* [admin](https://crystalcube.tistory.com/manage)
* [write](https://crystalcube.tistory.com/manage/entry/post)
* [홈](https://crystalcube.co.kr/)



[[ESP8266-01] 최신 Firmware 업그레이드](https://crystalcube.co.kr/215) [Hardware](https://crystalcube.co.kr/category/Hardware)

ESP8266-01 최신 펌웨어 업그레이드 (a.k.a 거지같은 것)

WiFi 통신을 위해서 모듈을 하나 구매했다. 그건 바로 가장 저렴한 ESP 모듈 중에서 01 이라고도 불리는 ESP8266 이다.

왜 그런지도 모르겠지만, 다들 펌웨어 업그레이드를 하길래 나도 한번 해 보았다.

사실,, 해야만 하는 줄 알았다 -\_-;;

대부분 아두이노 우노를 가지고 개발을 많이 하는데, 이 경우 Serial Port 가 1개 밖에 없기 때문에, ESP8266 과의 통신을 위해서는 SoftwareSerial 을 사용해야 했다. 그런데 SoftwareSerial 의 경우 115200 Baud Rate 를 지원하지 않는다. 그렇기 때문에 ESP8266 의 Baud Rate 를 57600 정도로 낮춰야 한다. 그럼 어떻게 낮춰야 하는가? "AT+UART\_DEF=57600,8,1,0,0" 이렇게 AT 커맨드로 낮출 수 있다. 그런데 이게 옛날 펌웨어에서는 지원되지 않는 커맨드이다. 만약 구매한 ESP8266 이 해당 커맨드를 지원하는 펌웨어가 설치되어 있다면, 아두이노를 이용해서 ESP8266 에 시리얼 연결 후, 해당 커맨드로 Baud Rate 를 낮추면 된다. 그러면 펌웨어 업그레이드 없이 SoftwareSerial 로도 통신이 가능하다. 안타깝게도 해당 커맨드를 지원하지 않는 펌웨어가 설치되어 있다면, 이 글에서처럼 펌웨어 업그레이드를 해 주어야 하는 것이다.

만약 ESP8266 을 가지고 아두이노와 통신할게 아니라, 자체적으로 아두이노 부트로더를 올려서 독립적인 ESP8266의 MCU 프로그래밍을 할 것이라면, 이 모든 것은 필요가 없다.

마지막으로 해당 글은 아두이노 메가 2560 기준이라는 점을 주의하기 바란다.

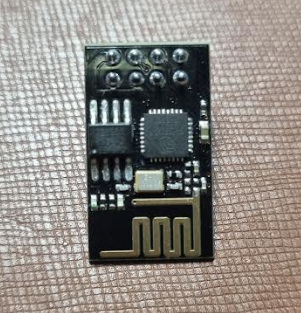
**하드웨어쟁이들은 상종하지 말아야지**

일단 내가 가장 크게 느낀것은 하드웨어쟁이들은 상종하지 말아야 겠다는 것이다. 소프트웨어도 불친절하고(90년대에도 이것보다 나았겠다!), UX 도 UI 도 너무나 불편했다. 제대로 된 자료도 없고 - 심지어 같은 내용인데 블로그마다 말이 다름 - 왜 그런지 설명 따위도 사치인 동네 같았다.

아무튼 불평불만은 여기까지만 하고, 본론으로 들어 보도록 하자.

**삽질의 시작. 저항 그거 달아야해?**

나는 ESP8266-01 모듈 세 개를 구매했다. 개당 천 몇백원 정도 준 것으로 기억한다. 아무튼 비싸도 2천원 내외.

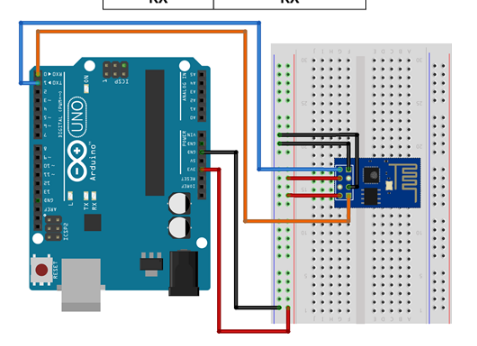


이것을 연결하는것부터 삽질의 시작이었다.

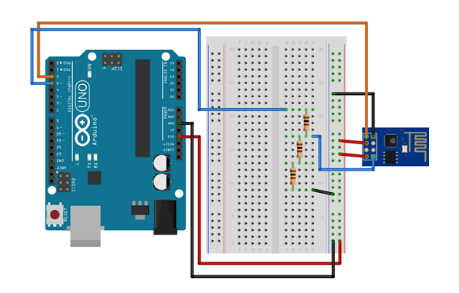
인터넷에 자료를 찾아보았으나, 말이 다 다르고 방식이 다 다르다. -\_-;;

누구는 저항을 달아야 한다고 하고, 어떤 곳에서는 저항을 따로 달지 않았다. 무슨 이야긴가면 아두이노는 5V 전압을 사용하는데 ESP8266 의 경우에는 3.3V 를 사용하므로, RX 단자에 저항을 이용하여 전압을 낮춰야 한다는 것이다.

결과부터 이야기 하자면, 전혀 필요가 없다. 아니 있으면 안된다. 이것 때문에 삽질을 엄청 했다. 아무리 AT 커맨드를 날려도 수신되는게 없더라..... 나중에 저항을 전부 제거했더니 그때부터 RX 단자로 데이터가 수신되기 시작했다. -\_-;; 난 참고로 1K옴과 2K옴을 이용하여 전압을 3.3V 로 만들었었다. 아무래도 전류량이 너무 낮아져서 수신되지 않는 것 같다. 아무튼 저항을 제거해도 이상이 없었다. 조금 작은 저항을 이용했으면 어떻게 되었을지는 궁금하긴 하나 귀찮다.



