

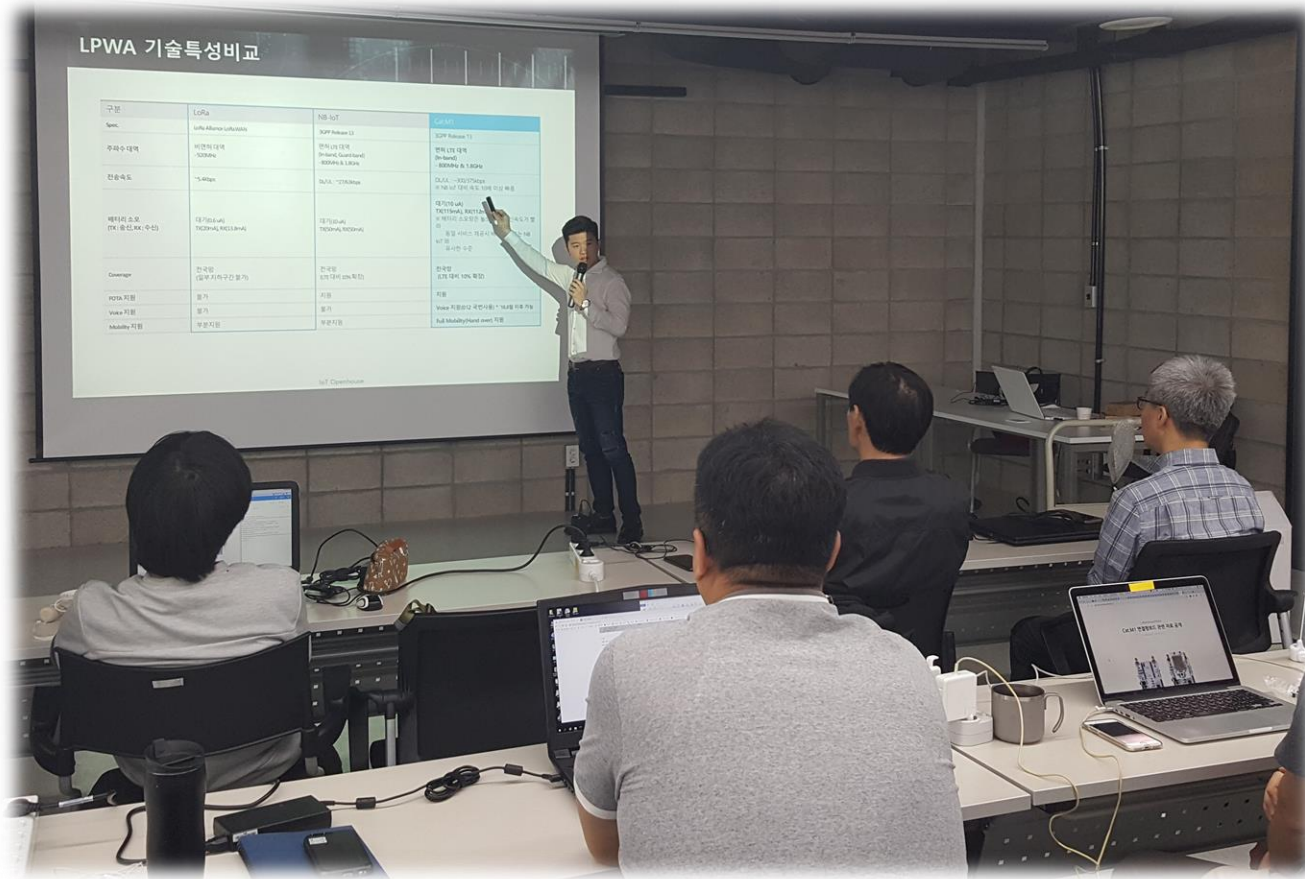
한국지능형사물인터넷협회 강좌

LTE CatM1 기반 사물인터넷 디바이스 플랫폼 연동 1-2

Arduino기반의 Cat.M1 디바이스 구현

김성태

tom@wiznet.io



김성태
Tom Kim

 tom@wiznet.io

INDEX

I. **Arduino 개발 환경 구축**

- HW Setting
- SW Setting

II. **F/W Library – AT Parser 란?**

- Parser Send()
- Parser Recv()
- Parser Flush()
- 공통 수행 절차 및 AT Parser 사용 예시

III. **Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)**

- LTE Cat.M1 Attach & Ping TEST
- SMS Send – CDS Triggering
- SMS Recv – Device Reset

IV. **Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (2)**

- MQTT 이론
- MQTT Local 테스트
- MQTT Publish – Temperature
- MQTT Subscribe – Device Reset

아두이노 환경 구축

- 1) H/W Setting
- 2) S/W Setting



개발 환경 설정하기

➤➤ Arduino Mega2560

- **MCU: Atmega2560**
- **Operating Voltage: 5V**
- **Digital/Analog Pin: 54/16**
- **SRAM/Flash: 8K/256K**
- **UART: 4**



| | |
|-----------------------------|---|
| Microcontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 54 (of which 15 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |
| LED_BUILTIN | 13 |
| Length | 101.52 mm |
| Width | 53.3 mm |
| Weight | 37 g |

개발 환경 설정하기

>> Arduino와 LTE 연결하기

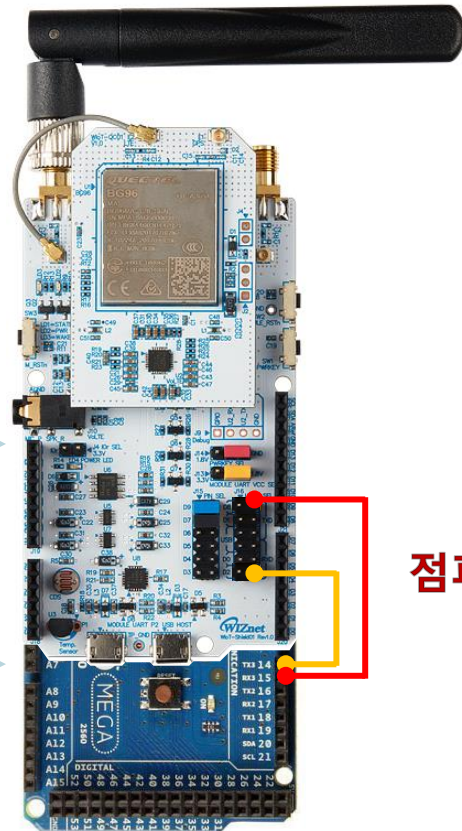
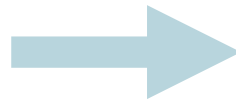
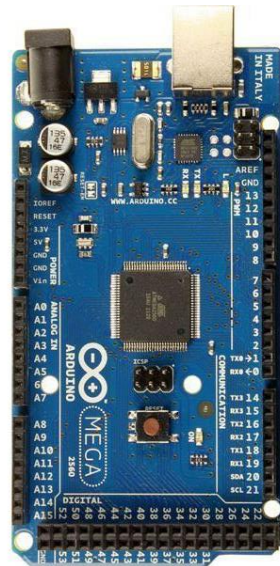
- 스타터 키트 점퍼캡 USB 부분 제거
- 아두이노 MEGA 위에 Start Kit Stacking
- 점퍼 케이블 연결
 - D8열 오른쪽(TXD) → Pin15(RX3)
 - D2열 오른쪽(RXD) → Pin14(TX3)



Stacking



점퍼캡 제거



점퍼 Wiring

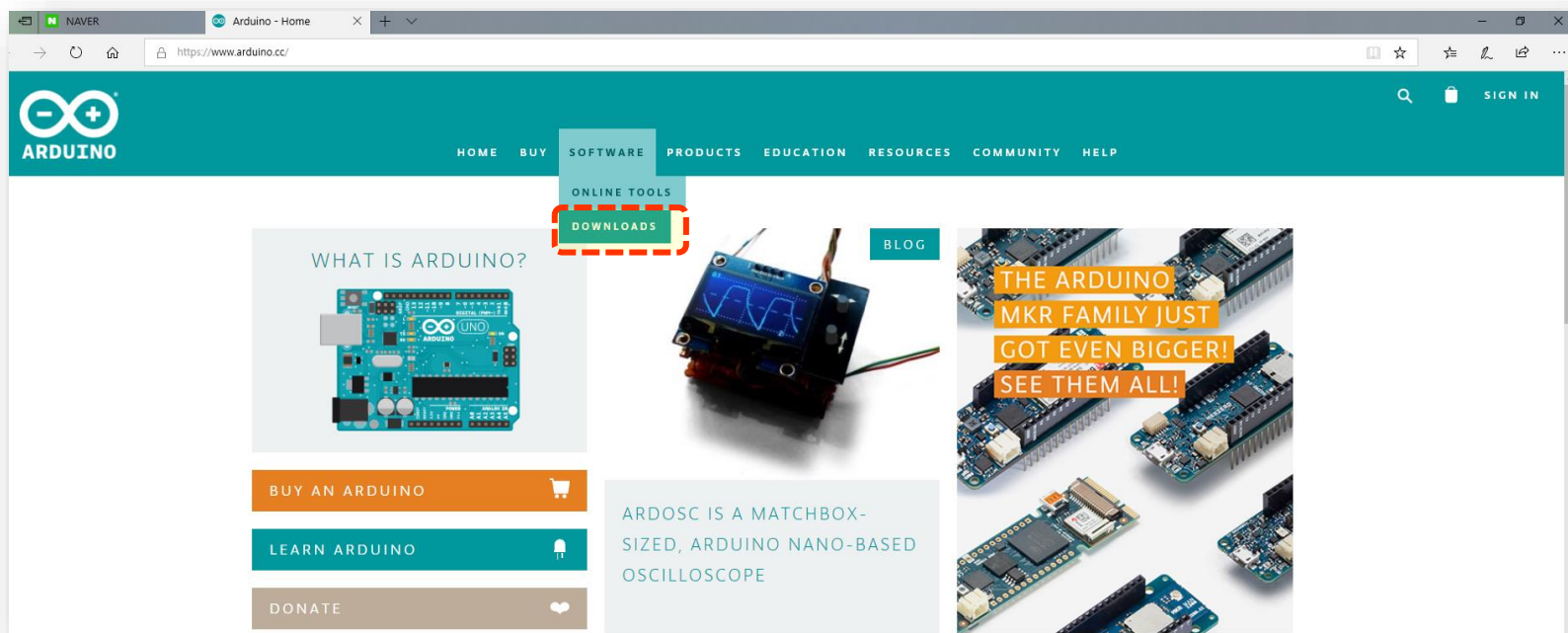
아두이노 환경 구축

- 1) H/W Setting
- 2) S/W Setting

개발 환경 설정하기

>> Arduino IDE 설치

- <https://www.arduino.cc/>



개발 환경 설정하기

>> Arduino IDE 설치

- <https://www.arduino.cc/>
- 강의 자료W2. Arduino기반의 Cat.M1 디바이스 구현WTool/
arduino-1.8.9-windows.exe 설치

Download the Arduino IDE



ARDUINO 1.8.7

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10

Get

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

HOURLY BUILDS

LAST UPDATE
20 September 2018 19:58:40 GMT

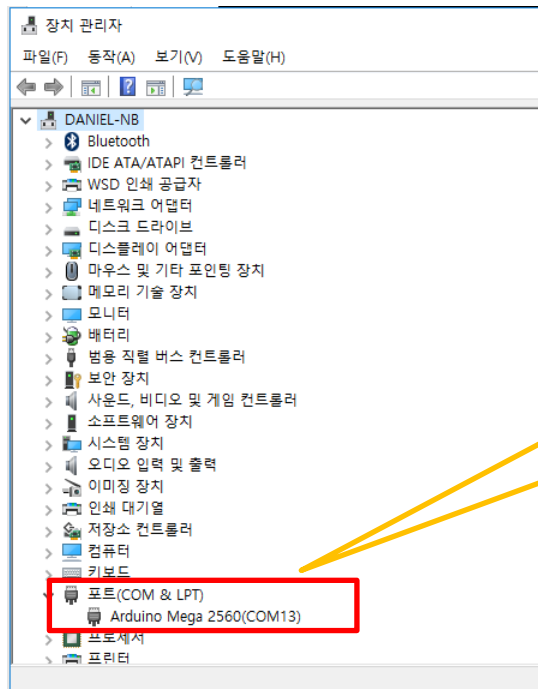
BETA BUILDS

BETA

개발 환경 설정하기

>> Arduino IDE Driver

- **Arduino를 연결** 하고, 장치관리자에서 확인
- 장치관리자는 내 컴퓨터 속성 → 장치관리자 클릭

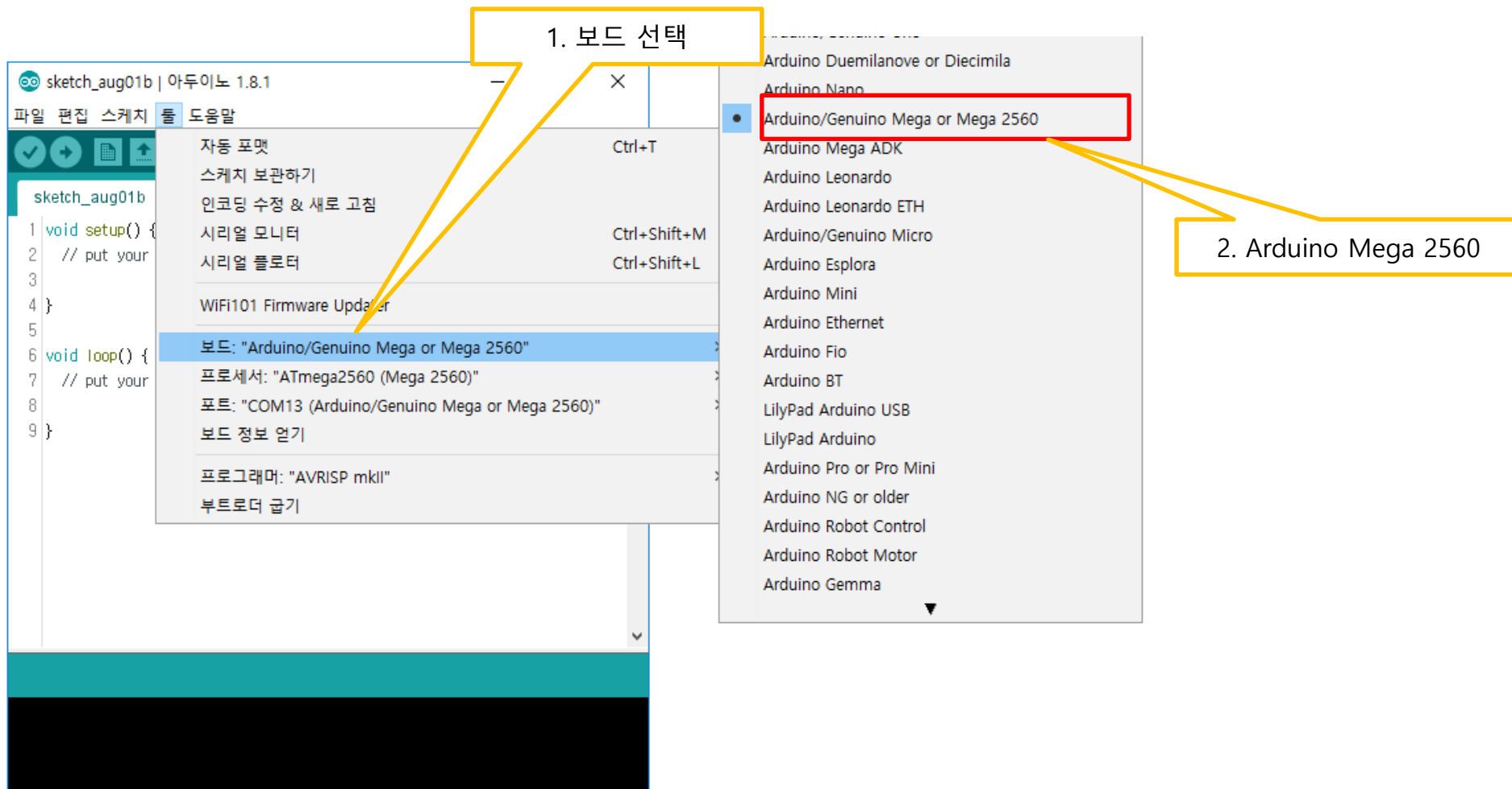


포트(COM & LPT)

