

TWM0X ESP8266 모듈 사용 설명서

24/11/2015
Written by Steve AHN

1. TWM0X 모듈 소개

ESP8266칩에 기초한 Wifi 모듈들은 꽤 인상적입니다. 저가의 Wifi 모듈로써 기존의 모든 마이크로 기반의 프로젝트에 UART 통신만으로 간단히 Wifi 기능들을 추가할 수 있기 때문입니다. 심지어 모듈을 다시 리프로그래밍 하기 위한 다양한 방법이 제공되기 때문에 기능이 개선된 새로운 버전이 나올 때 마다 신규 버전으로 업그레이드가 용이하며, ESP8266칩을 이용하여 만든 많은 다른 모듈들이 다른 여러개의 펌웨어들을 수정없이 사용할 수 있습니다.

또한 사용자 임의로 자기만의 Wifi 모듈을 설계할 수 있도록 오픈 소스의 개발자 환경을 제공하고 있으므로 원하는 형태의 커스터마이징이 매우 쉽다는 장점도 있습니다. 본 문서에서는 간단한 AT 명령을 이용한 모듈 사용법을 소개합니다. Wifi 모듈을 사용하기 위해 사용자는 단순히 전원을 연결하기만 하면 됩니다.

제공하는 기능들은 꽤 흥미로우며 다음 기능들을 기본적으로 포함합니다.

- 802.11 b/g/n 프로토콜 지원
- Wifi direct (P2P) 기능지원, SoftAP
- 내장된 TCP / IP 스택

본 문서는 처음 Wifi 모듈을 사용하는 사용자들을 위해 모듈을 쉽게 사용하는 것을 돕기 위한 빠른 사용자 설명서로써 작성되었습니다.

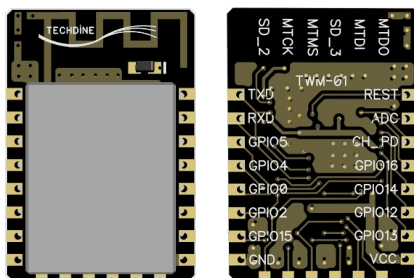
2. 모듈 하드웨어 연결

2-1. 브레드 보드를 이용한 연결

TWM0X Wifi 모듈의 연결은 아주 간단하지만 모듈의 연결전에 몇가지 주의 사항이 필요합니다.

- TWM0X 모듈은 3.3V 전원으로 동작하기 때문에 전원 공급시 주의를 요합니다 만약 실수로 5V, 또는 그 이상의 전원을 연결하면 모듈을 태워 버리게 됩니다.
- TWM0X 모듈은 3.3V로 통신을 하게되는데, 입력에 5V 전압 레벨의 통신 입력을 허용 하지 않습니다. 따라서 아두이노나 기타 다른 마이크로와 통신시 반드시 3.3V로 변환하여 통신하여야 합니다.
- 5V로 통신할 경우 통신은 가능합니다만, 칩의 안정성은 장담할 수 없으므로 사용자 리스크를 감당하고 사용하셔도 됩니다만 절대 권장 하지 않습니다.

다음은 TWM0X 모듈의 종류 및 연결 관련 내용을 살펴보도록 하겠습니다. 아래 [그림 1]은 모듈의 종류 및 그 핀 구성을 나타낸 그림입니다.



전원(3.3V)가 모듈에 공급되면, 보드 전면 상단에 있는 LED가 잠깐동안 깜박이는 것을 볼 수 있습니다.

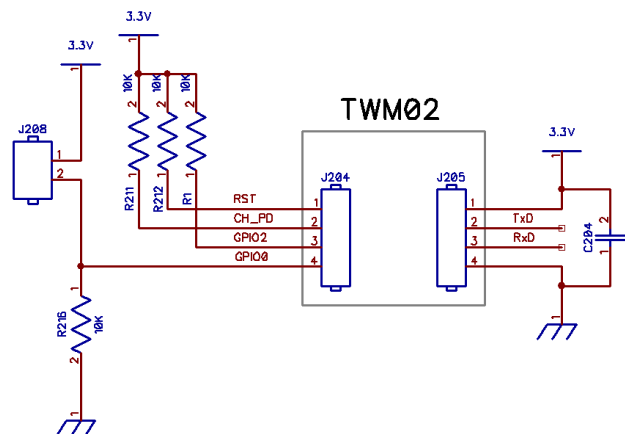
위 [그림 1]의 TWM02 모듈은 총 8핀으로 구성되어 있으며, 이제 이모듈을 사용하기 위한 연결을 살펴 보겠습니다.

핀번호	명칭	설명
1	RESET	Reset / 3.3V / Pull-up
2	CH_PD	3.3V / Pull-up
3	GPIO2	General Port I/O 2 (일반 I/O 포트 2) / 3.3V / Pull-up
4	GPIO0	General Port I/O 0 (일반 I/O 포트 0) 또는 펌웨어 업데이트 핀 - 0 : 업데이트 모드, 1 : 일반 모드
5	GND	전원 입력 0V
6	RxD	통신 단자
7	TxD	통신 단자
8	VCC	전원 입력 3.3V

[표 1] TWM02 모듈의 핀

위 [표 1]에 나타내었듯이 모듈을 사용하려면 전원 외에도 통신을 위한 통신 단자 TxD, RxD를 연결해야 하며 그 외의 나머지 모든 핀들은 3.3V에 Pull-up 시켜야 합니다.

또한 일반 모드로 사용할 때에는 GPIO0에 꼭 풀업을 연결해야 합니다. 칩이 리셋되거나 전원이 들어오면, 칩은 각포트의 Pull-up, Pull-down 상태에 따라, 동작 모드를 결정하게 되는데, GPIO0가 0V면 펌웨어 업데이트 기능이 동작하고, GPIO0가 3.3V면 일반 모드로 동작하기 때문입니다. 단, 펌웨어 업데이트 모드로 동작하려면, GPIO0 외에 나머지 핀들은 반드시 3.3V에 Pul-up이 되어야만 펌웨어 업데이트 모드로 진입됩니다. 따라서 전원과 통신단자 연결외, 나머지 핀들은 항상 풀업 시키고, GPIO0만 가지고 펌웨어 업데이트 모드인지, 일반 모드인지 조절 하는것이 좋습니다.

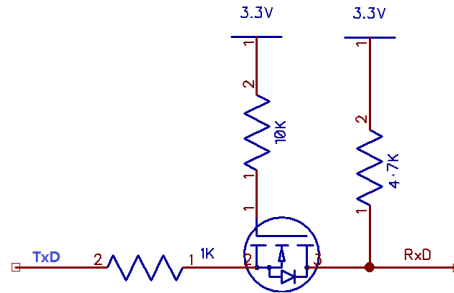


[그림 2] 모듈 연결도

위 [그림 2]에 간단한 연결도를 그렸습니다. TxD, RxD만 이제 연결하면 되는데... USB to Serial 젠더를 사용해 간단히 연결해도 되지만 이 때 주의 할 점이 있습니다. 이미 말씀 드렸듯이 모듈이 3.3V만 지원하기 때문에 TxD 단자는 상관 없지만, RxD 단자는 반드시 3.3 V로 통신해야 합니다.

