

# Telegram과 NodeMCU

2023학년도 2학기

김진숙 교수

동의과학대학 컴퓨터정보과

[jinsook@dit.ac.kr](mailto:jinsook@dit.ac.kr)

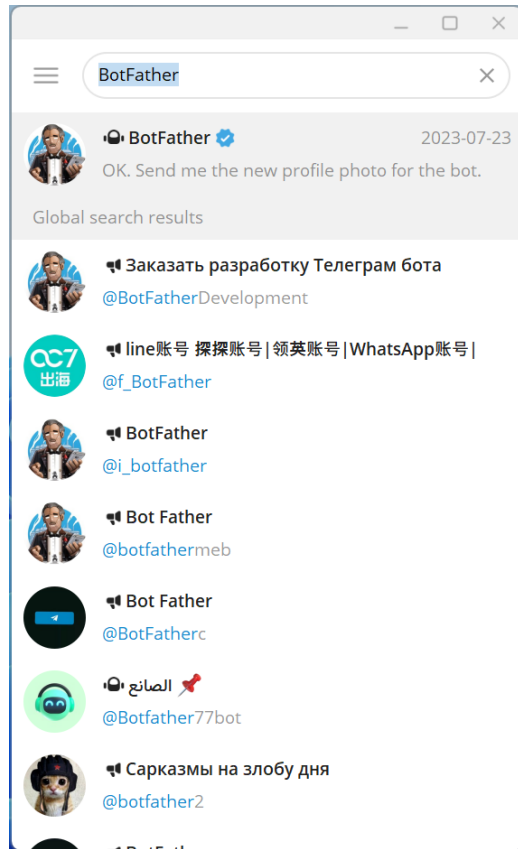
# 학습 내용

# 텔레그램에 메시지 보내기

- 주제 : nodeMCU에서 텔레그램에 문자 보내기
  - 텔레그램은 개인 서버 없이 무료로 사용할 수 있는 유일한 메신저
  - 양방향으로 메시지 통신 가능(nodeMCU <-> telegram)
  - nodeMCU에서 "Hello, World!" 문자를 텔레그램에게 전송
- 준비사항
  - 텔레그램 설치, 로그인, 설정
  - 라이브러리 : <UniversalTelegramBot.h>, <ESP8266WiFi.h>

