

LTE CAT.M1을 이용한 AWS 클라우드 서비스 설계

CodeZoo 장병남 대표





수많은 개발 도구를 마우저에서 한번에 검색하세요

1100+ 글로벌 브랜드에서 제공하는
수천 종류의 개발도구를 지원합니다



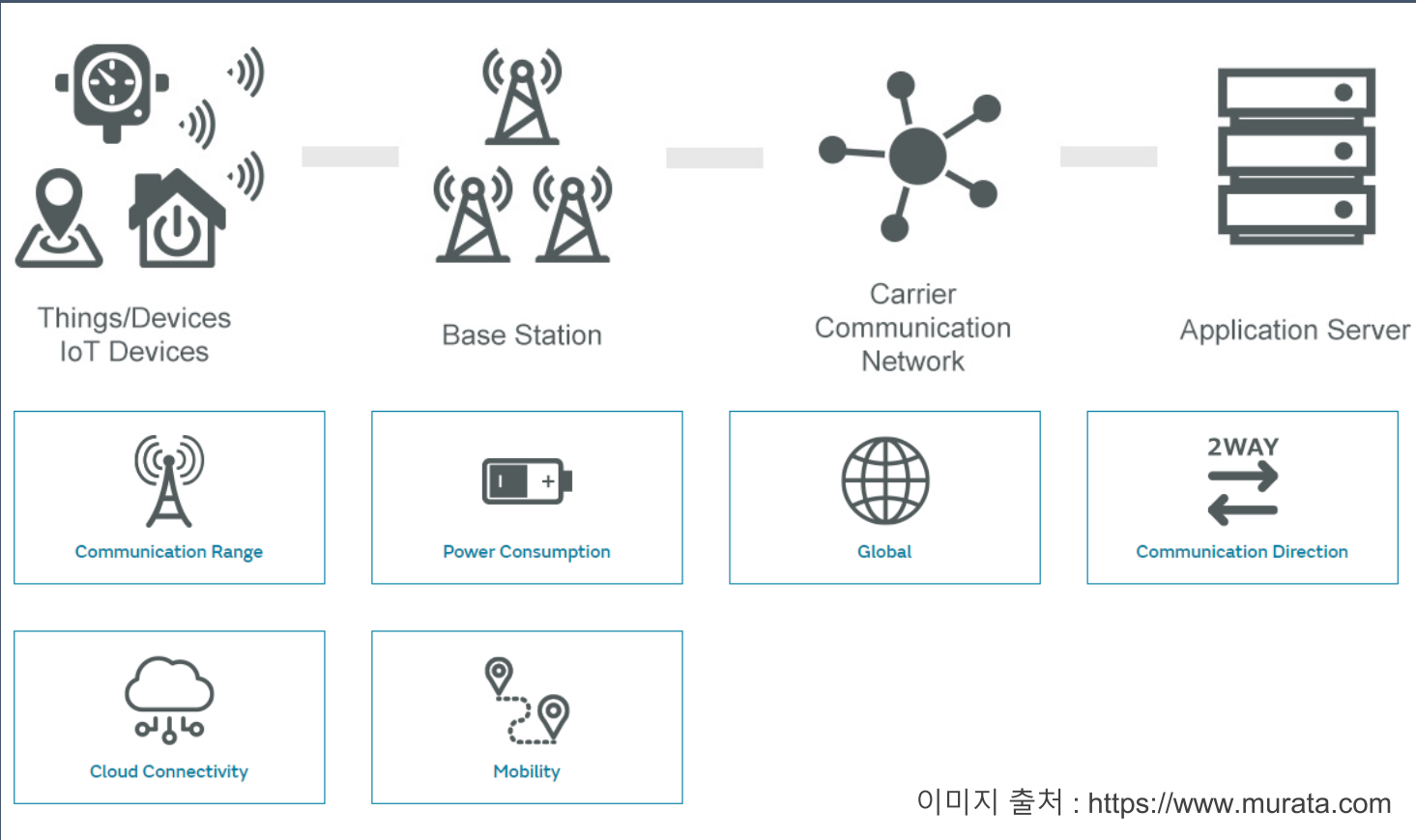
<https://www.mouser.kr/>

주요내용

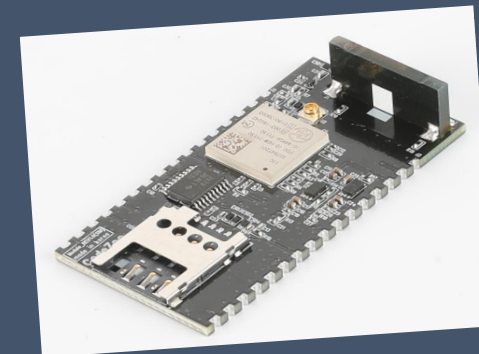
1. LTE Cat.M1 소개
2. 글로벌심 소개 (Vodafone)
3. Cat.M1 하드웨어 구성
4. Cat.M1 동작방식
5. CAT.M1 사용하기 위한 S/W 구성 및 동작테스트
6. CAT.M1과 AWS IoT SiteWise를 활용한 서비스 구현
 - AWS SiteWise
 - 데모구현
 - 실제 구현사례 및 기술 소개 : 아성 협동로봇

1. LTE Cat.M1 소개

LTE 이동통신망 기반의 저전력 광역 통신 기술 표준 LPWA(Low Power Wide Area)



- ☑ 통신 인프라가 없어도 사용가능
- ☑ 초소형 기기 설치 가능
- ☑ 저전력 기기 적합
- ☑ 보안이 강화된 무선통신
- ☑ 저렴한 통신 비용



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

TCP 패킷 송신헤더	230byte
사용자 패킷	xxbyte

TCP 송신 : $230 + xx$ (byte)

TCP 패킷 수신헤더	199byte
사용자 패킷	xxbyte

TCP 수신 : $199 + xx$ (byte)

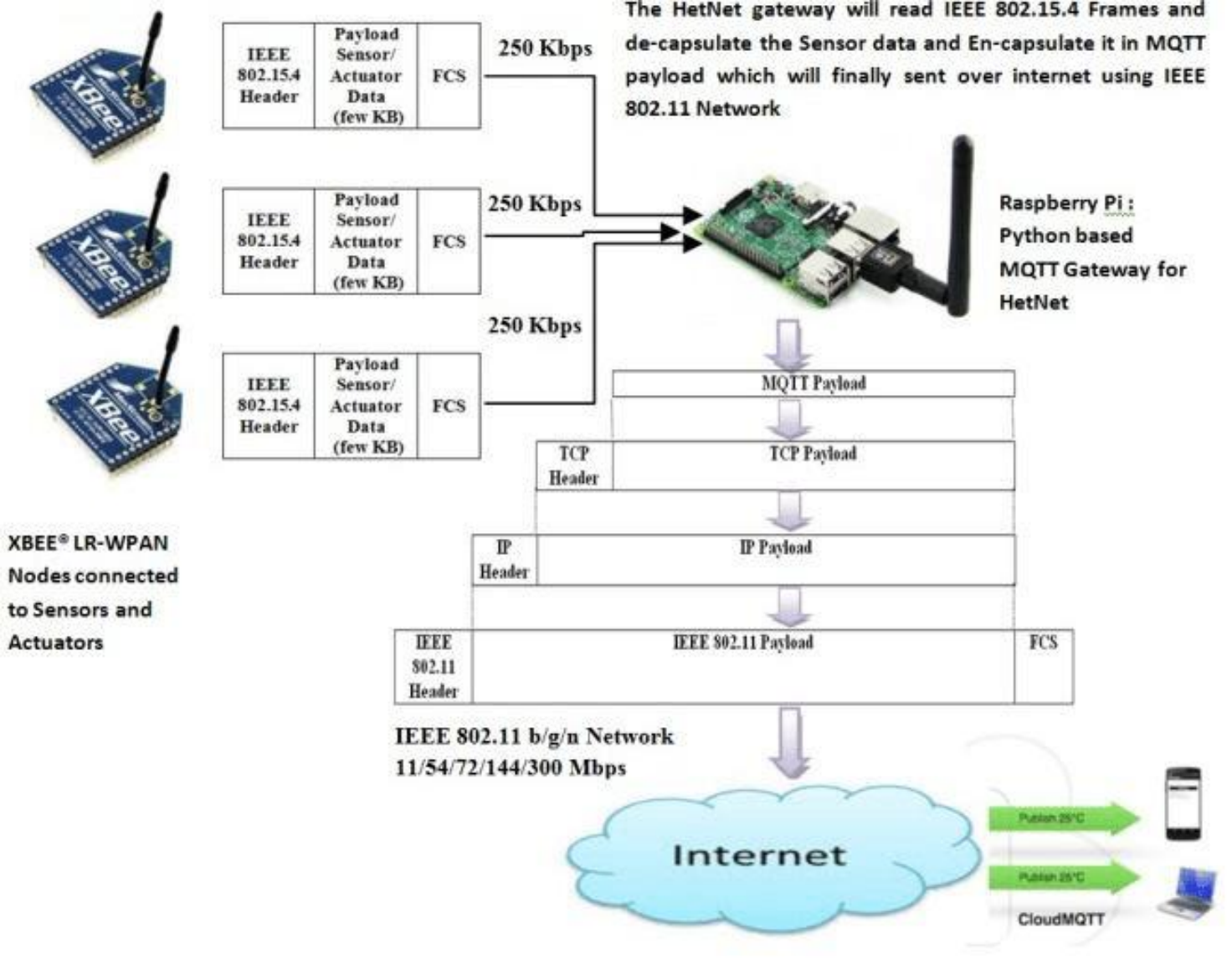
UDP 패킷 송신헤더	40byte
사용자 패킷	xxbyte

UDP 송신 : $40 + xx$ (byte)

UDP 패킷 수신헤더	40byte
사용자 패킷	xxbyte

UDP 수신 : $40 + xx$ (byte)

TCP 소켓으로 1Byte만 전송해도 실제 사용하는 송신데이터 사이즈는
ex) $230 + 1 = 231$ Byte
UDP 소켓으로 8Byte를 수신할 경우 실제 사용하는 수신데이터 사이즈는
ex) $40 + 8 = 48$ Byte



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

데이터 총량제(!)
CATM1은 사용하는 데이터 사이즈 기반으로 과금하기 때문에 월별 데이터 사용량과 송수신 횟수를 미리 계산 후 통신 서비스 운영비용을 예측해야 함 (매우 중요!!!)

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

Raspberry PI 4 구매 : <https://www.mouser.kr/ProductDetail/Seeed-Studio/102110421?qs=sGAepiMZZMv0NwlthflBi7uwz1hZK69BILE7fADipPk%3D>

2. 글로벌심 소개 (Vodafone)

보다폰(Vodafone Group Plc)은 영국의 이동 통신 사업자 중 하나로 잉글랜드 런던에 본사가 있고 매출액상으로 전세계 2위의 이동 통신 사업자입니다. 보다폰은 현재 31개국에서 사업 중이며, 40개국의 통신사와 협력 중이다.

Global IOT SIM?

1. 해외 시장 개척 시 유리

하나의 SIM으로 180개국 이상에서 로밍(roaming)하여 사용할 수 있으므로 해외에서 진행하는 프로젝트에 쉽게 적용할 수 있습니다. (각국 통신사 협의 불필요)

2. 디바이스 망인증 불필요

국내 이동사(MNO) 망인증을 받지 않아도 됩니다
국내에서는 SKT, KT망을 사용합니다.

3. 표준 디바이스 호환성

표준 규격의 모뎀을 탑재하고 carrier-lock 되지 않은 모든 디바이스와 호환됩니다



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.

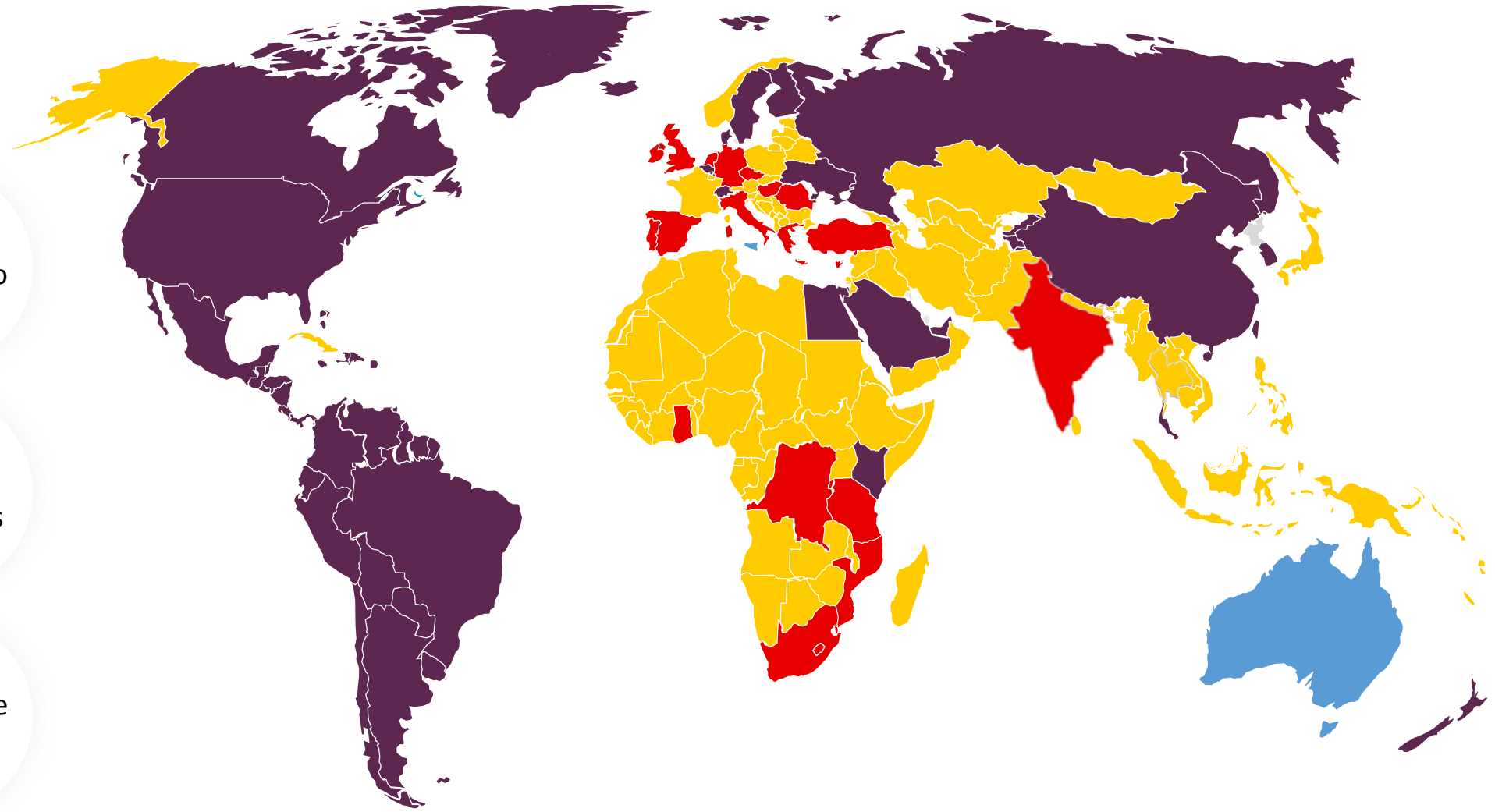
Source: Gartner (December 2019)



– 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

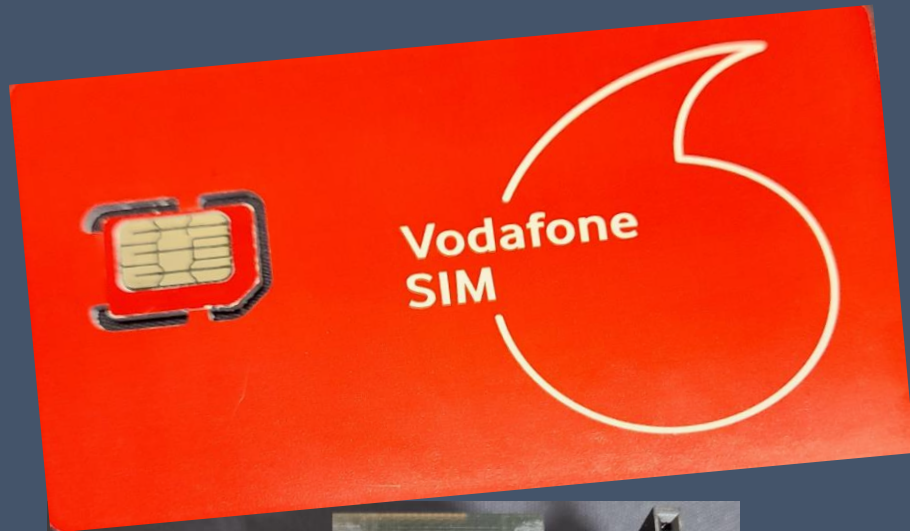
182 countries
+570 networks 2/3/4/5G, LPWA, satellite and fixed capability

Capabilities

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



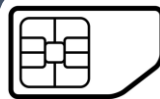
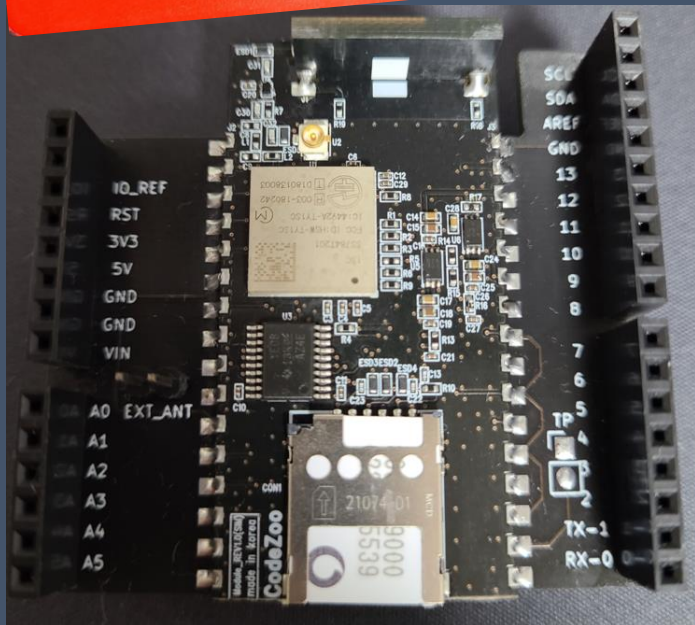
CodeZoo VodaFone Global Sim B2B Service



Global Sim B2B Service

CodeZoo에서 기업과 파트너십을 맺고
Vodafone Global Sim(요금제 포함) 및
IoT 셀룰러 디바이스 연동 기술지원 서비스 제공

문의 : rooney.jang@codezoo.co.kr



Vodafone Global Sim

SimCard, Data Plan 판매

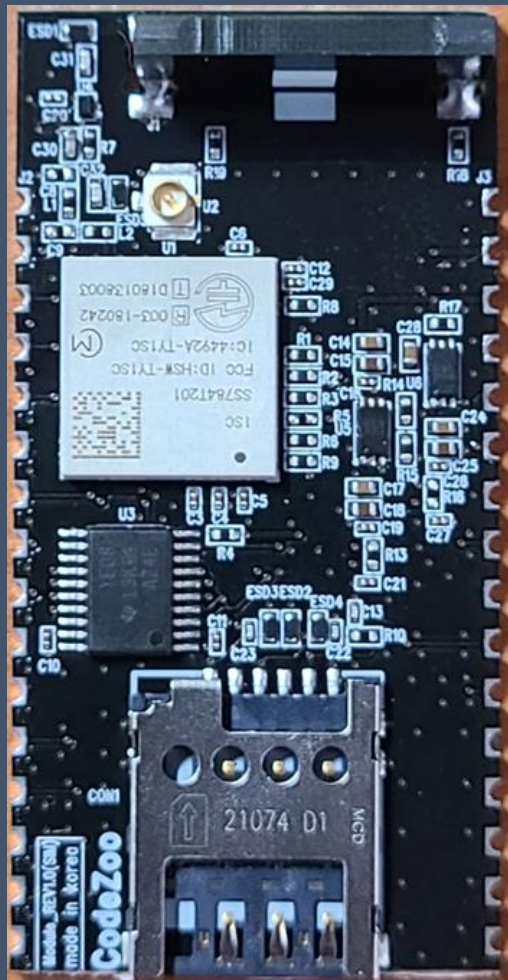


Vodafone Global Sim

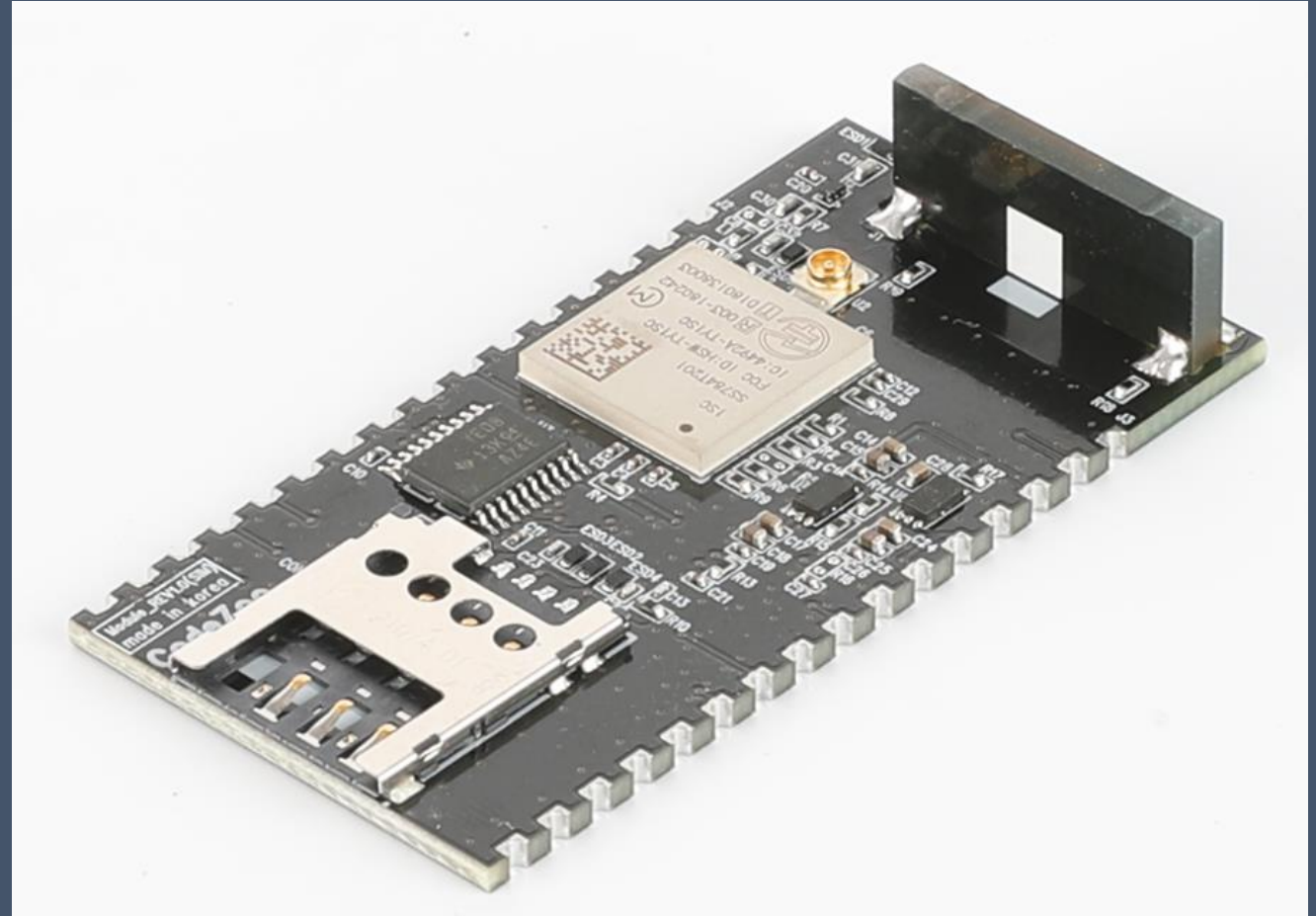
IoT 디바이스 연동
기술지원 서비스

3. Cat.M1 하드웨어 구성

LTE CATM1 내장형 모델
50mm*25mm




GND	GND
VBAT	TXD0
VBAT	RXD0
GND	RTS0
1V8	CTS0
1V8	GND
GND	TXD2
IOREF	RXD2
GND	RTS2
PWR_EN	CTS2
P_LED	GND
RESET	TXD1
RI	RXD1
WAKE	RTS1
TX_LED	CTS1
ANT_CON	GND

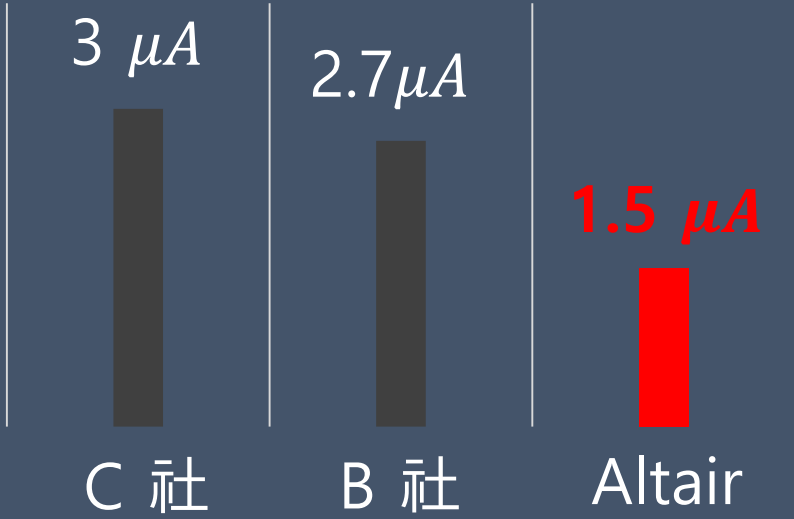


Murata Cat.M1 Module Type1SC 특징

사이즈

Type 1SC	
Size	11.1 x 11.4 x 1.4mm
Chipset	<u>SONY (ALT1250)</u>
PSM 소비전류	<u>1.5uA</u>

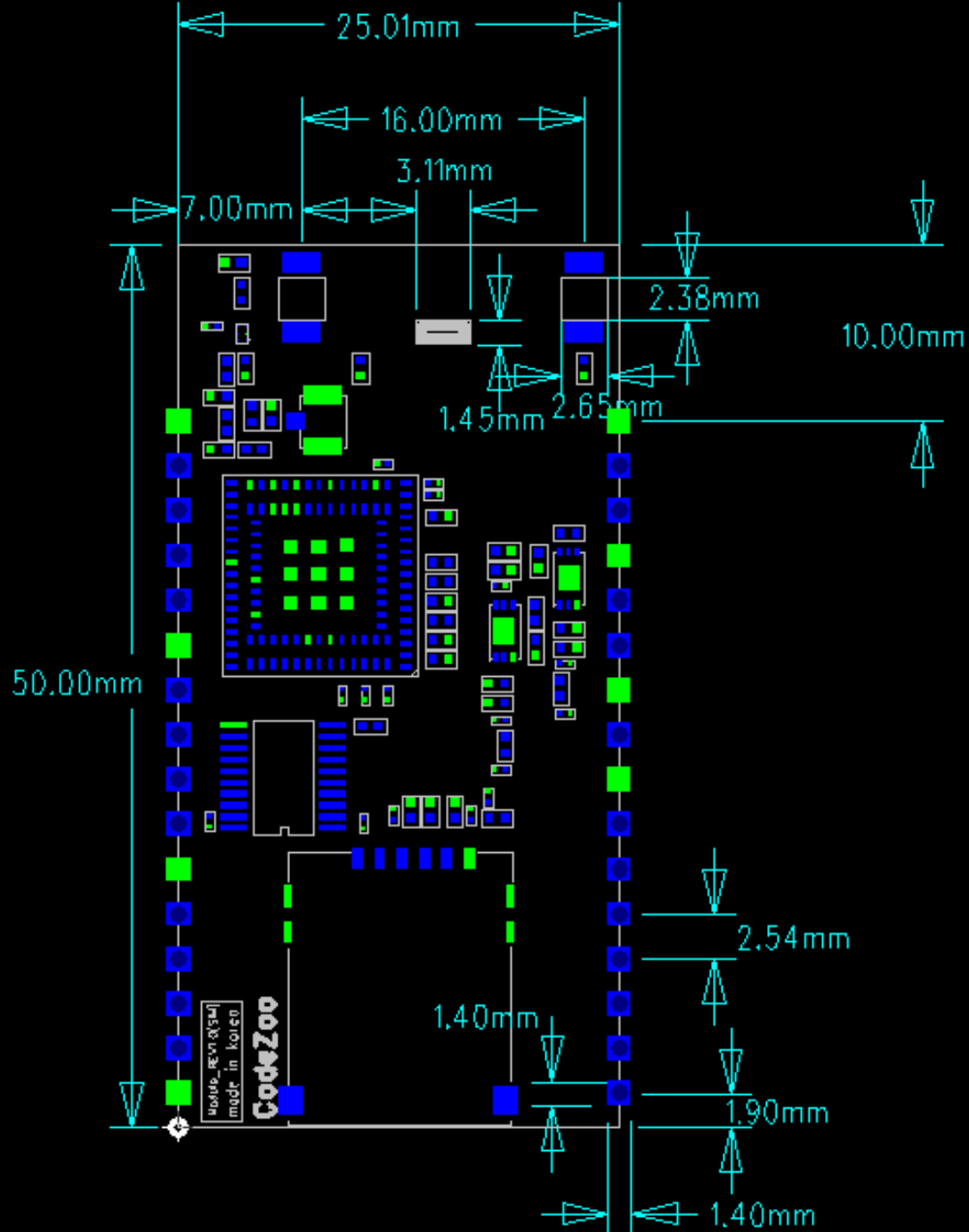
소비전류



⇒ 저전력
⇒ 긴 배터리 수명!



