



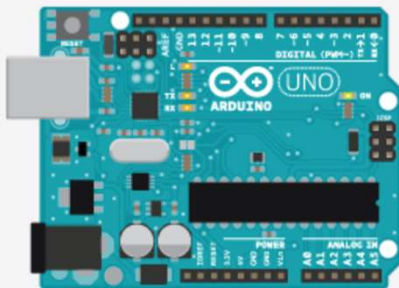
아두이노 실습

# Chapter 14

## 블루투스 모듈 제어해보기



WHAT IS ARDUINO?



Step 1. AT명령어를 사용하여 블루투스 모듈 제어하기

Step 2. BLE앱을 사용하여 블루투스를 연결하고 아두이노와 채팅하기

## AT명령어를 사용하여 블루투스 모듈 제어하기

### 블루투스(Bluetooth)

- 대표적인 근거리 무선 통신 규격
- Bluetooth Classic
  - 기존의 블루투스 (4.0 이전)
- Bluetooth 4.0
  - 저전력 (Low Energy)
- Bluetooth 5.0
  - IOT 사물인터넷
  - 전송 속도, 전송가능거리 증가



## AT명령어를 사용하여 블루투스 모듈 제어하기

### 블루투스 프로파일(Profile)

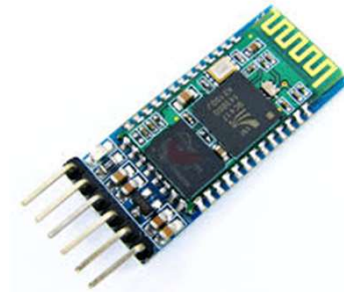
- 블루투스 장치를 어떻게 사용할 지 정의  
(예, 헤드폰, 리모컨 등)
- SPP(Serial Port Profile)
  - 블루투스 장치로 시리얼 통신을 하는 것
  - Bluetooth Classic에서만 지원
  - 활용하기가 간편함
- Custom Profile
  - Bluetooth 4.0에서 새롭게 등장
  - 개발자가 직접 블루투스 장치의 용도를 결정함
  - 활용하기가 어려움



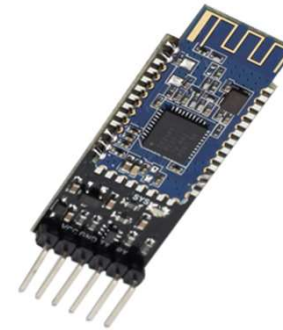
## AT명령어를 사용하여 블루투스 모듈 제어하기

### 아두이노와 블루투스

- HC-06
  - Bluetooth Classic
  - SPP지원: 시리얼 포트로 인식
  - Android에서만 가능
- HM-10
  - Bluetooth 4.0
  - SPP로 사용할 수 있는 Custom Profile 제공
  - Android/iOS 모두 사용 가능



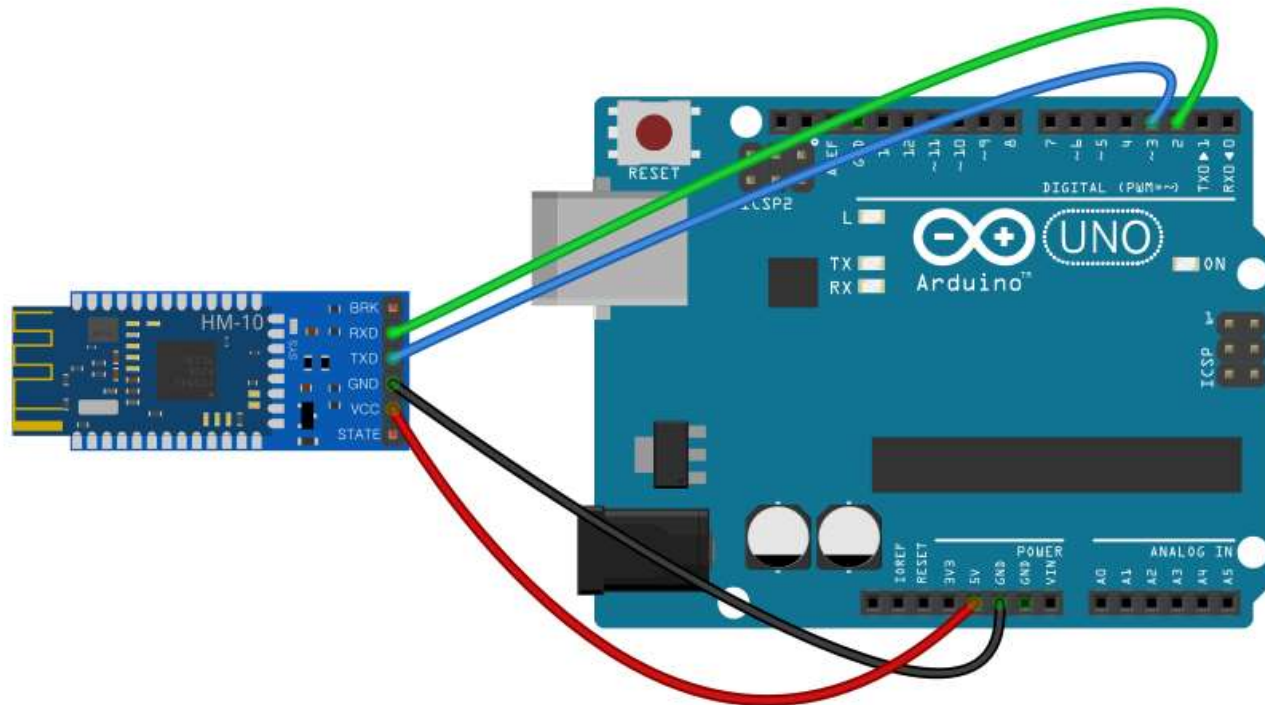
HC-06



HM-10

## AT명령어를 사용하여 블루투스 모듈 제어하기

### 회로구현



## AT명령어를 사용하여 블루투스 모듈 제어하기

### SoftwareSerial Class

- 아두이노에서 소프트웨어 시리얼 통신을 위해 사용되는 Class
  - `#include <SoftwareSerial.h>` 추가
- 선언시 사용되는 TX, RX 핀 지정
  - 예) `SoftwareSerial softSerial(3, 2);` // 3번 핀을 RX, 2번 핀을 TX로 사용
- 함수는 Serial Class와 동일

## AT명령어를 사용하여 블루투스 모듈 제어하기

### available

- 시리얼 통신으로 받은 데이터 크기를 확인한다.
- C언어 문법
  - `int available();`
  - 반환값 있음, 인자 없음
- 사용방법
  - 반환: 받은 데이터 크기를 바이트 단위로 알려줌
  - 예) `int num = Serial.avilable(); // num변수에 받은 데이터 크기를 저장`



## AT명령어를 사용하여 블루투스 모듈 제어하기

### write

- 시리얼 통신으로 데이터 1 byte를 전송한다.
- C언어 문법
  - `void write(data);`
  - 반환값 없음, 인자 1개
- 사용방법
  - data: 보낼 1byte 데이터
  - 예) `Serial.write('A');` // A 문자 하나를 시리얼 통신으로 보낸다.

