
Arduino Uno 보드를 이용한 JSN270 모듈 실험

JMPSYS JSN270



문의
제이엠펙시스
<http://jmpsys.com>

목 차

서문

1. 아두이노

1.1 아두이노 보드

1.2 아두이노 시작하기

1.3 아두이노 프로그래밍 언어 개요

2. 무선랜 표준 및 채널

2.1 Infra-structure

2.2 Access Point, Wireless client

2.3 IEEE 802.11

2.4 2.4GHz와 5GHz 무선랜 채널 및 채널 대역폭

2.5 SISO, MIMO

2.6 Tera term 프로그램

3. 무선랜 암호화 표준

3.1 WEP와 위험성

3.2 WPA와 WPA2 personal

3.3 WPS-PBC

4. IP주소, 서브넷 마스크, 게이트웨이, DNS

4.1 자동 IP (DHCP)와 고정 IP

4.2 IP 주소와 서브넷 마스크 관계

4.3 게이트웨이 기능

4.4 DNS 기능

5. JSN270 예제 실습

5.1 AT 명령어

5.2 TCP Client WPA2 실습

5.3 TCP Server Web 실습

5.4 UDP Client No_Encryption 실습

5.5 UDP Server WPA 실습

5.6 UDP NTP Client 실습

5.7 Simple Web Server 실습

Arduino Uno 보드를 이용한 JSN270 모듈 실험

JMPSYS JSN270

서 문

아두이노 보드는 가격이 저렴하고 관련 지식과 정보를 인터넷에서 쉽게 얻을 수 있어서 마이크로컨트롤러를 기반으로 하는 프로젝트를 구현하는데 매우 유용하다. 또한 전기전자에 대한 지식이 없는 초보자도 쉽게 사용할 수 있어서 인터랙티브 아트에서부터 공학적 해결책까지 널리 활용되고 있다.

아두이노는 보다 쉽게 물리적인 세계를 감지하고 제어할 수 있게 해주는 도구로 간단한 마이크로컨트롤러 보드에 기초한 오픈소스 컴퓨팅 플랫폼으로 이보드에서 실행할 프로그램을 작성하는 개발 환경을 포함한다. 아두이노는 다양한 스위치나 센서를 입력을 받아 조명, 모터 등의 출력을 제어하는데 사용된다. 아두이노 스케치는 독립적으로 동작하거나 PC에서 동작하는 소프트웨어(예:Flash, Processing, MaxMSP)와 연동하여 동작할 수 있다.

다수의 마이크로컨트롤러와 플랫폼(Parallax Basic Stamp, Netmedia BX-24, Phidgets, MIT Handyboard) 등이 존재하지만 아두이노는 Atmel의 ATmega8과 ATmega168 마이크로컨트롤러를 기반으로 단순하고 명확한 프로그래밍 환경을 갖추

고 있어 초보자도 쉽게 사용할 수 있는 장점이 있다.

아두이노 소프트웨어는 윈도, 맥킨토시 OSX와 리눅스 운영체제에서도 동작하는 크로스 플랫폼이며 오픈 소스 도구로 출시되어 고급 프로그래머에 의해 확장이 가능하다. 프로그래밍 언어는 C++ 라이브러리를 통해 확장될 수 있어 보다 고급 기술을 이해하고자 하는 사람은 AVR C 프로그래밍 언어로 옮겨갈 수 있다. 마찬가지로 AVR-C 코드를 직접 아두이노 스케치에 추가할 수도 있다.

1. 아두이노

1.1 아두이노 보드



아두이노 우노(Arduino Uno)는 ATmega328에 기초한 마이크로컨트롤러 보드이다. 이 보드는 14개의 디지털 입출력 핀(이 가운데 6개는 PWM 출력으로 사용가능) 6개의 아날로그 입력, 16MHz 크리스탈 오실레이터, USB 연결, 전원 잭, ICSP 헤더와 리셋 버튼을 가진다. 이 우노 보드는 기존 보드와 달리 USB를 직렬로 변환하는FTDI 드라이버 칩을 사용하지 않고 대신 ATmega8U2에서 이 변환을 프로그램으로 수행한다.

● 아두이노 우노 보드 특징

마이크로 컨트롤러 : ATmega328

구동전압: 5V

입력전압(권고): 7~12V

디지털 입출력 핀 : 14(이 중 6개는 PWM 출력)

아날로그 입력 핀 : 6

입출력 핀의 DC 전류 : 40mA

3.3V 핀의 DC 전류 : 50mA

플래시 메모리 : 32KB(ATmega328), 이 중 부트로더가 0.5KB사용

SRAM: 2KB(ATmega328)

EEPROM : 1KB(ATMega328)

클록 속도 : 16MHz

- ① 전원 : 이 보드는 USB 연결이나 외부 전원 장치를 통해 전기를 공급받으며 전원은 자동으로 선택된다. 외부 전원은 어댑터나 배터리에서 공급된다. 어댑터는 2.1mm 중앙이 양극인 플러그를 보드의 전원 잭에 끼운다. 배터리 연결선은 전원핀에 연결한다. 보드는 6~20V의 외부 전원에서 동작하며 7V이하의 경우 5V핀은 5V이하 전압을 공급해 보드가 불안정해질 수 있다. 12V 이상이면 전압 조정기가 과열되어 보드에 손상이 갈 수 있다. 따라서 전원 전압은 7~12V가 좋다.

● 전원 핀

VIN : 외부 전원 사용 시 아두이노 보드의 입력 전압

5V : 보드의 마이크로컨트롤러와 다른 구성요소에 공급할 전원. 이 전원은 VIN전압이 보드의 전압조정기를 통해 공급되거나 USB나 다른 5V 전압으로부터 공급된다.

3.3V : 보드의 전압 조정기에 의해 공급되며 최대 전류는 50mA이다.

GND : 접지핀

- ② 메모리 : Arduino Uno은 32KB의 메모리를 가지며 이 중 0.5KB를 부트로더가 사용한다. 2KB의 SRAM과 1KB의 EEPROM을 가진다.

- ③ 입출력 : Arduino Uno 보드는 14개의 디지털 핀이 입력과 출력으로 사용할 수 있다. 입출력 지정을 위해 pinMode()를, 출력을 위해 digitalWrite()와 입력을 위해 digitalRead() 함수를 사용한다. 이 핀은 5V에서 동작한다. 각 핀은 최대 40mA의 전류를 입출력할 수 있고 내부에 20~50kΩ의 풀업저항을 가진다.

● 특별한 기능을 수행하는 일부 핀

직렬통신 0와 1번 핀 : TTL 직렬 통신 데이터를 수신하고 송신하는데 사용, 이들 핀은 Atmega8U2의 USB를 TTL로 변환하는 직렬 칩에 연결되어있다.

외부 인터럽트 2와 3번 핀 : 이 핀은 전압상승 에지나 전압하강 에지에서 또는 전압이 변화 시에 인터럽트가 개시되도록 하는데 사용된다. 세부 내용은 attachInterrupt() 함수를 참조한다.

PWM 3,5,6,9,10번핀 및 11번 핀: analogWrite() 함수와 함께 8비트 PWM 출력을 제공한다.

SPI 통신 10(SS), 11(MOSI), 12(MISO) 및 13(SCK)번 핀 : 이들 핀은 SPI 라이브러리를 사용하여 SPI 통신을 지원한다.

LED 13번핀 : 디지털 13번 핀에 내장된 LED가 연결되어 있다. 이 핀이 하이가 되면 LED가 온(점등), 핀이 로우가 되면 LED가 오프된다.

아날로그 핀 : A0~A5까지 6개의 아날로그 핀을 가지며 각각은 10비트의 해상도를 가진다. 디폴트로 접지에서 5V까지 측정할 수 있다. 측정 상한값을 변경하려면 AREF 핀과 analogReference() 함수를 사용한다.

I2C 통신 A4(SDA)와 A5(SCL)번 핀 : “Wire” 라이브러리를 사용하여 I2C(TWI)통신을 지원한다.

AREF핀: 아날로그 입력에 대한 기준 전압(0~5V), analogReference()함수와 함께 사용할 수 있다.

Reset핀 : 마이크로컨트롤러를 리셋하려면 이 핀을 LOW에 연결, 리셋 버튼을 추가하려면 이 핀을 사용한다.

- ④ 통신 : 아두이노 우노는 컴퓨터 및 다른 아두이노나 마이크로 컨트롤러와 통신할 수 있는 다양한 수단을 제공한다. ATmega328은 UART TTL(5V) 직렬 통신을 제공한다(디지털 0 (RX))과 1((TX)번핀). 보드 상의 ATmega8U2는 USB를 통해 직렬 통신이 가능하게 하고 컴퓨터의 소프트웨어에 가상의 COM 포트로 나타나게 한다. 8U2 펌웨어는 표준의 USB COM 드라이버를 사용하므로 외부 드라이버가 필요 없다. 윈도 경우에는 .inf 파일이 필요하다. 아두이노 소프트웨어는 단순한 텍스트 데이터를 아두이노 보드와 컴퓨터 간에 전송하는 직렬 모니터를 포함하고 있다. 데이터가 전송될 때 보드의 RX와 TX LED가 깜박인다. “SoftwareSerial” 라이브러리는 우노의 디지털 핀에서 직렬 통신을 지원한다. ATmega328은 I2C(TWI)와 SPI 통신도 지원한다. 아두이노 소프트웨어는 “Wire” 라이브러리를 이용해서 I2C 버스를 쉽게 사용할 수 있다. SPI 통신의 경우에는 “SPI” 라이브러리를 사용한다.

- ⑤ 프로그래밍 : 아두이노 우노는 아두이노 개발환경 소프트웨어(무료 다운로드)로 프로그램 할 수 있다. 이 소프트웨어 메뉴 Tools>Board>Arduino Uno(보드의 마이크로컨트롤러에 따라)를 선택한다. 아두이노 우노에서 ATmega328은 부트 로더를 먼저 쓰고 새로운 코드를 업로드할 수 있게 해준다. 부트 로더

를 생략하고 마이크로컨트롤러를 프로그램하려면 ICSP(In-Circuit Serial programming) 헤더를 사용한다.

- ⑥ 자동 소프트웨어 리셋 : 업로드 전에 물리적으로 리셋 버튼을 누르는 대신에 아두이노 우노는 보드에 연결된 컴퓨터에서 동작하는 소프트웨어에 의해 리셋 되도록 설계되어 있다. ATmega8U2의 하드웨어 흐름 제어선(DTR) 가운데 하나가 100nF 축전기를 통해 ATmega328의 리셋 선에 연결되어 있다. 이 선이 LOW가 되면 리셋 선은 칩 리셋에 필요한 시간 동안 LOW 상태를 유지한다. 아두이노 개발환경에서 업로드 버튼을 클릭하면 이 기능을 이용하여 코드 업로드가 실행된다.
- ⑦ USB 과전류 보호 : 아두이노 우노는 컴퓨터의 USB 포트가 단락 및 과전류로부터 보호되도록 리셋이 가능한 폴리 퓨즈를 갖고 있다. 대부분의 컴퓨터가 자체적으로 내부 보호를 제공하지만 이 퓨즈는 500mA 이상의 전류가 USB 포트에 인가되면 단락이나 과부하가 제거될 때까지 자동으로 연결을 끊는다.

1.2 아두이노 시작하기

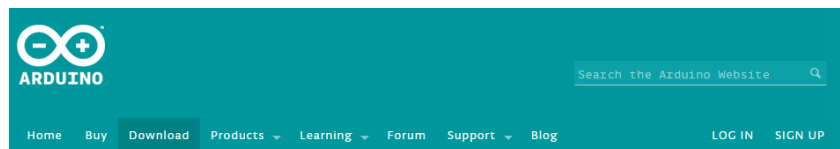
아두이노 개발환경(IDE: Integrated Development Environment)은 코드를 작성하는 텍스트 편집기, 메시지 영역, 텍스트 콘솔, 공통 기능을 위한 버튼을 갖는 툴바와 일련의 메뉴를 포함한다. 개발환경은 아두이노 하드웨어에 연결해서 프로그램을 업로드하고 통신할 수 있게 한다.

①아두이노 보드와 USB 케이블 연결

Arduino Uno 보드와 USB 케이블을 연결한다. Arduino Duemilanove, Nano 또는 Diecimila 보드의 경우도 같다.

②아두이노 개발환경 다운로드

다운로드 페이지에서 최신 버전을 받는다(다운로드사이트). 다운로드가 완료되면 폴더 구조를 유지하도록 해당 파일을 unzip한다.



Download the Arduino Software



③ 보드 연결

Arduino Uno, Mega, Duemilanove 와 Arduino Nano 보드는 자동으로 컴퓨터나 외부 전원으로부터 USB 연결을 통해 전원을 공급받는다. Arduino Diecimila 보드를 사용하는 경우에는 USB 연결로부터 전원을 공급받도록 구성해야 한다. USB와 전원 잭 사이의 3핀 가운데 2개(USB에 가장 가까운 두 핀)에 맞는 작은 플라스틱인 점퍼로 전원을 선택한다. USB 케이블로 아두이노 보드를 컴퓨터에 연결한다. 녹색 전원 LED(PWR로 이름 붙여진)가 켜진다.

④ 드라이버 설치

Windows 7,8, Vista 또는 XP 운영체제의 Arduino Uno용 드라이버를 설치한다.

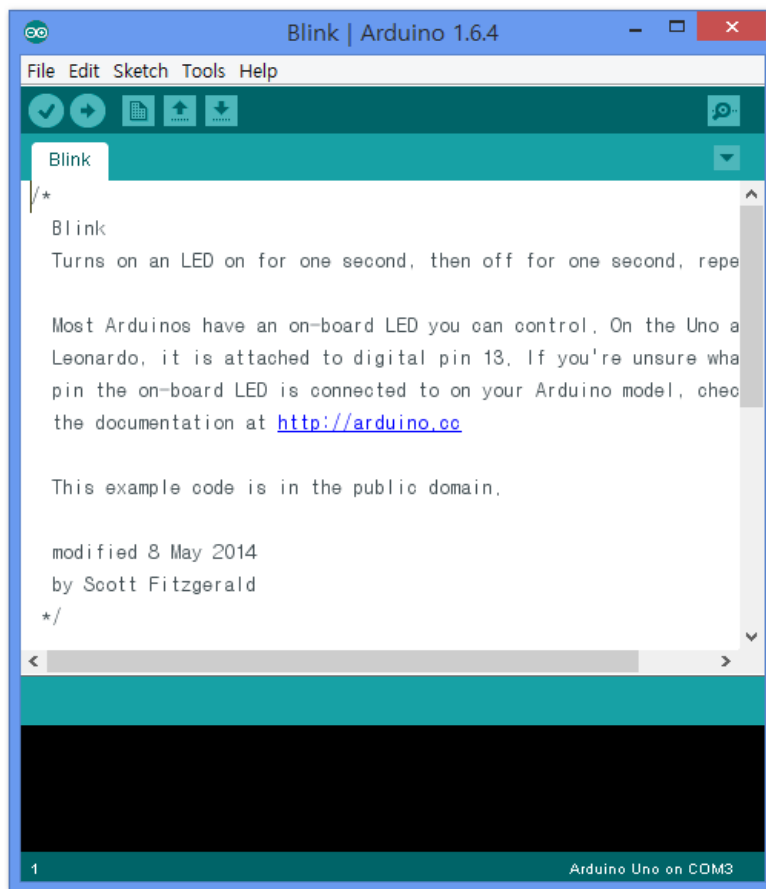
⑤ 아두이노 애플리케이션 시작

아두이노 애플리케이션(Arduino.exe)을 더블 클릭한다.

⑥ 깜빡임(Blink) 예제를 연다.

아래의 그림과 같이 예제(LED 깜빡임) 스케치를 연다.