Arduino Mega 2560(COM xx) 가 있으면 정상 인식

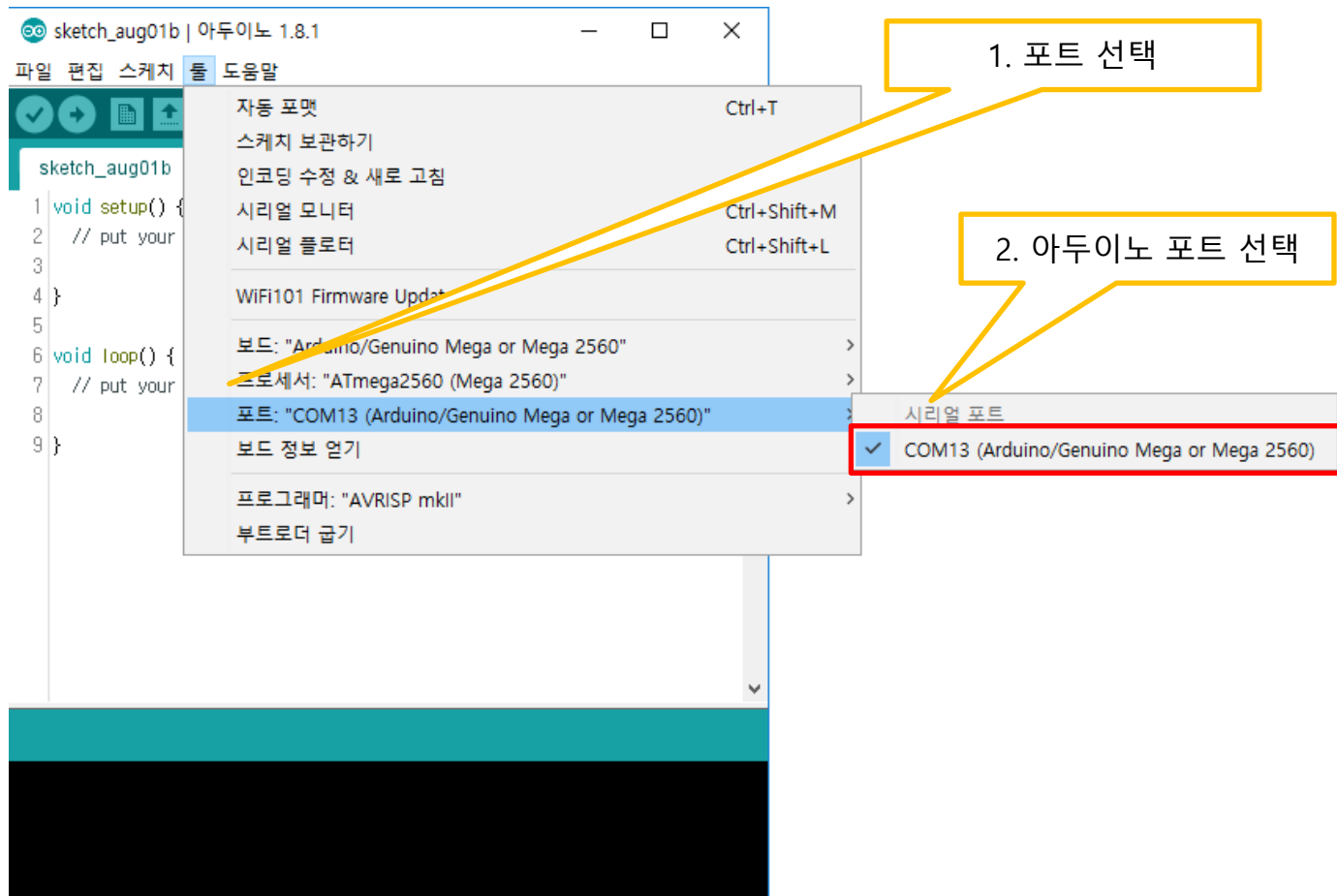
개발 환경 설정하기

>> Arduino 보드 선택



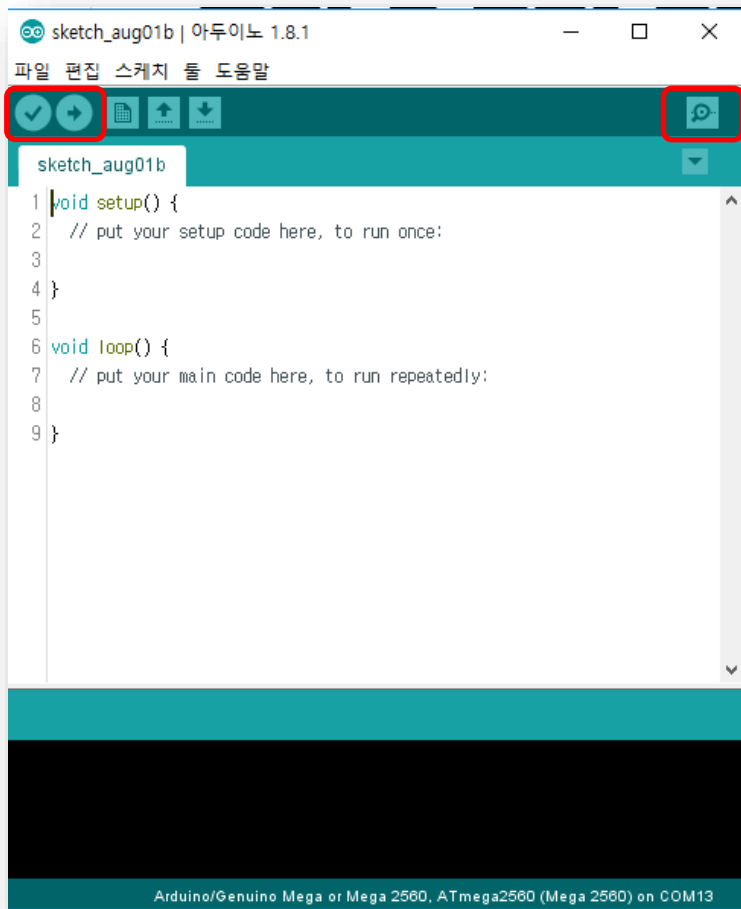
개발 환경 설정하기

>> Arduino가 연결된 포트 선택



개발 환경 설정하기

>> Arduino IDE 사용방법



확인:

내가 작성한 소스가 정상적으로 작성되었는지 확인!



업로드:

내가 작성한 소스를 디바이스에서 동작 가능하도록 업로드

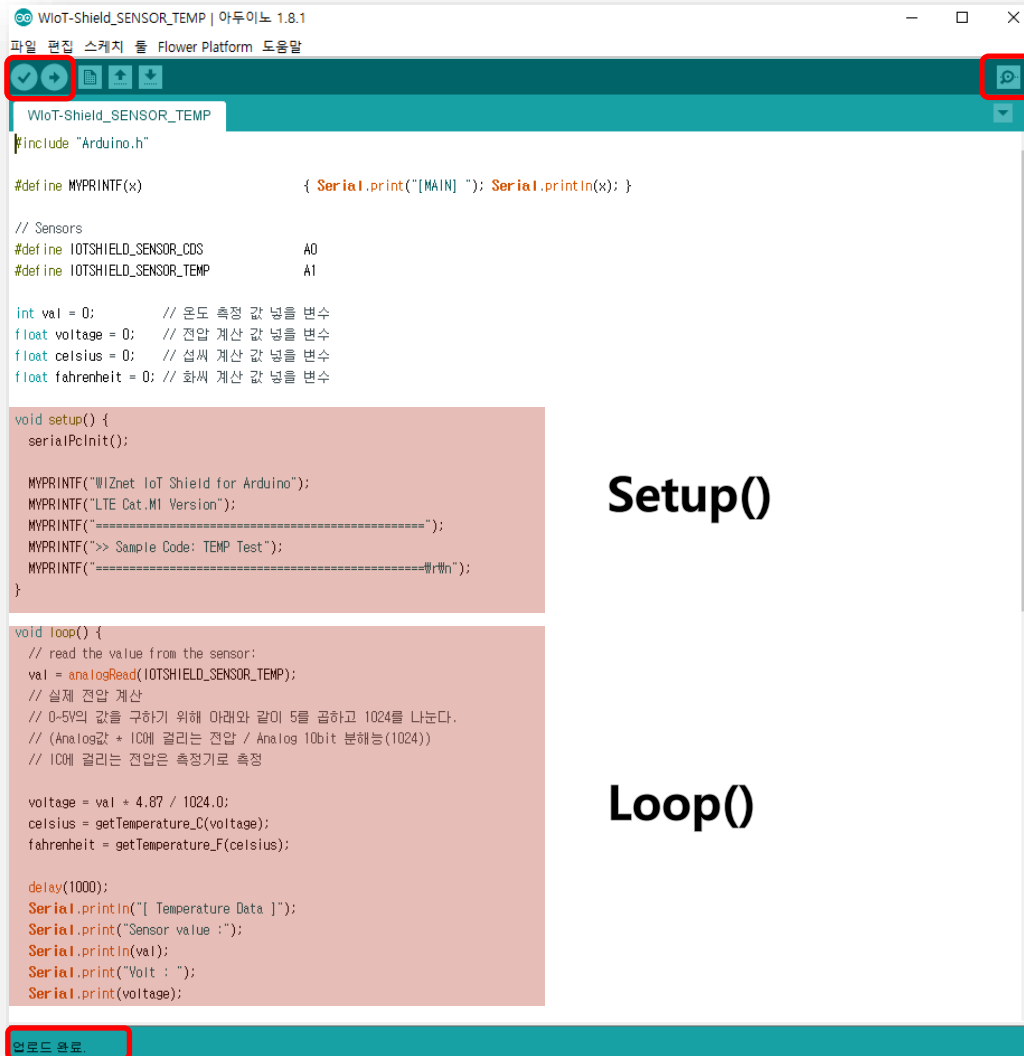


시리얼 모니터:

내가 작성한 소스가 잘 동작하는지 모니터링

개발 환경 설정하기

>> 강의자료\Example\WIoT-Shield_SENSOR_TEMP



```
WIoT-Shield_SENSOR_TEMP | 아두이노 1.8.1
파일 편집 스케치 툴 Flower Platform 도움말

WIoT-Shield_SENSOR_TEMP
#include "Arduino.h"

#define MYPRINTF(x)          { Serial.print("[MAIN] "); Serial.println(x); }

// Sensors
#define IOTSHIELD_SENSOR_CDS      A0
#define IOTSHIELD_SENSOR_TEMP    A1

int val = 0;          // 온도 측정 값 넣을 변수
float voltage = 0;    // 전압 계산 값 넣을 변수
float celsius = 0;    // 섭씨 계산 값 넣을 변수
float fahrenheit = 0; // 화씨 계산 값 넣을 변수

void setup() {
  serialPcInit();

  MYPRINTF("WIZnet IoT Shield for Arduino");
  MYPRINTF("LTE Cat.M1 Version");
  MYPRINTF("=====");
  MYPRINTF(">> Sample Code: TEMP Test");
  MYPRINTF("=====\\r\\n");
}

void loop() {
  // read the value from the sensor:
  val = analogRead(IOTSHIELD_SENSOR_TEMP);
  // 실제 전압 계산
  // 0-5V의 값을 구하기 위해 아래와 같이 5를 곱하고 1024를 나눈다.
  // (Analog값 * IC에 걸리는 전압 / Analog 10bit 분해능(1024))
  // IC에 걸리는 전압은 측정기로 측정

  voltage = val * 4.87 / 1024.0;
  celsius = getTemperature_C(voltage);
  fahrenheit = getTemperature_F(celsius);

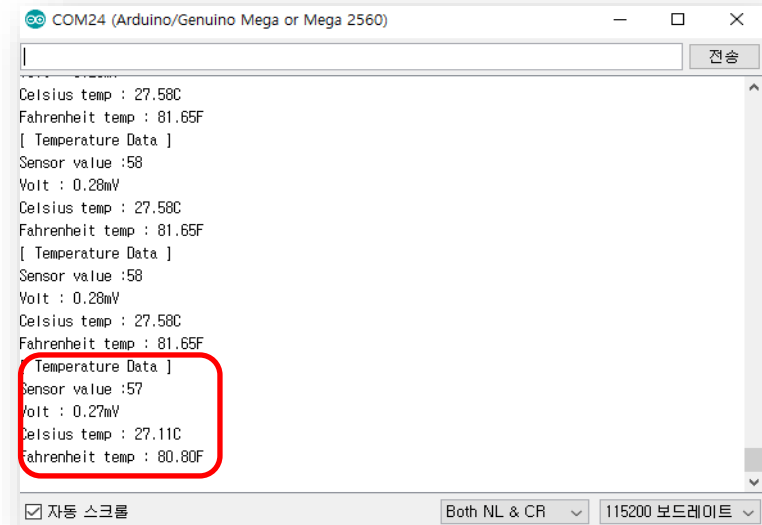
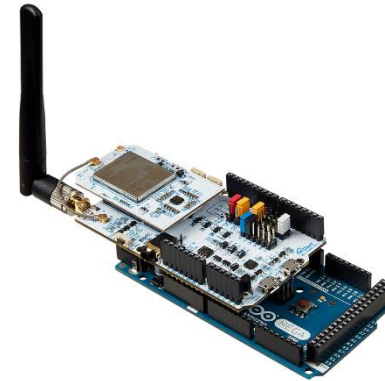
  delay(1000);
  Serial.println("[ Temperature Data ]");
  Serial.print("Sensor value :");
  Serial.println(val);
  Serial.print("Volt : ");
  Serial.print(voltage);
  Serial.print("Celsius temp : ");
  Serial.print(celsius);
  Serial.print("Fahrenheit temp : ");
  Serial.print(fahrenheit);
  Serial.println();
}
```

업로드 완료

Setup()

Loop()

실행 결과



```
COM24 (Arduino/Genuino Mega or Mega 2560)
전송

Celsius temp : 27.58C
Fahrenheit temp : 81.65F
[ Temperature Data ]
Sensor value :58
Volt : 0.28mV
Celsius temp : 27.58C
Fahrenheit temp : 81.65F
[ Temperature Data ]
Sensor value :58
Volt : 0.28mV
Celsius temp : 27.58C
Fahrenheit temp : 81.65F
[ Temperature Data ]
Sensor value :57
Volt : 0.27mV
Celsius temp : 27.11C
Fahrenheit temp : 80.80F

[ Temperature Data ]
Sensor value :57
Volt : 0.27mV
Celsius temp : 27.11C
Fahrenheit temp : 80.80F

자동 스크롤 Both NL & CR 115200 보드레이트
```

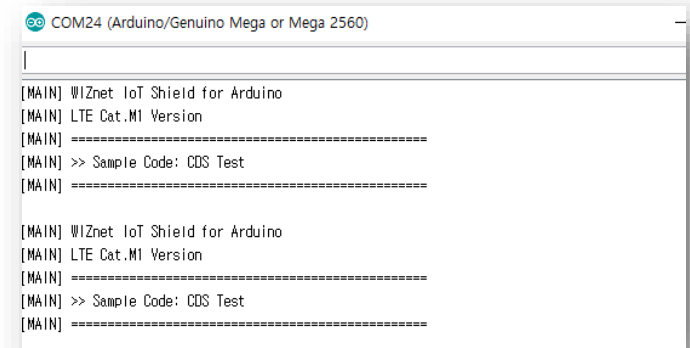
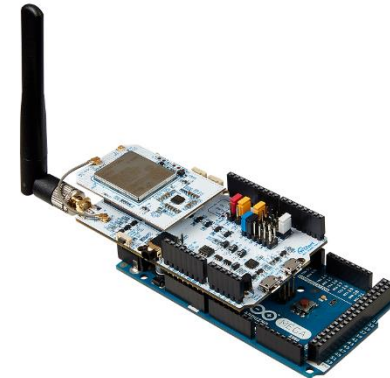

개발 환경 설정하기

>> 강의자료\Example\Wb_WIoT-Shield_SENSOR_CDS



실행 결과

- CDS를 가리면 MEGA LED 점등



AT Command Parser

- 1) Parser Send()
- 2) Parser Recv()
- 3) Parser Flush()
- 4) 공통 수행 절차 및 AT Parser 사용 예시

AT Command Parser

»» Arduino Serial 함수

- `Serial.println("AT")`
- `stringBuffer = Serial.readString()`
- `Strstr()` 함수를 이용해서 stringBuffer 중에 원하는 문자열이 있는지 체크



"AT"



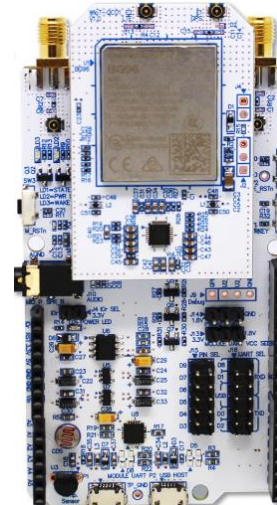
"OK"

"AT+CPIN?"



" +CPIN: READY"

⋮



AT Command Parser

» m_parser.send()

- m_parser.send("AT+CPIN?")



"AT"



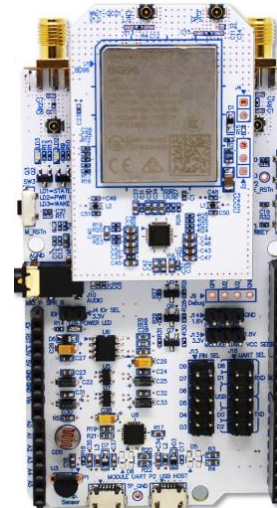
"OK"

"AT+CPIN?"



" +CPIN: READY"

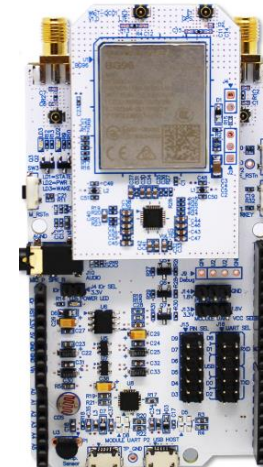
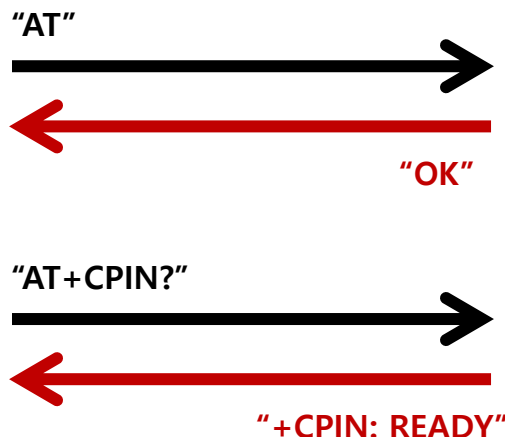
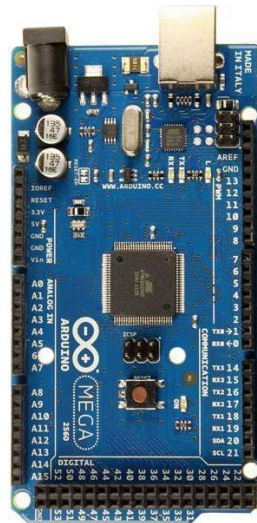
⋮



AT Command Parser

>> m_parser.recv()

- m_parser.recv(F("OK")) (Serial 데이터를 읽고 저장하고 비교하고!!)
- 대부분의 Connectivity 모듈은 AT Command 절차가 중요
 - Ex) 유심 개통 확인 → 망 접속 확인 → 데이터 송수신
- 만약, 모듈의 response를 확인하지 않고 시간에 따라 순서대로 진행 하면 망 접속이 안되었는데도 불구하고 데이터 송수신을 수행하는 경우 발생



⋮

AT Command Parser

>> F() 매크로

- `m_parser.send(F("AT+CPIN?"))`
- `m_parser.recv(F("OK"))`
- 문자열은 SRAM에 복사되어 사용됨
- F() 매크로를 사용하면 문자열을 SRAM에 복사 하지 않고, 플래시 메모리에 저장된 문자열에 접근하여 사용할 수 있음

| Arduino | Processor | Flash | SRAM | EEPROM |
|--|-------------|-------|------|--------|
| UNO, Uno Ethernet, Menta, Boarduino | Atmega328 | 32K | 2K | 1K |
| Leonardo, Micro, Flora, 32U4 Breakout, Teensy, Esplora | Atmega 32U4 | 32K | 2.5K | 1K |
| Mega, MegaADK | Atmega2560 | 256K | 8K | 4K |

AT Command Parser

» m_parser.flush()