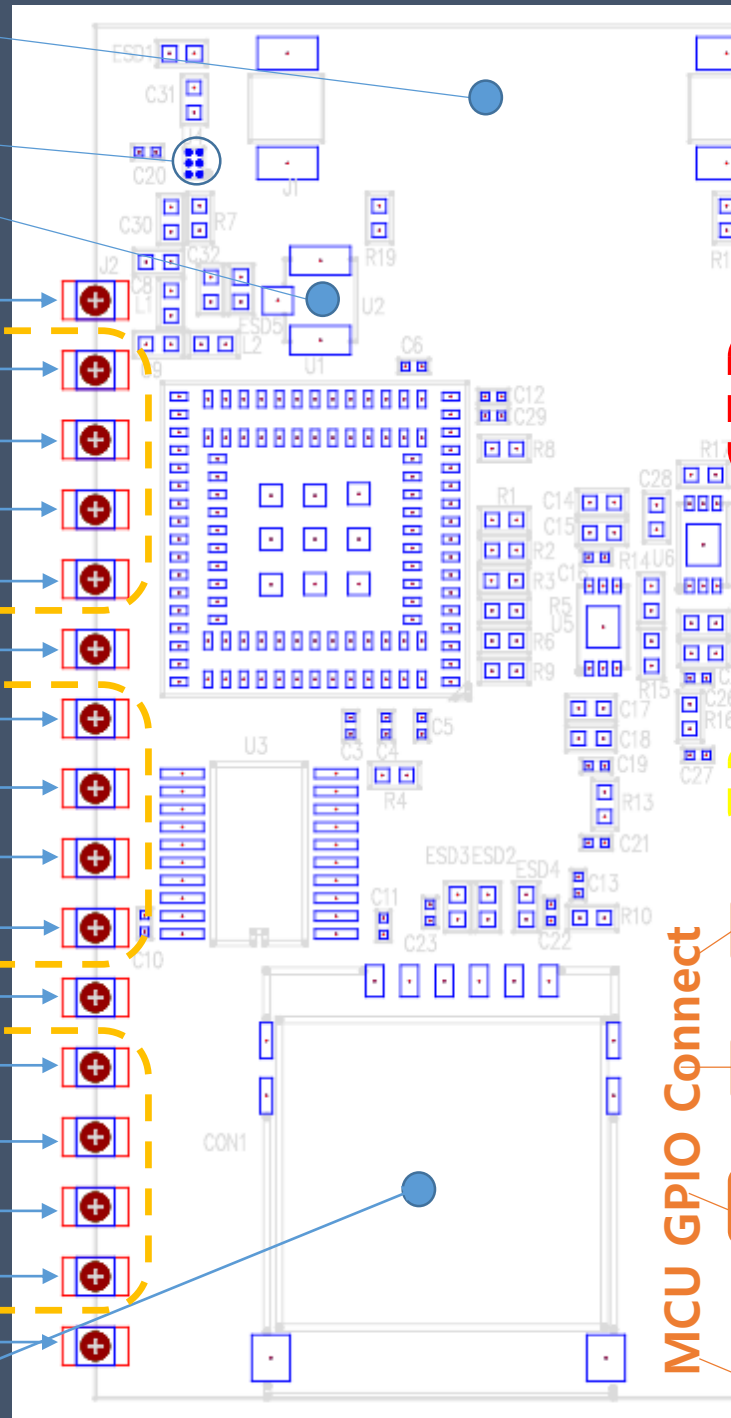
Murata Manufacturing Co., Ltd.



TYPE1SC_32_PAD

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

- LTE_Internal_ANTENNA
- RF_Switch_IC
- LTE_External_ANTENNA
- GND
- UART0_TX
- UART0_RX
- UART0_RTS
- UART0_CTS
- GND
- UART2_TX
- UART2_RX
- UART2_RTS
- UART2_CTS
- GND
- UART1_TX
- UART1_RX
- UART1_RTS
- UART1_CTS
- GND
- Micro Sim Socket



MCU GPIO Connect

- GND
- VBAT_IN(3V3, 5V) : input 2.4~5.5V
- GND
- VCC_1V8_OUT
- GND
- IO_REF : input MCU I/O Voltage 1.65 ~5.5V
- GND
- LDO_EN
- PMU_EXT_ALARM(If Alive, OUT 1V8)
- MRESET (Modem Reset)
- HWAKEUP(Host Wake Up)
- MWAKEUP (Modem Wake Up)
- DEVICE_TX_INDICATOR
(If Alive, OUT 1V8)
- EXT_ANT_Select
(If Pull-Up, Using EXT Antenna)

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) [단위: mm]	Width(25.2)*Height(50.0)*Depth(2.8[with Antenna 10.0])
7. 기능용도 (Function-Use)	LTE통신 모듈
8. 전원 타입 (Power Supply Type)	2.4V ~ 5.5V
9. 안테나 타입 (Antenna Type)	내외장 겸용 (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

4. Cat.M1 동작방식

AT Commands 방식

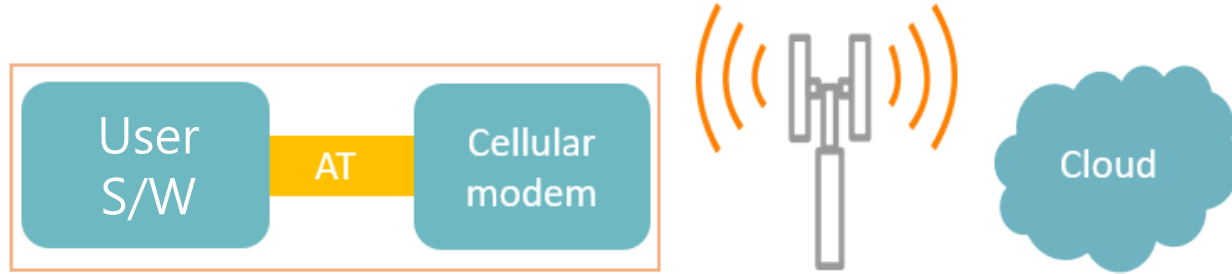


Fig 1. CellularInterface using raw AT



AT Commands' Classification

PPP 통신 방식

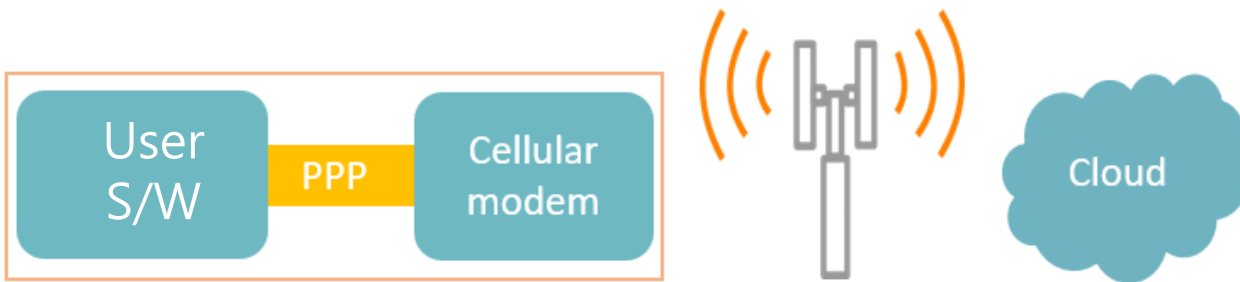
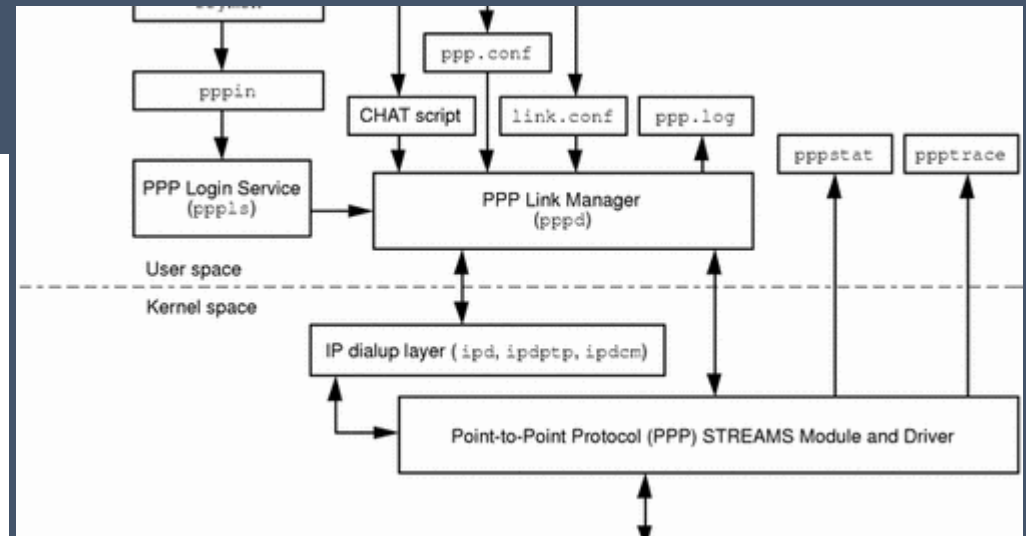


Fig 2. CellularInterface using PPP



4. Cat.M1 동작방식

AT Commands 방식

QCOM_V1.0

About

COM Port Setting

COM Port: 8 Baudrate: 115200 StopBits: 1 Parity: None

ByteSize: 8 Flow Control: No Ctrl Flow **Close Port**

AT

OK

AT+COPS?

+COPS: 0,0,"DATA ONLY",7

OK

AT+CEREG?

+CEREG: 0,5

OK

[2022-02-15_23:18:48] Open COM Port Success

Operation

DTR RTS View File Show Time

Input String: HEX String Show In HEX Send With Enter

C:\Users\jmbmas\Downloads\mosquito.org.crt

Save Log Z:\W02_PROJECTS\W70_NB_10TW5_Driver\QUECTEL\Tools\QCOM_V1.0\QCOM.LL

DUT Connection

COM1

Comport connection is OK!!!

Information

제조사 : Murata Manufacturing Co., Ltd.
모델명 : LBAD00C1SC

S/W Version

S/W Version :
Build Date : N

Network

Phone Number(Type) :
IMEI :
ICCID(Type) :
IMSI :

Network Information


MS Time :
USIM Status :
Network 상태 : I

Notification

registered, home network

Comport Log

```
[<<] at+cnum+CME ERROR: SIM PIN required ^
[>>] at+cgsn
[<<] at+cgsn
[<<] 351521100330281
[<<] OK
[>>] at%ccid
[<<] at%ccid
[<<] %CCID: 8982300420000006005
[<<] OK
[>>] AT+CMGF=0
[<<] AT+CMGF=0
[<<] OK
[>>] AT+CMGL
[<<] AT+CMGL
[<<] OK
[<<]
[>>] at+cpin="0000"
[<<] at+cpin="0000"
[<<] OK
[<<] %NOTIFYEV:"SIMSTATE",3
[<<] %NOTIFYEV:"RRCSTATE",2
[<<] %NOTIFYEV:"EMMSTATE",1
[<<] %NOTIFYEV:"EMMSTATE",4
[>>] at+cpin?
[>>] at+cnum
[>>] at+cnum
[>>] at+dck="SC",0,"0000"
[<<] at+dck="SC",0,"0000"
[<<] OK
[>>] at+cpin?
[>>] at+cnum
[>>] at+cnum
```

 codezoo-ltd Unix Timestamp Conversion Example ...	6656ac1	3 days ago	🕒 22 commits
examples	Unix Timestamp Conversion Example		3 days ago
manual	Arduino HandsOn page 17 modified		6 months ago
schematics_dimension	Manual, Schematics & Dimension file upload		6 months ago
.gitignore	Initial commit		8 months ago
Countdown.h	TYPE1SC Initial Commit		8 months ago
LICENSE	Initial commit		8 months ago
README.md	Initial commit		8 months ago
TYPE1SC.cpp	Type1SC HardReset Added		3 days ago
TYPE1SC.h	dos2unix clang-format & ESP32 Feature commit		6 months ago

README.md

TYPE1SC

Murata TYPE1SC Cat.M1 Arduino Library & example

About

Murata TYPE1SC Cat.M1 Arduino Library & example

- Readme
- MIT License
- 1 star
- 1 watching
- 0 forks



Releases

No releases published
[Create a new release](#)

Packages

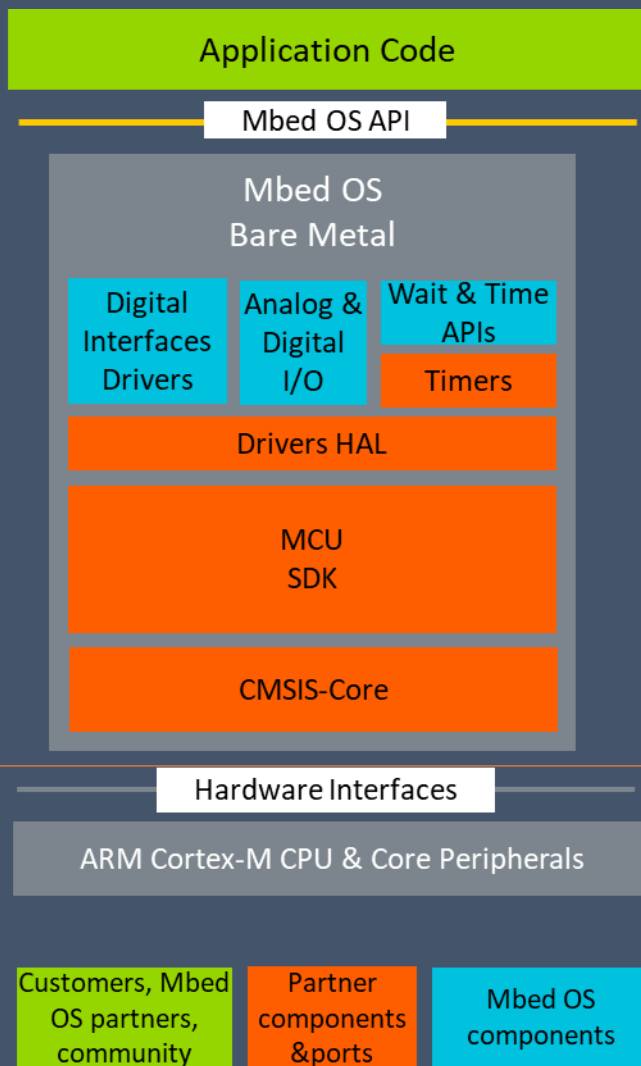
No packages published
[Publish your first package](#)

Contributors 2

-  codezoo-ltd CodeZoo
-  jbmater50

Open Source Library to Firmware Porting

✓ 오픈소스 하드웨어를 이용해서 빠르게 PoC 진행하여 평가 후 본 프로젝트 진행



Library Logic
AT Command
Parser, Logic Unit

Library Hardware
abstraction

Uart Send/Recv
Timer, Delay

Library
Hardware IP

Library Logic + Firmware Code

HAL_UART_Transmit()

HAL_UART_Transmit

Function name HAL_StatusTypeDef HAL_UART_Transmit (UART_HandleTypeDef * huart, uint8_t * pData, uint16_t Size, uint32_t Timeout)

Function description Send an amount of data in blocking mode.

Parameters

- **huart**: UART handle.
- **pData**: Pointer to data buffer.
- **Size**: Amount of data to be sent.
- **Timeout**: Timeout duration.

Return values

- **HAL**: status

Notes

- When UART parity is not enabled (PCE = 0), and Word Length is configured to 9 bits (M1-M0 = 01), address of user data buffer containing data to be sent, should be aligned on a half word frontier (16 bits) (as sent data will be handled using u16 pointer cast). Depending on compilation chain, use of specific alignment compilation directives or pragmas might be required to ensure proper alignment for pData.

HAL_UART_Receive_IT

Function name HAL_StatusTypeDef HAL_UART_Receive_IT (UART_HandleTypeDef * huart, uint8_t * pData, uint16_t Size)

Function description Receive an amount of data in interrupt mode.

Parameters

- **huart**: UART handle.
- **pData**: pointer to data buffer.
- **Size**: amount of data to be received.

Return values

- **HAL**: status

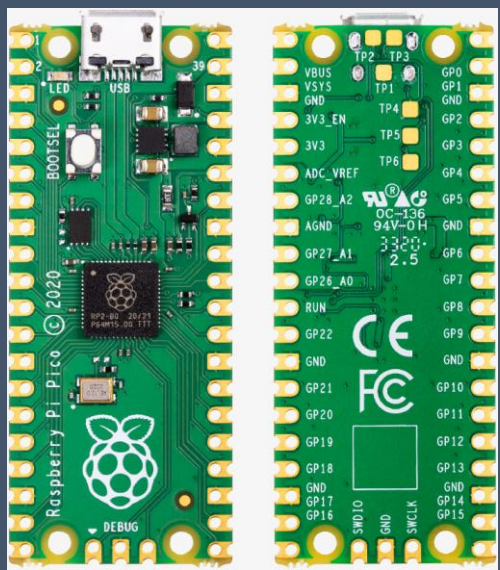
Notes

- When UART parity is not enabled (PCE = 0), and Word Length is configured to 9 bits (M1-M0 = 01), address of user data buffer for storing data to be received, should be aligned on a half word frontier (16 bits) (as received data will be handled using u16 pointer cast). Depending on compilation chain, use of specific alignment compilation directives or pragmas might be required to ensure proper alignment for pData.



AT Commands 방식

오픈소스 CATM1 Library (<https://github.com/codezoo-ltd/TYPE1SC>)



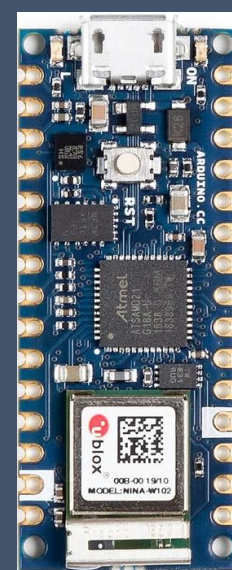
RaspberryPi PICO



ESP32 DevKit



Arduino NANO Board (Every, 33 IoT, 33 BLE Sense, RP2040 Connect)



CodeZoo CATM1 Library (Commercial License)

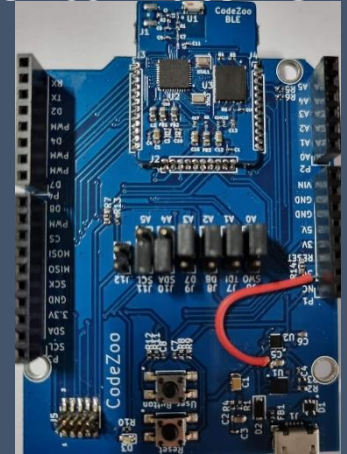
STM32G4



Renesas RA2L

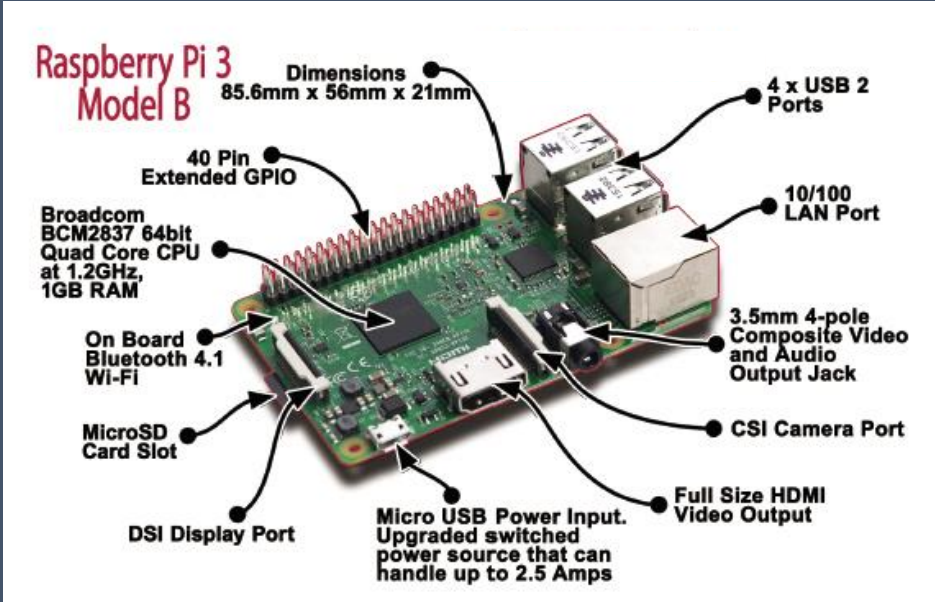


Silicon Labs BG22



NXP I.MX RT1010





LTE-CATM1 내장형 모뎀을 리눅스에서 사용하기

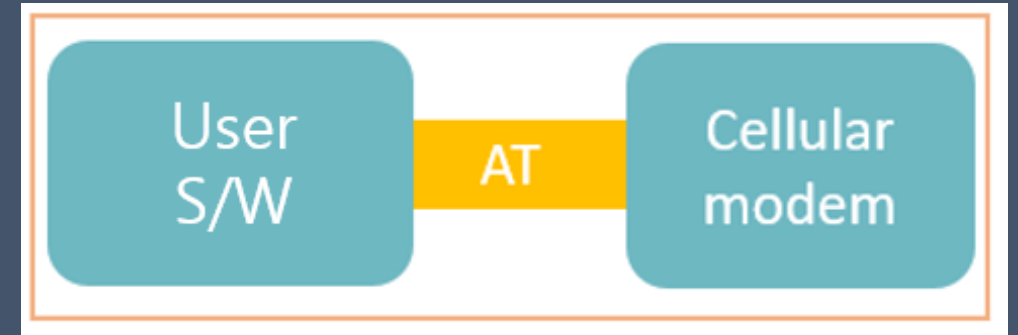
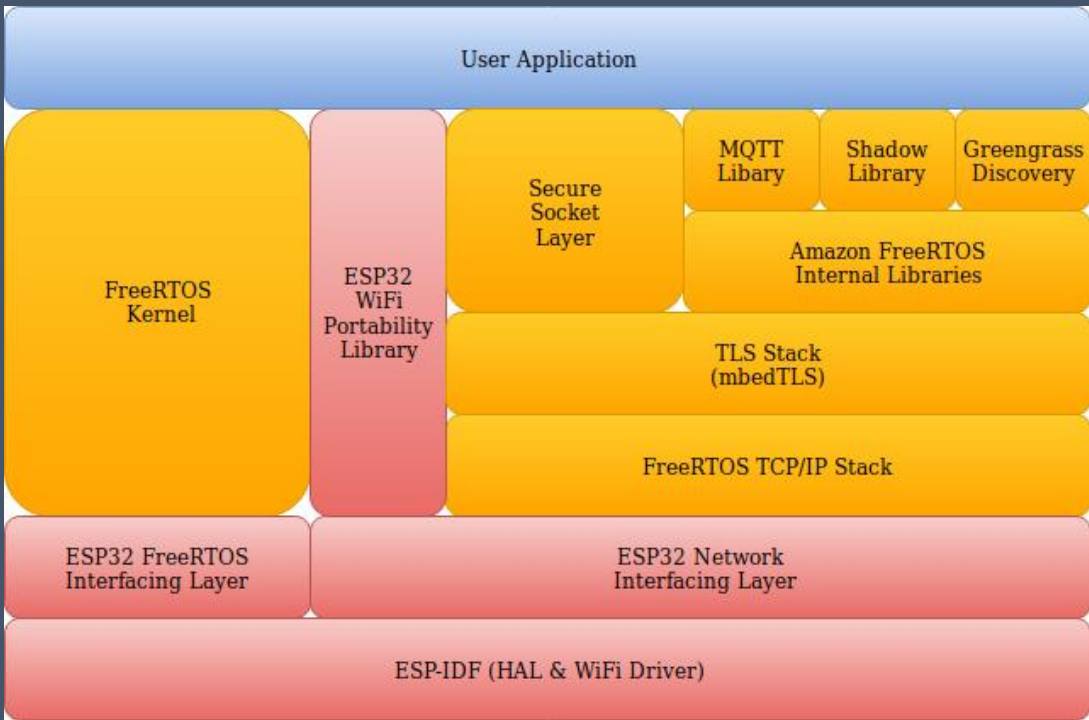
카테고리 없음 · 2021. 10. 21. 01:43

며칠 전 코드주 LTE-CATM1을 구매하신 개발자분에게 질문받았던 내용입니다. 리눅스 시스템에서 PPP로 동작시키고 싶은데 어떻게 하면 좋을지 방법을 알려 달...

