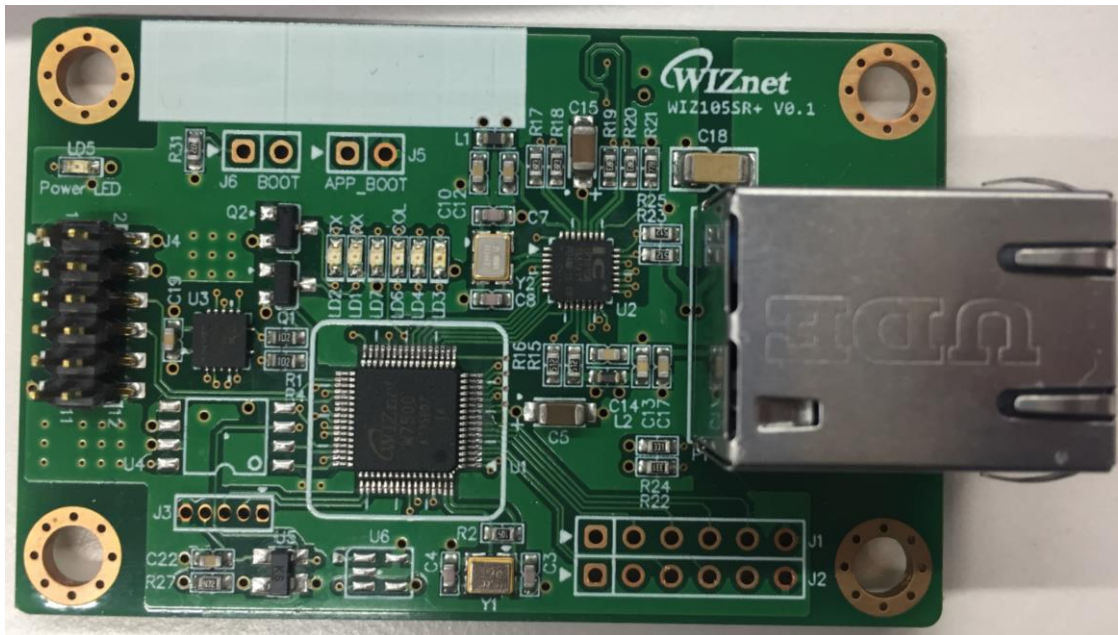


WIZ750SR-105 Production Guide

(Version 1.0)



2018-01-15

©2016 WIZnet Co., Inc. All Rights Reserved.

For more information, visit our website at <http://www.wiznet.io>

목차

1. 작업 환경 구성.....	3
1. 소프트웨어 환경	3
2. WIZwiki 보드 mbed COM Port 확인	3
3. 프로그램 설치	4
2. 하드웨어 연결.....	8
1. 테스트 환경 준비 및 하드웨어 연결.....	8
3. 펌웨어, 맥 어드레스 라이팅 및 기능 테스트	13
1. 펌웨어 라이팅	13
2. 맥 어드레스 라이팅	15
3. 기능 테스트.....	16
4. 맥 어드레스 잘 못 입력했을 경우	17
5. 모듈에서 Error가 발생했을 때	19
6. WIZ750SR-105 생산 과정 요약	22

1. 작업 환경 구성

1. 소프트웨어 환경

1. PC (Window 7 이상)
2. Python 2.7.13 program (파일로 제공)

파일이 없을 시 아래 설명 확인.

- i. (<https://www.python.org/downloads/release/python-2713/>) 링크로 접속, Python 2.7.13 Windows x86-64 MSI installer 다운로드 (PC가 64bit일 경우)
- ii. (32bit PC라면 Windows x86 MSI installer를 다운로드)
3. Mbed Serial Driver (파일로 제공)

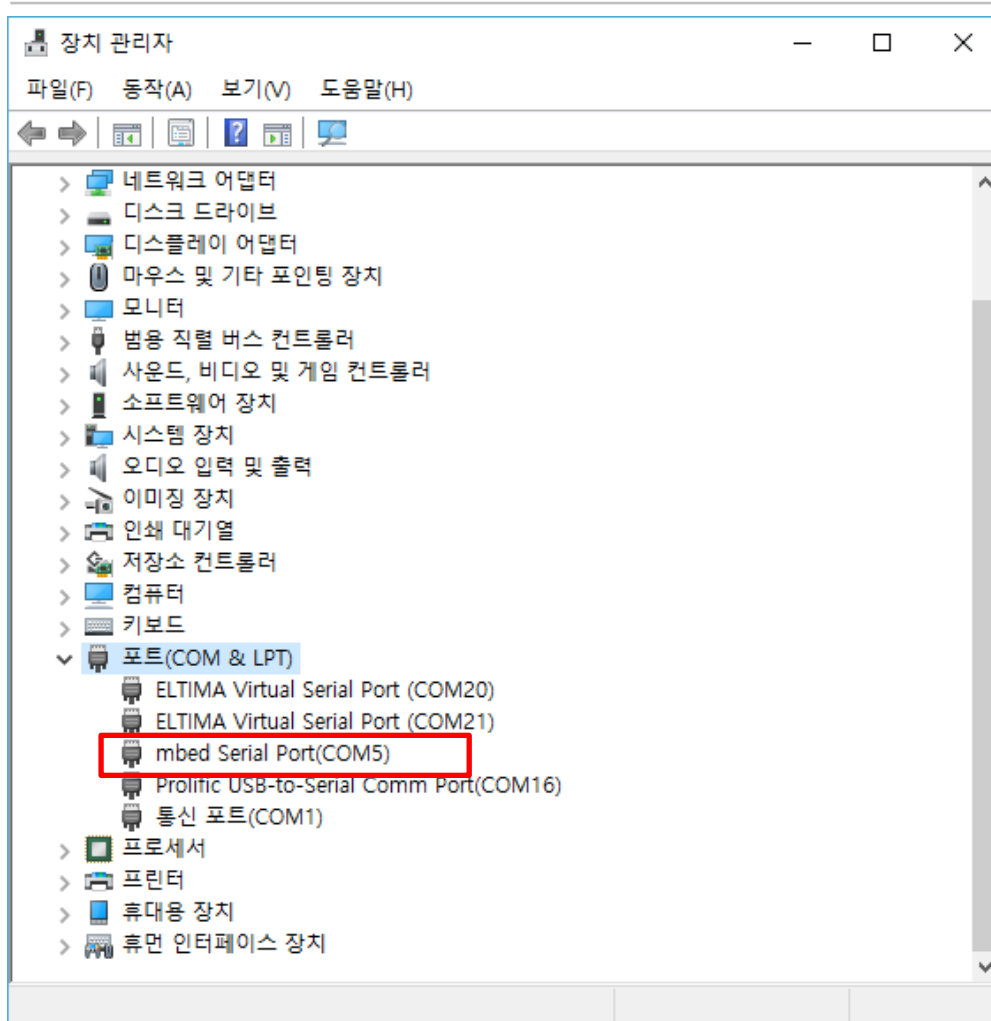
파일이 없을 시 아래 설명 확인.

- i. (<https://os.mbed.com/handbook/Windows-serial-configuration>) 링크로 접속.