따라서 일반 빵판(bread board)에 모듈을 꽂아서, FTDI 젠더칩이나, CP210X 젠더 칩을 이용해서 통신할 경우, 젠더칩의 TxD를 모듈의 RxD에, 젠더칩의 RxD를 모듈의 TxD에 연결하면 되는데요... 일단 TxD를 모듈의 RxD에 연결할때 전압 변경을 위해서 약간의 조치를 취해야 합니다.

가장 간단한 방법은 NMOS를 이용해서 레벨 쉬프트를 하는 방법으로 제가 사용한 방법은 아래와 같습니다.



[그림 3] RxD 레벨 쉬프트

젠더칩의 TxD 단자를 모듈의 RxD에 연결할 때 NMOS를 이용해서 3.3V로 레벨을 변화시켜 입력 시키면 간단히 해결 됩니다.

2-2. 아두이노와의 연결

아두이노에 Wifi 모듈을 연결하여 사용하려고 하면 몇 가지 문제가 있습니다. 따라서 Wifi 모듈을 절대 아두이노에 바로연결해서는 안됩니다. 그 이유를 살펴 보겠습니다.

1. ESP8266칩은 5V 동작을 지원하지 않으므로, 전원을 바로 연결하면 Wifi 모듈의 칩이 파괴됩니다.
2. ESP8266칩이 동작중에 아두이노의 3.3V 레귤레이터가 공급해주는 전류보다 많은 전류를 소비할 수 있습니다. 이 경우 아두이노 보드가 손상을 입게 됩니다.
3. ESP8266칩의 스펙에서 명기하는 최대점을 초과하여 동작시키면, 칩동작의 불안이나 소자의 파괴가 나타날 수 있습니다.

그럼에도 불구하고 아두이노를 이용해서 ESP8266 칩을 제어하면 아주 좋은 점은 제어가 너무 쉽다는 점이지요. 따라서 아두이노로 제어할경우 위예거 검토한 사항을 고려하여 사용하여야 합니다.

1. 아두이노에 3.3V 레귤레이터가 장착되어 있는지 여부와 있다면 충분한 전류를 공급 가능한지 확인해 보아야 합니다.
2. 아두이노의 전류 공급이 충분하다면 아두이노에서 공급하는 3.3V를 직접 Wifi 모듈에 인가해도 됩니다.
3. 아두이노의 통신 단자 TxD가 5V로 동작하므로 위에서 제시한 방법을 이용하여 3.3V로 변환한 다음에 모듈의 RxD에 연결 합니다.

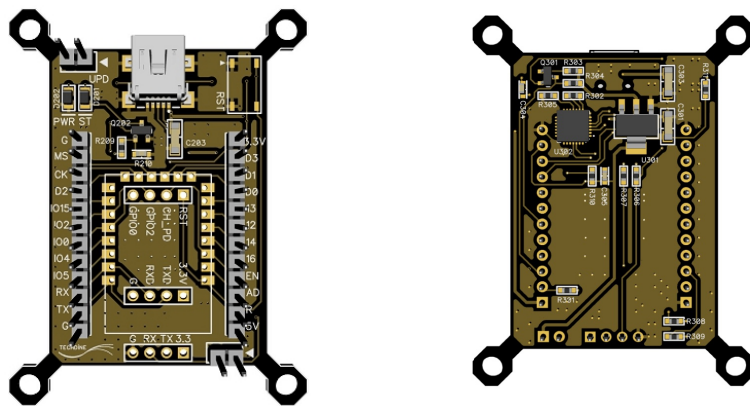
이 정도만 고려하면 바로 아두이노를 이용하여 모듈을 사용할 수 있습니다.

2-3. 개발 평가보드 (EVB)를 이용한 연결

이번에 소개할 방법은 가장 안전하고 편리한 방법으로, TWM0X모듈에 맞게 제작된 평가보드 TWM-EVB01을 이용한 방법입니다. 평가보드는 다음과 같은 특징이 있습니다.

- EVB 자체에 USB to Serial 젠더 내장
- EVB 자체에 3.3V 1A 공급가능한 레귤레이터 내장
- Wifi 모듈 TWM01과 TWM02를 바로 사용할수 있는 인터페이스 내장
- 아두이노와 바로 연결할 수 있는 전용 연결 인터페이스 내장
- 통신 단자 레벨 쉬프트 내장
- 펌웨어 업그레이드용 회로 내장
- 리셋 스위치 및 관련기능 내장

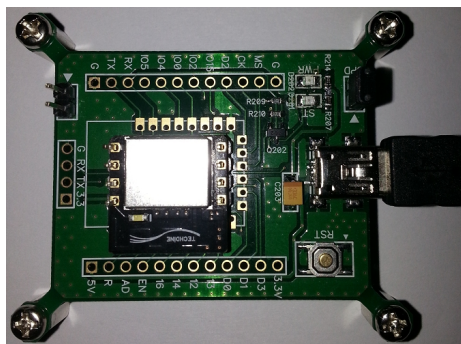
다음은 [그림 4]는 평가보드 TWM-EVB-01 입니다.



[그림 4] 개발 평가보드 TWM-EVB-01

개발 평가보드 TWM-EVB-01에는 TWM01, TWM02 모듈을 직접 연결할 수 있습니다. 이 중에서 TWM01 모듈의 경우에는 오픈소스 ESP8266 모듈들과 핀맵이 완전히 호환 되며, 따라서 현존하는 모든 ESP8266 모듈을 수용할 수 있도록 디자인 되었습니다. 즉, 오픈소스 프로젝트 모듈인 기존 ESP8266을 직접 평가보드에 연결하여 기능 테스트 및 개발에 이용할 수 있도록 도와줍니다.

이제 하드웨어 연결이 완료 되었으면 본격적으로 모듈기능을 하나하나 살펴 볼 차례입니다.