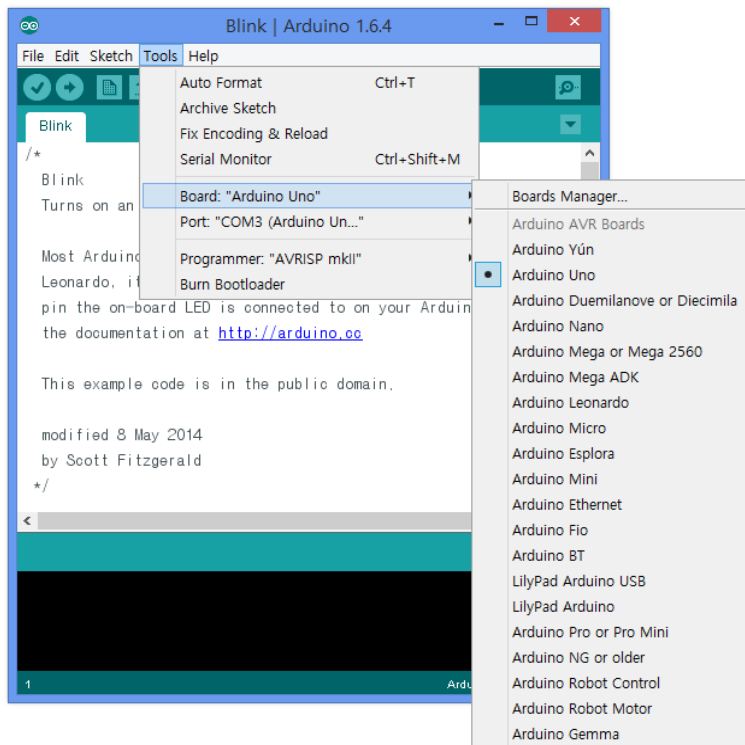
(File - Examples - 1.Basics - Blink)



⑦ 사용하고 있는 보드 선택

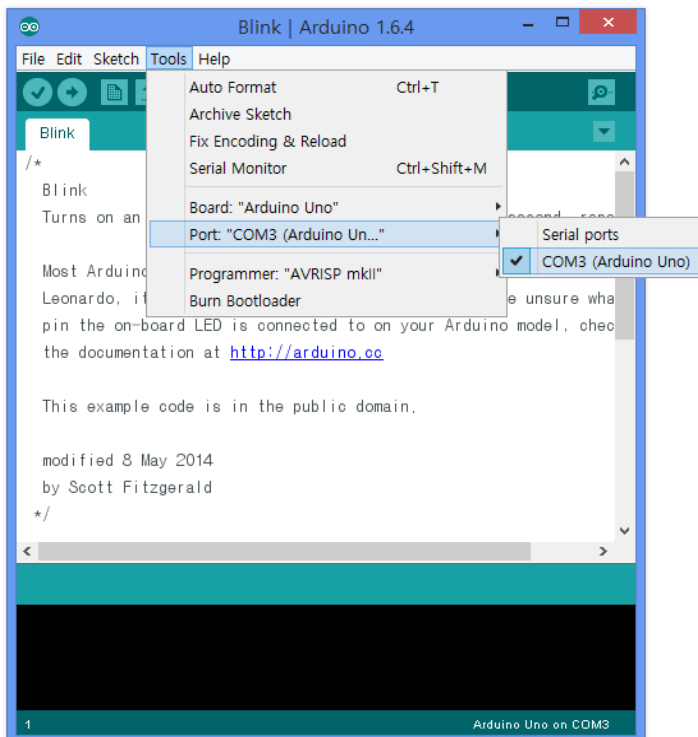
그림과 같이 Tool - Board 메뉴에서 사용하고 있는 아두이노 보드를 선택한다.

본 교재에서는 Arduino Uno 선택.



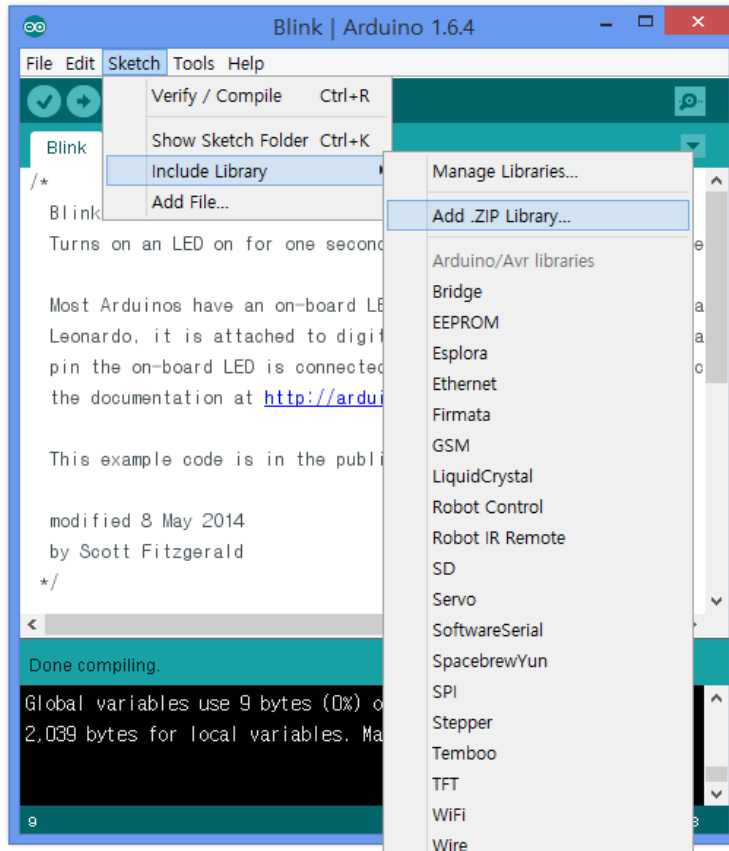
⑧ 직렬포트(Serial port)선택

Tools - Serial Port 메뉴에서 아두이노 보드의 직렬 디바이스 선택. USB 연결의 경우 대개 COM3이나 그 이상.



⑨ 라이브러리 추가

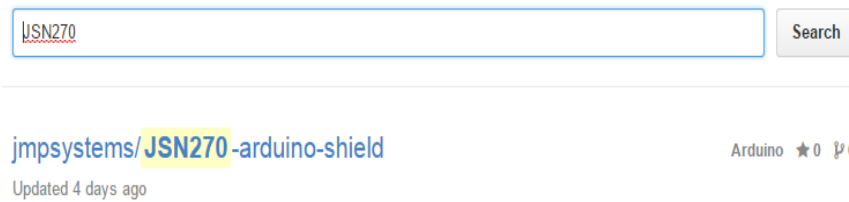
Sketch – Include Library – Add.ZIP Library 에서 JSN270폴더를 추가해준다.



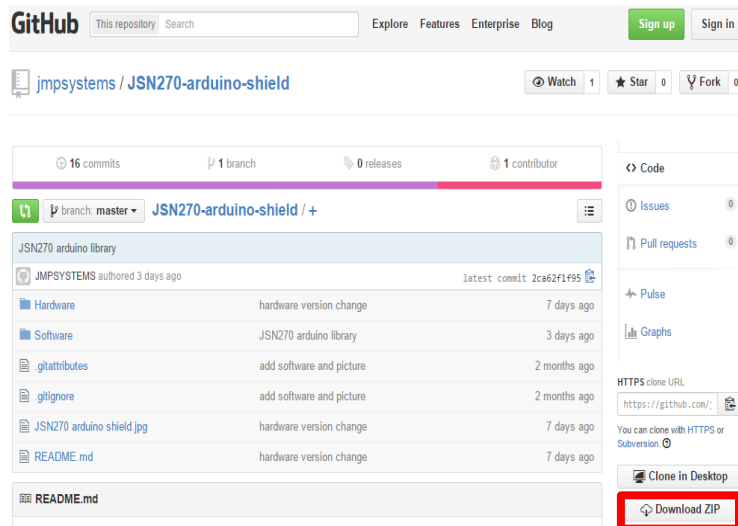
JSN270 자료 다운

① <http://github.com> 에 접속한다.



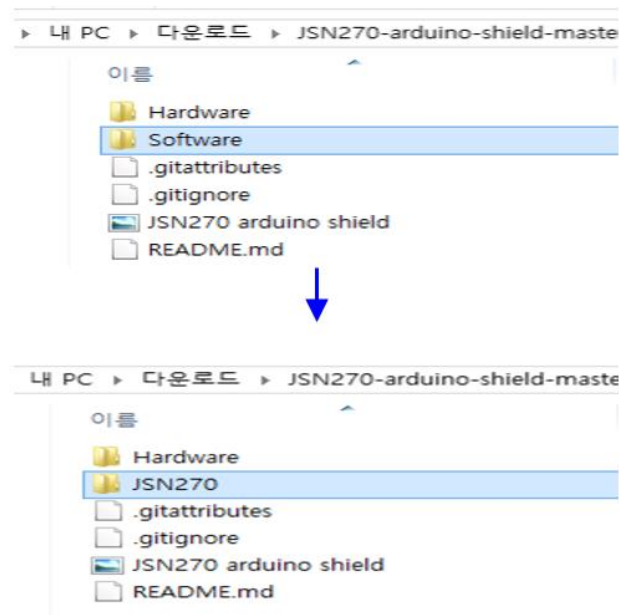


② 오른쪽 하단의 압축파일 다운 클릭

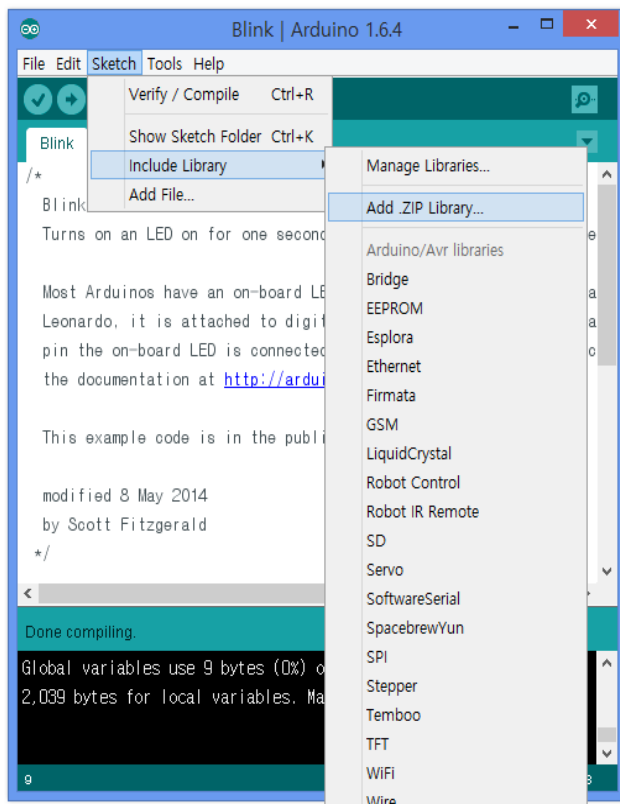


③ 다운받은 zip파일 unzip한다.

④ 폴더 내의 Software 파일명을 JSN270으로 변경해준다.

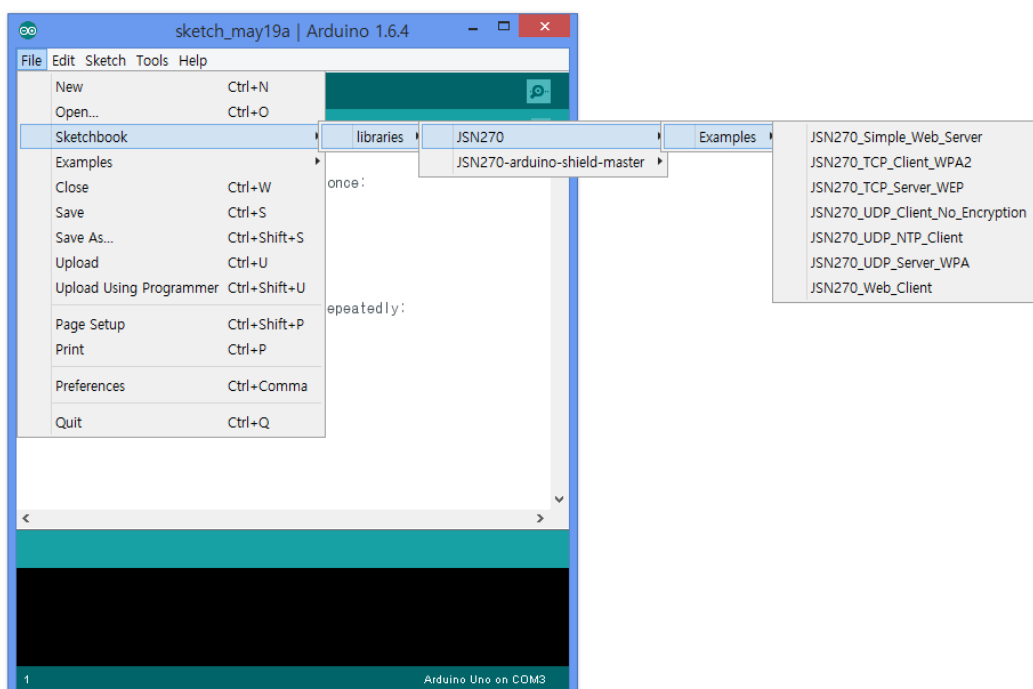


⑤ 앞쪽 참고 라이브러리 추가 : Sketch - Include Library - Add.ZIP Library 에서 JSN270폴더를 추가해준다.

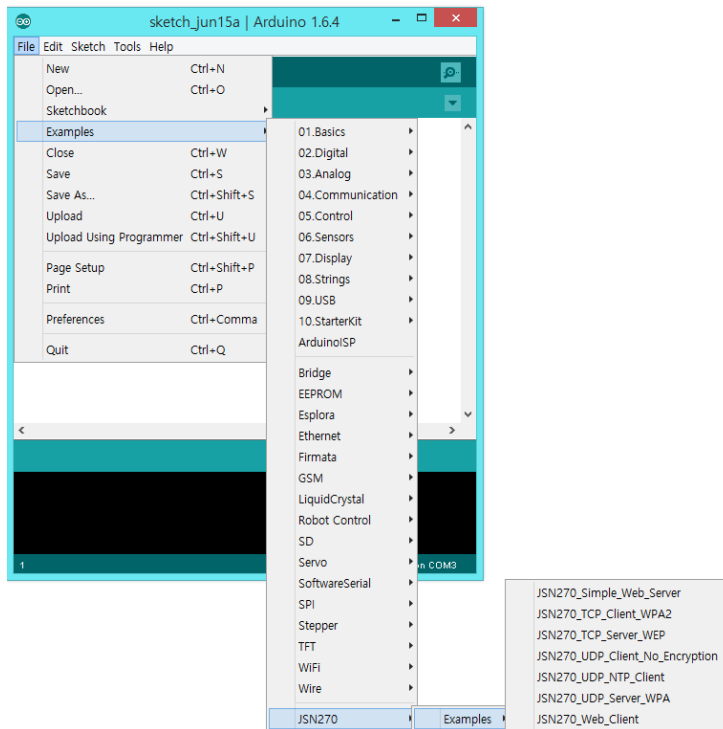


자세한건 <http://arduino.cc/en/Guide/Libraries> 참고

⑥ github에서 다운받은 라이브러리를 실행한다.



File – sketchbook – libraries – JSN270 – Examples



또는 File – Examples – JSN270으로 들어가서 예제를 선택한다.

1.3 아두이노 프로그래밍 언어 개요

아두이노 언어는 C/C++에 기초해서 모든 표준 C 구성과 C++특징을 지원한다. 아두이노 프로그램은 3개의 주요부분(구조, 변수와 상수, 함수)로 이루어진다. C 언어에 대한 기초 지식이 있으면 아두이노 프로그래밍 언어를 바로 사용할 수 있다.

* 아두이노 언어 소개 사이트 : <http://arduino.cc/en/Reference/HomePage>

1) 프로그래밍 구조

아두이노는 두 부분으로 동작한다. (setup() 및 loop()함수)

- Void setup(){} : 이 함수는 아두이노 프로그램이 시작될 때 맨 처음 한 번 실행되기 때문에 프로그램 동작에 필요한 설정 관련 코드를 포함한다. 이 함수는 프로그램의 맨 위에 만들어지며 핀 모드를 설정하고 통신을 초기화 하는 등의 작업을 수행한다.
- Void loop() {} : 이 함수는 setup() 함수 종료 후 실행되며 전원이 공급되는 한 반복해서 실행된다.

2) 일반규칙

- 한 줄 주석처리에 //을 사용한다. //이후 이 줄 끝까지의 문장은 코드 설명을 위한 주석으로 간주, 실행되지 않는다.
- 한 줄 이상의 문장을 주석 처리 할 때에는 /* */를 사용한다. /* ~ */ 사이의 문장은 주석으로 간주, 실행되지 않는다.
- 중괄호 {}는 블록 단위 코드의 시작과 끝을 지정하는데 사용한다.
- 명령문의 끝에는 ;(세미콜론)을 반드시 붙여야 한다.

3) 제어 구문

프로그램의 흐름을 제어하는 구조에는 아래와 같은 구문을 사용한다.

- 조건 제어문
 - If, if...else
 - Switch ~ case
- 반복 제어문
 - For

- While
- Do..while
- Break
- Continue

4) 연산자

산술 연산자 : +, -, *, /, %

비교 연산자 : ==, !=, <, >, <=, >=

부울연산자 : &&, ||, !

포인터 액세스 연산자: *, &

비트 단위 연산자 : &, |, ^, ~, <<, >>

포트 조작 : 포트 레지스터는 아두이노 보드의 마이크로컨트롤러의 입출력 핀을 저수준에서 빠르게 조작할 수 있도록 함.

아두이노 보드에 사용된 칩은 3개의 포트 즉, B(디지털핀8~13), C(아날로그 입력 핀), D(디지털 핀0~7)을 가진다.

복합 연산자 : ++, --, +=, -=, *=, /=, &=, |=

5) 변수와 상수

자료형	크기	범위
Boolean	1	0 또는 1 (참 또는 거짓)
Byte	1	0 ~ 255
Char	1	-128 ~ 127
Unsigned char	1	0 ~ 255
Int	2	-32,435 ~ 32,767
Unsigned int	2	0 ~ 65,535
Word	2	0 ~ 65,535
Long	4	-2,147,483,648 ~ 2,147,483,647
Unsigned long	4	0 ~ 4,294,967,295
Float	4	-3.4028235E+38 ~ 3.4028235E+38
Double	4	-3.4028235E+38 ~ 3.4028235E+38
String		문자 배열
array		변수 배열

2. 무선 표준 채널

2.1 Infra-structure

[Infra – structure]

infrastructure ; 인프라

정보기술이나 인터넷에서, 인프라는 컴퓨터와 사용자들을 연결하는데 사용되는 물리적인 하드웨어를 말한다. 인프라는 전화회선, 케이블 TV 회선, 인공위성 및 안테나 등과 같은 전송매체와, 라우터, 리피터 등의 전송제어장치 등을 포함한다. 인프라는 또한, 신호를 보내고 받고, 관리하는데 사용되는 소프트웨어를 포함한다.

