

## ESP8266 - AT 커맨드로 제어하기 예제

작성자



최고관리자

16-01-24 21:35

조회 13,946회

댓글 0건

[이전글](#)
[다음글](#)
[목록](#)


ESP8266 모듈을 사용하는 가장 기본적인 방법은 AT 커맨드를 이용하는겁니다. 보통 ESP8266 모듈을 구매하면 AT-based firmware(AT 기반 펌웨어)가 올라가 있기 때문에 이 방법을 이용해 모듈의 기본적인 통신, 동작 테스트를 해볼 수 있습니다.

이 문서에서 사용된 모듈은 ESP12E 모듈을 탑재한 **NodeMCU V3** 보드입니다.(위 사진 참고) 제가 구매한 이 보드도 AT 기반 펌웨어가 올라가 있습니다.

ESP8266 모듈은 대부분 3가지 모드를 지원합니다.

**STA(stand-alone) 모드** : 이 모드는 ESP8266 모듈이 AP(Access Point)에 접속해서 HTTP Request 를 보내거나 간단한 HTTP 서버가 될 수 있습니다. 가장 기본적인 모드입니다.

**AP 모드** : 이 모드에서는 ESP8266 모듈이 AP(Access Point) 역할을 할 수 있습니다. 일종의 공유기 역할을 하는 것으로 이해할 수 있습니다. 5개 미만의 wifi 장치들이 접속 가능하고 각 장치들에 고유한 주소(192.168.4.x)를 할당합니다. 하지만 접속한 장치들간의 통신은 불가능합니다. 예를 들어 A 장치와 B 장치가 ESP8266 AP에 연결되었다면, 각 장치들이 AP와 통신은 할 수 있지만 A와 B 장치간의 통신은 불가능합니다.

**STA+AP 모드** : STA 모드와 AP 모드 역할을 동시에 수행합니다.

그리고 ESP8266 모듈은 2가지 MUX 타입이 있습니다.

Single/Multiple 타입이 있는데 Single은 하나의 TCP 연결을 지원하고, Multiple 은 5개 까지의 TCP 연결을 지원합니다. Multiple인 경우 0~4 까지의 ID 값을 가지고, Single 일 때와는 달리 AT 커맨드에 ID 파라미터가 하나 더 늘어납니다.

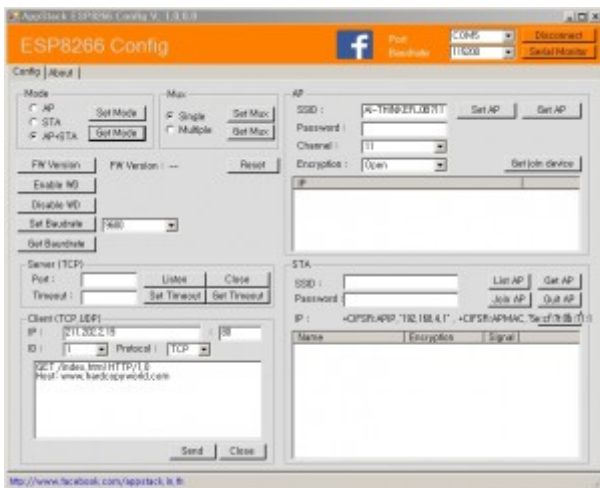
**NodeMCU V3** 보드에 AT 기반 펌웨어가 탑재된 경우 윈도우 PC에 USB 연결만하면 바로 테스트 해볼 수 있습니다. 그리고 윈도우 PC의 경우 Putty 같은 터미널 프로그램을 사용해서 Serial-COM 포트 연결하면 됩니다. 혹은 아두이노 IDE를 실행해서 Serial monitor를 실행해서 AT 커맨드를 전송해도 됩니다. NodeMCU Lua 개발을 할 때 사용하는 개발툴인 ESPlorer를 사용해도 편리하게 AT 커맨드를 보낼 수 있습니다. ESPlorer를 사용하면 굉장히 편리하긴 한데.. 한가지, HTTP Request를 보낼 때 줄넘김 문자를 넣기가 힘듭니다. 그래서 HTTP Request를 전송할 때는 ESP8266 Config 프로그램을 사용하시길 추천합니다. 물론 Putty/Serial Monitor를 이용해 직접 AT 커맨드를 입력해도 되긴 합니다.