- m_parser.recv("BG96") 하고 버퍼에 남은 Serial 데이터를 모두 비움
- Ex) 공공 API 서버에 날씨를 요청하고 많은 양의 데이터 중 필요한 데이터만 추출하고 나머지는 Flush() 함수로 버퍼에 남은 Serial 데이터를 비움



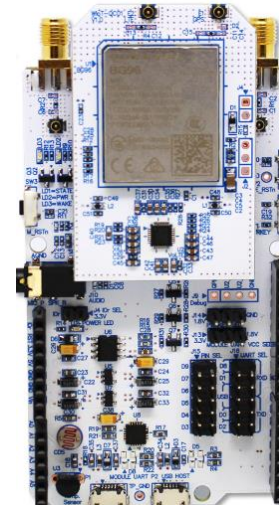
"ATI"



""
Quectel
BG96
Revision:
BG96MAR02A08M1G

OK

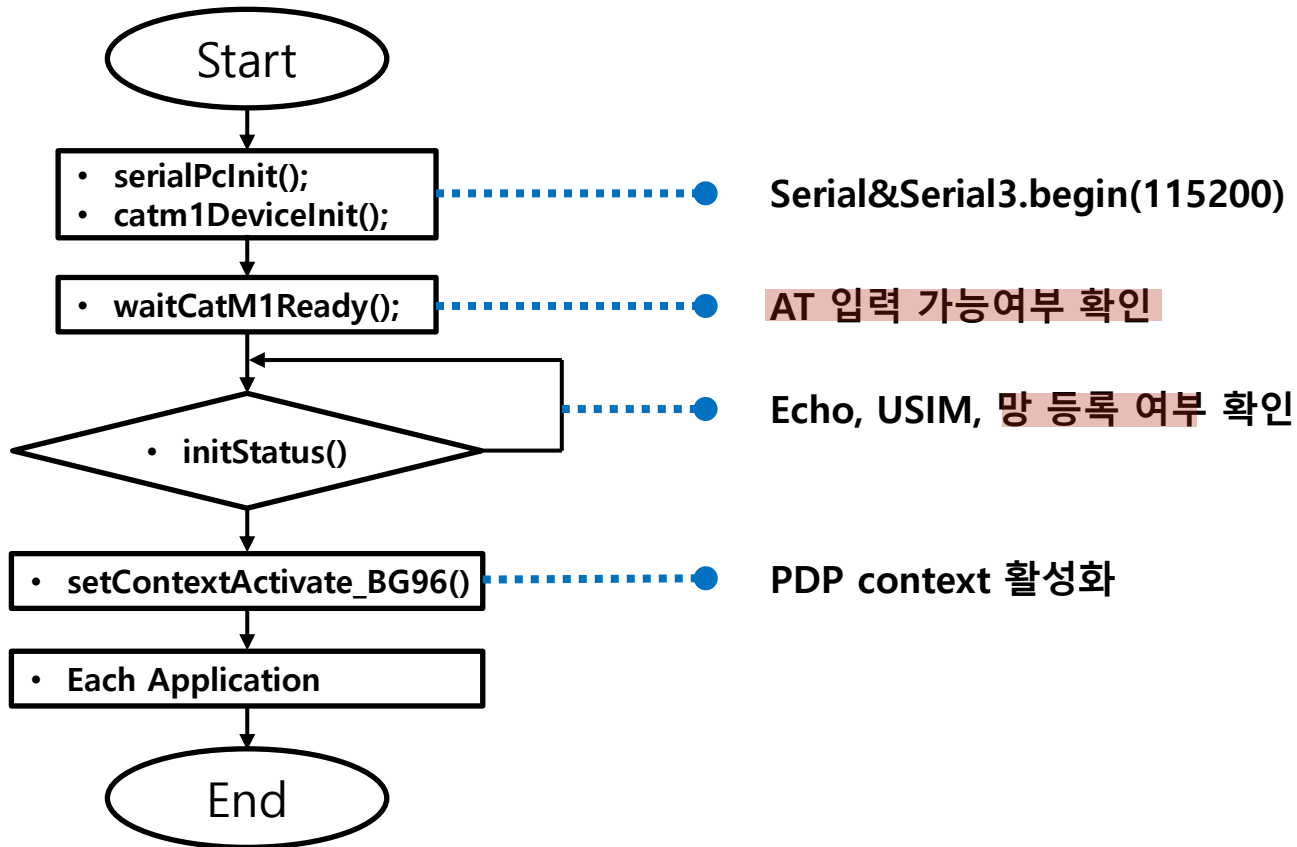
""



⋮

AT Command Parser

>> 공통 수행 절차



• Setup() 에 구현
→ Initialize는 한번만
수행하면 되기 때문에!

AT Command Parser

>> waitCatM1Ready()

- 가장 기본적인 형태 → if(명령어 && 응답)

```
int8_t waitCatM1Ready()
{
    while (1)
    {
        if (m_parser.recv(F("RDY"))) {
            MYPRINTF("BG96 ready\r\n");
            return RET_OK;
        }
        else if (m_parser.send(F("AT")) && m_parser.recv(F("OK")))
        {
            MYPRINTF("BG96 already available\r\n");
            return RET_OK;
        }
    }
}
```

AT Command Parser

» getNetworkStatus_BG96()

- 명령어에 대한 응답의 규칙이 있다면, 다음과 같이 문자열 추출 가능

```
int8_t getNetworkStatus_BG96(void)
{
    char mode[10], stat[10];
    char buf[10];

    if ( m_parser.send(F("AT+CEREG?")) &&
        m_parser.recv(F("+CEREG: %[^\n],%[^\n]\n"), mode, stat) &&
        m_parser.recv(F(ESP_OK)) ) {

        if ( (atoi(mode) == 0) && (atoi(stat) == 1) ) {
            LOGDEBUG("Network Status: Attach\r\n");
            return RET_OK;
        }

        else if (( atoi(stat) != 1 )) {
            sprintf((char *)buf, "Network Status: %d, %d", atoi(mode), atoi(stat));
            LOGDEBUG(buf);
            return RET_NOK;
        }
    }
}
```

"%[^a]": a라는 글자 전까지 추출

"%[abc]": abc문자만 추출

"%[a-z]": a-z문자만 추출

"%[0-9A-Za-z]": 대소문자 및 숫자만 추출

```
AT+CEREG?
+CEREG: 0,1
OK
```

"Attached"

```
AT+CEREG?
+CEREG: 0,0
OK
```

"Not Attached"

AT Command Parser

➤➤ `getNetworkStatus_BG96()`

- QUIZ 1

- 그렇다면, 다음과 같은 응답을 받는 명령어는 어떻게 처리할 수 있을까?

```
AT+QCDS
+QCDS: "SRV", "CAT-M", 45012, 2500, 13BD0B, R13, 3, 368, 2058, -83, -106, -6, 3, 128, 0, 0, 0
OK
```

```
AT+QCDS
+QCDS: "NO_SRV"

OK
```

- **HINT**

- if (m_parser.send(F(" ")) &&
m_parser.recv(F(" + QCDs: % ", mode) &&
m_parser.recv(F(RESP_OK)))
- 문자열 비교 함수는 strcmp() 함수를 사용
- “ ”안에 “를 삽입하려면, w” 으로 표현 가능

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

- 1) **LTE Cat.M1 Attach & Ping TEST**
- 2) SMS Send – CDS Triggering
- 3) SMS Recv – Device Reset



Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

» LTE Cat.M1 Attach & Ping TEST

- 강의 자료 Example Wc_WIoT-QC01_Arduino_Ping

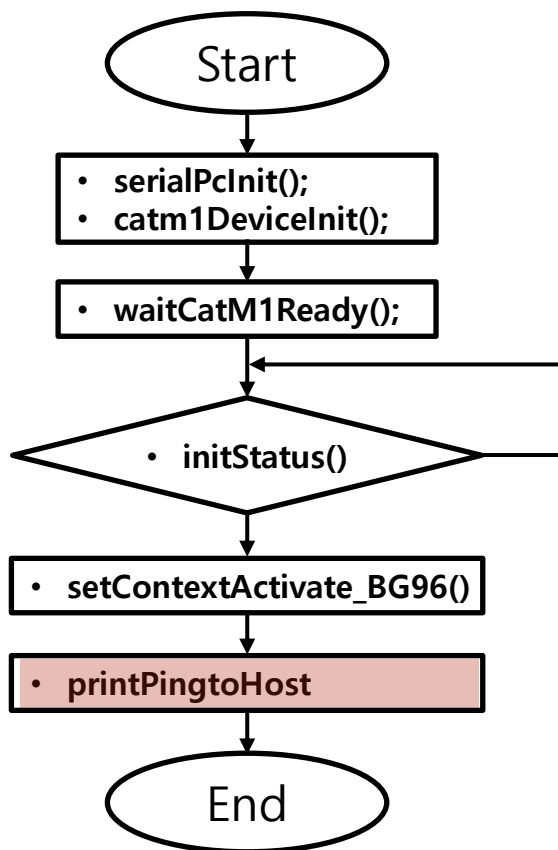
The screenshot displays the Arduino IDE interface with the 'WIoT-QC01_Arduino_Ping' sketch loaded. The sketch includes headers for 'Arduino.h' and 'at_cmd_parser.h'. It defines various macros for response codes, debug levels, and network settings. The main function initiates an APN check, PDP context activation, a ping to 'www.google.com', and PDP context deactivation. The serial monitor shows the following output:

```
[BG96] APN Check Done
[BG96] PDP Context Activation: Success
[MAIN] [Ping] Host: www.google.com
[BG96] www.google.com: 0, "2404:6800:4004:806:0:0:0:2004", 32, 172, 255
[BG96] www.google.com: 0, "2404:6800:4004:806:0:0:0:2004", 32, 137, 255
[BG96] www.google.com: 0, "2404:6800:4004:806:0:0:0:2004", 32, 161, 255
[BG96] www.google.com: 0, "2404:6800:4004:806:0:0:0:2004", 32, 159, 255
[BG96] PDP Context Deactivation: Success
```

The bottom status bar indicates 'Both NL & CR' and '115200 보드레이트'.

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

» LTE Cat.M1 Attach & Ping TEST



```
void printPingToHost_BG96(char * host, int timeout, int pingnum)
{
    char buf[100];
    int i;
    char resp_str[100] = {0, };

    if ((timeout < 1) || (timeout > 255)) {
        LOGDEBUG("Ping timeout range is 1-255, and the default value is 4 (unit: sec)\r\n");
        return;
    }

    if ((pingnum < 1) || (pingnum > 10)) {
        LOGDEBUG("The maximum number of sending Ping request range is 1-10, and the default value is 4\r\n");
        return;
    }

    m_parser.set_timeout((1000 * timeout) + 2000);

    if (m_parser.send("AT+QPING=%d,%s", %d, %d", 1, host, timeout, pingnum) && m_parser.recv("OK")) {
        for (i = 0; i < (pingnum); i++) {
            m_parser.recv("+QPING: %s\r\n", resp_str);
            sprintf((char *)buf, "%s: %s\r\n", host, resp_str);
            LOGDEBUG(buf);
        }
    }

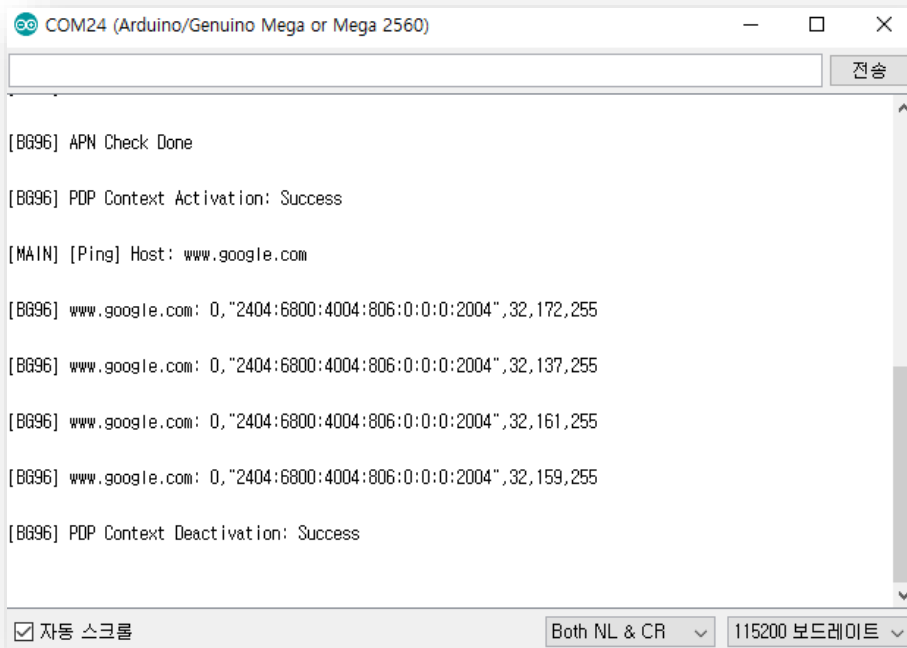
    m_parser.set_timeout(BG96_DEFAULT_TIMEOUT);
    m_parser.flush();
}
```

예외 처리

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

>> QUIZ1

- 강의 자료WExampleW**QUIZ1**
 - getNetworkStatus_BG96() 함수를 24p의 **HINT**를 참고하여
기존과 같이 동작하도록 수정
 - 정상 동작하면, PING TEST하는 것을 확인 가능



The screenshot shows the Arduino IDE serial monitor window for a COM24 port (Arduino/Genuino Mega or Mega 2560). The window displays the following output:

```
[BG96] APN Check Done  
[BG96] PDP Context Activation: Success  
[MAIN] [Ping] Host: www.google.com  
[BG96] www.google.com: 0,"2404:6800:4004:806:0:0:0:2004",32,172,255  
[BG96] www.google.com: 0,"2404:6800:4004:806:0:0:0:2004",32,137,255  
[BG96] www.google.com: 0,"2404:6800:4004:806:0:0:0:2004",32,161,255  
[BG96] www.google.com: 0,"2404:6800:4004:806:0:0:0:2004",32,159,255  
[BG96] PDP Context Deactivation: Success
```

At the bottom of the window, there are checkboxes for "자동 스크롤" (Auto Scroll) which is checked, and dropdown menus for "Both NL & CR" and "115200 보드레이트" (Baud Rate).

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

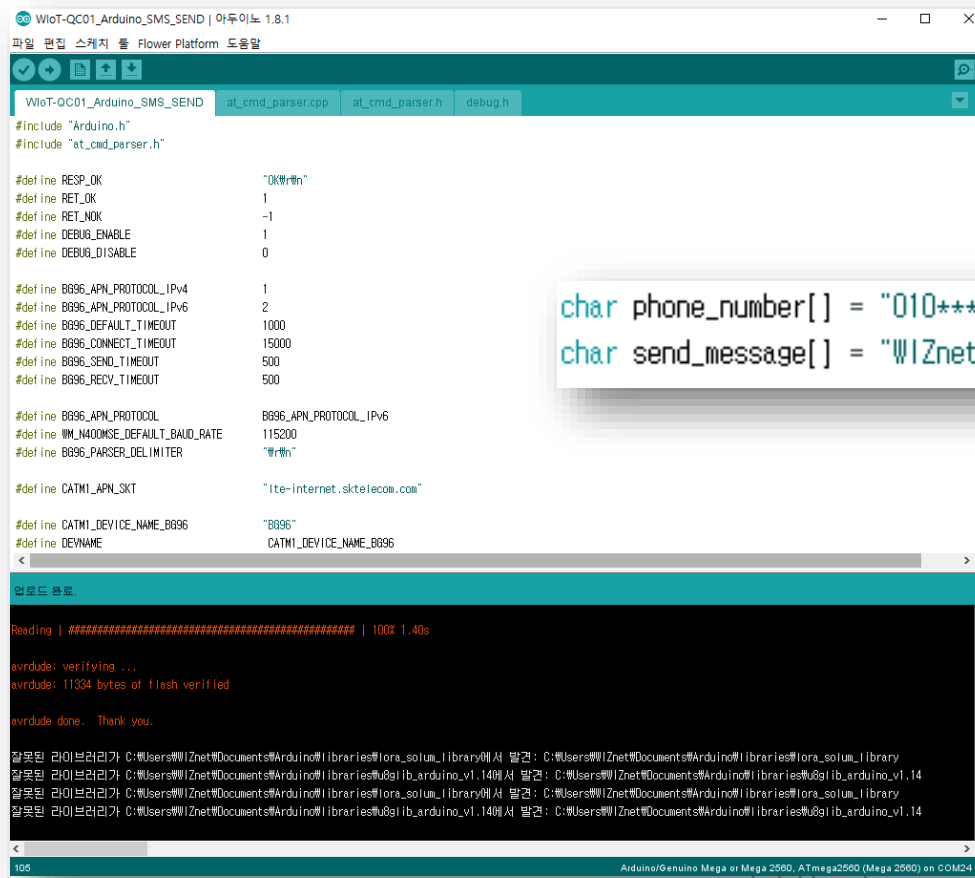
- 1) LTE Cat.M1 Attach & Ping TEST
- 2) **SMS Send – CDS Triggering**
- 3) SMS Recv – Device Reset



Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

» Send SMS – CDS Triggering

- 강의 자료 Example Wd_WIoT-QC01_Arduino_SMS_SEND
- 번호 및 메시지 수정 필요



```
#include "Arduino.h"
#include "at_cmd_parser.h"

#define RESP_OK "OK\r\n"
#define RET_OK 1
#define RET_NOK -1
#define DEBUG_ENABLE 1
#define DEBUG_DISABLE 0

#define B696_APN_PROTOCOL_IPV4 1
#define B696_APN_PROTOCOL_IPV6 2
#define B696_DEFAULT_TIMEOUT 1000
#define B696_CONNECT_TIMEOUT 15000
#define B696_SEND_TIMEOUT 500
#define B696_RECV_TIMEOUT 500

#define B696_APN_PROTOCOL B696_APN_PROTOCOL_IPV6
#define WMLN400KSE_DEFAULT_BAUD_RATE 115200
#define B696_PARSER_DELIMITER "\r\n"

#define CATM1_APN_SKT "lte-internet.sktelecom.com"

#define CATM1_DEVICE_NAME_B696 "B696"
#define DEVNAME CATM1_DEVICE_NAME_B696
```

업로드 완료

Reading | ##### | 100% 1.40s

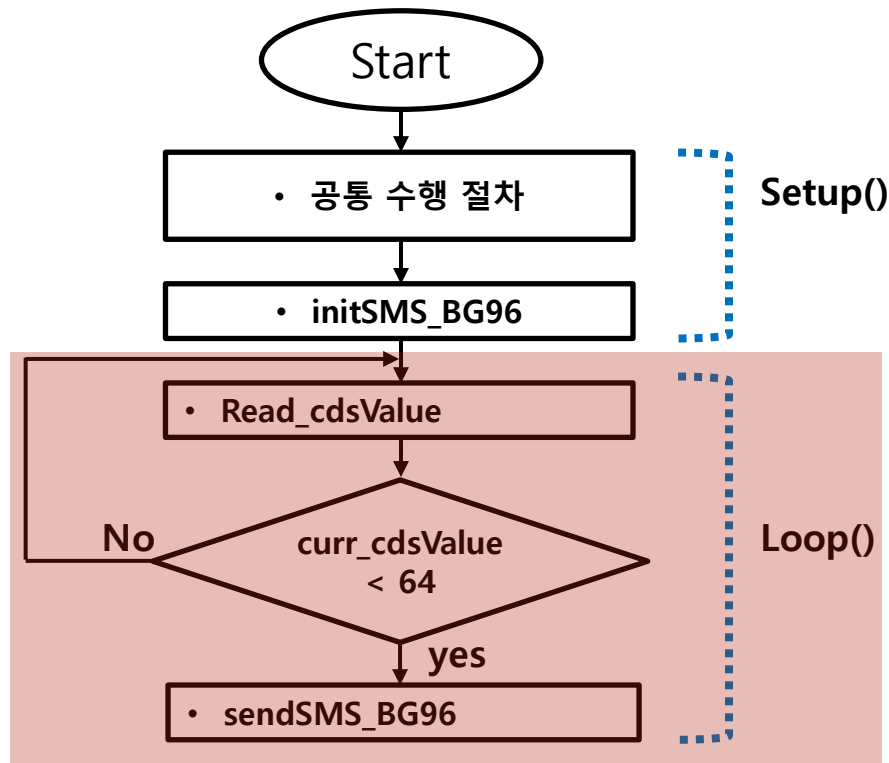
avrdude: verifying ...
avrdude: 11334 bytes of flash verified
avrdude done. Thank you.