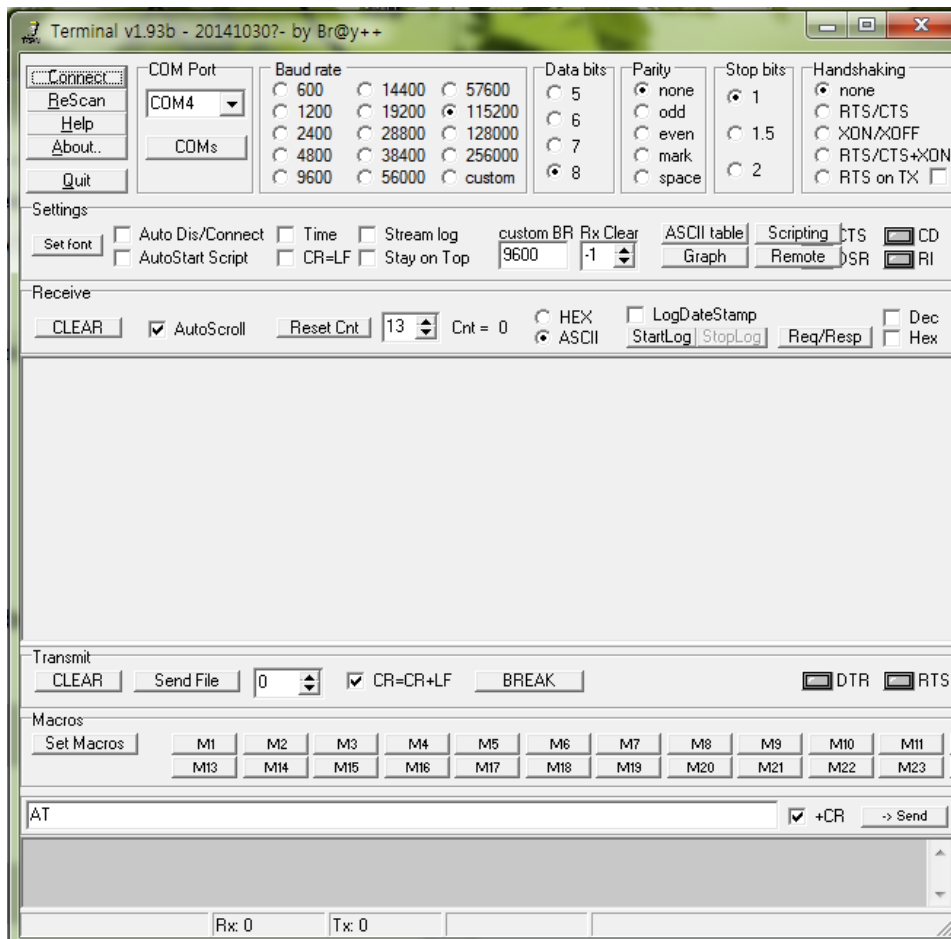
[그림 5] 평가보드 TWM-EVB01에 TWM02 모듈 연결

위에 제시된 여러 방법 중 한 가지를 선택해 모듈을 연결하게 되면, 이제 사용 준비는 다 끝난 것입니다. 요약하면 전원3.3V를 공급하고 시리얼 통신만 되도록 연결하면 됩니다.

3. 시리얼 통신을 이용한 제어

하드웨어의 연결이 끝나면 이제 시리얼 통신을 통해서 본격적으로 모듈을 제어 할 수 있습니다. 시리얼 통신은 간단한 시리얼 터미널 프로그램을 이용하면 됩니다. 아두이노 시리얼 모니터를 이용해서 제어 할 수도 있고, 별도의 시리얼 통신 터미널을 이용하셔도 됩니다.

우선 통신 포트 및 통신 속도 데이터 비트 등을 설정해 줍니다. 통신 포트는 각자 사용하는 젠더에 따라 지정된 포트 번호를 이용하면 되고 통신속도는 **115,200 bps** 로 설정하고 8데이터 비트에 **No Parity** 그리고 **1Stop** 비트로 설정한다. 이때 매 패킷마다 뉴라인과 캐리지 리턴 옵션을 주어야 합니다.



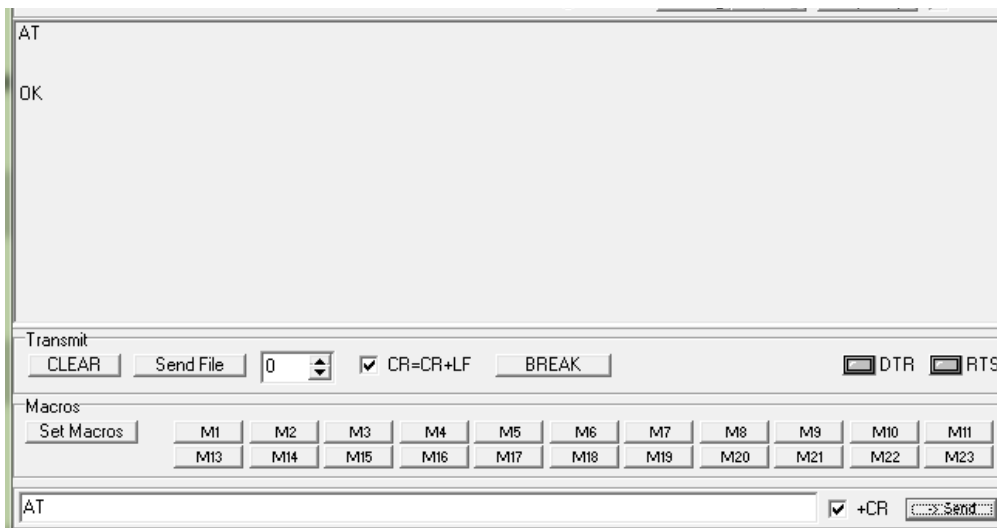
[그림 6] 시리얼 통신을 이용한 연결

위 [그림 6]은 연결에 필요한 설정 값을 적용한 예시입니다. 통신포트 번호는 개인의 컴퓨터마다 다르게 나타나므로 본인의 컴퓨터에 설정된 통신포트를 이용해서 접속하시면 됩니다. 이제부터 사용할 시리얼 통신 명령은 별도로 작성된 **AT명령어 요약집**을 참고하시기 바랍니다. 링크는 다음과 같습니다.

[AT 명령어 요약집 링크](#)

3-1. 첫번째 명령

이제 **AT** 명령을 보내보겠습니다. **AT** 명령이 제대로 입력되면 **OK** 라는 응답을 받게 되는데, 이때 주의할 점은 **AT** 라는 명령어는 반드시 대문자여야 합니다. Wifi모듈이 **Case-Sensitive** 형태이므로 대/소 문자에 유의하여 주시기 바랍니다.



[그림 7] AT 명령과 그 응답

만일 응답이 오지 않는다면 다음 사항을 검사해 보시길 바랍니다.

- 하드웨어 연결(흰색 LED의 깜빡이는 횟수를 확인)과 RxD, TxD 가 바뀌지 않았는지 확인합니다.
- 보드레이트가 확실히 115,200bps 인지 확인합니다. 펌웨어의 기본값은 항상 115,200bps 입니다.
- 명령의 끝에 위 그림과 같이 CR+LF 를 붙여서 명령을 전송하였는지 확인합니다.

위 3가지 사항이 확실하게 연결 되었다면, 제대로 응답을 받을 수 있습니다. 그래도 동작하지 않는다면, USB to Serial 젠더 드라이버의 오류나 USB 케이블 불량 등 외부 요소의 문제를 검토해 보아야 합니다.

3-2. 모듈 리셋 하기

모듈의 응답이 정상적으로 된다면, 이제 모듈을 리셋하여 보겠습니다.

