



CodeZoo CATM1 Hands-On

- Raspberry Pi (with PPP)-

version 1.0

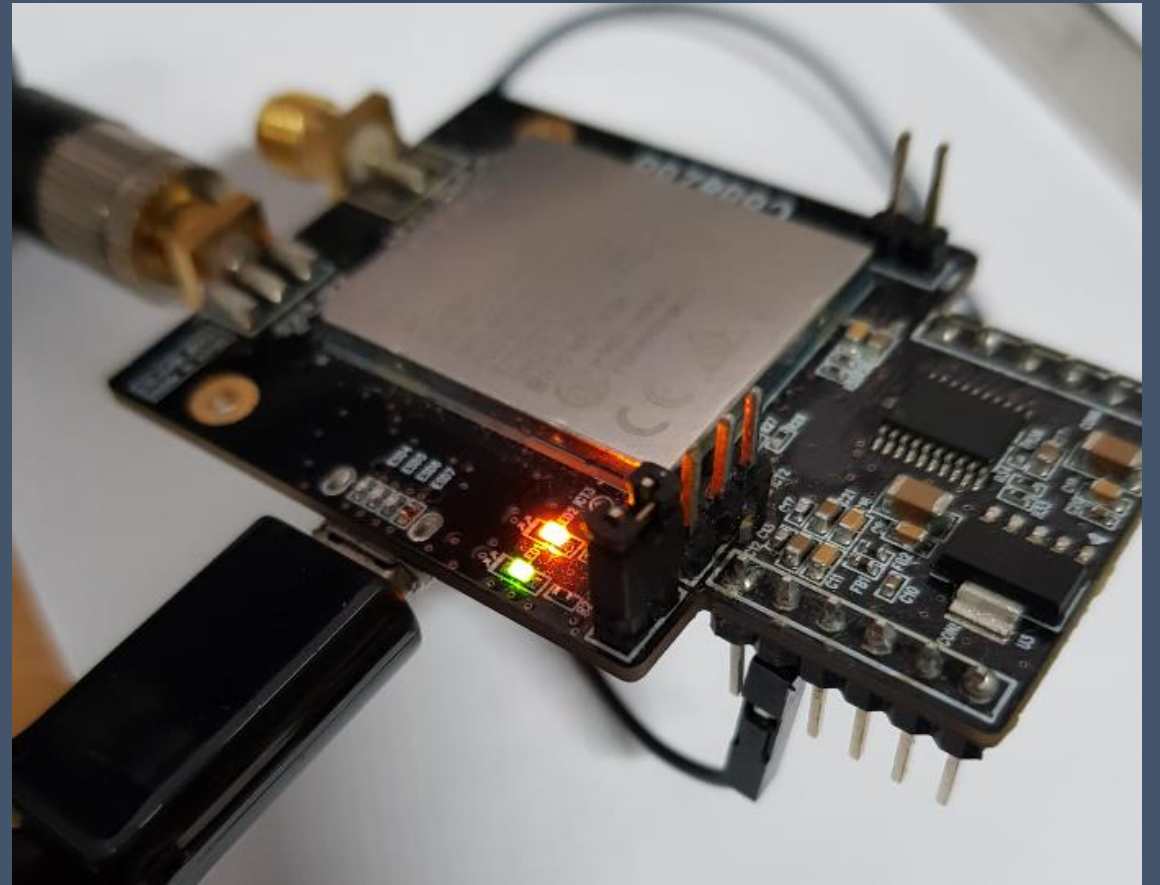
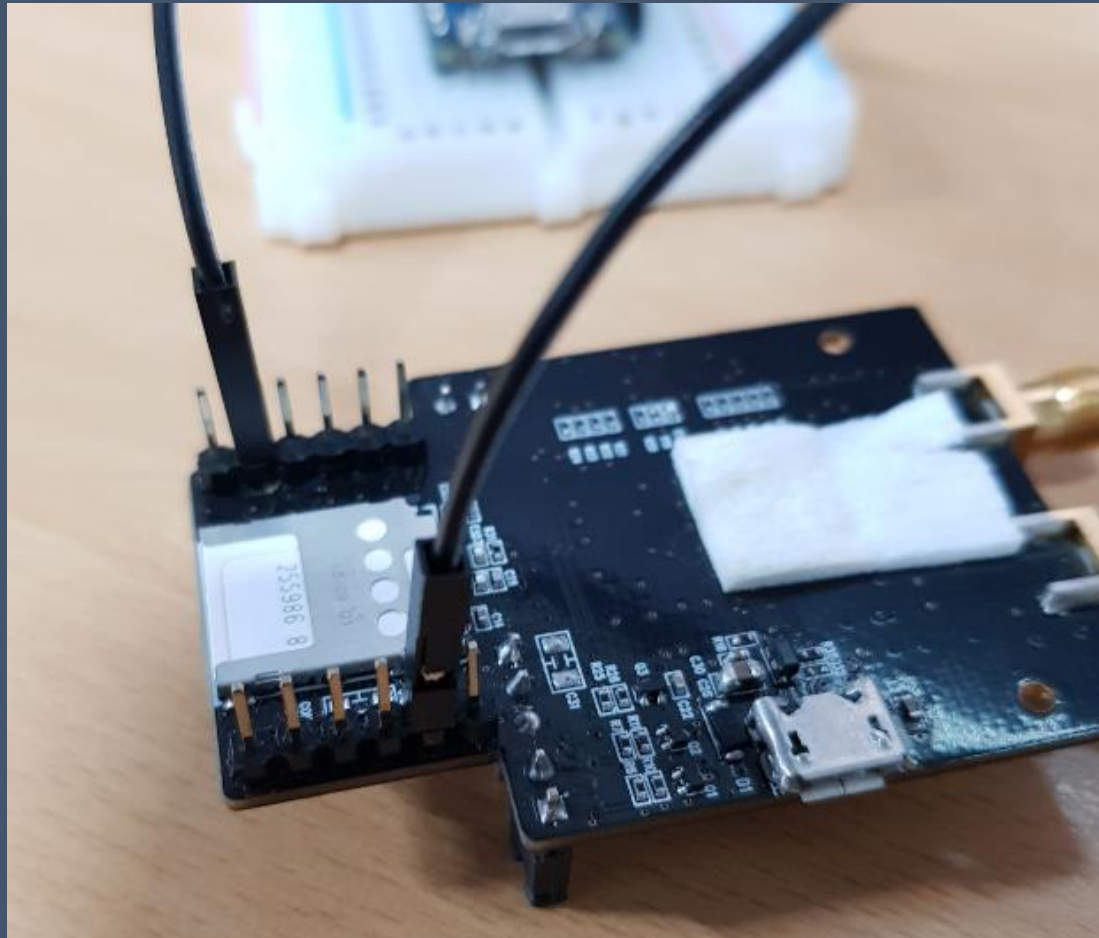
rooney.jang@codezoo.co.kr

