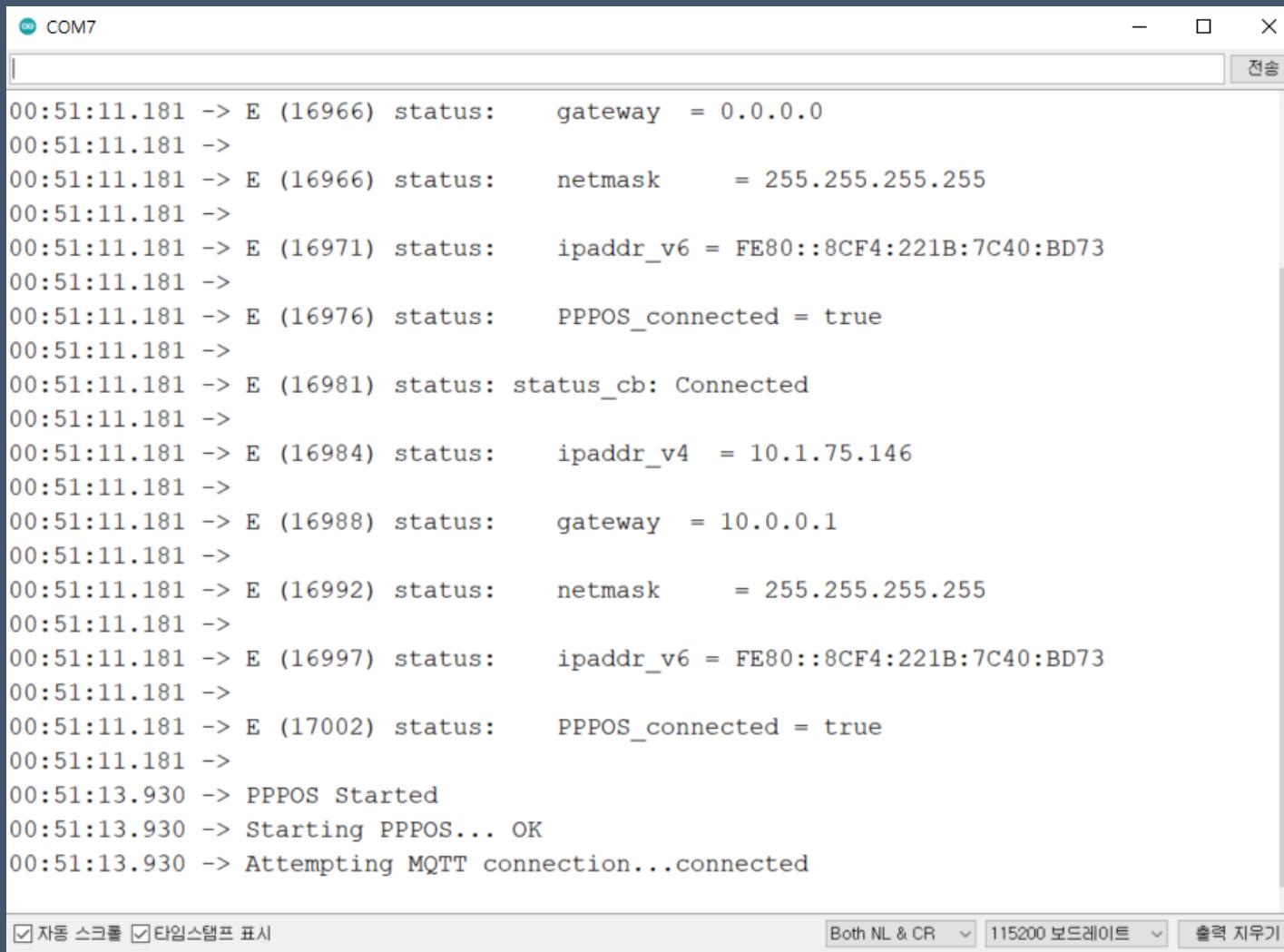
```
PubSubClient.h: No such file or directory
compilation terminated.
exit status 1
PubSubClient.h: No such file or directory
```

Build the example. If you don't already have the PubSubClient library installed if you don't already have it installed.