# 텔레그램 설정

1. 텔레그램 설치 및 가입, 로그인
2. BotFather 검색하여 BotFather 실행



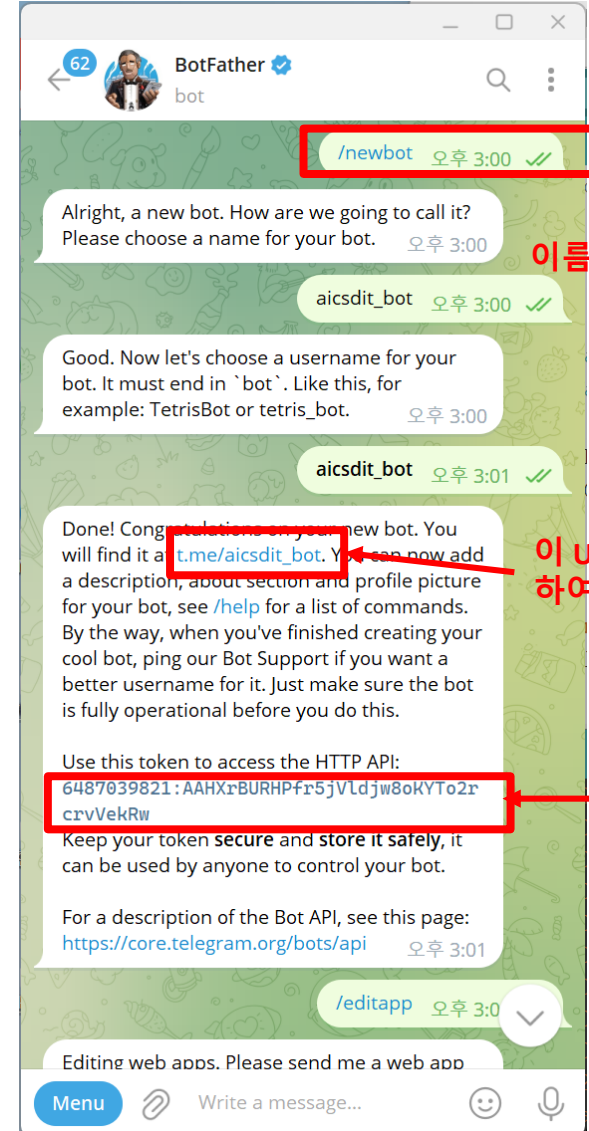
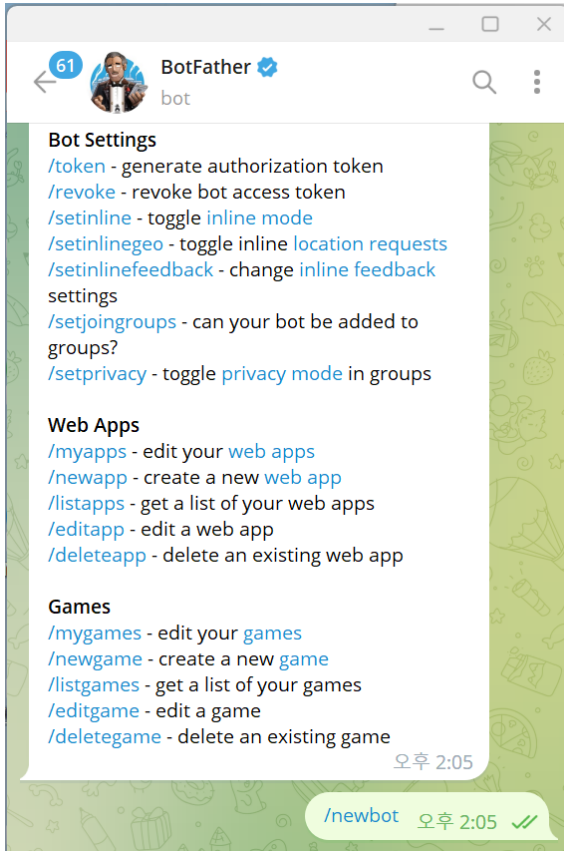
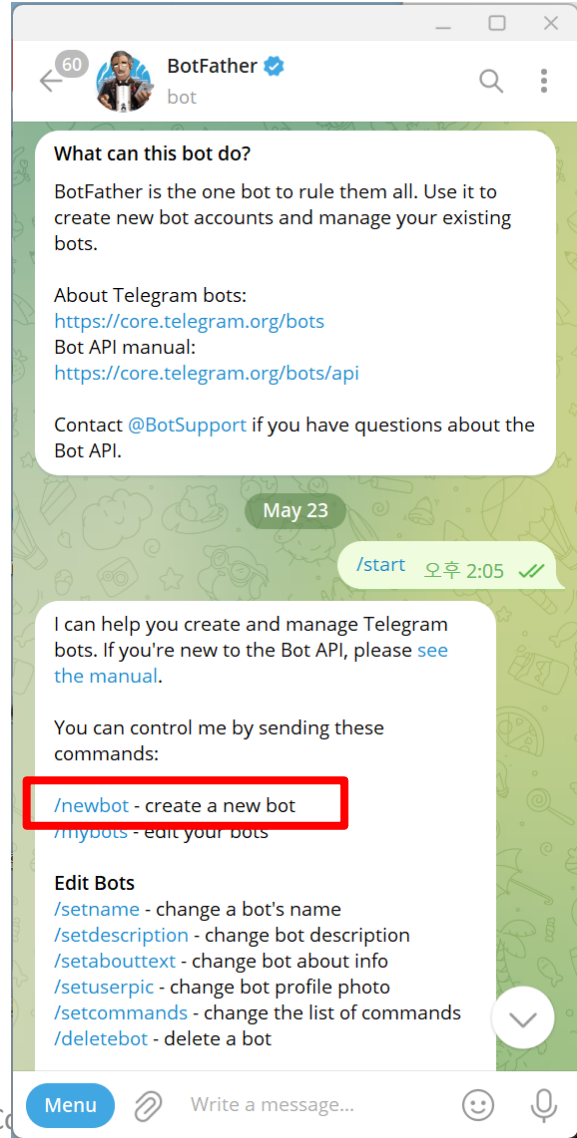
[참조]

<https://core.telegram.org/bots/features#botfather>

<https://core.telegram.org/bots/api#authorizing-your-bot>

# 텔레그램 설정

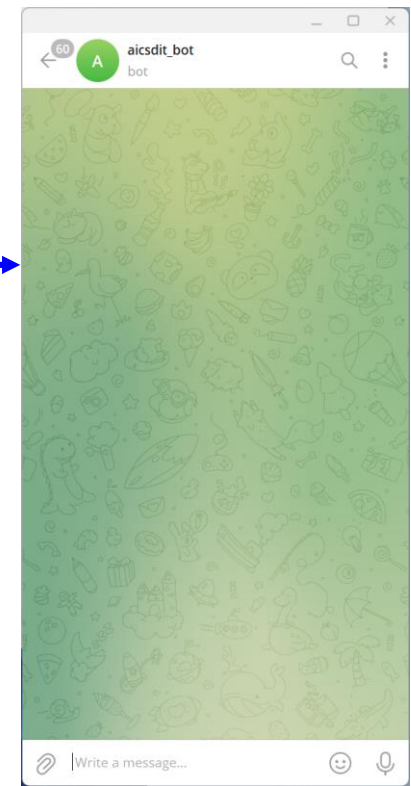
## 3. BotFather에서 새로운 봇 생성 – BOT\_TOKEN 알아내기