## AT명령어를 사용하여 블루투스 모듈 제어하기

### read

- 시리얼 통신으로 데이터 1 byte를 받는다.
- C언어 문법
  - `int read();`
  - 반환값 있음, 인자 없음
- 사용방법
  - 반환값: 받은 1byte 데이터
  - 예) `int data = Serial.read();` // 시리얼 통신으로 받은 1 byte 데이터를 data변수에 저장한다.

## AT명령어를 사용하여 블루투스 모듈 제어하기

### 소스코딩

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(3, 2);

void setup() {
  Serial.begin(9600);
  BTSerial.begin(9600);
}

void loop() {
  if (BTSerial.available()) {
    Serial.write(BTSerial.read());
  }
  if (Serial.available()){
    BTSerial.write(Serial.read());
  }
}
```

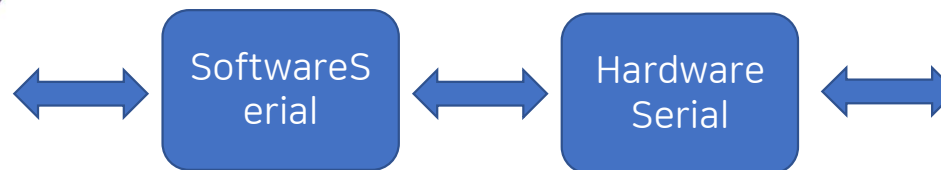
## AT명령어를 사용하여 블루투스 모듈 제어하기

### AT Command

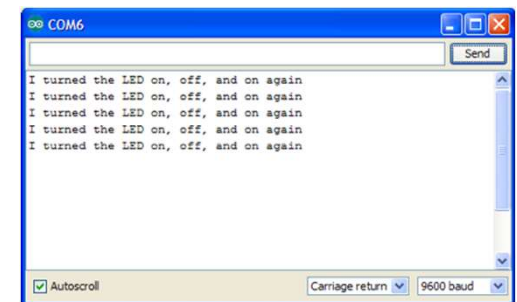
- HM-10을 설정하기 위해 사용하는 기능
- AT Command
  - AT+VERSION: 연결확인
  - AT+NAME: 장치 검색 이름 설정
  - AT+BAUD: 보레이트 설정



HM-10



아두이노 스케치

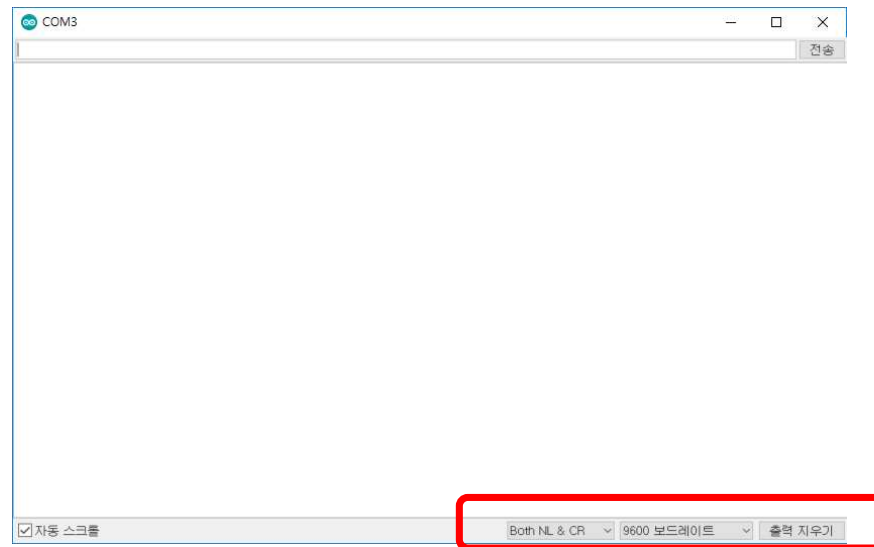


시리얼 모니터

## AT명령어를 사용하여 블루투스 모듈 제어하기

### 블루투스제어

- 시리얼 모니터에서 'Both NL & CR' 선택, '9600 보드레이트' 선택합니다.



## AT명령어를 사용하여 블루투스 모듈 제어하기

### 블루투스제어

AT+VERSION 입력