www.CodeZoo.co.kr

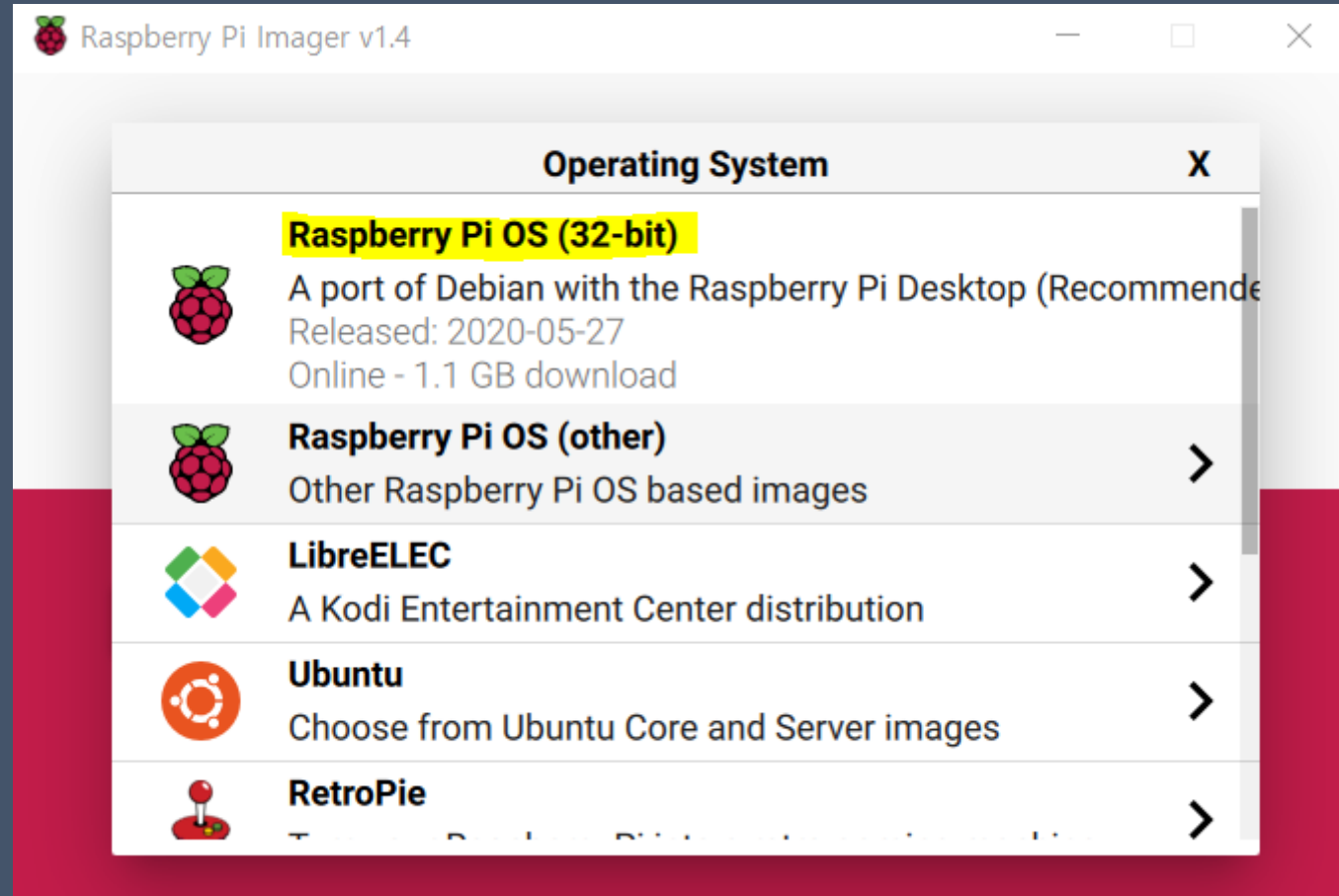
1. CAT.M1 Hardware 준비