이름은 XXX\_bot 으로 생성

이 URL을 클릭 하여 봇 시작

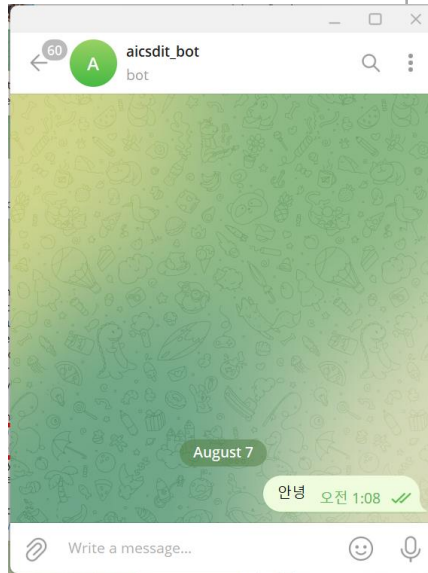
BOT\_TOKEN



# 텔레그램 설정

## ■ CHAT\_ID 알아내기

- 처음 접속하면 응답이 오지 않음
- 생성된 봇에서 문자를 보내고 난 후에 웹브라우저에서 다음의 사이트에 접속(request 만들기) 하면 파악 가능



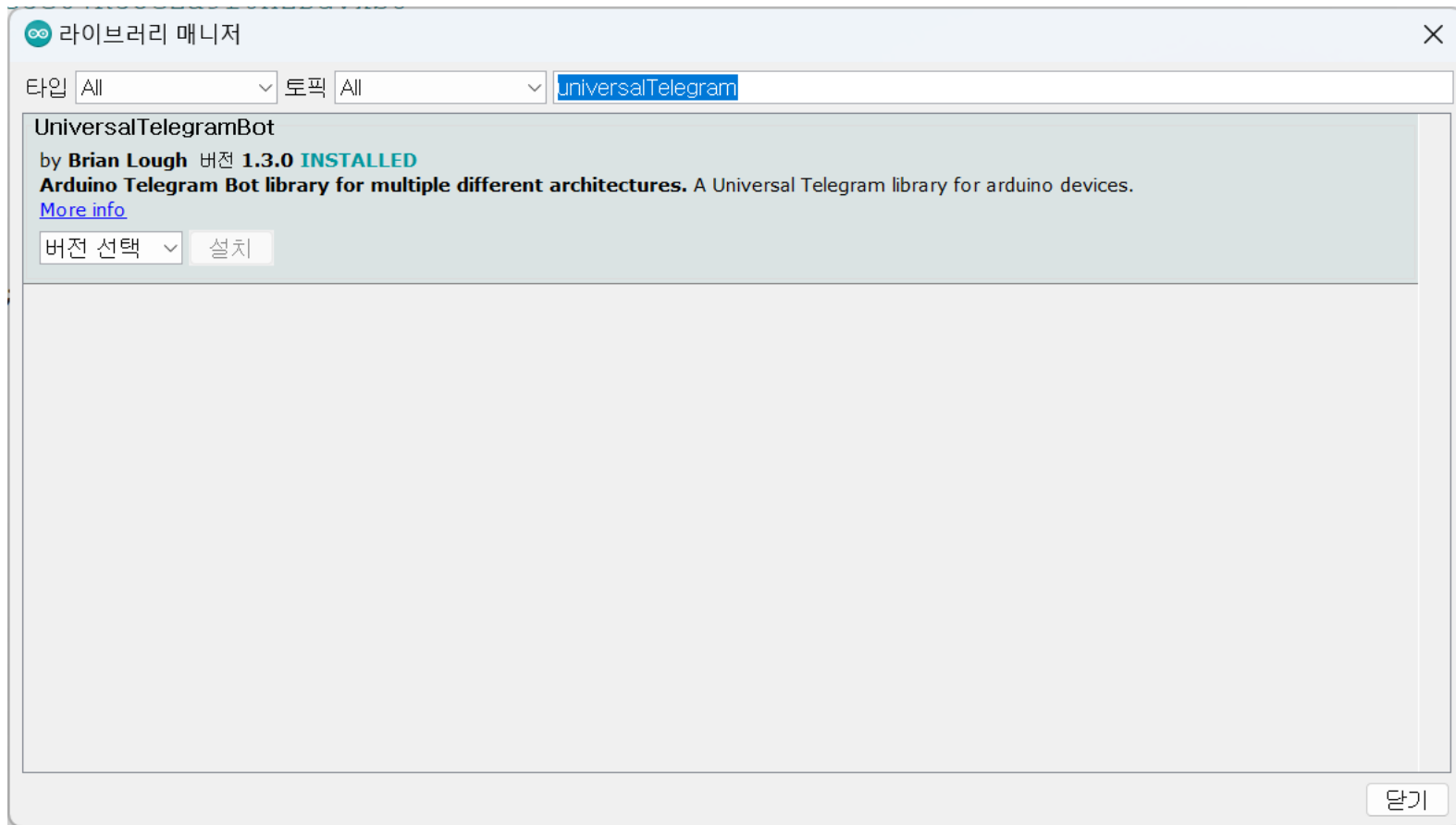
```
{
  "ok": true,
  "result": [
    {
      "update_id": 103755244,
      "message": {
        "message_id": 64,
        "from": {
          "id": 392679036,
          "is_bot": false,
          "first_name": "Jinsook",
          "last_name": "Kim",
          "language_code": "ko",
          "chat": {
            "id": 392679036,
            "first_name": "Jinsook",
            "last_name": "Kim",
            "type": "private",
            "date": 1691326540,
            "text": "hi"
          }
        },
        "date": 1691326540,
        "text": "hi"
      }
    }
  ]
}
```