Download the mbed Windows serial port driver의 "Download latest driver" 클릭하여 다운로드

2. WIZwiki 보드 mbed COM port 확인

1. WIZwiki-W7500P에 USB 케이블을 연결한 뒤, PC의 장치관리자에 들어가서 "mbed Serial Port"가 나오는지 확인한다."



3. 프로그램 설치

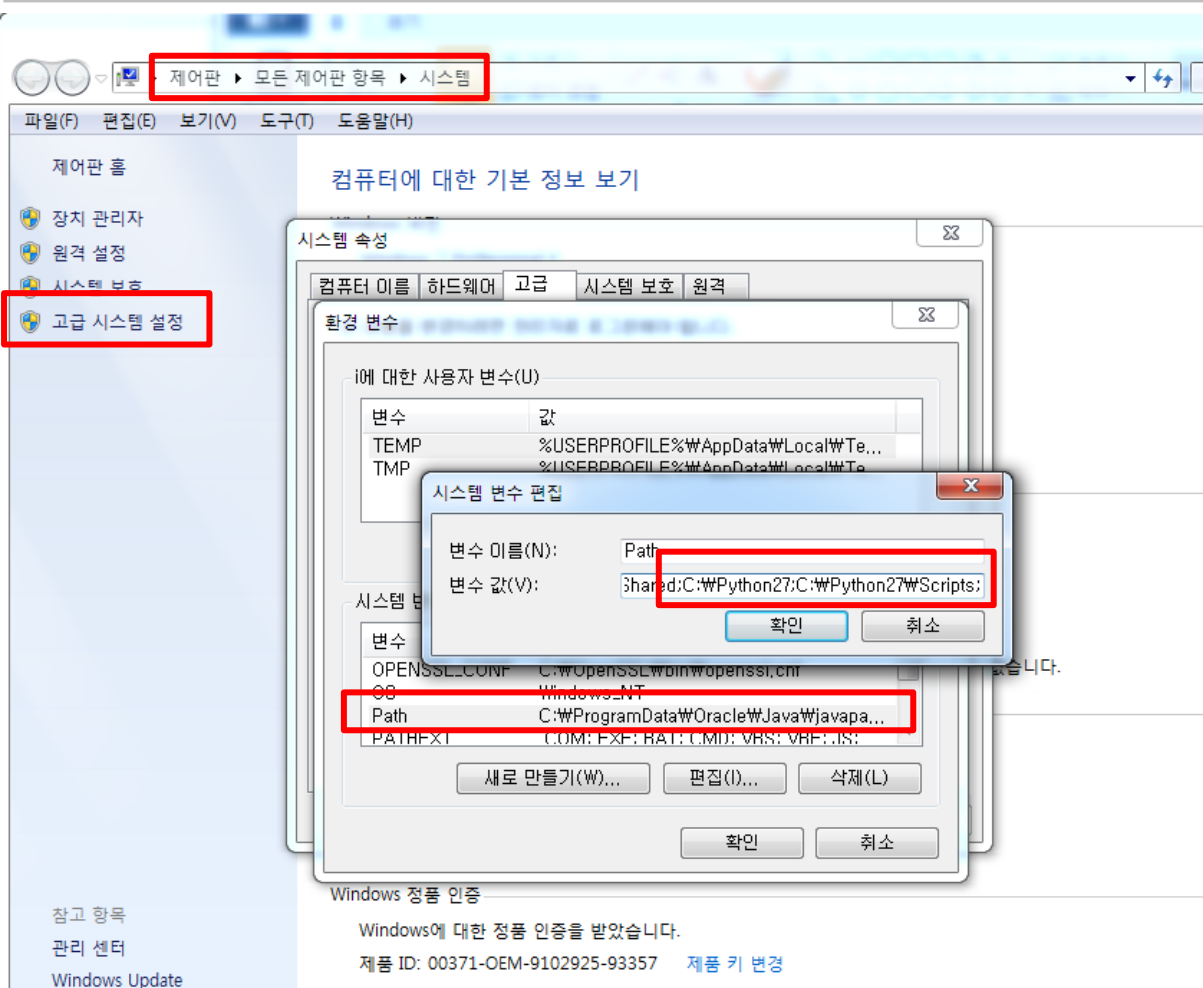
1. Python 2.7.13 프로그램 설치

설치파일 실행 후 전부 "Next" 누르고 설치.

2. 환경변수 추가하기.

아래 사진과 같이 제어판 -> 시스템 -> 고급 시스템 설정 -> 환경변수로 접속한다.

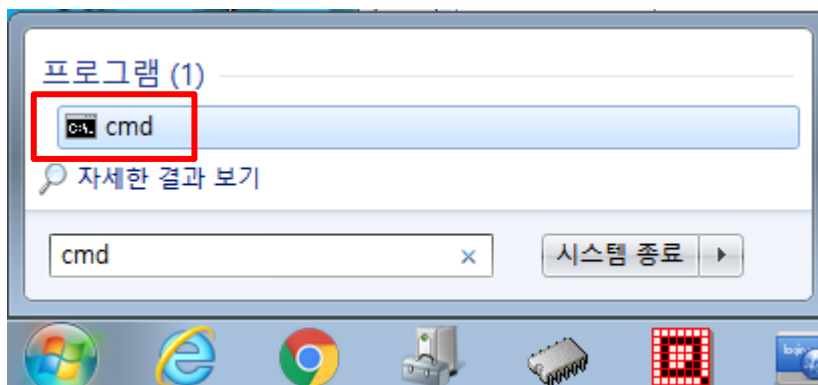
그리고 시스템 변수의 Path에 **C:\Python27;C:\Python27\Scripts** 을 기입한 뒤 확인을 누르고 나온다.



3. PyOCD 설치하기

- i. 아래 사진과 같이 시작메뉴의 찾기 메뉴에서 명령 프롬프트(cmd)를 실행한다.

(주의: 반드시 인터넷이 연결되어 있어야 함.)



- ii. cmd환경에서 `pip install --pre -U pyocd` 명령어를 입력한 뒤 enter를 누른다.

그러면 아래 사진 처럼 pyOCD가 설치 된다.

```
C:\Users\Wi>pip install --pre -U pyocd
Collecting pyocd
  Downloading pyOCD-0.8.1a1.tar.gz (441kB)
    100% |#####| 450kB 930kB/s
Collecting intelhex (from pyocd)
  Downloading intelhex-2.1-py2.py3-none-any.whl (46kB)
    100% |#####| 51kB 3.4MB/s
Collecting six (from pyocd)
  Downloading six-1.11.0-py2.py3-none-any.whl
Collecting enum34 (from pyocd)
  Downloading enum34-1.1.6-py2-none-any.whl
Collecting future (from pyocd)
  Downloading future-0.16.0.tar.gz (824kB)
    100% |#####| 829kB 819kB/s
Collecting websocket-client (from pyocd)
  Downloading websocket_client-0.46.0-py2.py3-none-any.whl (200kB)
    100% |#####| 204kB 939kB/s
Collecting pywinusb<=0.4.0 (from pyocd)
  Downloading pywinusb-0.4.2.zip (61kB)
    100% |#####| 71kB 1.2MB/s
Installing collected packages: intelhex, six, enum34, future, websocket-client,
pywinusb, pyocd
  Running setup.py install for future ... done
  Running setup.py install for pywinusb ... done
  Running setup.py install for pyocd ... done
Successfully installed enum34-1.1.6 future-0.16.0 intelhex-2.1 pyocd-0.8.1a1 pyw
inusb-0.4.2 six-1.11.0 websocket-client-0.46.0

C:\Users\Wi>
```

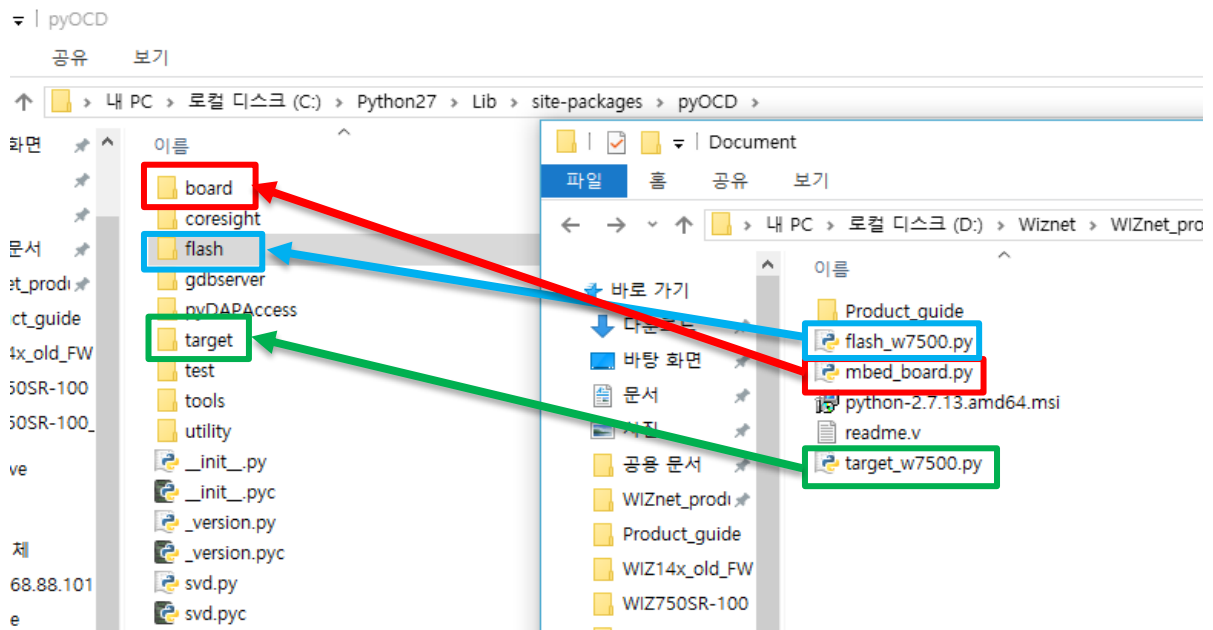
- iii. 그리고 제공해 준 파일을 아래 사진과 같이 아래의 경로에 각각 복사한다.