ESP8266 Config 프로그램은 아래에서 다운로드 받으세요.

- 다운로드 링크 : [http://esp8266.ru/download/esp8266-utils/ESP8266\\_Config.zip](http://esp8266.ru/download/esp8266-utils/ESP8266_Config.zip)

## HTTP REQUEST 테스트

간단히 HTTP Request 테스트를 해보죠. ESP8266 Config 프로그램을 실행하고 우측 최상단의 COM Port를 선택한 뒤 [Connect] 버튼을 누릅니다. 연결이 정상적으로 되면 [Serial Monitor] 버튼을 눌러 메시지 창을 엽니다.



ESP8266 모듈을 한번 리셋해주세요.(리셋버튼) 그러면 시리얼 모니터에 펌웨어 기본 정보가 뜹니다.

그리고 좌측 상단의 **Mode 영역**에서 [Get Mode] 버튼을 눌러 현재 모드를 확인하세요. AP+STA를 선택하고 [Set Mode] 버튼을 누르면 AT+CWMODE=3이 전송되면서 모드가 변경됩니다.

바로 옆에있는 **Mux 영역**도 확인해보세요. [Get Mux]로 현재 상태를 확인하고 Single로 선택한 뒤 [Set Mux] 버튼을 누르면 AT+CIPMUX=0 커맨드가 전송되면서 설정이 바뀝니다.

기본 설정이 됐으니 공유기에 접속해 보겠습니다. 우측 하단에 **STA 영역**이 있습니다. 여기서 [List AP] 버튼을 누르면 주변의 AP를 스캔합니다.(AT+CWLAP) 원하는 AP를 선택하고 password를 입력한 뒤 [Join AP] 버튼을 누르면 해당 AP에 접속합니다. (AT+CWLAP="ap\_name","ap\_password")

접속에 성공하면 아래와 같은 메시지가 뜹니다.

```
WIFI CONNECTED
WIFI GOT IP
```

이후 [Get IP] 버튼을 누르면 할당받은 IP 정보를 확인할 수 있습니다. (AT+CIFSR)

본격적으로 외부 서버에 접속해서 데이터를 받아와 보겠습니다. [http://www.hardcopyworld.com/esp\\_test.html](http://www.hardcopyworld.com/esp_test.html) 파일을 읽어와 볼겁니다.

좌측 하단에 Client(TCP,UDP) 영역이 보이실겁니다. IP 입력란에 [211.202.2.19] 를 입력합니다. 서버의 IP 입니다. 포트는 80, ID 는 1, Protocol은 TCP를 입력하세요. 그리고 하단에 아래 내용을 입력해줍니다.

```
GET /esp_test.html HTTP/1.0
Host: www.hardcopyworld.com

// 마지막에 2줄의 줄넘김 포함해야 합니다.
```

**반드시!!** 마지막에 2줄의 줄넘김이 포함되어야 합니다. 그리고 [Send] 버튼을 눌렀을 때 아래처럼 결과가 찍혀야 합니다.

```
AT+CIPSTART="TCP", "211.202.2.19", 80 CONNECT

OK
AT+CIPSEND=60
OK
> busy s...

Recv 60 bytes

SEND OK

+IPD,482:HTTP/1.1 200 OK
Date: Sun, 24 Jan 2016 13:18:58 GMT

Server: Apache/2
X-Powered-By: PHP/5.2.17
Content-Length: 317
Connection: close
Content-Type: text/html
.....
This is ESP8266 HTTP request test page. If you can see this, your request is successful!!
.....
CLOSED
```

ESP8266 Config 프로그램은 우리가 입력한 정보를 바탕으로 AT 커맨드를 생성해서 서버에 TCP 연결을 하고 메시지를 보내서 HTML 코드를 받아온 겁니다. 이 과정에서 사용된 AT 커맨드를 하나씩 살펴보겠습니다.