이렇게 달지 않은 사람도 많고,



이렇게 단 사람도 많단 말이다 -\_-;;;

내 생각에는 SoftwareSerial 을 이용할 경우 5V 라서 저항을 달아야 하는 것 같고, 그 Hardware Serial 인 경우에는 달지 않아야 하나? 싶기도 하다.

결론은 저항은 달지 않는편이 낫다!

**그래서 Serial 통신을 위한 결선도는 어떻게 되는거야?**

인터넷에 자료가 온통 아두이노 우노 기준이어서 삽질을 많이 했다. 참고로 난 우노가 없기 때문에 내가 한 방법이 우노에서도 동작하는건지, 아니면 메가에서만 그런지 잘 모르겠다.

일단 펌웨어를 업그레이드 하기 위해서는 다음과 같은 연결이 이루어 져야 한다.

1. PC 와 아두이노가 연결 되어 있어야 한다.

2. 아두이노는 USB 를 통해서 Serial 통신이 가능하도록, 컨버터(USB to TTL Serial) 같은 역할만 해야 한다. 즉, 아두이노에 의존하는건 없음.

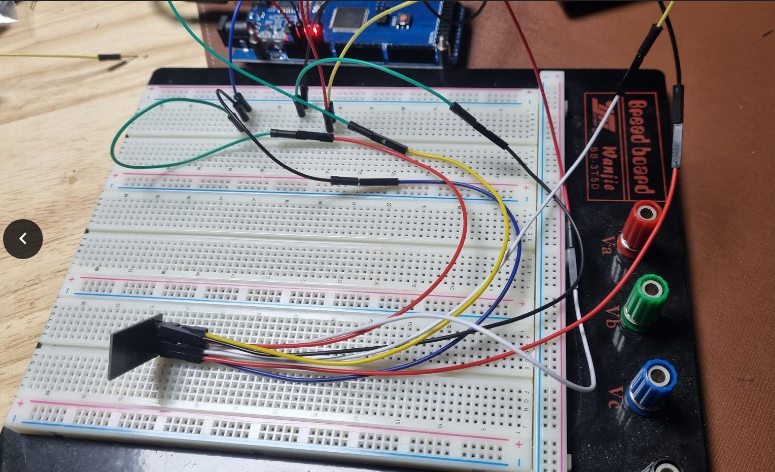
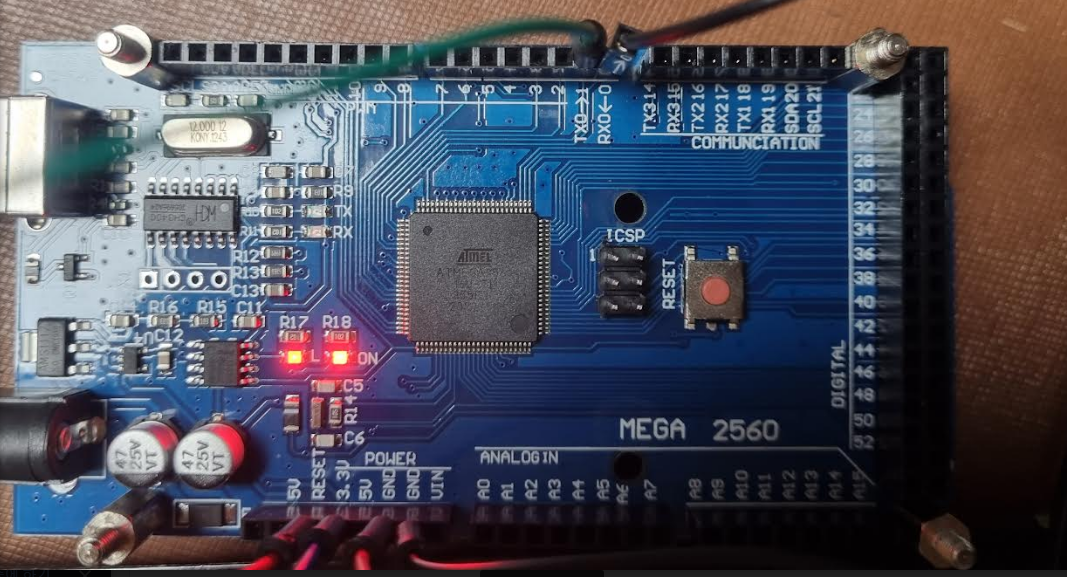
3. 아두이노의 Serial 단자를 ESP8266 과 연결해야 한다.

4. PC 는 현재 COM 포트에 연결된것이 아두이노가 아닌 ESP8266 으로 인식하고 통신할 수 있어야 한다.

이 개념을 잘 생각하고 진행해야 한다.

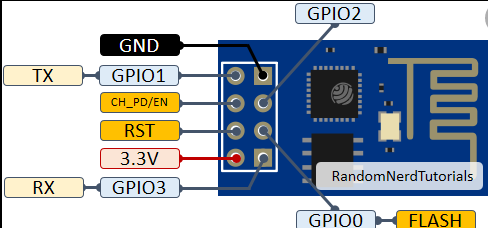
아두이노가 USB to TTL Serial 단자처럼 동작하게 하려면~~ 아두이노의 Reset 단자를 GND 와 연결시켜야 한다. 그렇지 않으면 아두이노는 업로드 되어 있는 스케치를 실행하기 때문이다. 그래서 그냥 단자만 살려두고, 아무짓도 못 하도록 Reset 을 GND 에 물려둔다. 그 이후에 전원을 연결한다.

PC 와 ESP8266 이 직접 연결되어야 하므로, 아두이노의 RX 단자는 ESP8266 의 RX 단자에, 아두이노의 TX 단자는 ESP8266 의 TX 단자에 물려야 한다. 일반적으로 두 개의 디바이스가 통신하려면 RX 와 TX, TX 와 RX 처럼 반대로 물려야 하는데, 지금 하려는건 그게 아니다. 아두이노와 ESP8266 이 통신하게 하려는 것이 아니라, PC 와 ESP8266 이 통신하게 하려는 것이다. 그러므로 그냥 전선을 연결해서 연장한것처럼 생각하면 된다. RX 는 RX로, TX 는 TX 로, 저항 따위 없이.