C:\Python27\Lib\site-packages\pyOCD

Flash_w7500.py >> Flash 폴더

Mbed_board.py >> board 폴더

Target_w7500.py >> target 폴더

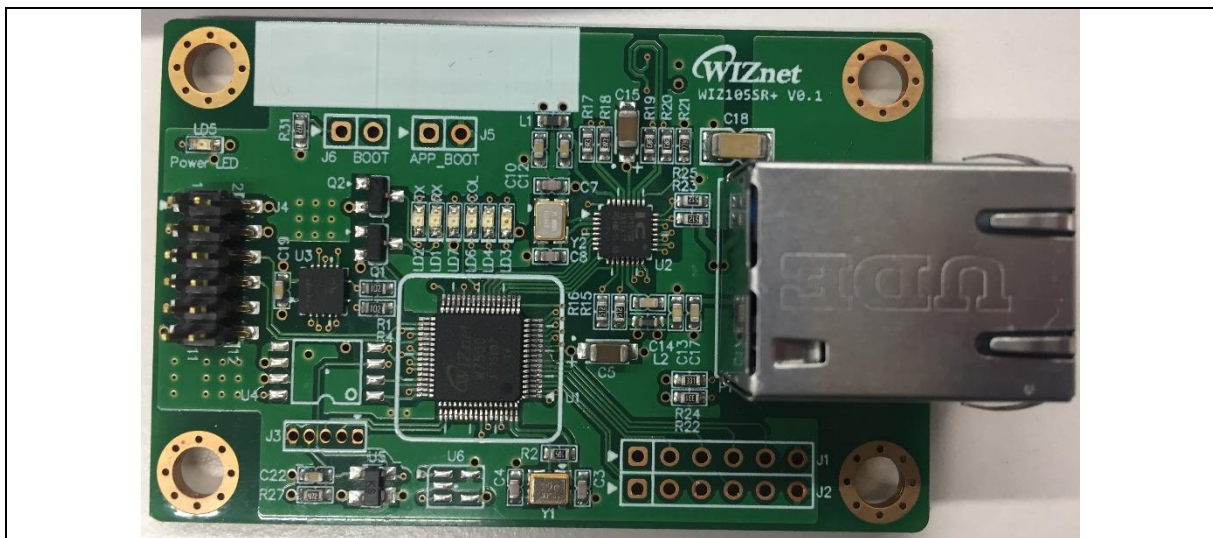


2. 하드웨어 연결

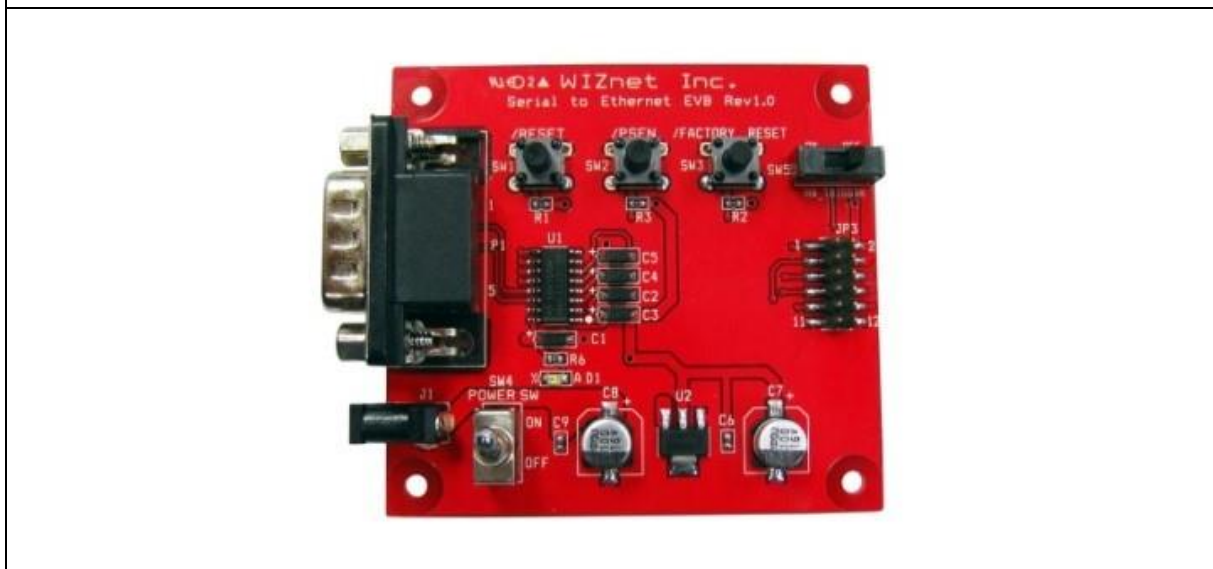
1. 테스트 환경 준비 및 하드웨어 연결

- 테스트 환경

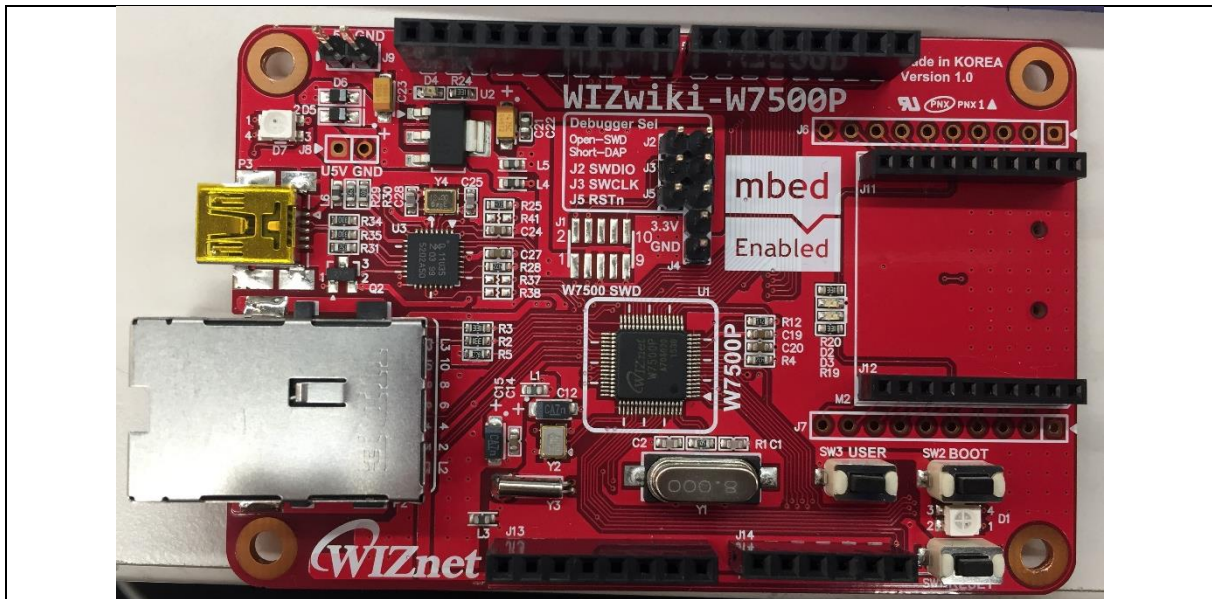
1. WIZwiki-W7500P & WIZwiki-W7500 board (둘 중에 하나 사용)
2. WIZ750SR-105
3. WIZ105SR-EVB Board



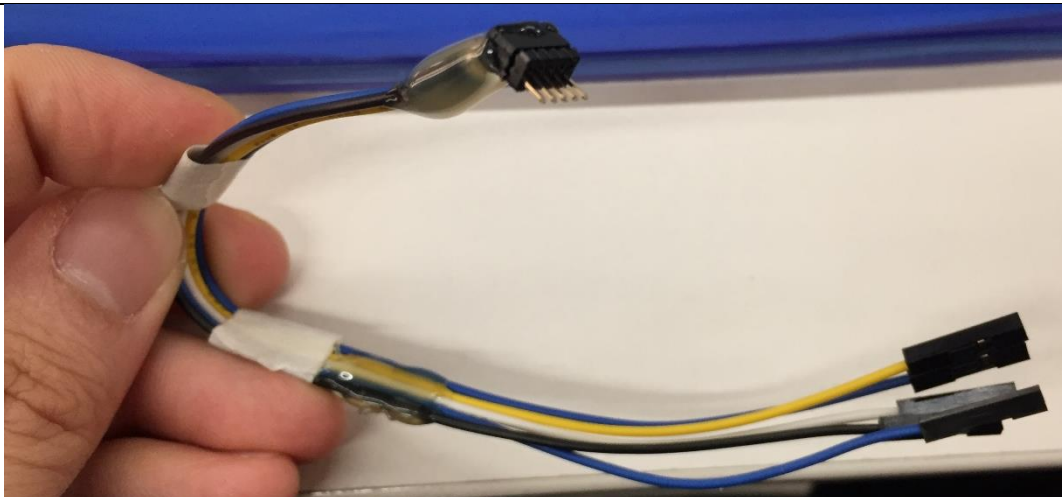
WIZ750SR-100 보드



WIZ105SR-EVB



WIZwiki-W7500 보드



2.54mm to 1.27mm 생산용 Cable

4. 2.54mm to 1.27mm 생산용 Cable 필요. (반드시 있어야 함.)