HTTP Request를 서버로 보내기 위해선 서버와 TCP 연결을 먼저 해야 합니다. 아래 명령어가 TCP 연결을 open 하는 AT 커맨드입니다.

- AT+CIPSTART="TCP","211.202.2.19",80

그리고 전송할 HTTP Request 메시지 길이를 입력해야 합니다. 아래 AT 커맨드를 사용합니다.

- AT+CIPSEND=60

여기까지 진행되면 [>] 프롬프트가 시리얼 모니터에 뜨는데, 이때 전송할 HTTP Request 메시지를 적어주면 됩니다. AT+CIPSEND 명령으로 정해둔대로 60자 길이로 적어줘야 합니다. 하단 입력창에 HTTP 메시지를 적어둔게 자동으로 입력되어 서버로 전송됩니다.

그럼 서버에서 esp\_test.html 파일의 내용을 응답으로 보내줍니다. ESP8266 모듈은 수신받은 응답을 주르륵 출력해줍니다.

이후 자동으로 TCP 연결은 해제됩니다. TCP 연결이후 아무 입력이 없이 일정 시간이 경과해도 자동으로 TCP 연결이 해제됩니다.

하드카피월드 게시판의 특정 게시물을 이런식으로 가져올 수 있습니다. 아래처럼 입력하면 lecture\_esp 게시판의 wr\_id=36 게시물을 읽어옵니다.

```
GET /gnuboard5/bbs/board.php?bo_table=lecture_esp&wr_id=36 HTTP/1.0
Host: www.hardcopyworld.com

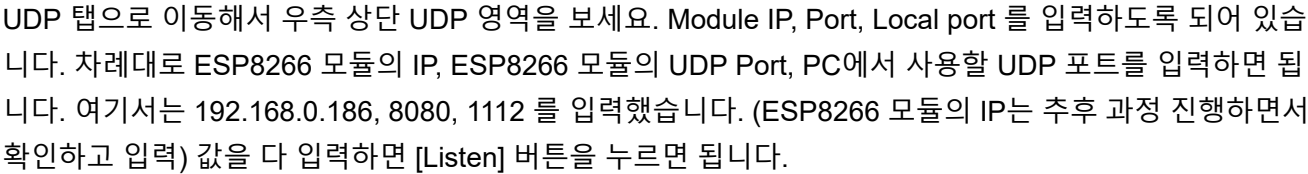
// 마지막에 2줄의 줄넘김 필요
```

이 방법을 응용하면 GET 방식의 HTTP Request 뿐 아니라 POST 방식의 HTTP Request 도 사용할 수 있습니다.

## UDP 전송 예제 (고정된 IP에 응답)

이번에는 UDP를 이용해 PC와 ESP8266 모듈간 데이터 전송을 구현해 보겠습니다.

먼저 PC에 네트워크 테스트용 프로그램을 깔아줍니다. 여러 프로그램들이 있는데, 간단하게 사용하기에는 허큘리스 프로그램이 좋습니다. [링크에서 다운로드 받아 실행할 수 있습니다.](#) (모바일 폰에서 네트워크 테스트 앱을 깔아 테스트해도 됩니다.)



- <http://www.14core.com/downloads-2/>

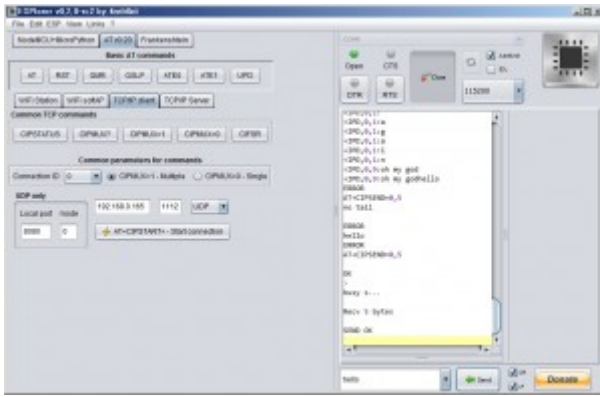
ESPlorer를 실행하면 우측 상단에서 ESP8266 모듈의 COM포트를 선택하고 [Open] 버튼 누르세요. 연결이 되면 좌측 상단에서 [AT v0.20] 탭을 선택하세요. 그리고 아래쪽 [WiFi Station] 탭을 선택합니다.