경우에 따라서는, 인프라는 컴퓨터나 기타 다른 장치에 연결되어 있는 것은 아니지만, 서로 연결되어 있는 하드웨어와 소프트웨어를 지칭하기도 한다. 그러므로, 정보기술 사용자들에게, 인프라는 정보의 흐름과 처리를 지원하는 모든 것처럼 비쳐지기도 한다.

인프라 회사들은, 상호 연결되어 접근할 수 있는 형태 그리고 얼마나 많은 정보를 얼마나 빠르게 보낼 수 있는냐는 두 가지 측면 모두에서 인터넷의 발전에 지대한 역할을 한다.

Infrastructure 모드는 Access Point라는 중계기를 통해 무선랜 카드와 데이터 통신을 한다.

Ap를 시야가 트인 곳에 설치하면, 데이터를 송수신하는 단말(PC, 프린터 등) 주변에 장애물이 있더라도 안정된 데이터 통신이 가능하다.

AP를 사용하면 다음과 같은 장점이 있다.

- HUB를 통해 유선 LAN과의 상호 접속이 가능하다.
- RG(Residential Gateway)를 사용하면 인터넷 접속이 가능하다.
- 여러 대의 AP를 일정한 간격으로 설치하면, AP간의 이동 중에도 네트워크 접속이 가능하여 로밍 기능을 이용할 수 있다.
- MAC Address를 제한하여 보안 기능을 사용할 수 있다.

2.2 Access Point, Wireless client

Access Point 공유기

Wireless Access Point라고 하여 WAP라고도 한다.

컴퓨터에 달려있는 무선랜 카드와 연결해 데이터를 전송하는 안테나 겸 접속장치로 "공유기"로 불린다. 무선 LAN에서 기지국 역할 하는 소출력 무선 기기이다.

유선과 무선을 잇는 브릿지 역할을 하게 되며 유선망 관점에서 AP는 라우터 또는 스위치 등에 붙게 되며 라우터나 스위치는 무선 스테이션을 인식하지 못하며 이는 전적으로 AP에 의존하게 됨. 따라서, AP는 유선망을 무선망으로 확장시켜주는 역할을 한다.

AP는 server가 되고, AP에 연결하려는 단말기(핸드폰, 노트북, 등)는 Wireless client가 된다.

Wireless Client

2.3 IEEE 802.11

IEEE 802.11은 흔히 무선랜, 와이파이(Wi-Fi)라고 부르는 무선 근거리 통신망(Local Area Network)을 위한 컴퓨터 무선 네트워크에 사용되는 기술로, IEEE의 LAN/MAN 표준 위원회 (IEEE 802)의 11번째 워킹 그룹에서 개발된 표준 기술을 의미한다.

IEEE 802.11은 현재 주로 쓰이는 유선 LAN 형태인 이더넷의 단점을 보완하기 위해 고안된 기술로, 이더넷 네트워크의 말단에 위치해 필요 없는 배선 작업과 유지관리 비용을 최소화하기 위해 널리 쓰이고 있다. 보통 폐쇄되지 않은 넓은 공간(예를 들어, 하나의 사무실)에 하나의 핫스팟을 설치하며, 외부 WAN과 백본 스위치, 각 사무실 핫스팟 사이를 이더넷 네트워크로 연결하고, 핫스팟부터 각 사무실의 컴퓨터는 무선으로 연결함으로써 사무실 내에 번거로이 케이블을 설치하고 유지보수를 하지 않아도 된다.

1) 802.11 (초기 버전)

802.11은 2Mbps의 최고속도를 지원하는 무선 네트워크 기술로, 적외선 신호나 ISM 대역인 2.4GHz 대역 전파를 사용해 데이터를 주고 받으며 여러 기기가 함께 네트워크에 참여할 수 있도록 CSMA/CA 기술을 사용한다.

하지만 규격이 엄격하게 정해지지 않아서 서로 다른 회사에서 만들어진 802.11 제품 사이에 호환성이 부족했고 속도가 느려서 널리 사용되지 않았다.

2) 802.11b

802.11b는 802.11 규격을 기반으로 더욱 발전시킨 기술로, 최고 전송속도는 11Mbps이나 실제로는 CSMA/CA 기술의 구현 과정에서 6-7Mbps 정도의 효율을 나타내는 것으로 알려져 있다.

표준이 확정되자마자 시장에 다양한 관련 제품이 등장했고, 이전 규격에 비해 현실적인 속도를 지원해 기업이나 가정 등에 유선 네트워크를 대체하기 위한 목적으로 폭넓게 보급되었으며, 공공장소에서 유무선 서비스를 제공하는 업체도 생겨났다.

3) 802.11a

세 번째로 등장한 전송방식인 802.11a는 5GHz 대역의 전파를 사용하는 규격으로, OFDM 기술을 사용해 최고 54Mbps까지의 전송 속도를 지원한다.

5GHz 대역은 2.4GHz 대역에 비해 다른 통신기기(무선 전화기, 블루투스 기기 등)와의 간섭이 적고, 더 넓은 전파 대역을 사용할 수 있다는 장점이 있지만, 신호의 특성상 장애물이나 도심 건물 등 주변 환경의 영향을 쉽게 받고, 2.4GHz 대역에서 54Mbps 속도를 지원하는 802.11g 규격이 등장하면서 현재는 널리 쓰이지 않고 있다.

4) 802.11g

네 번째로 등장한 802.11g 규격은 a 규격과 전송 속도가 같지만 2.4GHz 대역 전파를 사용한다는 점만 다르다. 널리 사용되고 있는 802.11b 규격과 쉽게 호환되어 현재 널리 쓰이고 있다.

5) 802.11n

802.11n은 상용화된 전송규격이다. 2.4GHz 대역과 5GHz 대역을 사용하며 최고 600Mbps까지의 속도를 지원하고 있다. 처음 Draft 1.0이 확정되었을 때, 대한민국의 경우 기술규격 내 주파수점유대역 폭의 문제(2개의 채널점유)로 최대150Mbps이하로 속도가 제한되었으나 2007년 10월 17일 전파연구소의 기술기준고시로 300Mbps이상까지 사용할 수 있게 되었다. 이 기술의 최종 표준안은 2008년 말

제정될 예정이었으나 2009년 9월 11일에서야 IEEE 802.11n-2009이 표준안으로 제정되었고 대한민국에 현재 상용화되어 있다. 다른 규격보다 승인 규격이 엄격하고 출력 규제가 심하여, 일부 회사에서는 이 규제를 지키지 않고 있다. IEEE 802.11n-2009 표준은 최대 600Mbps까지 전송속도를 높일수 있다.

6) 802.11ac

이 규격에 따르면 다중 단말의 무선랜 속도는 최소 1 Gbit/s, 최대 단일 링크 속도는 최소 500 Mbit/s 까지 가능하게 된다. 이는 더 넓은 무선 주파수 대역폭(최대 160 MHz), 더 많은 MIMO 공간적 스트림(최대 8 개), 다중 사용자 MIMO, 그리고 높은 밀도의 변조(최대 256 QAM) 등 802.11n 에서 받아들인 무선 인터페이스 개념을 확장하여 이루어진다.

7) 802.11ad

빔포밍 기술을 이용하여 최대 7Gb/s의 속도를 제공하는 전송규격이다. 기존 2.5GHz/5GHz 대신 60GHz 대역을 사용해 데이터를 전송하는 방식으로 대용량의 데이터나 무압축 HD 비디오등 높은 비트레이트 동영상 스트리밍에 적합하다. 하지만 60GHz는 장애물 통과가 어려워 10m이내 같은 공간 내에서만 사용이 가능하여 근거리 사용 기기만 이용 가능하다. 기존 2.4/5GHz 대역사이도 원활한 전환을 위해 '빠른 세션 전송'을 추가했으며 Tri-band 네트워킹, 무선 도킹, 유선과 동등한 데이터 전송 속도, 압축 스트리밍 비디오 지원 등의 보완이 이루어졌다.

IEEE 802.11 네트워크 환경은 인프라(infrastructure) 방식과 애드혹(Ad-Hoc) 방식으로 구성할 수 있다. 핫스팟에 여러 대의 클라이언트가 접속해 네트워크를 구성한다면 인프라망(하부구조 네트워크)이라고 부르고, 각 클라이언트가 핫스팟 없이 서로 데이터를 주고 받는다면 애드혹 네트워크라고 부른다. 보통 인프라망에는 핫스팟이 필요하므로 초기 설치 비용이 많이 들지만, 더 많은 클라이언트를 받아들일 수 있고 더 넓은 접속 반경을 제공해 주기 때문에 자주 쓰인다.

2.4 2.4GHz와 5GHz 무선랜 채널 및 채널 대역폭

IEEE 802.11ac는 높은 속도의 근거리 통신망(LAN)을 제공하기 위하여 현재 개발 중에 있는 802.11 무선 컴퓨터 네트워킹 표준 중의 하나이다. 5GHz 주파수에서 높은 대역폭(80MHz~160MHz)을 지원하고, 2.4GHz에선 802.11n과의 호환성을 위해 40MHz까지 대역폭을 지원한다. 2011년 1월 20일, IEEE 802.11 TGac (ac를 위한 Task Group)에 의해 초기 기술 사양 초안 0.1이 확정되었다. 표준의 마무리는 2012년 후기에, 최종 802.11 위원회의 승인은 2014년 초에 예상되고 있다. 연구 자료에 따르면, 802.11ac 사양의 장치들은 2015년에 일반화되어 전 세계에 10억 개 이상 퍼져 나갈 것으로 기대된다.

이론적으로, 이 규격에 따르면 다중 단말의 무선랜 속도는 최소 1 Gbit/s, 최대 단일 링크 속도는 최소 500 Mbit/s 까지 가능하게 된다. 이는 더 넓은 무선 주파수 대역폭(최대 160 MHz), 더 많은 MIMO

공간적 스트림(최대 8 개), 다중 사용자 MIMO, 그리고 높은 밀도의 변조(최대 256 QAM) 등 802.11n에서 받아들인 무선 인터페이스 개념을 확장하여 이루어진다.

2.5 SISO, MIMO

1) SISO

[Single Input Single Output]

단일 입력 단일 출력

수신 측과 송신 측에서 각각 1개의 안테나를 사용하는 무선 통신 안테나 기술. 무선 통신에서 송신 측과 수신 측이 각각 안테나 하나로 통신하는 경우 언덕, 계곡, 철탑 등 전파 경로상 장애로 다중 경로 현상이 나타나 페이딩으로 인한 문제가 발생하고, 무선 인터넷 등 디지털 통신에서는 데이터 속도 저하 및 오류 증가의 원인이 된다. 따라서 이러한 문제를 개선하기 위해 단일 입력 다중 출력(SIMO), 다중 입력 단일 출력(MISO), 다중 입력 다중 출력(MIMO) 등 스마트 안테나 기술이 사용된다.

2) MIMO

[Multiple Input Multiple Output]

다중 입력 다중 출력

다중의 입출력이 가능한 안테나 시스템을 이르는 말이다.

기지국과 휴대 단말기의 안테나를 2개 이상으로 늘려 데이터를 여러 경로로 전송하고 수신단에서 각각의 경로로 수신된 신호를 검출해 간섭을 줄이고 각각의 전송 속도를 낮출 수 있는 기술이다.

MIMO의 원리

기존 무선랜은 안테나가 2개 달려 있음에도 불구하고 유선망과 무선망을 연결시켜주는 AP(Access Point)의 방향에 따라 하나의 안테나만 이용했지만, MIMO는 두 개의 안테나가 동시에 동작하도록 해 고속의 데이터 교환을 가능하게 한다. N개의 송신안테나에 동일시간에 동일 주파수를 사용하여 독립적인 신호를 전송한다. 이렇게 송신된 신호들은 무선채널상에서 공간적으로 다른 페이딩(수신되는 전파가 지나온 매질의 변화에 따라 그 수신전파의 강도가 급격하게 변동되는 현상)을 겪게 되어 각 안테나로 수신되는 신호간에는 비상관성을 갖게 된다. 송신안테나마다 다른 신호를 송신함으로써, 기존보다 송신안테나 수(N개)만큼 더 많은 데이터를 송신할 수 있는 것이다.

MIMO의 장점

첫째, 기존 무선랜은 ieee802.11g라는 표준으로 54Mbps의 속도를 가졌다. 하지만 MIMO기술을 탑재한 무선랜 제품들은 250~500Mbps의 전송속도를 보여준다.

둘째, 도달거리도 획기적으로 개선된다. 기존 무선랜에 비해 전송속도와 도달거리를 4배정도 향상되었다. 바로 안테나 기술만으로 쉽게 전송속도와 통신거리를 획기적으로 향상시켰다.

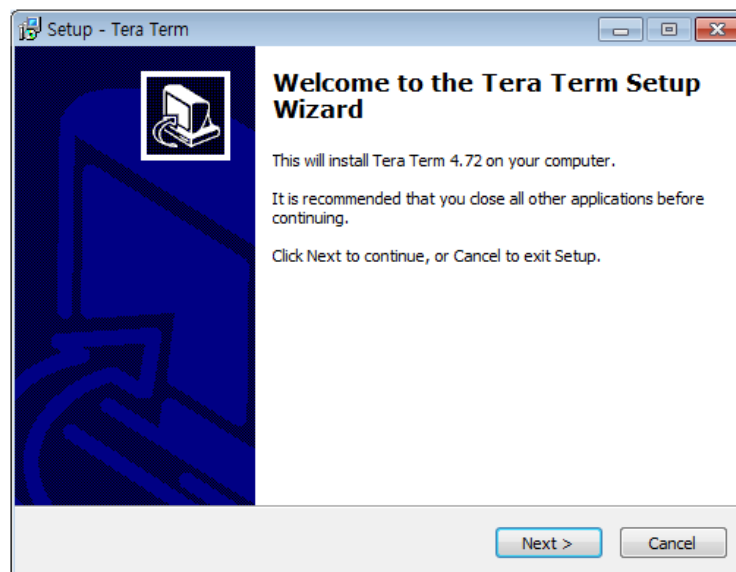
향후 전망

MIMO기반의 시스템이 소비자와 기업 분야에서 대량으로 사용되려면 여러 해가 지나야겠지만 그 동안에도 시스템의 개선과 향상은 계속될 것이다. MIMO는 무선랜 기술로 하여금 오늘날의 전형적인 유선통신대역폭(100Mbps/s)만큼 전달할 수 있는 미래의 잠재력을 보여줬다. 이러한 성능상의 돌파구로 소비자, 의료용, 산업용, 및 기업 데이터 통신에서 3G 무선 핫스팟(hotspot), 공중 무선 데이터 접근, HDTV 신호의 가정 전달 및 기타 여러 가지 응용에 사용될 것이다.

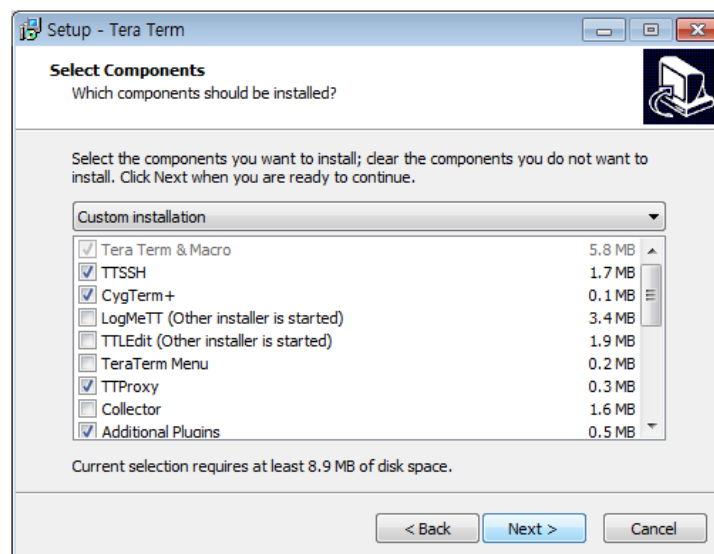
2.6 Teraterm

오픈 소스 기반으로 한 무료 터미널 프로그램 Teraterm이다. 본 교재에서는 Teraterm 터미널 프로그램을 사용하여 JSN270의 통신을 확인한다.