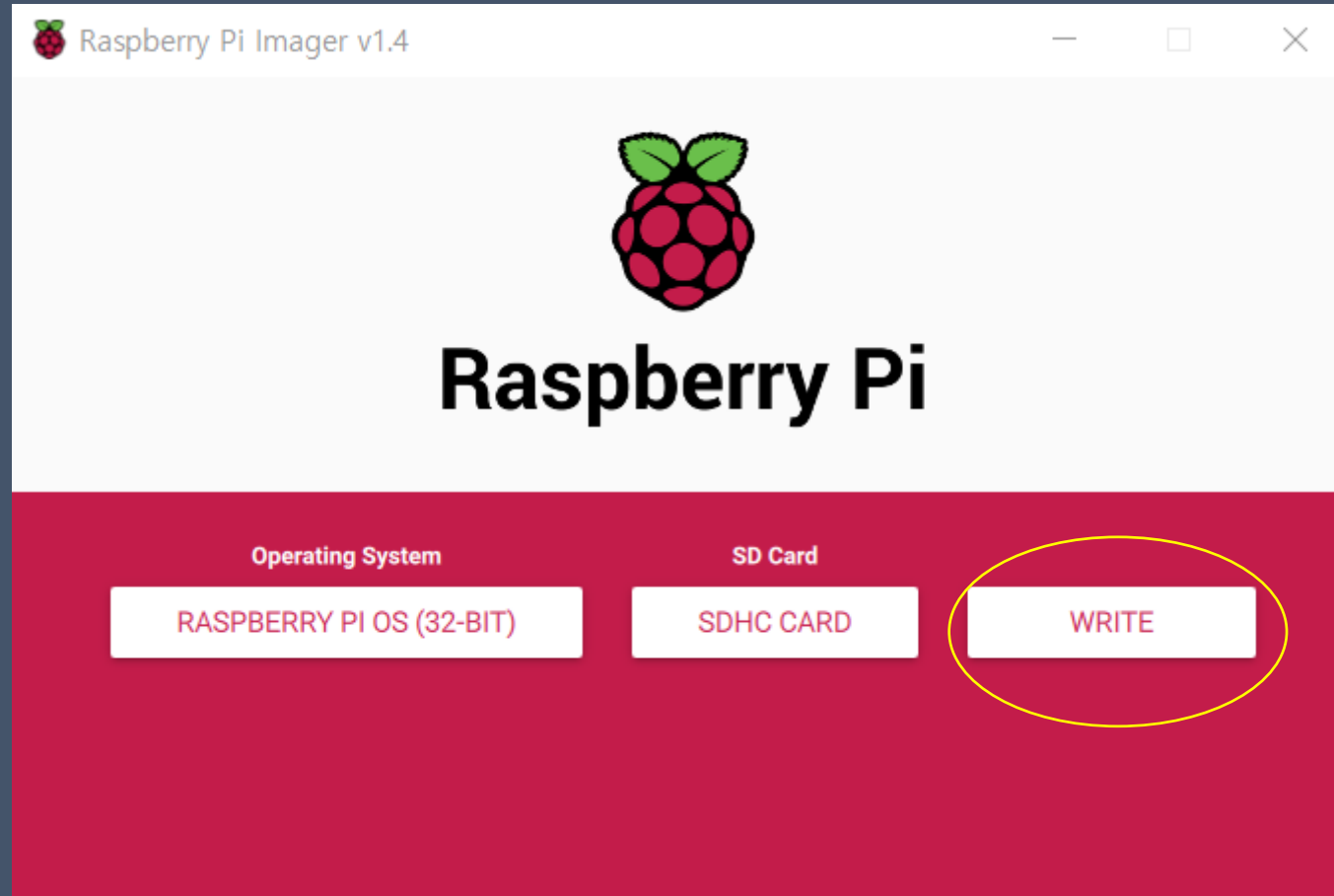
USB 연결 후 AT Command 테스트 절차

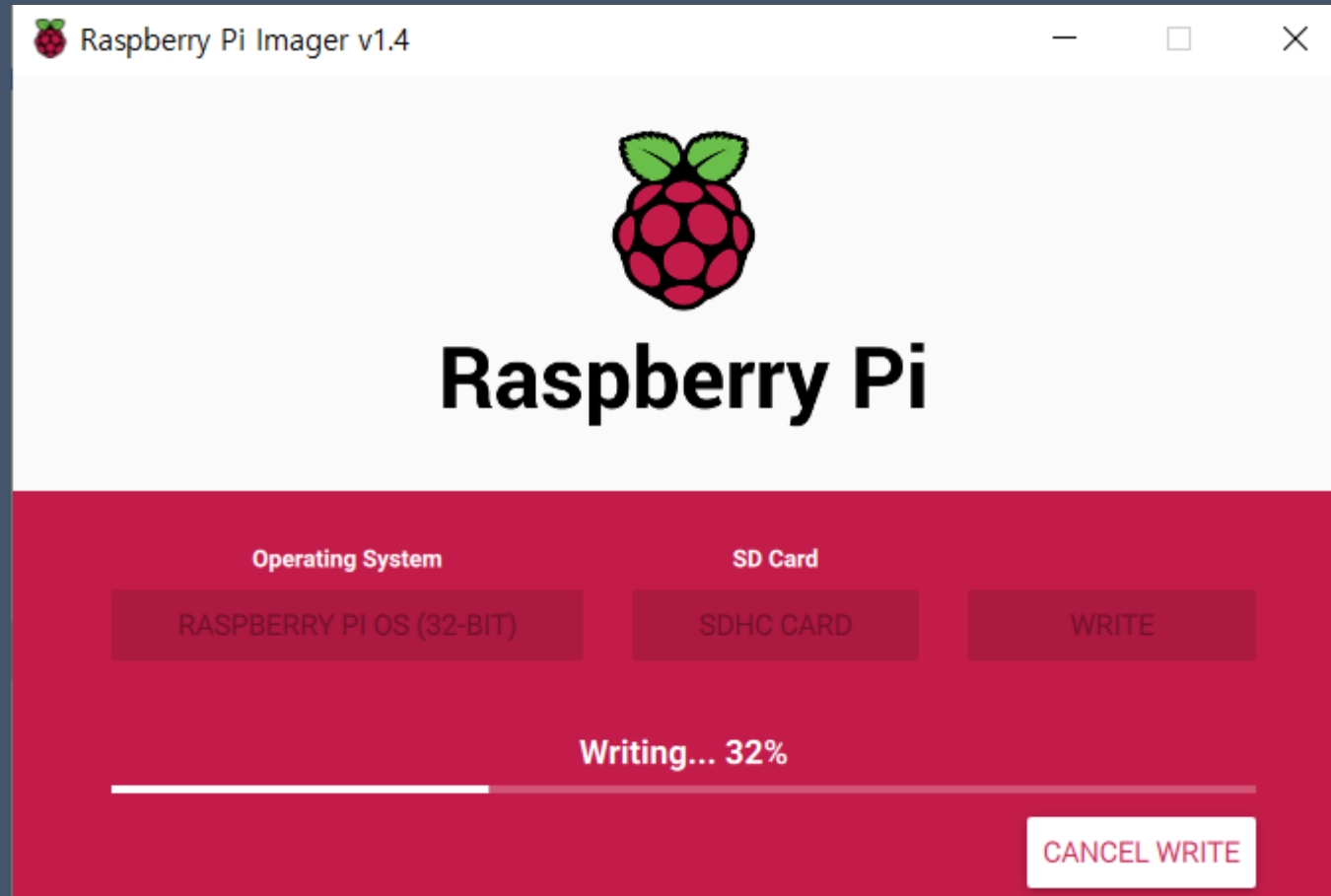
1. 후면 좌측 5번과 우측 2번을 점퍼선으로 연결
2. Micro USB 케이블을 라즈베리파이 USB 포트에 연결