[CWLAN - Get AP List]를 선택합니다. 그러면 주변 AP들을 검색해서 정보를 출력해줍니다.

출력된 정보를 보고 아래쪽에서 [SSID], [password] 항목을 입력, [CWJAP Connect to AP] 버튼을 누릅니다. 그러면 AP에 접속하고 IP를 받습니다. [CIPSTA? Get Station IP] 버튼을 누르면 IP를 확인할 수 있습니다. 여기서 받은 IP를 PC의 허큘리스 프로그램에 입력해주면 됩니다. 당연히 PC와 ESP8266은 같은 AP에 연결되어야 합니다.

이제 상단의 [TCP/IP client] 탭을 눌러 페이지를 전환합니다.



여기서 본격적인 UDP 세팅을 해줍니다. 일단 [CIPMUX=1] 버튼을 눌러줍니다. MUX - Multiple 모드로 맞춰주는겁니다. 이 모드에서는 최대 5개까지의 connection을 만들수 있으면 각 connection은 고유의 ID를 갖게 됩니다. 아래쪽에 [Connection ID] 선택 스크롤바가 있을겁니다. 기본 0으로 맞춰주면 됩니다.

가장 아래쪽에 UDP 연결 관련된 정보들을 넣어줘야 합니다.

PC의 IP와 UDP 포트를(1112) 입력하고 Connection type은 UDP로 선택합니다. Local port, mode 값은 ESP8266 모듈의 UDP 포트와 전송 모드를 말합니다. 8080, 0 을 입력하면 됩니다.

여기서 mode는 ESP8266 모듈에서 UDP 데이터를 전송할 때 특정 장치로만 고정해서 보낼것인지 결정하는 값입니다. ESP8266의 8080 포트로 여러대의 PC가 UDP 데이터를 전송할 수 있는데, 0으로 설정하면 처음 UDP 데이터를 보내준 상대방에게만 데이터를 전송합니다. 0이 아닌 다른 값을 적어주면 가장 최근에 데이터를 보내준 상대방에게 UDP 데이터를 전송합니다.

입력이 다 되었으면 PC 허큘리스 프로그램에서 [Listen] 버튼을 눌러 UDP 연결을 기다리게 합니다. 그리고 ESPlorer에서 [AT+CIPSTART = - Start connection] 버튼을 눌러 UDP 연결을 시작합니다. 정상적으로 연결 되면 PC 허큘리스 프로그램에서 문자를 입력했을 때 ESP8266 모듈이 UDP 로 받게되고, ESPlorer에 표시 됩니다.

반대로 ESPlorer에서 (ESP8266 모듈을 통해) 데이터를 보내려면 우측 최하단의 입력란에 [AT+CIPSEND=0,5] 를 입력하고 [Send] 버튼을 누릅니다. 0번 connection 에 5byte 데이터를 전송한다는 의미입니다. 그러면 [>] 쉘 프롬프트가 화면에 보일겁니다. 이제 5byte에 해당하는 문자열, 즉 영문자 5개를 입력하면 됩니다. hello 라고 입력해서 [Send] 버튼 눌러보세요. 그럼 PC 허큘리스 프로그램에도 받은 문자열이 보일겁니다.

[AT+CIPSTART = - Start connection] 버튼을 누르면 실제로는 아래 AT 커맨드가 전송되어 실행됩니다.

- AT+CIPSTART=0, "UDP", "192.168.101.110", 8080, 1112, 0

파라미터를 순서대로 살펴보면...