```
COM3
AT+VERSION
```

응답 : +VERSION=  
BLE 버전

```
COM3
+VERSION=Firmware V3.0.6,Bluetooth V4.0 LE
```

```
COM3
AT+NAMEMakist
```

AT+NAME이름

```
COM3
+VERSION=Firmware V3.0.6,Bluetooth V4.0 LE
+NAME=Makist
OK
```

응답 : +NAME=이름OK

이름은 영문자, 숫자, \_ 등 입력가능하고, 최소 6자리 이상 입력합니다.  
AT 명령어는 BLE모듈의 펌웨어 버전에 따라 다를 수 있습니다.

## BLE앱을 사용하여 블루투스를 연결하고 아두이노와 채팅하기

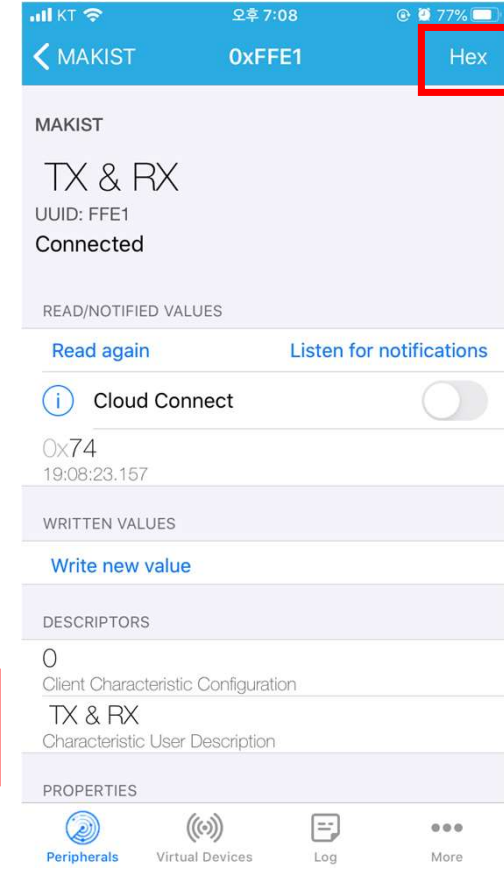
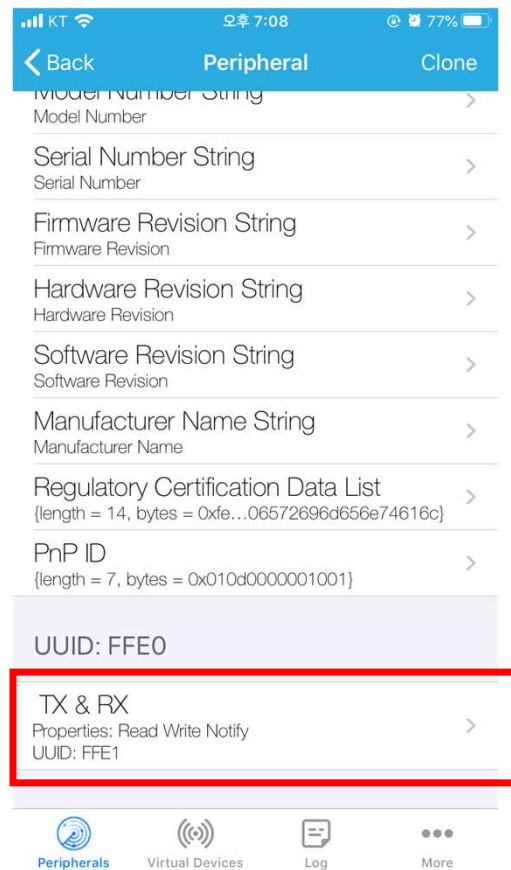
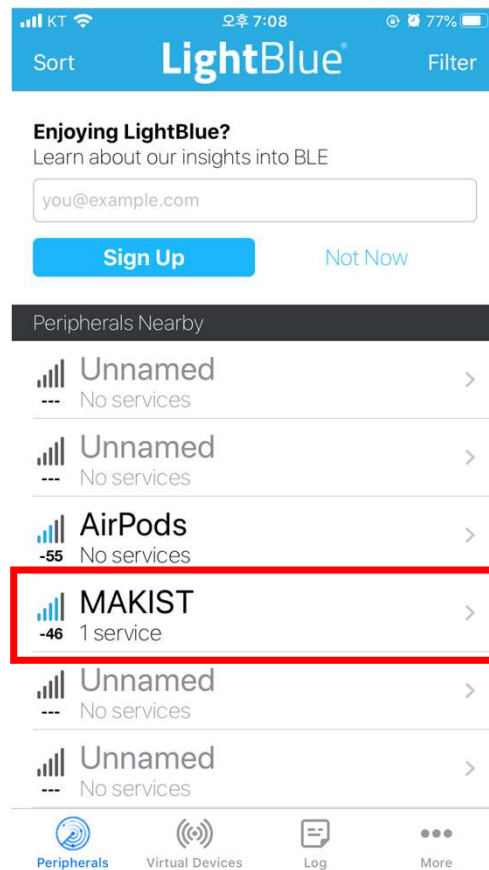
### 앱설치

- 아이폰은 앱스토어, 안드로이드는 플레이스토어에서 LightBlue 다운



## BLE앱을 사용하여 블루투스를 연결하고 아두이노와 채팅하기

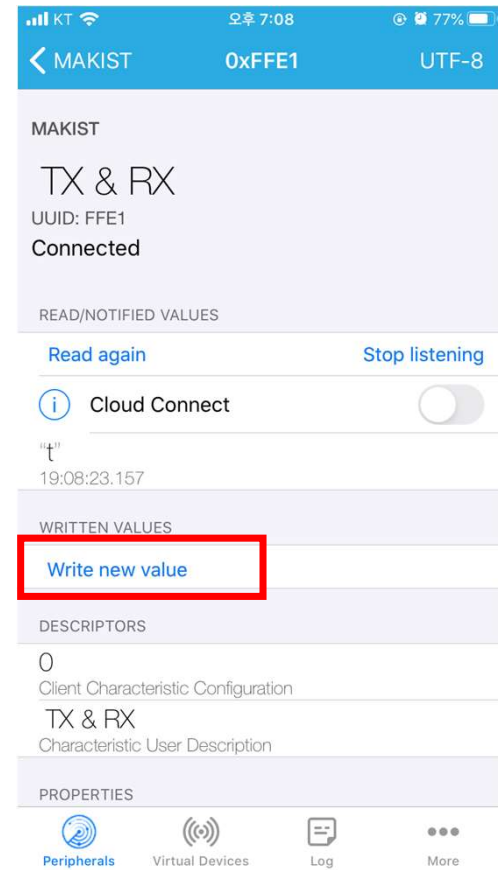
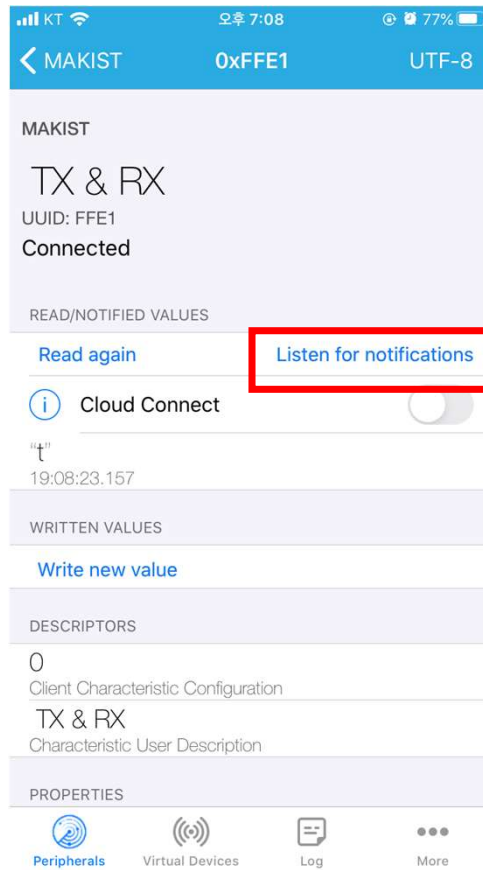
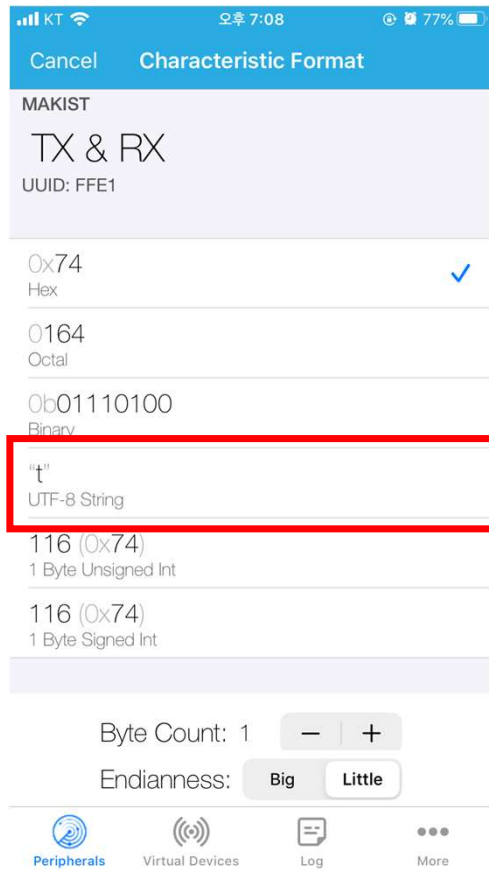
### 앱 사용법





## BLE앱을 사용하여 블루투스를 연결하고 아두이노와 채팅하기

### 앱 사용법

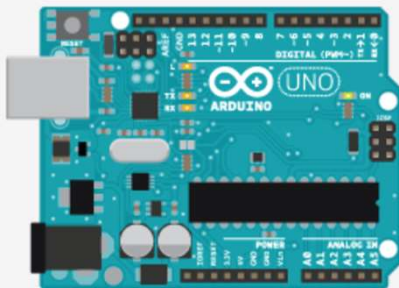


# Chapter 15

## WiFi 모듈 제어해보기



WHAT IS ARDUINO?



Step 1. AT명령어를 사용하여 wifi 모듈 제어하기

Step 2. WiFi 모듈을 공유기에 연결하기  
(STA모드)

Step 3. wifi모듈로 AP를 만들고 접속해보기  
(AP모드)

## AT명령어를 사용하여 WiFi 모듈 제어하기

### WIFI

- IEEE 802.11 통신규격을 만족하는 기기들끼리 무선으로 데이터를 주고받을 수 있도록 하는 기술을 뜻함
  - IEEE 802.11은 미국전기전자학회 (IEEE)에서 개발한 무선 랜 규격
- 와이파이를 사용하려면 무선 접속 장치(AP: Access point)가 있어야 함
- AP, STA
  - AP : Access point로 일종의 공유기 역할