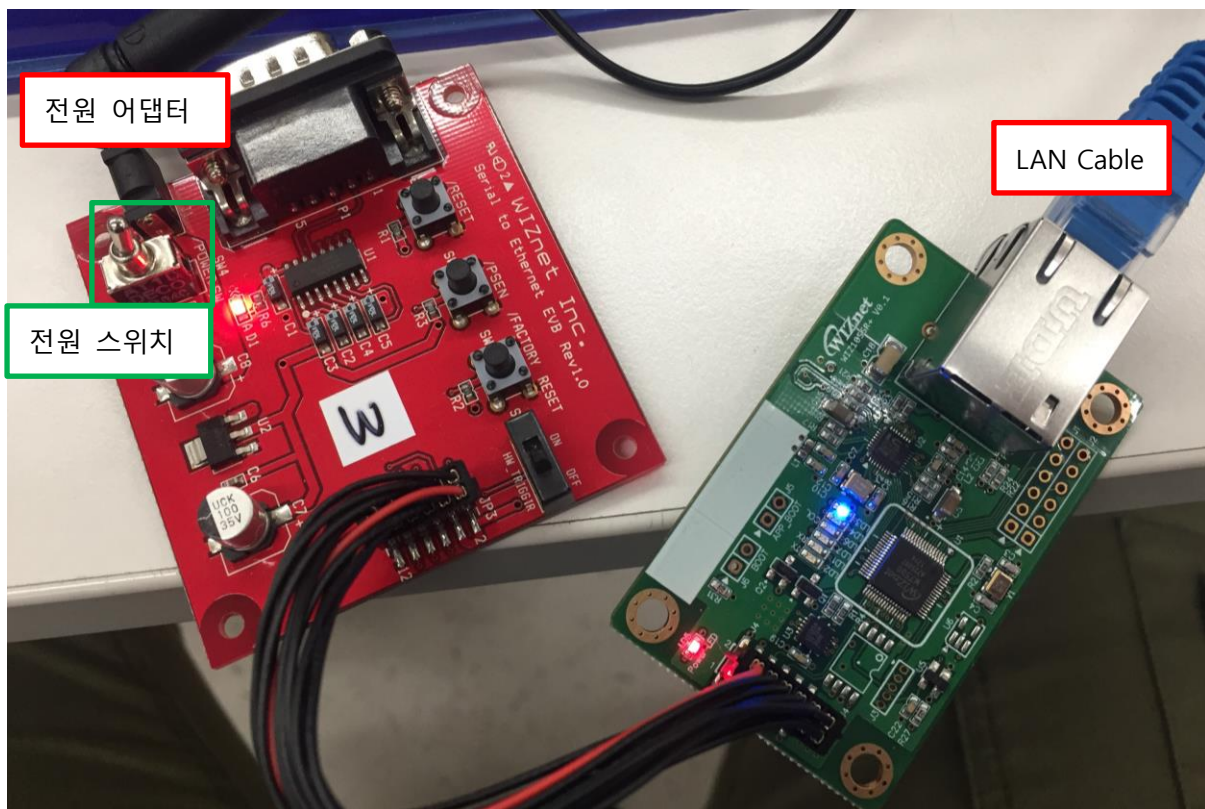
A. PIN 1번은 VCC (빨간색으로 색칠되어 있음.)

5. LAN 케이블 1EA (기능 테스트 시 필요)

6. USB mini B Type 케이블 1EA (WIZwiki-W7500에 연결)

- 하드웨어 연결

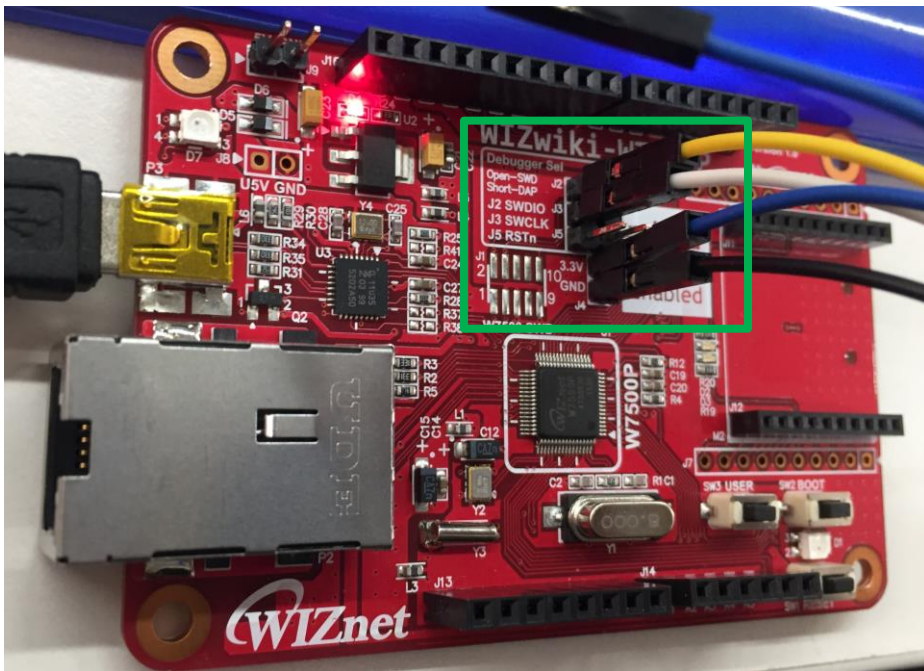
1. WIZ750SR-105을 WIZ105SR-EVB에 장착한다.



2. WIZ750SR-105에 LAN 케이블만 장착한다. (전원 어댑터는 장착하지 않는다.)
3. 아래 사진과 같이 점퍼를 WIZ105SR-EVB의 D-SUB9커넥터의 2번(RX), 3번(TX)에 연결한다.



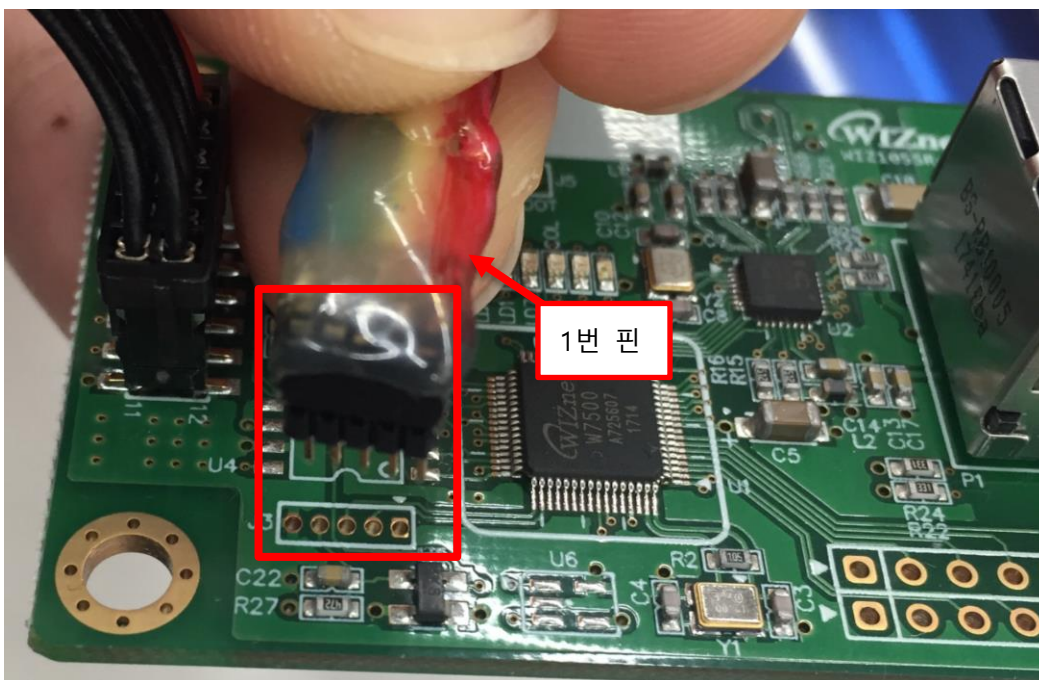
4. 아래 사진과 같이 mini USB B 케이블을 연결하고, WIZwiki-W7500P의 J2, J3, 3.3V, GND에 2.54mm to 1.27mm 생산용 Cable을 연결한다.



- 아래 사진과 같이 2.54mm to 1.27mm 생산용 Cable을 WIZ750SR-100의 J4(JTAG)에 연결한다.