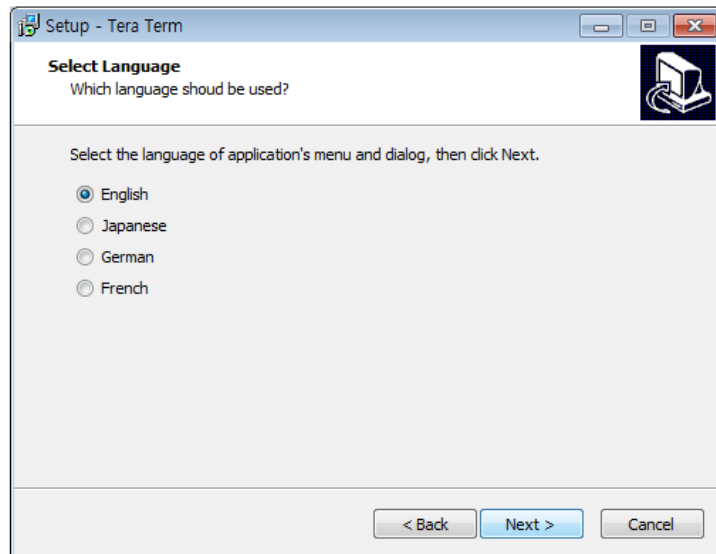
<http://en.osdn.jp/projects/ttssh2/> 에 접속하여 Teraterm 파일을 다운받는다.



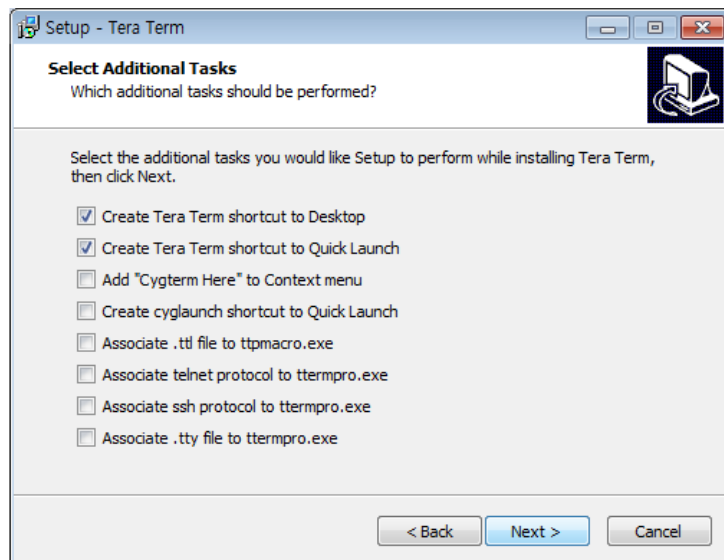
설치 첫 화면



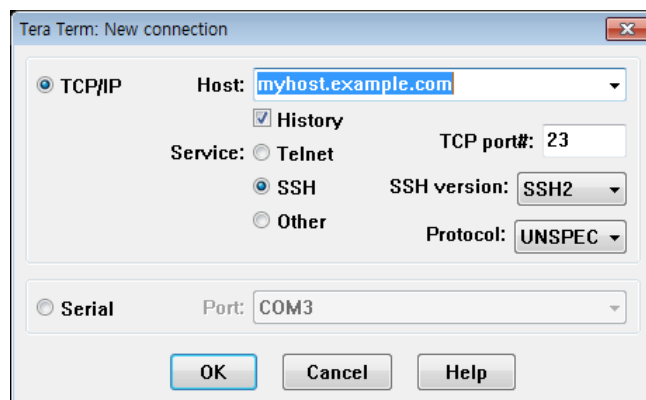
Teraterm에 포함된 여러 컴포넌트를 선택 할 수 있다. (본 교재에서는 기본 설정으로 사용)



언어 설정



추가 작업 선택 (본 교재는 기본으로 설정)



설치가 후 실행하면 시리얼통신, 텔넷, SSH통신을 할 수 있다.

3. 무선 암호화 표준

3.1 WEP

[Wired Equivalent Privacy]

유선급 보호

무선 LAN에서 사용하는 암호화 기법으로 동일한 키(key)와 알고리즘을 사용해 데이터를 암호화하고 해독하는 대칭키 알고리즘 방식을 기본으로 암호화 기법

유선 랜(LAN)에서 기대할 수 있는 것과 같은 보안과 프라이버시 수준의 무선 랜(WLAN)의 보안 프로토콜.

유선 랜은 일반적으로 건물 접근 통제와 같은 물리적 보안 체계로 되어 있지만, 무선 랜에서는 전파의 공간 전달 특성으로 인하여 별 효과가 없기 때문에 유선망의 보안도에 맞추어 유사한 보안 대책으로 데이터를 암호화한 것이다.

데이터 암호화는 노출되기 쉬운 회선 접속을 보호하고, 비밀 번호, 단말 간 암호, 가상 사설 통신망(VPN), 인증 등 전형적인 랜 보안 체계가 프라이버시를 보호하고 있다. IEEE WI-FI 표준 802.11b에 기술되어 있다

3.2 WPA

Wi-Fi Protected Access

무선 랜 보안 표준의 하나. WEP(Wired Equivalent Privacy) 키 암호화를 보완하는 TKIP(Temporal Key Integrity Protocol)라는 IEEE 802.11i 표준을 기반으로 하고 있으며, 인증 부문에서도 802.1x 및 EAP(Extensible Authentication Protocol)를 도입해 성능을 높인 것이다. 특히 패킷당 키 할당 기능, 키값 재설정 등 다양한 기능이 있기 때문에 해킹이 불가능하고 네트워크에 접근 시 인증 절차를 요구한다.

3.3 WPS

WPS [Wi-fi Protected Setup]

복잡한 공유기 보안 설정을 하지 않아도 WPS버튼만 누르면 보안이 적용된 무선 연결이 가능합니다. WPS 기능을 쓰려면 공유기와 연결하려는 기기가 WPS 를 지원해야 합니다. 간편하게 무선 보안 접속이 가능하지만 다른 보안 방식에 비해 해킹의 위험이 커 가급적 사용하지 않는 것이 좋습니다.

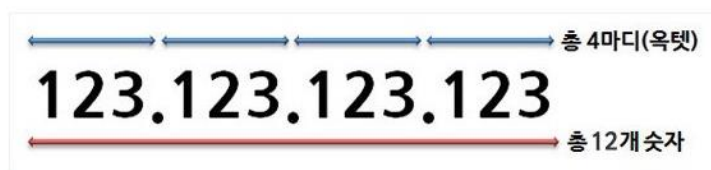
4. IP주소, 서브넷 마스크, 게이트웨이, DNS

1) IP 주소

편지는 받는 사람의 주소를 토대로 정확하게 배송된다. 이메일도 마찬가지로, 이메일 주소가 있어야 원하는 내용을 원하는 사용자에게 보낼 수 있다. 컴퓨터 네트워크에서 IP(Internet Protocol) 주소는 전 세계 컴퓨터에 부여된 고유의 식별 주소다. 기본적으로 네트워크(인터넷 등)에 연결된 모든 컴퓨터(또는 네트워크 기기)는 중복되지 않는 IP 주소를 가지고 있어야 한다. 그래야 상호 충돌 없이 네트워크에 연결될 수 있기 때문이다. 따라서 IP 주소는 컴퓨터끼리 서로 통신하기 위한 '전화번호'라 생각하면 이해가 쉽다. 참고로 IP 주소는 각 나라의 공인기관에서 할당/관리하는데, 우리나라의 경우 한국인터넷진흥원(www.krnic.or.kr)이 담당하고 있다.

- IP 주소의 기본 골격

2011년 1월 현재, 인터넷에 연결된 컴퓨터(혹은 모바일 기기 등)에 할당된 IP 주소는 거의 대부분 IPv4(IP version4) 형태다. 이 IPv4의 주소 체계는 3자리 숫자가 4마디로 표기된다(각 마디를 옥텟(octet)이라 한다). 표기되는 숫자는 총 12개다. 예를 들어, '123.123.123.123'과 같은 식이다. 단, 각 마디(옥텟)의 숫자는 255(0~255)를 넘을 수 없다. 컴퓨터 내부적으로는 이를 32개의 이진수(32비트)로 처리하기 때문이다. 따라서 IPv4 체계의 IP 주소로는 0.0.0.0부터 255.255.255.255까지 2의 32제곱, 약 42억 개의 주소를 사용할 수 있다. 숫자로 보면 IP 주소가 엄청나게 많은 듯하지만, 지금까지 인터넷이 발전되면서 점차 IP 주소 고갈 현상이 나타나고 있다. 전 세계 컴퓨터를 비롯한 다양한 기기가 고유한 IP 주소를 가져야 하기 때문이다.



IPv4의 주소 체계는 4마디, 총 12개 숫자로 표기된다. 컴퓨터 내부적으로는 32비트로 처리한다.

이와 같은 IP 주소 부족 문제를 해결하기 위해 제안된 새로운 IP 주소 체계가 [IPv6](#)다. 이는 IPv4가 32개의 이진수로 처리하던 IP 주소를 128개의 이진수(128비트)로 처리하게 된다. 따라서, 2의 128제곱인 약 3.4×10^{38} 에 해당하는 IP 주소를 사용할 수 있게 된다. 한 마디로 거의 무한대로 사용할 수 있는 정도다. 이렇게 되면 비단 컴퓨터 관련 기기뿐 아니라, 냉장고, TV, 세탁기, 전자레인지 등 일반 가전에도 IP 주소를 부여할 수 있어 각 기기 간의 통신이 가능해 진다(이러한 통

신 환경을 일컬어 '유비쿼터스(Ubiquitous)'라고도 한다). 다만 IPv6는 현재 테스트 단계라 현재의 IPv4 체계를 완전히 대체하기까지는 상당한 시간이 필요할 것으로 예상된다.

IPv4의 IP개수 = 2^{32} 개 = 42억 9496만 7296개

IPv6의 IP개수 = 2^{128} 개 = 340간 2823구 6692양 0938자 4634해 6337경 4607조 4317억 6821만 1456개

참고로 IP 주소는 대역에 따라 클래스(class) A, B, C, D, E 등으로 나눌 수 있다. 클래스 A는 대규모 네트워크 환경에 쓰이며, IP 주소 4마디 중 첫 번째 마디의 숫자가 1~126(약 1,600만 개 IP 주소 사용 가능)까지 사용된다. 클래스 B는 중규모 네트워크 환경용으로 129~191(약 65,000개 IP 주소 사용 가능)까지, 클래스 C는 소규모 네트워크 환경에 적용되는 것으로 192~223(254개 IP 주소 사용 가능)까지 사용된다. 클래스 D와 E가 223 이후의 숫자로 구성되는데, 이는 연구/개발용 IP 주소이므로 일반적인 용도로 활용되지 않는다. 한편 IP 주소는 사용 목적과 용도 등에 따라 여러 가지로 구분되기도 한다. 일반적으로 이들을 모두 자세하게 알 필요는 없지만, 인터넷과 관련 기기를 사용할 때 접할 수 있는 용어는 간단하게나마 숙지하는 것이 좋다.

- 사용 범위에 따른 구분

공인 IP 주소

단어 그대로, 공인기관에서 인증한 공개형(public) IP 주소다. 인터넷 유무선 공유기를 사용하지 않는 한 컴퓨터 등에서 사용하는 대부분의 IP 주소는 공인 IP 주소다. 우편물로 치면 우체국에서 배달하는 실제 주소인 셈이다. 이 주소는 외부로 공개되어 누구라도 그 주소로 우편물을 보낼 수 있는 것처럼, 공인 IP 주소도 외부에 공개되어 있어 다른 컴퓨터 등에서 검색, 접근이 가능하다. 예를 들어, 내 컴퓨터의 IP 주소가 100.100.100.100이라면 인터넷에 연결된 어떤 사용자(혹은 컴퓨터)라도 이 IP 주소를 토대로 내 컴퓨터에 (1차) 접근이 가능하다. 따라서 공인 IP 주소를 사용하려면 보안 장비(방화벽 등)가 반드시 필요하다. 다만 가정에서는 가입한 인터넷 서비스 회사(ISP, KT나 SK텔레콤 등)에서 보안 서비스를 제공하고 있기에 크게 걱정할 필요는 없다.

사설 IP 주소

공인 IP 주소가 공개형이라면 사설(private) IP 주소는 폐쇄형이다. 공인되지 않은 IP 주소라는 의미 때문이다. 즉, 이 사설 IP 주소는 외부에 공개되지 않아 외부에서 검색, 접근이 근본적으로 불가능하다. 사설 IP 주소는 주소 대역이 3개로 고정되어 있다. 이를테면, '192.168.xxx.xxx'와 '172.10.xxx.xxx', 그리

고 '10.xxx.xxx.xxx' 대역이다. 이러한 사설 IP 주소는 인터넷 유무선 공유기를 사용할 때 흔히 접하게 되는데, 하나의 공인 IP 주소를 공유하여 여러 대의 컴퓨터가 인터넷에 접속하게 하려면 사설 IP 주소가 필요하기 때문이다.

예를 들어, 그 동안 컴퓨터 한 대를 100.100.100.100라는 공인 IP 주소로 설정해 인터넷에 접속하다가 유무선 인터넷 공유기를 설치해 연결했다면, 이후로는 공유기 IP 주소가 100.100.100.100이 되고 공유기에 연결된 해당 컴퓨터에는 192.168.0.10 등과 같은 사설 IP 주소가 할당된다. 이러한 사설 IP 주소를 사용하는 이유는 두 가지다. 하나는 위에서 언급한 대로 IP 주소를 공유하기 위함이다. 이는 IPv4 체계의 IP 주소 부족 문제를 해결할 수 있는 방안이기도 하다.

공유기가 없다면 사무실에 있는 10대의 컴퓨터 각각에 모두 공인 IP 주소를 부여해야 하지만, 공유기가 있으면 1개 공인 IP 주소만 공유기에 할당하고, 10대의 컴퓨터는 사설 IP 주소를 각각 할당 받아 인터넷에 접속할 수 있게 된다. 또 하나의 이유는 보안 때문이다. 사설 IP 주소가 할당된 컴퓨터 등은 외부에서 검색, 접근이 기본적으로 불가능하다. 일반적으로 인터넷 공유기가 그러한 보안 장비의 역할(네트워크 방화벽)도 수행하고 있다.

4.1 자동 IP (DHCP)와 고정 IP

할당 방식에 따라 고정IP와 자동 IP로 나눌 수 있다.

고정 IP 주소

공인 IP 주소든 사설 IP 주소든 컴퓨터 등에 IP 주소를 설정하려면, 누군가(주로 서버 시스템)가 IP 주소를 컴퓨터에 배급, 할당해 줘야 한다. 공인 IP 주소라면 해당 인터넷 서비스 업체의 주소 할당 서비스가 사설 IP 주소라면 인터넷 공유기가 이 역할을 수행한다. 이때 IP 주소를 할당 받는 컴퓨터 등은 이를 자동으로 설정할지, 아니면 수동으로 설정할지를 선택할 수 있다. 고정 IP 주소는 사용자가 직접 IP 주소를 입력해 주소를 설정하는 방식을 말한다. IP 주소가 변경되면 안 되는 컴퓨터(파일 공유 서버 등) 등에 적합한 것으로 일반적으로 특별한 경우 외에는 사용할 기회가 거의 없다. 윈도우의 네트워크 연결 속성 페이지에서 '다음 IP 주소 사용' 옵션을 선택하고, 원하는 IP 주소를 4개 마디에 맞게 입력하면 된다(각 마디는 255를 넘을 수 없다).

유동 IP 주소

IP 주소를 할당하는 특정 서버가 보내 주는 정보 그대로 컴퓨터에 자동 설정되는 방식이다. 이를 'DHCP(Dynamic Host Configuration Protocol)' 서비스라 하는데, 컴퓨터가 부팅하면 DHCP 서버에 IP 주소 할당을 요청하고, 이를 수신한 DHCP 서버가 해당 컴퓨터에 IP 주소 등의 네트워크 정보를 전달하면, 컴퓨터에서는 이를 자동으로 등록, 설정하게 된다. 이 DHCP 서비스는 컴퓨터 수가 많은 환경에서 특히 간편하고 유용하다. 이론적으로 컴퓨터가 매번 부팅할 때마다(즉 IP 할당 요청이 발생할 때마다) IP 주소는 변경되지만, 이전에 할당 받았던 IP 주소 그대로 재할당 받는 경우가 더 많다. 인터넷 공유기는 사설 IP 주소를 생성(NAT 서비스)하여 각 컴퓨터 등에 자동 할당(DHCP 서비스)하는 역할을 동시에 수행하는 것이다.

- | |
|--|
| <ol style="list-style-type: none">1. 컴퓨터가 부팅되며 DHCP 서버가 존재함을 탐지한다2. DHCP 서버는 컴퓨터에 IP 주소 등이 할당되어 있는지 확인한다3. 컴퓨터가 DHCP 서버에 IP 주소 할당을 요청한다4. DHCP 서버는 IP 주소 할당 현황에 따라 IP 주소 등의 정보를 컴퓨터에 할당한다 |
|--|

물론 DHCP 서비스를 적용했어도 특정 컴퓨터에 고정 IP 주소를 할당해야 한다면, (앞서 설명한 대로) 윈도우 네트워크 속성 페이지에서 '다음 IP 주소 사용' 설정을 이용하면 된다. 그럼 그 컴퓨터는 IP 주소 자동 할당 목록에서 제외된다.