가진게 없어서,, 있는 점프선으로 대강 연결했다. 다시 말하지만 저항 따윈 없다. 있으면 안되더라.

여기서 결선도를 한번 넣어주면 좋은데, 그리기가 귀찮은 관계로 테이블로 대체하도록 하겠다.

하드웨어쟁이들은 결선도는 핀이 왼쪽에 가도록 그려놓고는, Pin map 은 우측으로 뒤집어진걸 그려놓더라 -\_-;; 난 동일하게 방향 맞춤!

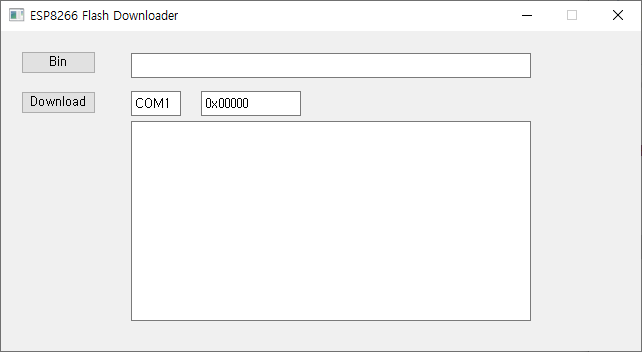
|  |  |  |
| --- | --- | --- |
| **ESP8266-01** | **아두이노 Mega 2560** | **비고** |
|  | Reset 을 GND 와 연결 | 아두이노가 터미널(젠더) 역할을 하도록 항상 연결유지 |
| TX (GPIO1) | TX0 |  |
| CH\_PD/EN | 3.3V |  |
| RST |  | 리셋이 필요할때만 GND 랑 연결/해제 할 예정 |
| 3.3V | 3.3V |  |
| GND | GND |  |
| FLASH (GPIO0) |  | **펌웨어를 올릴때만 GND 와 연결상태 유지** |
| RX (GPIO3) | RX0 |  |

RST 는 다른 모듈들처럼 리셋이 필요한 시점에 GND 에 잠깐 연결했다가 떼면 된다. 그리고 FLASH (GPIO0) 은 펌웨어를 ESP8266 에 올릴때 (다운로드 라는 용어가 본래 맞음)만 연결된 상태를 유지한다. 잘 올라갔는지 나중에 COM 포트로 연결해서 AT 커맨드를 날릴때는 제거되어 있어야 한다.

**그래서 어떤게 최신인데?**

이 부분도 참으로 삽질한 포인트 중 하나이다. 인터넷에 ESP8266 펌웨어 올리는 글을 찾아보면, 무지성으로 나온게 너무 많다. 2021년에 올라온 글에서 2016년에 만들어진 펌웨어를 올리고 있고, 또 어떤 글에서는 Flash Updater 를 쓰는데 어디서는 FLASH DOWNLOAD TOOL 을 쓴다. 누구는 2.0을 쓰고 누구는 3.0을 쓴다. 메뉴얼도 제 각각이다. -\_-;;

난 온라인에 떠도는 모든 툴을 다 써봤다고 감히 이야기 해 정도이다. 결론만 말하겠다. 과정을 일일이 설명하기에는 너무 많은 삽질들을 해 왔기 때문이다.



위와 같이 생긴 프로그램이 있다. 아마 다른 글들을 봤다면 한번씩 마주쳤을 툴이다. 결론부터 말한다.

이것이 필요한 시점은 딱 한가지다.

ESP8266 이 맛탱이가 가서 공장 초기화를 해야 할때.

그때 어떤것을 받아야 하는가?

[http://www.ai-thinker.com](http://www.ai-thinker.com/)

[Ai-Thinker Co., Ltd](http://www.ai-thinker.com/" \t "_blank)

[B&T provides customers with N92 series modules with wide coverage, multiple connections, high speed, low cost, low power consumption, excellent architecture, etc., which can be widely used in various IoT occasions, suitable for smart cities, smart meters ,](http://www.ai-thinker.com/" \t "_blank)

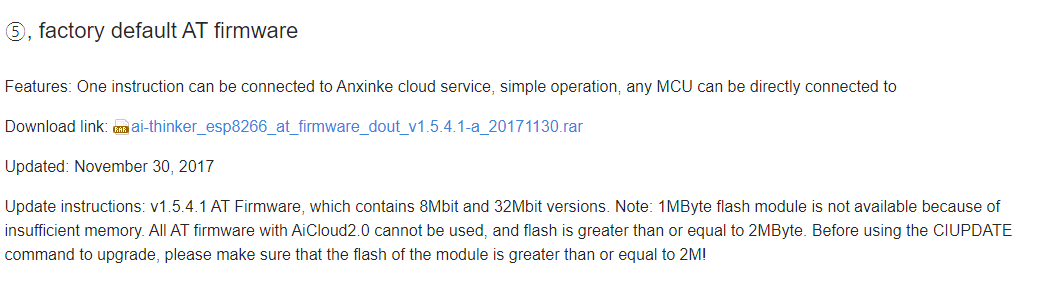
[www.ai-thinker.com](http://www.ai-thinker.com/" \t "_blank)

위 사이트에서 받으면 되는데.

'상단 Document' > '본문 중간쯤 WiFi Module Series' > '본문 ESP8266 series' > '본문 하단 Various AT firmware'

맨 아래 보면 다음과 같이 되어 있다.

공장 초기 AT 펌웨어!



일일이 찾아가기 귀찮다면 아래 링크를 클릭하면 된다. 홈페이지가 리뉴얼되면 주소가 종종 바뀌는 경우가 있으므로 위에서 굳이 설명한것 뿐.