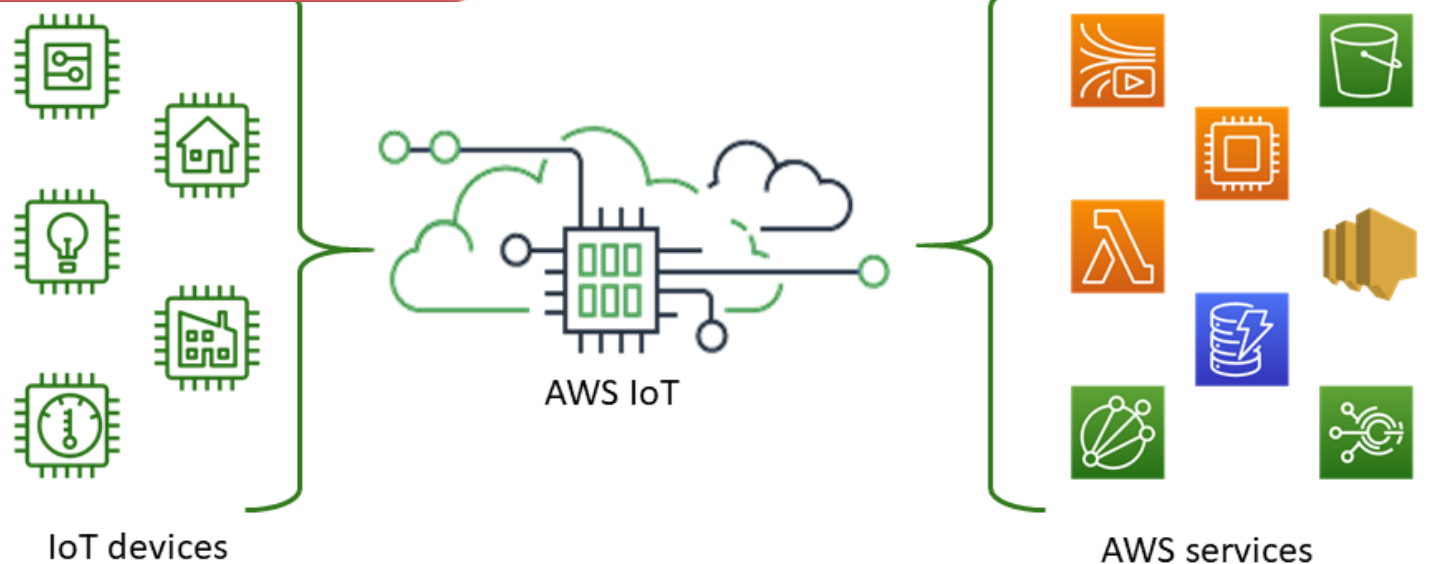
[Read More](#)

Linux PPP설정 방법
<https://codezoo.tistory.com/15>

AT Commands 방식 AWS-IoT 통신



ESP32 AWS-IoT S/W Architecture



<Security Setup>

```
at%certcfg="ADD",9,"rootCA.pem",,"cert.pem.crt","private.pem.key"
```

<AWS-IoT Setup>

```
at%awsiotcfg="CONN","a1[REDACTED]st2.amazonaws.com",9,"M[REDACTED]e01"
```

```
at%awsiotcfg="PROTOCOL",1200,0
```

```
at%awsiotev="all",1
```

<AWS-IoT Connect>

```
at%awsiotcmd="connect"
```

<AWS-IoT Test>

```
at%awsiotcmd="subscribe","sdkTest/sub"
```

```
at%awsiotcmd="publish","sdkTest/sub","CodeZoo Now Ready!!!"
```

```
at%awsiotcmd="UNSUBSCRIBE","sdkTest/sub"
```

<AWS-IoT Disconnect>

```
at%awsiotcmd="DISCONNECT"
```

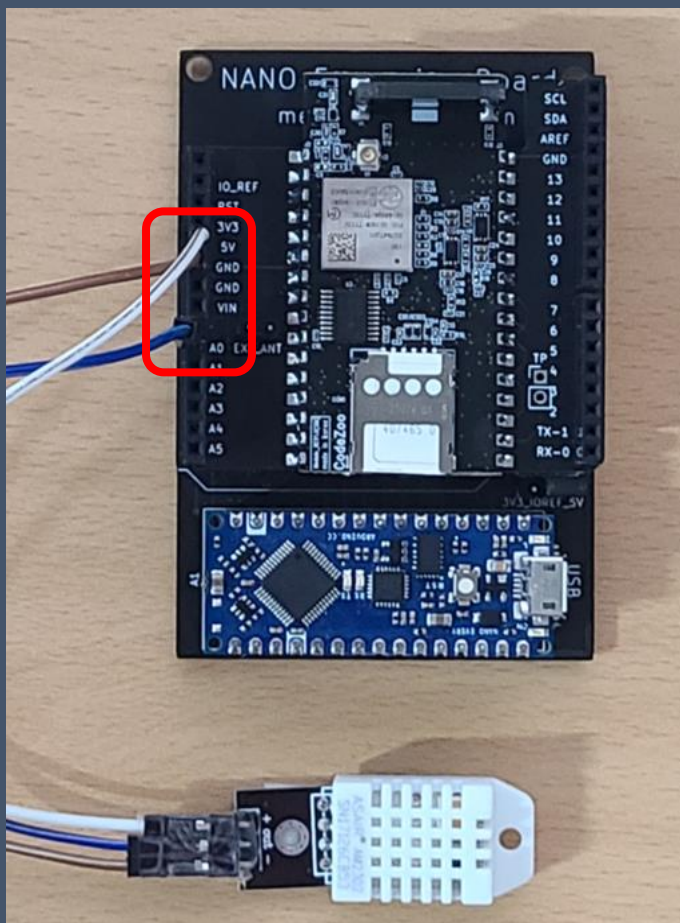


5. CAT.M1 사용하기 위한 S/W 구성 및 동작테스트

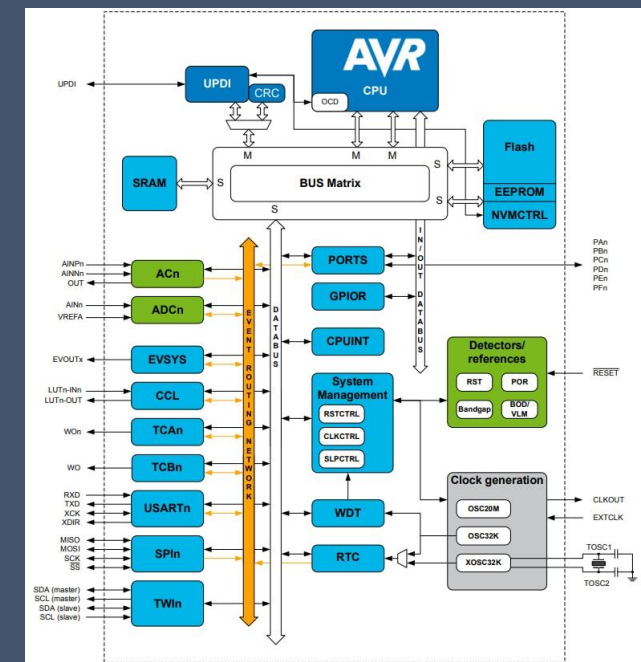
- 실습을 위한 하드웨어 구성

1. Arduino NANO Every (ATmega4809, 8bit/20Mhz, 6KB SRAM, 48KB Flash)
2. DHT22 온습도센서

VCC : 아두이노 쉴드 5V 연결
 DATA : 아두이노 쉴드 A0 연결
 GND : 아두이노 쉴드 GND 연결



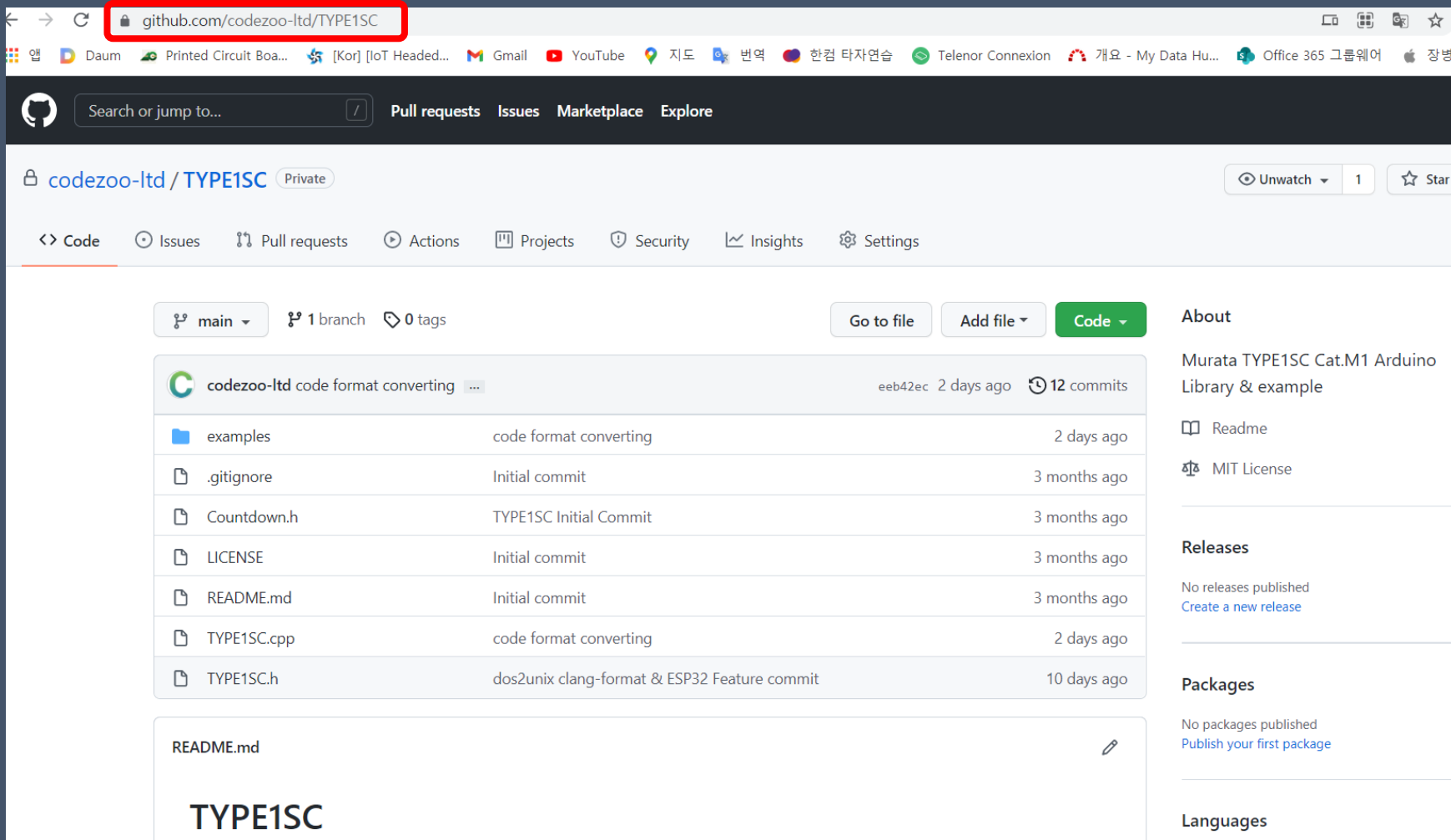
<p>Dht22 Dht 22 디지털 온도 및 ... korean.alibaba.com</p>	<table border="1"> <thead> <tr> <th colspan="2">DHT22 pins</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>VCC</td> </tr> <tr> <td>2</td> <td>DATA</td> </tr> <tr> <td>3</td> <td>NC</td> </tr> <tr> <td>4</td> <td>GND</td> </tr> </tbody> </table> <p>Arduino UNO에서 dht-22 온도 / 습도... webnautes.tistory.com</p>	DHT22 pins		1	VCC	2	DATA	3	NC	4	GND	<p>1 : VCC 2 : Data 3 : NC 4 : GND</p> <p>DHT22/AM230x/RHT0x Temperature... espruno.com</p>	<p>Eiechip® AM2302 DHT22 arduino용 디... ko.aliexpress.com</p>	<p>디지털 온습도 vctec.co.kr</p>
DHT22 pins														
1	VCC													
2	DATA													
3	NC													
4	GND													
		<p>Application typical</p> <p>1-Vcc 2-Data 3-NC 4-GND</p>												



- ✓ **Arduino Nano Every 구매**
<https://www.mouser.kr/ProductDetail/Arduino/ABX00033?qs=PzGy0jfpSMueXyfBfl2XSA%3D%3D>
- ✓ **DHT 22 Sensor 구매**
<https://www.mouser.kr/ProductDetail/Adafruit/385?qs=sGAEpiMZZMsKEdP9slC0YV3SHmTsbCQ6nxczzvjM7GQ%3D>

- LTE Cat.M1 라이브러리 설치

웹 브라우저를 열고 github.com/codezoo-ltd/TYPE1SC 입력

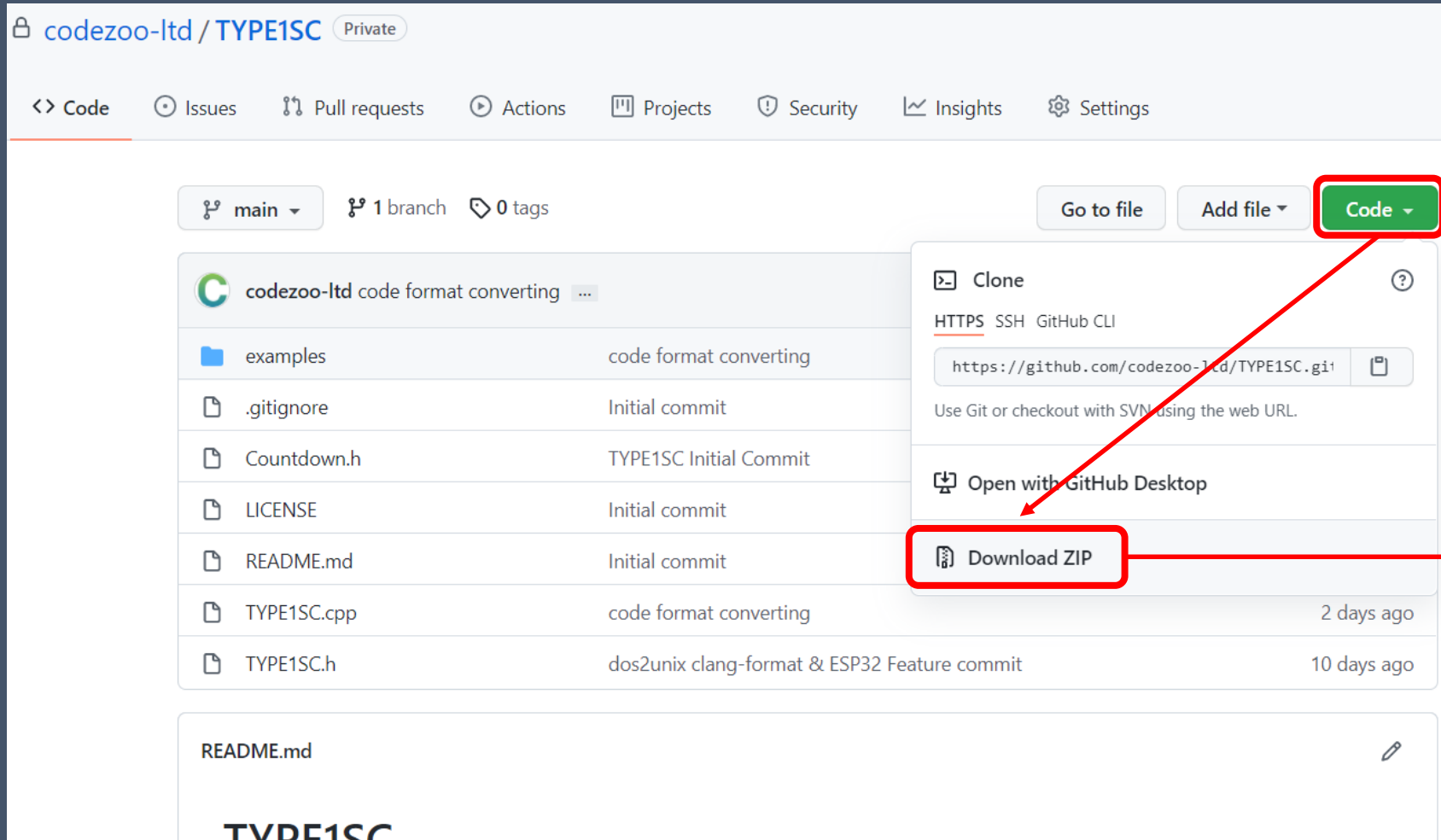


The screenshot shows a web browser displaying the GitHub repository page for `codezoo-ltd/TYPE1SC`. The address bar shows the URL `github.com/codezoo-ltd/TYPE1SC` highlighted in red. The repository is private and has 1 branch and 0 tags. The main branch is `main`. The repository contains the following files and folders:

File/Folder	Commit Message	Commit Date
examples	code format converting	2 days ago
.gitignore	Initial commit	3 months ago
Countdown.h	TYPE1SC Initial Commit	3 months ago
LICENSE	Initial commit	3 months ago
README.md	Initial commit	3 months ago
TYPE1SC.cpp	code format converting	2 days ago
TYPE1SC.h	dos2unix clang-format & ESP32 Feature commit	10 days ago

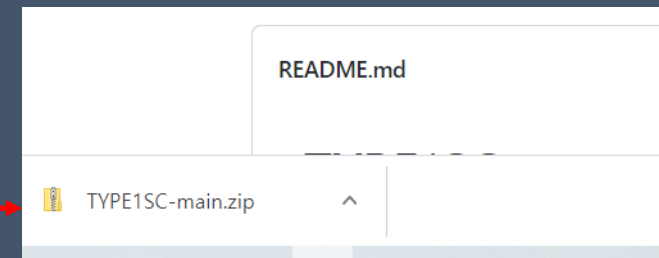
The README.md file is visible at the bottom of the page, with the title `TYPE1SC`.

녹색 Code 버튼 -> Download ZIP -> 라이브러리 다운로드 (다운로드 된 경로 확인!!)



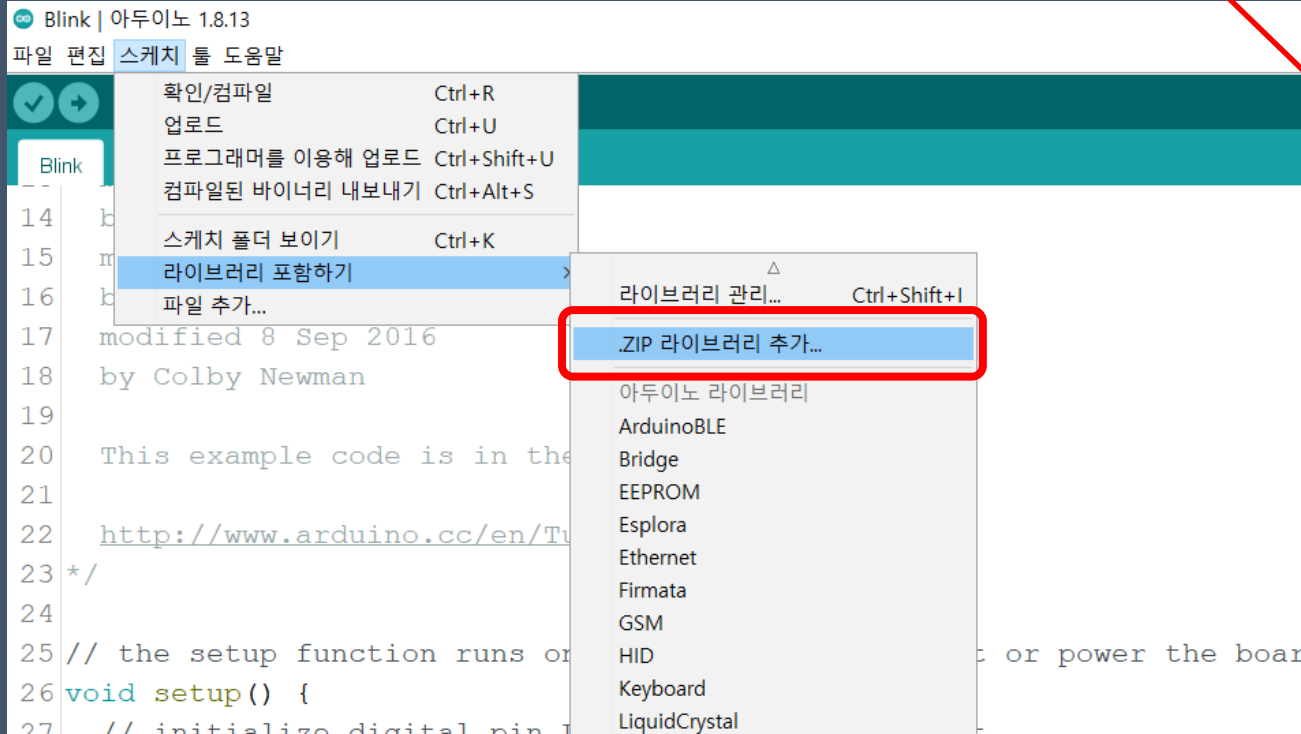
The screenshot shows the GitHub interface for the repository 'codezoo-ltd / TYPE1SC'. The 'Code' button is highlighted with a red box. A dropdown menu is open, showing options for cloning and downloading. The 'Download ZIP' option is also highlighted with a red box. A red arrow points from the 'Download ZIP' option to the 'TYPE1SC-main.zip' file in the file explorer on the right.

File Name	Commit Message	Last Commit
examples	code format converting	
.gitignore	Initial commit	
Countdown.h	TYPE1SC Initial Commit	
LICENSE	Initial commit	
README.md	Initial commit	
TYPE1SC.cpp	code format converting	2 days ago
TYPE1SC.h	dos2unix clang-format & ESP32 Feature commit	10 days ago

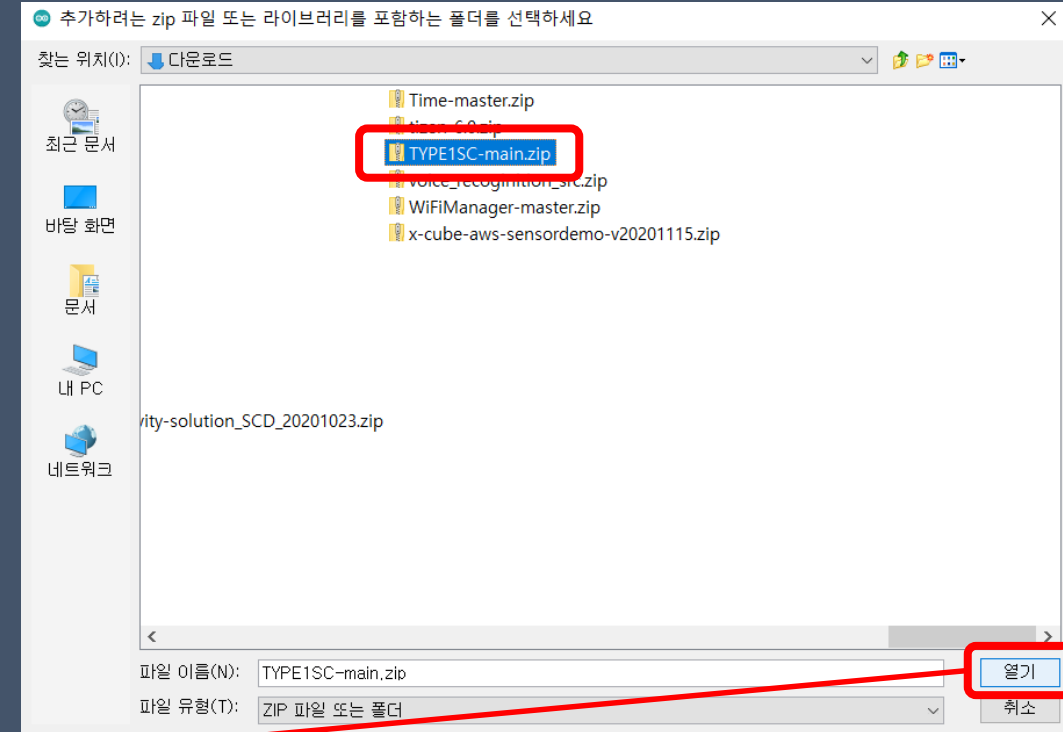


The screenshot shows a file explorer window with a 'README.md' file selected. Below it, the file 'TYPE1SC-main.zip' is visible, indicating the successful download of the repository's source code.

스케치 -> 라이브러리 포함하기 -> .ZIP 라이브러리 추가...



다운로드 된 디렉토리 -> TYPE1SC-main.zip -> 열기



라이브러리가 추가되었습니다. "라이브러리 포함하기" 메뉴를 확인하세요

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

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

avrdude: jtagmkII_initialize(): Cannot locate "flash" and "boot" memories in description

예제 실습을 위한 센서 라이브러리 설치

DHT22 온습도센서 라이브러리 설치

스케치 -> 라이브러리 포함하기 -> 라이브러리 관리
-> DHT sensor library 설치

```
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...
라이브러리 관리... Ctrl+Shift+I
ZIP 라이브러리 추가...
아두이노 라이브러리
ArduinoBLE
Bridge
EEPROM
Esplora
Ethernet
Firmata
GSM
HID
Keyboard
LiquidCrystal
Mouse
RTCZero
Robot Control
```

라이브러리 매니저

타입 토픽

DHT sensor library
by Adafruit 버전 1.4.2 **INSTALLED**
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 library 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

1) APN 설정

- APN (Access Point Name)

이동통신사의 셀룰러 네트워크와 인터넷을 연결하는 게이트웨이. APN설정을 사용하여 이동통신사에서 IP주소를 생성합니다.

결론, LTE CAT.M1으로 네트워크 통신을 하기 위해서는 반드시 설정해 주어야 합니다.
설정 후 CAT.M1 모뎀 내부 파일시스템에 APN 값 저장 됨. 이후에는 APN 변경시에만 사용

2) Basic Test

CAT.M1 모뎀을 초기화 하고 통신사 망에 접속 후 아래 정보를 가져 옵니다.

1. CIMI : IMSI (International Mobile Subscriber Identity) 값을 SIM 에서 가져 옵니다
2. IMEI : Internaitional Mobile Equipment Identity 모뎀의 고유값(phone's fingerprint)을 가져 옵니다.
3. CGMR : 모뎀 펌웨어 버전
4. Time : 현재시간을 가져 옵니다 (기지국 등록시 획득)
5. RSSI, RSRP, RSRQ, SINR, Cell ID, TX Power
무선통신 감도, 신호세기, 기지국망 ID, 전송 세기를 가져 옵니다.

3) HTTP Test

CAT.M1 모뎀 내부 통신스택을 이용해서 아두이노 보드에서 측정된 온도, 습도 데이터를 HTTP 프로토콜로 Thingspeak.com 서버에 업로드 해서 PC, 스마트폰에서 실시간 모니터링 합니다.

1. Thingspeak.com 을 사용하기 위해 가입 및 설정 합니다.

2. CAT.M1 모뎀 내부 TCP 소켓으로 Thingspeak.com과 연결하고 아두이노 보드에서 DHT22 센서로 온도, 습도 데이터를 측정된 후 HTTP 프로토콜을 만들어서 Thingspeak.com에 데이터를 업로드 합니다.

ThingSpeak 란?

- IoT 클라우드 서비스 중 하나로 MathWorks 에서 만든 서비스
- Thingspeak 시작하려면, MathWorks 가입해야 함
- www.thingspeak.com

무료 계정으로 4개 채널, 채널당 8개의 필드를 구성할 수 있음
Data Export 기능을 이용해서 저장한 데이터를 CSV format으로 Download 받을 수 있음.

ASI A-SUNG
International CO.,LTD.

www.CodeZoo.co.kr



ThingSpeak™ Channels Apps Support

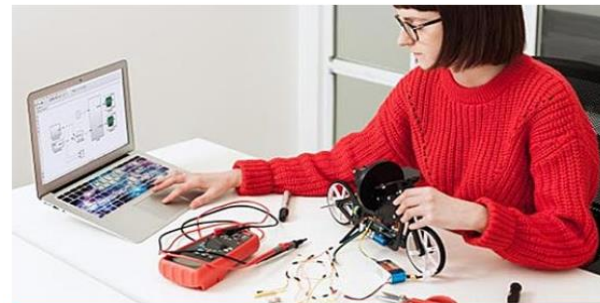
ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

Channels Learn More

M MOUSER ELECTRONICS

The banner features a hand holding a tablet displaying various data visualizations: a bar chart, a line graph, and a 3D surface plot. The background is a blurred outdoor scene.



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

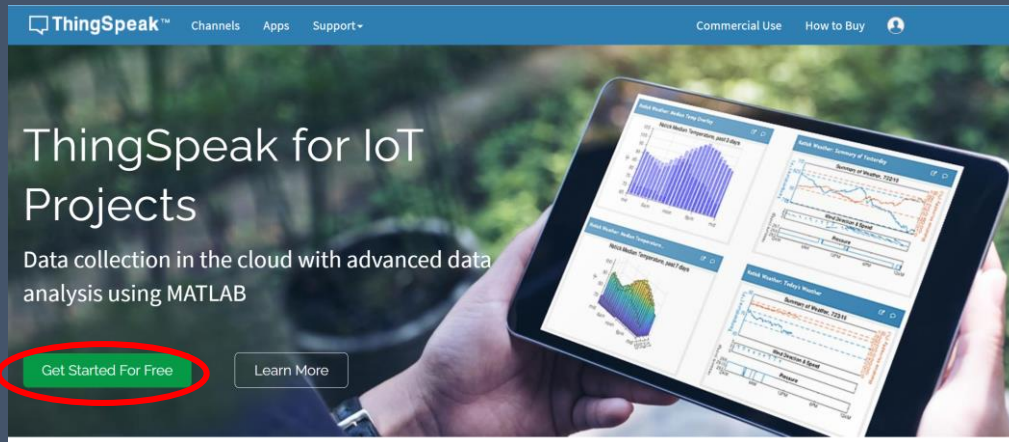


ThingSpeak for Smart Farming

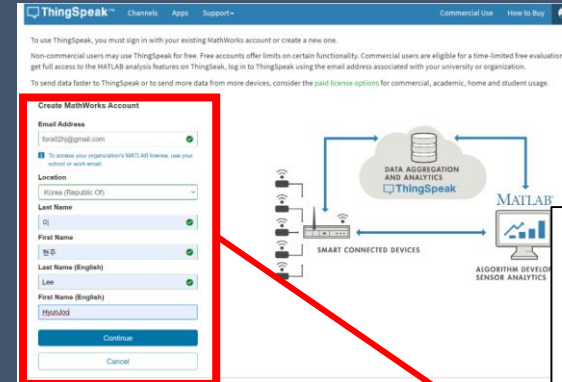
Build applications for remote monitoring of sensors deployed on farms, and develop MATLAB models to increase crop yields and reduce costs

ThingSpeak 계정생성

1. www.thingspeak.com 접속한 후 'Get Started For Free' 클릭



3. 정보입력 후 'Continue' 클릭



Create MathWorks Account

Email Address:

To access your organization's MATLAB license, use your school or work email.