AT+RST

응답:

```
ets Jan 8 2013,rst cause:1, boot mode:(3,7)

load 0x40100000, len 1396, room 16
tail 4
chksum 0x89
load 0x3ffe8000, len 776, room 4
tail 4
chksum 0xe8
load 0x3ffe8308, len 540, room 4
tail 8
chksum 0xc0
csum 0xc0

2nd boot version : 1.4(b1)
SPI Speed      : 40MHz
SPI Mode       : QIO
SPI Flash Size & Map: 32Mbit(1024KB+1024KB)
```

```

jump to run user1 @ 1000

rlr

techdine.co.ltd.
http://www.techdine.com
ready

```

모듈마다 차이는 있겠지만 위와 비슷한 응답 결과를 얻을 수 있습니다. 중요한 것은 가장 마지막의 **ready** 라는 응답입니다.

3-3. 펌웨어 버전 확인 하기

```
AT+GMR
```

응답 :

```

AT version:0.30.0.0(Jul 3 2015 19:35:49)
SDK version:1.2.0
compile time:Oct 5 2015 13:10:02
OK

```

3-4. 모듈을 **Station + softAP** 둘 다 가능하게 설정하기

```
AT+CWMODE=3
```

응답:

```
OK
```

모드를 변경한 후에는 일부 오래된 모델은 반드시 리셋을 실행하여야 합니다.

3-5. 주변의 연결 가능한 **Wifi** 네트워크 검색하기

```
AT+CWLAP
```

응답 :

```

+CWLAP:(4,"AP-B-901",-65,"e8:54:84:02:10:45",1,133)
+CWLAP:(4,"AP-B-801",-78,"e8:54:84:01:8c:39",3,108)
+CWLAP:(4,"AP-B-902",-90,"e8:54:84:02:0f:c1",3,113)
+CWLAP:(4,"techdine",-59,"90:9f:33:19:5b:be",8,41)
+CWLAP:(3,"HP-Print-70-Photosmart 5520",-85,"64:51:06:e0:a3:70",5,50)
+CWLAP:(4,"AP-B-1001",-89,"e8:54:84:01:83:3f",9,113)
+CWLAP:(2,"JL-N704bcm",-94,"64:e5:99:70:90:de",13,55)

OK

```

3-6. 와이파이 (**Wifi**) 액세스 포인트에 접속하기

와이파이에 접속할 때에는 다음의 명령을 사용합니다. < > 안에는 접속하기 위한 액세스 포인트 이름과 비밀번호를 입력하시면 됩니다.

```
AT+CWJAP="<access point name>","<password>"
```


예를 들어 3-5의 리스트 중에서 **techdine** 에 접속하기 위한 명령은 다음과 같습니다.

```
AT+CWJAP="techdine","MyNameisMyPW"
```

응답 :

```
WIFI CONNECTED
WIFI GOT IP

OK
```

일단 접속이 확인 되었으면, 내 접속 정보를 조회해 볼 수 있습니다.

```
AT+CIFSR
```

응답 :

```
+CIFSR:APIP,"192.168.4.1"
+CIFSR:APMAC,"1a:fe:34:f5:01:ea"
+CIFSR:STAIP,"192.168.1.124"
+CIFSR:STAMAC,"18:fe:34:f5:01:ea"

OK
```

접속 정보를 조회하여 보면 응답 결과에서 2개의 IP 주소를 확인할 수 있습니다. 이것은 3-4에서 모듈을 **Station + softAP** 모드로 설정했기 때문인데 **APIP**는 액세스 포인트로 동작할 때의 IP 주소이며, **STAIP**는 **Station** 모드로 동작할 때의 IP 주소 입니다. 마찬가지로 각각의 아이피에 대한 **MAC** 주소도 개별적으로 할당 되어 있습니다.

예제에서의 서버 IP는 **192.168.4.1** 이며, 클라이언트 IP 주소는 **192.168.1.124** 임을 알 수 있습니다.

3-7. TCP 클라이언트로 동작 시키기

TCP 클라이언트 모드로 바로 인터넷에 접속할 수 있습니다. 방법은 다음과 같습니다.

1. 다중 접속 모드 설정하기

```
AT + CIPMUX=1
```

```
OK
```

2. 다중 접속 모드가 활성화 되었으면, 접속할 채널(0~4)과 프로토콜 타입(TCP/UDP), 그리고 IP주소 (또는 도메인 주소) 및 포트 번호를 지정하여 **CIPSTART** 명령을 통해 접속할 수 있습니다. 구글에 한 번 접속해 보도록 하겠습니다.

```
AT+CIPSTART=4,"TCP","google.com",80
```

응답 :

```
4,CONNECT

OK
```

3. 이제 보낼 데이터를 계산 하여야 하는데, 실례로 보낼 데이터는 **GET/HTTP/1.0\r\n\r\n** 입니다. 데이터는 총 18 바이트가 됩니다. 뉴라인 명령과 캐리지 리턴 명령을 두 번 전송하는 이유는, HTTP에서 마지막에 빈 라인의 입력을 한 번 더 요구하기 때문입니다.

이제 위에서 설정한 4번 채널로 18바이트를 전송하기 위해 다음의 명령어를 입력합니다.

```
AT+CIPSEND=4,18
```

모듈은 **OK** 응답 대신에 **>** 형태의 입력 **Prompt**를 응답으로 보냅니다.

```
>
```

이 **Prompt**는 이제부터 보낼 18바이트의 데이터를 입력하라는 의미입니다.
데이터를 전송해 보겠습니다.

```
GET/HTTP/1.0
```

전송이 끝나면 모듈은 발송이 완료 되었다고 응답합니다.

```
Recv 18 bytes
```

```
SEND OK
```

위 메시지 뒤에 공백의 라인을 보내기 위해서 `\n\r`을 한번 더 보냈습니다.

이제 모듈은 웹서버로부터 응답을 기다리는데 응답이 들어오면 그것을 시리얼 통신을 통해 보내줍니다.

```
+IPD,4,494:HTTP/1.0 302 Found
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Location: http://www.google.co.kr/?gfe_rd=cr&ei=pblSVuGqKYqT8Qfy3YOoAg
Content-Length: 261
Date: Mon, 23 Nov 2015 07:00:53 GMT
Server: GFE/2.0

<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/?
gfe_rd=cr&ei=pblSVuGqKYqT8Qfy3YOoAg">here</A>.
</BODY></HTML>
4,CLOSED
```

응답의 의미를 살펴보도록 하겠습니다.

```
+IPD,4,494:HTTP/1.0 302 Found
Cache-Control: private
Content-Type: text/html; charset=UTF-8
```

+IPD는 모듈이 외부로 부터 데이터를 수신하였음을 의미하고, 숫자 **4**는 채널 번호를 의미하며, **494**는 494 바이트의 데이터를 수신 하였음을 의미합니다. **494:** 이후는 실제 수신한 데이터가 되는 것입니다.