**CHAT\_ID**

<https://api.telegram.org/bot발급받은HTTP-API키/getUpdates>

# 라이브러리 : <UniversalTelegramBot.h>

- Telegram은 봇 생성을 허용하는 인스턴트 메시징 서비스
- 봇을 생성하려면 액세스 토큰이 필요



# 라이브러리 : <UniversalTelegramBot.h>

## ■ 주요 기능

<https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot>

| 주요 함수   | 설명  | 비고 |
|---|---|----|
| int <b>getUpdates</b> (long offset)   | <ul style="list-style-type: none"> <li>Telegram에서 대기 중인 메시지를 가져오고 bot.messages에 저장</li> <li>오프셋은 bot.last_message_received + 1 로 설정</li> <li>수신된 새 메시지 수를 반환</li> </ul> |    |
| bool <b>sendMessage</b> (String chat_id, String text, String parse_mode = "")   | <ul style="list-style-type: none"> <li>chat_id로 메시지를 보냅니다. 메시지 전송 여부를 반환</li> <li>참고: 봇은 먼저 메시지를 보낸 경우에만 메시지를 보낼 수 있음</li> </ul>  |    |
| bot. <b>sendPhoto</b> (chat_id, test_photo_url, "text");  | <ul style="list-style-type: none"> <li>chat_id로 url의 사진을 전송</li> </ul>  |    |
| bot. <b>sendPhotoByBinary</b> (chat_id, "image/jpeg", myFile.size(), isMoreDataAvailable, getNextByte, nullptr, nullptr); | <ul style="list-style-type: none"> <li>chat_id로 파일 사진을 전송</li> </ul>  |    |
| UniversalTelegramBot<br><b>bot</b> (BOT_TOKEN, secured_client);   | <ul style="list-style-type: none"> <li>Telegram 봇 생성</li> </ul>   |    |



# 텔레그램에 메시지 보내기

## 1) 프로그래밍

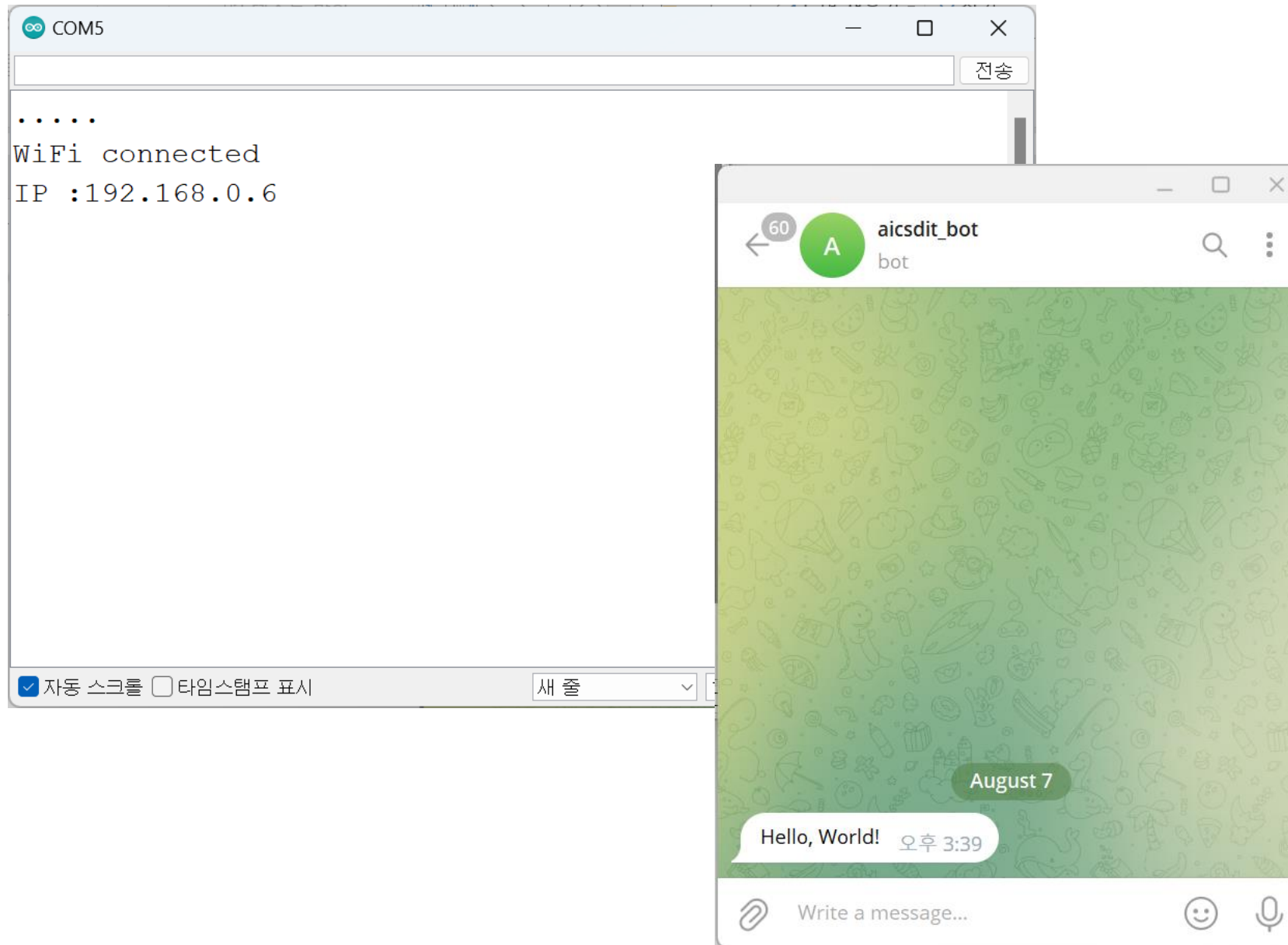
```
1 #include <UniversalTelegramBot.h>
2 #include <WiFiClientSecure.h>
3 #include <ESP8266WiFi.h>
4
5 const char *ssid = "iptime";
6 const char *password = "";
7
8 #define BOT_TOKEN "6487039821:AAHXrBURHPfr5jVldjw8oKYTo2rcrvVekRw"
9 #define CHAT_ID "392679036"
10
11 //X.509는 암호학에서 공개키 인증서와 인증알고리즘의 표준 가운데에서 공개 키 기반 (PKI) 의 ITU-T 표준
12 X509List cert (TELEGRAM_CERTIFICATE_ROOT);
13
14 WiFiClientSecure secured_client;
15 UniversalTelegramBot bot (BOT_TOKEN, secured_client);
```