잘못된 라이브러리가 C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library에서 발견: C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library
잘못된 라이브러리가 C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library에서 발견: C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library
잘못된 라이브러리가 C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library에서 발견: C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library
잘못된 라이브러리가 C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library에서 발견: C:\Users\W\Znet\Documents\Arduino\libraries\lora_somul\library

```
char phone_number[] = "010*****";  
char send_message[] = "WIZnet Cat.M1 IoT shield is powered on";
```

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

>> Send SMS – CDS Triggering

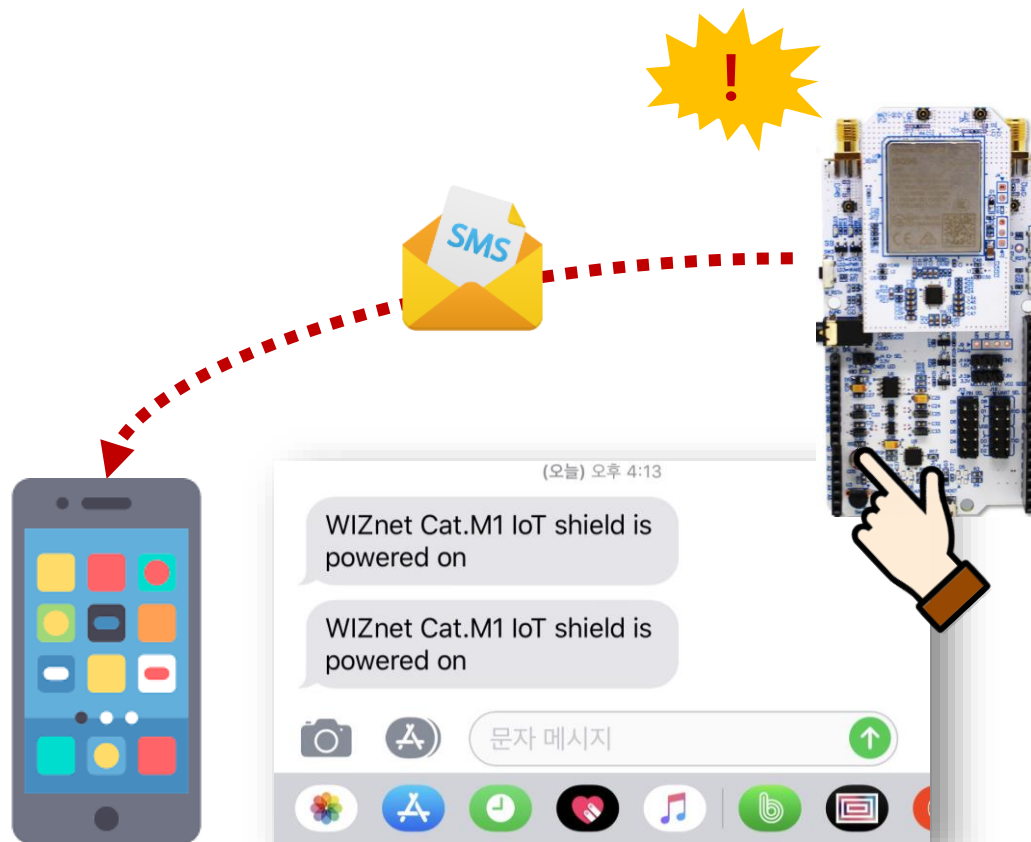


```
void loop() {  
  // put your main code here, to run repeatedly:  
  //map : 특정 범위에 속하는 값을 다른 범위의 값으로 변환해주는 명령어  
  curr_cdsValue = map(analogRead(IOTSHIELD_SENSOR_CDS), 0, 1023, 0, 256);  
  //curr_cdsValue = (analogRead(IOTSHIELD_SENSOR_CDS) * 5v); // 다른 계산 방법  
  
  if ( curr_cdsValue < 64) { // 손가락으로 가리면 조도 저항값이 높아지기 때문에 아날로그 값이 낮아진다  
    if (sendSMS_BG96(phone_number, send_message, strlen(send_message)) == RET_OK) {  
      MYPRINTF("SEND [OK]");  
      delay(5000);  
    }  
  }  
}
```


Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

>> Send SMS – CDS Triggering

- 강의 자료 Example Wd_WIoT-QC01_Arduino_SMS_SEND



```
COM24 (Arduino/Genuino Mega or Mega 2560)

[MAIN] =====
[MAIN] >> Target Board: WIoT-QC01 (Quectel BG96)
[MAIN] >> Sample Code: SMS Test
[MAIN] =====

[BG96] Echo Off: Success

[BG96] USIM Status: READY

[BG96] Network Status: Attach

[BG96] Checking APN...

[BG96] APN Check Done

[BG96] SMS message format: Text mode

[BG96] SMS character set: GSM

[BG96] >> SMS send success: index 2

[MAIN] SEND [OK]
```

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

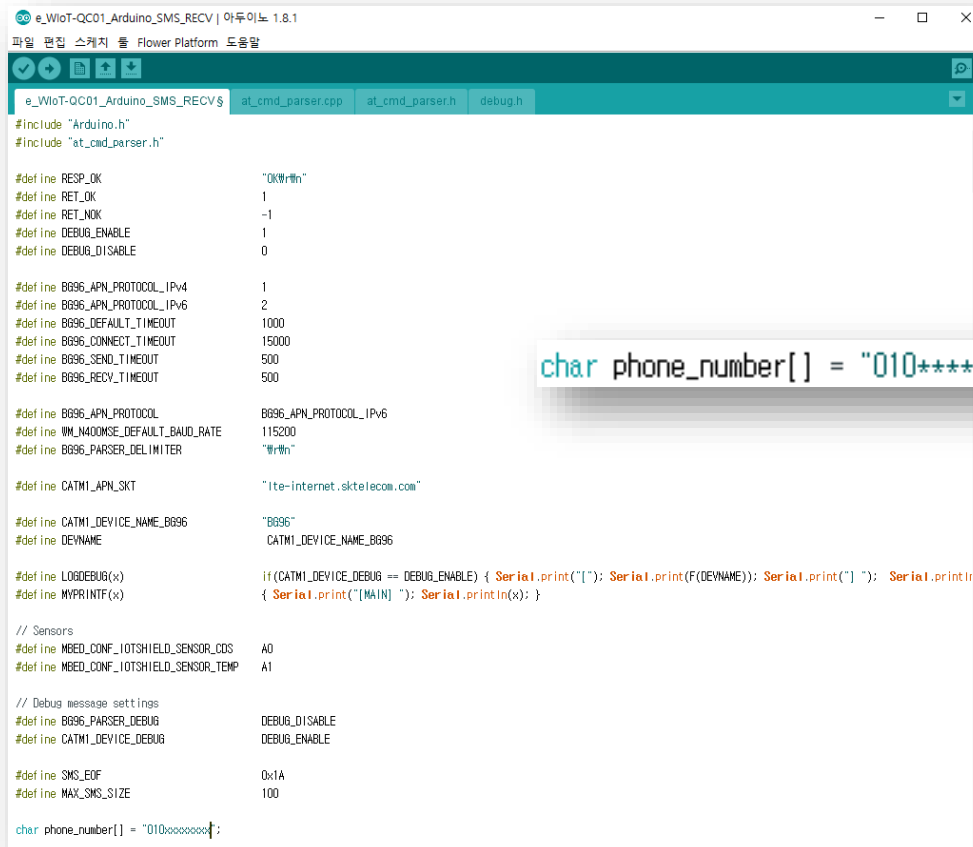
- 1) LTE Cat.M1 Attach & Ping TEST
- 2) SMS Send – CDS Triggering
- 3) **SMS Recv – Device Reset**



Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

» SMS Recv – Device Reset

- 강의 자료 Example e_WIoT-QC01_Arduino_SMS_RECV
- 휴대폰 번호 수정 필요



```
e_WIoT-QC01_Arduino_SMS_RECV | 아두이노 1.8.1
파일 편집 스케치 플 Flower Platform 도움말

e_WIoT-QC01_Arduino_SMS_RECV $ at_cmd_parser.cpp at_cmd_parser.h debug.h

#include "Arduino.h"
#include "at_cmd_parser.h"

#define RESP_OK "OK\r\n"
#define RET_OK 1
#define RET_NOK -1
#define DEBUG_ENABLE 1
#define DEBUG_DISABLE 0

#define B096_APN_PROTOCOL_IPV4 1
#define B096_APN_PROTOCOL_IPV6 2
#define B096_DEFAULT_TIMEOUT 1000
#define B096_CONNECT_TIMEOUT 15000
#define B096_SEND_TIMEOUT 500
#define B096_RECV_TIMEOUT 500

#define B096_APN_PROTOCOL B096_APN_PROTOCOL_IPV6
#define WM_NADOMSE_DEFAULT_BAUD_RATE 115200
#define B096_PARSER_DELIMITER "\r\n"

#define CATM1_APN_SKT "lte-internet.sktelecom.com"

#define CATM1_DEVICE_NAME_B096 "B096"
#define DEVNAME CATM1_DEVICE_NAME_B096

#define LOGDEBUG(x) if(CATM1_DEVICE_DEBUG == DEBUG_ENABLE) { Serial.print(""); Serial.print(F(DEVNAME)); Serial.print(" "); Serial.println(x); }
#define MYPRINTF(x) { Serial.print("[MAIN] "); Serial.println(x); }

// Sensors
#define MBED_CONF_IOTSHIELD_SENSOR_CDS A0
#define MBED_CONF_IOTSHIELD_SENSOR_TEMP A1

// Debug message settings
#define B096_PARSER_DEBUG DEBUG_DISABLE
#define CATM1_DEVICE_DEBUG DEBUG_ENABLE

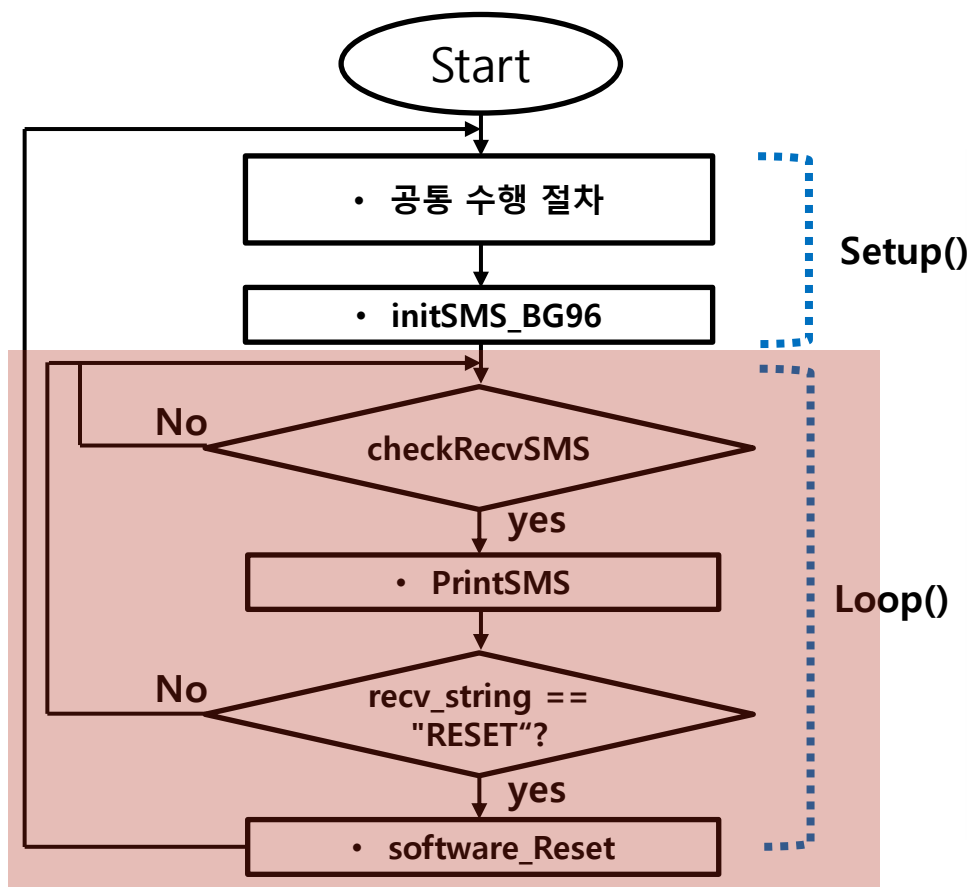
#define SMS_EOF 0x1A
#define MAX_SMS_SIZE 100

char phone_number[] = "010*****";
```

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

>> SMS Recv – Device Reset

- 강의 자료 Example We_WIoT-QC01_Arduino_SMS_RECV



```
void loop() {
    // put your main code here, to run repeatedly:

    // SMS receive check
    msg_idx = checkRecvSMS_BG96();    +CMTI: "ME",1 ← 문자 알림 체크

    if (msg_idx > RET_NOK) { // SMS received
        // Receive a message
        if (recvSMS_BG96(msg_idx, recv_message, dest_addr, date_time) == RET_OK) {
            sprintf((char *)sms_buf, "[SMS Recv] from %s, %s, %s, %s", dest_addr, date_time, recv_message);
            MYPRINTF(sms_buf);

            recv_string = recv_message; // 문자열 비교를 위해 c_str --> String 으로 캐스팅
            if (recv_string == "RESET") {
                delay(50);
                MYPRINTF("MCU RESET");
                delay(50);
                software_Reset();
            }
            memset(recv_message, 0, sizeof(recv_message)); // recv_message 초기화, 만약 하지 않으면 메시지가 '뒤'
                                                         // Msg1: 'Hello', Msg2: 'Hi' --> Msg2: 'Hillo'로 표시됨
        }
    }
}
```

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (1)

>> SMS Recv – Device Reset

- 강의 자료 Example We_WIoT-QC01_Arduino_SMS_RECV

The diagram illustrates the process of sending an SMS to an Arduino WIoT Cat.M1 Shield to trigger a system reset. On the left, a smartphone is shown with a red dashed arrow pointing from its screen to a yellow envelope icon labeled 'SMS'. This envelope icon has another red dashed arrow pointing to the Arduino WIoT Cat.M1 Shield. Below the smartphone, a screenshot of a text message conversation is displayed. The messages are: 'SMS TEST 1' and 'RESET'. The status bar at the top of the screenshot shows 'LG U+ LTE' and the time '오후 9:02'. The contact name is '012-2771-2274 >'. The bottom of the screenshot shows the iOS home screen with various app icons.

On the right, a screenshot of the Arduino IDE serial monitor window is shown. The window title is 'COM24 (Arduino/Genuino Mega or Mega 2560)'. The serial output shows the following messages:

```
[BG96] << SMS receive success: index 0
[MAIN] [SMS Recv] from 01090373914, 19/05/29,20:35:12+36, "F6"
[BG96] << SMS received: index 1
[BG96] << SMS receive success: index 1
[MAIN] [SMS Recv] from 01090373914, 19/05/29,21:01:44+36, "SMS TEST 1"
[BG96] << SMS received: index 2
[BG96] << SMS receive success: index 2
[MAIN] [SMS Recv] from 01090373914, 19/05/29,21:01:55+36, "RESET"
[MAIN] MCU RESET
[MAIN] Waiting for Cat.M1 Module Ready...
[MAIN] BG96 already available
[MAIN] System Init Complete
```

The text 'System Reset()' is written in red next to the 'RESET' message in the serial output.