<https://docs.ai-thinker.com/en/%E5%9B%BA%E4%BB%B6%E6%B1%87%E6%80%BB>

[Various AT firmware summary | 安信可科技](https://docs.ai-thinker.com/en/%E5%9B%BA%E4%BB%B6%E6%B1%87%E6%80%BB" \t "_blank)

[docs.ai-thinker.com](https://docs.ai-thinker.com/en/%E5%9B%BA%E4%BB%B6%E6%B1%87%E6%80%BB" \t "_blank)

아무튼 위 파일을 다운로드 해서 압축을 풀고나면(아니 하드웨어 쟁이들은 언제적 rar 을 쓰고 있는거야 -\_-;;;),,

Ai-Thinker\_ESP8266\_DOUT\_8Mbit\_v1.5.4.1-a\_20171130 > Ai-Thinker\_ESP8266\_DOUT\_8Mbit\_v1.5.4.1-a\_20171130.bin

이 파일이 있다.

이것을 위에 'ESP8266 Flash Downloader' 에 넣어 준 뒤, COM1 대신 본인 아두이노가 연결된 포트를 써 주면 된다.

그런 다음에 ESP8266 의 'FLASH (GPIO0)' 핀을 아두이노 GND 와 연결한다. 마지막으로 ESP8266 의 RST 단자를 GND 와 잠깐 연결했다가 뗀다. 그리고나서 Download 버튼을 누르면 해당 bin 파일이 ESP8266 의 Flash 칩에 복사가 된다.

여기서 잠깐!!

펌웨어는 어디에 저장되는가? 바로 FLASH 메모리에 저장된다. 그렇다면 FLASH 메모리는 어디에 있는가? ESP8266 에 보면 다리가 총 8개 달린 칩이 하나 있다. 그것이 FLASH 메모리다.

그럼 요 녀석의 용량은 얼마인가? 1MB 이다. 즉 8Mbit 로 환산된다. 그게 왜 중요한가? 그것은 바로 펌웨어가 이곳에 저장될 것이므로 우리가 올릴 수 있는 펌웨어의 용량은 1MB 보다 작아야 한다는 이야기다. 즉 1024KB 이하만 가능하다.

인터넷에 뒤지다보면 ESP8266 의 32Mbit (즉 4MB) 짜리 펌웨어들이 있는데, 이것은 ESP8266-01 에는 올릴수가 없는 것이다. 그러니 이 부분에서 삽질하지 않도록 하자. ESP8266-IDF-AT\_V2.2.0.0.zip 이런 파일도 굴러다니는데(공식 사이트에 있음), 이것도 용량이 2MB 이므로 당연히 ESP01 에는 올릴수 없다.

파일의 업로드가 다 끝났다면 이제 FLASH 핀을 GND 에서 제거한다. 그리고 아두이노(혹은 다른)의 시리얼 모니터를 연다. 보드레이트는 74880 이고, 줄바꿈은 'Both NL & CR' 로 해야한다. 소프트웨어 개발자라면 알겠지만, 개행문자를 New line 과 Carriage return 모두 넣겠다는 뜻이다. '\n\r' 를 의미.

주의하라 115200 이 아니라 74880 이다

여기서 인터넷을 좀 본 사람들은 의문을 가질 것이다. 왜 115200 이나 9600 도 아닌 74880 인가?!

그것은 ESP8266-01 의 기본 보드레이트가 74880 이기 때문이다. 이렇게 하지 않고 115200 이나 9600 으로 설정하면, 처음 ESP8266-01 이 구동될때 나오는 메세지를 볼 수가 없다. 깨진 문자가 출력되어 내용을 볼 수 없다. 그러므로 의심하지 말고 74880 으로 세팅하자.

이제 ESP8266-01 의 RST 단자를 GND 와 연결한 뒤 1초정도 있다가 제거를 한다. 그러면 시리얼 모니터에 ESP8266 의 부팅 메세지가 출력된다.

ets Jan 8 2013,rst cause:2, boot mode:(3,6)

load 0x40100000, len 1856, room 16

tail 0

chksum 0x63

load 0x3ffe8000, len 776, room 8

tail 0

chksum 0x02

load 0x3ffe8310, len 552, room 8

tail 0

chksum 0x79

csum 0x79

2nd boot version : 1.5

SPI Speed : 40MHz

SPI Mode : DOUT

SPI Flash Size & Map: 8Mbit(512KB+512KB)

jump to run user1 @ 1000

rf cal sector: 249

rf[112] : 00

rf[113] : 00

rf[114] : 01

SDK ver: 1.5.4.1(39cb9a32) compiled @ Jul 1 2016 20:04:35

phy ver: 972, pp ver: 10.1

)?NO

이어서 보드레이트를 115200 으로 변경 한 후, AT+GMR 이라고 커맨드를 날려보면 아래처럼 추가 정보를 볼 수 있다.

AT version:1.2.0.0(Jul 1 2016 20:04:45)

SDK version:1.5.4.1(39cb9a32)

v1.0.0

Mar 11 2018 18:27:31

OK

2nd boot version 과 SDK 의 정보를 보라. 공장초기화의 맛이 어떠냐?!! 무려 6년전의 파일이다. ㅎㅎ.....

그리고 SPI Mode 를 보면 DOUT 으로 되어 있다. 이건 나중에 설명하도록 하겠다. 간략히 설명하자면 우리가 받은 초기화 bin 파일이 DOUT 이다.(파일명 참고)

그리고 맨 아랫줄에 깨진듯한 문자는 Ready 라는 문자열이다. 이것을 확인하고 싶다면, 보드레이트를 115200 으로 바꾸고나서, ESP8266 의 Reset 단자를 GND 와 잠깐 연결했다가 떼어 보아라. 위 화면과는 다르게 앞 부분이 전부 깨진 문자로 나오고, 맨 아랫줄만 Ready 라고 제대로 보일 것이다.