# 텔레그램에 메시지 보내기

## 1) 프로그래밍

```
17 void setup() {  
18     Serial.begin(115200);  
19  
20     //컴퓨터의 시간을 네트워크 시간과 맞추기 위한 설정  
21     configTime(0, 0, "pool.ntp.org");  
22  
23     // Telegram의 API에 대한 루트 인증서 추가(api.telegram.org)  
24     secured_client.setTrustAnchors(&cert);  
25  
26     WiFi.mode(WIFI_STA);  
27     WiFi.begin(ssid, password);  
28  
29     while(WiFi.status() != WL_CONNECTED) {  
30         Serial.print(".");  
31         delay(500);  
32     }  
33     Serial.println("");  
34     Serial.println("WiFi connected");  
35     Serial.print("IP :"); Serial.println(WiFi.localIP());  
36 }  
37  
38 void loop() {  
39     //chat_id로 메시지를 보내고 메시지 전송 여부를 반환  
40     bot.sendMessage(CHAT_ID, "Hello, World!", "");  
41  
42     delay(10000); //10초  
43 }
```

## 2) 실행

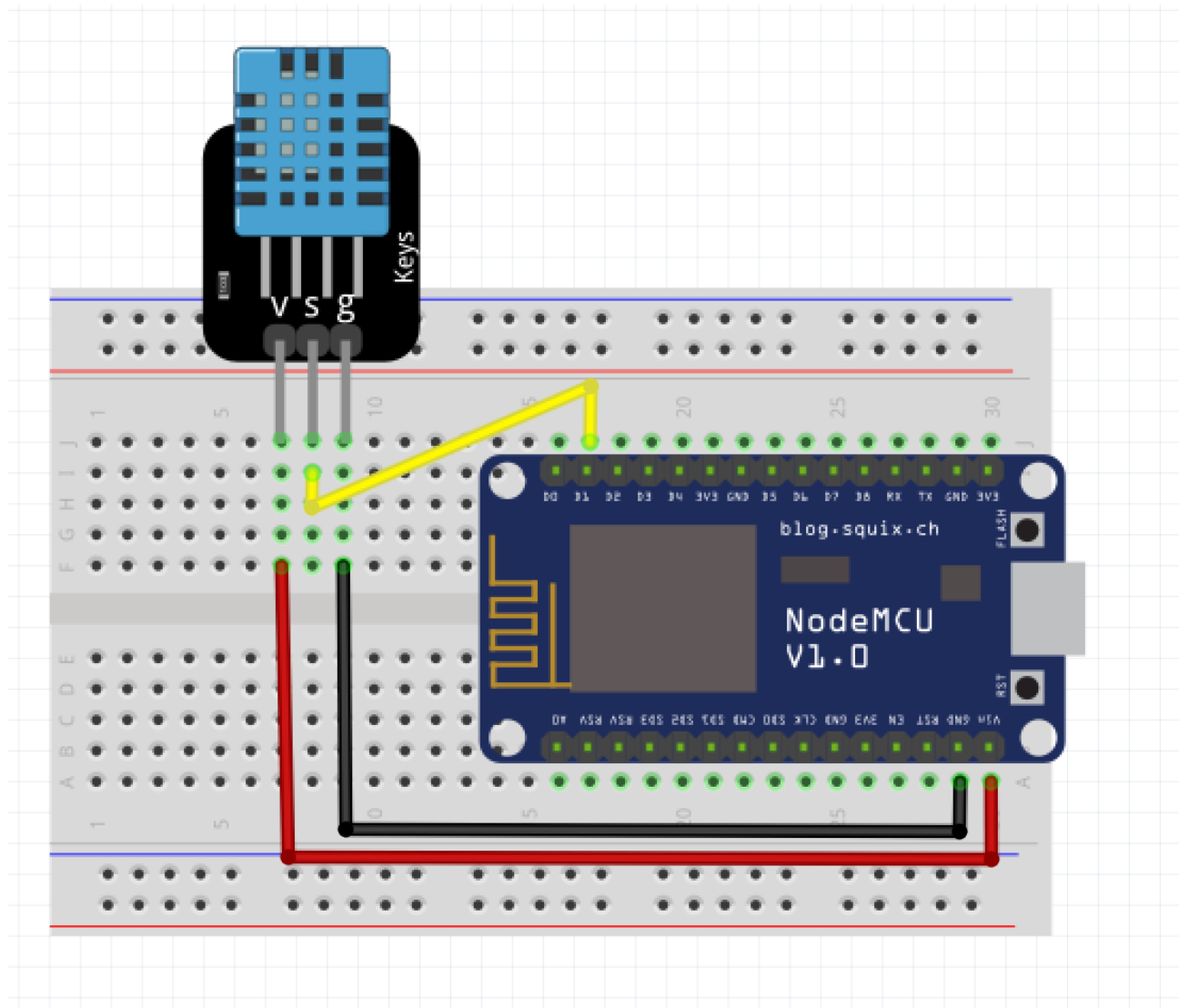