4.2 IP 주소와 서브넷 마스크 관계

IP 주소가 '바늘'이라면, 서브넷 마스크(subnet mask)는 '실'이라 할 수 있다. 네트워크에 연결되려면 이 두 정보는 반드시 짝을 이뤄 입력되어야 한다. 서브넷 마스크라는 용어를 풀어 보면 '하위(sub) 네트워크(net)를 구분하는 표기(mask)법'이라 할 수 있다. 다시 우편물로 예를 들자. 한 집에 두 가구가 사는 경우 주소는 같지만 최종 목적지가 다르다. 이때는 주소 마지막에 '1층' 또는 '2층'이라 표기해야 하는 것처럼, 서브넷 마스크도 네트워크 구역을 정확하게 구분하는데 사용된다. 즉, 같은 IP 주소 대역이라도 네트워크를 A, B, C 등의 구역으로 나눌 수 있는데, 이를 구분하는 기준이라는 것이다.



```
연결별 DNS 접미사. . . . : local
링크-로컬 IPv6 주소 . . . : fe80::5437:21a3:4d
IP 주소 . . . . . : 192.168.11.15
서브넷 마스크 . . . . . : 255.255.255.0
기본 게이트웨이 . . . . . : 192.168.11.1
```

이러한 서브넷 마스크는 사용자가 임의로 설정하는 게 아니라, IP 주소를 할당하는 인터넷 서비스 업체에서 제공하는 값을 그대로 입력해야 한다. 따라서 IP 주소가 정확하더라도 서브넷 마스크가 틀리면 인터넷 접속이 불가능하다. 서브넷 마스크는 일반적으로 '255.0.0.0(클래스 A)' 또는 '255.255.0.0(클래스 B)', '255.255.255.0(클래스 C)' 등의 형식으로 표기된다.

IP 주소 확인하는 방법

자신의 컴퓨터에 설치된 운영체계가 MS 윈도우 XP, 비스타, 7이라면, '시작' - '실행' - 'cmd' 입력하고 엔터를 치면 '명령 프롬프트' 창이 나타난다. 여기서 'ipconfig /all' 명령을 실행하면 현재 IP 주소 등을 비롯한 각종 네트워크 정보를 확인할 수 있다. 이 방법 외에도 IP 주소를 확인할 수 있는 방법은 다양하다.

4.3 게이트 웨이 기능

게이트웨이(gateway, 문화어: 망관문)는 컴퓨터 네트워크에서 서로 다른 통신망, 프로토콜을 사용하는 네트워크 간의 통신을 가능하게 하는 컴퓨터나 소프트웨어를 두루 일컫는 용어, 즉 다른 네트워크로 들어가는 입구 역할을 하는 네트워크 포인트이다. 넓은 의미로는 종류가 다른 네트워크 간의 통로의 역할을 하는 장치이다. 또한 게이트웨이를 지날 때마다 트래픽(traffic)도 증가하기 때문에 속도가 느려질 수 있다. 쉽게 예를 들자면 해외여행을 들 수 있는데 해외로 나가기 위해서 꼭 통과해야 하는 공항이 게이트웨이와 같은 개념이다.

게이트웨이는 서로 다른 네트워크 상의 통신 프로토콜(protocol, 통신규약)을 적절히 변환해주는 역할을 한다.

게이트웨이는 OSI 참조 모델의 전계층을 인식하여 전송방식이 다른 통신망도 흡수하여, 서로 다른 기종끼리도 접속을 가능하게 한다.

가장 잘 알려진 예로는 랜이나 무선 랜을 인터넷이나 다른 원거리 통신망에 연결하는 것이다. 이 경우 게이트웨이는 랜을 제공자 지정 네트워크에 연결함으로써 인터넷에 연결할 수 있게 된다. 가정의 경우 이 게이트웨이는 상주 게이트웨이라 부른다.

가까운 예로는 인터넷 공유기로 볼 수 있다.

마이크로소프트 윈도우 기준으로 컴퓨터에서 직접적으로 찾아보고 싶은 경우, 명령 프롬프트에서 ipconfig을 타이핑하여 기본 게이트웨이를 보면 시각적으로 확인할 수 있다.



! 게이트웨이는 현재 사용자가 위치한 네트워크에서 다른 네트워크로 이동하기 위해 반드시 거쳐야 하는 거점을 의미한다

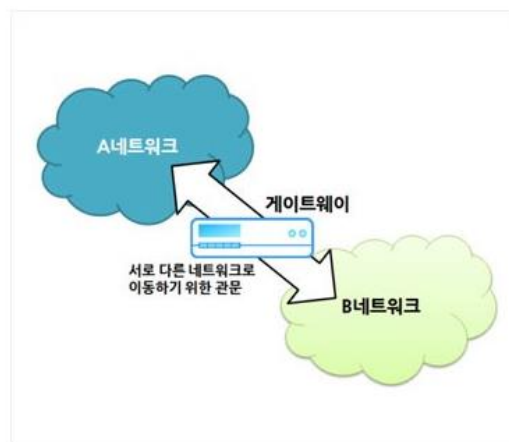
두 컴퓨터(노드-node라고도 함)가 네트워크 상에서 서로 연결되려면 동일한 통신 프로토콜(protocol,

통신 규약)을 사용해야 한다. 따라서 프로토콜이 다른 네트워크 상의 컴퓨터와 통신하려면 두 프로토콜을 적절히 변환해 주는 변환기가 필요한데, 게이트웨이가 바로 이러한 변환기 역할을 한다. 한국인과 미국인 사이에 원활한 의사소통을 위해 통역사를 두는 것과 동일하다.

- 게이트웨이는 네트워크간 통게이트

게이트웨이는 일반적으로 하드웨어 형태로 제공되며, 내부적으로 복잡한 원리로 작동하지만 외형은 생각보다 간단하다. 흔히 보는 네트워크 허브나 스위치 등과 비슷하게 생겼다. 또한 기능이나 용도, 적용 범위 등에 따라 손바닥만한 제품부터 소형 냉장고만한 제품까지 크기도 다양하다. 물론 설치와 설정 작업은 네트워크 전문가가 아니면 처리하기가 매우 어렵다.

게이트웨이는 또한 라우터(router)와 동일한 개념으로 이해할 수 있다. 라우터는 네트워크 장비의 일종으로, 패킷(packet, 네트워크 전송 데이터의 최소 단위)을 다른 네트워크 보내주는(forward) 역할을 한다. 이와 함께 최적의 네트워크 경로를 찾아주는 역할도 함께 수행한다. 이렇듯 라우터도 이기종 네트워크를 연결한다는 부분에서 게이트웨이와 상통한다(다만 게이트웨이는 라우터보다 포괄적인 개념이다).



흔히 사용하는 인터넷 유무선 공유기도 게이트웨이의 역할을 한다

우리는 컴퓨터 사용 환경에서 게이트웨이를 늘 사용하고 있다. 다만 그에 대한 관심과 지식이 적어 우리들 눈에 잘 띄지 않을 뿐이다. 가깝게는 인터넷 유무선 공유기(이하 공유기)가 우리가 만나는 첫 번째 게이트웨이이다. 공유기는 사용자 컴퓨터의 네트워크와 인터넷을 연결하여 사용자가 웹 사이트에 접근할 수 있도록 관문을 열어 준다. 사용자가 속해 있는 (로컬)네트워크의 통신 프로토콜(예, netbios)과 인터넷의 통신 프로토콜(예, http)이 다르기 때문이다. 참고로 공유기는 게이트웨이의 역할과 라우터의 역할, 방화벽 역할 등을 동시에 제공하는 종합 네트워크 장비다.

한편 자신의 컴퓨터에서 목적지 네트워크까지 도달하기까지 여러 개의 게이트웨이를 거칠 수 있다.

고속도로를 갈아탈 때마다 톨게이트를 지나야 하는 것과 다름 없다. 또한 톨게이트를 지날 때마다 통행료가 부가되듯, 게이트웨이를 거칠 때마다 네트워크 부하(트래픽, traffic)도 증가하여 전송 속도가 느려질 수 있다(이때 거치는 게이트웨이의 수를 '홉 카운트'-hop count-라고도 한다).

- 인터넷을 위한 필수 조건

해당 컴퓨터가 속해 있는 (로컬)네트워크 구역 내에서는 IP 주소 와 서브넷 마스크(subnet mask)만 있어도 주변 컴퓨터와 통신이 가능하다. 다른 네트워크 구역으로 나갈 필요가 없기 때문이다. 하지만 인터넷 등의 이기종 네트워크로 나가기 위해서는 게이트웨이(라우터 등)가 있어야 하고, IP 주소, 서브넷 마스크와 함께 게이트웨이 주소까지 정확하게 설정해야 한다.

컴퓨터가 서로 통신하기 위해서는 모든 컴퓨터마다 유일한 IP 주소를 할당해야 하듯, 게이트웨이에도 중복되지 않는 IP 주소가 필요하다. 이 IP 주소를 토대로 각 컴퓨터가 다른 네트워크와 연결된다. 일반적으로 게이트웨이의 IP 주소는 해당 네트워크 내 컴퓨터에 할당된 IP 주소 중 끝자리(4번째 옥텟)만 다른 형태다. 대개 1을 지정한다. 이를 테면 컴퓨터 IP 주소가 123.123.123.123이라면, 게이트웨이 주소는 123.123.123.1이 된다. 물론 게이트웨이 IP 주소 설정이 잘못되면 외부 네트워크(인터넷) 연결이 불가능하다.

4.4 DNS

도메인 네임 시스템(Domain Name System, **DNS**)

도메인 네임 시스템(Domain Name System, DNS)는 호스트의 도메인 이름을 호스트의 네트워크 주소로 바꾸거나 그 반대의 변환을 수행할 수 있도록 하기 위해 개발되었다. 특정 컴퓨터(또는 네트워크로 연결된 임의의 장치)의 주소를 찾기 위해, 사람이 이해하기 쉬운 도메인 이름을 숫자로 된 식별 번호(IP 주소)로 변환해준다. 도메인 네임 시스템은 흔히 "전화번호부"에 비유된다. 인터넷 도메인 주소 체계로서 TCP/IP의 응용에서, `www.example.com`과 같은 주 컴퓨터의 도메인 이름을 `192.168.1.0`과 같은 IP 주소로 변환하고 라우팅 정보를 제공하는 분산형 데이터베이스 시스템이다.

1) DNS 구조

도메인 이름 공간

도메인 이름 공간은 도메인 이름을 트리 형태로 구성한 것이다. 트리의 각 노드는 0개 이상의 리소스 레코드(resource record)를 가진다. 트리는 루트 존(root zone)에서 시작하여 여러 개의 하위 존으로 나뉜다. 각 DNS 존은 하나의 권한있는 네임서버(authoritative nameserver)에 의해 관리되는 노드들의 집합이다. (하나의 네임서버가 여러 개의 존을 관리할 수도 있다.)

관리 권한은 분할되어서 새로운 존을 형성할 수 있다. 이 때 기존의 도메인 이름 공간의 일부분이 서브도메인 형태로 다른 네임서버에 권한이 위임된다고 한다. 그리고 기존의 존은 새로운 존에 대한 권한을 잃게 된다.

도메인 이름 형성

도메인 이름 형성에 관한 규칙은 RFC 1035, RFC 1123, RFC 2181에 정의되어 있다. 도메인 이름은 한 개 이상의 부분(레이블)로 이루어 지고, 점으로 구분하여 붙여 쓴다.(예: `example.com`)

- 가장 오른쪽 레이블은 최상위 도메인을 의미한다. 예를 들어, 도메인 이름 `www.example.com`은 최상위 도메인 `com`에 속한다.
- 도메인의 계층 구조는 오른쪽부터 왼쪽으로 내려간다. 왼쪽의 레이블은 오른쪽의 서브도메인이다. 예를 들어, 레이블 `example`은 `com` 도메인의 서브도메인이며, `www`는 `example.com`의 서브도

메인이다. 서브도메인은 127단계까지 가능하다.

- 각 레이블은 최대 63개 문자를 사용할 수 있고, 전체 도메인 이름은 253개 문자를 초과할 수 없다. 실제로는 도메인 레지스트리에 더 짧은 제한을 가질 수 있다.
- 도메인 이름은 기술적으로 옥텟으로 표현할 수 있는 모든 문자를 사용할 수 있다(RFC 3696). 그러나 DNS 루트 존과 대부분의 서브도메인에서는 제한된 형식과 문자들만 허용한다. 레이블에 사용될 수 있는 문자는 아스키 문자 집합의 부분집합과 알파벳 a부터 z, A부터 Z, 숫자 0부터 9 와 하이픈(-)을 포함한다. 이 규칙은 letter(문자), digit(숫자), hyphen(하이픈)의 첫 글자를 따서 LDH 규칙이라고 한다. 도메인 이름은 대소문자 구분 없이 해석되고, 레이블은 하이픈으로 시작하거나 끝날 수 없다.

2) 주요 DNS 목록

DNS를 잘못 설정할 경우 인터넷 사용이 힘들 수 있다. 신뢰할 수 없는 DNS는 해킹, 파싱 등에 노출 되므로, 신뢰할 수 있는 DNS 서버만 이용해야 한다.

운영주체	기본 DNS	보조 DNS
KT	168.126.63.1	168.126.63.2
LG유플러스	164.124.101.2	203.248.252.2
SK브로드밴드	210.220.163.82	219.250.36.130
구글	8.8.8.8	8.8.4.4
시만텍	199.85.126.10	199.85.127.10
코모도	8.26.56.26	8.20.247.20

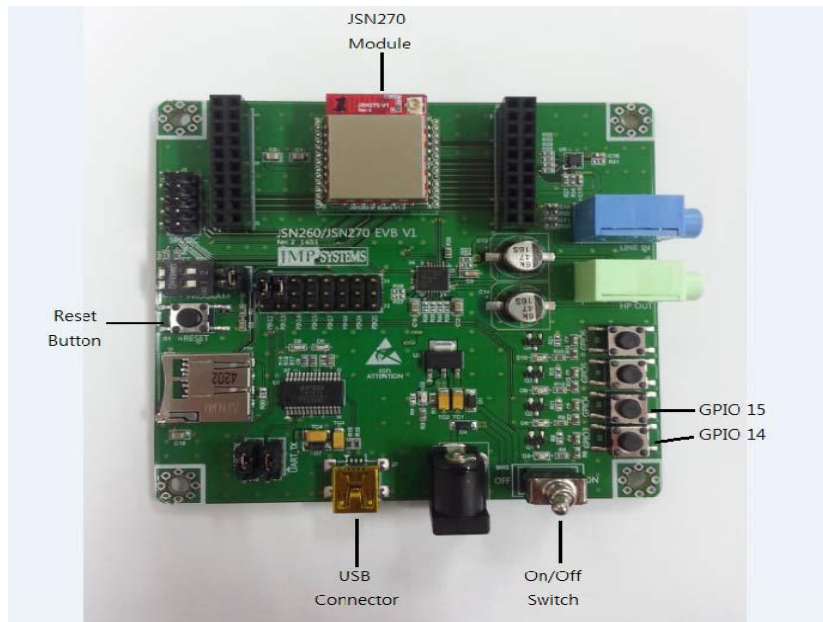
기본적으로 통신사가 제공하는 DNS는 서버가 한국에 소재하여 빠른 응답속도를 보여준다.

구글과 시만텍의 DNS는 한국에 서버가 소재하진 않지만, 인접국인 일본에 서버가 소재하고 있어, 40ms 미만의 응답속도를 보여준다.

5. JSN270 예제 실습

5.1 AT 명령어 실습

명령어 실습은 JSN270 EVB 보드를 이용한다.



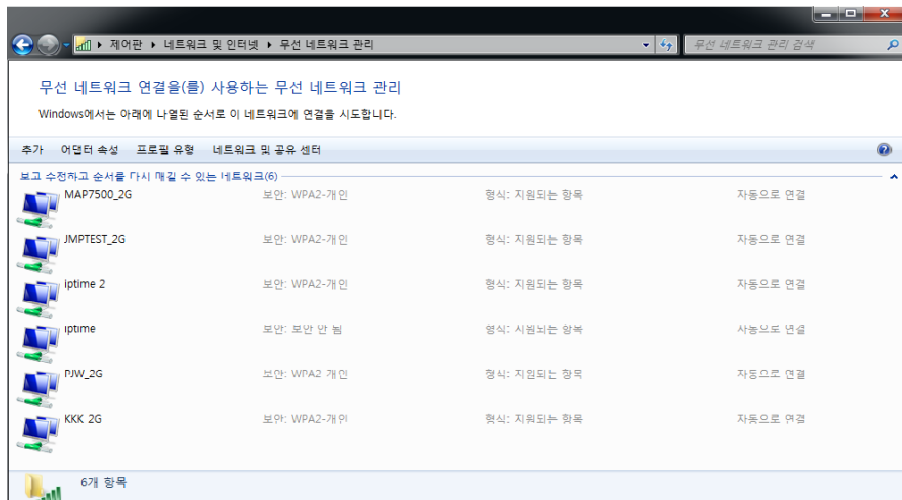
Teraterm 설치 후, 아래의 사이트에서 소켓 시험 프로그램인 SocketTest를 다운로드 받아서 설치한다.