    - 802.11 Wireless LAN 인터페이스와 802.3 Ethernet 인터페이스를 가지는 장비로, STA가 보낸 데이터를 무선으로 받아 Ethernet Port를 통해 유선망으로 보내 주는 장비이다
  - STA : AP에 접속해서 HTTP Request를 보내거나 간단한 HTTP서버가 될수 있다.
    - IEEE 802.11 용어이며, Wi-Fi 단말(Wi-Fi interface를 가진 단말)을 지칭합니다.

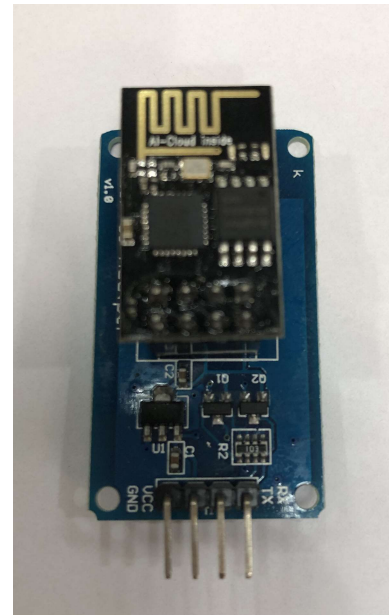
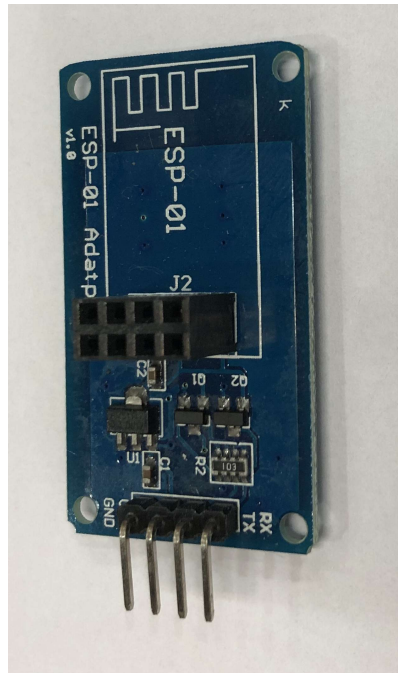
## AT명령어를 사용하여 WiFi 모듈 제어하기

### Esp8266

- 아두이노기반의 개발이 가능한 마이크로프로세서와 wifi모듈이 결합된 칩.
- IOT 사물인터넷, 스마트홈 등을 구성할때 필요한 와이파이 제품에서 가장 보편적인 모듈 중 하나.
- 시리얼을 통해 AT Command 명령으로 설정 변경 가능
- 802.11 b/g/n 지원

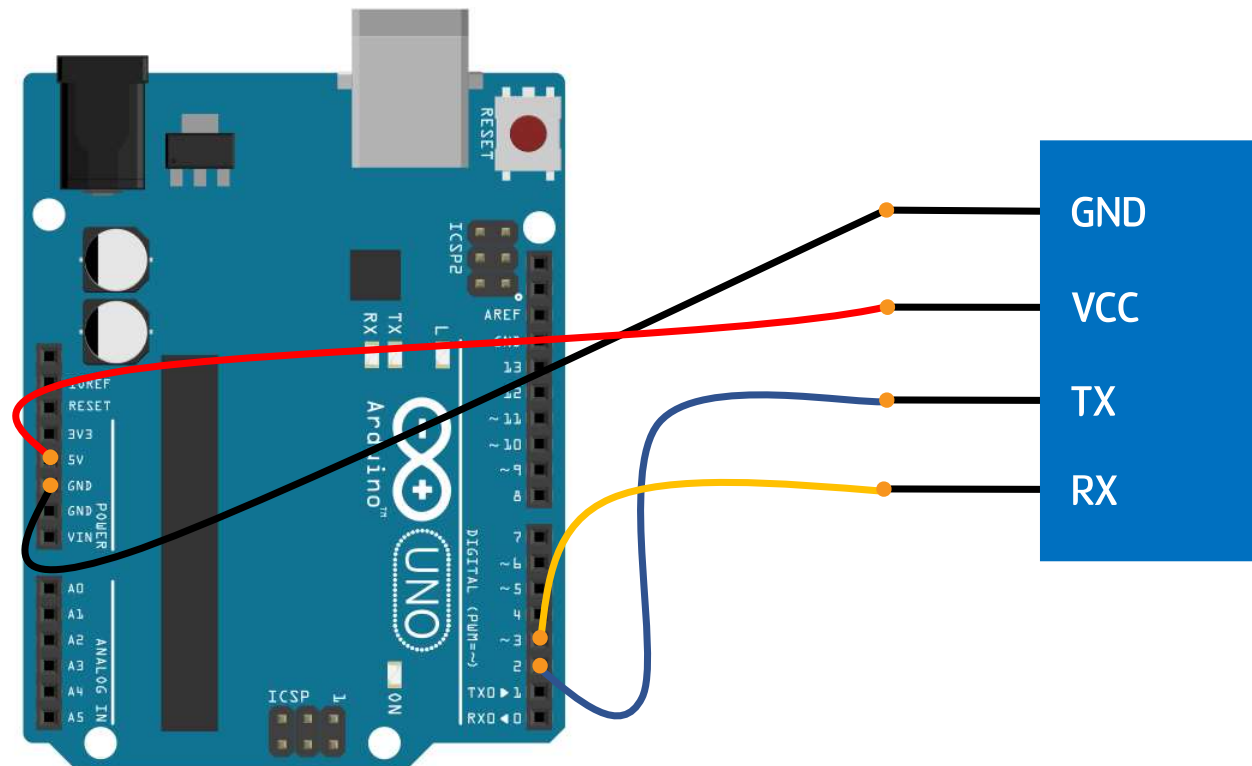
## AT명령어를 사용하여 WiFi 모듈 제어하기

### 회로구현



## AT명령어를 사용하여 WiFi 모듈 제어하기

## 회로구현



## AT명령어를 사용하여 WiFi 모듈 제어하기

### AT Command

- AT - 응답 : OK
  - ESP8266이 정상적으로 동작하고 있음을 나타내며, 정상적으로 동작되지 않을 경우 아무응답이 없습니다.
- AT+RST : 모듈 리셋 명령
- AT+GMR : 모듈 버전 확인 명령
- AT+CWMODE : WiFi 모드 확인 또는 설정 명령
  - AT+CWMODE? : 현재 ESP8266의 모드 확인 명령
  - AT+CWMODE=mode
    - 1 : Station Mode, WiFi 디바이스 기능
    - 2 : AP Mode, Access Point 기능
    - 3 : AP +Station Mode, 디바이스 + AP 기능
- [https://www.espressif.com/sites/default/files/documentation/4a-esp8266\\_at\\_instruction\\_set\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf)

## AT명령어를 사용하여 WiFi 모듈 제어하기

### WiFi 제어