Location:

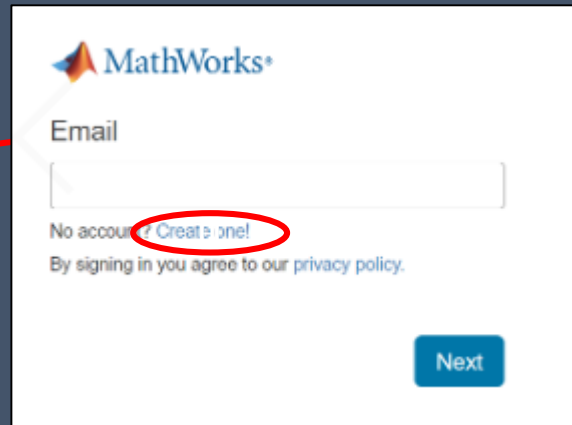
Last Name:

First Name:

Last Name (English):

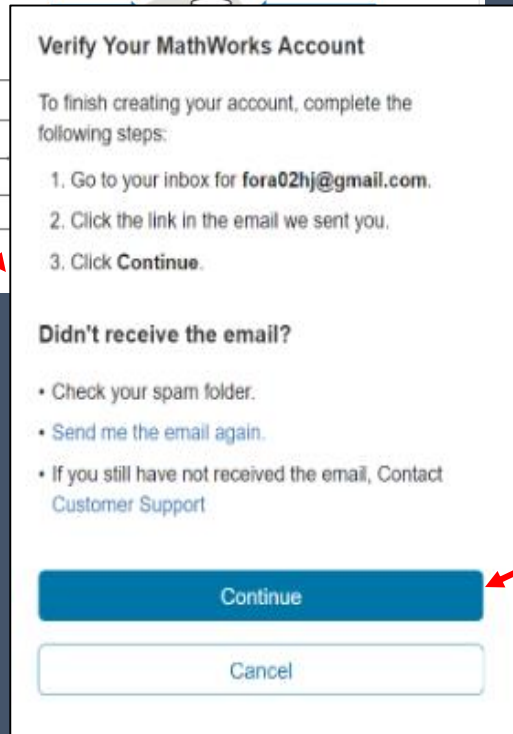
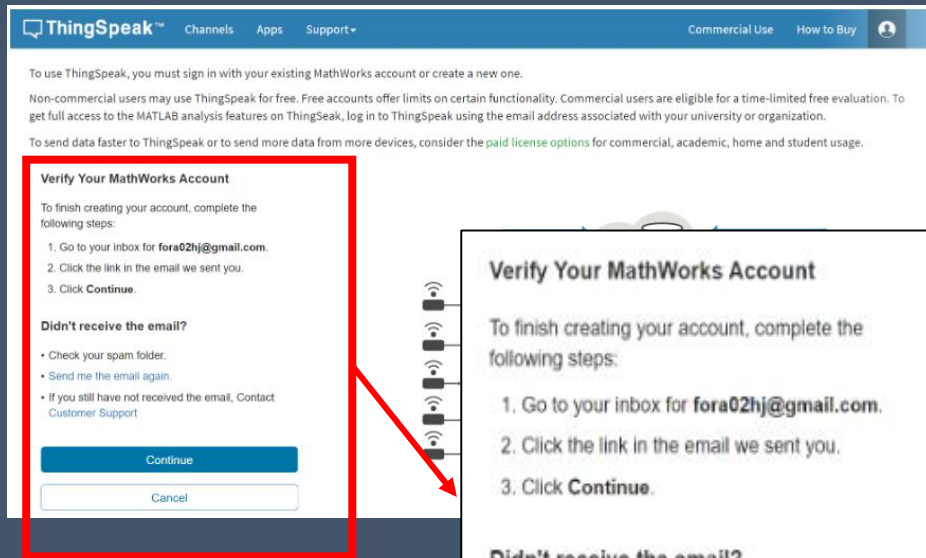
First Name (English):

2. 'Create one!' 클릭



ThingSpeak 계정생성

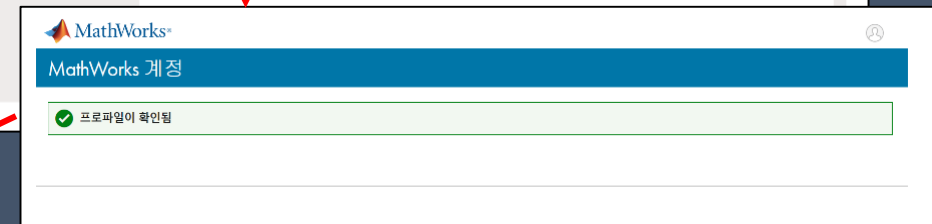
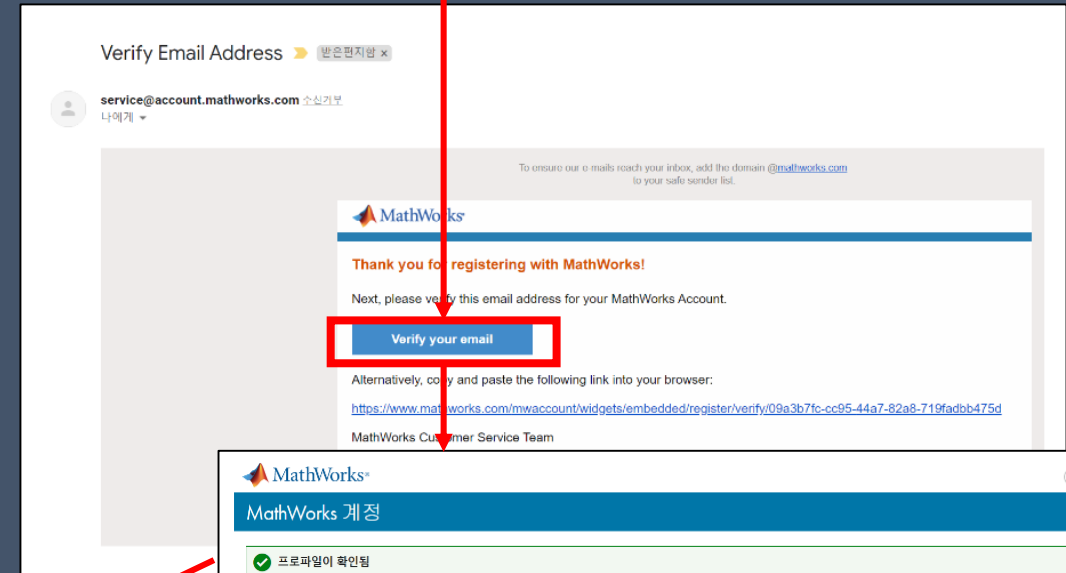
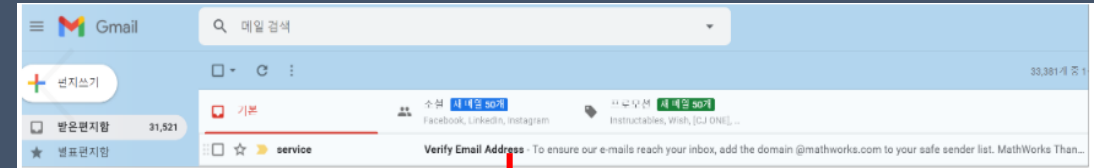
4. 인증하는 방법 확인하기



인증 확인 되면
이전으로 돌아가서
'Continue' 클릭

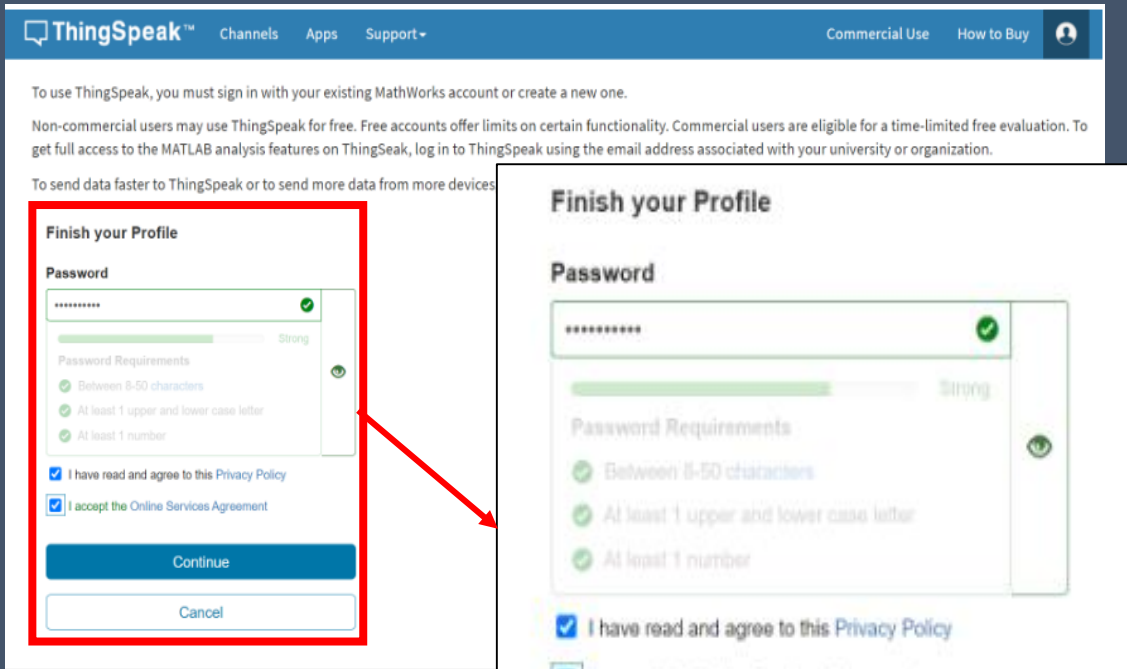
5. 이메일로 인증하기

이메일이 안 왔으면 스팸 폴더를 확인해보세요.



ThingSpeak 계정생성

6. 비밀번호 설정한 후 'Continue' 클릭



ThingSpeak™ Channels Apps Support Commercial Use How to Buy

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices

Finish your Profile

Password

***** Strong

Password Requirements

- Between 8-50 characters
- At least 1 upper and lower case letter
- At least 1 number

I have read and agree to this Privacy Policy

I accept the Online Services Agreement

Continue

Cancel

7. 가입 성공 '확인' 클릭



ThingSpeak™ 채널 앱 지원하다 상업적 사용 구입 방법 HL

성공적으로 로그인되었습니다. 엑스

가입 성공

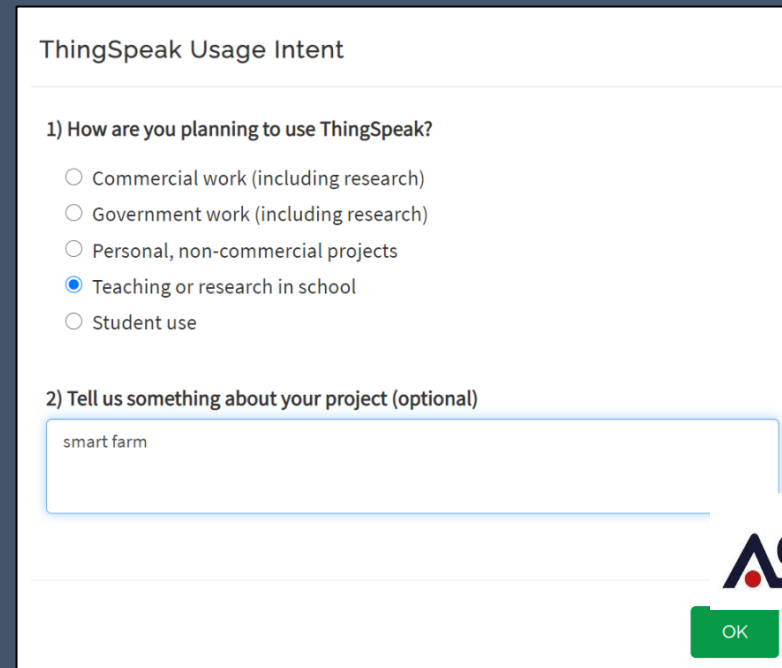
축하합니다. MathWorks 계정을 ThingSpeak에 성공적으로 연결했습니다. ThingSpeak에 대한 모든 후속 로그인에서 다음 이메일 ID 및 관련 MathWorks 계정 비밀번호를 사용합니다.

이메일 ID: fora02hj@gmail.com

ThingSpeak에 오신 것을 환영합니다!

확인

8. 설문 작성



ThingSpeak Usage Intent

1) How are you planning to use ThingSpeak?

- Commercial work (including research)
- Government work (including research)
- Personal, non-commercial projects
- Teaching or research in school
- Student use

2) Tell us something about your project (optional)

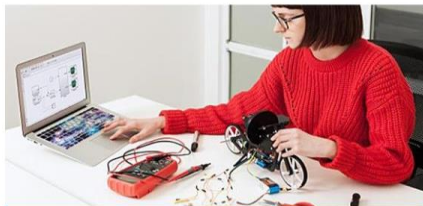
smart farm

ASI A-SUNG International CO.,LTD.

OK

ThingSpeak 로그인

1. 오른쪽 상단 사람 아이콘 클릭



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

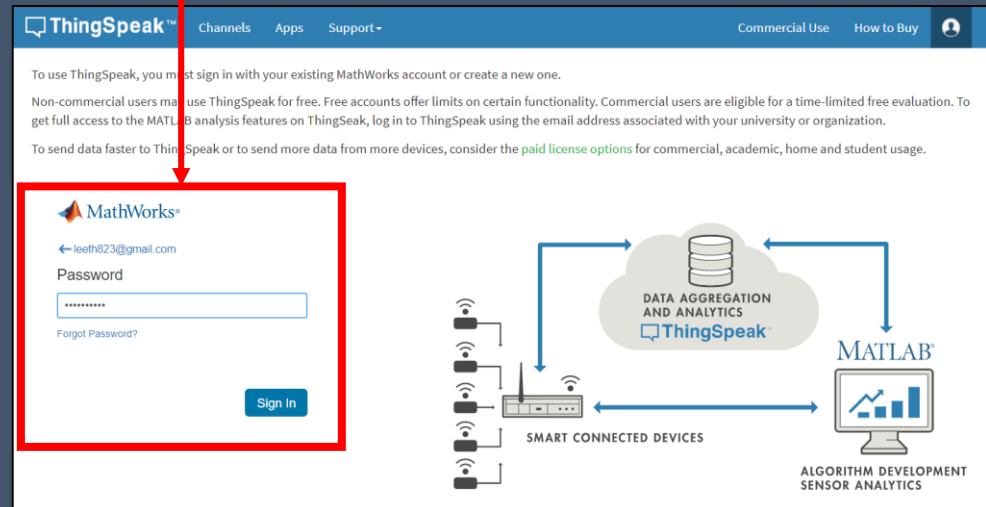
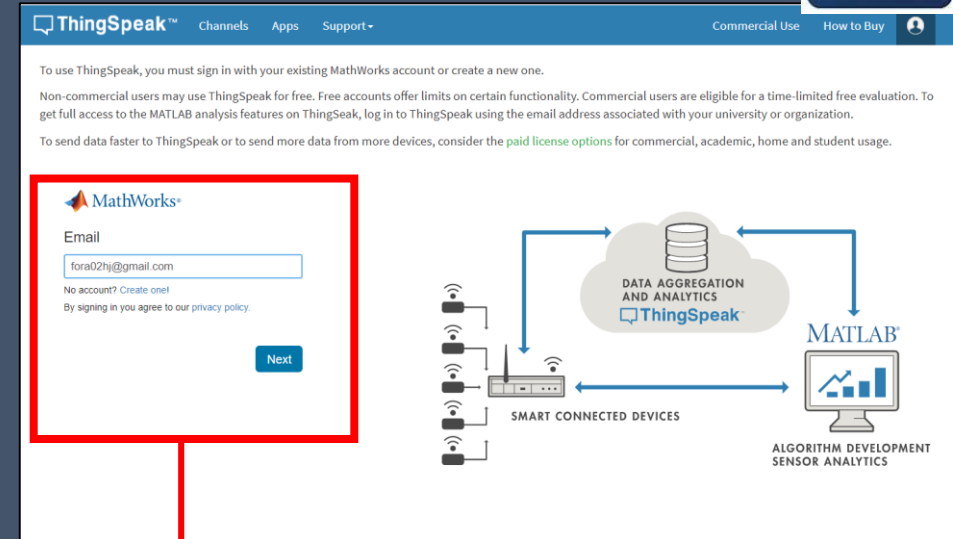


ThingSpeak for Energy Monitoring
Build IoT applications to monitor energy usage, and develop MATLAB models for power signature identification and load forecasting



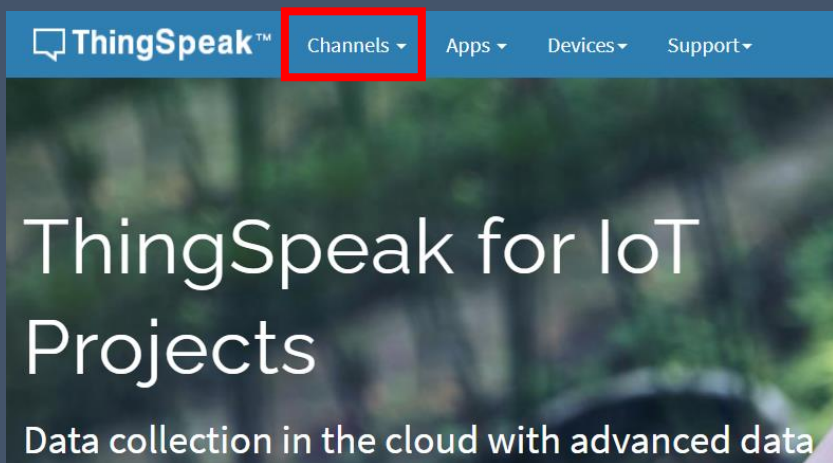
ThingSpeak for Smart Farming
Build applications for remote monitoring of sensors deployed on farms, and develop MATLAB models to increase crop yields and reduce costs

2. 로그인 Email/Password 입력

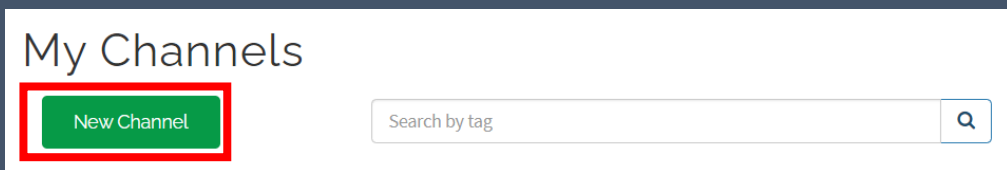


ThingSpeak 채널 생성 및 설정

1. 상단 메뉴에서 'channels' 클릭



2. 새로운 채널 생성



New Channel

Name

Description

Field 1	<input type="text" value="Temperature"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="Humidity"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text"/>	<input type="checkbox"/>
Field 4	<input type="text"/>	<input type="checkbox"/>
Field 5	<input type="text"/>	<input type="checkbox"/>
Field 6	<input type="text"/>	<input type="checkbox"/>
Field 7	<input type="text"/>	<input type="checkbox"/>
Field 8	<input type="text"/>	<input type="checkbox"/>

Metadata

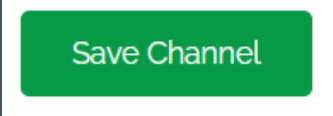
Help

Channels store all the data that a ThingSpeak app sends to it. Each channel has up to eight fields that can hold any type of data, plus status data. Once you collect data in a channel, you can view and visualize it.

Channel Settings

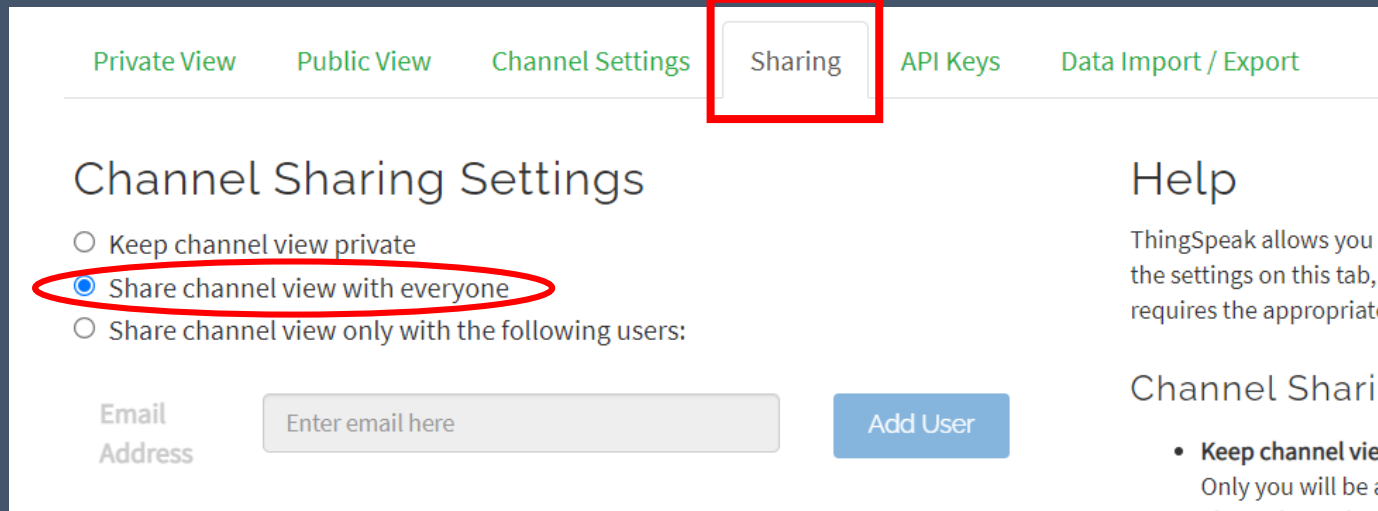
- Percentage complete:** Calculated based on the amount of data collected in the channel. Enter the name, description, location, and metadata for the channel.
- Channel Name:** Enter a unique name for the channel.
- Description:** Enter a description of the channel.
- Field#:** Check the box to enable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

위와 같이 입력하고 'Save Channel' 클릭



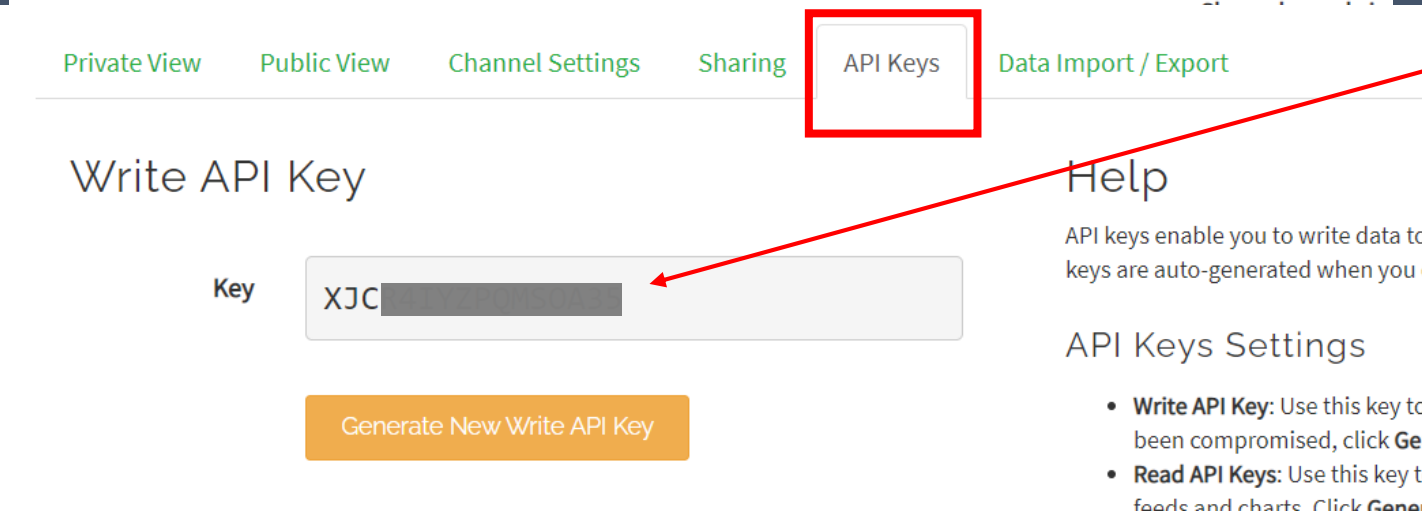
ThingSpeak 채널 생성 및 설정

Public View를 활성화 해서 모바일앱, 웹사이트에서 언제나 데이터 확인 가능.
데이터를 공유하지 않을 경우 Keep channel view private 체크



The screenshot shows the 'Channel Sharing Settings' page in ThingSpeak. The 'Sharing' tab is highlighted with a red box. Under 'Channel Sharing Settings', the option 'Share channel view with everyone' is selected and circled in red. Below this, there is a form for adding users with an 'Email Address' field and an 'Add User' button.

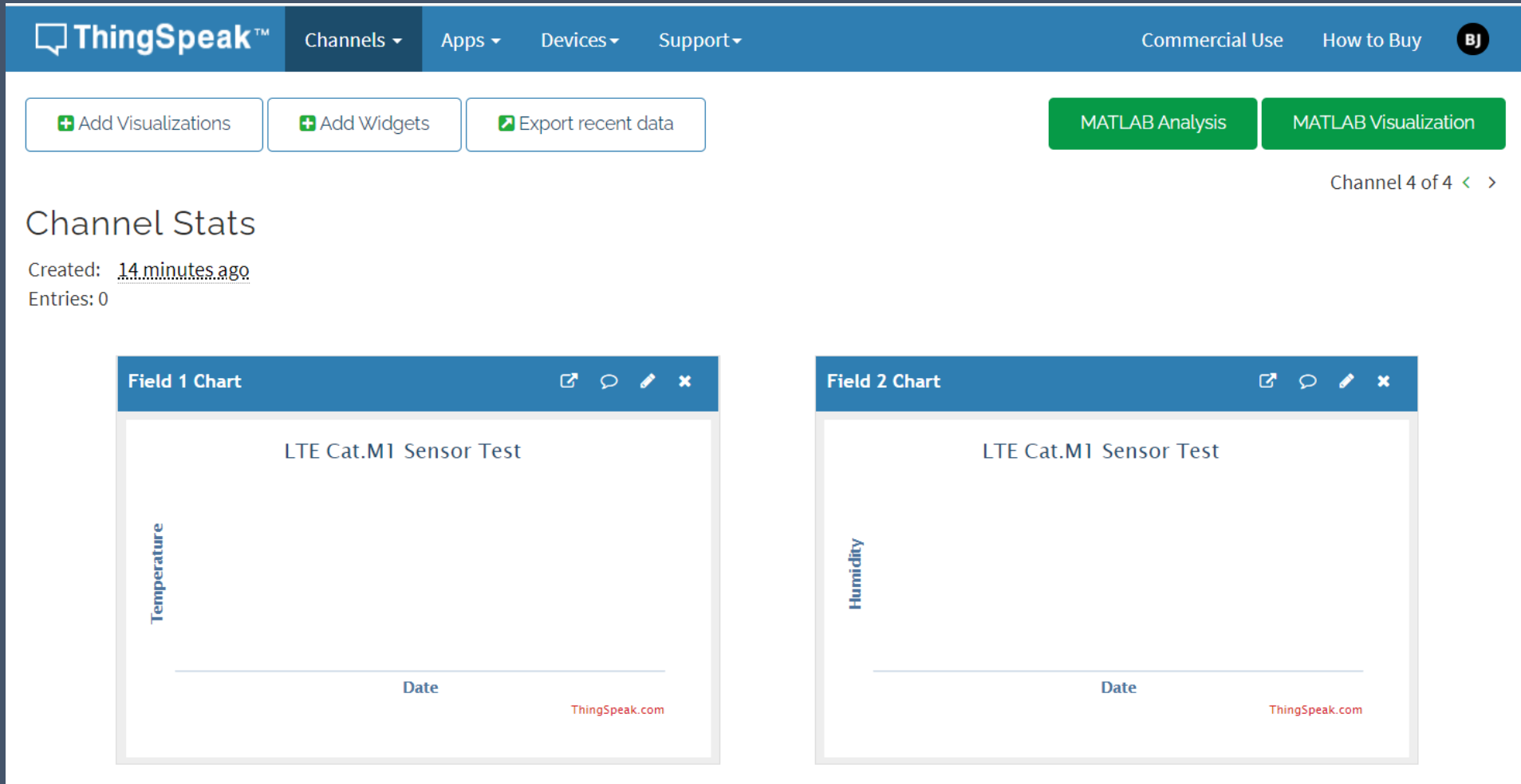
Thingspeak.com에 센서 데이터를 업로드 하기 위해 Write API Key가 필요합니다.



The screenshot shows the 'Write API Key' page in ThingSpeak. The 'API Keys' tab is highlighted with a red box. A 'Key' field is visible, containing the text 'XJC' followed by a redacted area. A red arrow points from the text 'Write API Key가 필요합니다.' to this field. Below the field is a 'Generate New Write API Key' button.