아무튼 이건 공장초기화 방법이고.. 이제 본래 목적대로 최신 펌웨어를 업그레이드 해 보자.

**그래서 최신버전은 어떻게 올리는데?**

최신버전 관련해서 삽질을 많이 했다. 온통 어디서 주워온 파일들만 존재하고, 그래서 도대체 그 파일이 어디서 생산되는지 정보가 없다. 하드웨어쟁이들은 아직도 90년대초 HD 5.25인치 플로피 디스크로 파일을 주고 받던 - 출처도 모르는 - 시절에 머물러 있는 것 같다. 그 누구하나 제대로 된 정보를 제공하는 곳이 없다. -\_-

아무튼 인터넷에서 펌웨어를 구하다보면 아래 사이트 이야기가 나온다.

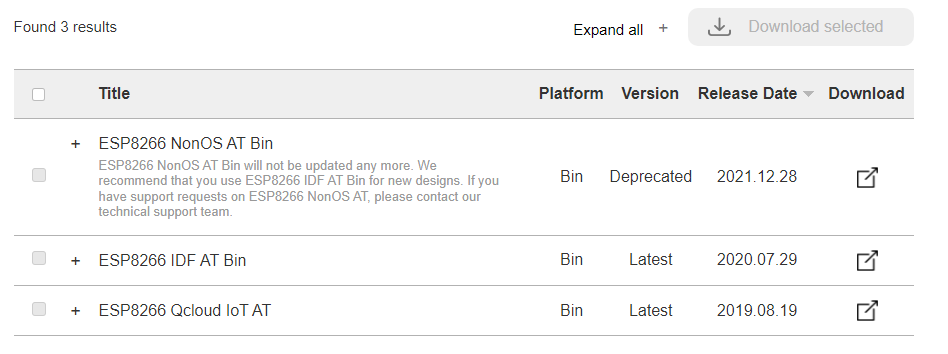
<https://www.espressif.com/>

[Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems](https://www.espressif.com/" \t "_blank)

[Welcome to Espressif's Meeting Series at CES 2019 Jan 8 - 11, 2019 CES attendees are invited to book one-to-one meetings with Espressif's representatives at CES 2019, so they can have a more structured and productive business discussion. To book a meeting,](https://www.espressif.com/" \t "_blank)

[www.espressif.com](https://www.espressif.com/" \t "_blank)

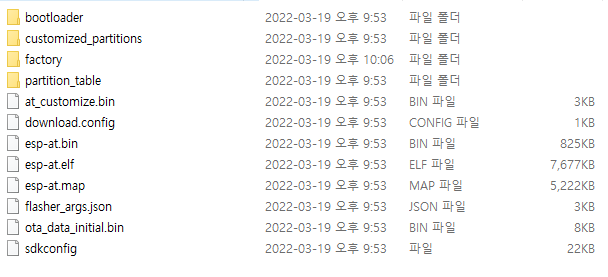
그리고 펌웨어를 찾아서 가 보면,, 아래처럼 절망적인 글이 보인다.



ESP8266 NonOS AT Bin 이 Deprecated 되었다. -\_-a

날짜가 무려 2021.12.28 이다. 그리고 내용을 읽어보면 ESP8266 IDF AT Bin 을 사용하란다. 바로 밑에 있다. 이녀석의 Release Date 를 보면 2020.07.29 이다. 앵?? 왜 더 과거지???

모쪼록 이것을 눌러서 들어가보면, 링크의 링크의 링크... 결국 Github 에 도착한다. ㅅㅂ... 어쨋든 그렇게해서 파일을 얻은것을 보면..

어쩌라고....

도대체 어떤 파일을 올려야 하는건지 알 수가 없다. 일단 확실한것은 앞서 이야기

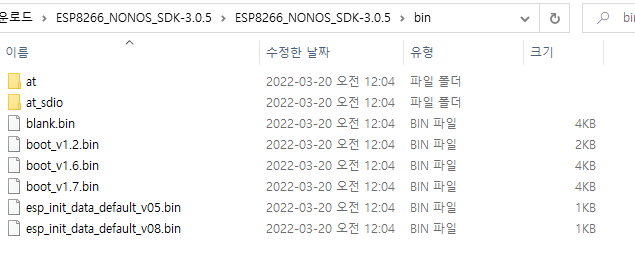
했듯이 1024KB 를 초과하는것은 ESP8266-01 용 펌웨어가 아니다. 그러니 시도조차 하지 마시라.

난 여기서 몇가지 bin 파일을 가지고 실험을 했으나 모두 실패하였다. 혹시라도 위 파일을 가지고 성공한 사람이 있다면 코멘트를 달아주면 좋겠다.

나는 그래서 여기저기 찾다가 끝내 해당 펌웨어의 출처를 찾았다. Github 가 존재하고 있었다. 제조사인 espressif 의 Github 주소이다.

<https://github.com/espressif/ESP8266_NONOS_SDK/releases>

가 보면 현재날짜 기준으로 최신 파일인 1.7.5 (v3.5.0) 이 존재한다. 2021년 10월 18일 버전이다. 다운로드 받아서 열어봤더니, 빌드된 파일이 포함되어 있다. 대략 압축을 풀면 이렇게 되어 있다.



아무튼 이제 업로드하는 툴이 필요한데, 이것도 삽질을 많이 해봤다. 최신버전도 받아보고 여러가지 버전을 받아서 테스트 해 보았으나 내 경우에는 'FLASH\_DOWNLOAD\_TOOLS\_v2.4\_150924' 버전이 문제없이 정상동작 하였다. (웃긴건 실행하면 v2.3으로 나옴)

'flash\_download\_tool\_3.9.2' 로도 해봤는데 안되더라 -\_-;

[FLASH\_DOWNLOAD\_TOOLS\_v2.4\_150924.zip](https://blog.kakaocdn.net/dn/ESDSM/btrwnaSbOit/YsgknimYSqoLyjOqxU0EG1/FLASH_DOWNLOAD_TOOLS_v2.4_150924.zip?attach=1&knm=tfile.zip)

[7.44MB](https://blog.kakaocdn.net/dn/ESDSM/btrwnaSbOit/YsgknimYSqoLyjOqxU0EG1/FLASH_DOWNLOAD_TOOLS_v2.4_150924.zip?attach=1&knm=tfile.zip)

아무튼 별개로 공식적인 Tool 은 아래에서 다운로드 가능하다.