이후 데이터는 정확히 494 바이트의 데이터를 받게되고, 마지막 라인에서는 데이터를 수신한 후 모듈과 서버와의 접속이 끊어졌음을 알려줍니다.

```
4,CLOSED
```

3-8. TCP 서버로 동작시키기

모듈을 TCP 서버로 동작 시키려면, 다음과 같이 몇 가지의 작업이 필요합니다.

1. 모듈을 액세스 포인트에 연결하기
2. 다중 접속이 가능도록 설정

```
AT+CIPMUX=1
```

3. 모듈의 IP 주소를 확인

```
AT+CIFSR
```

응답 :

```
+CIFSR:APIP,"192.168.4.1"
+CIFSR:APMAC,"1a:fe:34:f5:06:9a"
+CIFSR:STAIP,"192.168.1.126"
+CIFSR:STAMAC,"18:fe:34:f5:06:9a"

OK
```

위 주소중 station IP주소인 192.168.1.126을 기억해 두시기 바랍니다.

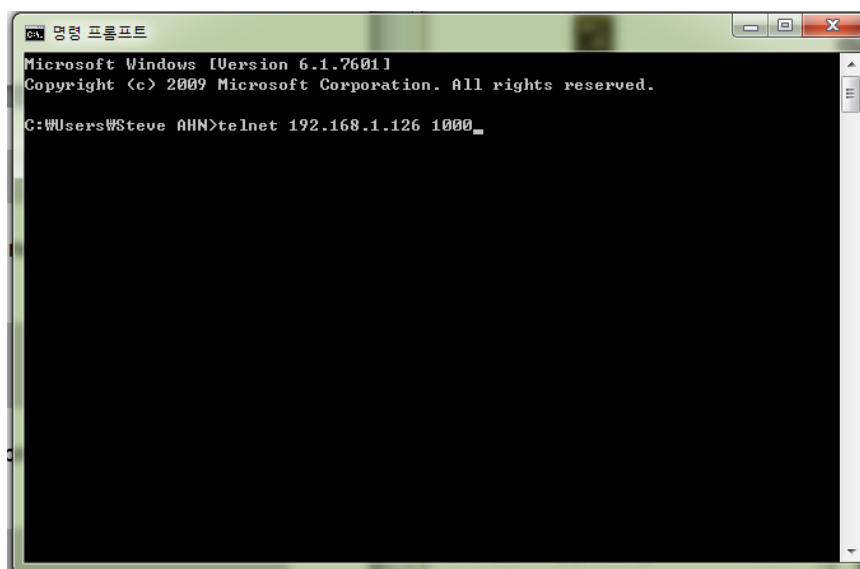
4. 모듈이 서버로 동작하기 위해선 특정 포트를 지정해서 계속 감시하는 상태를 만들어야 합니다. 다음 명령으로 서버를 지정할수 있습니다.

```
AT+CIPSERVER=1,1000
```

첫번째 파라미터인 1은 모드를 의미하는데, 서버를 시작한다는 의미 입니다. 1000은 서버가 리스닝을 할 포트 번호를 의미합니다. 서버 모드가 시작되면 **OK** 응답을 받을 수 있으며, 이로써 서버를 테스트할 준비가 되었습니다.

서버에 접속하기 위해서 도스창을 하나 엽니다. 그리고 다음과 같이 텔넷 접속을 시도합니다.

```
telnet 192.168.1.126 1000
```



[그림8] 텔넷으로 접속

접속하게 되면 다음과 같이 접속된 채널의 번호와 상태가 표시됩니다.

```
0,CONNECT
```

이제 텔넷 창에 글자를 입력해 보겠습니다.

```
+IPD,0,2:ab
+IPD,0,1:c
+IPD,0,1:d
+IPD,0,1:h
+IPD,0,11:ello world!
```

입력이 들어올때마다 +IPD 다음에 전송된 채널 번호와 함께 글자의 갯수 그리고 실제 입력된 글자를 확인할 수 있습니다.

이번에는 반대로 서버에서 텔넷 창으로 메시지를 보내보겠습니다. 위의 3-7에서 한 번 살펴본 CIPSEND 명령을 이용하면 됩니다. 현재 채널은 0번이고 8바이트를 송신해 보도록 하겠습니다.

```
AT+CIPSEND=0,8
```

응답:

```
>
```

입력 대기 상태가 되지요? Hello hi 라고 입력해 보겠습니다.

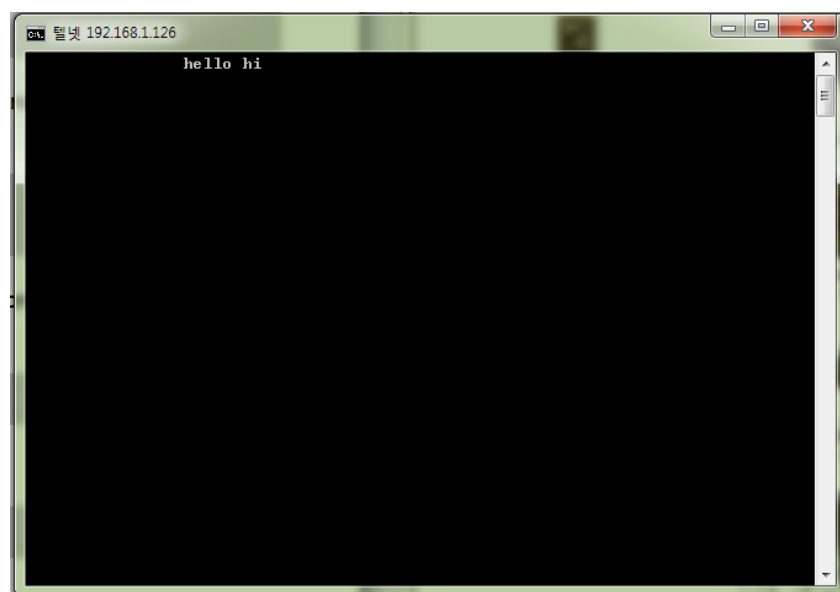
```
hello hi
```

응답 :

```
Recv 8 bytes
```

```
SEND OK
```

설정한 8바이트가 입력되자마자 전송을 하고 전송완료 되었다고 메시지를 보냅니다. 이때 텔넷 도스창엔 전송된 메시지가 보입니다.



[그림 9] 전송된 메시지

텔넷 전송을 끝내려면, 텔넷 창을 닫으면 됩니다. 그러면 텔넷이 접속이 끊겼다는 메시지를 확인할 수 있습니다.

0,CLOSED

3-9, 와이파이(Wifi) 액세스 포인트로 동작하기

지금까지 주변에 있는 액세스 포인트에 연결해서 **TWX02** 모듈을 서버와 클라이언트로 동작시키는 것을 살펴보았습니다. 이번에는 모듈 자체가 스스로 인터넷 공유기가 되는것을 실험해 보도록 하겠습니다. 이 말은, 모듈끼리 다른 액세스 포인트 없이 모듈간의 직접 접속이 가능하다는 의미와 같습니다. 즉 센서 네트워크나, 혹은 로컬 사설 공유 시스템 구축 등이 가능해 집니다.