ThingSpeak 채널 생성 및 설정

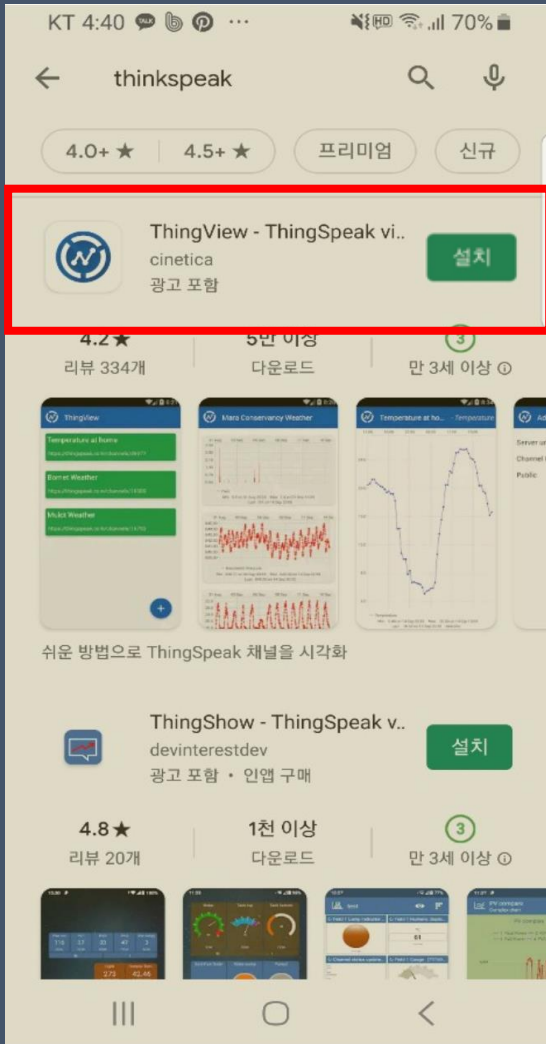
4. Private View 조회



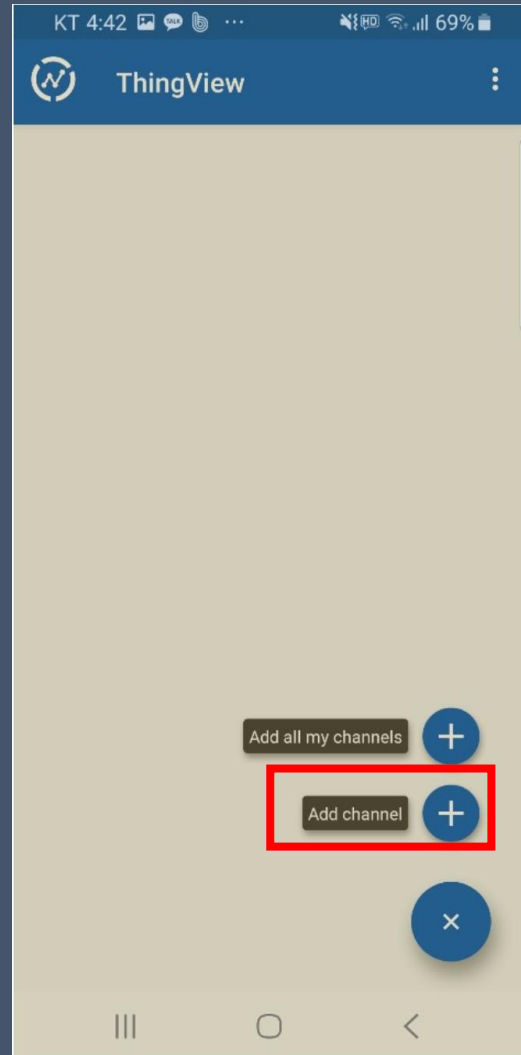
The screenshot displays the ThingSpeak web interface. At the top, there is a navigation bar with the ThingSpeak logo and menu items: Channels, Apps, Devices, Support, Commercial Use, How to Buy, and a user profile icon labeled 'BJ'. Below the navigation bar, there are several action buttons: '+ Add Visualizations', '+ Add Widgets', 'Export recent data', 'MATLAB Analysis', and 'MATLAB Visualization'. The main content area shows 'Channel 4 of 4' and 'Channel Stats' with the following information: Created: 14 minutes ago, Entries: 0. Two empty charts are visible, both titled 'LTE Cat.M1 Sensor Test'. The left chart is labeled 'Field 1 Chart' and has a vertical axis for 'Temperature' and a horizontal axis for 'Date'. The right chart is labeled 'Field 2 Chart' and has a vertical axis for 'Humidity' and a horizontal axis for 'Date'. Both charts include a 'ThingSpeak.com' watermark at the bottom right.

ThingSpeak 채널 조회 (앱)

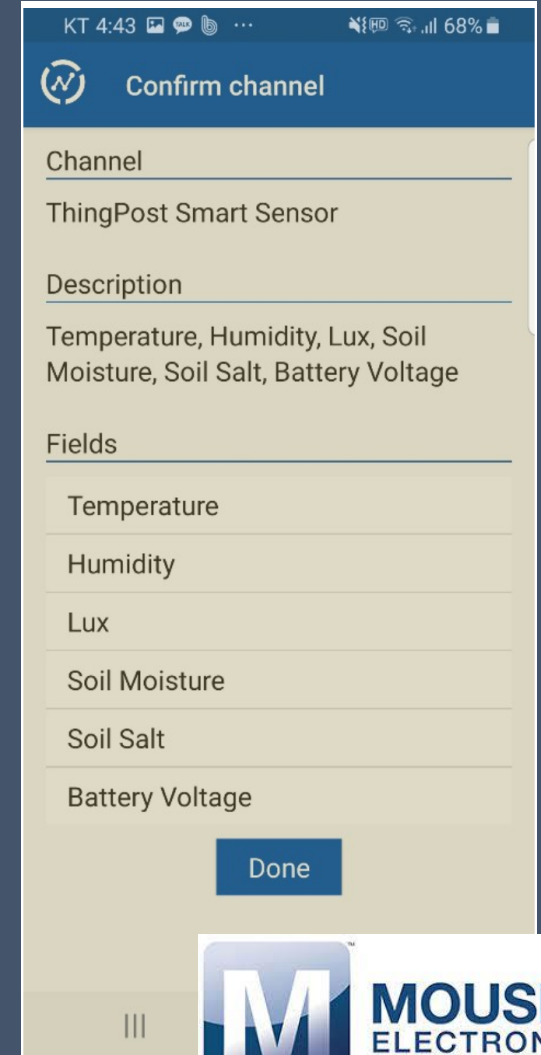
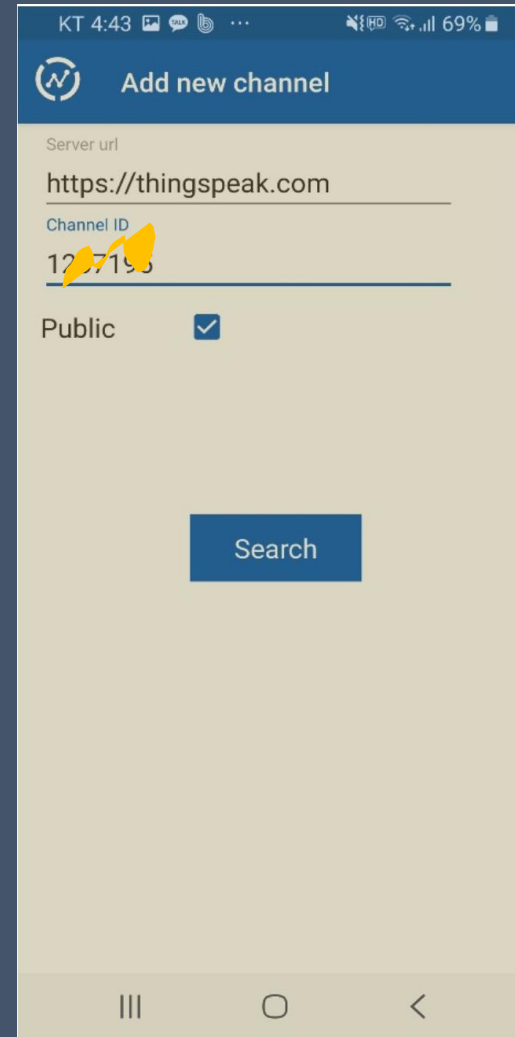
1. 앱 다운로드 및 설치



2. 채널 설정



3. 채널아이디 입력(자신의 것!!) 4. 채널 확인

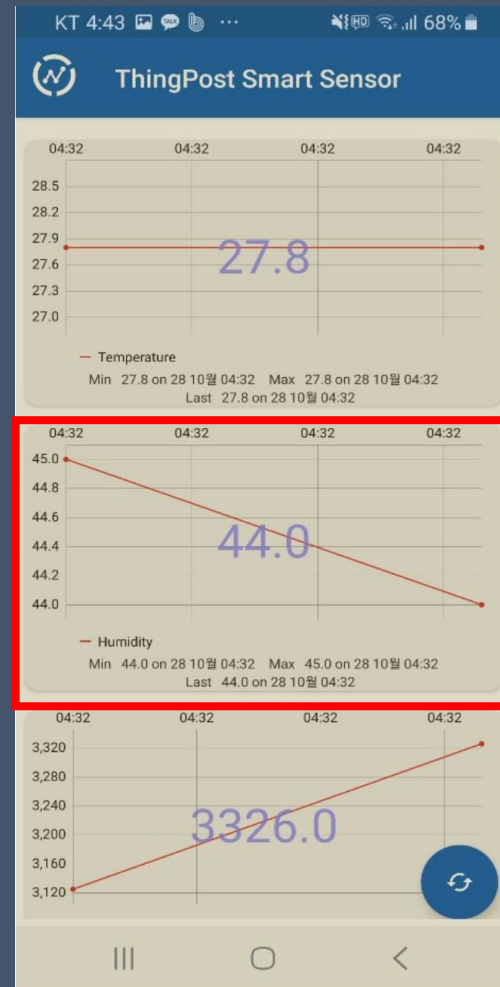


ThingSpeak 채널 조회 (앱)

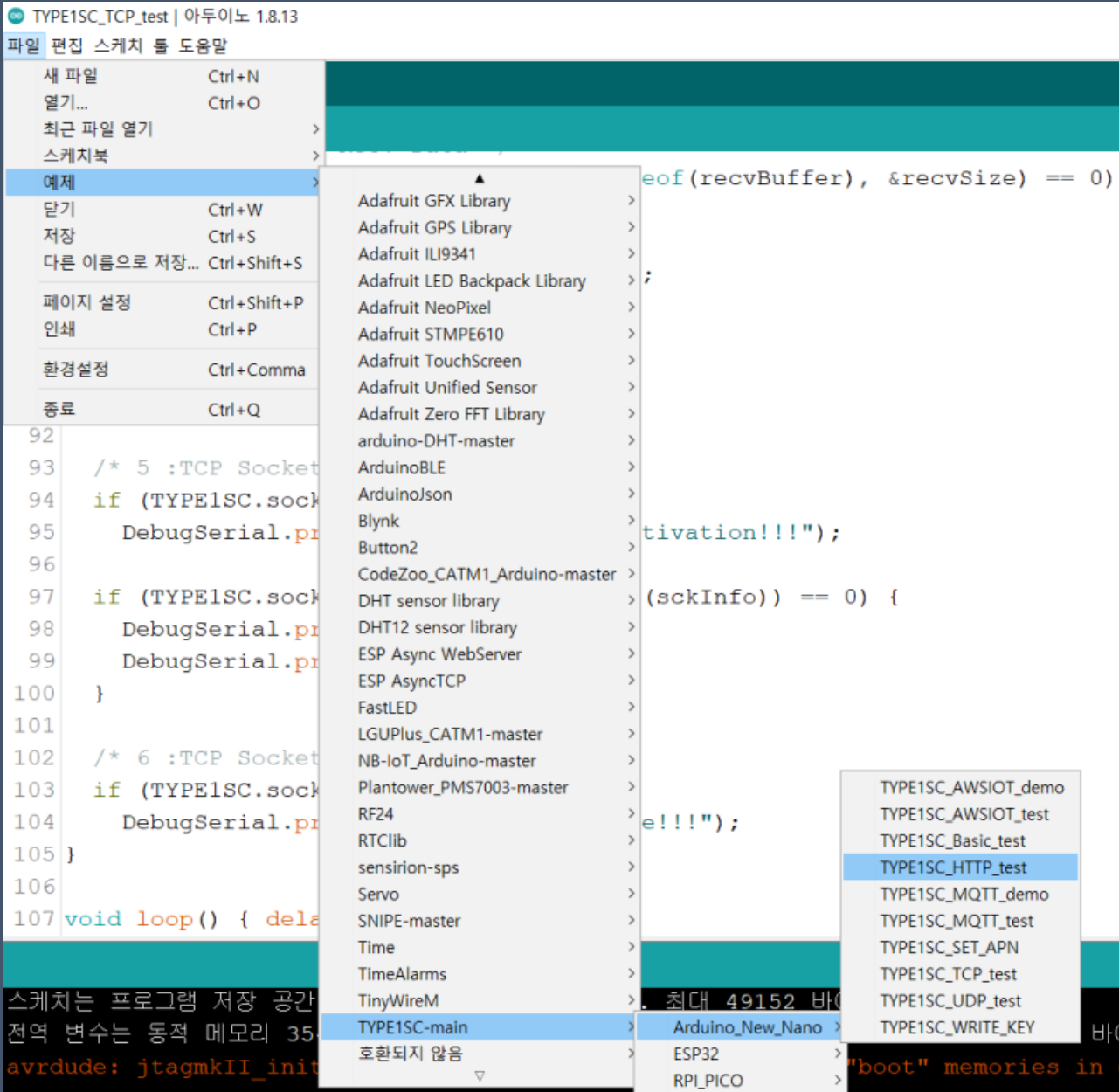
5. 채널 선택



6. 채널 조회



3) HTTP Test



✓ 예제 -> TYPE1SC-main -> Arduino_New_Nano -> TYPE1SC_HTTP_test 선택

3) HTTP Test

```
67
68 String WApiKey = "*****"; // Thing Speak Write API Key 16Character
69 float temp = 0.0;
70 float humi = 0.0;
71 String fieldTemp = "field1"; // Air temperature
72 String fieldHumi = "field2"; // Air humidity
73
74 /*Get Temperature & Humidity */
75 while (1) {
76     /* Get DHT22 Sensor */
77     temp = dht.readTemperature();
78     humi = dht.readHumidity();
79     if (String(temp) != "nan" && String(humi) != "nan")
```

***** 지우고 Thingspeak.com 채널에서
생성한 Write Key 입력

```
TYPE1SC_HTTP_test
1 #include "TYPE1SC.h"
2 #if !defined(__AVR_ATmega4809__)
3 #include <avr/dtostrf.h>
4 #endif
5
6 #define DebugSerial Serial
7 #define M1Serial Serial1
8
9 #define DHTPIN A0
10 #include "DHT.h" /* https://github.com/markruys/arduino-DHT */
11 // Uncomment whatever type you're using!
12 // #define DHTTYPE DHT11 // DHT 11
13 #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
14 // #define DHTTYPE DHT21 // DHT 21 (AM2301)
15
16 DHT dht(DHTPIN, DHTTYPE);
17
18 TYPE1SC TYPE1SC(M1Serial, DebugSerial);
```

DHT22 센서 Data 핀을 A0에 연결

3) HTTP Test

```
COM9
04:28:19.277 -> TCP Socket Activation!!!
04:28:19.325 -> Socket Info : ACTIVATED
04:28:19.325 -> [HTTP Send] >> GET /update?api_key=X-XXXXXXXXXXXXXXXXXXXX&field1=26.90&field2=90.90 HTTP/1.1
04:28:19.325 -> Host: api.thingspeak.com
04:28:19.325 -> Connection: close
04:28:19.325 ->
04:28:22.444 -> [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!!!
```

HTTP로 Thingspeak.com에 DHT22 센서에서 수집한 온도, 습도 데이터 업로드

데이터가 전송이 성공하면 200 OK 반환

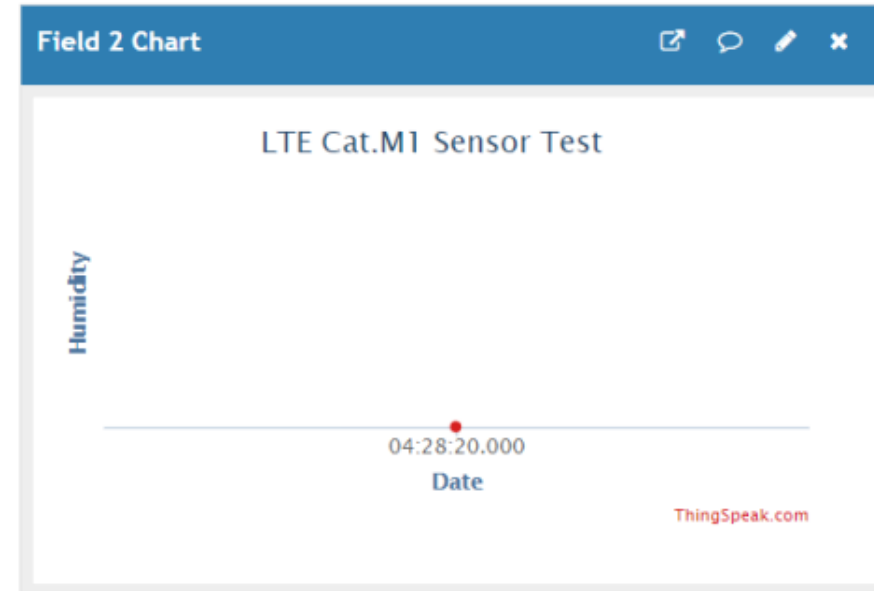
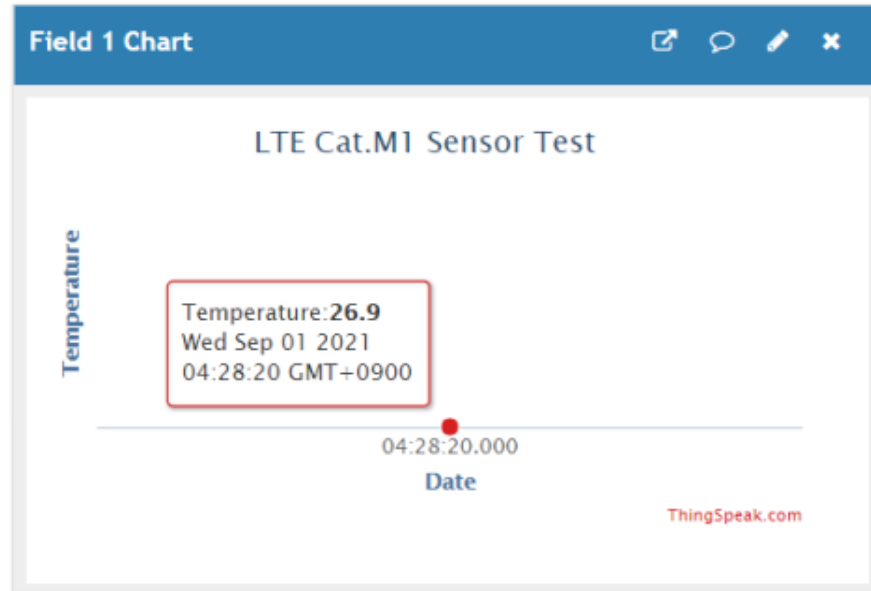
3) HTTP Test

Channel Stats

Created: [about an hour ago](#)

Last entry: [14 minutes ago](#)

Entries: 1

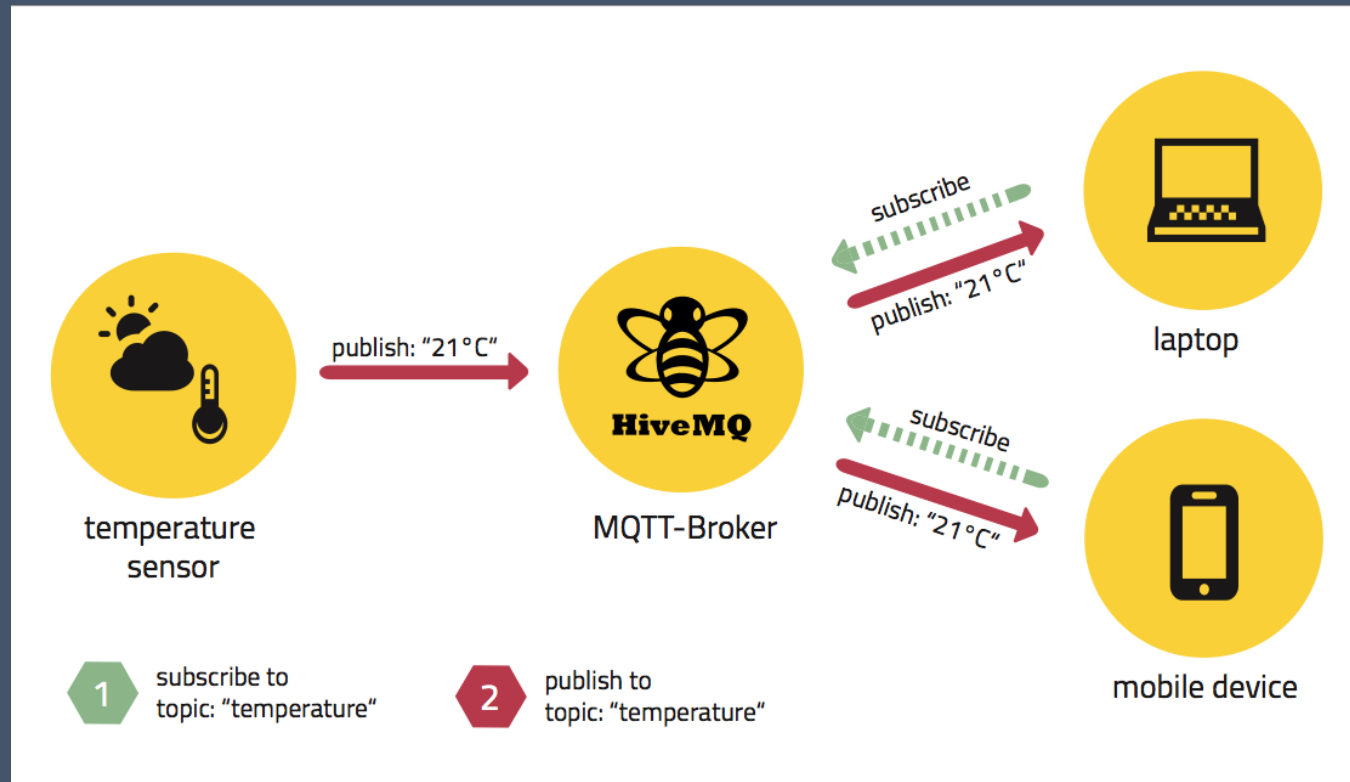


4) MQTT Test, Demo

CAT.M1 모뎀 내부 MQTT 통신스택을 이용해서 HiveMQ 에 아두이노 보드에서 측정한 온도, 습도 데이터를 Publish 하고 PC에서 실시간 모니터링 합니다.

1. PC에서 MQTT 데이터를 모니터링 하기 위해 설정 합니다.

2. CAT.M1 모뎀 내부 MQTT 통신스택으로 HiveMQ Broker와 연결하고 아두이노 보드에서 DHT22 센서로 온도, 습도 데이터를 측정한 데이터를 Publish 합니다.

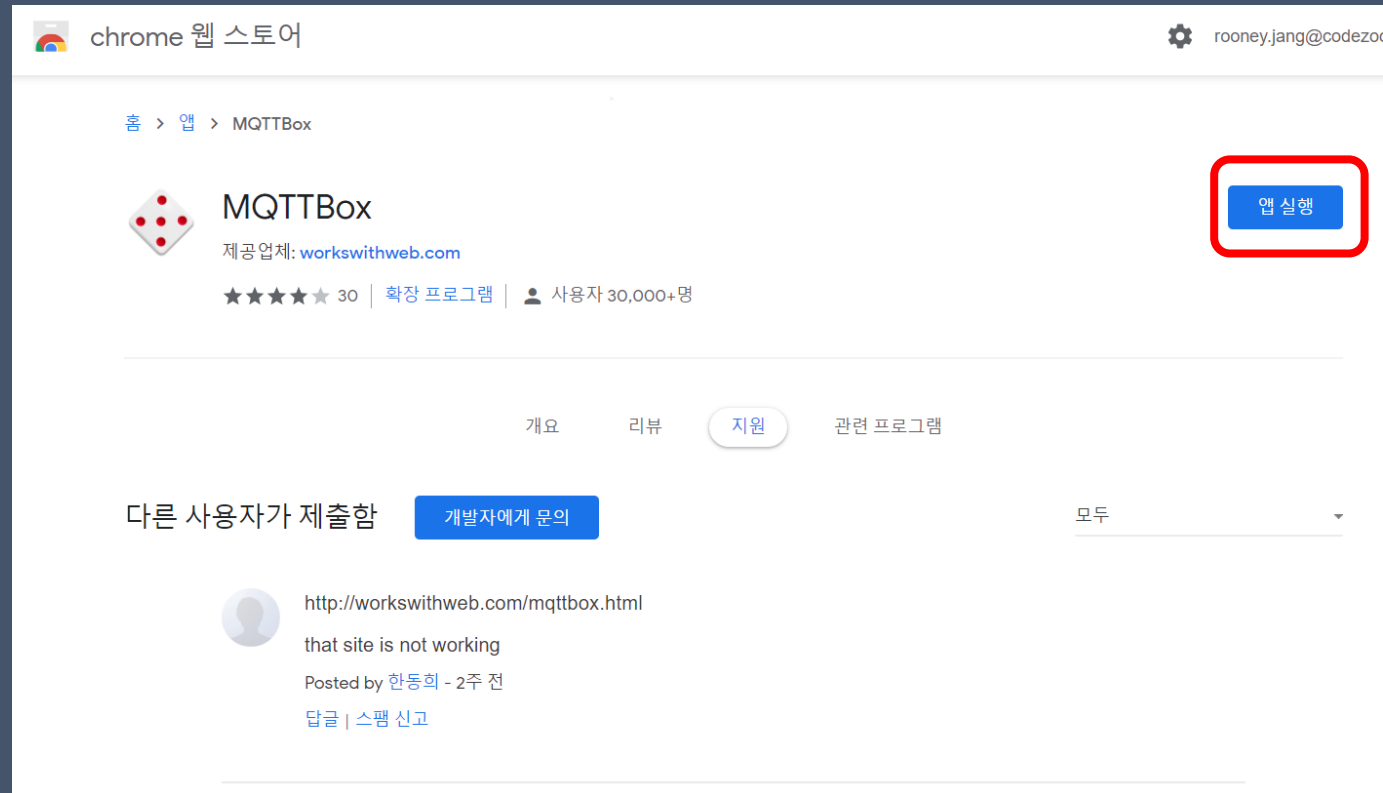


Chrome Extension 설치

- PC환경에서 MQTT broker 에 원하는 토픽을 구독(subscribe)하고 발행(Publish)하기 위해 크롬 확장 프로그램 설치합니다.