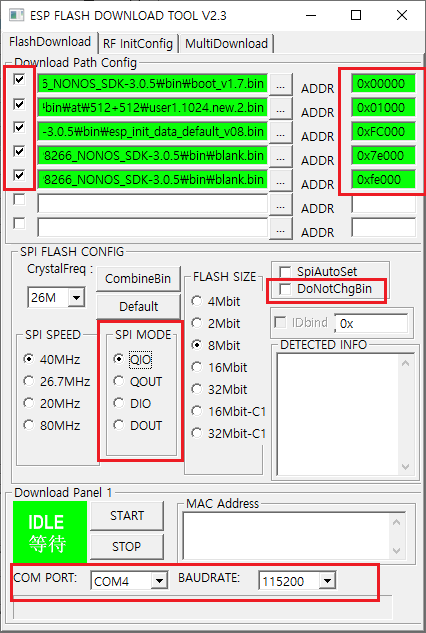
<https://www.espressif.com/en/support/download/other-tools>

[Tools | Espressif Systems](https://www.espressif.com/en/support/download/other-tools" \t "_blank)

[Quickly get Espressif's RF performance test tool, test firmware, test guide and certification test guide for ESP32, ESP32-C3, ESP32-S2, and ESP8266 series of product.](https://www.espressif.com/en/support/download/other-tools" \t "_blank)

[www.espressif.com](https://www.espressif.com/en/support/download/other-tools" \t "_blank)

다운로드 받아서 실행하면 다음과 같이 창이 뜨는데, 여기서 이렇게 설정하면 된다.



빨간색으로 쳐 놓은 박스는 주의할 부분이다. 첫번째로 좌측에 체크박스가 모두 체크되어 있어야 한다. 프로그램이 ㅂㅅ이라서 bin 파일명쪽의 녹색 하이라이트와 체크박스 싱크가 안 맞는 경우가 있다. 무조건 체크박스만 믿어라.

그리고 우측에 ADDR 이라고 플래시 메모리 어디에 써야 하는지 메모리 주소가 나와 있는데, 이건 다들 무지성으로 설명하더라 -\_-;; 이건 bin 파일 설명을 보아야 한다.

bin > at 폴더에 보면, README.md 파일이 있다. 이것을 열어보자.

# Notice:

AT firmware becomes larger since it supports more functions. So, we provide two firmwares here:

\* 1024+1024: Normal AT, which is compiled from ESP8266\_NONOS\_SDK/examples/at, and uses mbedTLS lib.

\* 512+512: Nano AT, which is compiled from ESP8266\_NONOS\_SDK/examples/at\_nano, and uses SSL lib that supports less cipher suites.

It is suggested to use normal AT(1024+1024), if your flash size is 2MB or larger.

# BOOT MODE

## download

### Flash size 8Mbit: 512KB+512KB

boot\_v1.2+.bin 0x00000

user1.1024.new.2.bin 0x01000

esp\_init\_data\_default.bin 0xfc000

blank.bin 0x7e000 & 0xfe000

### Flash size 16Mbit-C1: 1024KB+1024KB

boot\_v1.2+.bin 0x00000

user1.2048.new.5.bin 0x01000

esp\_init\_data\_default.bin 0x1fc000

blank.bin 0xfe000 & 0x1fe000

대충 이런식으로 쭉~ 뭐가 써있다.

우리는 누차 이야기하지만 8Mbit 플래시를 가지고 있는 ESP8266 이므로 첫번째 부분에 내용을 따라야 한다.

boot\_v1.2+.bin 이라고 되어 있는데 -\_- 1.2 이상 버전이면 된다는 뜻이다. + 가 파일명이 아니라는거에 주의하자.

\bin\boot\_v1.7.bin

\bin\at\512+512\user1.1024.new.2.bin

\bin\esp\_init\_data\_default\_v08.bin

\bin\blank.bin

\bin\blank.bin

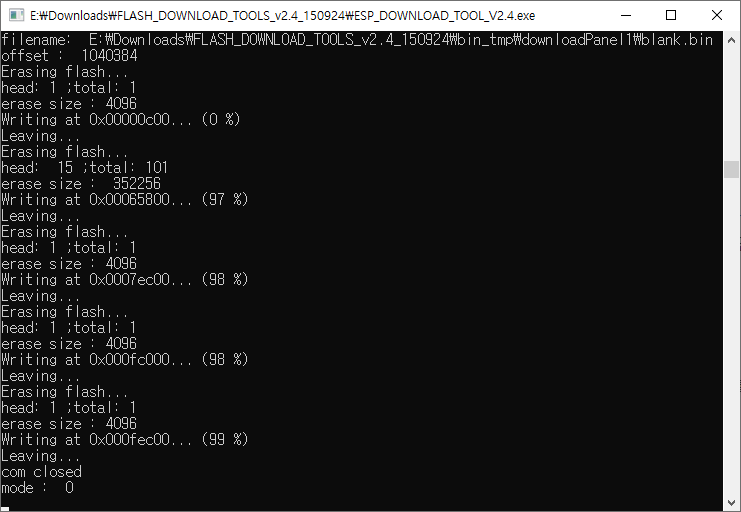
이렇게 4개의 파일을 우측에 나와있는 각 주소에 넣으면 된다. blank.bin 은 잘 보면 주소가 두 개 나와있다. & 표시에 헷갈리지 말자(하드웨어쟁이들은 왜 & 를 함부로 쓰는가 -\_-; 그냥 콤마 찍으면 안되나?)