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (2)

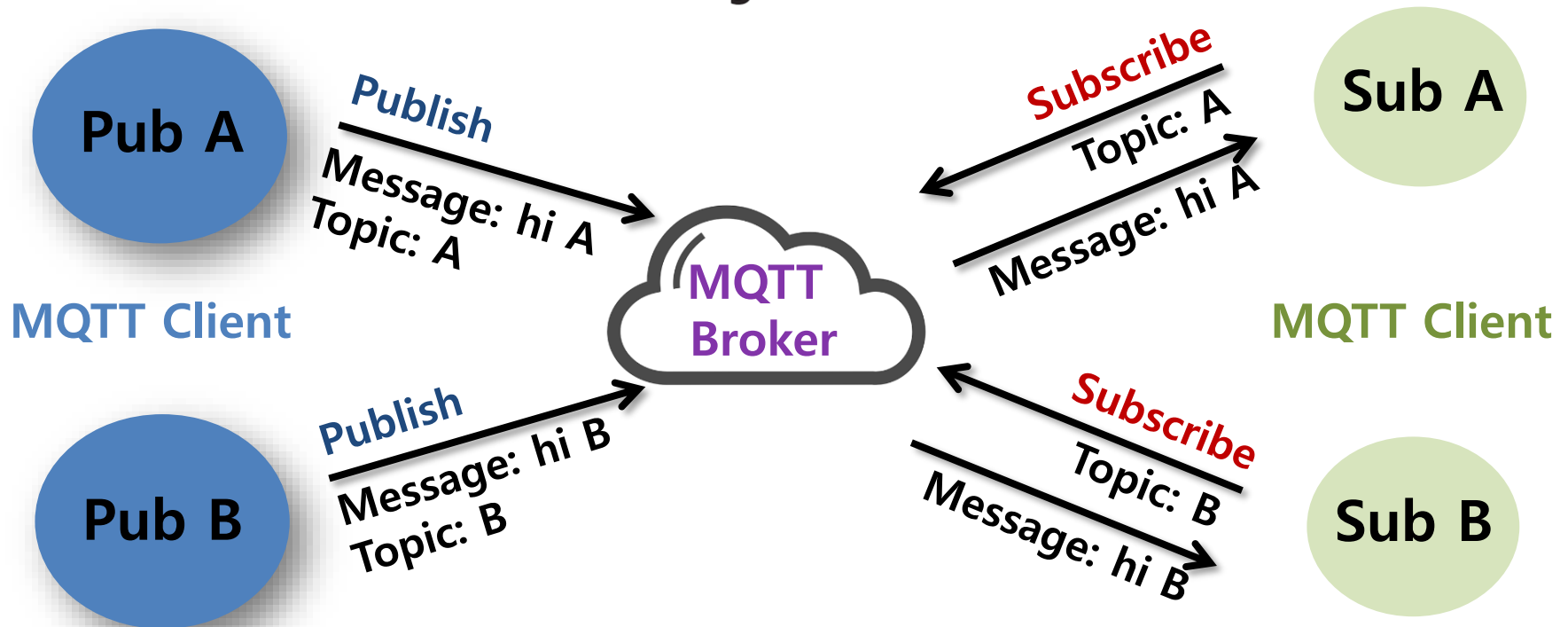
- 1) MQTT 이론 및 환경 설정
- 2) MQTT Local 테스트
- 3) MQTT Publish – Temperature
- 4) MQTT Subscribe – Device Reset



MQTT 이론 및 환경 설정

>> MQTT(Message Queue Telemetry Transport) 란

- 경량화 프로토콜 이므로 M2M, IoT분야에 적합
- Topic 기반의 Publish/subscribe
- QoS 0~2 지원
- 대표적인 사용 예: Facebook messenger



MQTT 이론 및 환경 설정

>> MQTT 브로커 – Mosquitto 설치

- <http://mosquitto.org/download/>

Source

- [mosquitto-1.6.2.tar.gz](#) (319kB) (GPG signature)
- [Git source code repository](#) (github.com)

Older downloads are available at <https://mosquitto.org/files/>

Binary Installation

The binary packages listed below are supported by the Mosquitto project. In many cases Mosquitto is also available directly from official Linux/BSD distributions.

Windows

- [mosquitto-1.6.2-install-windows-x64.exe](#) (~360 kB) (64-bit build, Windows Vista and up, built with Visual Studio Community 2017) ✓
- [mosquitto-1.6.2-install-windows-x32.exe](#) (~360 kB) (32-bit build, Windows Vista and up, built with Visual Studio Community 2017) ✓

See also readme-windows.txt after installing.

Mac

Mosquitto can be installed from the homebrew project. See [brew.sh](#) and then use `brew install mosquitto`








MQTT 이론 및 환경 설정

>> MQTT 클라이언트 – MQTT.fx 설치

- <https://mqttfx.jensd.de/index.php/download>

Latest Release

MQTT.fx Version 1.7.1 (more information) ✓

| Name | Last modified | Size | Description |
|---|-------------------------------|----------------------|-----------------------------|
|  Parent Directory | | - | |
|  mqttfx-1.7.1-1.i386.rpm | 2018-10-02 09:44 | 99M | |
|  mqttfx-1.7.1-1.x86_64.rpm | 2018-10-02 09:35 | 96M | |
|  mqttfx-1.7.1-32bit.deb | 2018-10-02 09:43 | 75M | |
|  mqttfx-1.7.1-64bit.deb | 2018-10-02 09:34 | 72M | |
|  mqttfx-1.7.1-macos.dmg | 2018-09-28 14:27 | 55M | |
|  mqttfx-1.7.1-windows...> | 2018-09-28 14:27 | 51M | 64bit |
|  mqttfx-1.7.1-windows...> | 2018-09-28 14:27 | 47M | 32bit |

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (2)

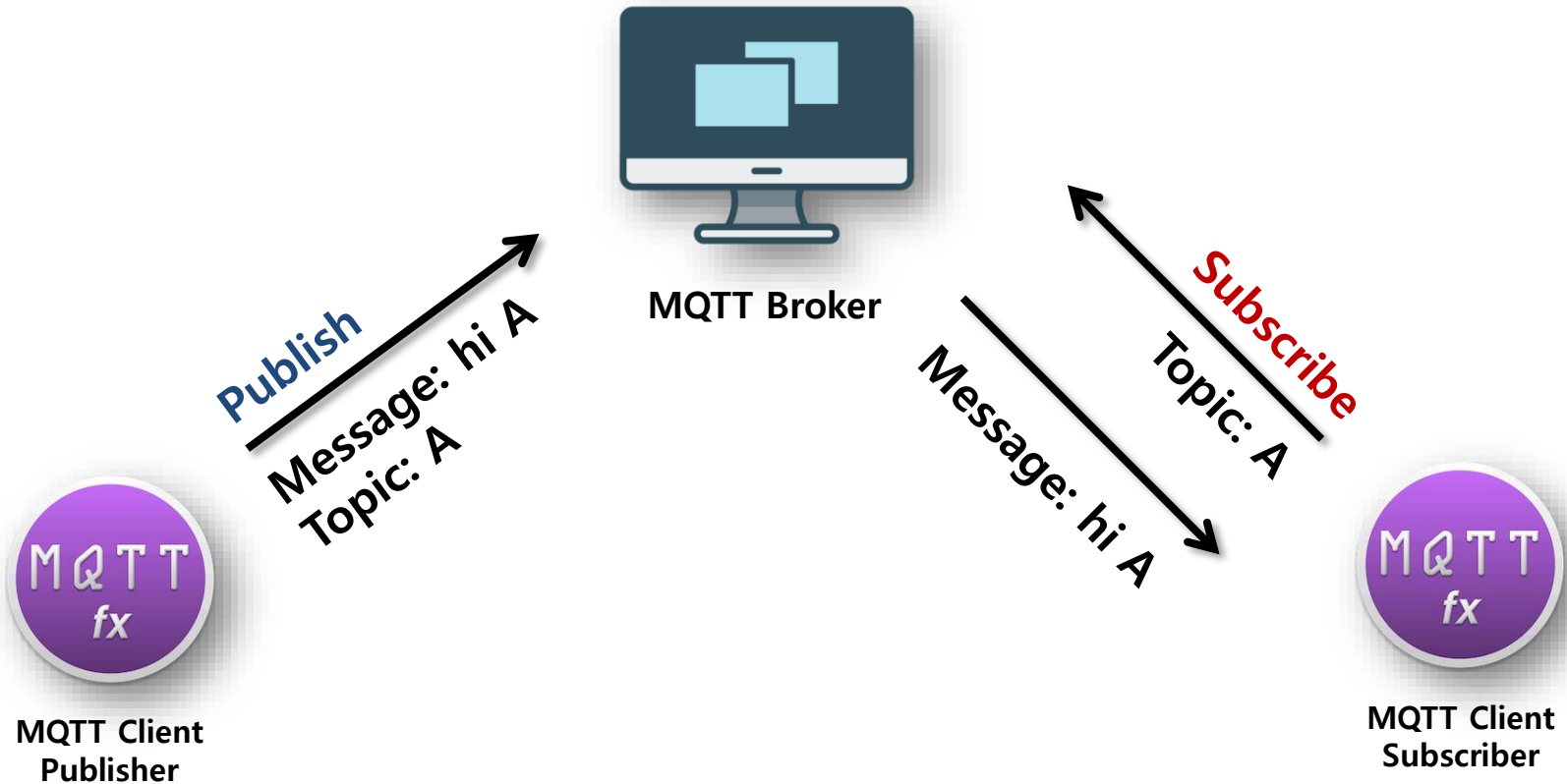
- 1) MQTT 이론 및 환경 설정
- 2) **MQTT Local 테스트**
- 3) MQTT Publish – Temperature
- 4) MQTT Subscribe – Device Reset



MQTT Local 테스트

>> Local 실습 환경

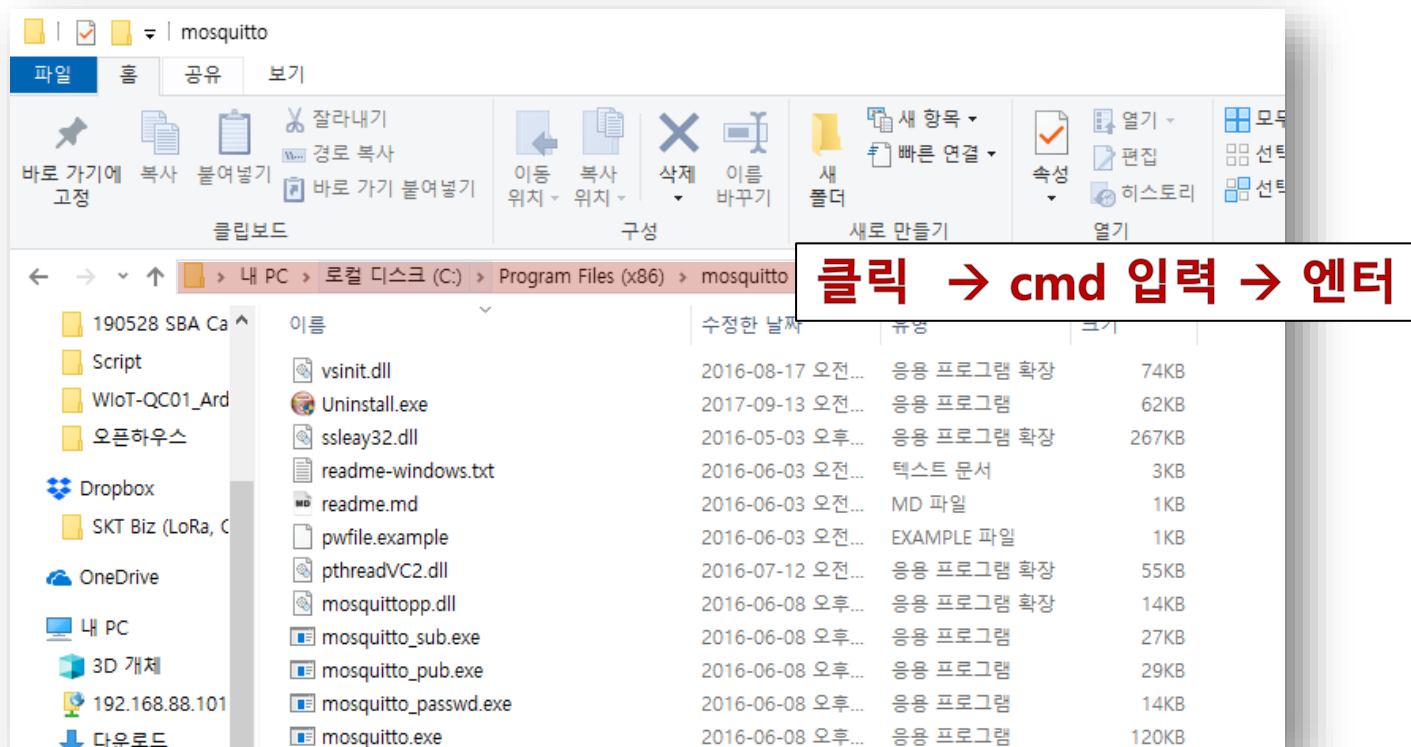
- MQTT 동작 이해



MQTT Local 테스트

>> MQTT 브로커 실행

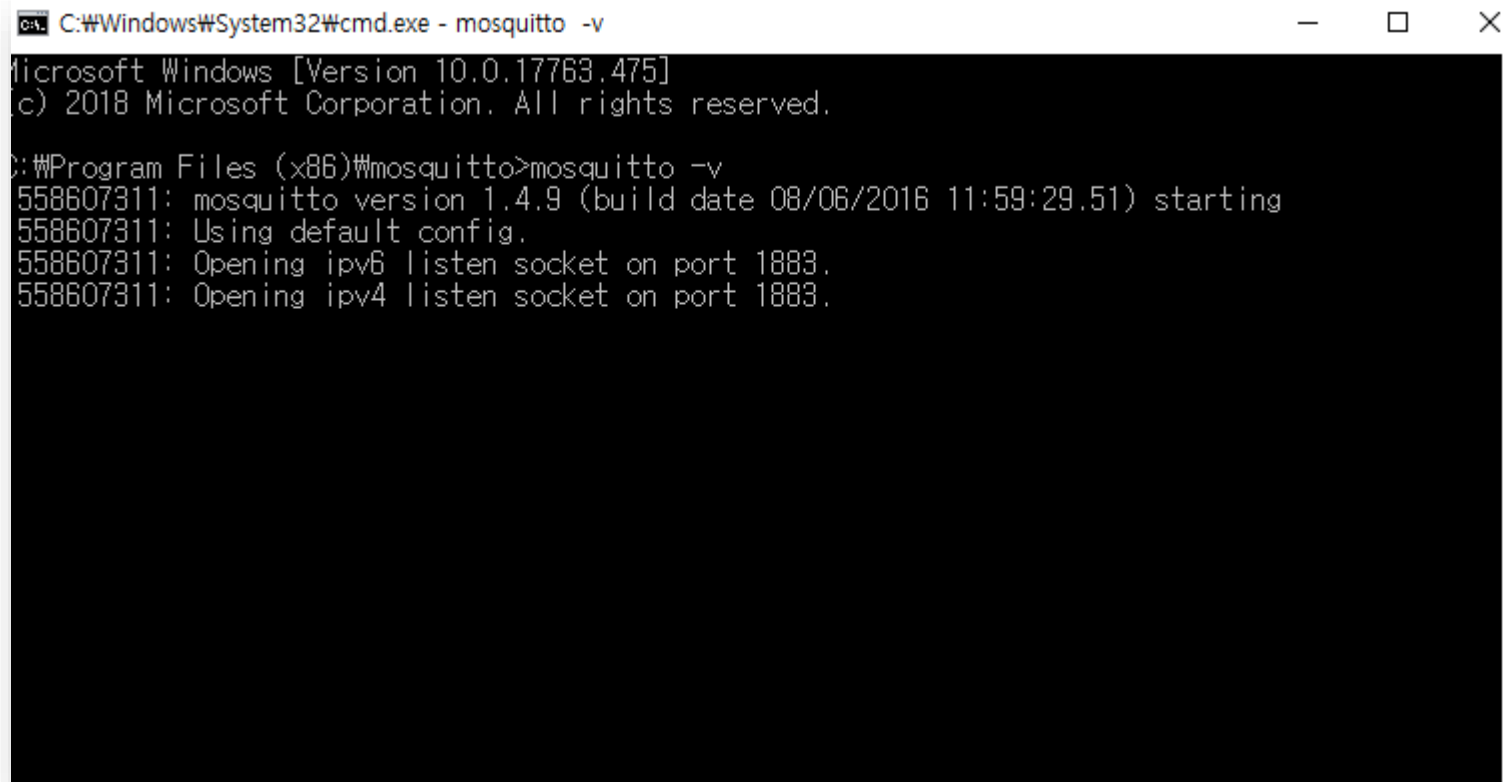
- 설치된 곳으로 이동
 - 일반적으로 program files → mosquitto



MQTT Local 테스트

>> MQTT 브로커 실행

- **mosquitto -v**



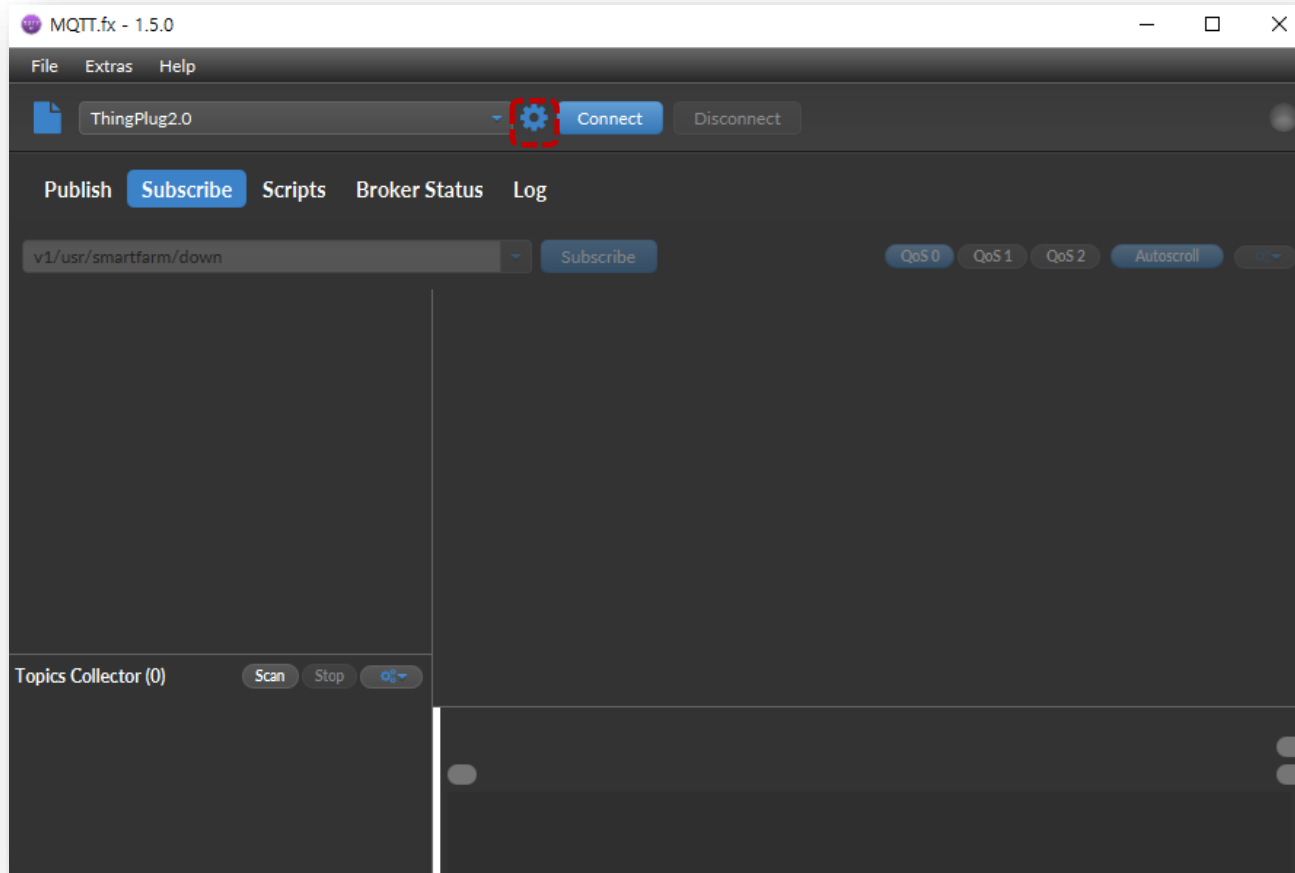
```
C:\Windows\System32\cmd.exe - mosquitto -v
Microsoft Windows [Version 10.0.17763.475]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\mosquitto>mosquitto -v
558607311: mosquitto version 1.4.9 (build date 08/06/2016 11:59:29.51) starting
558607311: Using default config.
558607311: Opening ipv6 listen socket on port 1883.
558607311: Opening ipv4 listen socket on port 1883.
```