(주의: 핀에 맞게 연결해야 한다. J4의 1번 핀에 케이블 1번핀이 연결되어야 한다.)

해당 케이블은 보드와 단순 접촉을 해야 하기 때문에, 케이블을 보드에 그대로 닿은 형태로 사용된다. 그래서 케이블을 세게 누르면 케이블이 망가질 수 있다.)



6. 정상적으로 연결 되었다면, 아래 사진과 같이 LD1 LED에 빨간 불이 점등된다.
7. 이 상태에서 펌웨어, 맥 어드레스 라이팅, 기능테스트를 진행한다.



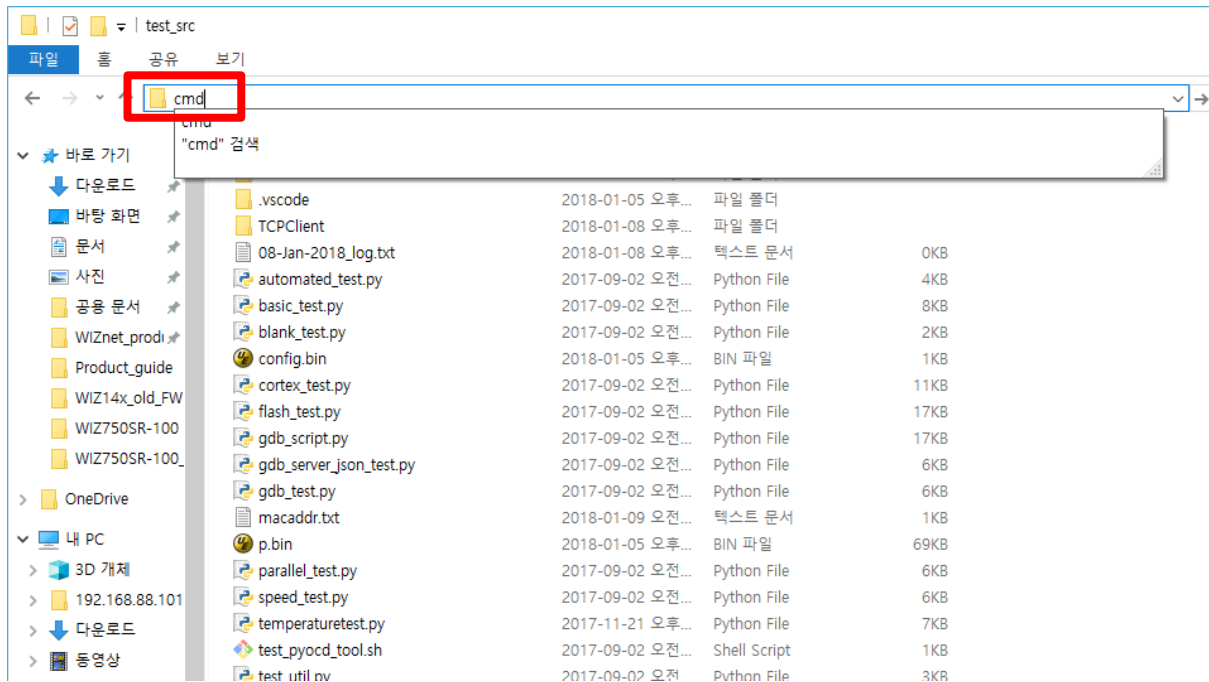
3. 펌웨어, 맥 어드레스 라이팅 및 기능 테스트

1. 펌웨어 라이팅

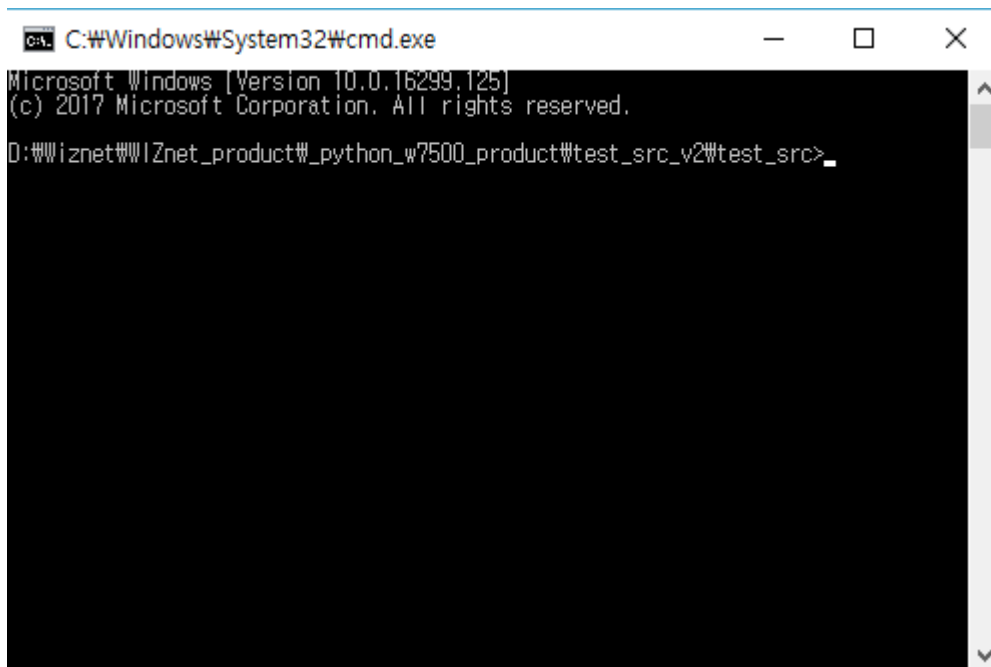
1. 펌웨어 다운로드 폴더로 접속한다. (제공한 파일 중 펌웨어 파일)

Ex) D:\Wiznet\WIZnet_product\python_w7500_product\test_src_v2\test_src

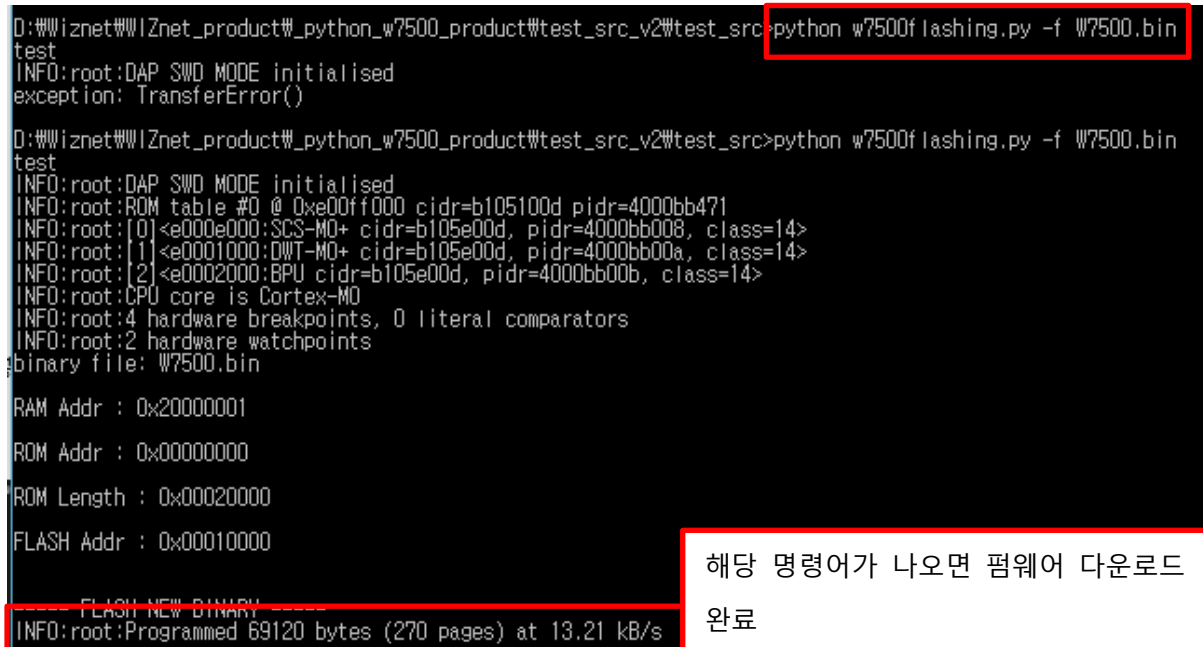
2. 아래 사진과 같이 주소 표시줄에 cmd를 입력 후 Enter를 한다.



3. 그림 아래 사진과 같이 cmd화면이 실행된다.



4. `python w7500flashing.py -f W7500.bin` 명령어를 입력하면 펌웨어 다운로드 시작한다.