<http://sourceforge.net/projects/sockettest/>

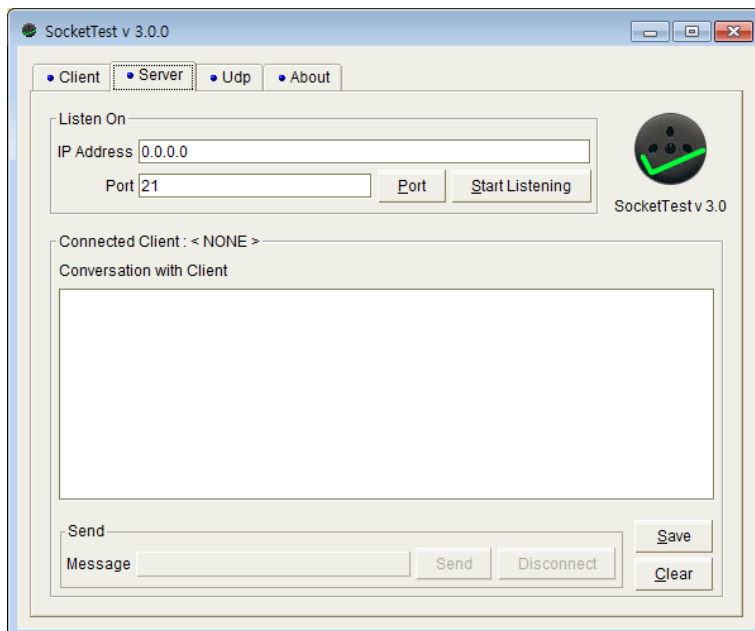
1) JSN270을 STA 모드로 동작하기

다음의 예제는 JSN270 을 STA 모드로 동작해서 무선 네트워크에 연결하고 TCP 클라이언트로 동작하여 PC에 설정해 놓은 TCP 서버에 연결하도록 합니다.

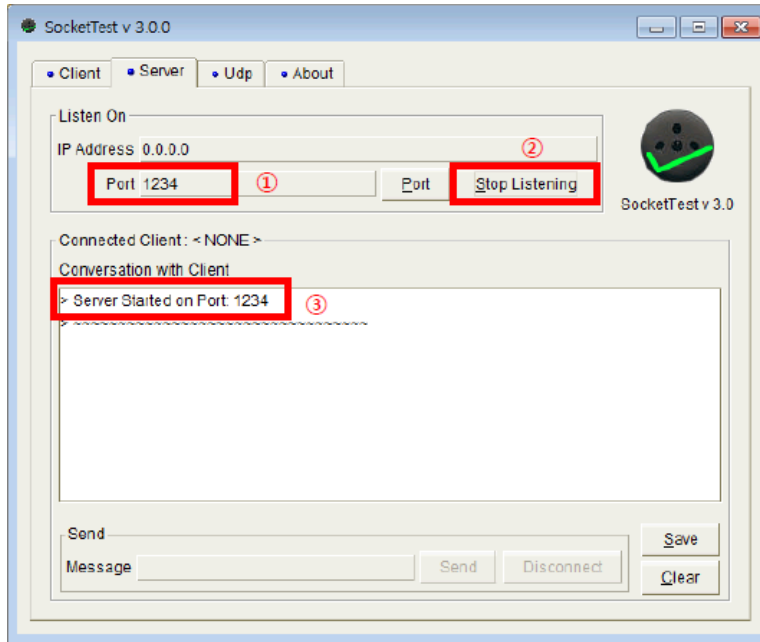
Step 1. PC를 SSID가 iptime인 AP에 무선 접속합니다.



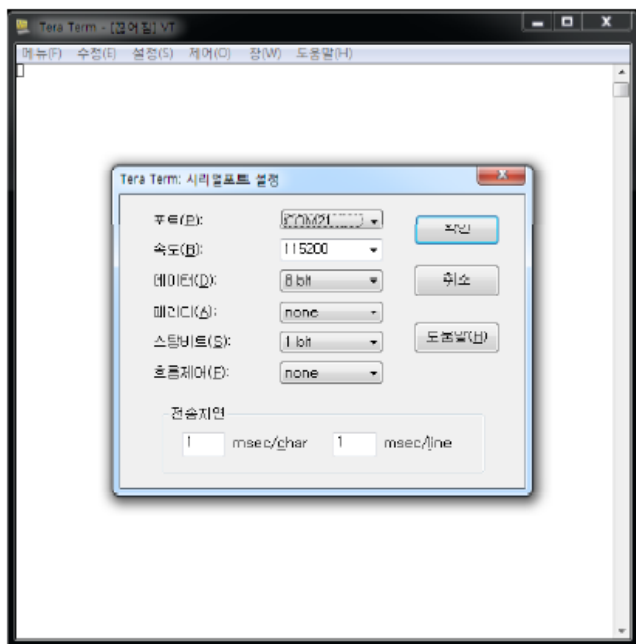
Step 2. SocketTest 프로그램을 실행하고 Server를 선택합니다.



Step 3. 아래 그림과 같이 Port를 설정한 후 Start listening 버튼을 누르면 TCP 서버가 실행됩니다.



Step 4. PC 에서 TeraTerm 을 실행하고 설정메뉴에서 시리얼 포트를 설정합니다. 이때 포트는 PC 에 따라 달라질 수 있습니다.

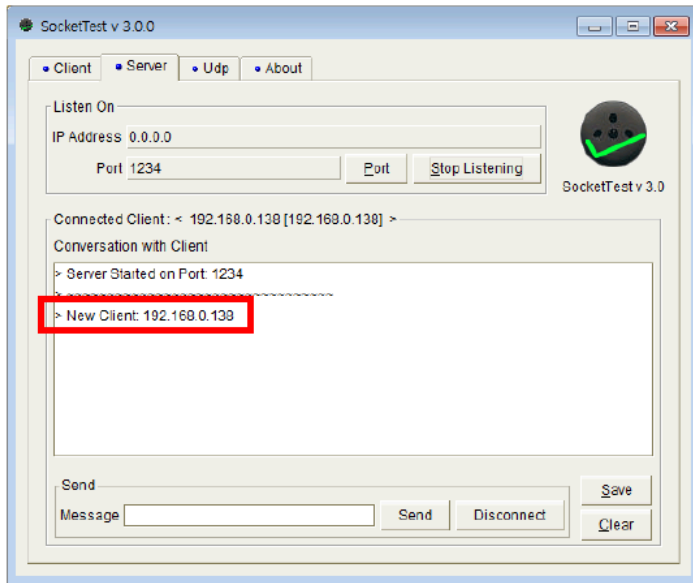


Step 5. 다음과 같이 AT 명령어를 이용하여 JSN270 을 AP 에 연결 후 위에서 실행한 PC 의 TCP Server 에 접속합니다.

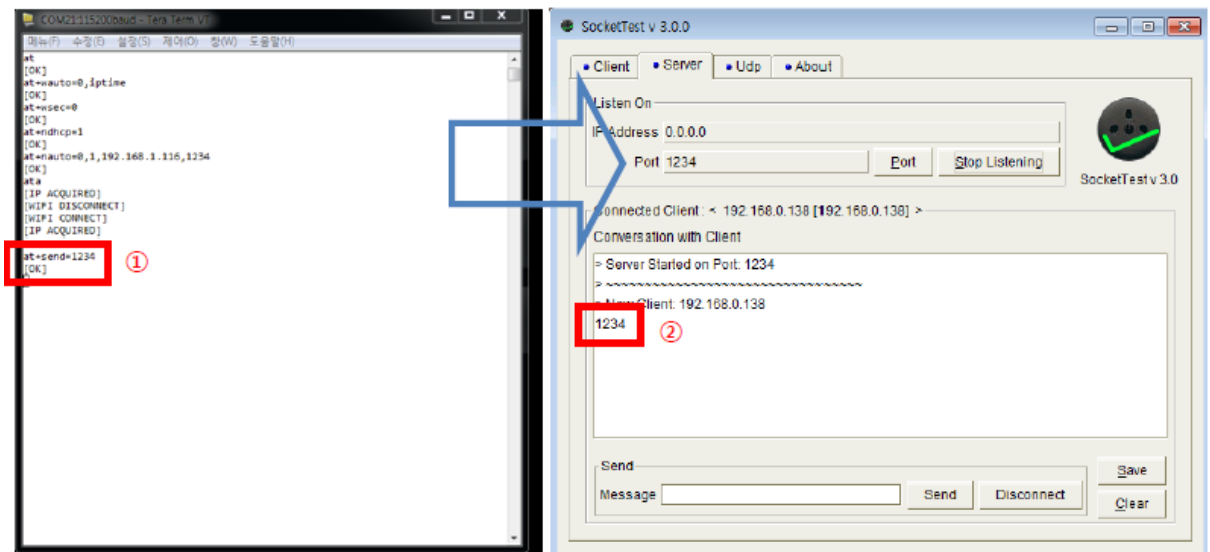
```
COM23:115200baud - Tera Term VT
File Edit Setup Control Window Help

Serial2HiFi APP
at+if
[OK]
at+uauto=0, iptime          ← SSID가 iptime인 AP에 연결
[OK]
at+uapsec=0                ← 무선 암호화 사용 안함
[OK]
at+ndhcp=1
[OK]
at+nauto=0, 1, 192.168.0.7, 1234 ← TCP 서버 192.168.0.7의 포트 1234에 접속
[OK]
ata
[WIFI CONNECT]
[IP ACQUIRED]
CONNECT 0                  ← TCP 서버에 접속됨
```


Step 6. JSN270 이 TCP 서버에 연결이 완료되면 아래와 같이 문구가 뜹니다.



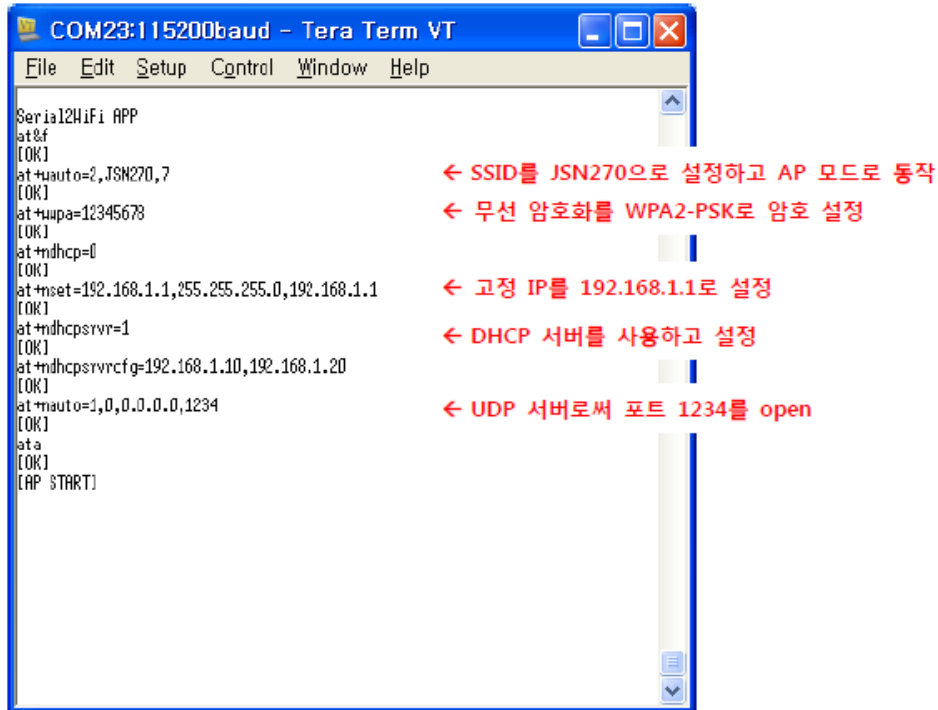
Step 7. 다음과 같이 JSN270 터미널에서 at+send 명령어를 사용해서 PC 로 데이터를 전송하면 PC 의 SocketTest 에서 전송된 데이터를 확인할 수 있습니다.



2) JSN270 을 AP 모드로 동작하기

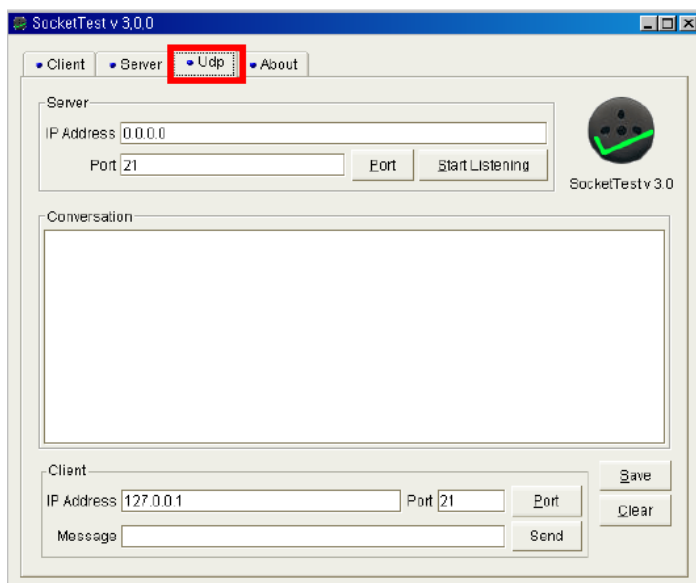
다음의 예제는 JSN270 을 AP 모드로 설정하고 UDP 서버로 동작해서 PC 를 JSN270 에 무선 접속 후 UDP로 연결합니다.

Step 1. 다음과 같이 JSN270 을 AP 모드 설정하고 UDP 서버를 엽니다.

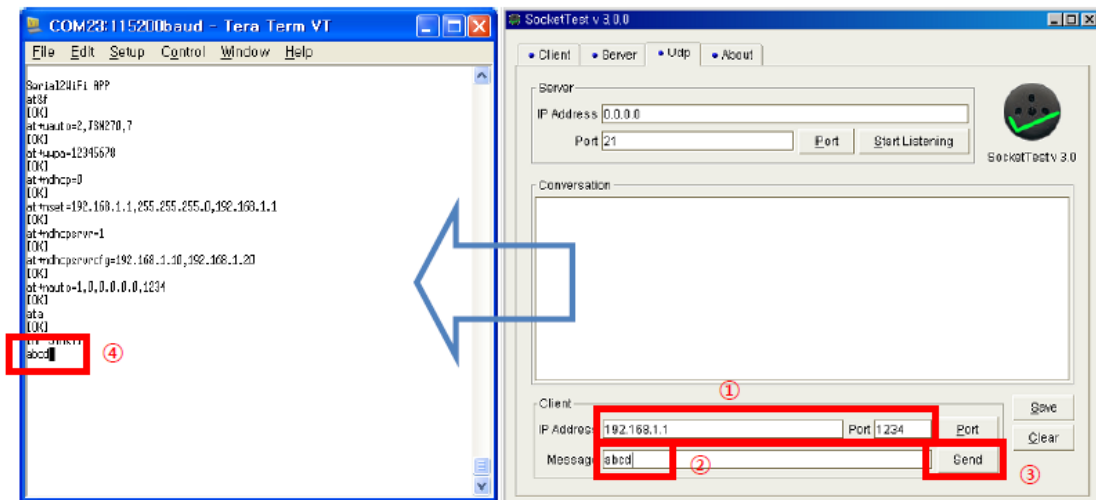


Step 2. PC 를 JSN270 에 무선 접속합니다. SSID 는 JSN270, 암호는 12345678

Step 3. SocketTest 를 실행하고 UDP 탭을 선택합니다.



Step 4. 아래 그림과 같이 PC 의 SocketTest 에서 IP Address 와 Port 를 설정하고 데이터를 전송하면 JSN270 터미널에 전송한 데이터가 표시됩니다.

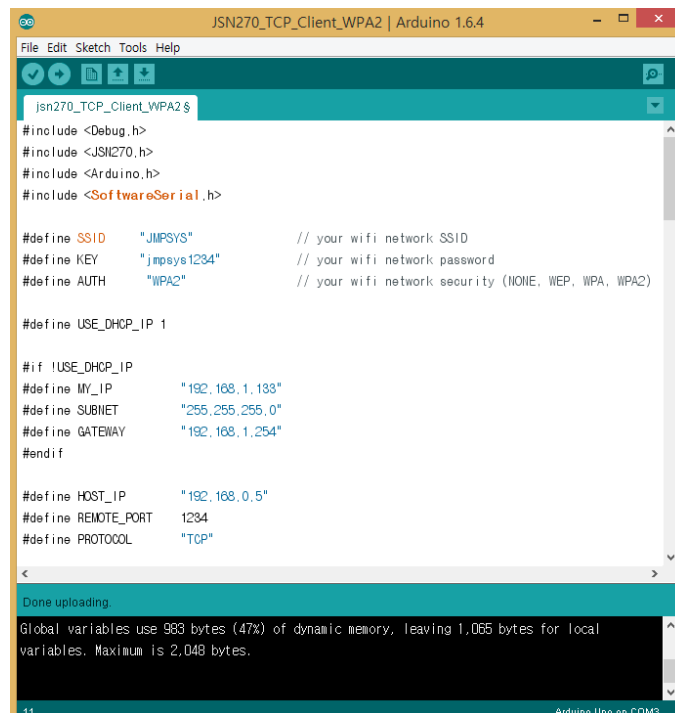


5.2 TCP Client WPA2

TCP Client WPA2 부터의 실습은 Arduino Uno 보드를 사용하여 실습한다.