- MQTTBox 프로그램을 설치 합니다

<https://chrome.google.com/webstore/detail/mqttbox/kaajoficamnjjhkeomgfljpicifbkaf/support?hl=ko>



chrome 웹 스토어


홈 > 앱 > MQTTBox

 **MQTTBox**
제공업체: workswithweb.com
★★★★★ 30 | 확장 프로그램 | 사용자 30,000+명

앱 실행

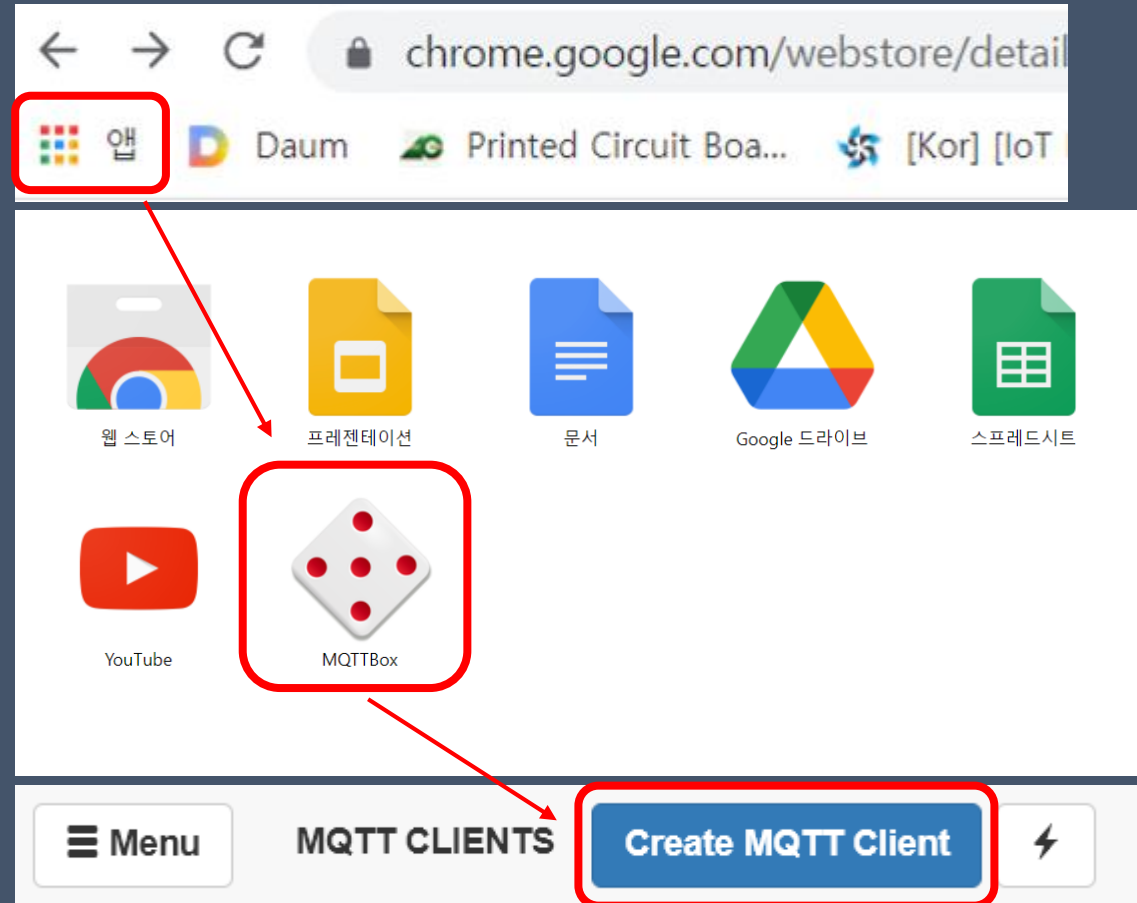
개요 리뷰 지원 관련 프로그램

다른 사용자가 제출함 [개발자에게 문의](#) 모두

 <http://workswithweb.com/mqttbox.html>
that site is not working
Posted by [한동희](#) - 2주 전
[답글](#) | [스팸 신고](#)

MQTTBox 설정하기

- 앱 클릭 -> MQTTBox 선택
- Create MQTT Client 선택



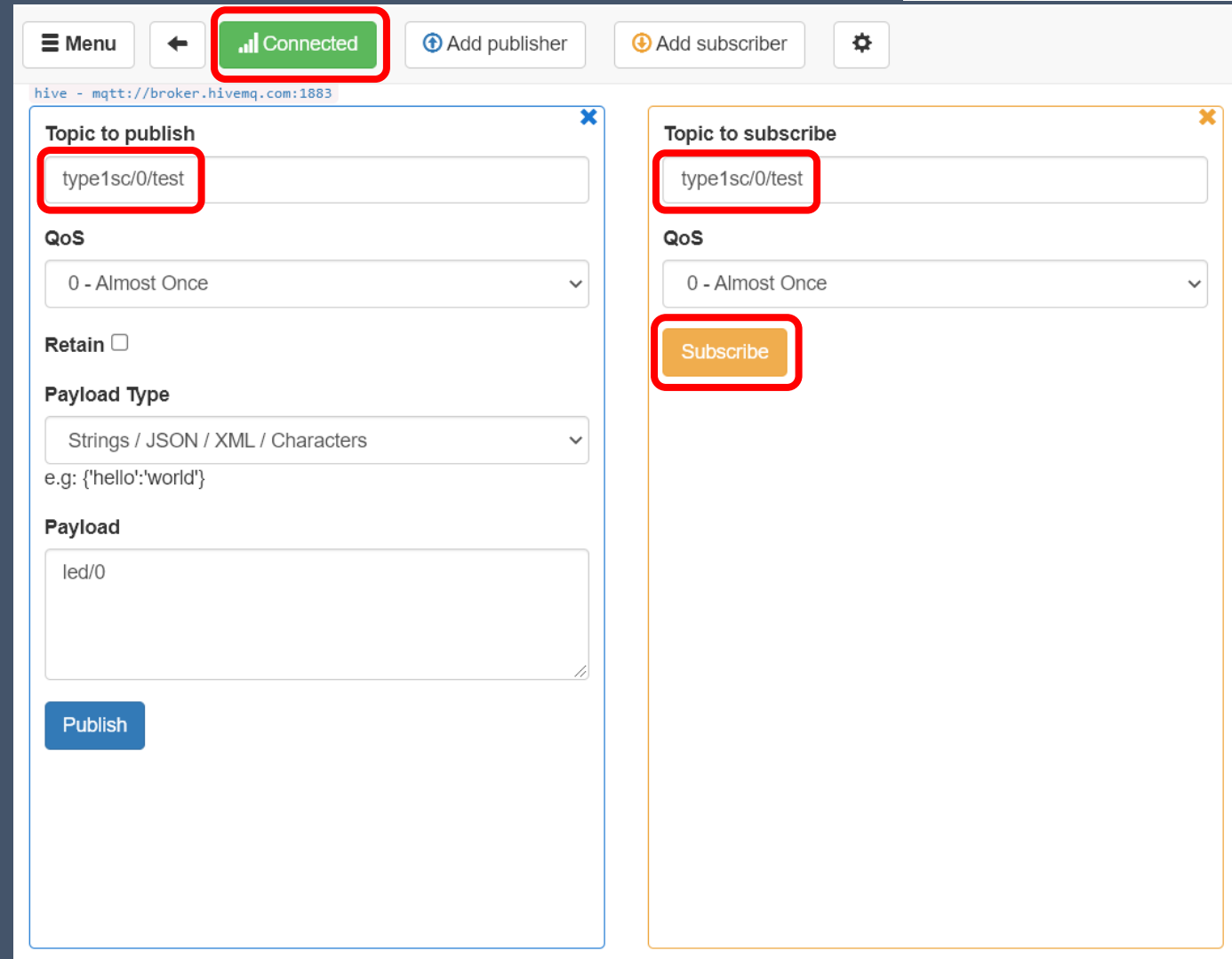
MQTTBox 설정하기

Menu ← MQTT CLIENT SETTINGS [Client Settings Help](#)

MQTT Client Name <input type="text" value="hive"/>	MQTT Client Id <input type="text" value="90f2e9c3-6e70-4a90-93c8-3cc3aaeC"/>	Append timestamp to MQTT client id? <input checked="" type="checkbox"/> Yes	Broker is MQTT v3.1.1 compliant? <input checked="" type="checkbox"/> Yes
Protocol <input type="text" value="mqtt / tcp"/>	Host <input type="text" value="broker.hivemq.com:1883"/>	Clean Session? <input checked="" type="checkbox"/> Yes	Auto connect on app launch? <input checked="" type="checkbox"/> Yes
Username <input type="text" value="Username"/>	Password <input type="text" value="Password"/>	Reschedule Pings? <input checked="" type="checkbox"/> Yes	Queue outgoing QoS zero messages? <input checked="" type="checkbox"/> Yes
Reconnect Period (milliseconds) <input type="text" value="1000"/>	Connect Timeout (milliseconds) <input type="text" value="30000"/>	KeepAlive (seconds) <input type="text" value="10"/>	
Will - Topic <input type="text" value="Will - Topic"/>	Will - QoS <input type="text" value="0 - Almost Once"/>	Will - Retain <input type="checkbox"/> No	Will - Payload <input type="text"/>

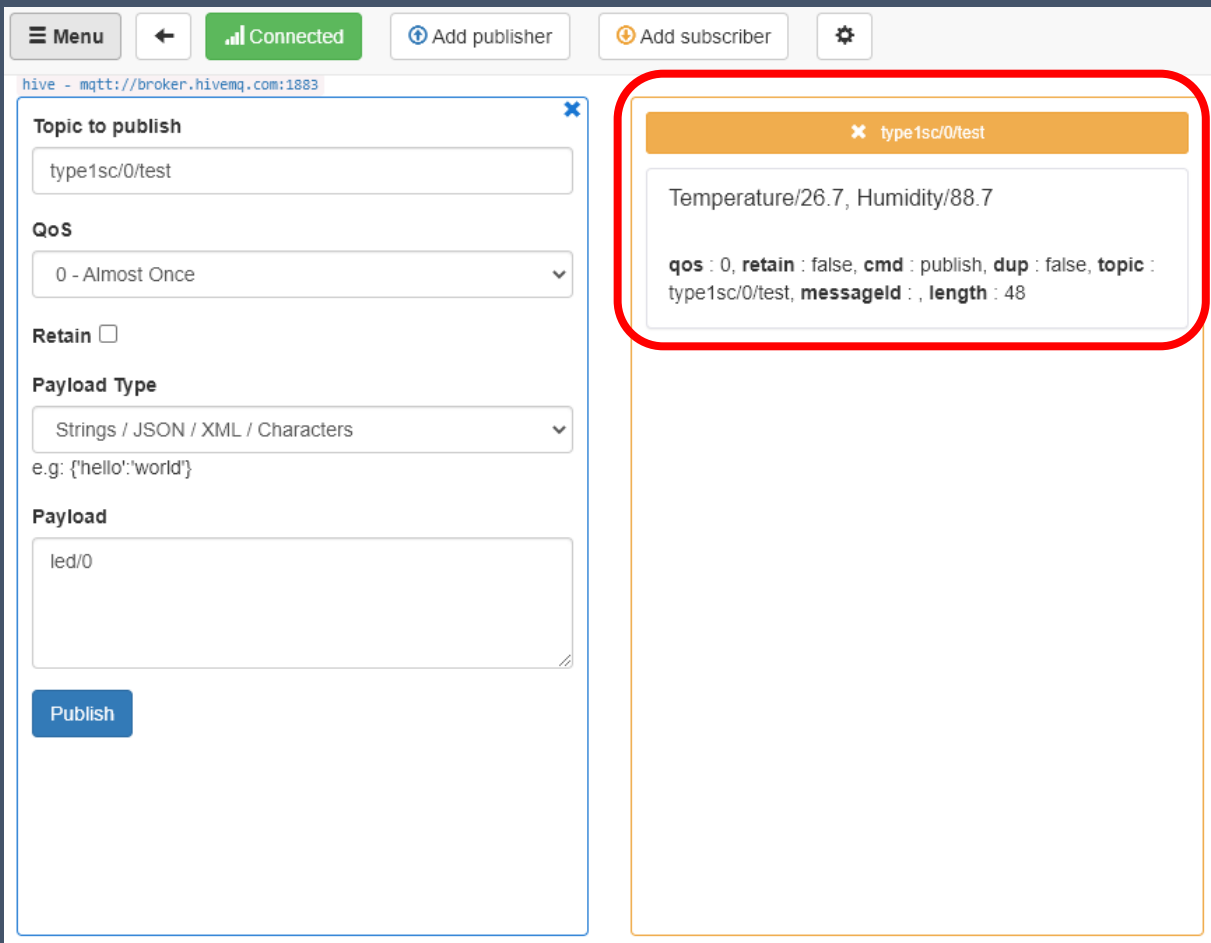
MQTTBox 설정하기

- 상단 Connected 되었는지 확인
- Topic to subscribe 에 구독 주제 입력,
단 아두이노 보드의 Topic과 동일해야
구독한 데이터를 확인할 수 있음
- 준비 되었으면 Subscribe 버튼 선택



The screenshot shows the MQTTBox web interface. At the top, a green 'Connected' status indicator is highlighted with a red box. Below it, the 'Topic to publish' field contains 'type1sc/0/test' and the 'Topic to subscribe' field also contains 'type1sc/0/test', both highlighted with red boxes. The 'Subscribe' button is also highlighted with a red box. The interface includes a 'Publish' button and various configuration options like QoS, Retain, and Payload Type.

4) MQTT Test, Demo



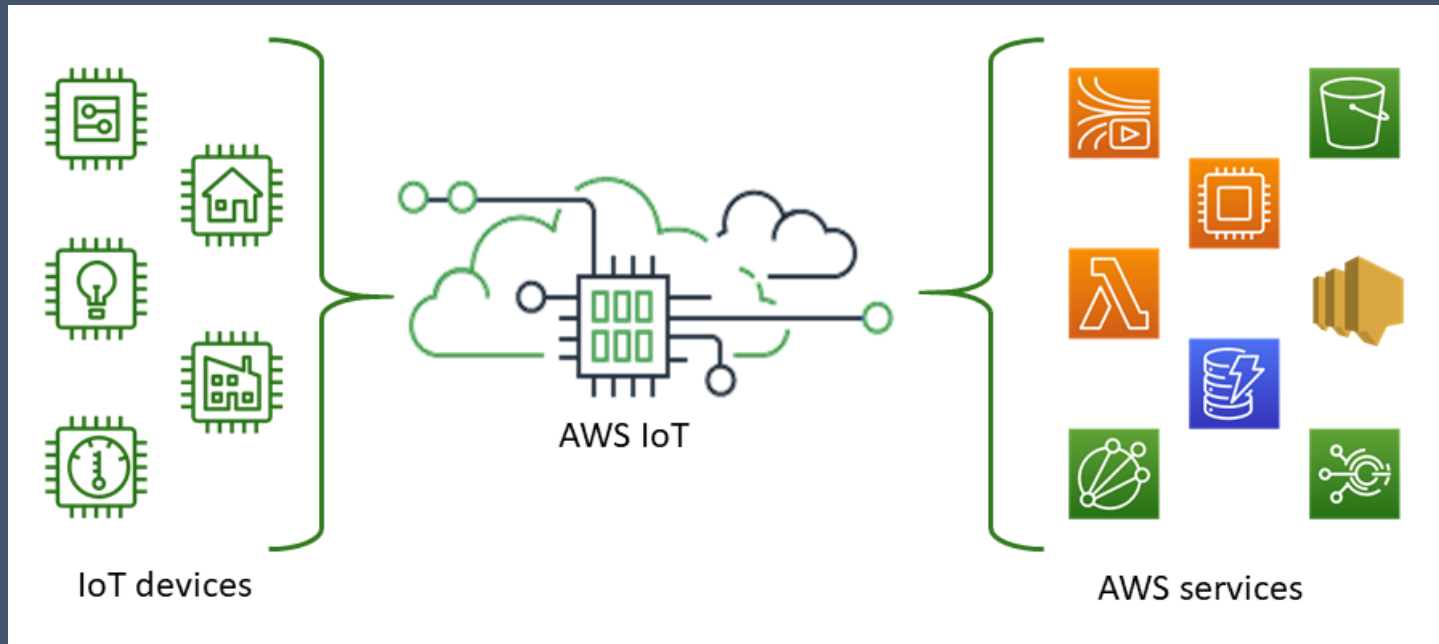
The image shows a MQTT client interface. On the left, the 'Topic to publish' field contains 'type1sc/0/test'. The 'QoS' is set to '0 - Almost Once'. The 'Payload Type' is 'Strings / JSON / XML / Characters'. The 'Payload' field contains 'led/0'. A 'Publish' button is visible. On the right, a message is displayed in a box with a red border, containing the text 'Temperature/26.7, Humidity/88.7' and metadata: 'qos : 0, retain : false, cmd : publish, dup : false, topic : type1sc/0/test, messageid : , length : 48'.

```
COM9
05:21:08.304 -> TYPE1SC Module Start!!!
05:21:10.371 -> Network not Ready !!!
05:21:13.184 -> Network not Ready !!!
05:21:15.196 -> Network not Ready !!!
05:21:17.165 -> TYPE1SC Module Ready!!!
05:21:17.165 -> 1.Enable MQTT events
05:21:17.213 -> 2.Configure node parameter:ID, Address
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
```

5) AWS-IoT 설정

CAT.M1 모뎀 내부 AWS-IoT 통신스택을 사용해서 아두이노 보드에서 측정된 온도, 습도 데이터를 AWS Cloud 로 전송 합니다.

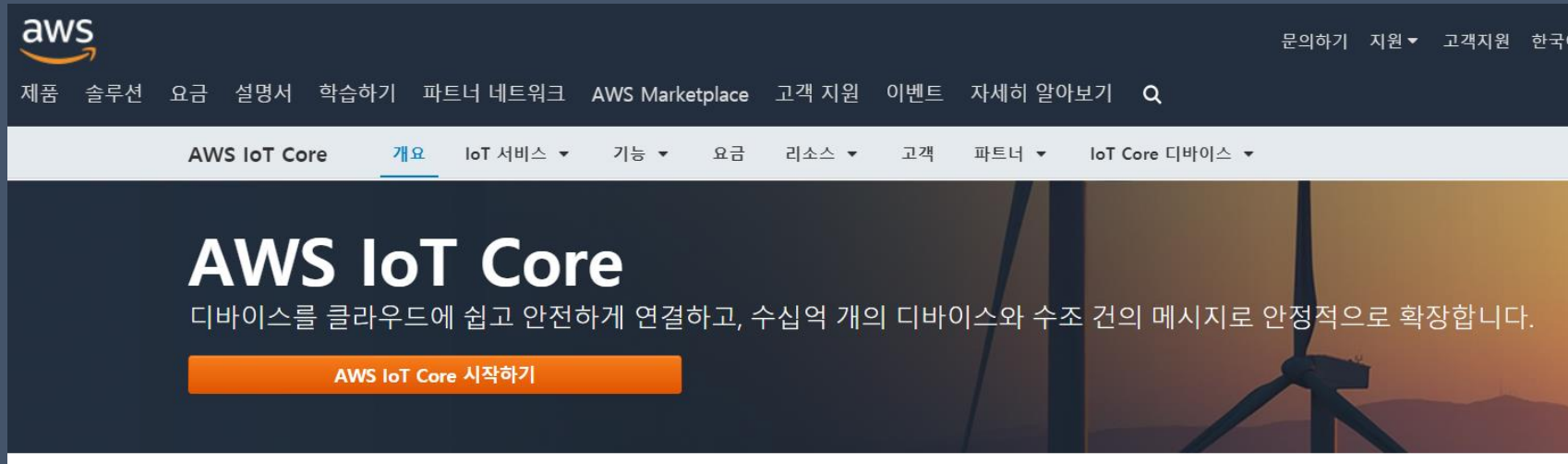
1. AWS-IoT를 MCU에서 직접 다루는 것보다 훨씬 쉽고 간단하게 사용하실 수 있습니다.
2. 필요한 소프트웨어 스택이 모두 Cat.M1 모뎀 내부에 구축되어 있어서 명령어 호출방식으로 손쉽게 이용하실 수 있습니다. 8비트 MCU (NANO Every, ATmega4809) 제품도 AWS-IoT 사용 가능



5) AWS-IoT 설정

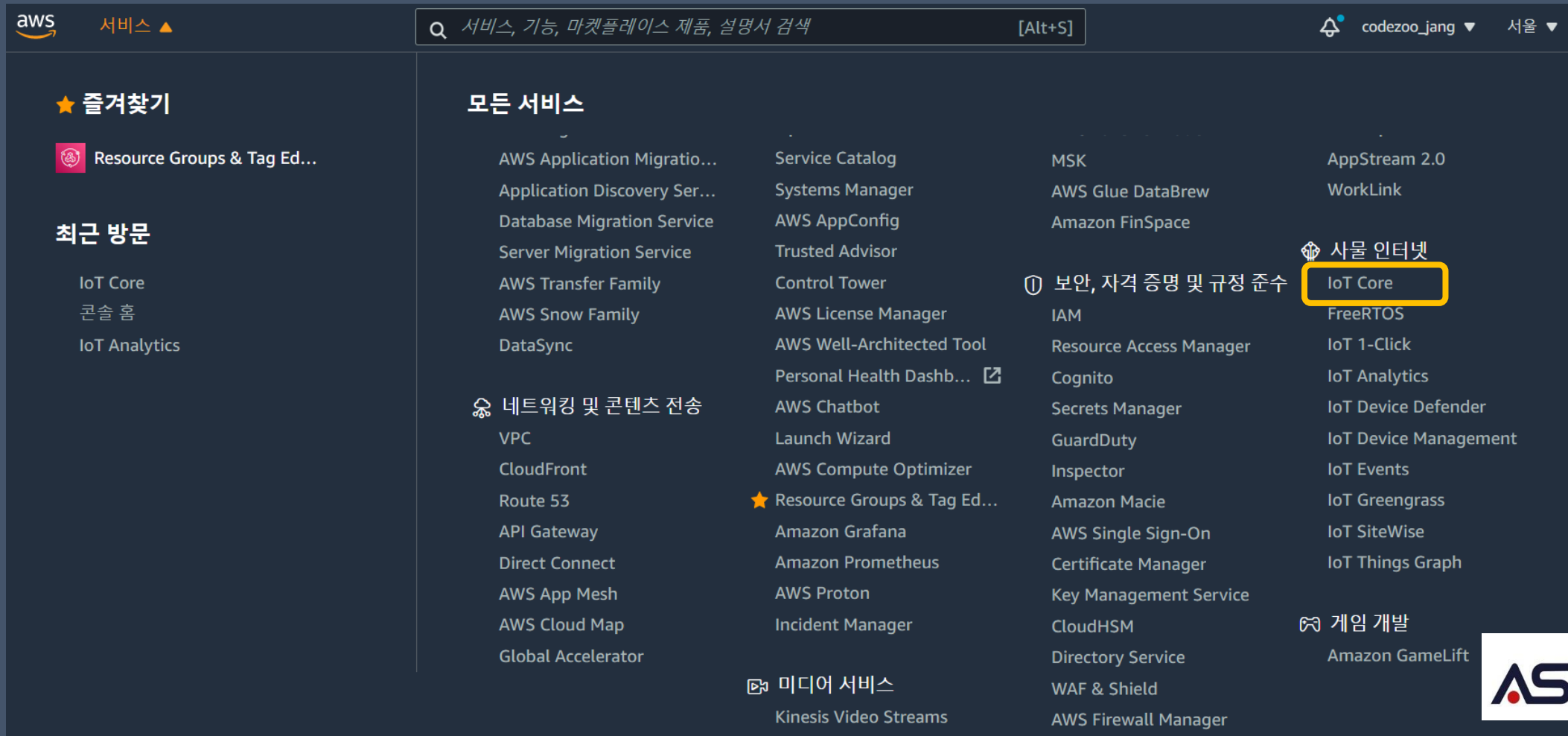
AWS 계정 생성

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



5) AWS-IoT 설정

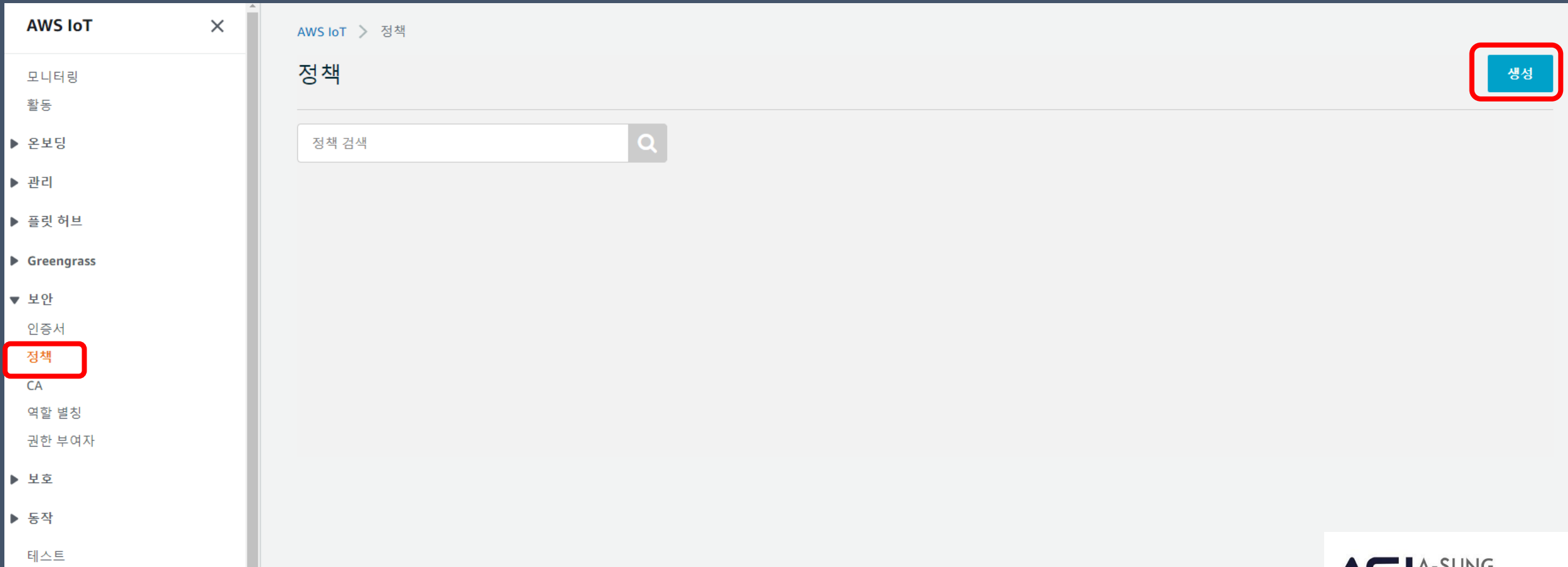
IoT Core 접속



The screenshot shows the AWS Management Console interface. At the top, there is a search bar with the text "서비스, 기능, 마켓플레이스 제품, 설명서 검색" and a search icon. The user's name "codezoo_jang" and location "서울" are visible in the top right corner. On the left sidebar, under "즐거찾기" (Discover), there is a link for "IoT Core" under the "최근 방문" (Recently visited) section. The main content area, titled "모든 서비스" (All services), displays a grid of various AWS services. The "IoT Core" service is highlighted with a yellow rectangular box. Other services visible include AWS Application Migration Service, Service Catalog, MSK, AppStream 2.0, Application Discovery Service, Systems Manager, AWS Glue DataBrew, WorkLink, Database Migration Service, AWS AppConfig, Amazon FinSpace, Server Migration Service, Trusted Advisor, 보안, 자격 증명 및 규정 준수 (Security, IAM, and Compliance), 사물 인터넷 (IoT Core), AWS Transfer Family, Control Tower, FreeRTOS, AWS Snow Family, AWS License Manager, IAM, IoT 1-Click, DataSync, AWS Well-Architected Tool, Resource Access Manager, IoT Analytics, Personal Health Dashboard, Cognito, IoT Device Defender, AWS Chatbot, Secrets Manager, IoT Device Management, Launch Wizard, GuardDuty, IoT Events, AWS Compute Optimizer, Amazon Macie, IoT Greengrass, Amazon Grafana, AWS Single Sign-On, IoT SiteWise, Amazon Prometheus, Certificate Manager, IoT Things Graph, AWS Proton, Key Management Service, Incident Manager, CloudHSM, 게임 개발 (Game development), and Amazon GameLift. At the bottom of the main content area, there is a section for "미디어 서비스" (Media services) including Kinesis Video Streams and "AWS Firewall Manager".

5) AWS-IoT 설정

정책 생성



The screenshot shows the AWS IoT console interface. On the left, a navigation sidebar is visible with the 'Policy' menu item highlighted in red. The main content area displays the 'Policy' page, which includes a search bar and a 'Create' button in the top right corner, also highlighted in red.

AWS IoT X

- 모니터링
- 활동
- ▶ 온보딩
- ▶ 관리
- ▶ 플릿 허브
- ▶ Greengrass
- ▼ 보안
 - 인증서
 - 정책**
 - CA
 - 역할 별칭
 - 권한 부여자
- ▶ 보호
- ▶ 동작
- 테스트

AWS IoT > 정책

정책

정책 검색

생성

5) AWS-IoT 설정

정책 생성2

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

설명문 추가

정책 설명문에서는 리소스를 사용하여 수행 가능한 작업의 유형을 정의합니다.