# 텔레그램에 온습도 데이터 전송하기

- 주제 : 온습도 데이터 전송하기
  - nodeMCU에서 측정한 온습도 데이터를 텔레그램 문자로 보내기
- 준비사항
  - 텔레그램 설치, 로그인, 설정
  - 라이브러리 : <UniversalTelegramBot.h>, <ESP8266WiFi.h>, <DHT.h>

# 텔레그램에 온습도 데이터 전송하기

## 1) 회로 구성



# 텔레그램에 온습도 데이터 전송하기

## 2) 프로그래밍

```
1 #include <DHT.h>
2 #include <UniversalTelegramBot.h>
3 #include <WiFiClientSecure.h>
4 #include <ESP8266WiFi.h>
5
6 const char *ssid = "iptime";
7 const char *password = "";
8
9 #define BOT_TOKEN "6487039821:AAHXrBURHPfr5jVldjw8oKYTo2rcrvVekRw"
10 #define CHAT_ID "392679036"
11
12 int dhtPin = D1;
13 X509List cert(TELEGRAM_CERTIFICATE_ROOT);
14 DHT dht(dhtPin, DHT11);
15
16 WiFiClientSecure secured_client;
17 UniversalTelegramBot bot(BOT_TOKEN, secured_client);
```

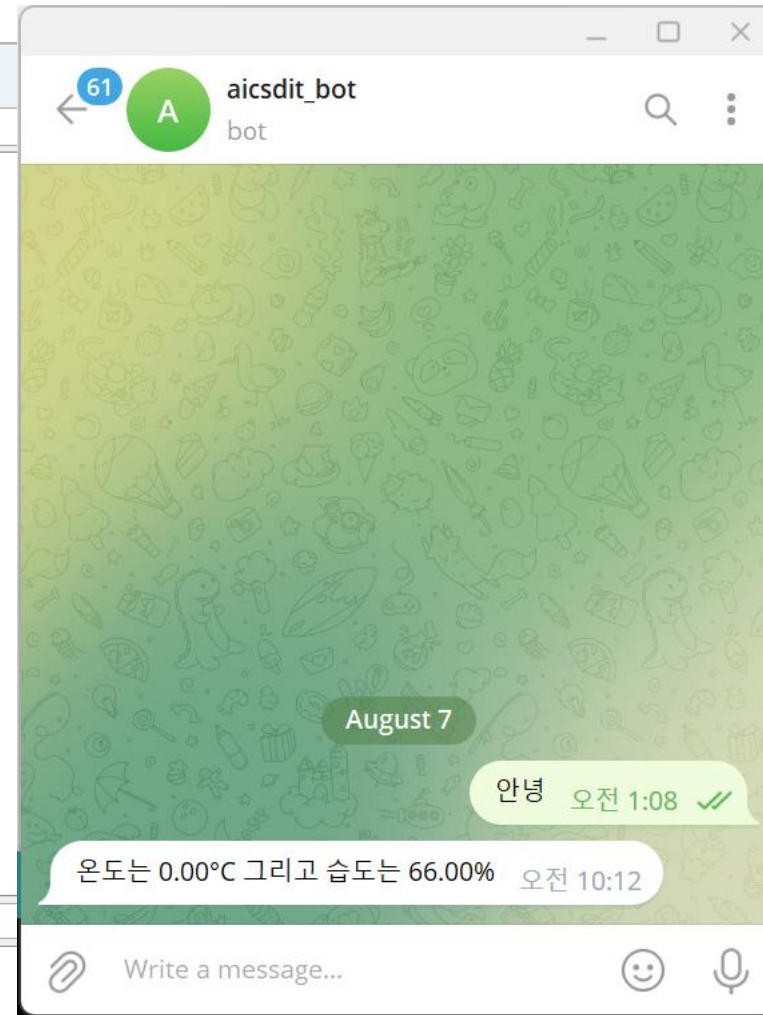
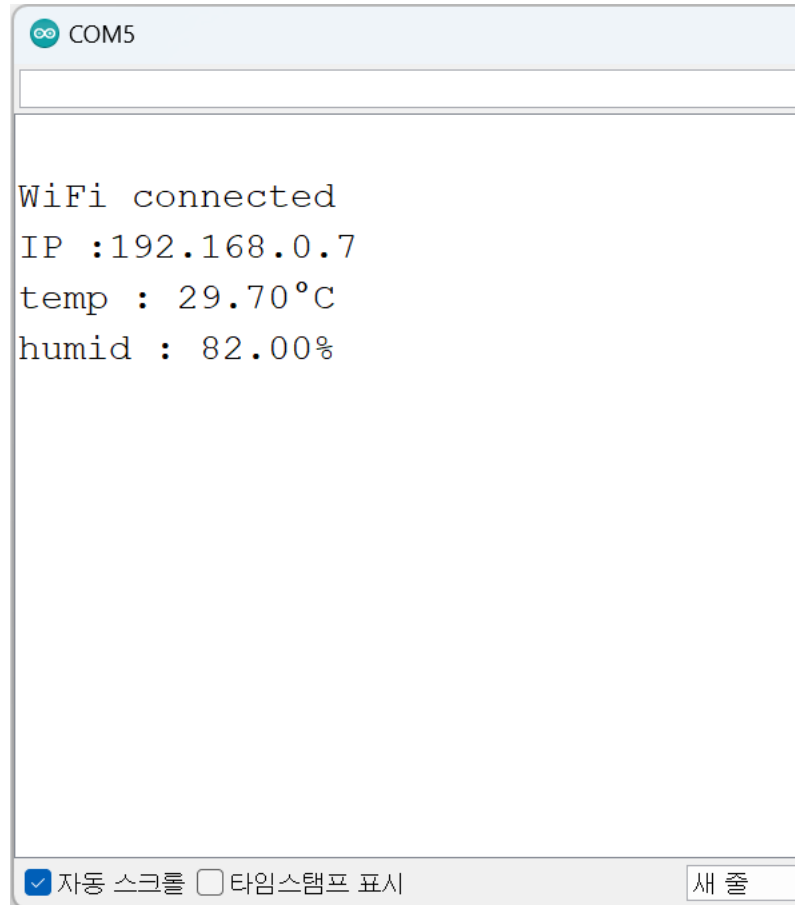
# 텔레그램에 온습도 데이터 전송하기