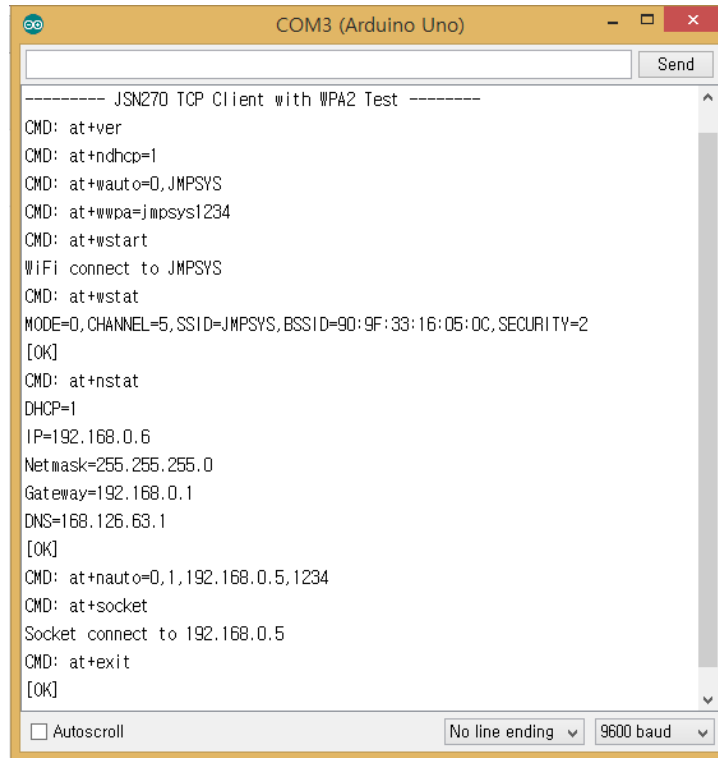


AP의 암호화 방식을 WPA2로 설정해준다.



File – Examples – JSN270 – TCP_Client_WPA2 에서 예제를 불러온다.

AP의 SSID와 KEY를 설정해준 뒤, 암호화 방식을 WPA2로 설정해준다.



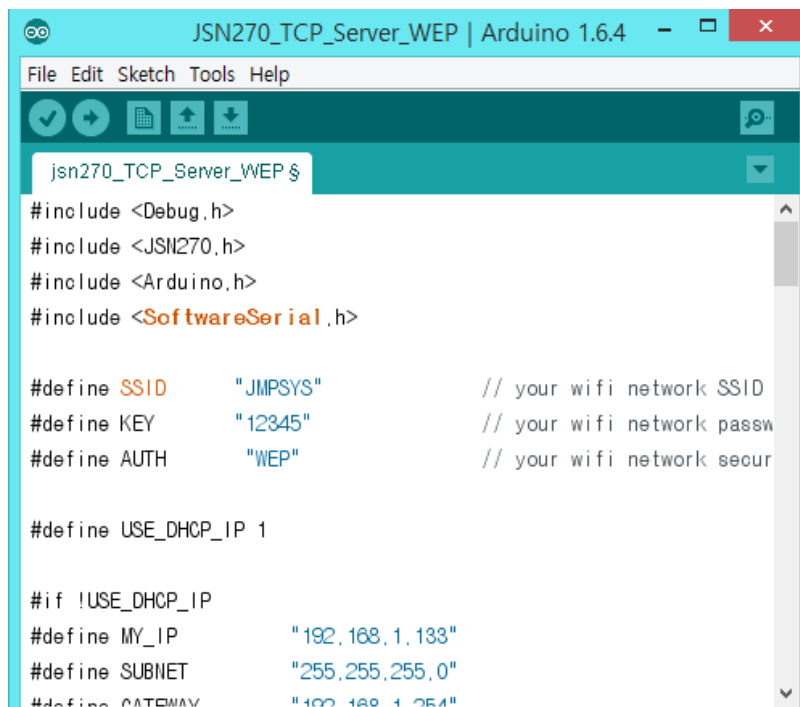
< 시리얼 모니터 창 결과 >

5.2 TCP Server Web



AP의 암호화 방식을 WEP로 설정해준다.

JSN270은 개방모드 WEP를 사용한다.



File – Examples – JSN270 – TCP_Server_WEP 에서 예제를 불러온다.

AP의 SSID와 KEY를 설정해준 뒤, 암호화 방식을 WEP로 설정해준다.

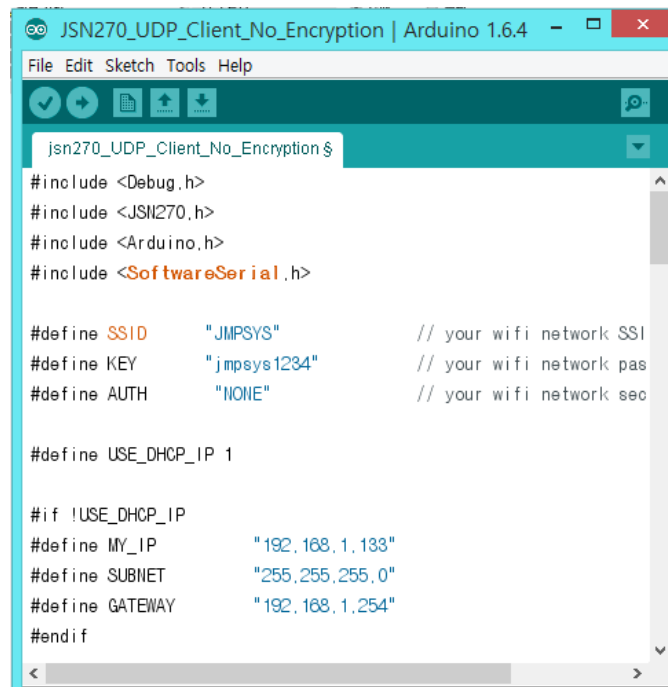
```
COM3 (Arduino Uno)
|
|
|
|
|----- JSN270 TCP Server with WEP Test -----
CMD: at+ver
CMD: at+ndhcp=1
CMD: at+wauto=0,JMPSYS
CMD: at+wep=12345
CMD: at+wstart
WiFi connect to JMPSYS
CMD: at+wstat
MODE=0, CHANNEL=5, SSID=JMPSYS, BSSID=90:9F:33:16:05:0C, SECURITY=1
[OK]
CMD: at+nstat
DHCP=1
IP=192.168.0.6
Netmask=255.255.255.0
Gateway=192.168.0.1
DNS=168.126.63.1
[OK]
CMD: at+nauto=1,1,0.0.0.0,1234
CMD: at+socket
Waiting for connection...
CMD: at+exit
[OK]
```

< 시리얼 모니터 창 결과 >

5.3 UDP Client No_Encryption

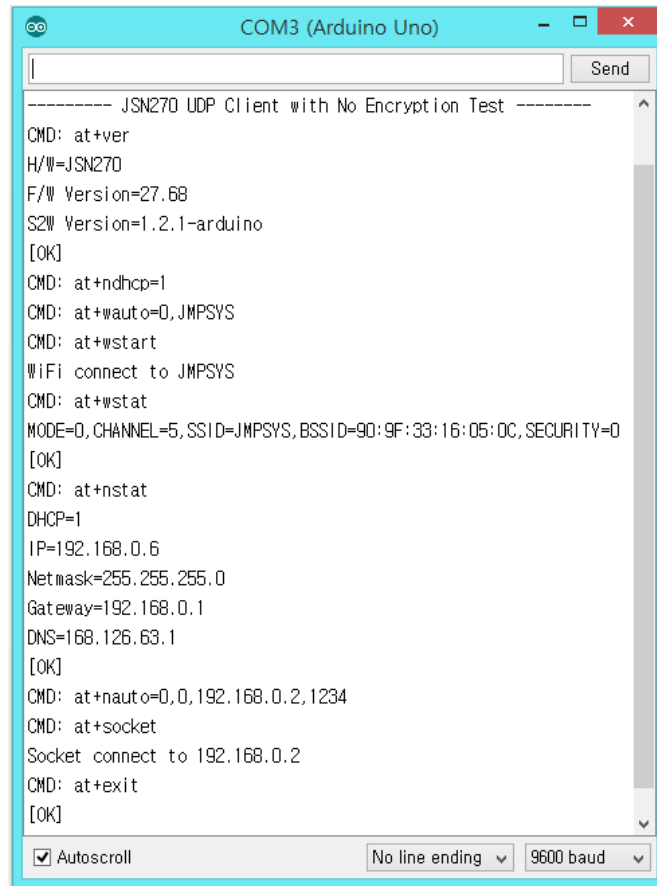


AP의 암호화를 사용하지 않음으로 설정한다.



File – Examples – JSN270 - UDP_Client_No_Encryption 에서 예제를 불러온다.

AP의 SSID와 KEY를 설정해준 뒤, 암호화 방식을 None로 설정해준다.



< 시리얼 모니터 창 결과 >

5.4 UDPTCP Server WPA



```
JSN270_UDP_Server_WPA | Arduino 1.6.4
File Edit Sketch Tools Help

jsn270_UDP_Server_WPA$

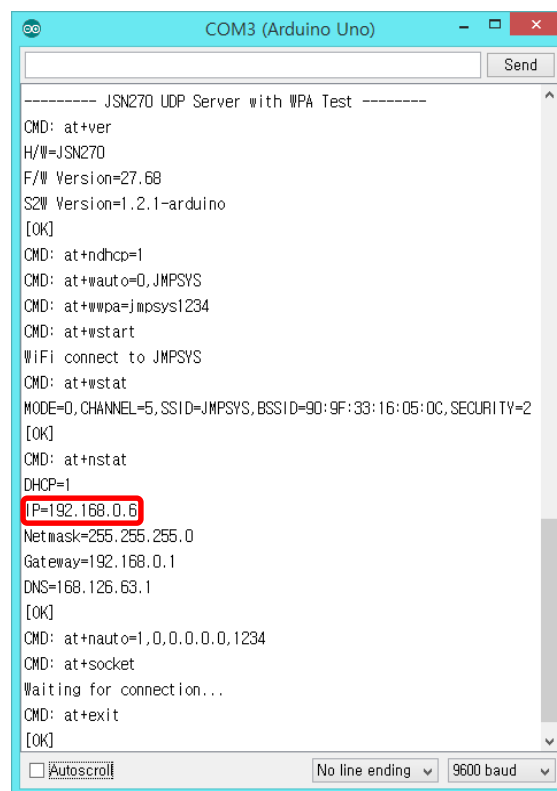
#include <Debug.h>
#include <JSN270.h>
#include <Arduino.h>
#include <SoftwareSerial.h>

#define SSID      "JMPSYS"           // your wifi network SSID
#define KEY       "jmpsys1234"       // your wifi network password
#define AUTH      "WPA"             // your wifi network security mode

#define USE_DHCP_IP 1

#if !USE_DHCP_IP
#define MY_IP      "192.168.1.133"
#define SUBNET     "255.255.255.0"
#define GATEWAY    "192.168.1.254"
#endif
```

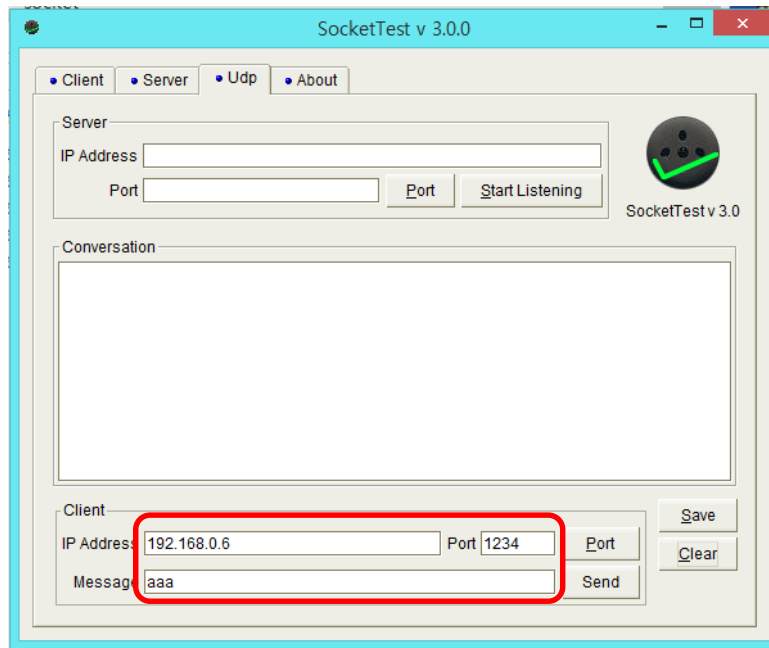
File – Examples – JSN270 - UDP_Server_WPA 에서 예제를 불러온다.
AP의 SSID와 KEY를 설정해준 뒤, 암호화 방식을 WPA로 설정해준다



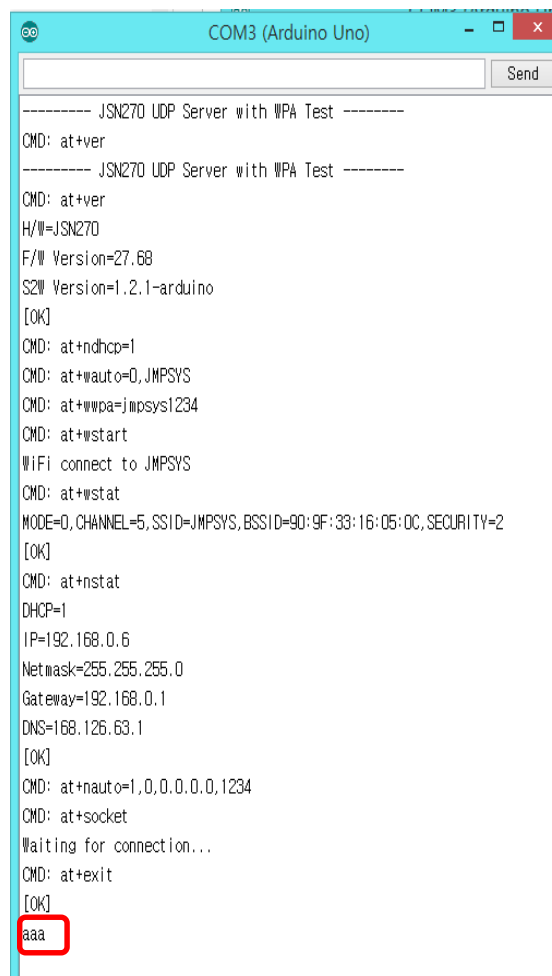
```
COM3 (Arduino Uno)
Send

----- JSN270 UDP Server with WPA Test -----
CMD: at+ver
H/W=JSN270
F/W Version=27.68
S2W Version=1.2.1-arduino
[OK]
CMD: at+ndhcp=1
CMD: at+wauto=0,JMPSYS
CMD: at+wpa=jmpsys1234
CMD: at+wstart
WiFi connect to JMPSYS
CMD: at+wstat
MODE=0,CHANNEL=5,SSID=JMPSYS,BSSID=90:9F:33:16:05:0C,SECURITY=2
[OK]
CMD: at+nstat
DHCP=1
IP=192.168.0.6
Netmask=255.255.255.0
Gateway=192.168.0.1
DNS=168.126.63.1
[OK]
CMD: at+nauto=1,0,0.0.0.0,1234
CMD: at+socket
Waiting for connection...
CMD: at+exit
[OK]
```

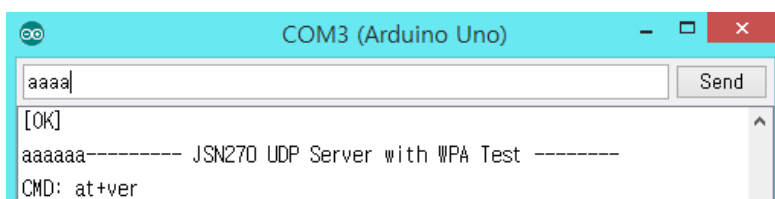
< 시리얼 모니터 창 결과 >

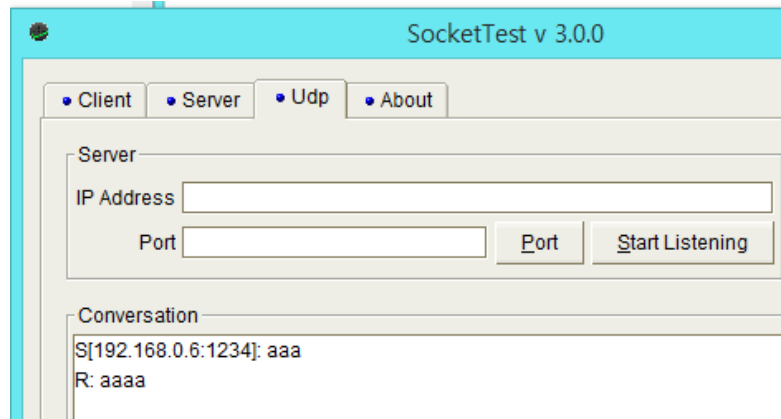


JSN270의 IP의 1234포트로 접속한 뒤, 'aaa' 를 전송



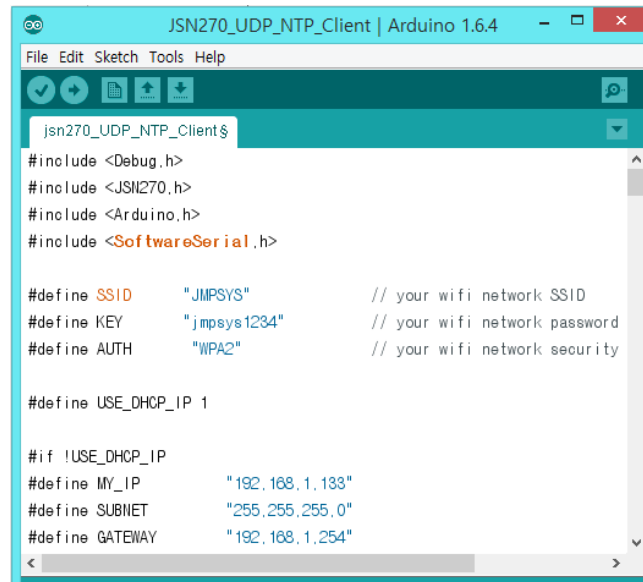
Arduino 시리얼 모니터 창에 aaa가 수신된 것을 확인할 수 있다.