```
#include <SoftwareSerial.h>

#define WIFI_RX    2
#define WIFI_TX    3

SoftwareSerial WIFI (WIFI_RX, WIFI_TX);

void setup() {
  Serial.begin(9600);
  WIFI.begin(115200);
}

void loop() {
  if (WIFI.available()) {
    Serial.write(WIFI.read());
  }
  if (Serial.available()) {
    WIFI.write(Serial.read());
  }
}
```

※ 명령어 : AT+UART\_DEF=9600,8,1,0,0 를 사용하여

Esp01의 통신속도를 9600으로 변경하고

WIFI.begin(115200); -> WIFI.begin(9600);

으로 바꾼후에 사용하시면 됩니다.

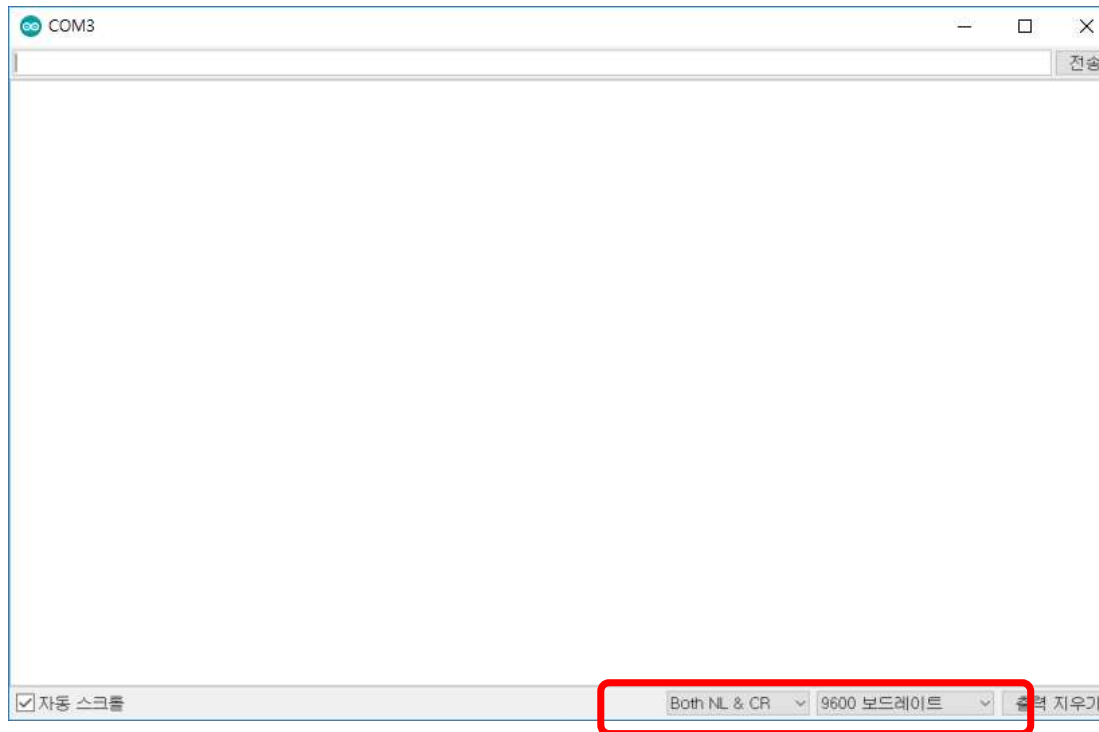