그리고 'DoNotChgBin' 이라는게 있는데, 이건 체크하면 안된다. 앞서 공장초기화처럼 단일 bin 파일을 업로드 할때는 체크를 해야하고, 지금처럼 여러개의 bin 파일을 올려서 합치는 경우에는 체크하면 안된다. 절대로 안된다.

SPI Mode 는 일단 QIO 로 하자. 자세한 설명은 나중에 하도록 하겠다.

마지막으로 COM 포트를 본인 환경에 맞게 설정하고 Baud Rate 는 115200 으로 한다.

이제 준비가 다 끝났다. ESP8266 의 FLASH 단자를 GND 에 연결하고, RST 단자는 GND 에 잠시 붙였다가 떼자. 그리고나서 START 버튼을 누르면, 쭉~~ 파일이 올라간다.



누가 하드웨어 쟁이 아니랄까봐 100% 찍히도록 못 만들고 99% 에서 끝난다. 소프트웨어쟁이들은 왜 저렇게 나오는지 알 것이다.

이제 FLASH 단자를 GND 에서 제거하고, 시리얼 모니터를 열자. 그리고나서 Baud rate 를 다시 74880 으로 변경한다. 이제 RST 단자를 GND 에 붙였다가 떼어서 출력되는 값을 확인하자.

ets Jan 8 2013,rst cause:2, boot mode:(3,7)

load 0x40100000, len 2592, room 16

tail 0

chksum 0xf3

load 0x3ffe8000, len 764, room 8

tail 4

chksum 0x92

load 0x3ffe82fc, len 676, room 4

tail 0

chksum 0x22

csum 0x22

2nd boot version : 1.7(5d6f877)

SPI Speed : 40MHz

SPI Mode : QIO

SPI Flash Size & Map: 8Mbit(512KB+512KB)

jump to run user1 @ 1000

correct flash map

V2

Mo

Backup

⸮rf cal sector: 251

freq trace enable 0

rf[112] : 00

rf[113] : 00

rf[114] : 01

SDK ver: 3.0.5(b29dcd3) compiled @ Oct 9 2021 09:52:05

phy ver: 1156\_0, pp ver: 10.2

)⸮NOAT+GMR

AT version:1.7.5.0(Oct 9 2021 09:26:04)

SDK version:3.0.5(b29dcd3)

compile time:Oct 15 2021 18:05:30

Bin version(Wroom 02):1.7.5

OK

싱싱한 버전이 올라갔다. :)

이것으로 완료!!

**전 안되는데요???**

마지막 부분에서 기대했던 로그가 아니라 다음과 같은 거지같은 화면을 마주하고 있는 사람도 있을 것이다.

ets Jan 8 2013,rst cause:2, boot mode:(3,7)

load 0x40100000, len 2592, room 16

tail 0

chksum 0xef

load 0x88888888, len -2004318072, room 8

flash read err, ets\_unpack\_flash\_code

ets\_main.c

ets Jan 8 2013,rst cause:2, boot mode:(3,6)

load 0x40100000, len 2592, room 16

tail 0

chksum 0xef

load 0x88888888, len -2004318072, room 8

ets Jan 8 2013,rst cause:2, boot mode:(3,2)

load 0x40100000, len 2592, room 16

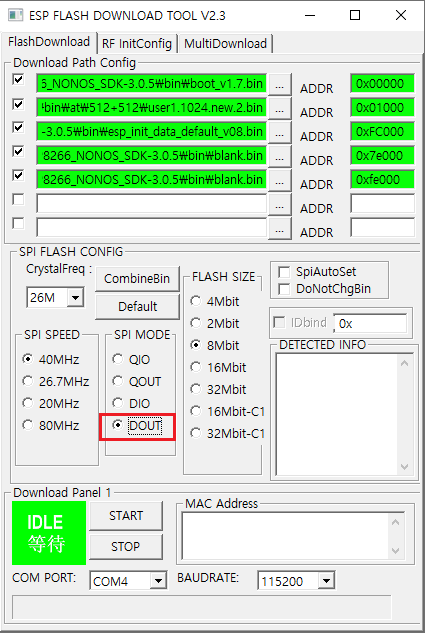
tail 0

chksum 0xef

load 0x88888888, len -2004318072, room 8

절망적인 것이다. 나도 잘 한다. 당해봤으니까..............

이 이유는 앞서 언급했던 SPI 모드 때문이다. 본래 ESP8266-01 의 스팩상 달려있는 플래시 메모리는 QIO 모드가 지원되어야 하는 것 같다. 그런데 ㅅㅂ 아닌 모듈이 있더라. 난 구매한 세 개 중에서 한 개가 그랬다. 이게 Quad IO 인지 Dual IO 인지 등등 그런건데, 나도 정확하게는 모른다. 중요한건지도 사실 잘 모르겠다. 대충 속도랑 관련 있는것 같은데, 해결 방법은 안다.



제일 구린,, DOUT 으로 하고 펌웨어를 올린 후에 해 보면 된다. 그러면 아래처럼 출력을 볼 수 있다.