- 0 : connection ID를 말합니다. (0~4)
- UDP : UDP 연결을 의미합니다. TCP가 사용될 수도 있겠죠.
- 192.168.101.110 : 연결할 상대방 IP 주소
- 8080 : 연결할 상대방 포트
- 1112 : ESP8266 자신의 UDP 포트
- 0 : 전송 모드. (0: 고정, 기타: 최근 통신 IP)

이상의 내용을 참고해서 UDP 전송을 다양하게 테스트 해보세요. TCP 연결때와는 달리 UDP 연결 상태는 아래 AT 커맨드를 입력하기 전까지 해제되지 않습니다.

- AT+CIPCLOSE=x (x: connection ID)

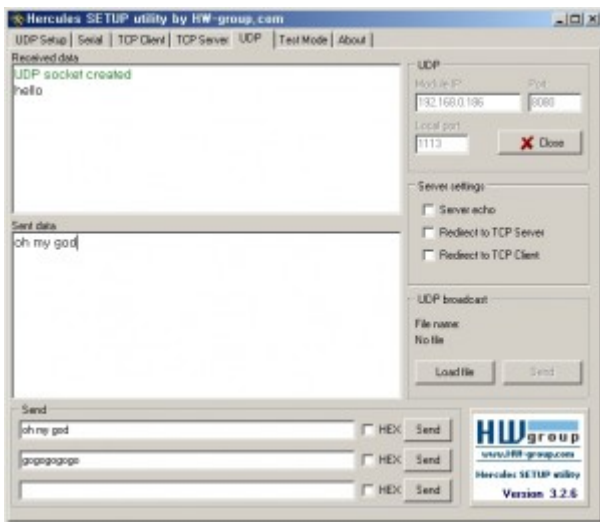
## UDP 전송 예제 2 (마지막 통신 IP에 응답)

앞선 예제와 거의 유사한 테스트입니다. 다만 ESP8266 모듈에서 UDP를 시작할 때 마지막 파라미터만 틀려집니다. 기존에는 0 이었는데 여기서는 2를 사용할겁니다.

- AT+CIPSTART=0, "UDP", "192.168.101.110", 8080, 1112, 2

위 AT 커맨드를 직접 입력해서 UDP를 실행해보세요. 그리고 이전 예제와 같이 PC에서는 허큘리스 프로그램을 실행해서 서로 데이터가 오가는지 테스트 합니다.

그리고 PC에서 허큘리스 프로그램을 하나 더 실행합니다. UDP 세팅을 똑같이 맞춰주는데 local port 값만 살짝 바꿔줍니다. 1112 --> 1113 으로 바꿔줍니다.



자, 이제 PC에서 2개의 허큘리스 프로그램으로 번갈아가면 UDP 메시지를 보내보세요. 그리고 ESP8266으로 데이터를 받으면 아래 AT 커맨드로 답장을 보내보세요.

- AT+CIPSEND=0,5
- >hello

그럼 hello 라는 메시지가 방금 메시지를 전송해준 허큘리스 프로그램으로 전달될겁니다. 1112, 1113 포트를 사용하는 허큘리스의 프로그램 각각에 응답을 보낼 수 있습니다.

## TRANSPARENT TRANSMISSION

Transparent Transmission 이란 마치 두 장치가 Serial 통신 케이블로 직접 연결된 것처럼 사용할 수 있도록 WiFi 장치가 중간에서 무선 데이터 통로를 만들어주는 기능입니다. 즉, 마이크로 컨트롤러에서 Serial(UART) 데이터 전송 선인 TX(송신), RX(수신) 선으로 데이터를 전송하면 그대로 상대방에 전달되도록 WiFi 가 무선 송수신에 필요한 모든 작업을 알아서 다 처리해주는 기능입니다. 이 기능을 사용하면 마이크로 컨트롤러는 TCP/IP, WiFi 컨트롤 등의 작업을 계속 관리할 필요 없이 무선 데이터 전송이 가능합니다.