## 2) 프로그래밍

```
19 void setup() {  
20     Serial.begin(115200);  
21     configTime(0, 0, "pool.ntp.org");  
22     secured_client.setTrustAnchors(&cert);  
23  
24     WiFi.mode(WIFI_STA);  
25     WiFi.begin(ssid, password);  
26  
27     while(WiFi.status() != WL_CONNECTED) {  
28         Serial.print(".");  
29         delay(500);  
30     }  
31     Serial.println("");  
32     Serial.println("WiFi connected");  
33     Serial.print("IP :"); Serial.println(WiFi.localIP());  
34 }  
35  
36 void loop() {  
37     float temp = dht.readTemperature();  
38     float humid = dht.readHumidity();  
39     Serial.print("temp : "); Serial.print(temp);  
40     Serial.print("\xc2\x00"); Serial.println("C\t");  
41     Serial.print("humid : "); Serial.print(humid); Serial.println("%");  
42  
43     String msg = "온도는 " + String(temp) + "\xc2\x00" + "C 그리고 습도는 " + String(humid) + "%";  
44     bot.sendMessage(CHAT_ID, msg, "Markdown");  
45  
46     delay(10000); //10초  
47 }
```

# 텔레그램에 온습도 데이터 전송하기

## 3) 실행



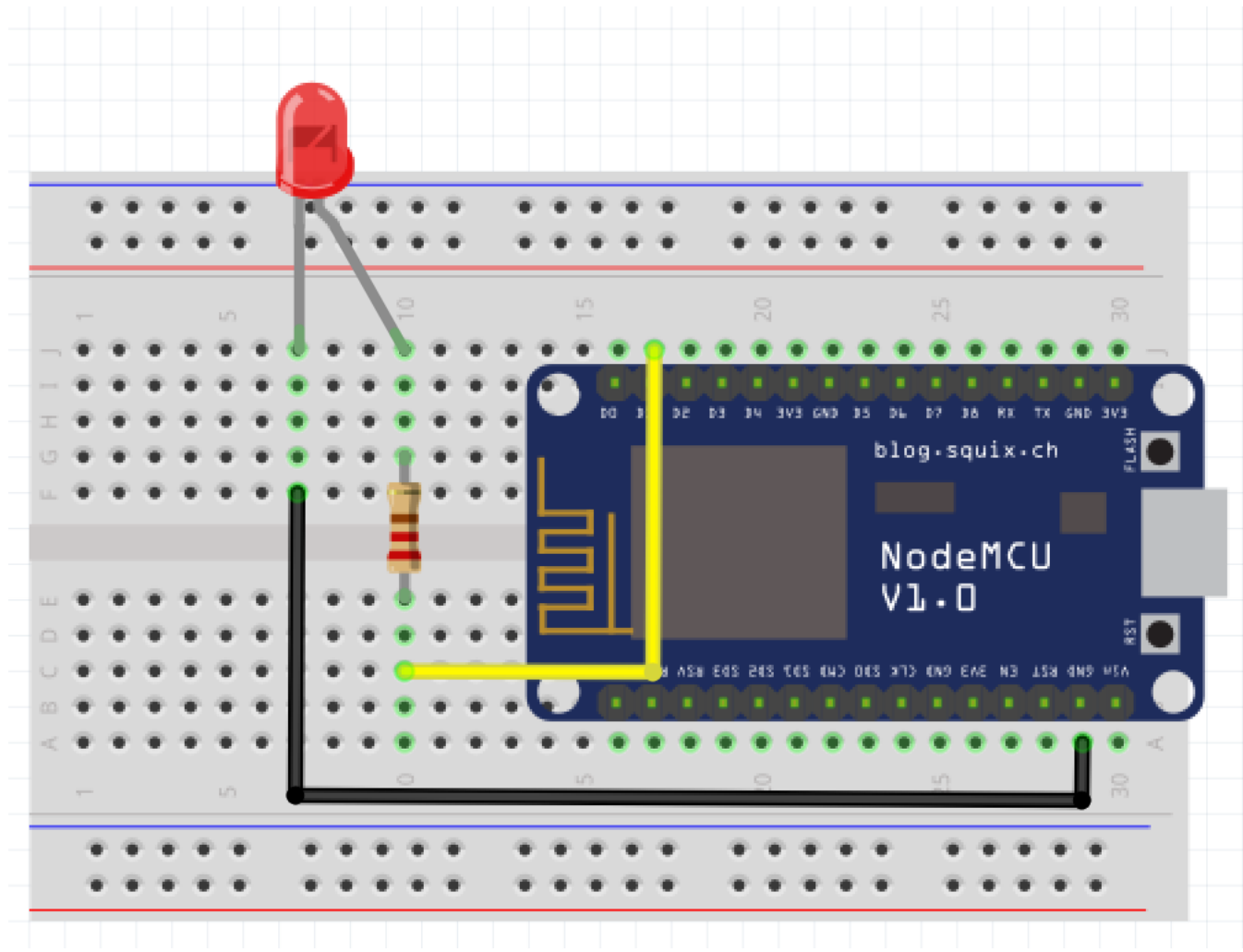


# 텔레그램에서 LED ON/OFF 제어하기

- 주제 : LED ON/OFF 제어하기
  - 텔레그램 챗봇에서 명령문(LEDON, LEDOFF)을 nodeMCU에 보내 LED를 제어
- 준비사항
  - 텔레그램 설치, 로그인, 설정
  - 라이브러리 : <UniversalTelegramBot.h>, <ESP8266WiFi.h>

# 텔레그램에서 LED ON/OFF 제어하기

## 1) 회로 구성



# 텔레그램에서 LED ON/OFF 제어하기

## 2) 프로그래밍

```
1 #include <UniversalTelegramBot.h>
2 #include <WiFiClientSecure.h>
3 #include <ESP8266WiFi.h>
4
5 const char *ssid = "iptime";
6 const char *password = "";
7
8 #define BOT_TOKEN "6487039821:AAHXrBURHPfr5jVldjw8oKY"
9