모듈은 처음에 이미 지정된 액세스 포인트 이름을 가지고 있는데, (**TWM01XXX**, 또는 **ESP_...**등) **AT** 명령을 이용하여 직접 공유기 이름을 설정할 수 있습니다.

AT+CWSAP="MyAPSSID","12345678",2,0

공유기 이름을 "**MyAPSSID**"로, 패스워드를 "**12345678**" 로 채널 ID를 2 번으로, 마지막으로 보안 설정은 0번, 즉 오픈으로 설정 합니다.

액세스 포인트 (공유기 모드)를 설정하려면 **CWMODE** 명령으로 모듈의 작동 모드를 2번, 또는 3번으로 설정해서 **softAP** 모드를 사용할 수 있는 상태에서만 가능하므로 다음과 같이 미리 설정 해 두어야 합니다.

AT+CWMODE=3

이제 액세스 포인트를 확인해 보도록 하겠습니다

OS의 "네트워크 및 공유센터 열기"를 보면 네트워크 목록에서 우리의 공유기를 확인 할 수 있습니다.



[그림 10] 액세스 포인트 확인

방금 전에 우리가 설정했던 **MyAPSSID**가 활성화 되어 있고, 보안은 해제되어 일반 접속이 가능한 상태를 확인할 수 있습니다.

이제 우리의 공유기인 **MyAPSSID** 에 연결해 보도록 하겠습니다. 모듈끼리의 **P2P** 접속을 위해서 모듈 하나를 더 연결하도록 하겠습니다.

두 번째 모듈은 **station** 모드로 동작시키기 위해 설정을 변경합니다.

```
AT+CWMODE =1
```

그리고 주변에 연결 가능한 **AP**를 검색해 봅니다.

```
AT+CWLAP
```

응답에 우리가 설정한 **AP**인 **MyAPSSID** 가 보이는군요..

```
+CWLAP:(4,"AP-B-901",-70,"e8:54:84:02:10:45",1,0)
+CWLAP:(4,"AP-B-801",-84,"e8:54:84:01:8c:39",3,0)
+CWLAP:(3,"HP-Print-70-Photosmart 5520",-89,"64:51:06:e0:a3:70",5,0)
+CWLAP:(3,"enews",-70,"64:e5:99:71:7b:de",5,0)
+CWLAP:(0,"MyAPSSID",-24,"1a:fe:34:f5:06:9a",8,0)
+CWLAP:(4,"techdine",-56,"90:9f:33:19:5b:be",8,53)
+CWLAP:(4,"AP-B-1001",-89,"e8:54:84:01:83:3f",9,0)
```

```
OK
```

이제 우리가 설정한 **AP**에 접속해 보도록 하겠습니다.

```
AT+CWJAP="MyIPSSID","12345678"
```

다음과 같은 응답이 옵니다.

```
WIFI CONNECTED
WIFI GOT IP
```

```
OK
```

이제 얻어온 **IP** 주소를 확인해 봅니다.

```
+CIFSR:STAIP,"192.168.4.4"
+CIFSR:STAMAC,"18:fe:34:f8:66:70"
```

```
OK
```

192.168.4.4 라는 주소를 **DHCP**로 받았습니다. 동작 모드를 **station** 모드로 설정하였기 때문에 **station IP**만 보이는걸 알 수 있습니다.

이제 접속이 끝났으니 **AP** 쪽에서 접속된 클라이언트를 확인해 보겠습니다.

```
AT+CWLIF
```

응답은 다음과 같습니다.

```
192.168.4.4,18:fe:34:f8:66:70
```

```
OK
```

접속된 클라이언트는 **192.168.4.4** 의 주소를 가지고 그에대한 **MAC** 주소를 보여줍니다.

이제 두 개의 모듈간의 통신을 실습해 볼 차례입니다.

실습을 하기 위해 현재 상태를 정리해보면,

- 우선 모듈 1을 액세스 포인트(공유기)로 설정을 했고, 그 아이피 주소는 **192.168.4.1** 입니다.
- 모듈 2를 **station**(무선랜)으로 설정을 했고, 그 아이피 주소는 **192.168.4.4** 입니다.

서로간의 통신을 하려면 스테이션 주소를 가진 쪽이 서버가 되어야 됩니다. 따라서 무선 랜 모드 **station** 모듈이 서버가 되어 포트간의 연결을 만들어야 합니다

모듈 2에서 다음 순서로 설정 합니다.

```
AT+CIPMUX=1
OK
```

복수 연결을 허용하고, 300번 포트를 오픈하여 감시합니다.

```
AT+CIPSERVER=1,300
OK
```

이제 액세스 포인트 모듈 1에서 모듈 2에 접속합니다.

```
AT+CIPSTART=0,"TCP","192.168.4.4",300
0,CONNECT
OK
```

스테이션 모듈 **192.168.4.4**의 300번 포트로 TCP접속을 시도합니다. id는 0번으로 설정합니다. 설정과 명령에 이상이 없으면 연결되었다는 응답을 받을 수 있습니다.

정상적으로 접속이 되었으면 모듈 2 역시 접속이 성공했음을 다음과 같이 알려줍니다.

```
0,CONNECT
```

이제 서로 메시지를 발송해 보도록 하겠습니다.

3-9-1, 모듈 2에서 모듈 1로 메시지 보내기

모듈 2에서 메시지를 보냅니다. 연결된 채널 id 0번으로 5글자를 보냅니다.

```
AT+CIPSEND=0,5
OK
>
```

문자를 입력하라고 대기중입니다.
이제 **hello**를 전송합니다.

```
busy s...
Recv 5 bytes
SEND OK
```

모듈 1에선 방금 모듈 2로부터 받은 메시지가 도착하였습니다.

```
+IPD,0,5:hello
```

3-9-2, 이제 모듈 1에서 모듈 2로 답신 보내기

```
AT+CIPSEND=0,2
```

```
OK  
>
```

채널 id 0번으로 2글자를 보냅니다. 입력 대기중입니다.

```
Hi
```

라고 전송해 봅니다.

```
busy s...
```

```
Recv 2 bytes
```

```
SEND OK
```

채널 2에서 문자를 받았습니다.

```
+IPD,0,2:Hi
```

이제 통신을 종료합니다. 채널1에서 열린 ip 접속을 끊습니다.

```
AT+CIPCLOSE=5
```

id 5번은 모든 채널을 닫으라는 의미입니다.

```
0,CLOSED
```

```
OK
```

열려 있는 채널은 하나 이므로 0번만 채널만 닫혔습니다.