참고 : <http://blog.naver.com/makist2015/221417792688>



## AT명령어를 사용하여 WiFi 모듈 제어하기

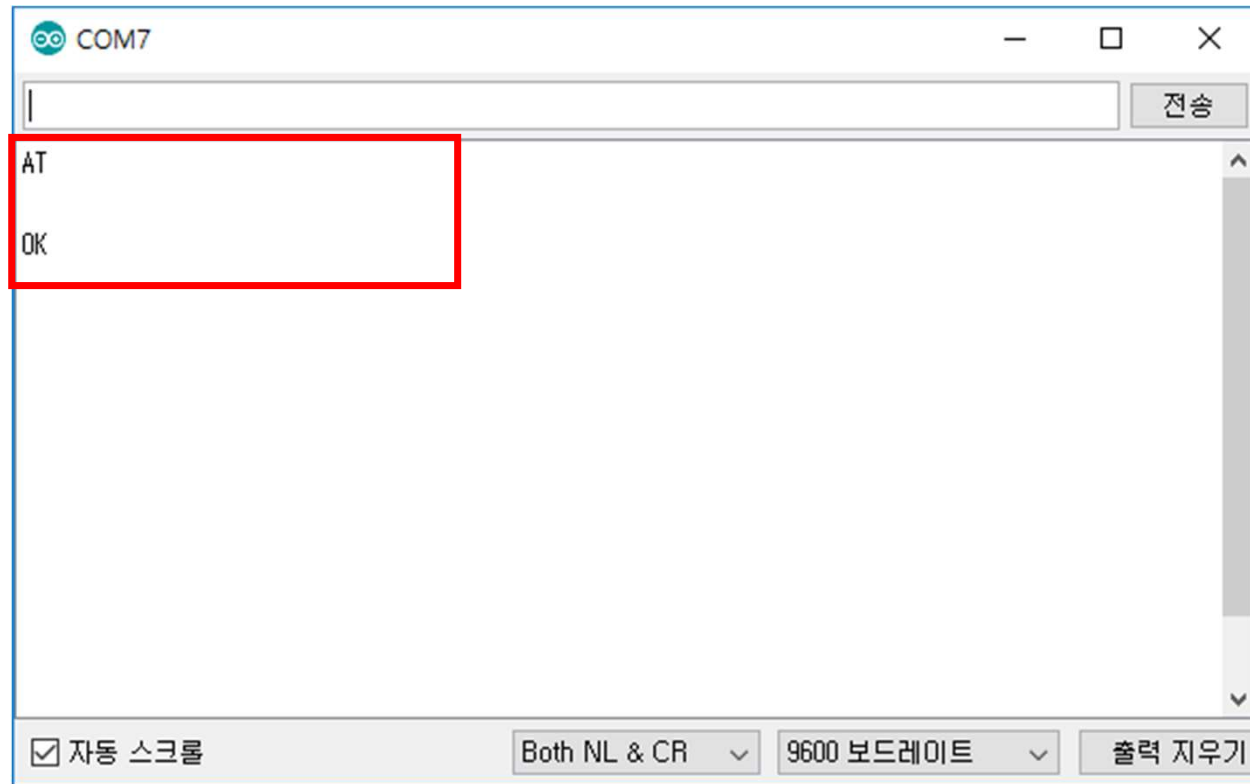
### WIFI 제어

- 시리얼 모니터에서 'Both NL & CR' 선택, '9600 보드레이트' 선택합니다.



## AT명령어를 사용하여 WiFi 모듈 제어하기

### WiFi 제어

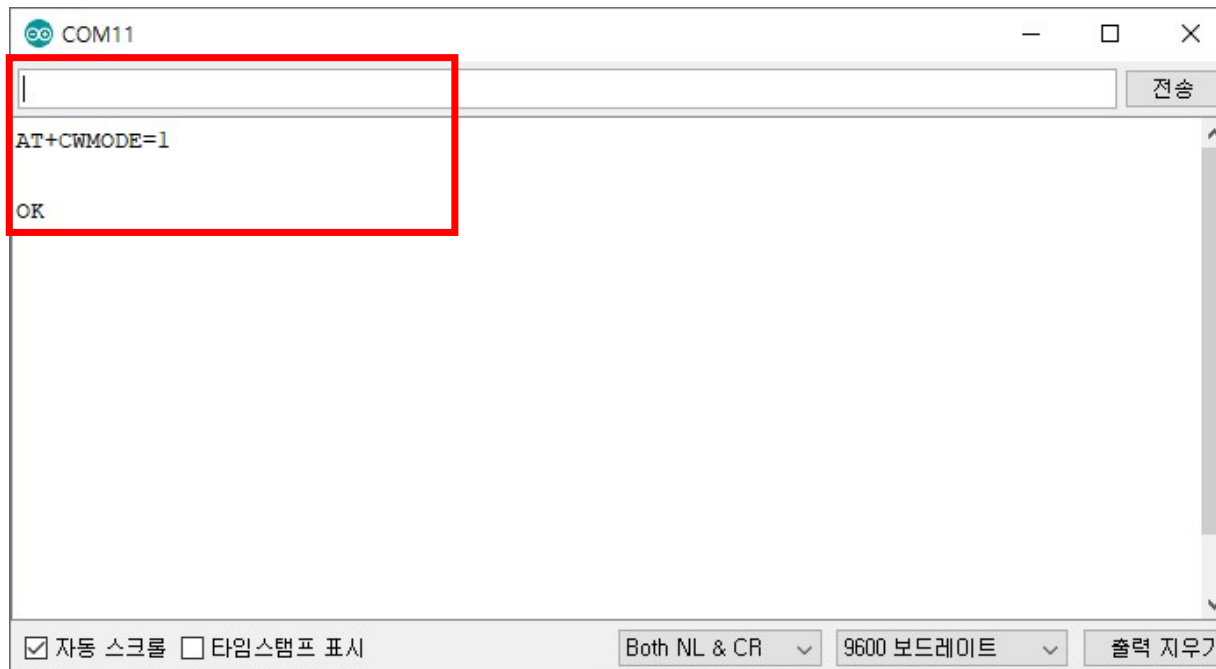


## WiFi 모듈을 공유기에 연결하기(STA모드)

### STA모드

- AT+CWMODE=1

OK 응답을 받게 되면 네트워크 연결에 대한 모드가 Station Mode로 설정이 됩니다.

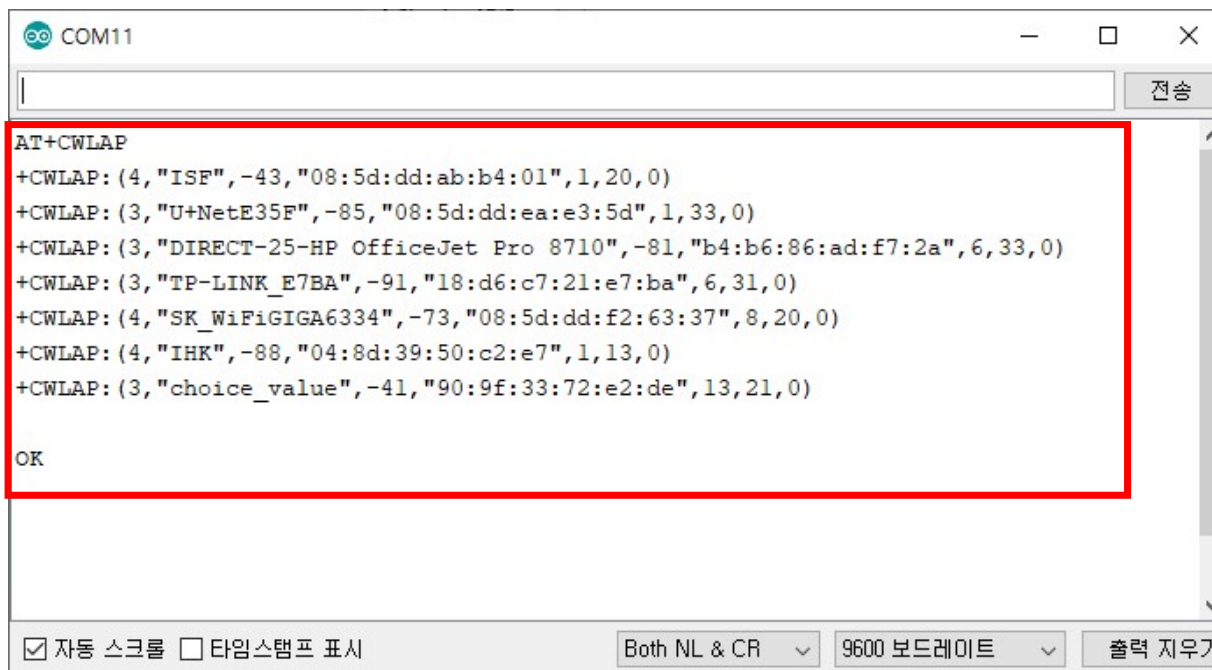


## WiFi 모듈을 공유기에 연결하기(STA모드)

### STA모드

- AT+CWLAP

연결할수 있는 WIFI 목록이 출력됨