슬프지만 ISP 부분을 보면 DOUT 이라고 마빡에 찍혀있다 -\_-;

ets Jan 8 2013,rst cause:2, boot mode:(3,7)

load 0x40100000, len 2592, room 16

tail 0

chksum 0xf3

load 0x3ffe8000, len 764, room 8

tail 4

chksum 0x92

load 0x3ffe82fc, len 676, room 4

tail 0

chksum 0x22

csum 0x22

2nd boot version : 1.7(5d6f877)

SPI Speed : 40MHz

SPI Mode : DOUT

SPI Flash Size & Map: 8Mbit(512KB+512KB)

jump to run user1 @ 1000

V2

Mo

Backup

yrf cal sector: 251

freq trace enable 0

rf[112] : 00

rf[113] : 00

rf[114] : 01

SDK ver: 3.0.3(8427744) compiled @ Mar 19 2020 18:11:05

phy ver: 1156\_0, pp ver: 10.2

)͈⸮

⸮⸮

AT+GMR

AT version:1.7.3.0(Mar 19 2020 18:15:04)

SDK version:3.0.3(8427744)

compile time:Apr 27 2020 10:06:00

Bin version(Wroom 02):3.0.3

OK

이게 중국산 짝퉁인건지, 잘못 만들어진건지 사기당한건지 모르겠지만, 이 녀석은 죽어도 DOUT 밖에 안되더라. DIO, QOUT 등등 다 해봤다. DOUT 하나밖에 안되더라. 수없이 많이 해 봤으므로 여러분은 삽질하지 않길 바란다.

이것으로 펌웨어 업그레이드 하는 설명을 마치도록 하겠다.

삽질을 그렇게 했음에도 아직 남는 궁금증이 있다. 왜 아두이노 우노의 경우에는 아두이노의 RESET 과 GND 를 연결한 상태로 유지하지 않는가? 안해도 동작하는걸까? 그리고 펌웨어를 꼭 업그레이드 해야만 AT 커맨드로 보드레이트를 9600 설정 가능한것처럼 하던데, 그냥 시리얼 모니터랑 ESP8266 이랑 연결한 상태에서 AT 커맨드로 보드레이트를 세팅하면 되는거 아닌가?

**연결에 대한 요약**

마지막 요약을 하자면,

1. 아두이노의 RESET 과 GND 를 연결한다.

2. 아두이노와 PC 를 USB 케이블로 연결한다.

3. ESP8266의 RESET 만 제외하고 나머지(FLASH 포함)을 모두 아두이노와 연결한다.

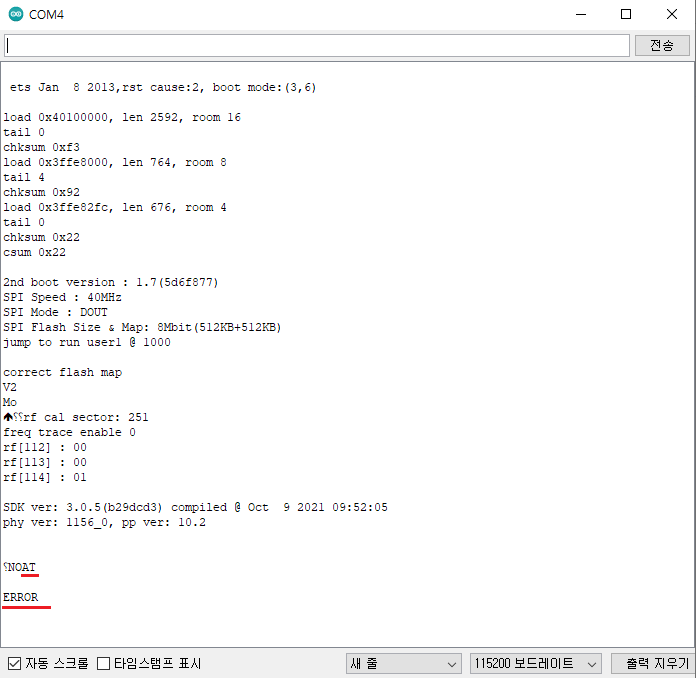
4. 펌웨어를 올리기 직전에 ESP8266 의 RESET 을 GND 와 1초간 연결했다가 끊는다.

5. 펌웨어를 올린다.(공식적으로는 다운로드라는 표현을 씀)

6. 완료되면, FLASH 단자의 연결을 제거하고 RESET 을 GND 와 약 1초간 연결했다 끊어서 초기화 한다.

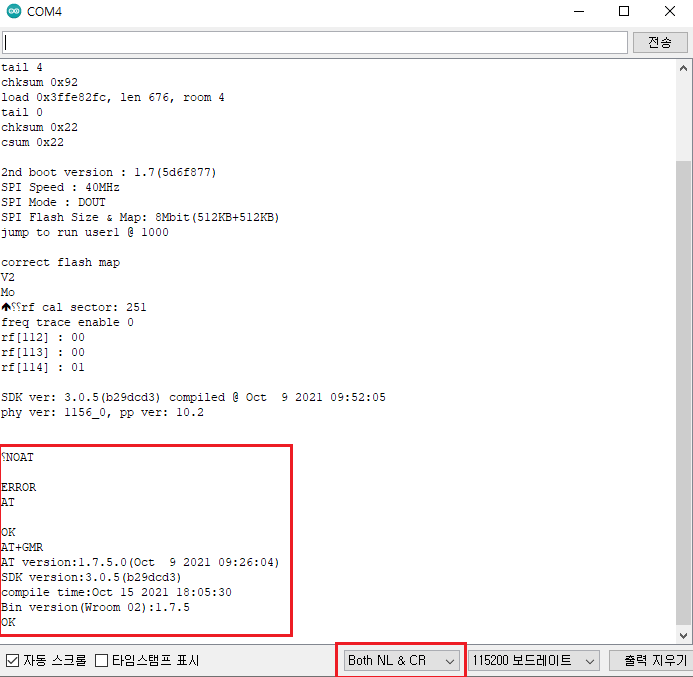
**기타 문제 상황**

다음날 AT 커맨드로 작업을 하려고 했는데 다음과 같은 상황이 벌어졌다.



AT 라고 날렸는데, ERROR 이 찍혔다. 뭐지?????????????

이때는 반드시 하단의 new line 정책을 확인하자. '새 줄' 로 되어 있어서 그렇다.



'Both NL & CR' 로 변경하고 나니 잘 동작한다.

아무튼 내가 삽질한 결과는 여기까지다. :) 많은 도움이 되었길 바란다.