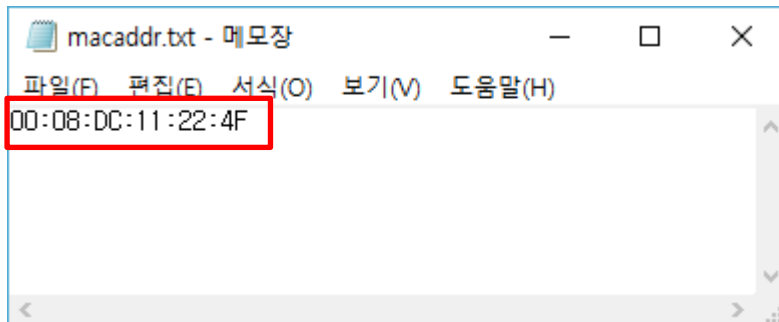


2. 맥 어드레스 라이팅

1. 맥 어드레스 입력은 폴더의 macaddr.txt를 수정하여 MAC address를 수정할 수 있다.

아래 사진과 같이 맥 어드레스를 수정할 수 있다.

수정 후 저장한 뒤에 문서를 닫는다.



2. `python w7500dataflashing.py` 명령어를 입력하면 macaddr.txt에서 수정했던 맥 어드레스가 라이팅 된다.

라이팅 된 후 자동 1씩 증가한다.

```
D:\WIZnet\WIZnet_product\python_w7500_product\test_src_v2\test_src>python w7500dataflashing.py
None
test
INFO:root:SWD MODE initialised
INFO:root:ROM table #0 @ 0xe00ff000 cidr=b105100d pidr=4000bb471
INFO:root:[0]<e000e000:SCS-M0+ cidr=b105e00d, pidr=4000bb008, class=14>
INFO:root:[1]<e0001000:DWT-M0+ cidr=b105e00d, pidr=4000bb00a, class=14>
INFO:root:[2]<e0002000:BPU cidr=b105e00d, pidr=4000bb00b, class=14>
INFO:root:CPU core is Cortex-M0
INFO:root:4 hardware breakpoints, 0 literal comparators
INFO:root:2 hardware watchpoints
binary file: config.bin
bytebuffer length: 512

RAM Addr : 0x20000001
ROM Addr : 0x00000000
ROM Length : 0x00020000

----- MAC Address Update
'00:08:DC:11:22:4E'
mac: '0008DC11224E'
int_mac: 38051848782L
00 08 DC 11 22 4E
----- TEST PROGRAM/ERASE PAGE -----
TEST PASSED
```

3. 여기까지 완료되었으면 라이팅 끝.

3. 기능 테스트

1. `python wiz750_multiple_test.py -r 1` 명령어를 입력하면 아래 사진처럼 기능 테스트가 실행된다. 기능테스트 시 맥 어드레스도 확인 가능하다.

(주의: 반드시 WIZ105SR-EVB가 Loopback 형태로 구성되어 있어야 한다. (D-SUB9포트 TX/RX 연결되어 있어야 함.))

```

D:\WIZnet\WIZnet_product#\python_w7500_product#\test_src_v2#\test_src>python
wiz750_multiple_test.py -r 1
1 devices are detected
=====
mac addr: 00:08:DC:11:22:75
=====
thread for '192.168.11.50' is sta
socket.socket() called
['192.168.11.50'] is OPEN
['192.168.11.50'] is CONNECTED
[192.168.11.50] sent Hello WIZ750SR
[192.168.11.50] received Hello WIZ750SR
[192.168.11.50] 15 Jan 2018 15:26:34: success, success rate: 100.00%, [1/1]

thread for '192.168.11.50' is shutdowning
[192.168.11.50] stopped at 15 Jan 2018 15:26:34
Total try: 1
Success count: 1
Fail count: 0
Success Rate: 100.0%
=====
all threads are dead
[['MA', '00:08:DC:11:22:75'], ['PW', ''], ['FR', ''], ['RT', '']]
D:\WIZnet\WIZnet_product#\python_w7500_product#\test_src_v2#\test_src>

```

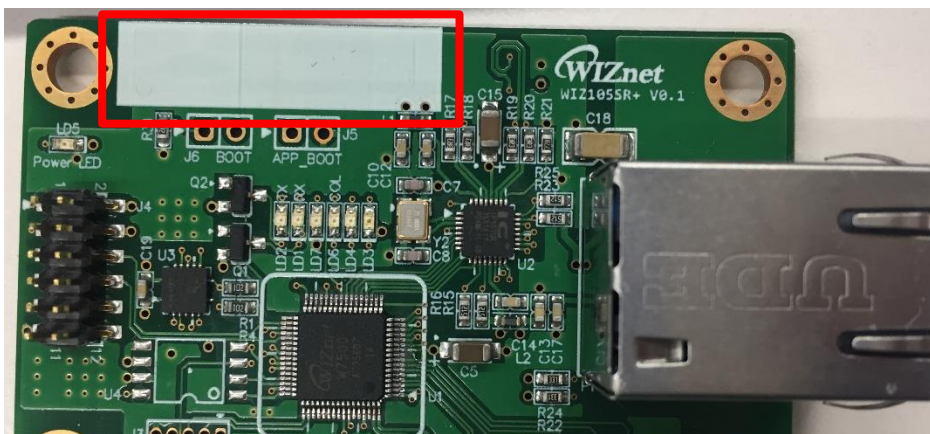
모듈에 라이팅 된 맥 어드레스

해당 데이터로 맥 어드레스 확인 가능

해당 명령어에 Success가 되면 통과

4. 위와 같이 MAC 정상적으로 출력되었다면, 생산 끝. 전원을 OFF한다.

5. 테스트가 완료되었으니, 아래의 빨간 테두리에 맥 어드레스 스티커를 모듈에 부착한다.