고급 모드

작업

리소스 ARN

효과

허용 거부

제거

5) AWS-IoT 설정

정책 생성3 : 리소스 입력 후 생성

설명문 추가

정책 설명문에서는 리소스를 사용하여 수행 가능한 작업의 유형을 정의합니다. 기본 모드

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "",
6       "Action": "",
7       "Resource": ""
8     }
9   ]
10 }
```

설명문 추가

생성

복사

```
{
  "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": "*"
    }
  ]
}
```

정책

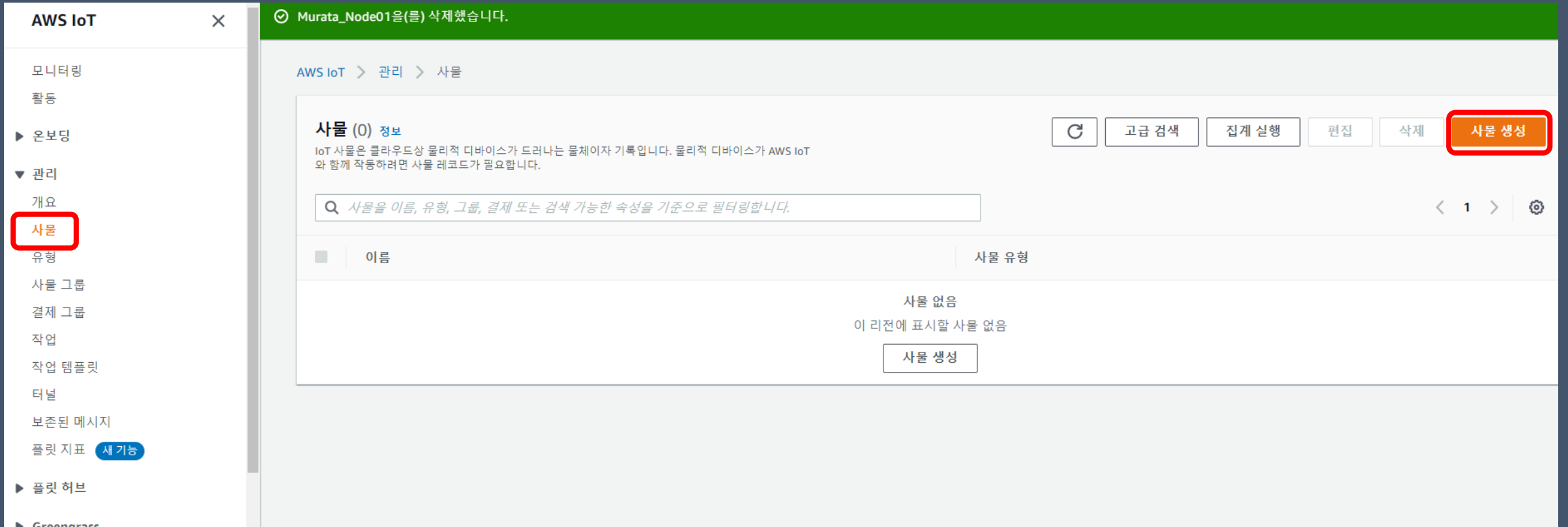
정책 검색

이름

Arduino_AWS_Policy

5) AWS-IoT 설정

사물 생성



Murata_Node01을(를) 삭제했습니다.

AWS IoT > 관리 > 사물

사물 (0) 정보 ↻ 고급 검색 집계 실행 편집 삭제 **사물 생성**

IoT 사물은 클라우드상 물리적 디바이스가 드러나는 물체이자 기록입니다. 물리적 디바이스가 AWS IoT와 함께 작동하려면 사물 레코드가 필요합니다.

🔍 사물을 이름, 유형, 그룹, 결제 또는 검색 가능한 속성을 기준으로 필터링합니다. < 1 > ⚙️

이름	사물 유형
사물 없음 이 리전에 표시할 사물 없음	

사물 생성

5) AWS-IoT 설정

사물 생성2 : 단일 사물 생성

사물 생성 정보

사물 리소스란 AWS IoT의 물리적 디바이스 또는 논리적 엔터티를 디지털로 표현한 것입니다. 디바이스 새도우, 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

5) AWS-IoT 설정

사물 생성3 : 사물 이름

사물 속성

사물 이름

문자, 숫자, 하이픈 또는 밑줄만 포함하는 고유한 이름을 입력합니다. 사물 이름에는 공백을 포함할 수 없습니다.

추가 구성

이러한 구성을 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 새도우

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

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

취소

5) AWS-IoT 설정

사물 생성4 : 새 인증서 자동 생성(권장) 선택

디바이스 인증서 구성 - 선택 사항 정보

디바이스가 AWS IoT에 연결하려면 인증서가 필요합니다. 지금 디바이스에 인증서를 등록하는 방법을 선택하거나 나중에 디바이스에 대한 인증서를 생성하고 등록할 수 있습니다. 디바이스는 적절한 정책이 적용된 활성 인증서가 있어야 AWS IoT에 연결할 수 있습니다.

디바이스 인증서

새 인증서 자동 생성(권장)
AWS IoT의 인증 기관을 사용하여 인증서, 퍼블릭 키 및 프라이빗 키를 생성합니다.

내 인증서 사용
자체 인증 기관에서 서명한 인증서를 사용합니다.

CSR 업로드
CA를 등록하고 하나 이상의 디바이스에 자체 인증서를 사용합니다.

지금은 인증서 생성 건너뛰기
이 사물에 대한 인증서를 생성하고 나중에 정책을 인증서에 연결할 수 있습니다.

취소

이전

다음

5) AWS-IoT 설정

사물 생성4 : 새 인증서 자동 생성(권장) 선택

CREATE A THING STEP 2/3

Add a certificate for your thing

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.

1 [Create certificate](#)

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

[Create with CSR](#)

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 **4** [Download](#)

5 [Activate](#)

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

5) AWS-IoT 설정

사물 생성5 : 인증서에 정책 연결

인증서에 정책 연결 - 선택 사항 정보

AWS IoT 정책은 AWS IoT 리소스에 대한 액세스를 허용하거나 거부합니다. 정책을 디바이스 인증서에 연결하면 이 액세스가 디바이스에 적용됩니다.

정책 (1/1)

이 인증서에 연결할 정책을 최대 10개까지 선택합니다.

정책 필터링 < 1 > ⚙️

<input checked="" type="checkbox"/>	이름
<input checked="" type="checkbox"/>	Arduino_AWS_Policy

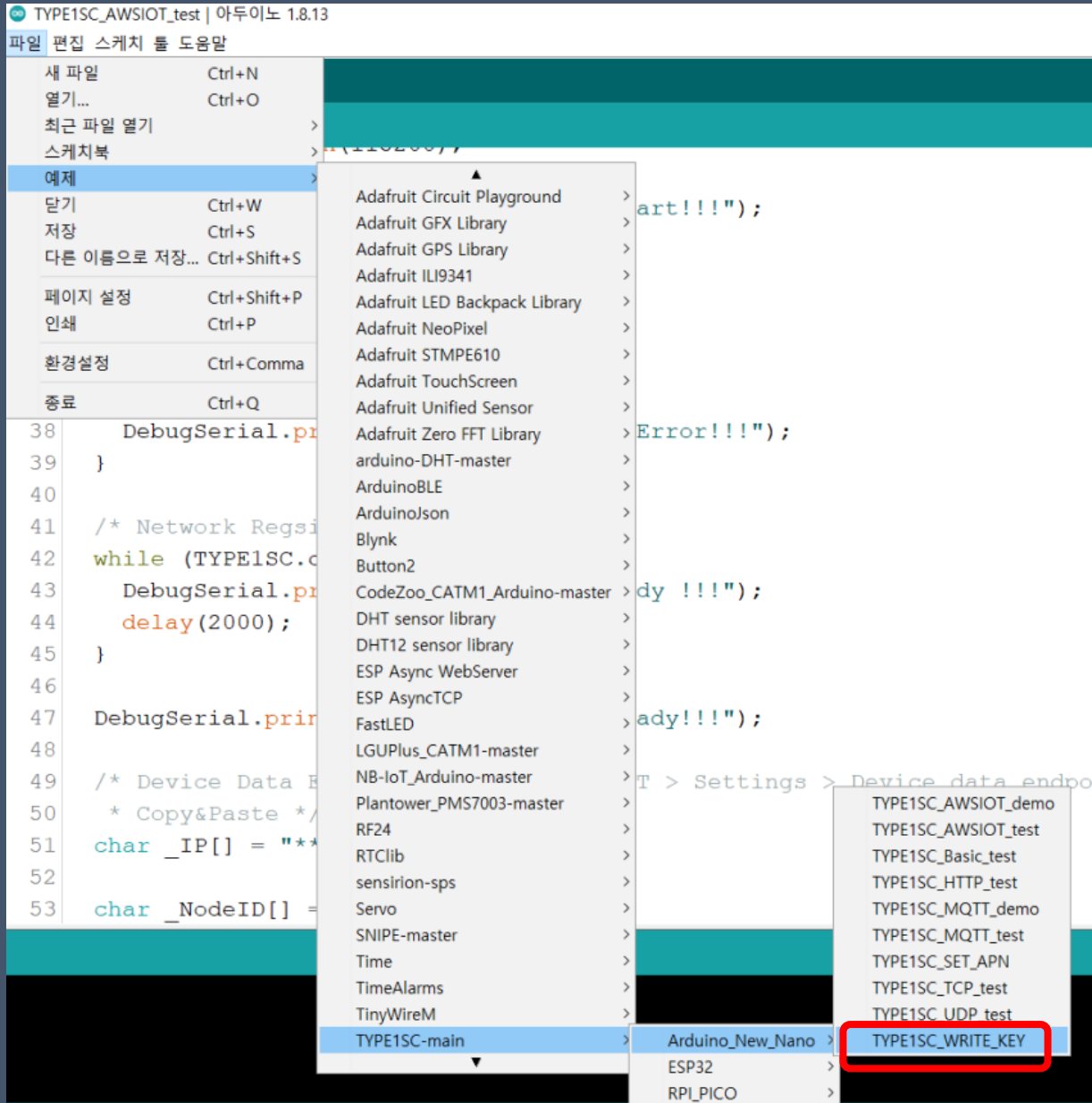
취소 이전 **사물 생성**

6) Write KEY (AWS-IoT 보안설정)

CAT.M1 모뎀 내부 파일 시스템에 AWS-IoT로 IoT 서비스에 접속하기 위한 보안 KEY를 저장합니다.

1. 사용하는 키는 AmazonRootCA 파일, certificate.pem.crt 파일, private.pem.key 파일 3가지 입니다.
2. 에디터 프로그램으로 해당 키를 열어서 예제에 추가 합니다.

6) Write KEY (AWS-IoT 보안설정)



✓ 예제 -> TYPE1SC-main -> Arduino_New_Nano -> TYPE1SC_WRITE_KEY 선택

6) Write KEY (AWS-IoT 보안설정)

```
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 "ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQWlhem9uMRkwFwYDVQQDExBBbWF6\n"
6 "b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTM4MDExNzAwMDAwMFowOTEL\n"
7 "MAkGA1UEBhMCVVMxMzANBgNVBAoTBkFtYXpvcjEzMDEwMDAwMDAwMDAwMDAwMDAw\n"
8 "b3QgQ0EgMTCCASIAwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj\n"
9 "ca9HgFB0fW7Y14h29Jl091ghYP10hAEvrAIthtOgQ3pOsqTQNroBvo3bSMgHFzZM\n"
10 "906II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw\n"
11 "IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRwa6\n"
12 "VOujw5H5SNz/0egwLX0tdHAL14gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L\n"
13 "93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GFID5yHI9Y/QCB/IIDEgEw+OyQm\n"
14 "jgSubJrIqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC\n"
15 "AYYwHQYDVR0OBBYEFIQYzIU07LwMlJQUcFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA\n"
16 "A4IBAQC8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI\n"
17 "U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxh1bI1Bjtt/msv0tadQ1wUs\n"
18 "N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKW1JbYK8U90vv\n"
19 "o/ufQJVtMVT8QtPHRh8jrdkPSHca2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU\n"
20 "5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy\n"
21 "rqXRfboQnoZsG4q5WTP468Sqvvg5\n"
22 "-----END CERTIFICATE-----";
23
24 const char clientCert[] = "-----BEGIN CERTIFICATE-----\n"
```

- ✓ aws_credentials.h 에 AmazonRootCA 파일, certificate.pem.crt 파일, private.pem.key 파일 내용을 줄 단위로 넣어 줍니다. const char 선언된 아래 이름은 내부 라이브러리에서 사용하기 때문에 임의로 변경하시면 안됩니다.
- ✓ 줄의 마지막에 \n 를 꼭 넣어주셔야 합니다.
 - rootCA : AmazonRootCA 파일
 - clientCert : certificate.pem.crt 파일
 - clientKey : private.pem.key
- ✓ AWS-IoT 보안키 파일은 Cat.M1 모뎀 내부 파일시스템에 안전하게 보관 되므로 매번 다시 넣어주지 않아도 됩니다. 단, 키가 변경되면 다시 Write Key 예제에 키를 넣어서 실행합니다.

6) Write KEY (AWS-IoT 보안설정)

```
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!!!
```

7) AWS-IoT Test, Demo

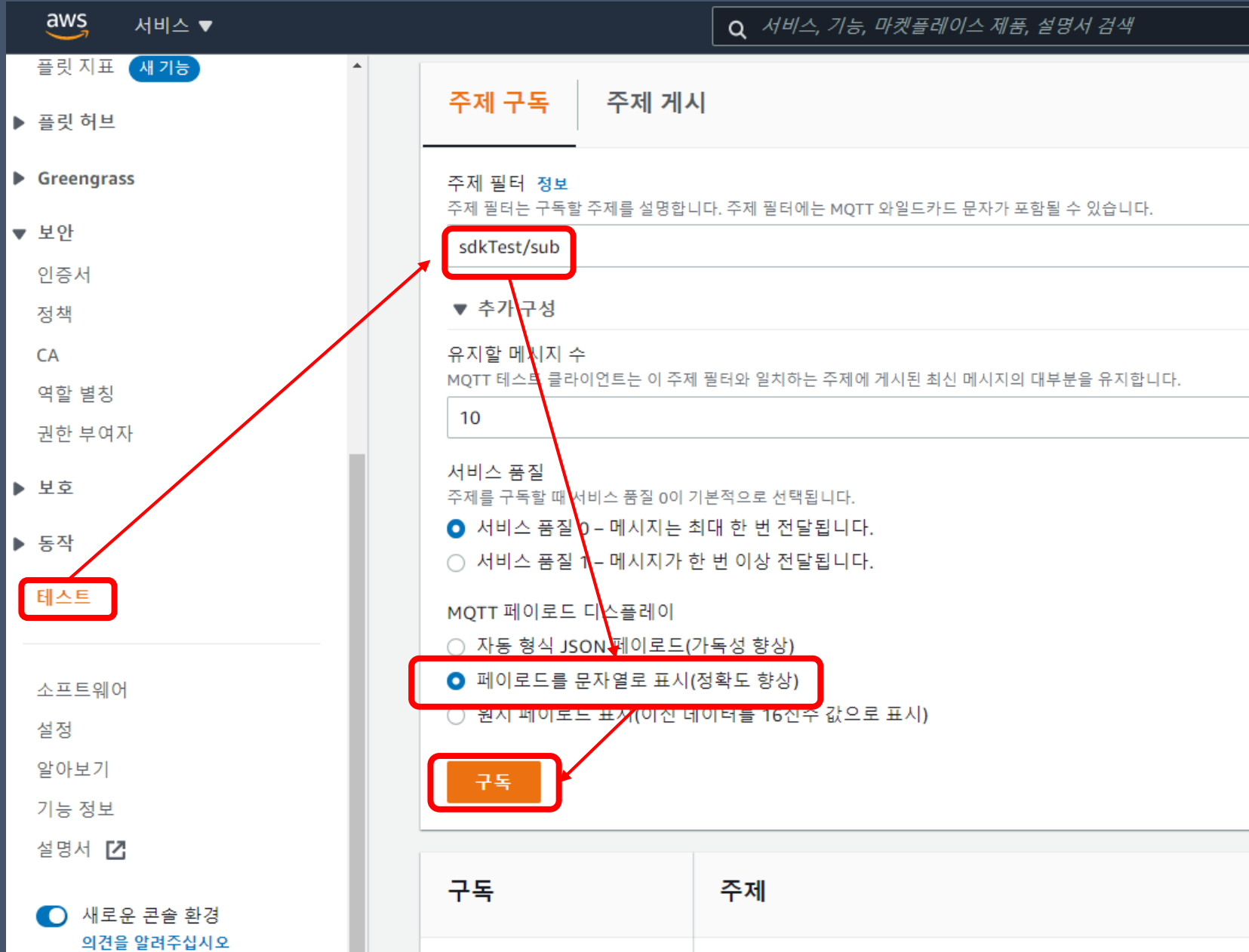
CAT.M1 모뎀 내부 AWS-IoT 통신 스택을 이용하여 아두이노에서 측정한 온도, 습도 데이터를 AWS IoT Core 로 업로드 합니다.

1. 4), 5) 과정을 통해 AWS-IoT를 사용할 준비가 되어 있어야 하고 CAT.M1 모뎀 내부에 AWS IoT Core에 접속하기 위한 보안 KEY가 Write 되어 있어야 합니다. 접속 주소는 디바이스 엔드포인트를 사용 합니다.

4) 마지막 내용 참고

2. AWS IoT Core 연동하기 위해 AWS 콘솔을 설정 합니다.

7) AWS-IoT Test, Demo



aws 서비스 ▾

플릿 지표 새 기능

▶ 플릿 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

▼ 추가구성

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)

원시 페이로드 표시(이전 데이터를 16진수 값으로 표시)

구독

구독	주제
----	----

7) AWS-IoT Test, Demo

```
47 DebugSerial.println("TYPE1SC Module Ready!!!");
48
49 /* Device Data EndPoint Address AWS IoT > Settings > Device data endpoint >
50 * Copy&Paste */
51 char _IP[] = "alpf!                .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 설정 -> 디바이스 엔드포인트 주소 입력

7) AWS-IoT Test, Demo

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 중지

지우기

내보내기

편집

sdkTest/sub



▼ sdkTest/sub

Temperature/26.0, Humidity/79.3

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

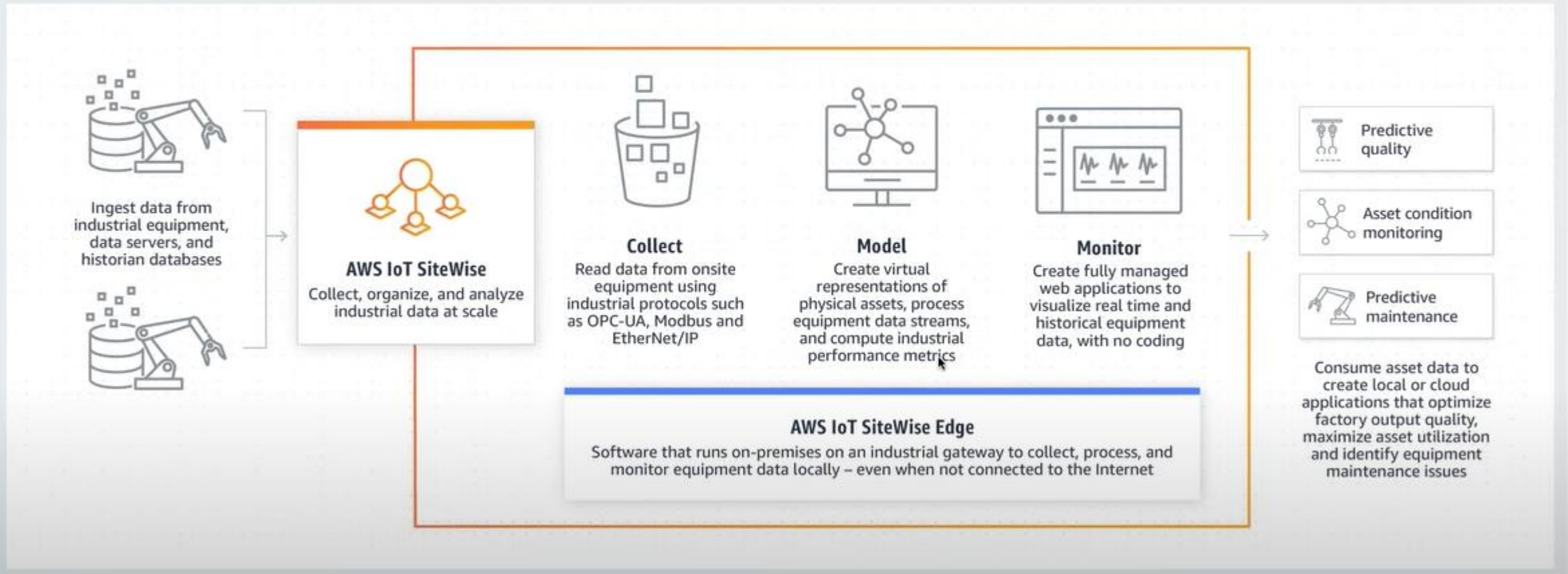
COM9

```
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
```

6. CAT.M1과 AWS IoT SiteWise를 활용한 서비스 구현

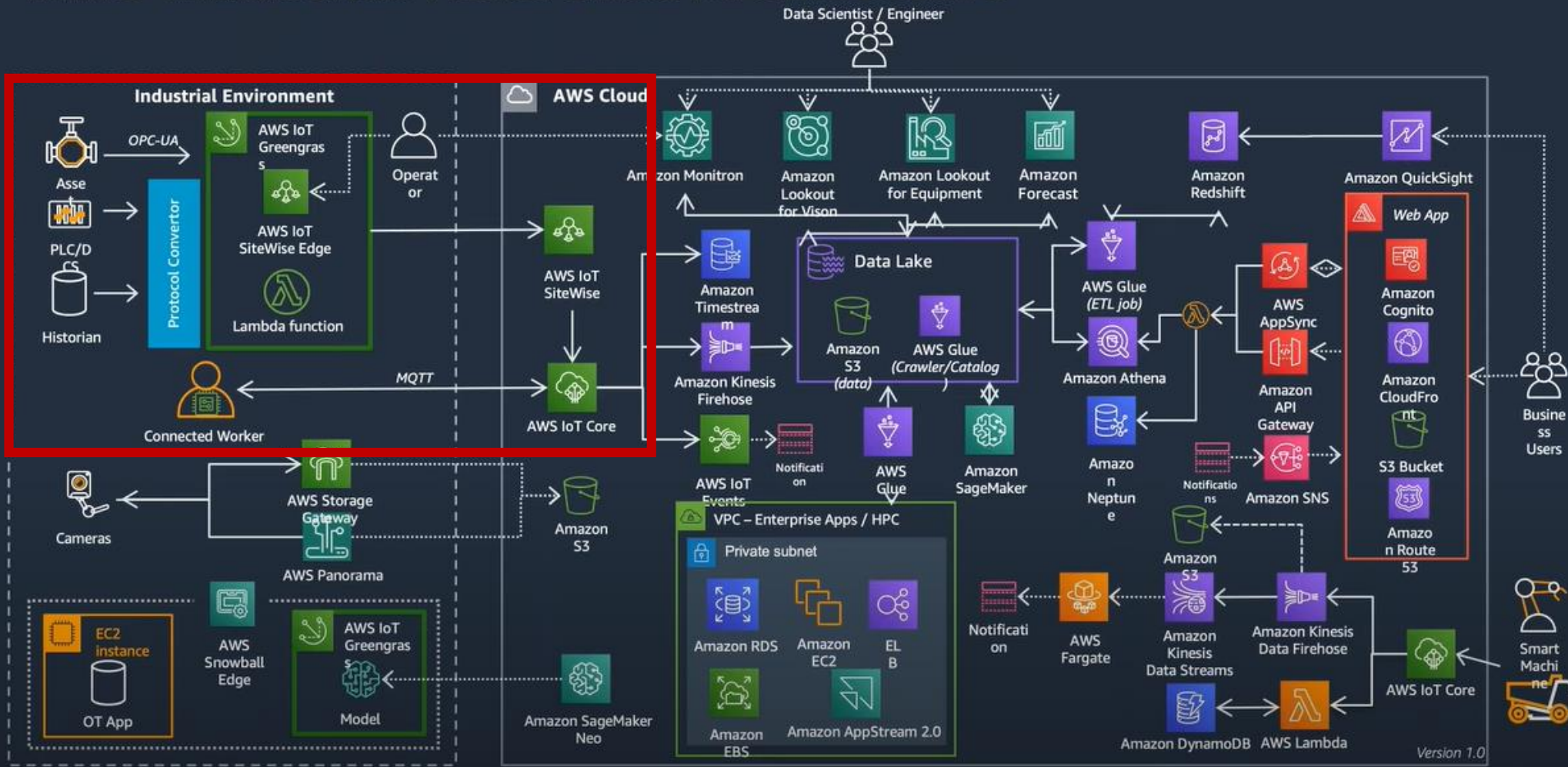


AWS IoT SiteWise

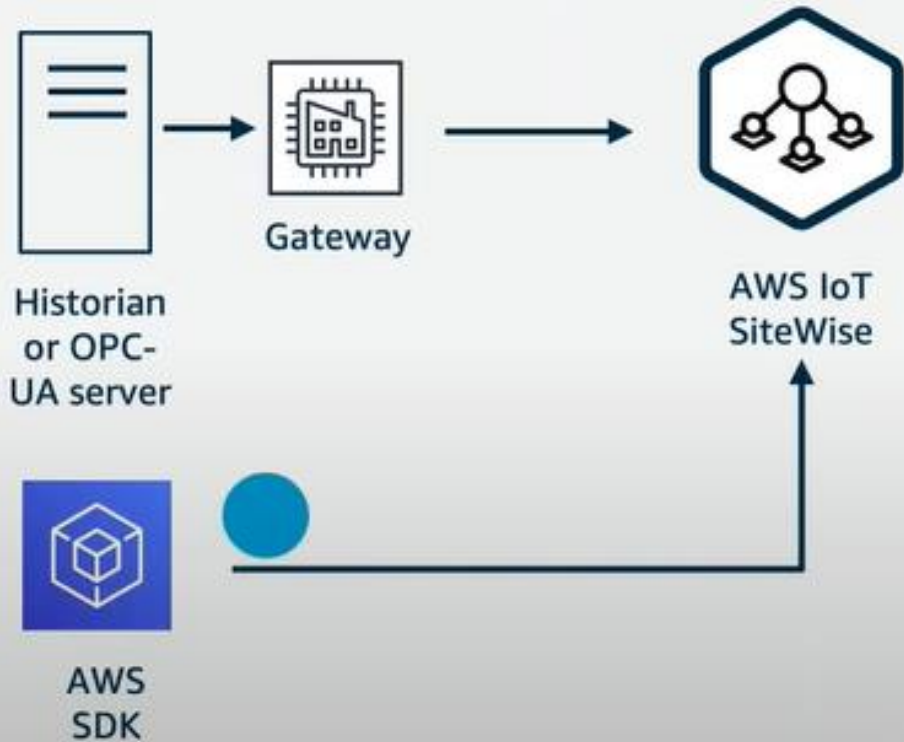
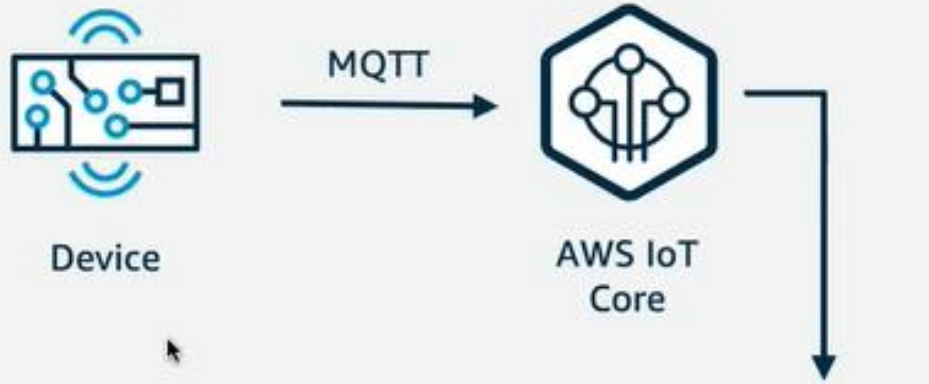