MQTT Local 테스트

>> MQTT 클라이언트(1) 실행

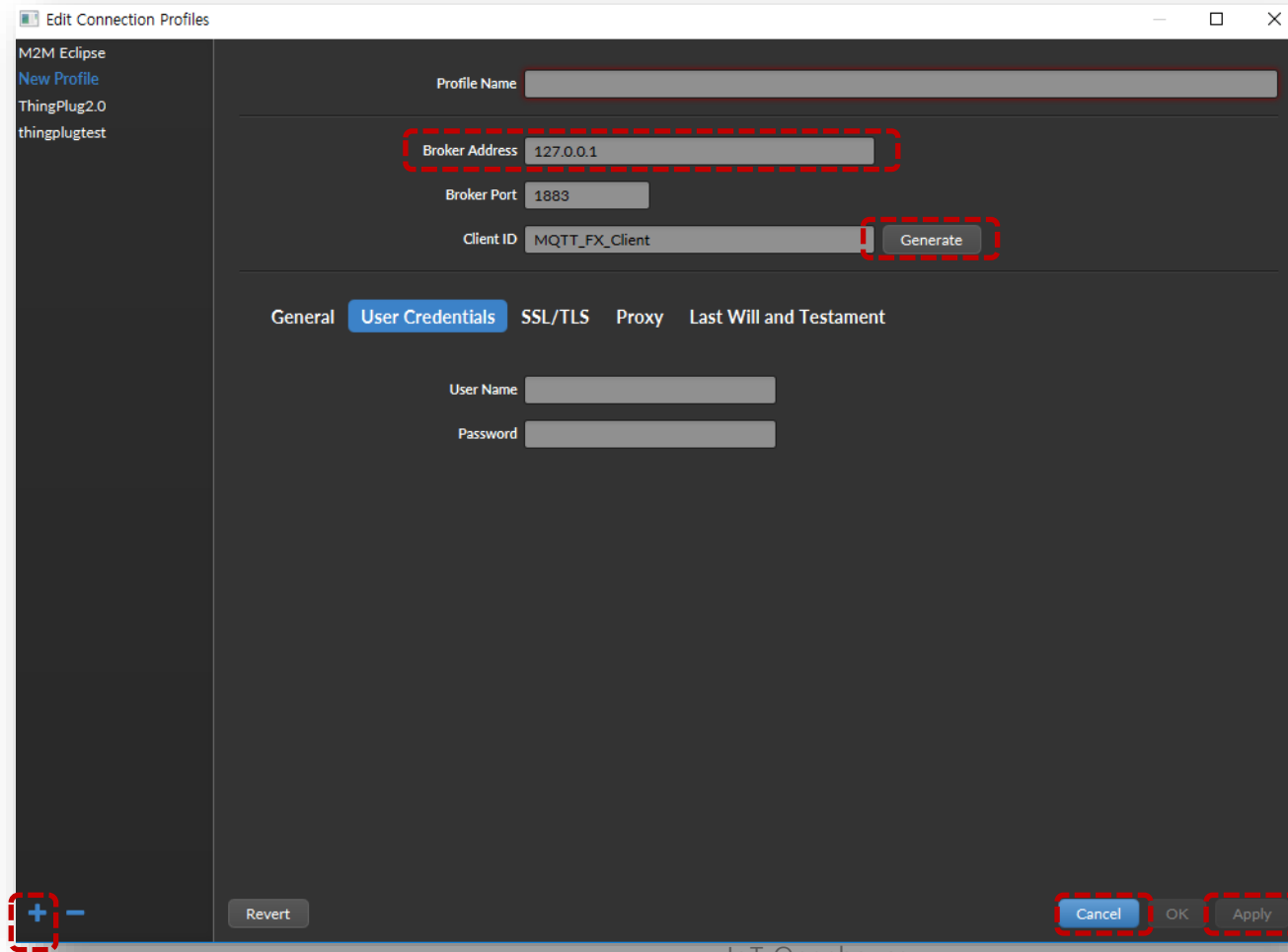
- 윈도우 --> mqttfx 검색



MQTT Local 테스트

>> MQTT 클라이언트(1) 실행

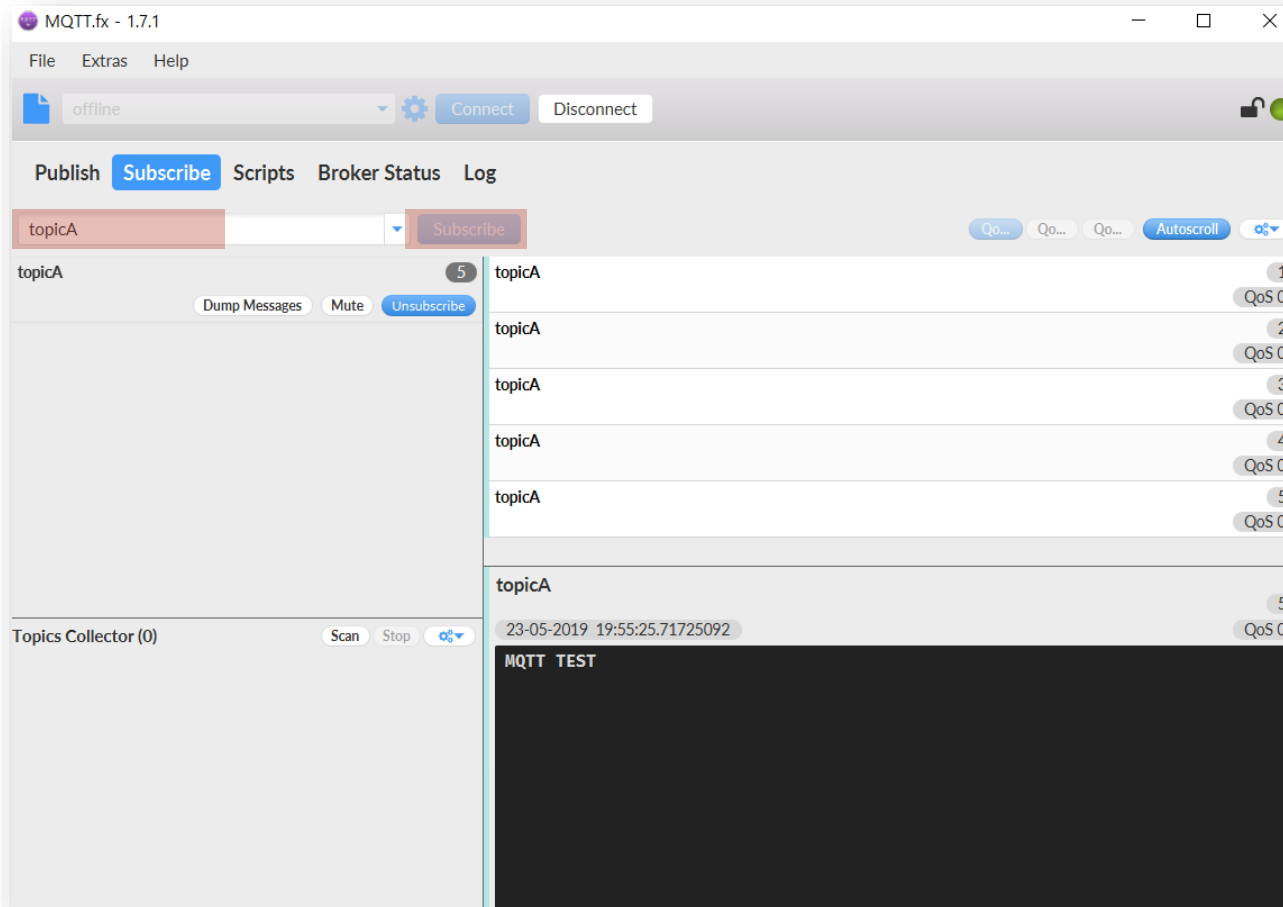
- Broker Address: 127.0.0.1



MQTT Local 테스트

>> MQTT 클라이언트(1) 실행

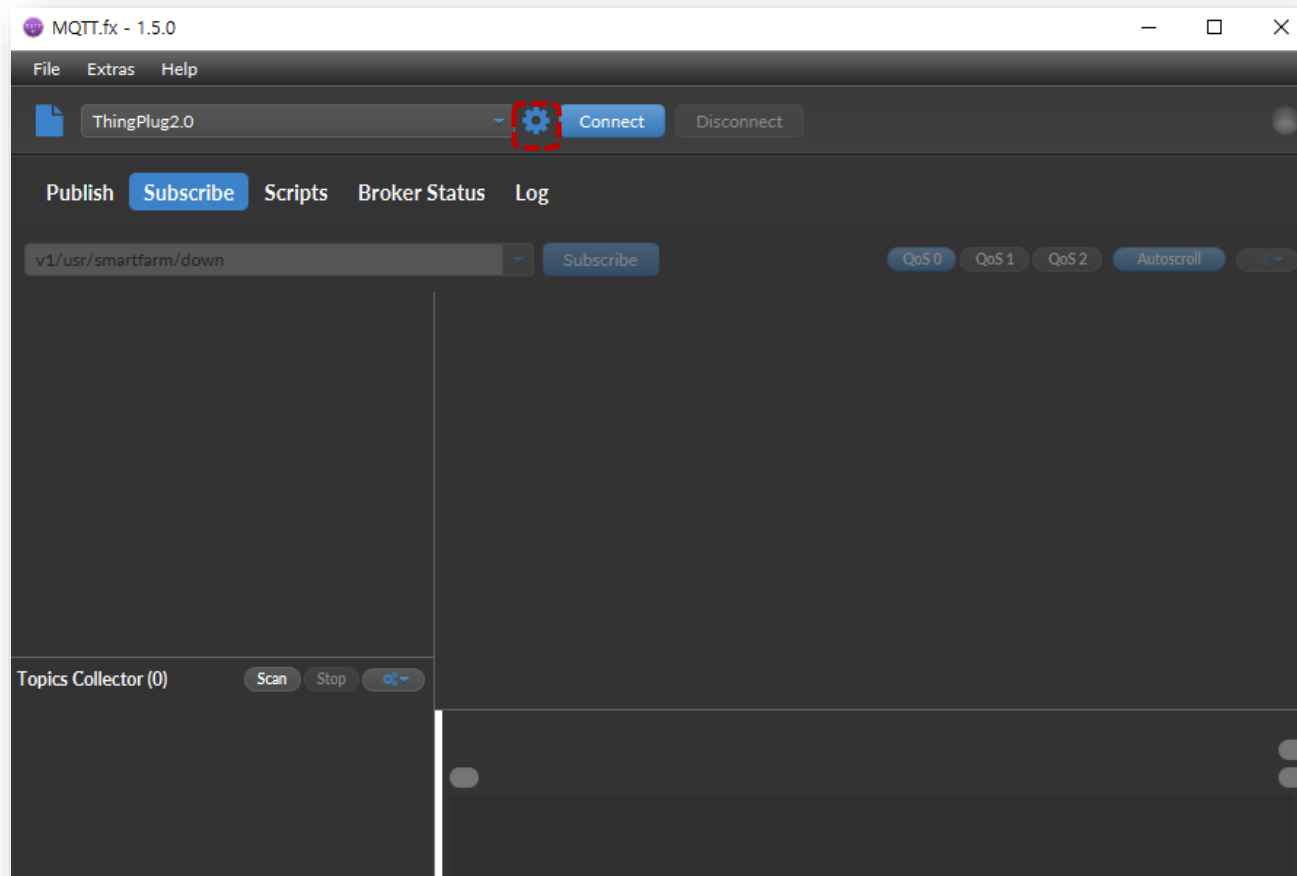
- Subscribe Topic: **[Input Topic]**



MQTT Local 테스트

>> MQTT 클라이언트(2) 실행

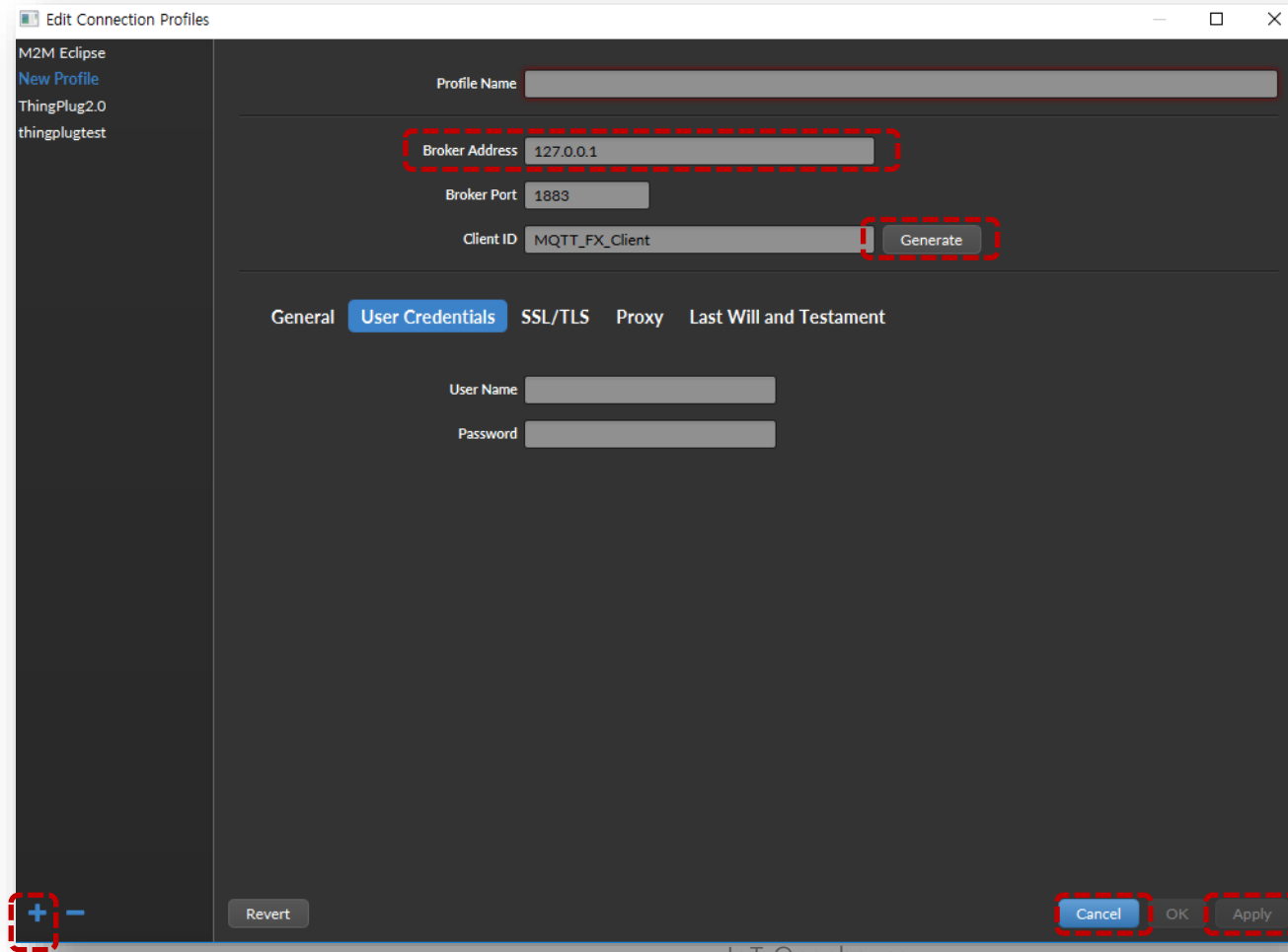
- 윈도우 --> mqttfx 검색



MQTT Local 테스트

>> MQTT 클라이언트(2) 실행

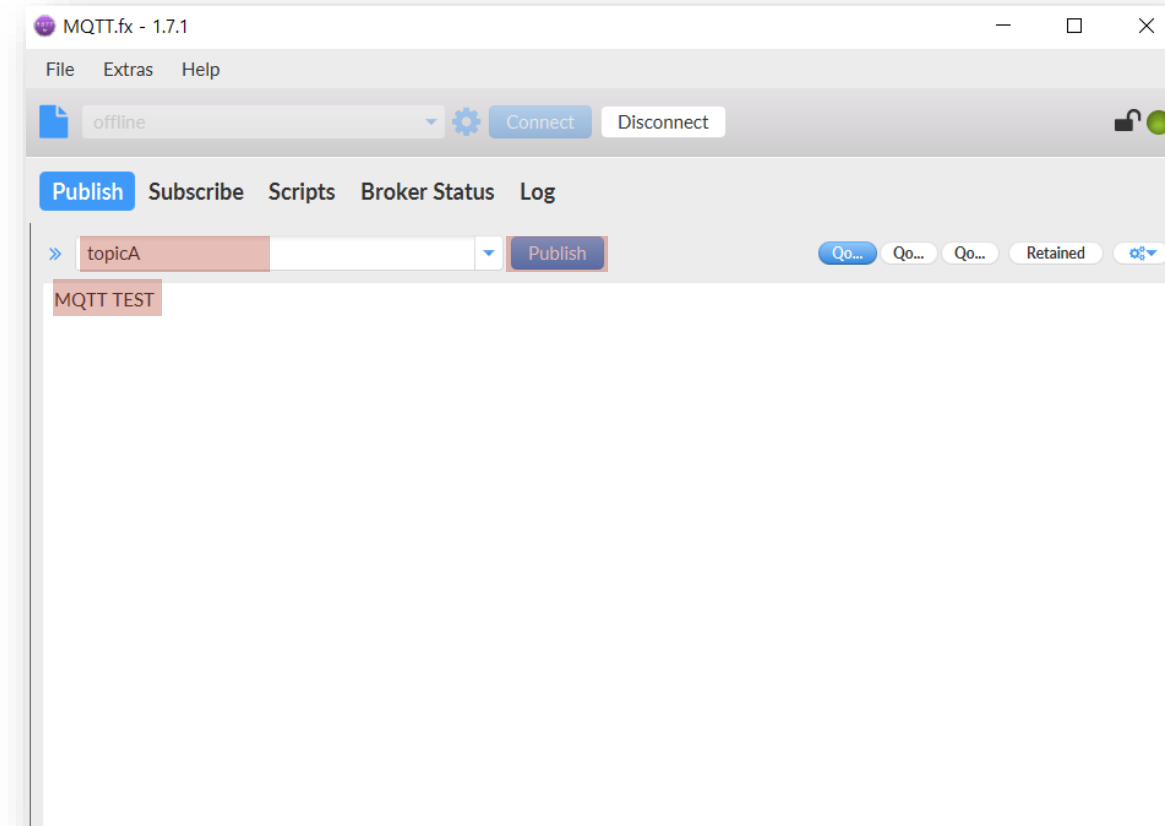
- Broker Address: 127.0.0.1



MQTT Local 테스트

>> MQTT 클라이언트(2) 실행

- Publish Topic: **[Input Topic]**
- Message: **[Input Message]**



MQTT Local 테스트

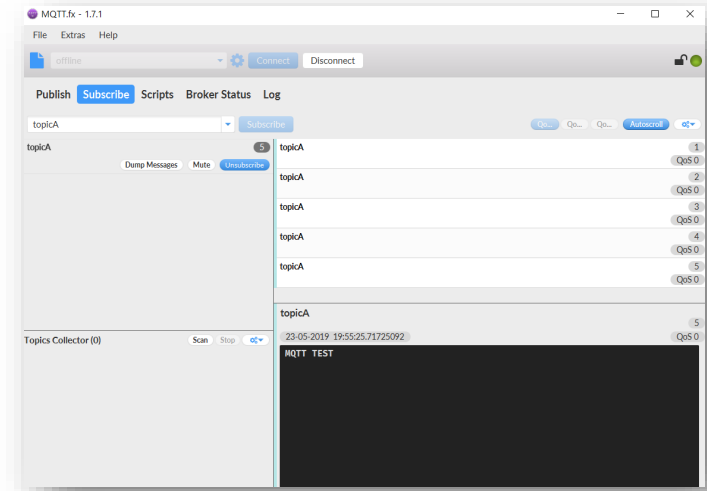
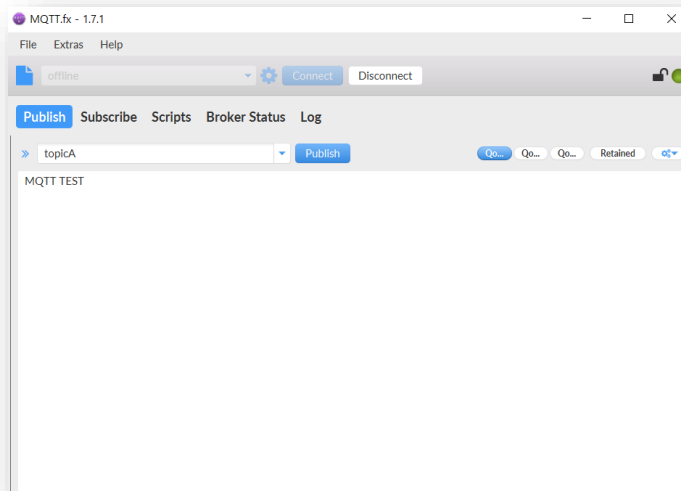
>> MQTT 클라이언트 실행

- 동작 확인

- 127.0.0.1

Publish

Subscribe

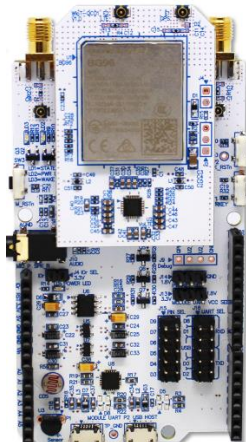


MQTT Local 테스트

>> MQTT 클라이언트 실행

- MQTT Client
→ Cat.M1 ??

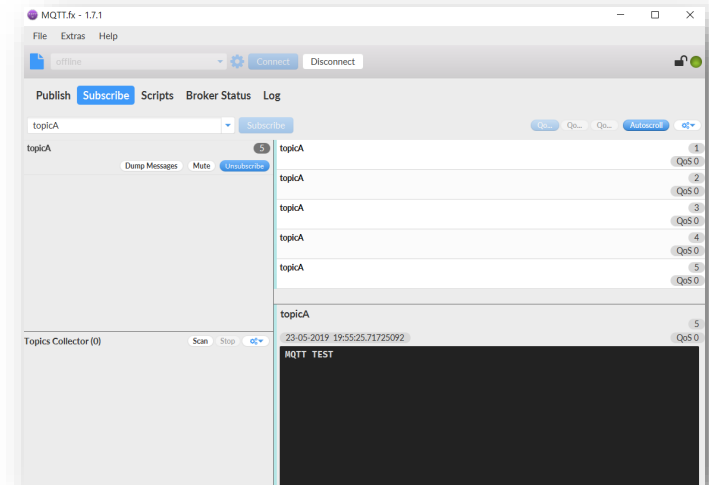
Publish



Local: 127.0.0.1

```
C:\Windows\System32\cmd.exe - mosquitto -v
558608826: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558608826: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558608865: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558608865: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
558608886: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558608886: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558608925: Received PUBLISH from 2ab74e2eb73a43efb1ac501d64ed767c (d0, q0, r0, m0, 'topicA', ... (9 bytes))
558608925: Sending PUBLISH to aac4e9c1b77d49808c0e682dae0ad136 (d0, q0, r0, m0, 'topicA', ... (9 bytes))
558608925: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558608925: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
558608946: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558608946: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558608985: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558608985: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
558609006: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558609006: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558609045: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558609045: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
```

Subscribe



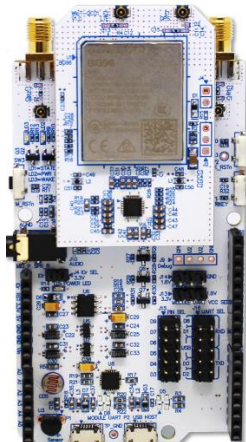
MQTT Local 테스트

>> MQTT 클라이언트 실행

- Why **not** ?

--> 브로커 : **사설 IP**

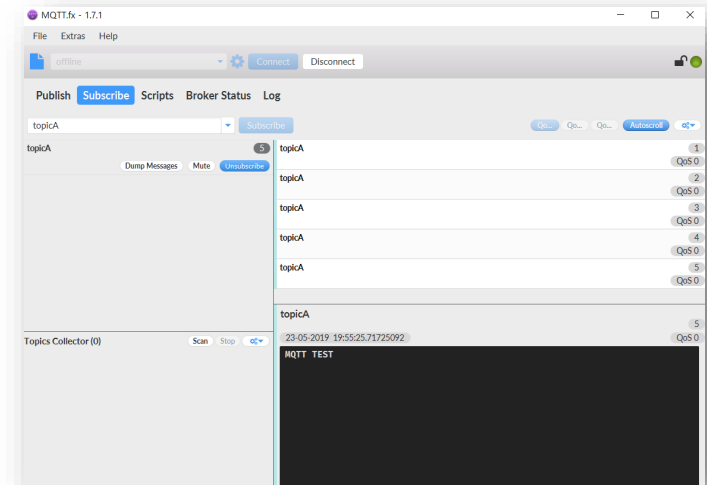
Publish



Local: 127.0.0.1

```
C:\Windows\System32\cmd.exe - mosquitto -v
558608826: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558608826: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558608865: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558608865: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
558608886: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558608886: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558608925: Received PUBLISH from 2ab74e2eb73a43efb1ac501d64ed767c (d0, q0, r0, m0, 'topicA', ... (9 bytes))