```
COM11
AT+CWLAP
+CWLAP: (4, "ISF", -43, "08:5d:dd:ab:b4:01", 1, 20, 0)
+CWLAP: (3, "U+NetE35F", -85, "08:5d:dd:ea:e3:5d", 1, 33, 0)
+CWLAP: (3, "DIRECT-25-HP OfficeJet Pro 8710", -81, "b4:b6:86:ad:f7:2a", 6, 33, 0)
+CWLAP: (3, "TP-LINK_E7BA", -91, "18:d6:c7:21:e7:ba", 6, 31, 0)
+CWLAP: (4, "SK_WiFiGIGA6334", -73, "08:5d:dd:f2:63:37", 8, 20, 0)
+CWLAP: (4, "IHK", -88, "04:8d:39:50:c2:e7", 1, 13, 0)
+CWLAP: (3, "choice_value", -41, "90:9f:33:72:e2:de", 13, 21, 0)
OK
```

☒ 자동 스크롤 ☐ 타임스탬프 표시 Both NL & CR 9600 보드레이트 출력 지우기

## WiFi 모듈을 공유기에 연결하기(STA모드)

### STA모드

- AT+CWJAP="SSID","비밀번호"  
ESP01 모듈이 와이파이기에 연결. AT+CWJAP="SSID"," "

```
COM11
AT+CWJAP="ISF","a1234567890"
WIFI CONNECTED
WIFI GOT IP
OK
```

## WiFi 모듈을 공유기에 연결하기(STA모드)

### STA모드

- AT+CWQAP

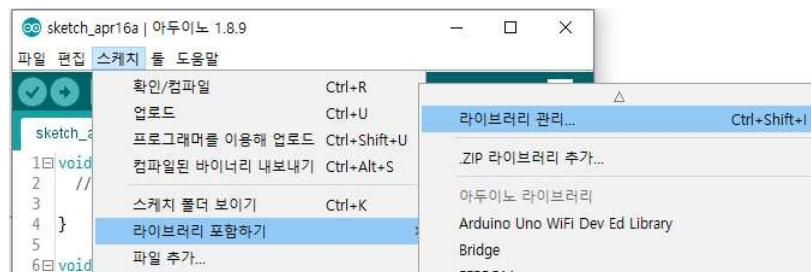
와이파이 연결해제

```
COM11
|
| 전송
|
AT+CWJAP="ISF","a1234567890"
WIFI CONNECTED
WIFI GOT IP
OK
AT+CWQAP
OK
WIFI DISCONNECT

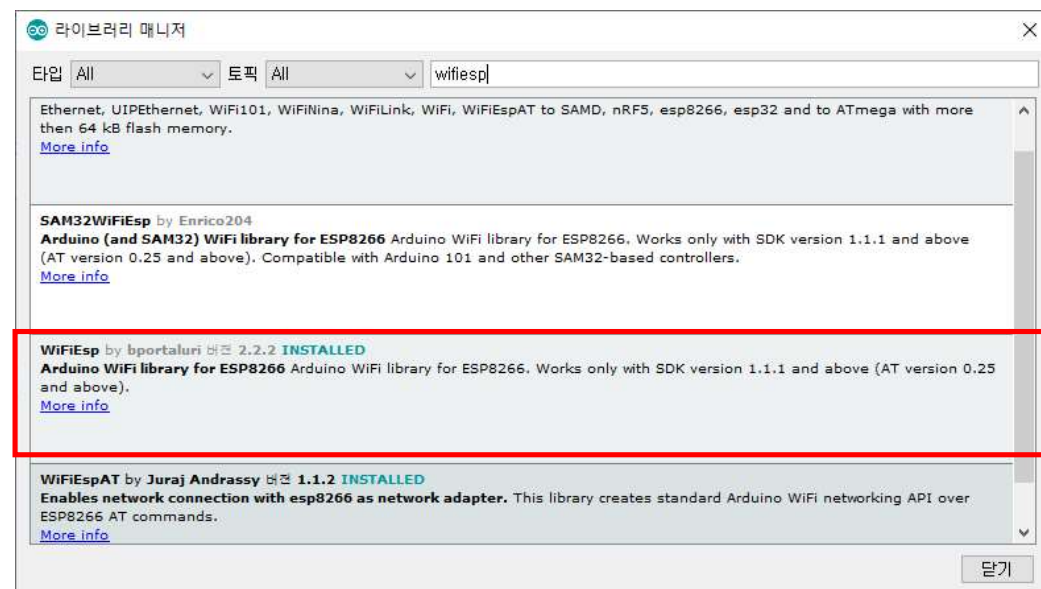
☒ 자동 스크롤 ☐ 타임스탬프 표시 Both NL & CR 9600 보드레이트 출력 지우기
```

## WiFi 모듈을 공유기에 연결하기(STA모드)

### 라이브러리 설치하기



스케치 - 라이브러리 포함하기 라이브러리 설치



Wifesp 를 검색하여 라이브러리를 설치해줍니다.

## WiFi 모듈을 공유기에 연결하기(STA모드)

### 소스코딩

```
#include "WiFiEsp.h"
#include "SoftwareSerial.h"

const char* wifi_ssid = "SSID"
const char* wifi_password = "PASSWORD";
unsigned int wifi_connect_timer;

SoftwareSerial WIFI(2,3); // RX, TX

void setup() {

  Serial.begin(9600);
  WIFI.begin(9600);
  WiFi.init(&WIFI);
```

```
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  WiFi.begin(wifi_ssid, wifi_password);
  wifi_connect_timer++;
  if (wifi_connect_timer >= 3) {
    WiFi.reset();
    wifi_connect_timer = 0;
    break;
  }
}

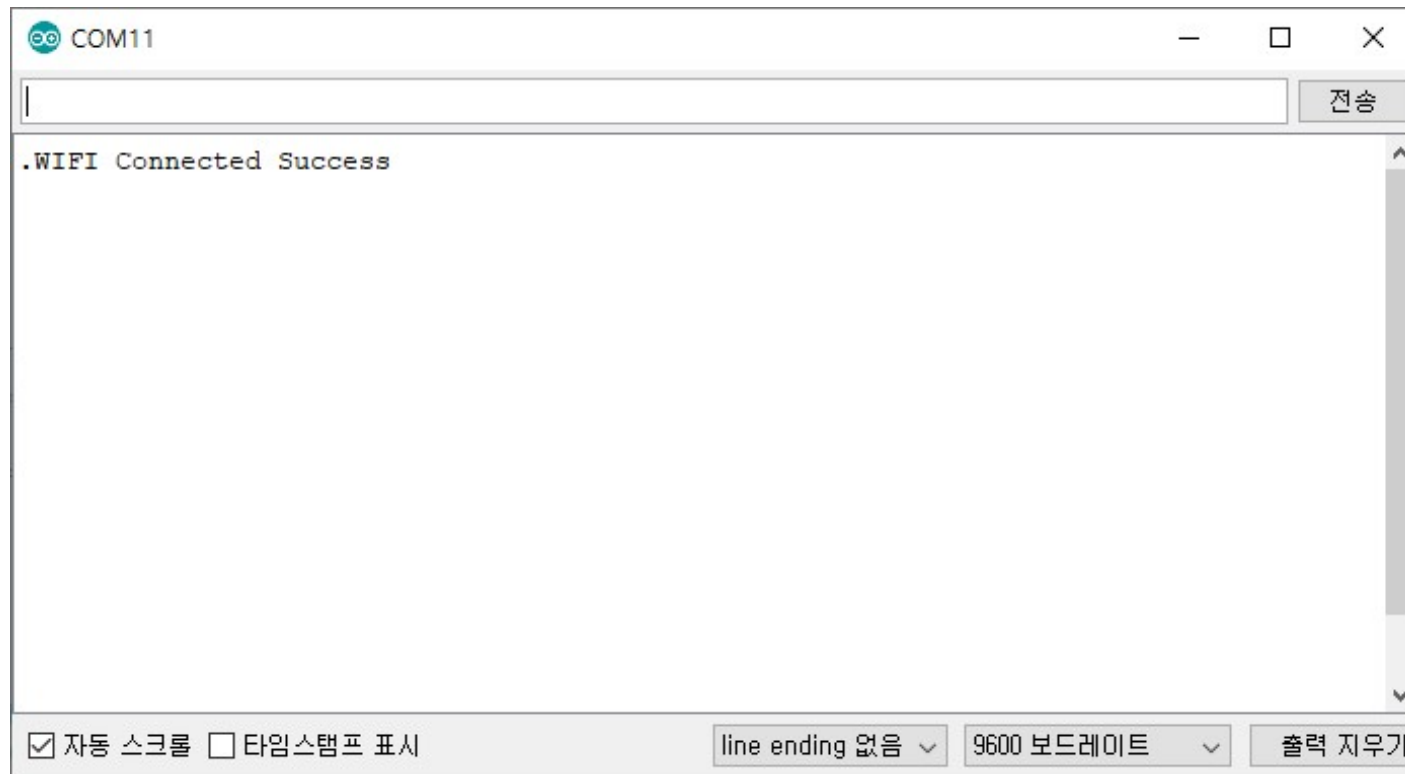
if (WiFi.status() == WL_CONNECTED) {
  Serial.println("WIFI Connected Success");
}
}