출처 : <https://aws.amazon.com/ko/iot-sitewise/>

AWS Industrial Reference Architecture



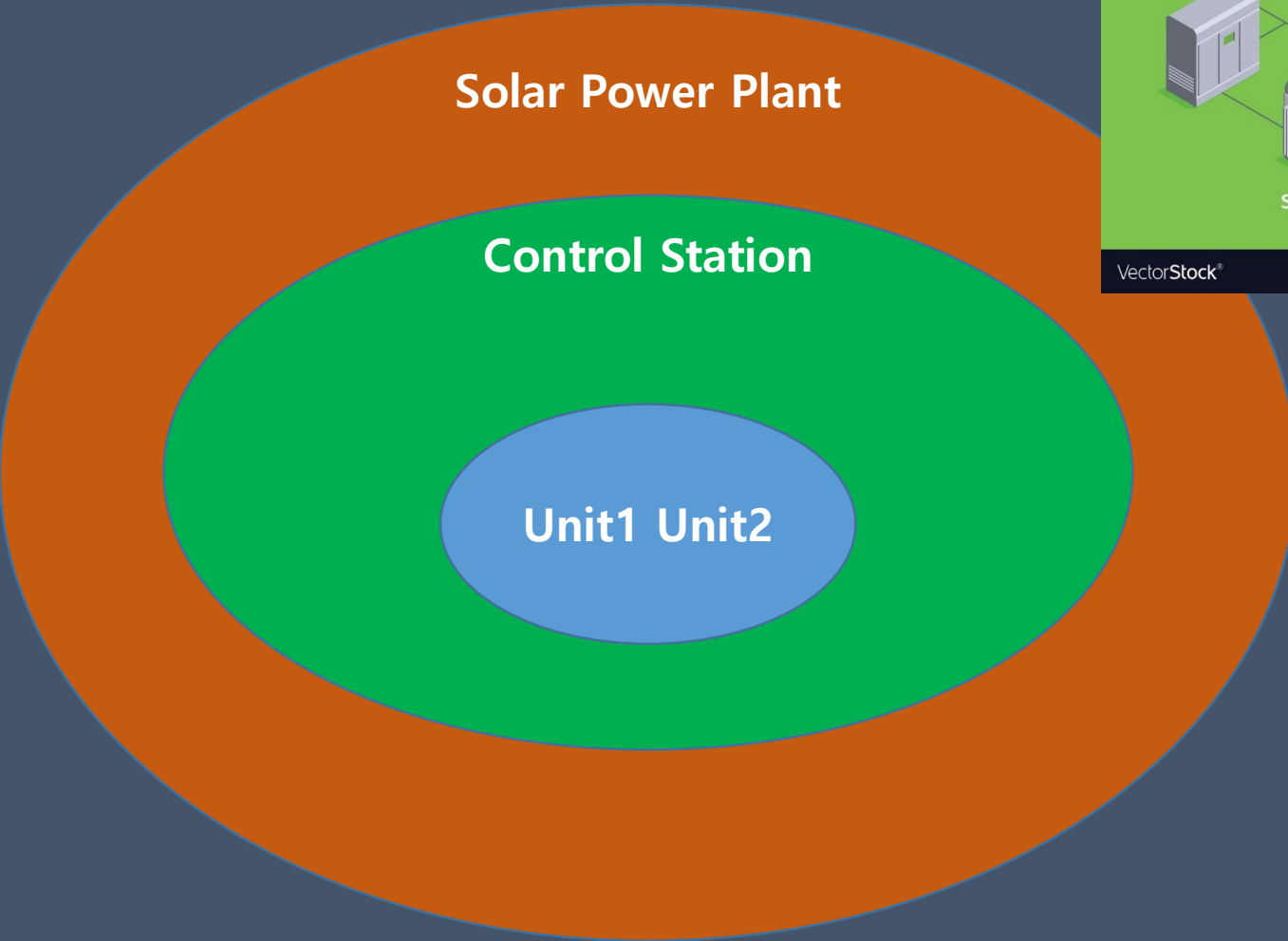
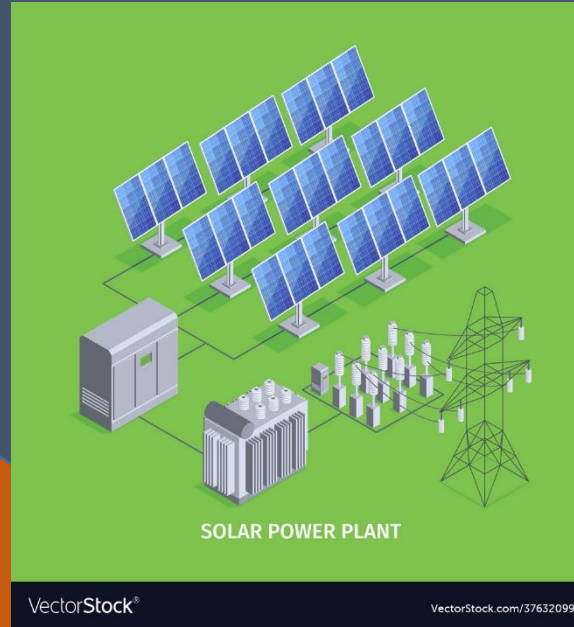
출처: AWS IoT SiteWise のご紹介
<https://youtu.be/2sQEDuCQLks>



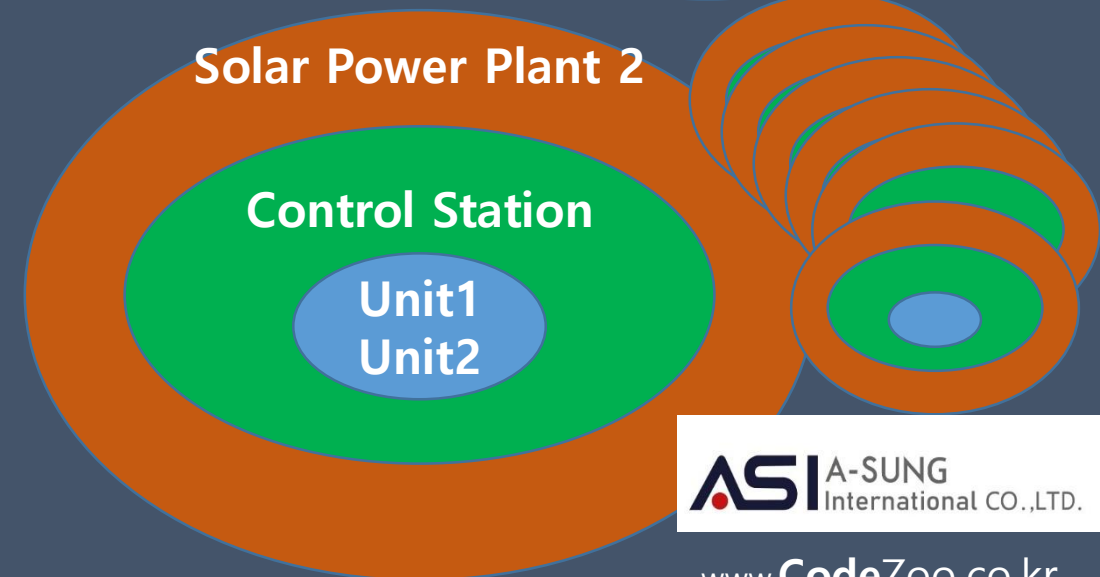
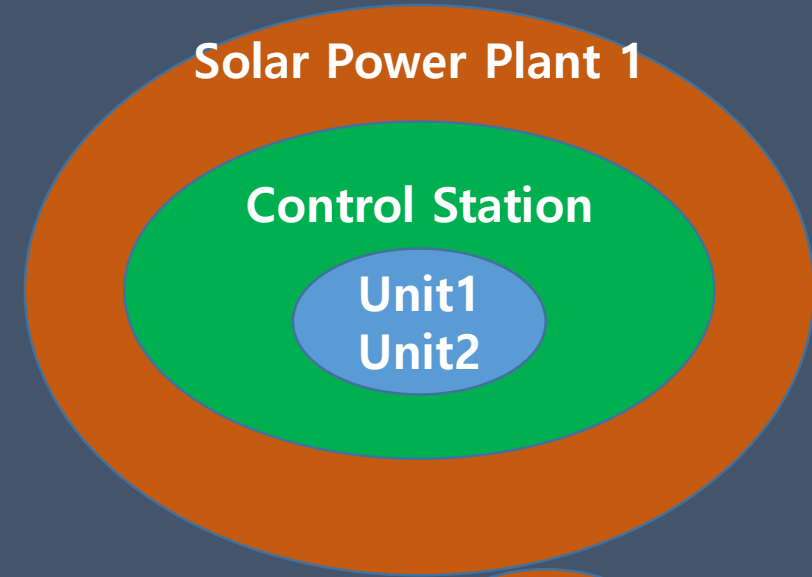
- ✓ AWS IoT SiteWise 관련 정보는 대부분 OPC-UA server + AWS GreenGlass 로 공개되어 있음
- ✓ OPC-UA Protocol 묶이지 않는 장비들의 경우 데이터를 MQTT로 AWS IoT Core 로 전송하여 AWS IoT SiteWise 서비스와 연동할 수 있음
- ✓ AWS IoT Core의 MQTT주제에 메시지를 게시하고 AWS IoT 규칙이 있는 데이터를 AWS IoT SiteWise에 전달

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

무엇을 만들 것인가? 모델, 자산 개념부터



모델 (Model)



자산 (Asset)

AWS IoT SiteWise

- ▼ 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

이름	상태	생성된 날짜
Demo Solar Power Plant Model	ACTIVE	2022년 2월 07일 17:47:03 (UTC+9:00)
Control Station Model	ACTIVE	2022년 2월 07일 17:44:22 (UTC+9:00)
Unit 2 Model	ACTIVE	2022년 2월 07일 17:40:52 (UTC+9:00)
Unit 1 Model	ACTIVE	2022년 2월 07일 17:38:00 (UTC+9:00)

새로운 기능 4

모델 정의

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](#)

정의변환 : Things에서 얻어 온 수치에 계산식을 더해서 필요한 데이터값을 생성
지표정의 : 집계 함수를 사용하여 지정된 시간 간격 동안의 데이터 포인트를 처리하는 수식
특정 기간 동안의 평균 온도와 같은 것을 입력하는 곳

AWS IoT SiteWise X

- ▼ 엣지
 - 게이트웨이
- ▼ 빌드
 - 모델
 - 자산**

자산 (1)

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.

 **자산 생성**

Q 최상위 수준 자산 필터링

< 1 > 

이름	상태	생성된 날짜	수정한 날짜
<ul style="list-style-type: none"> ☐ Demo Solar Power Plant 1 <ul style="list-style-type: none"> ☐ Control Station 1 <ul style="list-style-type: none"> Control Station 1 - Unit 1 Control Station 1 - Unit 2 	<ul style="list-style-type: none"> ☑ ACTIVE ☑ ACTIVE ☑ ACTIVE ☑ ACTIVE 	<ul style="list-style-type: none"> 2022년 2월 07일 18:03:38 (UTC+9:00) 2022년 2월 07일 18:00:11 (UTC+9:00) 2022년 2월 07일 17:49:50 (UTC+9:00) 2022년 2월 07일 17:51:15 (UTC+9:00) 	<ul style="list-style-type: none"> 2022년 2월 08일 16:18:09 (UTC+9:00) 2022년 2월 08일 16:17:35 (UTC+9:00) 2022년 2월 08일 20:33:35 (UTC+9:00) 2022년 2월 08일

새로운 기능 **4**

자산 정의

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 Station Model	Control Station 1	

AWS IoT SiteWise

SiteWise demo Solar Power Plant

- ▼ 엣지
 - 게이트웨이
- ▼ 빌드
 - 모델**
 - 자산
 - 데이터 스트림
- ▼ 모니터링
 - 시작하기
 - 포털**
- ▼ 설정
 - 로깅 옵션
 - 암호화
 - 스토리지

포털 세부 정보

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

권한

서비스 역할	엔터프라이즈 자격 증명 소스
AWSIoTSiteWiseMonitorServiceRole_28uMKybn	AWS SSO


추가 기능

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

새로운 기능 4

Demo Solar Power Plant 1

Last day Live 2022년 2월 15일 오전 2:47:43 | 2022년 2월 16일 오전 2:47:43 GMT+9 Edit

Dashboards 

Assets 

Models 

Projects 

Users 

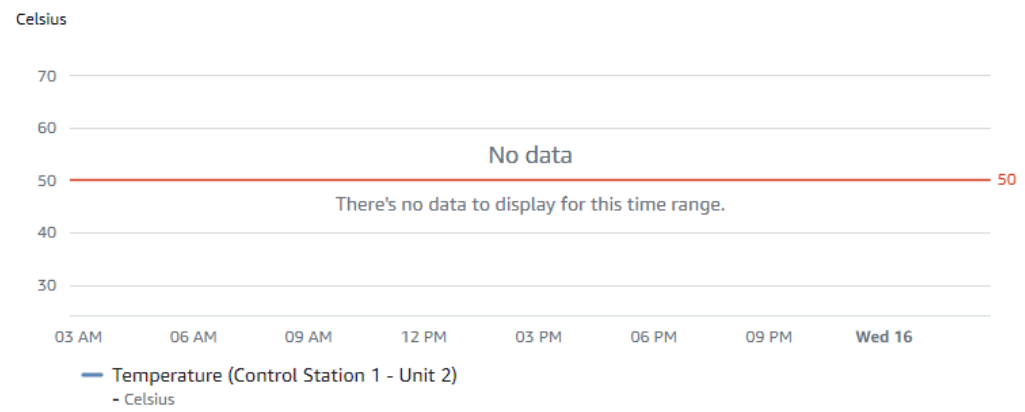
BYOUNG NAM JANG

English (US) >

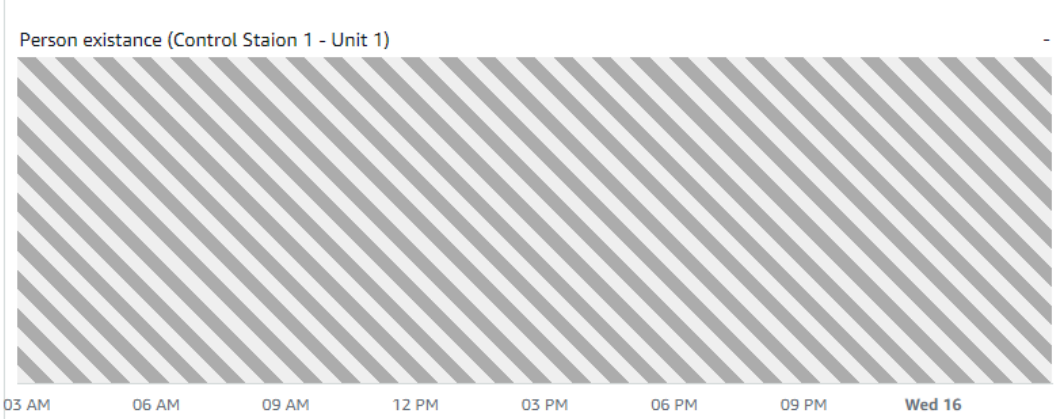
Help ?

Log Out 

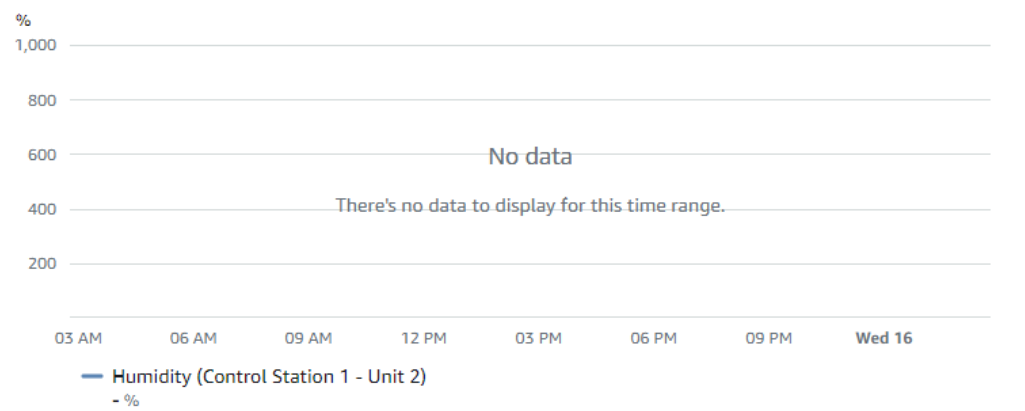
Temperature



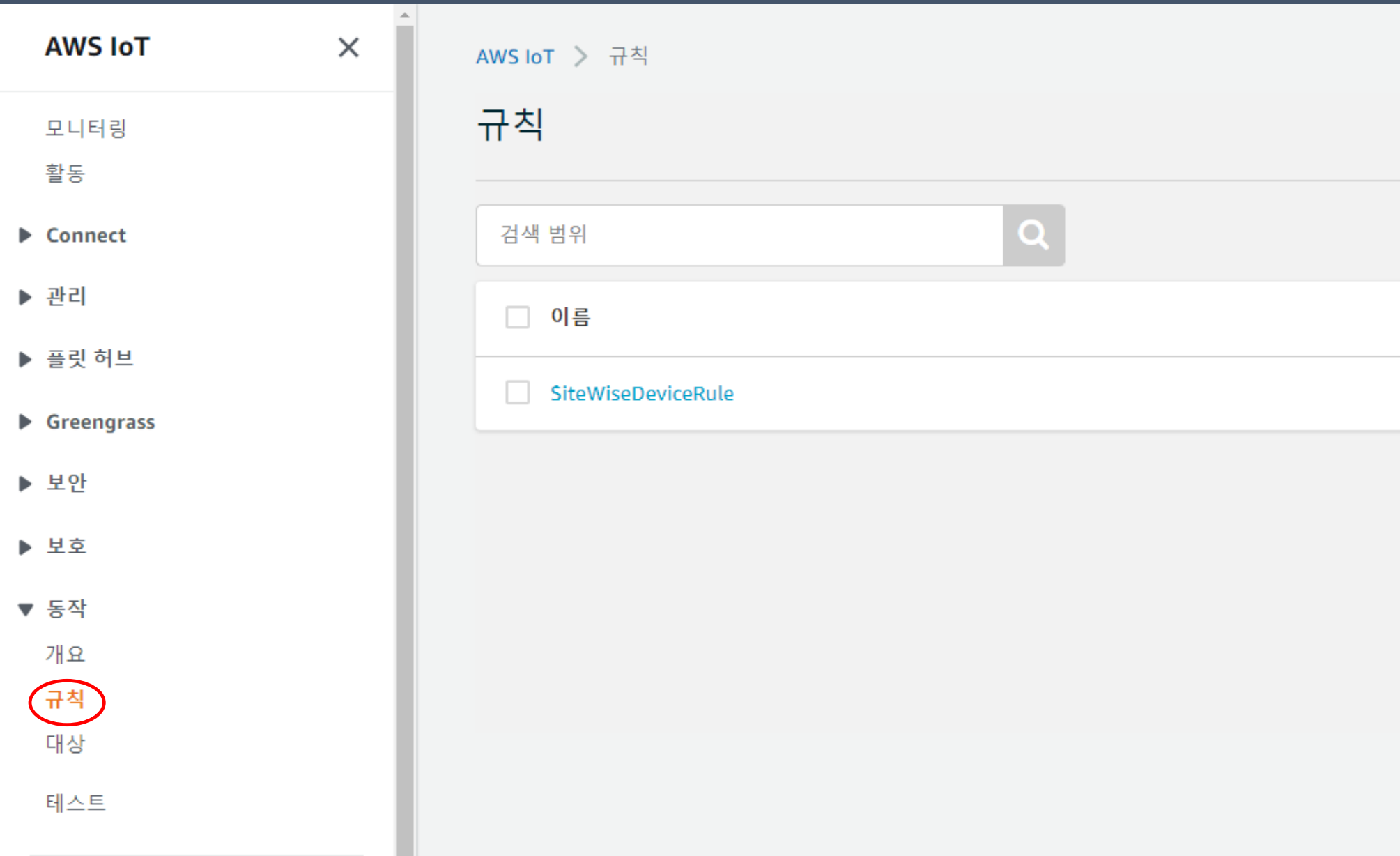
Person existence



Humidity



AWS IoT 규칙



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

21ae2c1c-5796-49fa-a00b0784eb46970

*속성 ID ?

87cad420-54fd-4cd2-b0b9-28af56757ca

항목 ID ?

예: \${entryId}

*시간(초) ?

\${timeInSeconds}

오프셋(나노초) ?

예: \${offsetInNanos}

*값 ?

\${Temperature}

*데이터 유형 ?

Double

*시간(초) [?](#)

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



LTE-CATM1 내장형 모뎀으로 Unix Timestamp 처...

카테고리 없음 · 2022. 2. 13. 16:53

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

[Read More](#)

*자산 ID [정보](#) *속성 ID [?](#)

Control Station 1 - Unit 2

자산 세부 정보

자산 ID <input type="text" value="21ae2c1c-5796-49fa-a00b-0784eb469703"/>	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
측정값 (2)						
이름 ▲	ID ▼	Alias ▼	MQTT 알림 상태 ▼			
Humidity	f35a464a-acdd-42c2-8d9e-e803a0bc2b52	/ControlStation1/ControlStation1-Unit2/ControlStationHumidity	⊖ 비활성화됨			
Temperature	87cad420-54fd-4cd2-b0b9-28af56757ca5	/ControlStation1/ControlStation1-Unit2/ControlStationTemperature	⊖ 비활성화됨			

속성 **측정값** 변환 지표 연결된 자산 경보 태그

AWS IoT

- 모니터링
- 활동
- ▶ Connect
- ▶ 관리
- ▶ 플릿 허브
- ▶ Greengrass
- ▶ 보안
- ▶ 보호
- ▼ 동작
 - 개요
 - 규칙
 - 대상
- 테스트**

구독

주제 구독

주제 게시

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

주제 게시

```

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

```

```

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

```

AWS IoT SiteWise

- ▼ 옛지
 - 게이트웨이
- ▼ 빌드
 - 모델
 - 자산**
 - 데이터 스트림
- ▼ 모니터링
 - 시작하기
 - 포털
- ▼ 설정
 - 로깅 옵션
 - 암호화
 - 스토리지

새로운 기능 4

Unit 2 Model

2022년 2월 07일 17:51:15 (UTC+9:00)

속성 | **측정값** | 변환 | 지표 | 연결된 자산 | 경보 | 태그

측정값 (2)

MQTT 알림 상태	알림 주제	최신 값	최신 값 타임스탬프
⊖ 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

SiteWise demo Solar Power Plant

- Dashboards
- Assets
- Models
- Projects
- Users

BYOUNG NAM JANG

English (US)

Help

Log Out

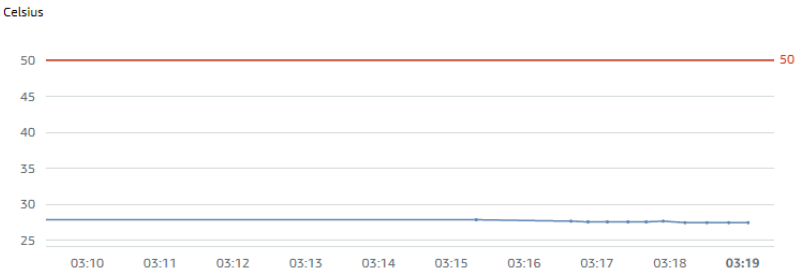
SiteWise demo Solar Power Plant > 프로젝트 > SiteWise Demo Project > Demo Solar Power Plant 1

Demo Solar Power Plant 1

Last 10 minutes Live

Temperature

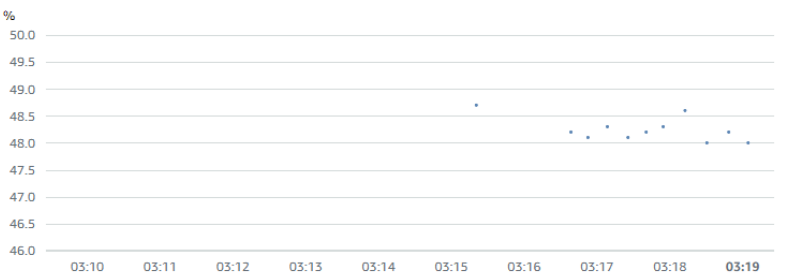
Celsius



Temperature (Control Station 1 - Unit 2)
27.4 Celsius

Humidity

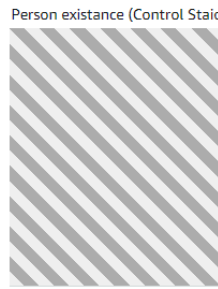
%



Humidity (Control Station 1 - Unit 2)
48%

Person existance

Person existance (Control Staio

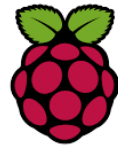
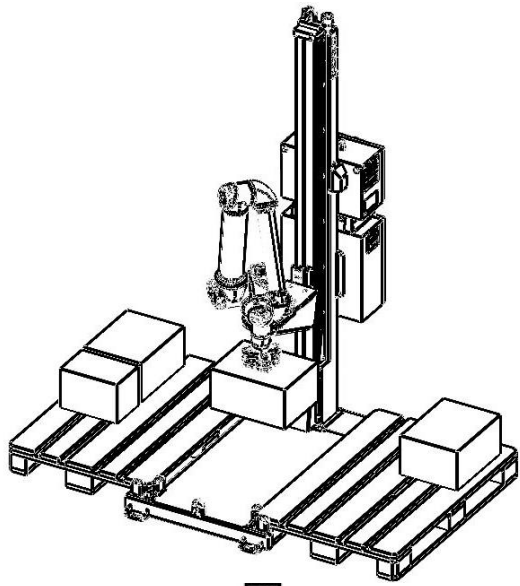


```
03:19:43.690 -> /=====/  
03:19:43.690 ->  
03:19:43.690 -> 1.Configure AWS_IOT parameter:ID, Address, tls Profile  
03:19:43.690 -> 2.Configure AWS_IOT parameter:Timeout  
03:19:43.690 -> 3.Enable AWS_IOT events  
03:19:52.495 -> 4.Establish connection  
03:19:55.040 -> 5.Subscribe to the topic 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
```

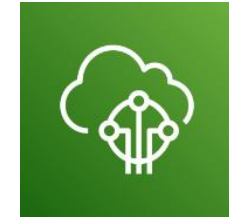
실제 구현사례 및 기술 소개 : 아성 협동로봇



협동로봇과 AWS IoT 연결



MQTT



AWS IoT Core



IoT SiteWise



로봇 상태에 대한 데이터 처리

Modbus TCP/IP

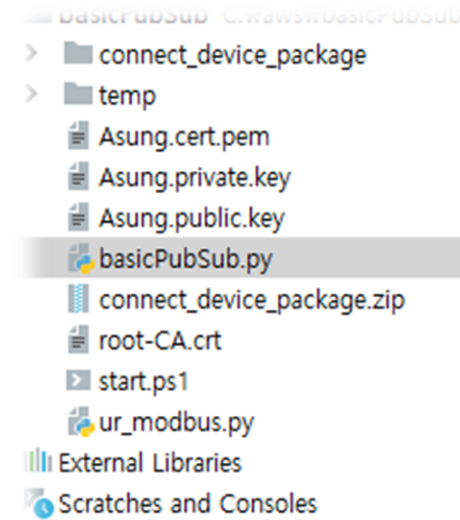
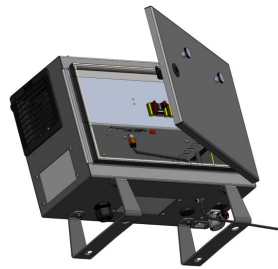


이미지 출처 : www.universal-robots.com



- ☑ 대다수 산업용 제어기기에서 사용됨
- ☑ 설치 유지보수가 비교적 쉬움
- ☑ 비트 또는 워드 단위로 데이터 조작
- ☑ 무료 공개 프로토콜
- ☑ 서버 - 클라이언트

파이썬 프로그램으로 로봇 데이터 처리



```

4 SERVER_PORT = 502
5
6
7 def modbustcp_client(host = IP
8     c = ModbusTcpClient(host,
9     return c
10
11 def connect(c):
12     if c.connect():
13         print("Modbus/TCP Conn
14         return True
15     else:
16         print("Modbus/TCP Conn
17         return False
18
19 def read_input_regs(c):

```

파이썬 프로그램으로 로봇 데이터 처리

```
while True:
    if args.mode == 'both' or args.mode == 'publish':
        t = datetime.utcnow()
        timeInSeconds = int((t-datetime(1970,1,1)).total_seconds())
        message = {}
        message['TimeStamp'] = timeInSeconds

        robot1 = ur.modbustcp_client(host="1[REDACTED]3")
        readInputRegisters = ur.read_input_regs(robot1)

        message['gp0'] = 1 if check_nth_bit(readInputRegisters[16].registers[0], 1) else 0
        message['gp1'] = 1 if check_nth_bit(readInputRegisters[16].registers[0], 1) else 0
        message['gp2'] = readInputRegisters[2].registers[0]
        message['gp3'] = readInputRegisters[3].registers[0]

        message['gp5'] = 0 if check_nth_bit(readInputRegisters[16].registers[0], 2) else 1
        message['gp6'] = readInputRegisters[6].registers[0]
        message['gp7'] = readInputRegisters[7].registers[0]
        message['gp8'] = readInputRegisters[8].registers[0]
        message['gp9'] = readInputRegisters[9].registers[0]
        message['tcp_x'] = tcp_xyz(readInputRegisters[10].registers[0])
        message['tcp_y'] = tcp_xyz(readInputRegisters[11].registers[0])
        message['tcp_z'] = tcp_xyz(readInputRegisters[12].registers[0])
        message['tcp_rx'] = tcp_rxyz(readInputRegisters[13].registers[0], to_deg=True) #
        message['tcp_ry'] = tcp_rxyz(readInputRegisters[14].registers[0], to_deg=True)
        message['tcp_rz'] = tcp_rxyz(readInputRegisters[15].registers[0], to_deg=True)
        message['Base_temp'] = readInputRegisters[19].registers[0]
        message['Shoulder_temp'] = readInputRegisters[20].registers[0]
        message['Elbow_temp'] = readInputRegisters[21].registers[0]
        message['Wrist1_temp'] = readInputRegisters[22].registers[0]
        message['Wrist2_temp'] = readInputRegisters[23].registers[0]
        message['Wrist3_temp'] = readInputRegisters[24].registers[0]
```

제품사양서, 개발자 매뉴얼 : <https://github.com/codezoo-ltd/TYPE1SC/tree/main/manual>

제품 회로, 도면 및 치수 자료 : https://github.com/codezoo-ltd/TYPE1SC/tree/main/schematics_dimension

오픈소스 예제 : <https://github.com/codezoo-ltd/TYPE1SC/tree/main/examples>

오픈소스 라이브러리 : <https://github.com/codezoo-ltd/TYPE1SC>

블로그 (세상에서 가장 작은 LTE Cat.M1 모듈) : <https://codezoo.tistory.com/>



Murata Type1SC 모듈 문의
(주)아성코리아 전지만이사 mlcc@asung.com



Type1SC 내장형 모뎀 문의
(주)코드주 장병남 대표
rooney.jang@codezoo.co.kr



www.CodeZoo.co.kr

감사합니다.