ESP8266 모듈은 TCP/UDP client로 동작할 때 Transparent Transmission(이하 TT) 기능을 활성화 할 수 있습니다.

### TCP client single connection UART - WiFi passthrough

먼저 TCP client 로 동작하면서 TT 기능을 사용하는 방법입니다. ESP8266 테스트를 위해서는 앞서 사용했던 어떤 툴을 사용해도 좋으며, PC에는 허큘리스 프로그램을 준비합니다.

WiFi 모드 설정

- AT+CWMODE=3 // softAP+station mode  
Response :OK

공유기에 접속

- AT+CWJAP="SSID", "password" // SSID and password of router  
Response :OK

할당받은 IP 확인

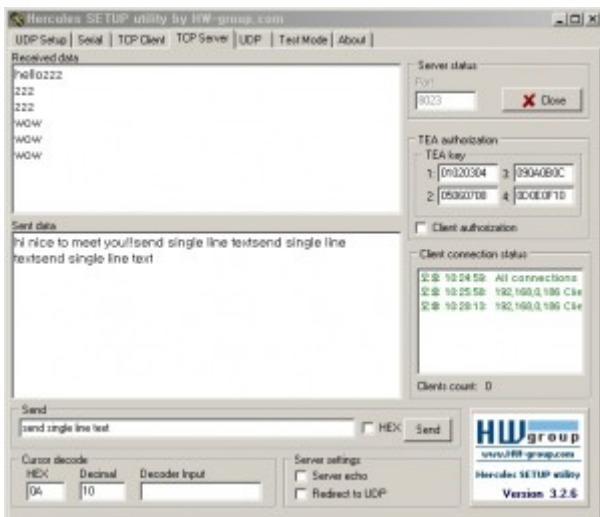
- AT+CIFSR  
Response :192.168.101.105 // Device's IP that got from router.

MUX=0 설정 (Single connection)

- AT+CIPMUX=0

MUX 모드를 single connection으로 설정해야만 TT 기능이 동작합니다.

같은 공유기에 접속된 PC에서 허큘리스 프로그램을 실행하고 TCP 서버를 실행합니다. Port를 8023으로 설정한 뒤 Listen 하세요.



이제 ESP8266 모듈에서 PC의 TCP 서버로 연결합니다. 아래처럼 TCP 접속 AT 커맨드를 보내면 됩니다.

- AT+CIPSTART="TCP","192.168.0.165",8023

앞선 예제에서 사용했던 CIPSTART 명령어와 유사한데 파라미터 부분 맨 앞에 숫자 하나가 빠져있습니다. 왜냐면 지금 MUX=0(single connection) 상태여서 connection ID를 지정할 필요가 없기 때문입니다.



TCP 연결이 되면 TT 모드를 활성화 합니다.

- AT+CIPMODE=1

이제 전송 프롬프트를 열어주면 됩니다.

- AT+CIPSEND
- >

지금부터 입력하는 모든 데이터는 TCP 연결된 장치로 바로 전송됩니다. 심지어는 AT 커맨드를 입력해도 처리하지 않고 전송해 버립니다.

그럼 TT 모드를 종료하려면? [+++] 문자를 입력하면 됩니다.(줄넘김 문자 없이 순수 3byte만) 그럼 1초 뒤 텍스트 자동 전송을 종료하고 AT 커맨드를 인식합니다. 아래 커맨드를 입력하면 TT 모드가 종료됩니다.

- AT+CIPMODE=0

아래 커맨드를 입력해서 TCP 연결도 해제할 수 있습니다.

- AT+CIPCLOSE

## UDP transmission UART - WiFi passthrough

이번에는 ESP8266 모듈을 Soft-AP 로 세팅하고 UDP 로 연결한 상태에서 TT 기능을 사용해 볼겁니다.

softAP+station 모드로 설정

- AT+CWMODE=3

TT 모드를 위해서는 MUX=0 로 설정해 줍니다.