558608925: Sending PUBLISH to aac4e9c1b77d49808c0e682dae0ad136 (d0, q0, r0, m0, 'topicA', ... (9 bytes))
558608925: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558608925: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
558608946: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558608946: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558608985: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558608985: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
558609006: Received PINGREQ from 2ab74e2eb73a43efb1ac501d64ed767c
558609006: Sending PINGRESP to 2ab74e2eb73a43efb1ac501d64ed767c
558609045: Received PINGREQ from aac4e9c1b77d49808c0e682dae0ad136
558609045: Sending PINGRESP to aac4e9c1b77d49808c0e682dae0ad136
```

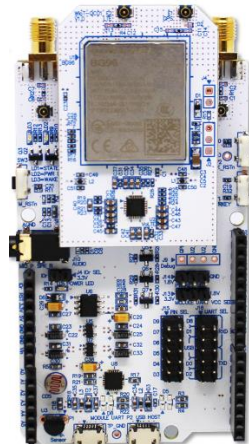
Subscribe



MQTT Local 테스트

>> MQTT 클라이언트 실행

- Public 브로커 사용

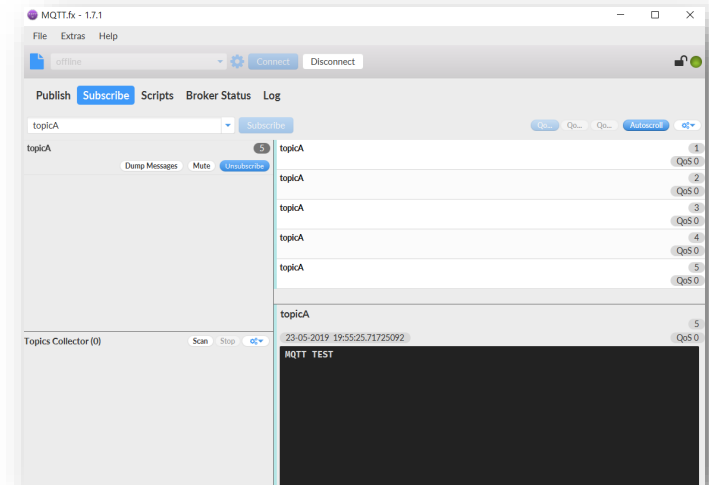


IoT 디바이스

Publish



Subscribe



서버

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (2)

- 1) MQTT 이론 및 환경 설정
- 2) MQTT Local 테스트
- 3) **MQTT Publish – Temperature**
- 4) MQTT Subscribe – Device Reset



MQTT Publish – Temperature

» 주기적으로 온도를 측정하여 Publish

- 강의 자료 Example Wf_WIoT-QC01_Arduino_MQTT_SEND
- 토픽 수정 필요

The screenshot displays the Arduino IDE interface. The top window shows the sketch 'Wf_WIoT-QC01_Arduino_MQTT_SEND' with the following code:

```
int8_t closeMqttBroker_B696(void)
{
    int8_t ret = RET_NOK;
    int id = 0;
    int result = 0;
    unsigned long lastClosedTime = 0;    // last time you connected to the server, in milliseconds

    bool done = false;
    //Timer t;

    //t.start();
    lastClosedTime = millis();

    if (m_parser.send("AT+QMQTIDISC=%d", id) && m_parser.recv("OK")) {
        do {
            done = (m_parser.recv("+QMQTIDISC: %d,%d", &id, &result));
        } while (!done && (millis() - lastClosedTime) < B696_CONNECT_TIMEOUT + 2);

        if (done) {
            ret = RET_OK;
        }
    }
    m_parser.flush();
}
```

The bottom window shows the serial monitor output for COM24 (Arduino/Genuino Mega or Mega 2560):

```
[B696] APN Check Done

[MAIN] [MQTT] Try to connect to broker "broker.hivemq.com:1883"

[MAIN] [MQTT] Socket open success

[MAIN] [MQTT] Connected. Start to send a message.

[ Temperature Data ]
Sensor value :58
Volt : 0.28mV
Celsius temp : 27.58C
Fahrenheit temp : 81.65F
[MAIN] [MQTT] Published Topic: "topic/openhouse-B", Message: "27.584"

[ Temperature Data ]
Sensor value :53
Volt : 0.25mV
```

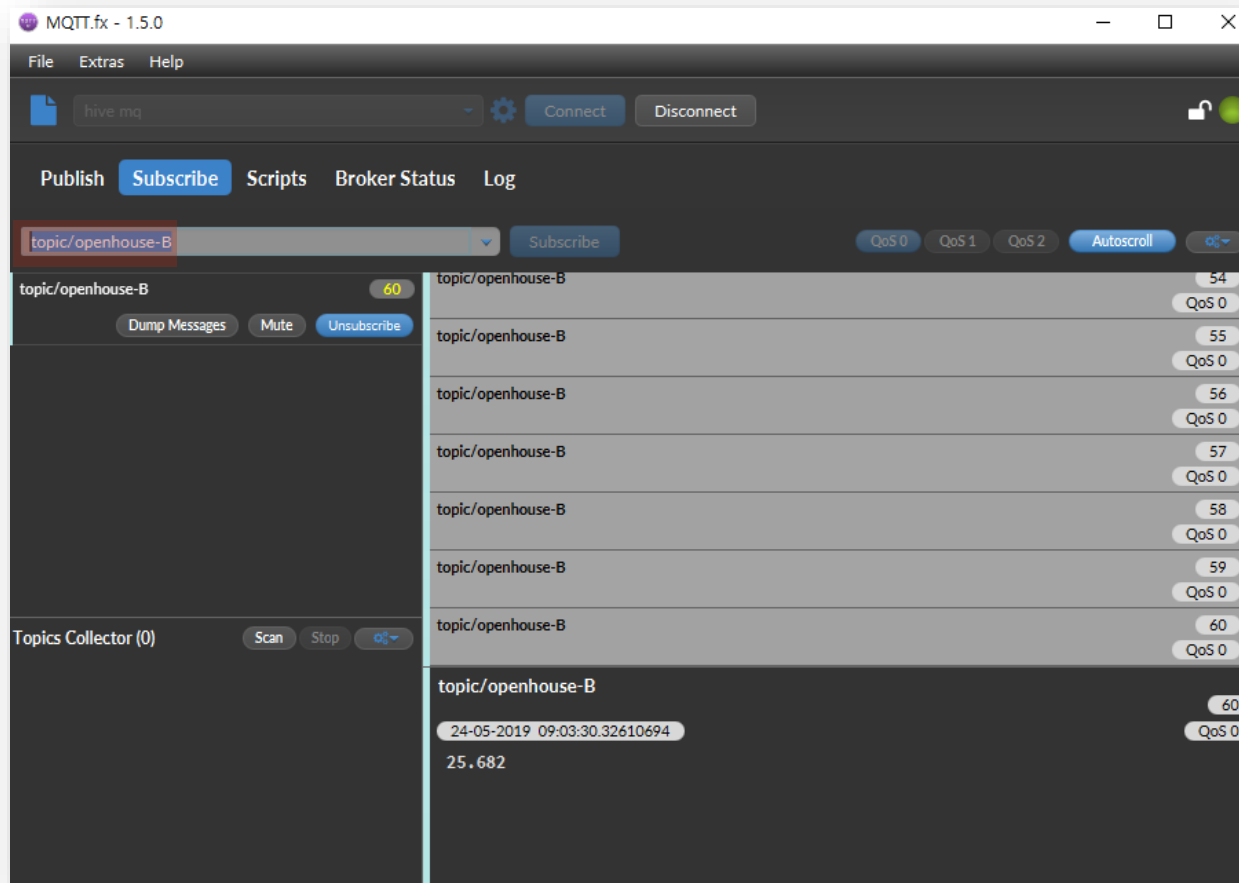
The serial monitor also shows the upload progress and completion message:

```
avrdude: verifying ...
avrdude: 17186 bytes of flash verified
avrdude done. Thank you.
```

MQTT Publish – Temperature

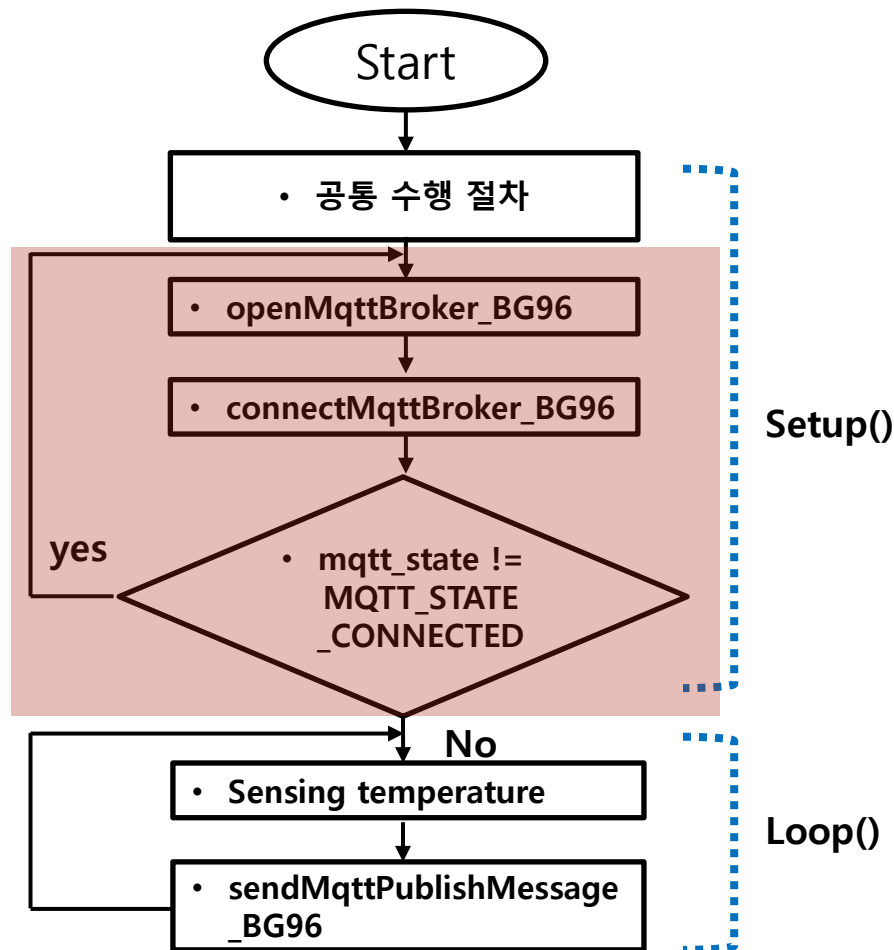
>> Subscribe

- Subscribe – “InputTopic” (수정된 토픽)