void loop() {
}
```



## WiFi 모듈을 공유기에 연결하기(STA모드)

### 실행결과



The screenshot shows a serial terminal window with the title bar 'COM11'. The main text area displays the message '.WIFI Connected Success'. The window includes a text input field at the top with a '전송' (Send) button. At the bottom, there are checkboxes for '자동 스크롤' (checked) and '타임스탬프 표시' (unchecked), along with dropdown menus for 'line ending 없음' and '9600 보드레이트', and a '출력 지우기' (Clear Output) button.

```
.WIFI Connected Success
```

## WiFi로 AP를 만들고 접속해보기(AP모드)

### 소스코딩

```
#include "WiFiEsp.h"
#include "SoftwareSerial.h"

SoftwareSerial WIFI(2,3); // RX, TX

char ssid[] = "MAKIST";
char pass[] = "12345678";
int status = WL_IDLE_STATUS;
int reqCount = 0;

WiFiEspServer server(80);
RingBuffer buf(8);

void setup()
{
  Serial.begin(9600);
  WIFI.begin(9600);
  WiFi.init(&WIFI);

  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while (true);
  }
```

```
if (WiFi.status() == WL_NO_SHIELD) {
  Serial.println("WiFi shield not present");
  while (true);
}

Serial.print("Attempting to start AP ");
Serial.println(ssid);

status = WiFi.beginAP(ssid, 10, pass, ENC_TYPE_WPA2_PSK);

Serial.println("Access point started");
printWifiStatus();

// start the web server on port 80
server.begin();
Serial.println("Server started");
}
```

## WiFi로 AP를 만들고 접속해보기(AP모드)

### 소스코딩

```
void loop()
{
  WiFiEspClient client = server.available();

  if (client) {
    Serial.println("New client");
    buf.init();
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        buf.push(c);

        if (buf.endsWith("\r\n\r\n")) {
          WiFiResponse(client);
          break;
        }
      }
    }
    delay(10);

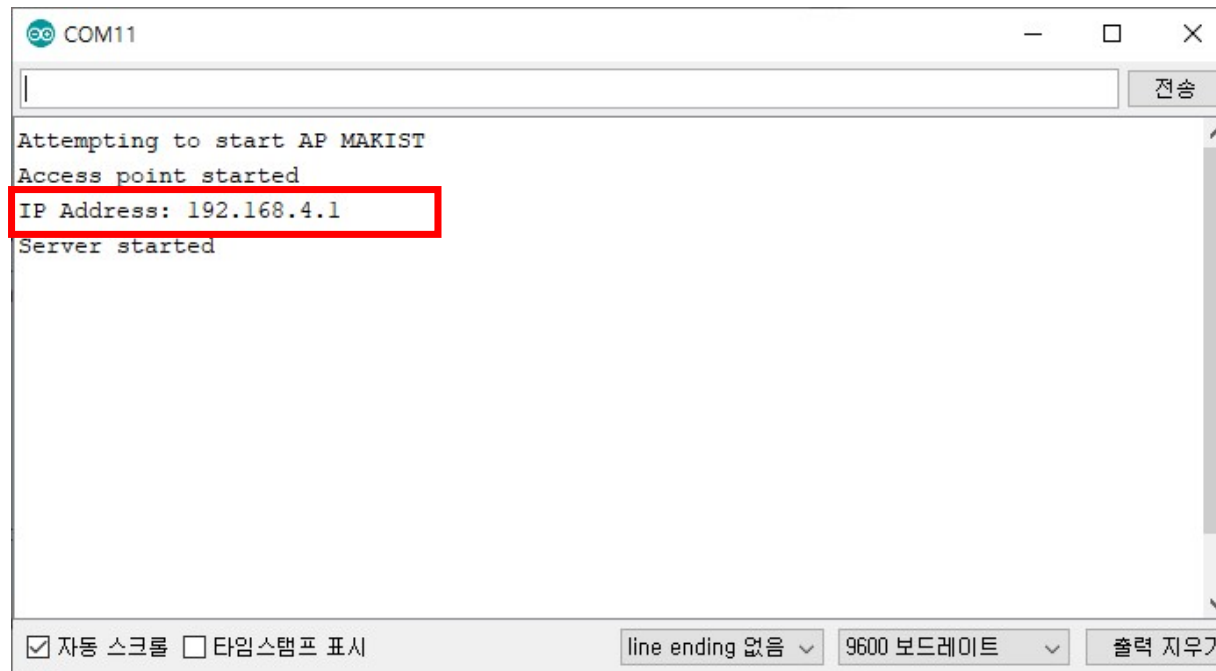
    client.stop();
    Serial.println("Client disconnected");
  }
}
```

```
void WiFiResponse(WiFiEspClient client)
{
  client.print(
    "HTTP/1.1 200 OK\r\n"
    "Content-Type: text/html\r\n"
    "Connection: close\r\n"
    "\r\n");
  client.print("<!DOCTYPE HTML>\r\n");
  client.print("<html>\r\n");
  client.print("<h1>Hello World!</h1>\r\n");
  client.print("</html>\r\n");
}

void printWifiStatus()
{
  // print your WiFi shield's IP address
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
}
```

## WiFi로 AP를 만들고 접속해보기(AP모드)

### 실행결과



```
COM11
Attempting to start AP MAKIST
Access point started
IP Address: 192.168.4.1
Server started
```

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

## WiFi로 AP를 만들고 접속해보기(AP모드)

### 실행결과