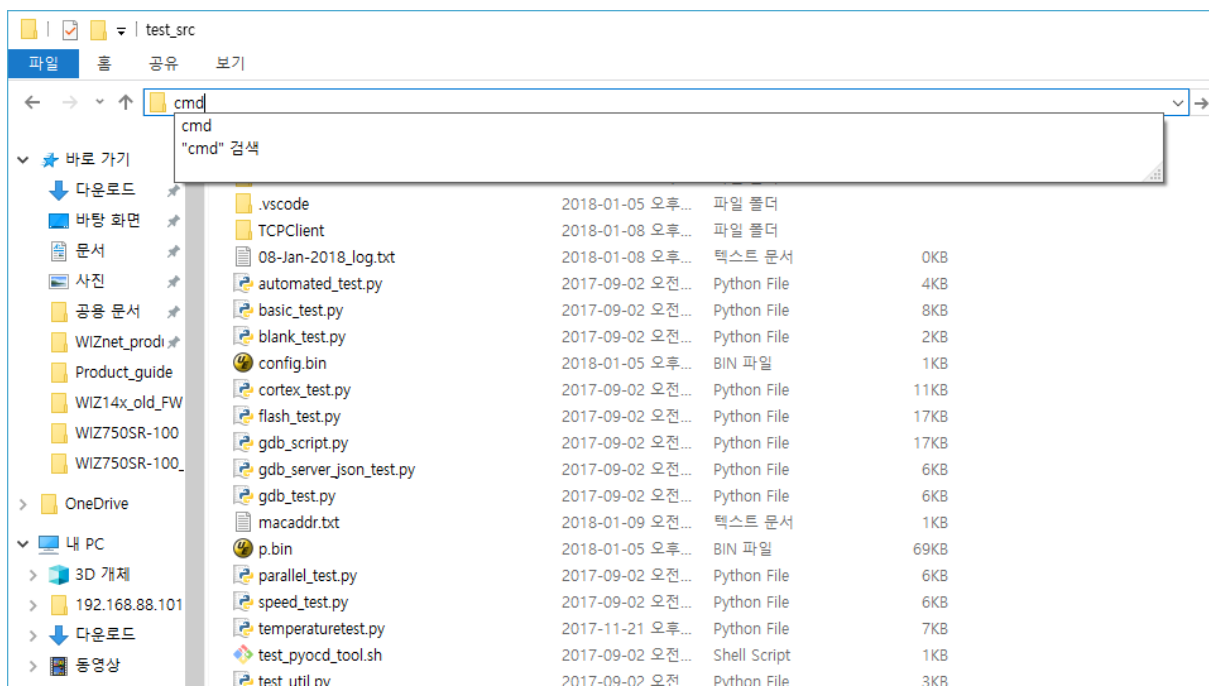


4. 맥 어드레스 잘 못 입력했을 경우

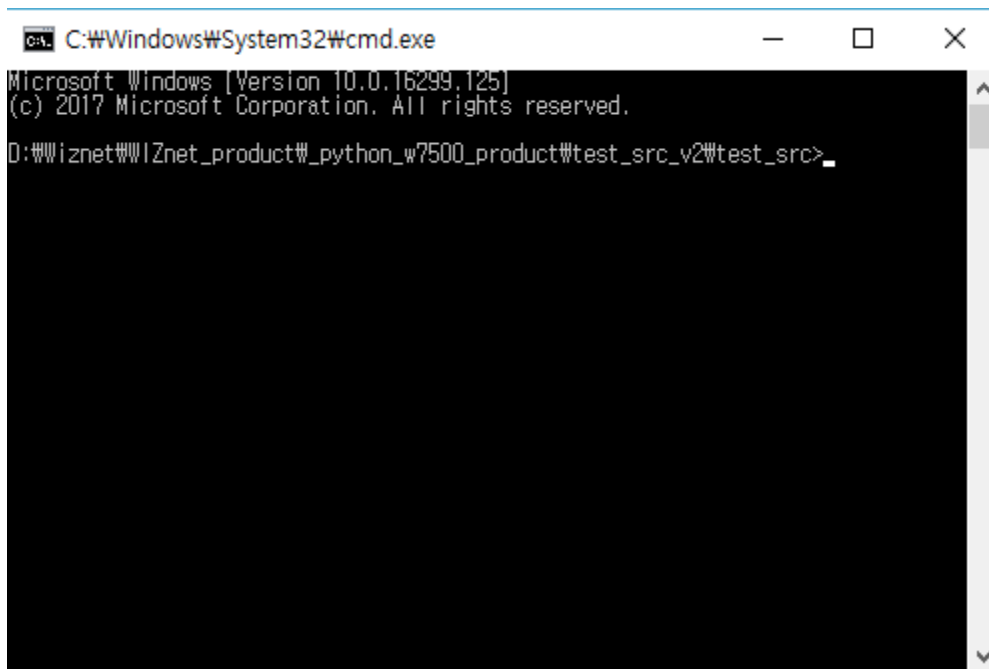
1. 2. 하드웨어 연결에서 2. 2.54mm to 1.27mm 생산용 Cable을 WIZ750SR-105의 J4(JTAG)에 연결한다.
2. 펌웨어 다운로드 폴더로 접속한다. (제공한 파일 중 펌웨어 파일)

Ex) D:\Wiznet\WIZnet_product\python_w7500_product\test_src_v2\test_src

3. 아래 사진과 같이 주소 표시줄에 cmd를 입력 후 Enter를 한다.

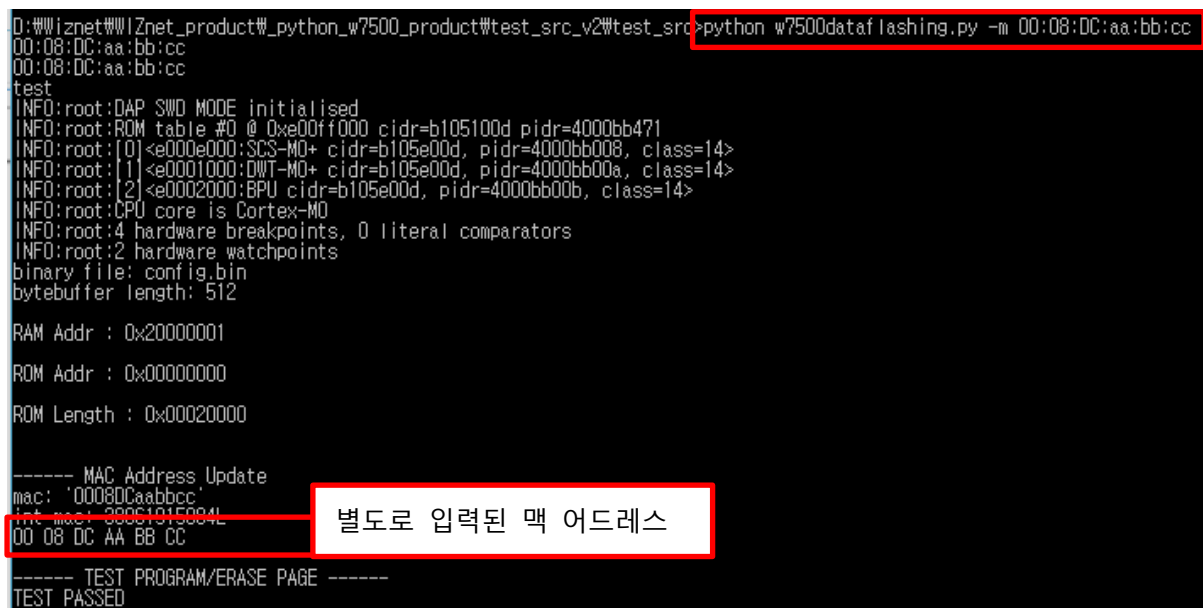


4. 그림 아래 사진과 같이 cmd화면이 실행된다.



5. `python w7500dataflashing.py -m 00:08:DC:xx:xx:xx` 명령어를 입력하면 입력한 맥 어드레스를 입력한다.

(주의: 해당 명령어는 `macaddr.txt`파일에 영향을 주지 않는다. 별도로 맥 어드레스를 입력할 때 사용한다.)



5. 모듈에서 Error가 발생했을 때

1. 아래 사진과 같이 에러가 발생한다면, 다음과 같이 조치하면 된다.
 - A. Python W7500flashing.py(펌웨어 라이팅)할 때 발생하는 문제
 - i. 다시 한 번 라이팅 해본다.
 - ii. 동일한 에러가 발생한다면, WIZwiki-W7500의 전원 OFF후 다시 ON한다.
 - iii. 마찬가지로 W7500P-EVB보드도 전원 OFF후 다시 ON한다.

```
PS D:\Wiznet\WIZnet_product\python_w7500_product\test_src_v2\test_src> python w7500flashing.py -f W7500x_S2E_Production.bin
test
INFO:root:DAP SWD MODE initialised
INFO:root:ROM table #0 @ 0xe00ff000 cidr=b105100d pidr=4000bb471
INFO:root:[0]<e000e000:SCS-M0+ cidr=b105e00d, pidr=4000bb008, class=14>
INFO:root:[1]<e0001000:DWT-M0+ cidr=b105e00d, pidr=4000bb00a, class=14>
INFO:root:[2]<e0002000:BPU cidr=b105e00d, pidr=4000bb00b, class=14>
INFO:root:CPU core is Cortex-M0
INFO:root:4 hardware breakpoints, 0 literal comparators
INFO:root:2 hardware watchpoints
binary file: W7500x_S2E_Production.bin

RAM Addr : 0x20000001

ROM Addr : 0x00000000
Traceback (most recent call last):
  File "w7500flashing.py", line 286, in <module>
    basic_test(None, file)
  File "w7500flashing.py", line 73, in basic_test
    print "\r\nROM Length : " + '0x{0:08X}'.format(rom_region.length)
IOError: [Errno 0] Error
```

- B. Python w7500dataflashing(맥 어드레스 라이팅)할 때 발생하는 문제.
 - i. WIZwiki-W7500의 전원 OFF후 다시 ON한다.
 - ii. 마찬가지로 W7500P-EVB보드도 전원 OFF후 다시 ON한다.