- AT+CIPMUX=0

다른 장치가 ESP8266 모듈에 연결될 수 있도록 Soft AP로 만들어줍니다. SSID는 [espap1], 비밀번호는 [12345678], 채널은 1, 인증 방식은 WPA2\_PSK (2) 로 설정해줍니다. 이것 AT 커맨드로 만들면 아래와 같습니다.

- AT+CWSAP="espap1","12345678",1,2

ESPlorer를 사용하시는 분은 [WiFi softAP] 탭에서 편리하게 설정할 수 있습니다. SSID, password, channel, encryption 설정하고 [CWSAP= Set SoftAP co...] 버튼을 눌러주면 됩니다. (ESP8266 Config 앱은 문제가 생김;;)

그러면 ESP8266 모듈은 공유기처럼 동작하고 스스로 192.168.4.1 IP를 사용하게 됩니다. 여기에 접속하는 WiFi 장치들은 순서대로 192.168.4.2~5 까지의 IP를 사용하게 됩니다.

이제 PC를 ESP8266 공유기(AP)에 접속시켜 줍니다. 그리고 허큘리스 프로그램을 실행해서 UDP 서버를 셋팅해둡니다. 앞선 예제에서 해봤던 UDP 셋팅과 같은 방식으로 하며 됩니다. module IP=192.168.4.1, Port=8080, Local port=1112 로 맞추세요.

그리고 ESP8266 모듈에서 UDP 연결을 시도합니다. 1112는 접속할 PC의 UDP 포트이고 8080은 ESP8266 모듈이 사용하는 UDP 포트입니다.

- AT+CIPSTART="UDP","192.168.4.2",1112,8080,0

UDP 연결이 성공하면 PC에서 데이터를 보내보세요. 데이터가 ESP8266쪽에 표시되면 연결은 성공입니다. 이제 TT 기능을 활성화 합니다.

- AT+CIPMODE=1
- AT+CIPSEND
- >

[>] 프롬프트가 뜨면 데이터를 전송할 준비는 끝. 이제부터 Serial(UART)로 입력되는 모든 데이터는 연결된 장치(PC)로 보내집니다.

연결을 해제하려면? 앞선 예제와 같습니다. [+++] 문자를 줄넘김 문자 없이 보내면 1초 후에 AT 커맨드를 인식합니다.

이제 TT 기능을 끄고 UDP 연결을 종료할 수 있습니다.

- AT+CIPMODE=0
- AT+CIPCLOSE

## MULTIPLE CONNECTION AS TCP SERVER

ESP8266 모듈을 TCP 서버로 만들 수 있습니다. 그리고 당연히 이 때는 여러 TCP 클라이언트들과 다중 접속이 가능해야 합니다. 이번 예제에서는 ESP8266 모듈을 SoftAP 모드로 설정하고 TCP 서버를 활성화 할 겁니다. 그리고 다른 여러 장치들이 여기에 TCP 접속을 해서 데이터 교환을 해보도록 할 겁니다.

ESP8266 모듈을 SoftAP + Station 모드로 맞춰줍니다.

- AT+CWMODE=3 // softAP+station mode

여러 장치들과 TCP 연결을 맺어야 하므로 MUX=1 (multiple connection)으로 설정해줍니다.

- AT+CIPMUX=1

다른 장치가 ESP8266 모듈에 연결될 수 있도록 Soft AP로 만들어줍니다. SSID는 [espap1], 비밀번호는 [12345678], 채널은 1, 인증 방식은 WPA\_PSK (=2) 로 설정해줍니다. 이것 AT 커맨드로 만들면 아래와 같습니다. (WPA2\_PSK 인 경우 마지막에 3을 기입)

- AT+CWSAP="espap1","12345678",1,2

이 상태에서 ESP8266 모듈이 TCP 서버 역할을 하도록 설정해줍니다.