10 int ledPin = D2;
11
12 X509List cert(TELEGRAM_CERTIFICATE_ROOT);
13 WiFiClientSecure secured_client;
14 UniversalTelegramBot bot(BOT_TOKEN, secured_client);
```

```
16 void setup() {
17     Serial.begin(115200);
18     pinMode(ledPin, OUTPUT);
19
20     // Telegram의 API에 대한 루트 인증서 추가(api.telegram.org)
21     secured_client.setTrustAnchors(&cert);
22
23     WiFi.mode(WIFI_STA);
24     WiFi.begin(ssid, password);
25
26     while(WiFi.status() != WL_CONNECTED) {
27         Serial.print(".");
28         delay(500);
29     }
30     Serial.println("");
31     Serial.println("WiFi connected");
32     Serial.print("IP :"); Serial.println(WiFi.localIP());
33 }
```

# 텔레그램에서 LED ON/OFF 제어하기

## 2) 프로그래밍

```
35 void loop() {  
36     //텔레그램에서 온 메시지를 받고 메시지 갯수 반환  
37     int numNewMsg = bot.getUpdates(bot.last_message_received + 1);  
38  
39     for(int i=0; i<numNewMsg; i++){  
40         //전송된 메시지에서 chat_id 알아내기  
41         String chat_id = bot.messages[i].chat_id;  
42         String text = bot.messages[i].text;  
43         Serial.println(text);  
44         Serial.println(chat_id);  
45  
46         if (text == "on") {  
47             digitalWrite(ledPin, HIGH);  
48             bot.sendMessage(chat_id, "LED turned ON");  
49         }  
50         if (text == "off") {  
51             digitalWrite(ledPin, LOW);  
52             bot.sendMessage(chat_id, "LED turned OFF");  
53         }  
54     }  
55  
56 }
```

# 텔레그램에서 LED ON/OFF 제어하기

다음과 같이 동작하도록 프로그램을 변경하세요.

- 명령어가 nodeMCU에 들어왔을 때만 응답하도록 한다.
- 잘못된 명령어일 경우 잘못되었다는 메시지를 전송한다.