모듈 2번에서도 모듈 1번이 채널을 종료 하였음을 확인 합니다.

```
0,CLOSED
```

이로써 서버를 설정하고 채널을 설정해서, 서로 데이터를 주고받는것을 실험해 보았습니다.

지금까지의 과정을 아두이노 혹은 안드로이드에서 또는 일반 마이컴의 시리얼 통신 루틴에서 코딩으로 제어하면, 전용 IoT 사물인터넷이 구성됩니다.

차후에 마이컴(아두이노 등)에서 코딩을 통해서 제어하는 실습을 하도록 하겠습니다.

4. 펌웨어 (Firmware) 업데이트

이제 모듈의 펌웨어를 업데이트하는 방법을 알아보도록 하겠습니다. 펌웨어 업데이트는 다양한 방법이 있습니다만, 대표적인 두가지 방법을 알아보도록 하겠습니다.

4-1. 전용 툴을 이용한 다운로드

우선은 모듈에 들어가는 펌웨어의 종류를 살펴보면 약 5-6개의 파일로 구성되어 있으며, 그 파일들은 다음과 같습니다.

1, esp_init_date_default.bin	0x3fc000	기본 RF 파라미터 값이 들어 있는 파일 입니다.
2, blank.bin	0x3fe000	기본 시스템 파라미터 값이 들어있는 파일 입니다.
3, boot.bin	0x0000	부트 로더가 들어있는 파일 입니다.
4, user1.bin	0x1000	사용자가 만든 파일 입니다.
5, user2.bin	0x101000	사용자가 만든 파일 2 입니다. 보통은 초기에 라이팅 하지 않고, 와이파이를 통해서 다운로드 됩니다.

일반적으로 ESP8266 공개 프로젝트의 경우 펌웨어 다운로드는 위의 5번파일을 다운로드 하는 과정을 의미합니다. 본 문서에서는 위의 5개 파일을 직접 다운로드 하는 방법을 살펴보고, 그 외 다른 다운로드 방법을 알아보겠습니다.

다운로드 할 파일의 주소는 컴파일된 파일 및 메모리 크기에 따라 다른데, 본 문서에서 제공하는 파일 이미지는 32메가 메모리를 기준으로 작업되어 있고 user1, user2 파일의 크기가 1MB로 컴파일 된 것을 기준으로 정하였습니다.

처음 제작된 IoT 모듈은 비어있기 때문에 위에 언급한 모든 파일을 다운로드 해줘야 하는데, 펌웨어를 업데이트 하기 위해선 하드웨어적으로 몇가지 기본 셋팅이 필요합니다.

다음의 5가지 신호를 셋팅한 상태에서, 전원을 넣거나 리셋을 시키면, 칩 내부의 부트 과정에서 펌웨어 업데이트 모드로 진입하게 됩니다.

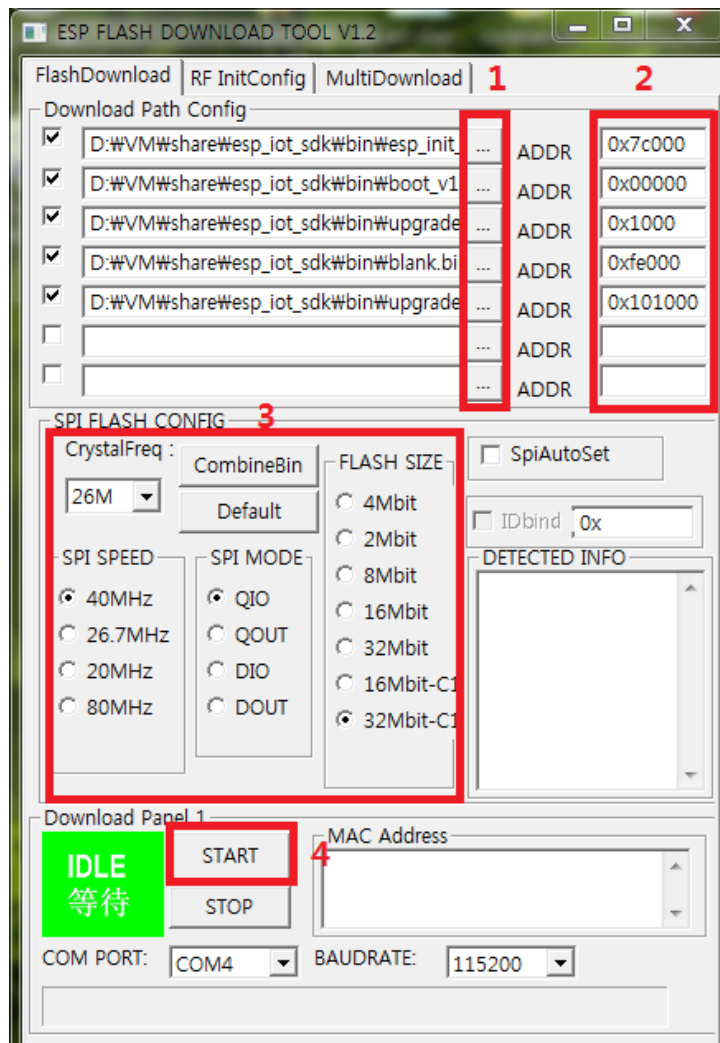
MTDO	0V
RST	3.3V
CH_PD	3.3V
GPIO2	3.3V
GPIO0	0V

이때 시리얼 통신을 통해 다운로드 하게 되어 있는데, ESP8266의 칩 제조사에서는 별도의 다운로드 프로그램을 제공하고 있습니다. 최신버전은 ESP FLASH DOWNLOAD TOOL V1.2로써, 이 프로그램을 이용하여 다운로드를 진행 합니다.

다음 [그림11]은 툴의 설정 순서를 보여줍니다.

1. 위에 나열한 5개의 파일들을 지정 합니다.
2. 각 파일이 다운로드될 주소를 정확히 입력합니다.
3. 크리스탈 26Mhz 를 설정하고, SPI Speed 40Mhz, SPI Mode, Flash Size 32M를 지정합니다.
4. 다운로드 START 버튼을 누릅니다.

파일이 전부 다운로드 되는데 약 10초정도의 시간이 걸립니다.



[그림 11] 다운로드 순서

이로써, 전용 다운로드 툴을 이용한 다운로드 방법을 살펴 보았습니다. 이 ESP Flash Download Tool은 Espressif의 웹사이트를 통하여 구할 수 있습니다.

다운로드에 필요한 이미지 파일은 테크다인 홈페이지 <http://www.techdine.com>의 자료실에서 내려받기 할 수 있습니다.

또한 무선모듈 자체가 공개 프로젝트로 표준화가 되어 있기 때문에 다양한 공개 프로젝트 이미지와도 100% 호환 됩니다.

Espressif사에서 제공하는 이미지는 웹사이트 다음의 링크에서 구하실 수 있습니다.

[Espressif 이미지 링크](#)

또는 ESP8266 커뮤니티 포럼을 통해서 다운로드 받을 수 있습니다.