We tested with version 2.8.0.

Run the firmware build and upload.

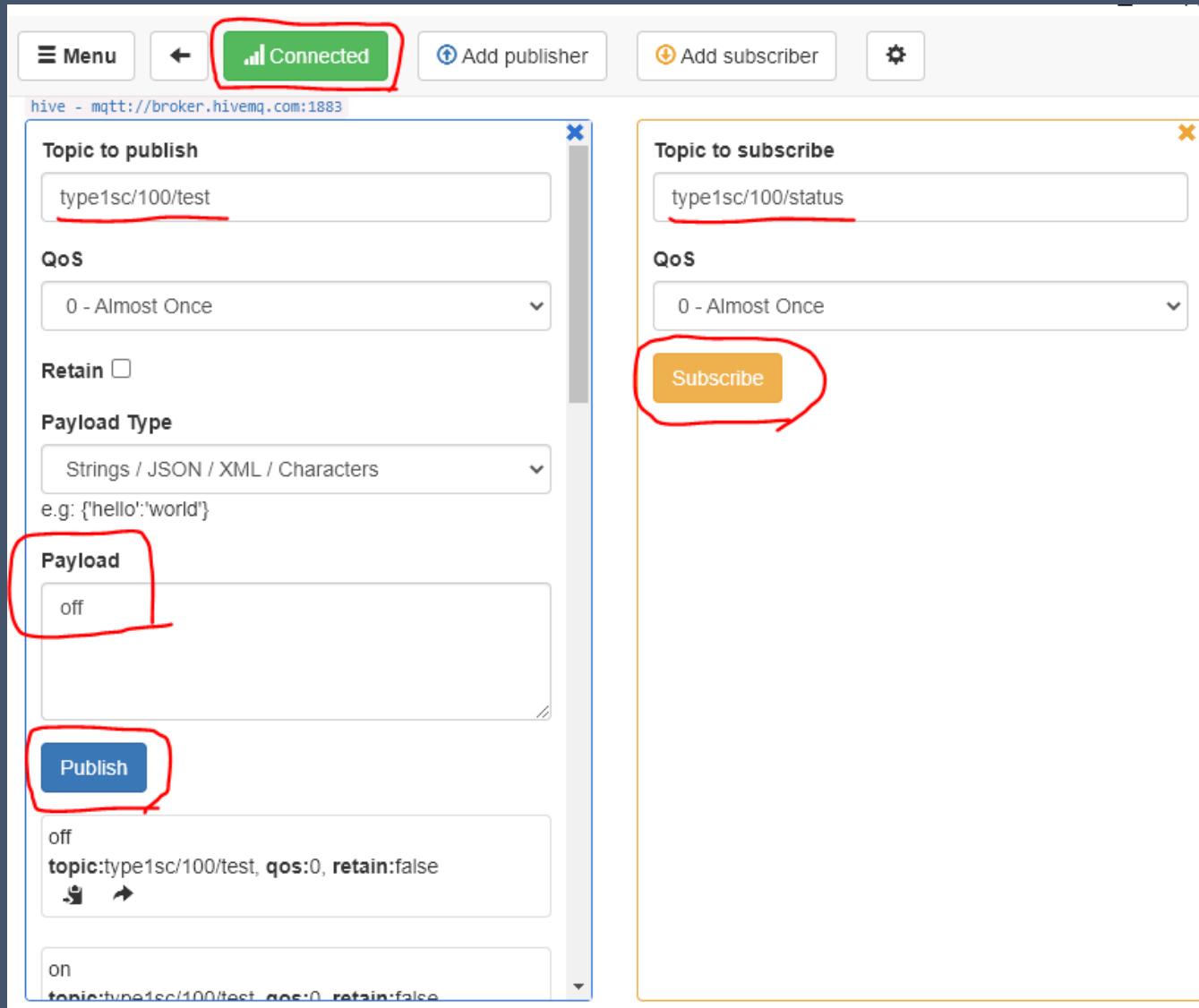
If the SIM is installed normally and the APN input is not incorrect, boot up and switch to PPP mode as shown below to obtain IP and check the status of connecting to the MQTT Broker. You can check the status of connecting to MQTT Broker. If the upload does not go well, hold down the boot button and the upload will proceed. upload will proceed.



```
00:51:11.181 -> E (16966) status: gateway = 0.0.0.0
00:51:11.181 ->
00:51:11.181 -> E (16966) status: netmask     = 255.255.255.255
00:51:11.181 ->
00:51:11.181 -> E (16971) status: ipaddr_v6 = FE80::8CF4:221B:7C40:BD73
00:51:11.181 ->
00:51:11.181 -> E (16976) status: PPPoS_connected = true
00:51:11.181 ->
00:51:11.181 -> E (16981) status: status_cb: Connected
00:51:11.181 ->
00:51:11.181 -> E (16984) status: ipaddr_v4   = 10.1.75.146
00:51:11.181 ->
00:51:11.181 -> E (16988) status: gateway   = 10.0.0.1
00:51:11.181 ->
00:51:11.181 -> E (16992) status: netmask     = 255.255.255.255
00:51:11.181 ->
00:51:11.181 -> E (16997) status: ipaddr_v6 = FE80::8CF4:221B:7C40:BD73
00:51:11.181 ->
00:51:11.181 -> E (17002) status: PPPoS_connected = true
00:51:11.181 ->
00:51:13.930 -> PPPoS Started
00:51:13.930 -> Starting PPPoS... OK
00:51:13.930 -> Attempting MQTT connection...connected
```