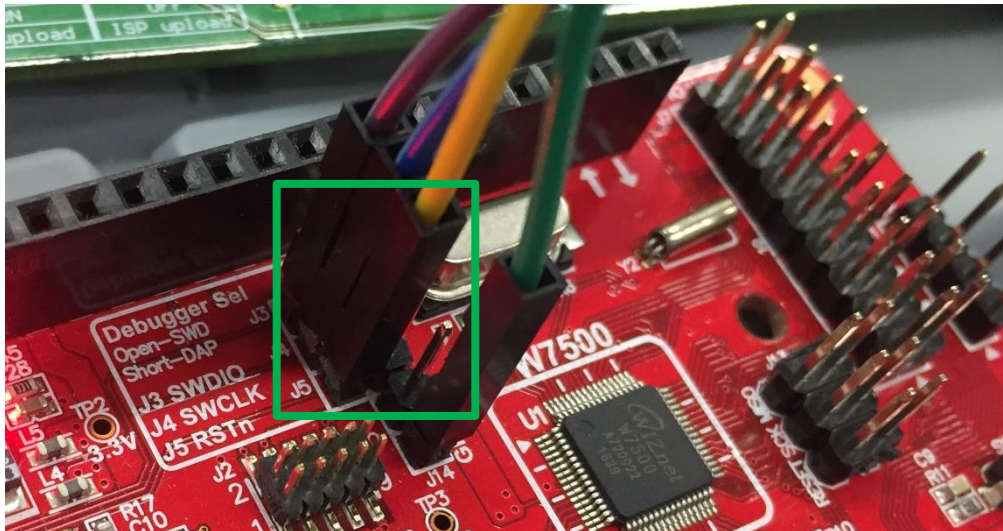
```
D:\Wiznet\WIZnet_product\python_w7500_product\test_src_v2\test_src>python w7500dataflashing.py
None
test
INFO:root:DAP SWD MODE initialised
exception: TransferError()

D:\Wiznet\WIZnet_product\python_w7500_product\test_src_v2\test_src>

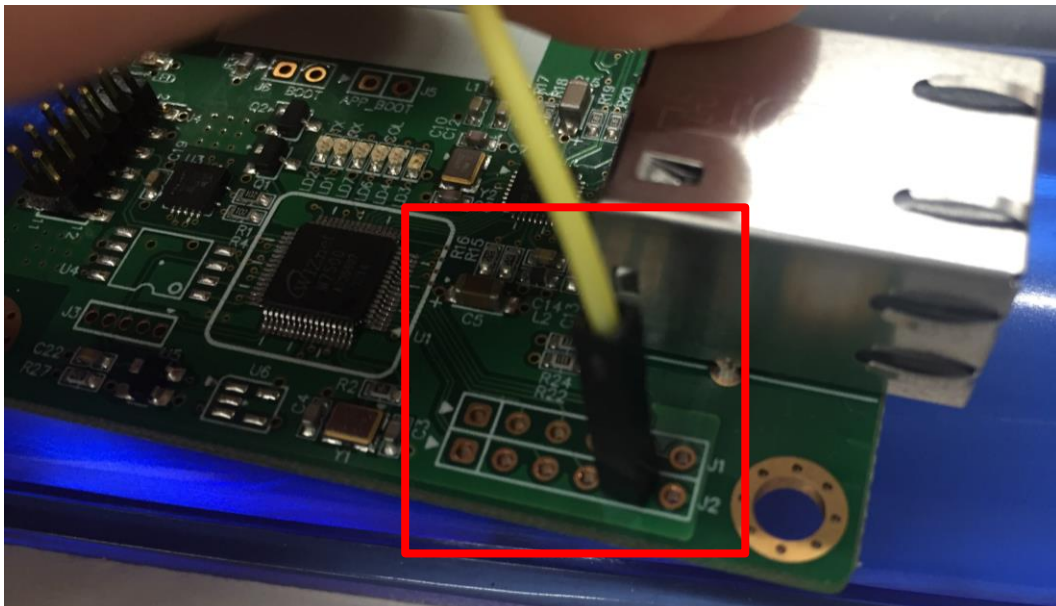
D:\Wiznet\WIZnet_product\python_w7500_product\test_src_v2\test_src>python w7500dataflashing.py
None
test
INFO:root:DAP SWD MODE initialised
```

iii. 그럼에도 동일하다면 아래와 같은 방법을 사용해보자.

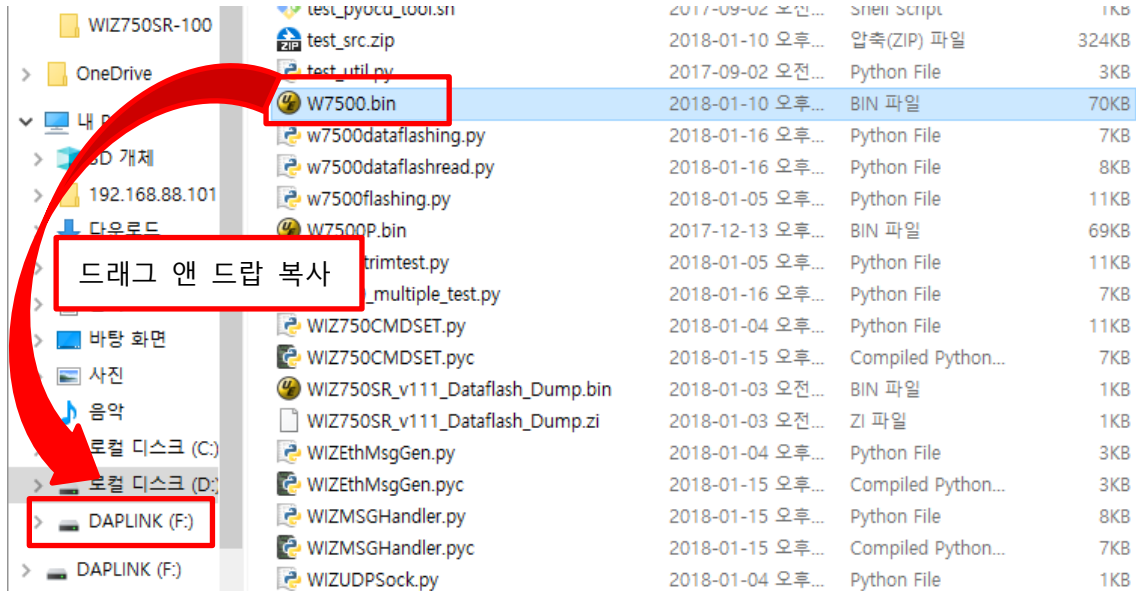
1. 아래 사진과 같이 WIZwiki-W7500의 J5에 점퍼 케이블을 연결한다.



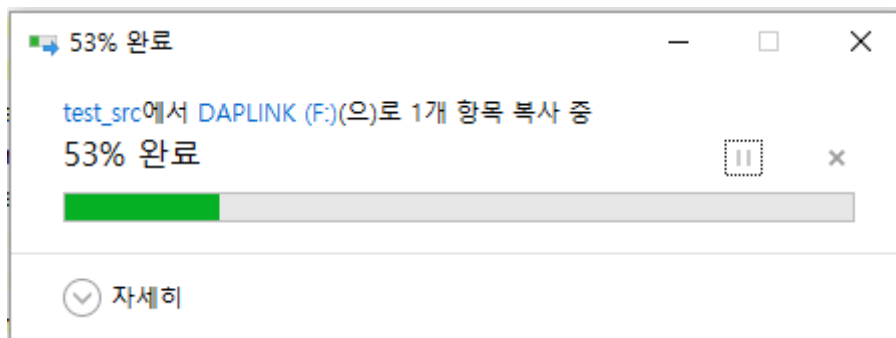
2. 그리고 해당 점퍼 케이블을 아래 사진과 같이 WIZ750SR-105의 J2의 5번 핀의 Reset에 연결한다.



3. 펌웨어 바이너리 파일을 WIZwiki-W7500의 이동식디스크에 **드래그 앤 드랍**으로 복사한다.



4. 그럼 아래 사진과 같이 다운로드가 시작된다.



5. 여기까지 완료되었다면, 다시 정상 동작한다.

6. WIZ750SR-105 생산 과정 요약

해당 생산 과정 요약본은 프로그램이 전부 설치되었다는 기준으로 설명한다.

1. WIZ750SR-105을 WIZ105SR-EVB에 장착한다.
2. WIZ105SR-EVB에 LAN 케이블만 장착한다. (전원 어댑터는 장착하지 않는다.)
3. 점퍼를 WIZ105SR-EVB의 D-SUB9커넥터의 2번(RX), 3번(TX)에 연결한다.
4. WIZwiki-W7500P보드에 mini USB B 케이블을 연결하고, J2, J3, 3.3V, GND에 2.54mm to 1.27mm 생산용 Cable을 연결한다.
5. 2.54mm to 1.27mm 생산용 Cable을 WIZ750SR-105의 J4(JTAG)에 연결한다.

(주의: 핀에 맞게 연결해야 한다. J4의 1번 핀에 케이블 1번핀이 연결되어야 한다.)

해당 케이블은 보드와 단순 접촉을 해야 하기 때문에, 케이블을 보드에 그대로 닿은 형태로 사용된다. 그래서 케이블을 세게 누르면 케이블이 망가질 수 있다.)

6. 펌웨어 다운로드 폴더로 접속한다. (제공한 파일 중 펌웨어 파일)
7. 아래 사진과 같이 주소 표시줄에 cmd를 입력 후 Enter를 한다.
8. `python w7500flashing.py -f W7500.bin` 명령어를 입력하면 펌웨어 다운로드 시작한다.
9. 맥 어드레스 입력은 폴더의 macaddr.txt를 수정하여 MAC address를 수정할 수 있다.
10. `python w7500dataflashing.py` 명령어를 입력하면 macaddr.txt에서 수정했던 맥 어드레스가 라이팅 된다.
11. `python wiz750_multiple_test.py -r 1` 명령어를 입력하면 기능 테스트가 실행된다. 기능테스트 시 맥 어드레스도 확인가능하다.

(주의: 반드시 WIZ105SR-EVB가 Loopback 형태로 구성되어 있어야 한다. (D-SUB9포트 TX/RX 연결되어 있어야 함.))

12. 테스트가 완료되었으니, 맥 어드레스 스티커를 모듈에 부착한다.