[ESP8266 커뮤니티 포럼 링크](#)

4-2. 공개 업데이트 툴을 이용한 다운로드

공개 업데이트 툴은 인터넷에서 쉽게 구할 수 있는데, 이 경우 위의 공개 툴에서 다운로드한 바이너리 파일들을 하나의 이미지로 만들어서 업데이트 하는 방법입니다. 일반적으로 펌웨어 업데이트는 전체가 아닌 하나의 파일만 업데이트 하는 내용이 골자를 이루고 있습니다.

먼저 공개 펌웨어를 라이팅 하는 방법을 살펴보겠습니다. 우선 다운로드할 유틸리티 프로그램을 받아야 하는데 아래 링크에서 다운로드 받을 수 있습니다.

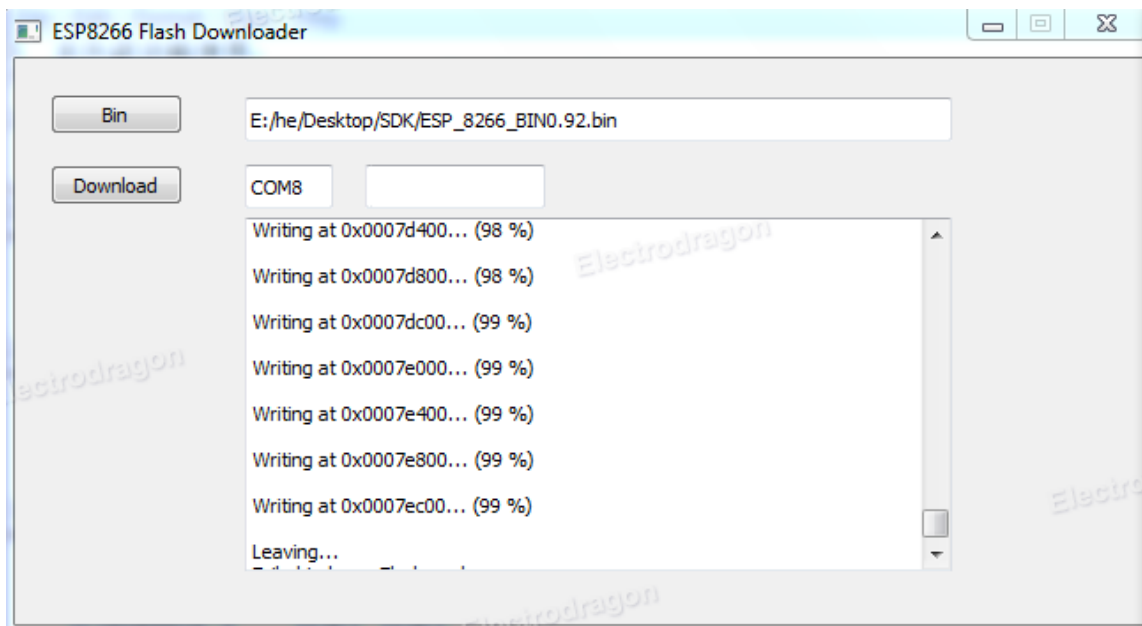
[ESP8266 Flasher 와 이미지 다운로드 링크](#)

다운로드 할 공개 이미지는 위의 링크에 포함되어 있으나 아래 링크에서도 받을 수 있습니다. 이 링크는 초기 통신속도가 9600bps로 작업되어 있는 이미지 입니다.

[ESP8266 AT 다운로드 링크](#)

이제 다운로드 받은 업데이트 프로그램 **ESP8266_Flasher.exe**을 실행하여 다음의 순서로 진행합니다.

1. **BIN** 버튼을 눌러서 이미지 파일을 지정해 줍니다.
파일의 위치는 한글이름이나 공백이 없는 디렉토리에 저장하셔야 합니다.
2. 다음은 연결된 시리얼 포트 번호를 지정합니다.
3. 마지막으로 **Download** 버튼을 누르면 자동으로 다운로드가 진행 됩니다.
4. 99%까지 진행되고 이후 **Leaving** 메시지가 나타나면 다운로드가 완료된 상태입니다.

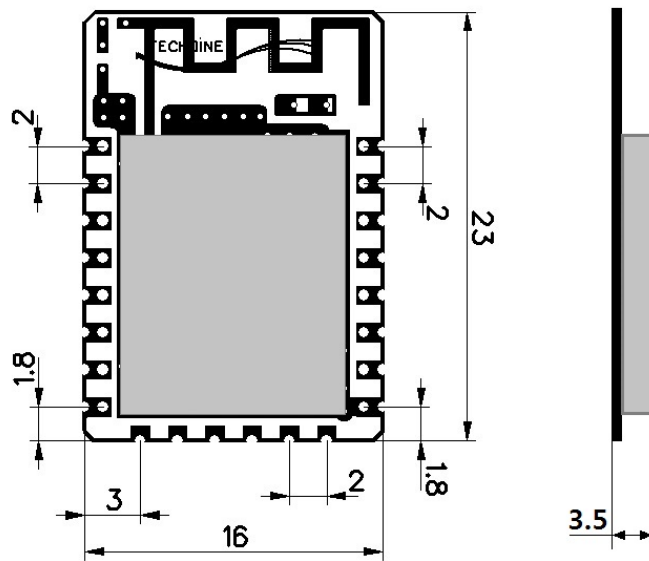


[그림 12] 이미지 다운로드 프로그램을 이용한 다운로드

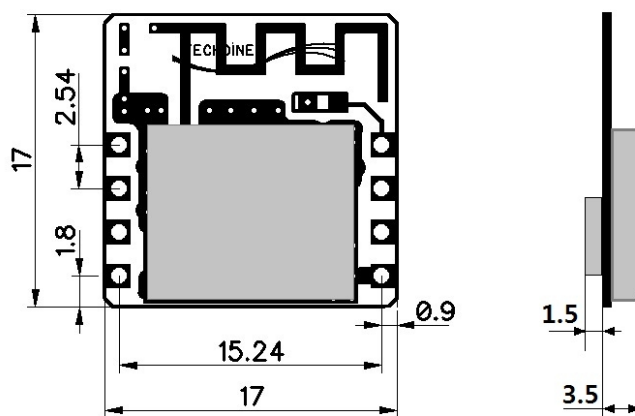
지금까지 기본적인 모듈 사용법을 검토해 보았습니다. 차후 검토할 사항은 아두이노등과 같은 마이컴과 연동하여 프로그램을 작성하는 것을 살펴 보겠습니다.

이어지는 아두이노 **Cortex ARM** 관련 문서를 먼저 살펴보고, 아두이노를 이용한 **wifi** 모듈 프로그래밍 관련 문서를 참고 하십시오.

5. TWM01, TWM02 모듈 규격



[그림 13] TWM01 모듈 규격



[그림 14] TWM02 모듈 규격

Note :