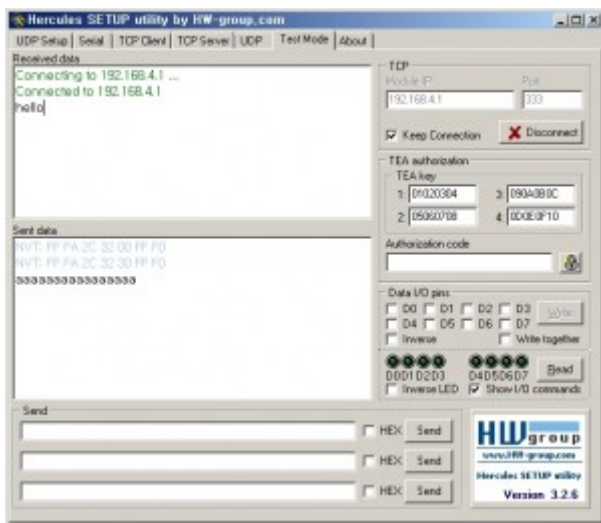
- AT+CIPSERVER=1 // default port = 333

이제부터 ESP8266 모듈은 공유기(SoftAP) 역할을 하면서 접속한 장치에서 오는 TCP 연결 요청을 수락하고 관리합니다. 즉 TCP로 데이터를 주고 받을 수 있습니다. ESP8266은 TCP 서버일 때 333 포트를 사용합니다. 외부 장치는 이 포트로 TCP 연결 요청을 보내면 됩니다.

PC를 ESP8266 공유기(SoftAP)에 접속시켜 줍니다. 그리고 PC에서 허큘리스 프로그램을 실행합니다. 허큘리스 프로그램의 [Test Mode] 탭을 선택하세요. (TCP Client 탭을 써도 됨)

PC가 WiFi로 연결된 ESP8266 모듈은 SoftAP로 동작하면서 192.168.4.1 IP를 사용합니다. 따라서 192.168.4.1 - 333 번 포트로 TCP 접속을 해주면 됩니다. [Keep connection] 체크박스가 있다면 체크해 주세요. ESP8266 모듈은 TCP 연결 후 아무런 데이터 전송 없이 5~10초가 지나면 연결을 해제합니다. [Keep connection] 기능은 연결이 해제되지 않도록 주기적으로 신호를 보내줍니다.

TCP 연결 후 양방향으로 데이터가 잘 전송되는지 확인하세요.



ESP8266 모듈은 softAP로 동작하면서 여러개의 TCP연결을 처리한다고 했습니다. PC에서의 TCP 연결 하나는 처리하고 있으니 여기에 하나 더 연결시켜서 확인해 보겠습니다. 이를 위해 PC에서 허큘리스 프로그램을 하나 더 실행해도 됩니다. 하지만 이번에는 색다르게 ESP8266 모듈을 하나 더 준비해서 테스트 해보겠습니다.

추가로 준비한 ESP8266 모듈을 PC에 연결하고 Putty 또는 ESPlorer/ESP8266 Config 프로그램을 하나 더 실행합니다. 새로 연결된 모듈의 COM 포트에 접속해서 설정할 준비를 하세요. 새로 연결한 ESP8266 모듈은 softAP로 동작하는 모듈에 연결한 뒤 TCP client 로 접속하도록 할겁니다.

새로 연결한 ESP8266 모듈은 WiFi Station 모드로 동작하도록 설정합니다.

- AT+CWMODE=1

MUX=0 (single connection) 모드로 설정합니다.

- AT+CIPMUX=0

아래처럼 TCP 접속 AT 커맨드를 보냅니다.

- AT+CIPSTART="TCP","192.168.4.1",333

TCP 연결이 되면 TT 모드를 활성화 합니다.

- AT+CIPMODE=1

이제 전송 프롬프트를 열어주면 됩니다.

- AT+CIPSEND
- >

데이터를 전송해서 ESP8266 모듈이 다중 접속을 처리하는지 확인하세요. (PC, ESP8266 모듈과 TCP 연결 2개가 유지되는지 확인하세요.)

## 참고자료

- [ESP8266 AT Command Set Example](#)