Arduino 시리얼 모니터 창에 'aaaa'를 송신하였더니 SocketTest에서 'aaaa'가 수신 된 것을 확인 할 수 있다.

5.5 UDP NTP Client



```
jsn270_UDP_NTP_Client$
#include <Debug.h>
#include <JSN270.h>
#include <Arduino.h>
#include <SoftwareSerial.h>

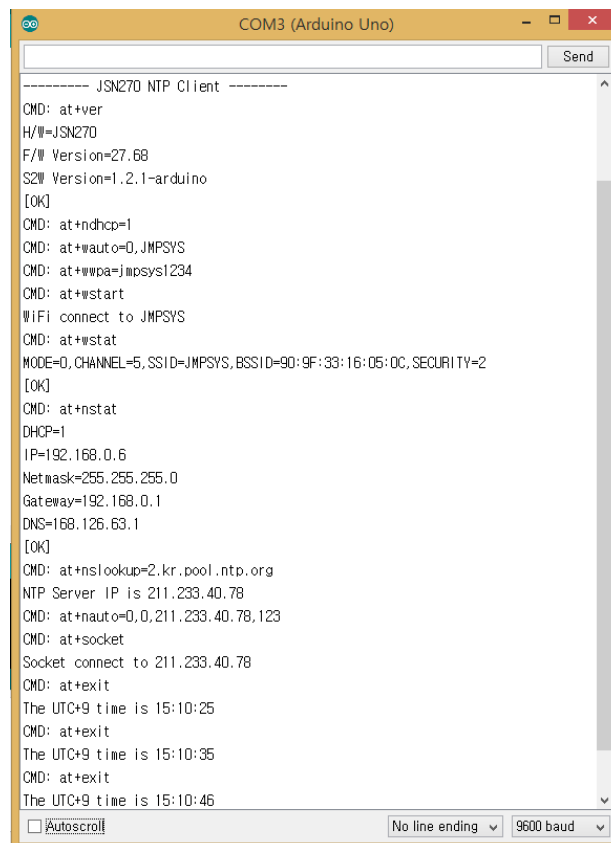
#define SSID      "JMPSYS"           // your wifi network SSID
#define KEY       "jmpsys1234"       // your wifi network password
#define AUTH      "WPA2"            // your wifi network security

#define USE_DHCP_IP 1

#if !USE_DHCP_IP
#define MY_IP      "192.168.1.133"
#define SUBNET     "255.255.255.0"
#define GATEWAY    "192.168.1.254"
```

File – Examples – JSN270 - UDP_NTP_Client에서 예제를 불러온다.

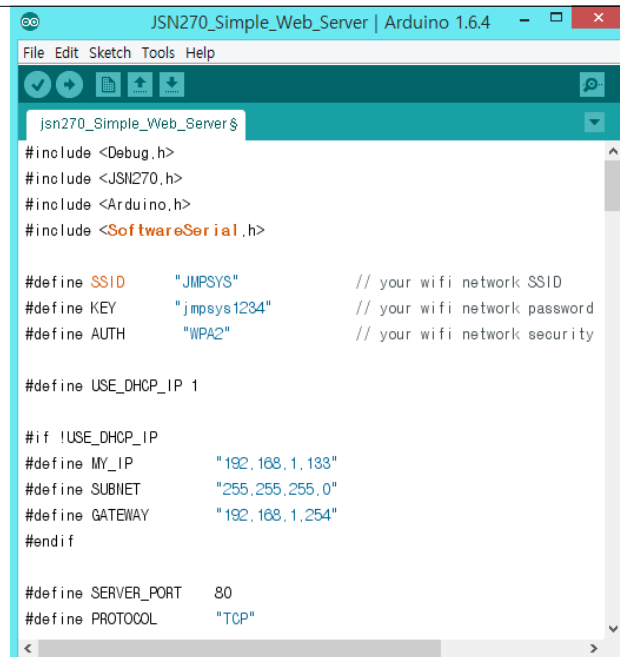
AP의 SSID와 KEY를 설정해준 뒤, 암호화 방식을 WPA2로 설정해준다



```
COM3 (Arduino Uno)
----- Jsn270 NTP Client -----
CMD: at+ver
H/W=JSN270
F/W Version=27.68
S2W Version=1.2.1-arduino
[OK]
CMD: at+ndhcp=1
CMD: at+wauto=0,JMPSYS
CMD: at+wwpa=jmpsys1234
CMD: at+wstart
WiFi connect to JMPSYS
CMD: at+wstat
MODE=0,CHANNEL=5,SSID=JMPSYS,BSSID=90:9F:33:16:05:0C,SECURITY=2
[OK]
CMD: at+nstat
DHCP=1
IP=192.168.0.6
Netmask=255.255.255.0
Gateway=192.168.0.1
DNS=168.126.63.1
[OK]
CMD: at+nslookup=2.kr.pool.ntp.org
NTP Server IP is 211.233.40.78
CMD: at+nauto=0,0,211.233.40.78,123
CMD: at+socket
Socket connect to 211.233.40.78
CMD: at+exit
The UTC+9 time is 15:10:25
CMD: at+exit
The UTC+9 time is 15:10:35
CMD: at+exit
The UTC+9 time is 15:10:46
Autoscroll No line ending 9600 baud
```

< 시리얼 모니터 창 결과 >

5.6 Simple Web Server



```
jsn270_Simple_Web_Server $
#include <Debug.h>
#include <JSN270.h>
#include <Arduino.h>
#include <SoftwareSerial.h>

#define SSID      "JMPSYS"           // your wifi network SSID
#define KEY       "jmpsys1234"       // your wifi network password
#define AUTH      "WPA2"            // your wifi network security

#define USE_DHCP_IP 1

#if !USE_DHCP_IP
#define MY_IP      "192.168.1.133"
#define SUBNET     "255.255.255.0"
#define GATEWAY    "192.168.1.254"
#endif

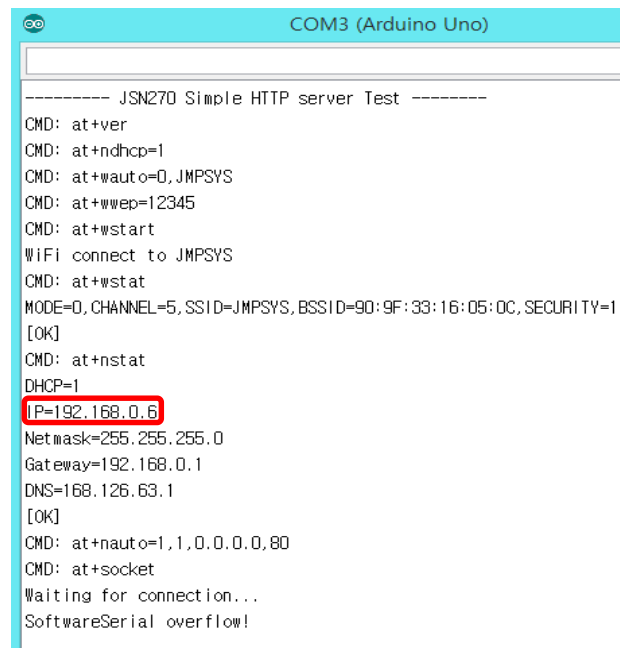
#define SERVER_PORT 80
#define PROTOCOL   "TCP"
```

File – Examples – JSN270 - Simple Web server 에서 예제를 불러온다.

AP의 SSID와 KEY를 설정해준 뒤, 암호화 방식을 WPA2로 설정해준다.

```
#define SERVER_PORT 80
#define PROTOCOL    "TCP"
```

Server의 포트를 80으로 설정하였다.

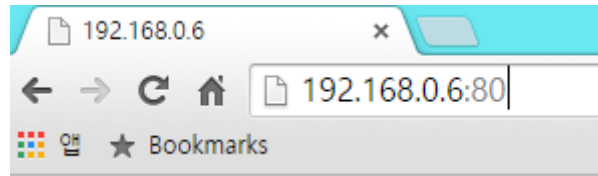


```
COM3 (Arduino Uno)

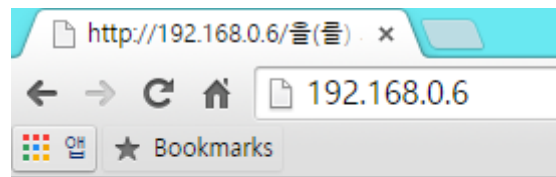
----- JSN270 Simple HTTP server Test -----
CMD: at+ver
CMD: at+ndhcp=1
CMD: at+wauto=0,JMPSYS
CMD: at+wwep=12345
CMD: at+wstart
WiFi connect to JMPSYS
CMD: at+wstat
MODE=0, CHANNEL=5, SSID=JMPSYS, BSSID=90:9F:33:16:05:0C, SECURITY=1
[OK]
CMD: at+nstat
DHCP=1
IP=192.168.0.6
Netmask=255.255.255.0
Gateway=192.168.0.1
DNS=168.126.63.1
[OK]
CMD: at+nauto=1,1,0.0.0.0,80
CMD: at+socket
Waiting for connection...
SoftwareSerial overflow!
```

< 시리얼 모니터 창 결과 >

JSN270의 IP를 이용하여 웹 서버에 접속한다.

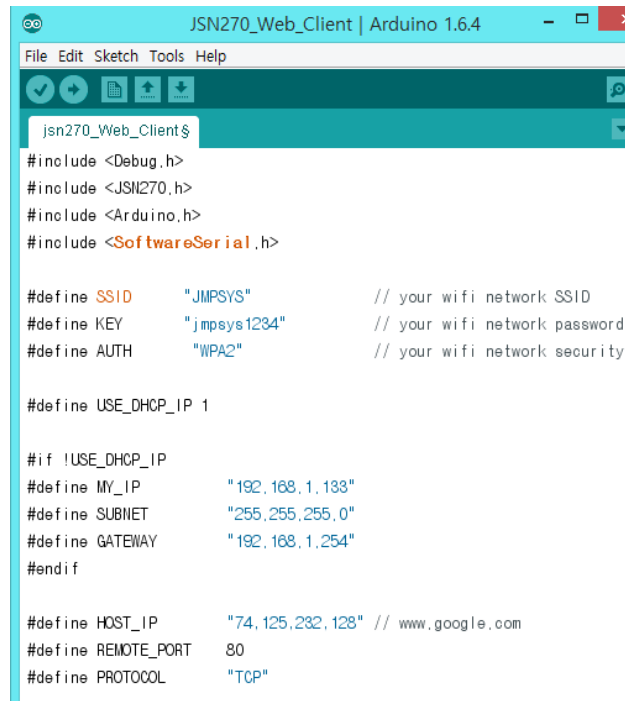


* 192.168.0.6 : 80 (IP : 포트)



접속하면 LED를 제어 할 수 있는 창이 뜬다.

5.7 Web Client



```
jsn270_Web_Client$
#include <Debug.h>
#include <JSN270.h>
#include <Arduino.h>
#include <SoftwareSerial.h>

#define SSID "JMPSYS" // your wifi network SSID
#define KEY "jimpsys1234" // your wifi network password
#define AUTH "WPA2" // your wifi network security

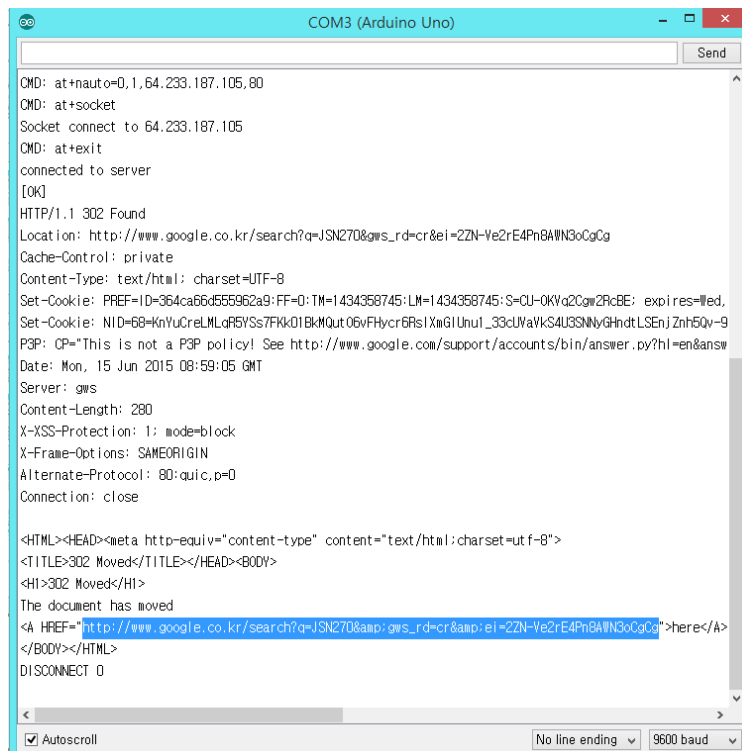
#define USE_DHCP_IP 1

#if !USE_DHCP_IP
#define MY_IP "192.168.1.133"
#define SUBNET "255.255.255.0"
#define GATEWAY "192.168.1.254"
#else
#define HOST_IP "74.125.232.128" // www.google.com
#define REMOTE_PORT 80
#define PROTOCOL "TCP"
#endif
```

File – Examples – JSN270 - Web Client 에서 예제를 불러온다.

AP의 SSID와 KEY를 설정해준 뒤, 암호화 방식을 WPA2로 설정해준다.

google.com에서 JSN270을 검색 하도록 설정해준다.



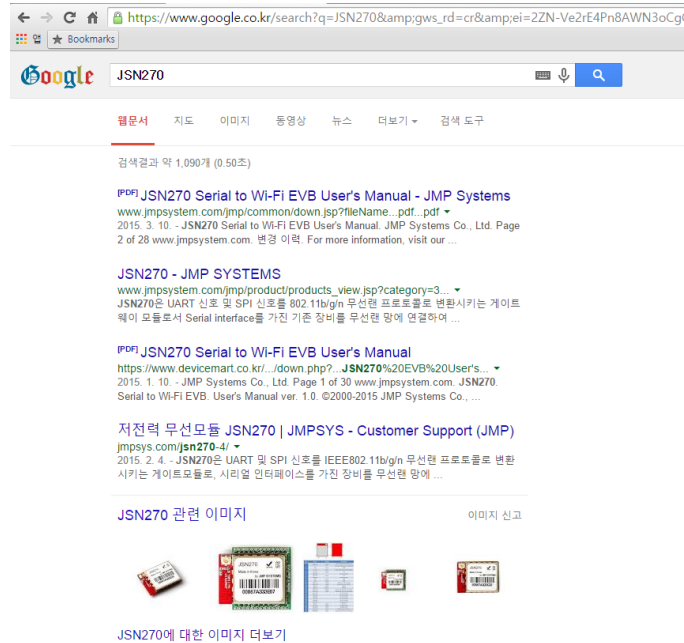
```
COM3 (Arduino Uno)
Send
CMD: at+nauto=0,1,64.233.187.105,80
CMD: at+socket
Socket connect to 64.233.187.105
CMD: at+exit
connected to server
[OK]
HTTP/1.1 302 Found
Location: http://www.google.co.kr/search?q=JSN270&gws_rd=cr&ei=2ZN-Ve2rE4Pn8A3oCgOg
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Set-Cookie: PREF=ID=364ca86d55952a9:FF=0:TM=1434358745:LM=1434358745:S=CU-OKVq2Cgw2RcBE: expires=Wed,
Set-Cookie: NID=68=KnYuCreLMLqR5YSs7FKk01BkMQu05vFHycr6RslXmGIUnu1_33dUvAykS4USNNyGHndtLSEnjZnh5Qv-9
P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bin/answer.py?hl=en&answ
Date: Mon, 15 Jun 2015 08:59:05 GMT
Server: gws
Content-Length: 280
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Alternate-Protocol: 80: quic, p=0
Connection: close

<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.kr/search?q=JSN270&gws_rd=cr&ei=2ZN-Ve2rE4Pn8A3oCgOg">here</A>
</BODY></HTML>
DISCONNECT 0
Autoscroll No line ending 9600 baud
```


< 시리얼 모니터 창 결과 >

* google.com에서 JSN270을 검색한 결과의 주소가 나왔다.

그 주소를 복사하여 접속해보면 JSN270의 결과가 나타난다.



<JSN270검색 결과>