MQTT Publish – Temperature

MQTT Publish Flow Chart



```
do {  
    switch (mqtt_state) {  
        case MQTT_STATE_OPEN:  
            if (openMqttBroker_BG96(mqtt_broker_url, mqtt_broker_port) == RET_OK) {  
                MYPRINTF("[MQTT] Socket open success\r\n");  
                mqtt_state = MQTT_STATE_CONNECT;  
            } else {  
                MYPRINTF("[MQTT] Socket open failed\r\n");  
                mqtt_state = MQTT_STATE_DISCON;  
            }  
            break;  
  
        case MQTT_STATE_CONNECT:  
            if (connectMqttBroker_BG96(MQTT_CLIENTID, MQTT_USERID, MQTT_PASSWORD) == RET_OK) {  
                mqtt_state = MQTT_STATE_CONNECTED;  
            } else {  
                MYPRINTF("[MQTT] Connect failed\r\n");  
                mqtt_state = MQTT_STATE_DISCON;  
            }  
            break;  
  
        case MQTT_STATE_DISCON:  
            if (closeMqttBroker_BG96() == RET_OK) {  
                MYPRINTF("[MQTT] Disconnected\r\n");  
            }  
            mqtt_state = MQTT_STATE_OPEN;  
            break;  
  
        default:  
            mqtt_state = MQTT_STATE_OPEN;  
            break;  
    }  
} while (mqtt_state != MQTT_STATE_CONNECTED);
```

Arduino 기반 WIoT Cat.M1 Shield 활용과 실습 (2)

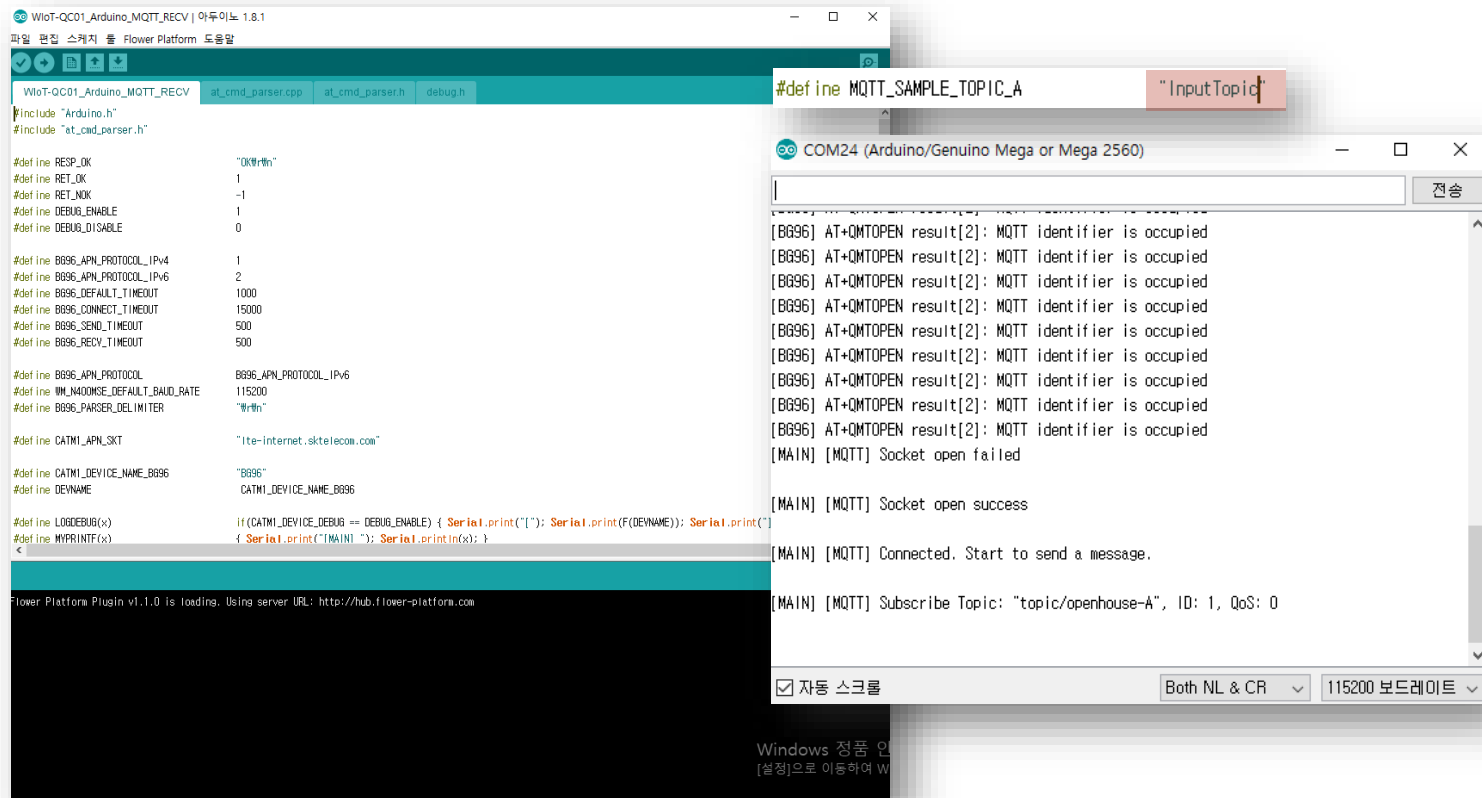
- 1) MQTT 이론 및 환경 설정
- 2) MQTT Local 테스트
- 3) MQTT Publish – Temperature
- 4) **MQTT Subscribe – Device Reset**



MQTT Subscribe – Device Reset

» “RESET” 메시지를 받으면 아두이노 재부팅

- 강의 자료 Example Wg_WIoT-QC01_Arduino_MQTT_RECV
- 토픽 수정 필요



The screenshot displays the Arduino IDE interface for the file `WIoT-QC01_Arduino_MQTT_RECV`. The code includes standard Arduino headers and defines various MQTT-related constants. A key definition is `#define MQTT_SAMPLE_TOPIC_A "InputTopic"`. The `main` function contains logic for MQTT initialization and subscription. Below the code editor, the Serial Monitor window for COM24 (Arduino/Genuino Mega or Mega 2560) shows the following output:

```
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[B096] AT+QMQTOPEN result[2]: MQTT identifier is occupied
[MAIN] [MQTT] Socket open failed

[MAIN] [MQTT] Socket open success

[MAIN] [MQTT] Connected. Start to send a message.

[MAIN] [MQTT] Subscribe Topic: "topic/openhouse-A", ID: 1, QoS: 0
```

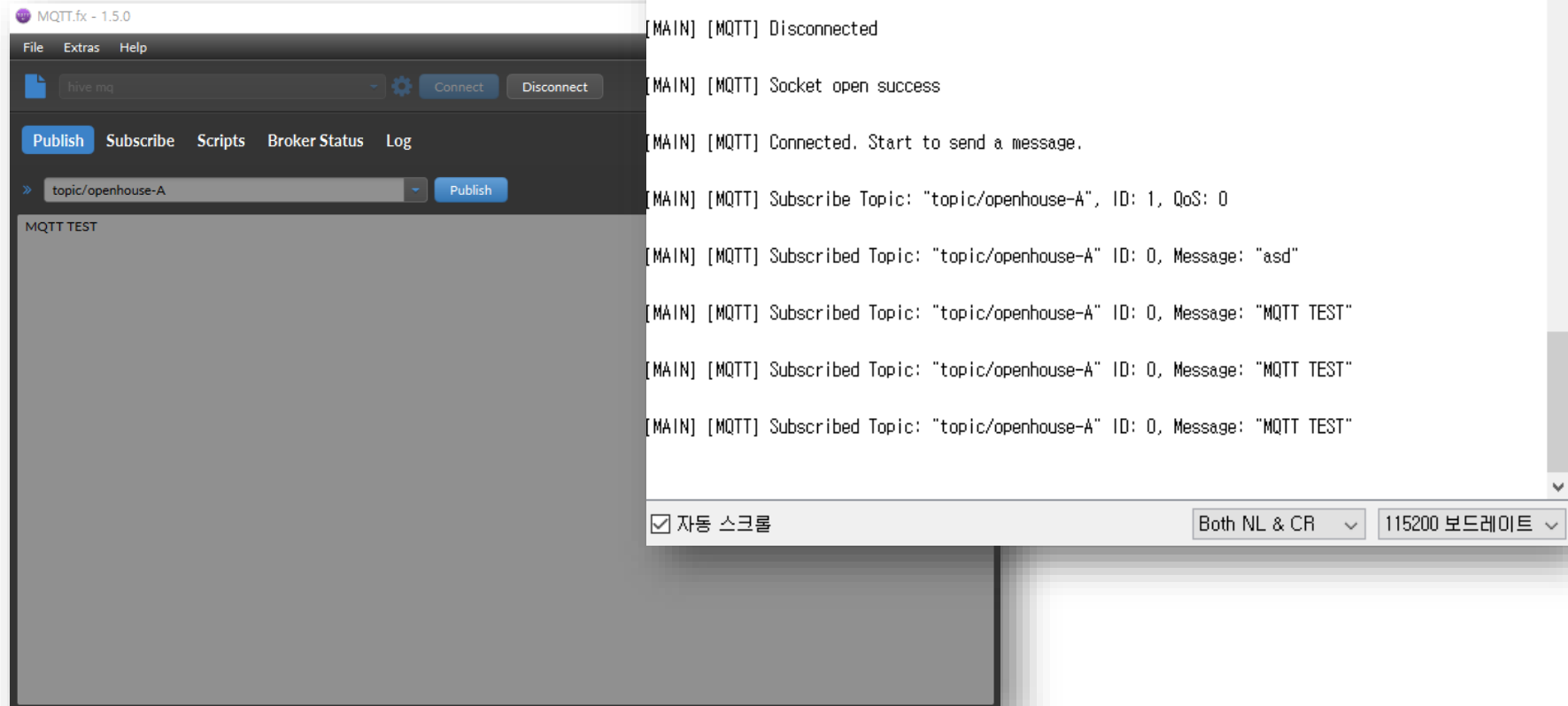
At the bottom of the Serial Monitor, the settings are configured to "자동 스크롤" (Auto Scroll), "Both NL & CR", and "115200 보드레이트" (115200 Baud Rate).

Windows 정품 인증
[설정]으로 이동하여 W

MQTT Subscribe – Device Reset

» MQTTfx Message Publish

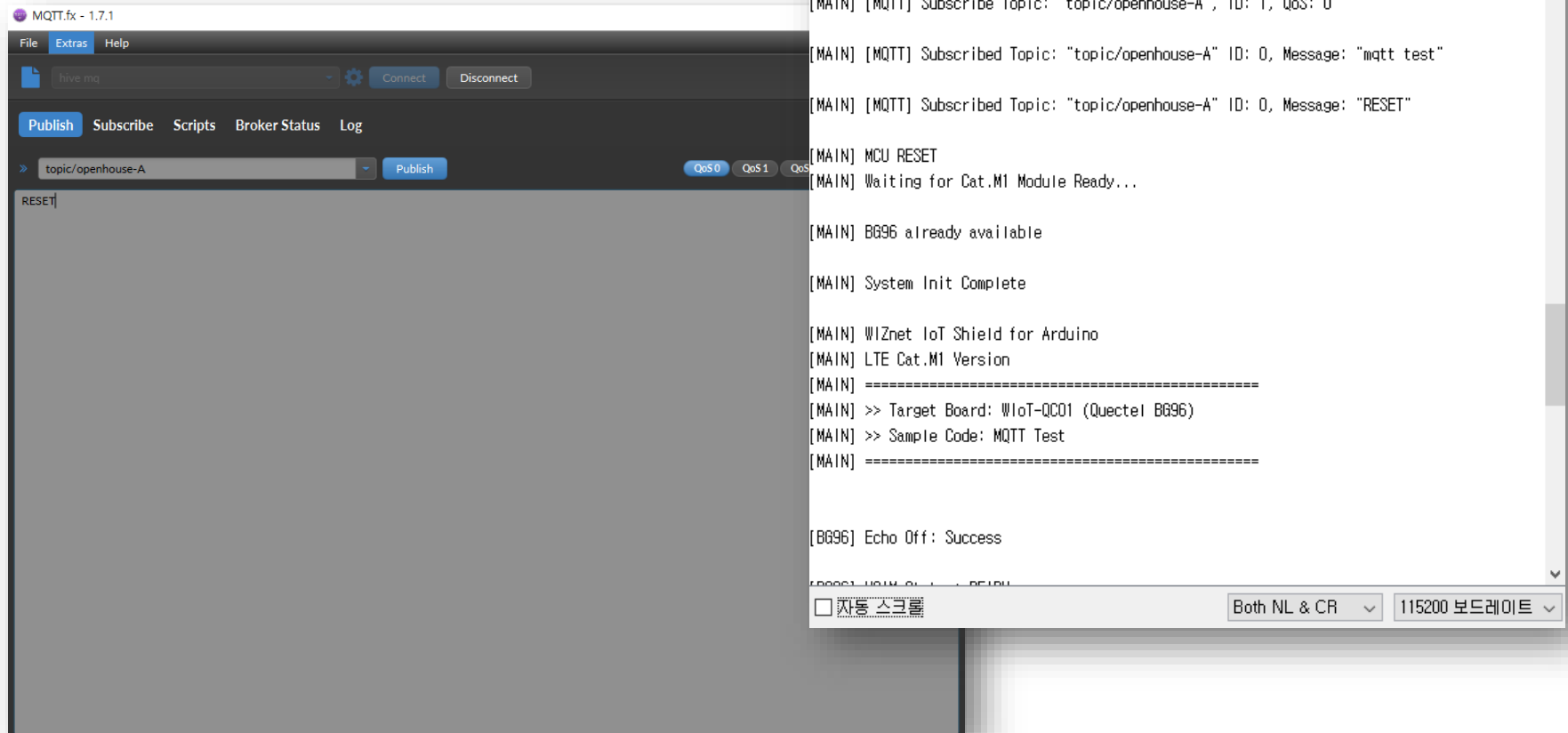
- Publish- “InputTopic” (수정한 토픽)
- Message – MQTT TEST



MQTT Subscribe – Device Reset

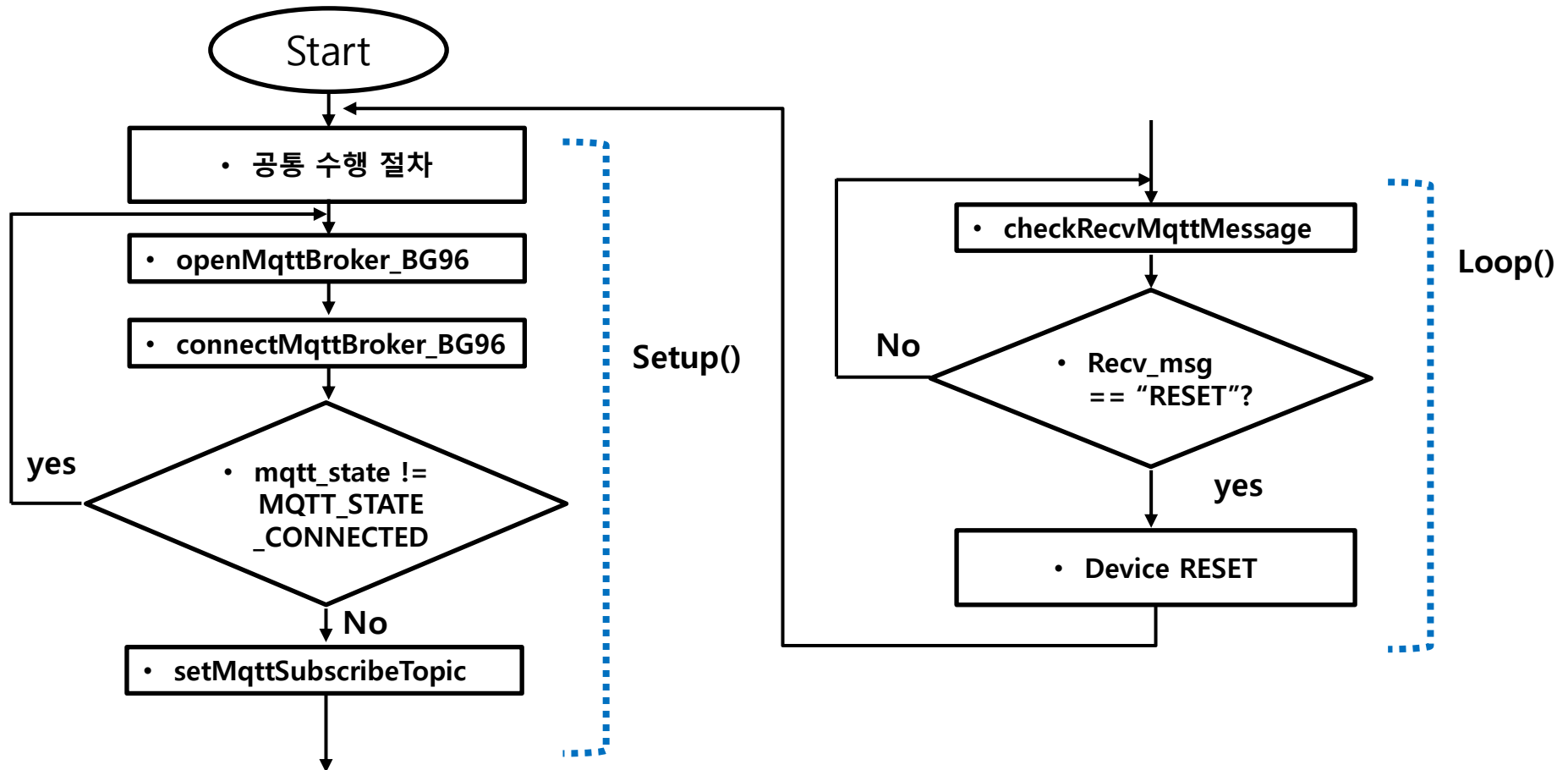
» MQTTfx Message Publish

- Publish- “InputTopic” (수정한 토픽)
- Message – RESET



AT Command Parser

>> 공통 수행 절차



SESSION 3, Q&A

자유롭게 질문 부탁드립니다.

WIZnet