자동 스크롤 타임스탬프 표시 Both NL & CR 115200 보드레이트 출력 지우기

Refer to the MQTTBox settings below to set up an Android or iPhone MQTT app. The broker address is the same as Host : broker.hivemq.com Port :1883. Payload and Message are used interchangeably.



Use two topics as shown below

type1sc/100/test :

For controlling the device externally

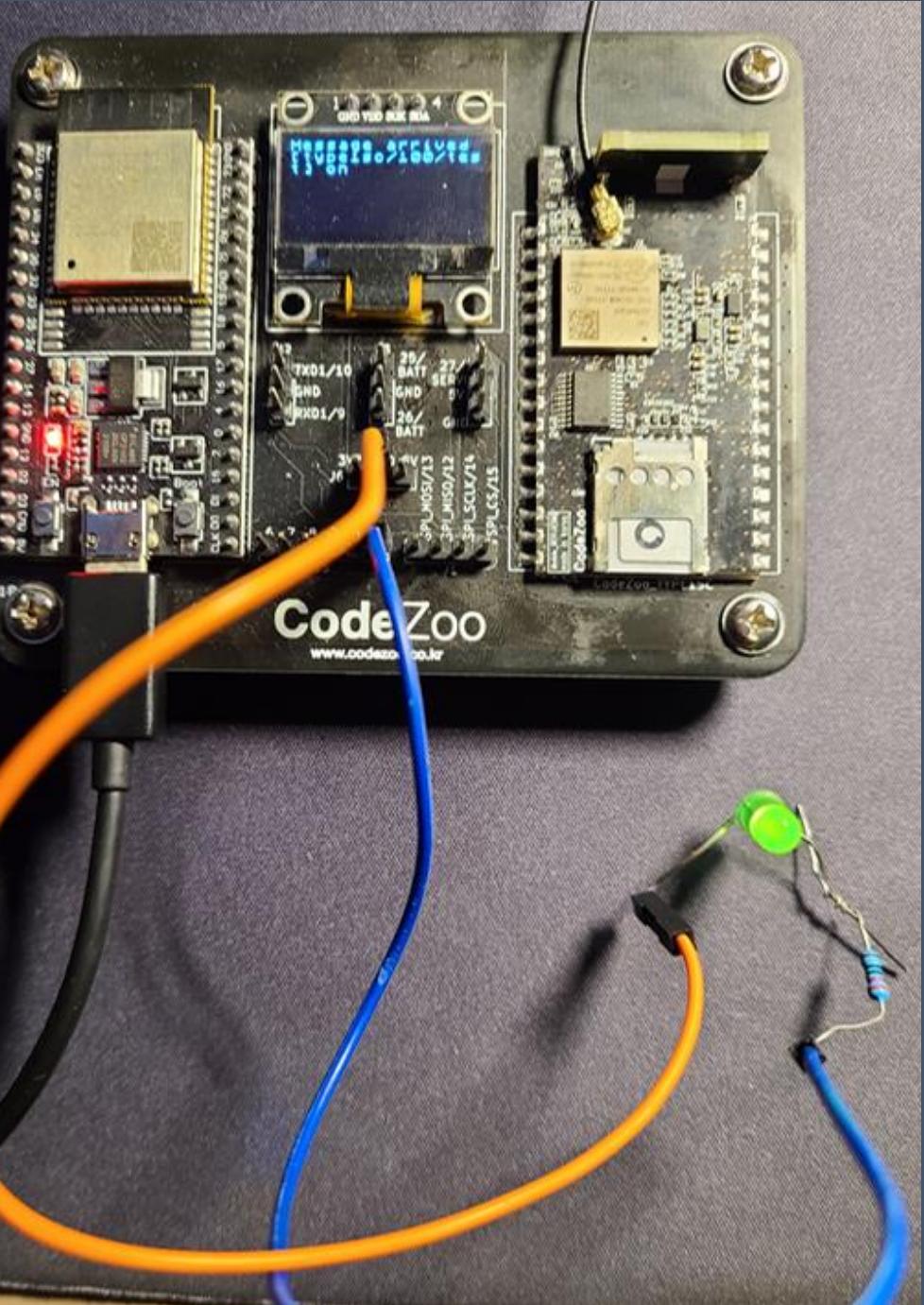
type1sc/100/status :

Report the current status of the device that received the command

Payload

on --> Publish (ESP_IoT board LED remote ON)

off --> Publish (ESP_IoT board LED remote OFF)



Connected

Add publisher Add subscriber

hive - mqtt://broker.hivemq.com:1883

Topic to publish: type1sc/100/test

QoS: 0 - Almost Once

Retain:

Payload Type: Strings / JSON / XML / Characters

e.g: {hello:'world'}

Payload: on

Publish

on
topic:type1sc/100/test, qos:0, retain:false

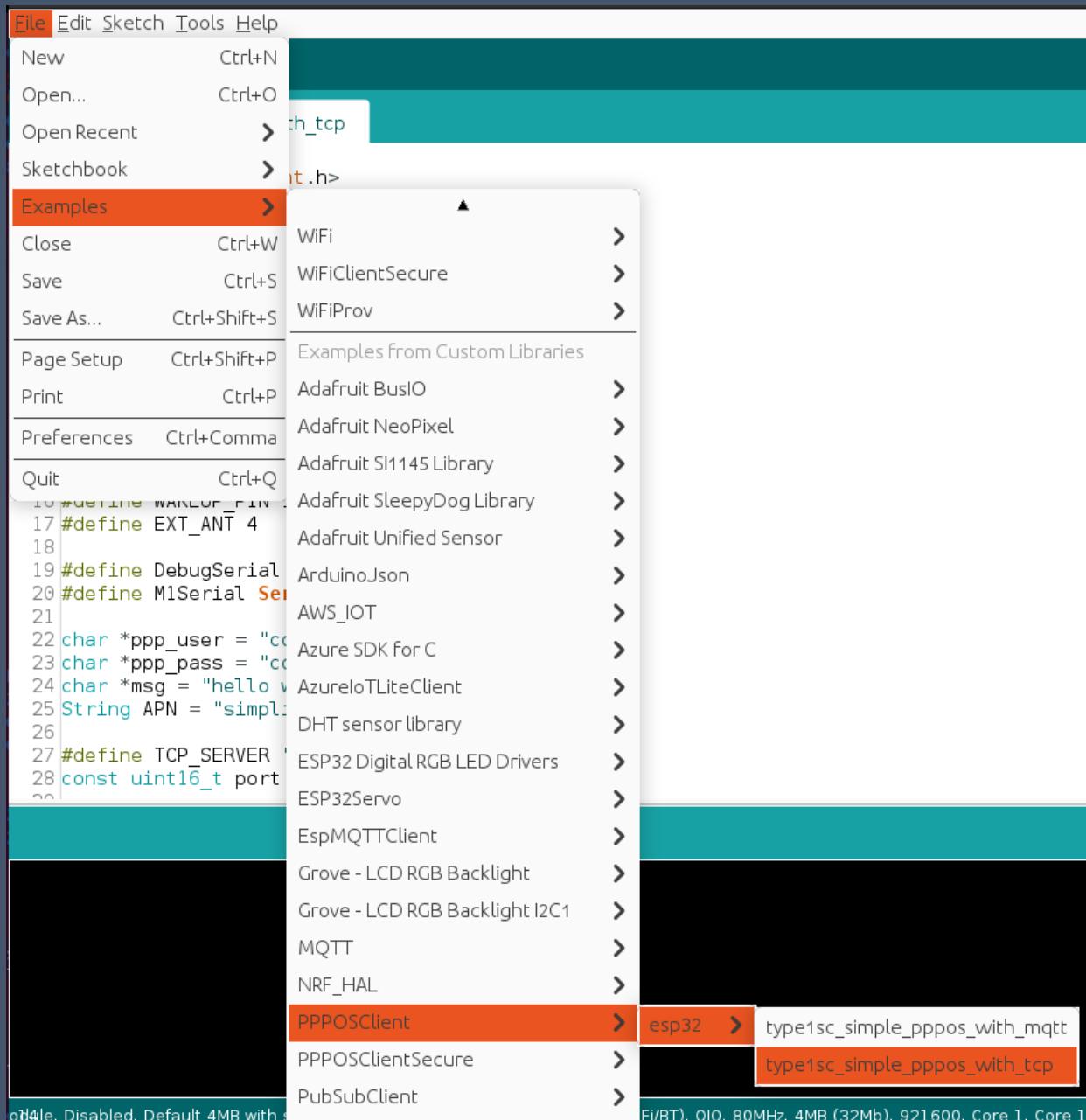
off
topic:type1sc/100/test, qos:0, retain:false

type1sc/100/status

on
qos : 0, retain : false, cmd : publish, dup : false, topic : type1sc/100/status, messageId : , length : 22

off
qos : 0, retain : false, cmd : publish, dup : false, topic : type1sc/100/status, messageId : , length : 23

2) TCP Socket Remote Control



Examples -> PPPOSClient ->
type1sc_simple_pppos_with_tcp

2) TCP Socket Remote Control

Incoming TCP packets from the outside can be received and processed in ppposClient.available() to receive and process them.

```
199 void loop() {  
200   if (PPPOS_isConnected()) {  
201     if (!ppposClient.connected()) {  
202       reconnect();  
203     }  
204  
205     // Data Receive with TCP Socket  
206     // Read all the lines of the reply from Server and print them to Serial  
207     while (ppposClient.available() > 0) {  
208       String line = ppposClient.readStringUntil('\r');  
209       DebugSerial.print(line);  
210       u8x8log.print(line);  
211     }  
212  
213     // Data Transfer with TCP Socket  
214     if (cnt < TEST_CNT) {  
215       ppposClient.println(msg);  
216       delay(1000);  
217     } else if (cnt == TEST_CNT) {  
218       DebugSerial.println("socket Test complete.");  
219       u8x8log.print("socket Test complete.\n");  
220     }  
221     cnt++;  
222   }  
223 }
```