2. CAT.M1 실습 (Raspberry Pi OS install)

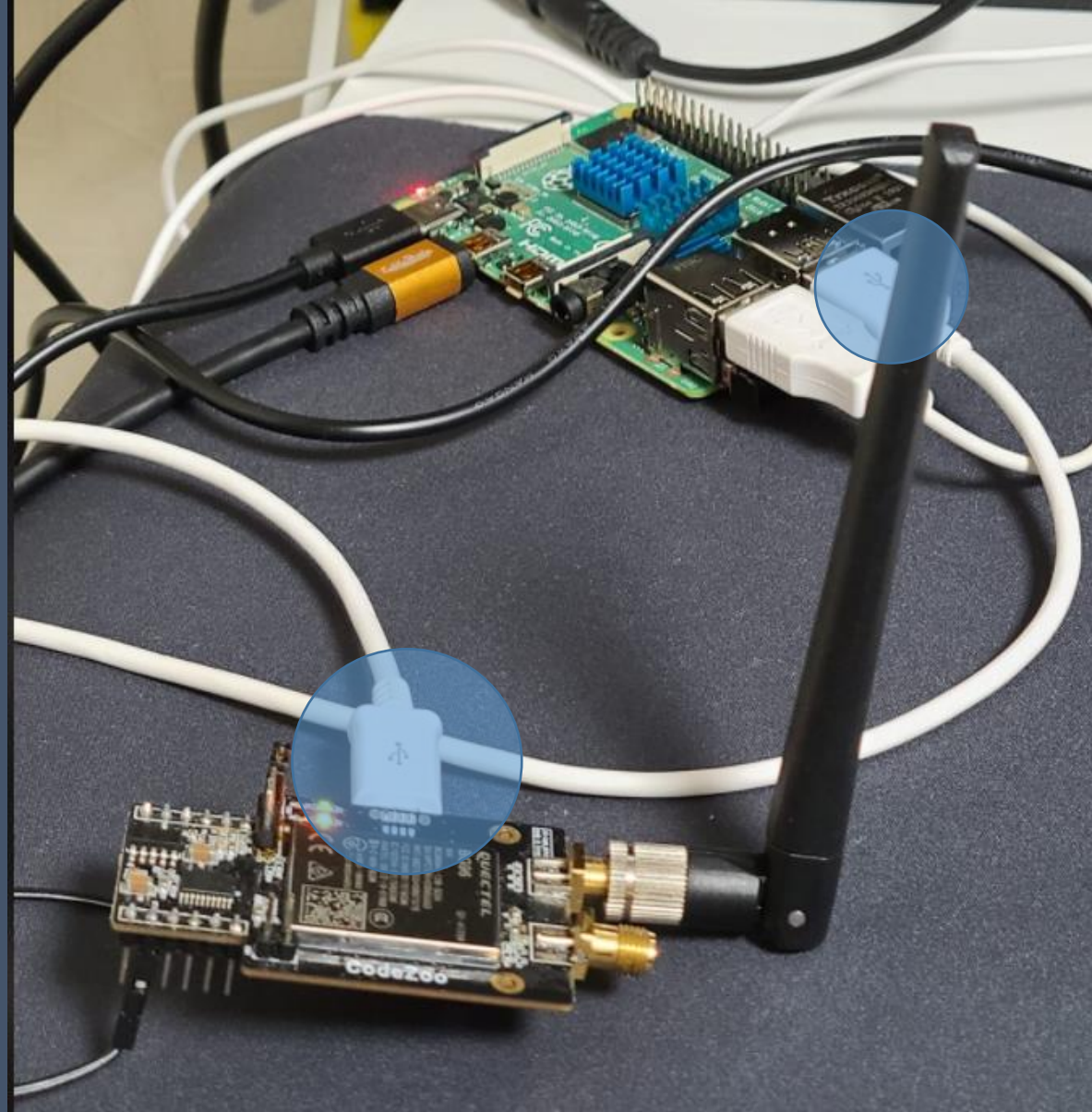






3. CAT.M1 연결

마이크로 USB 케이블로 CATM1과
라즈베리파이 연결



4. CAT.M1 PPP install

1. 라즈베리파이 터미널을 실행해서 installer file 을 다운로드 받습니다.

```
wget https://raw.githubusercontent.com/sixfab/Sixfab\_PPP\_Installer/master/ppp\_install\_standalone.sh
```

2. 파일의 퍼미션을 변경합니다.

```
sudo chmod +x ppp_install_standalone.sh
```

3. ppp_install_standalone.sh 을 실행 합니다.

```
sudo ./ppp_install_standalone.sh
```

4. CAT.M1 PPP install

4. 6: 3G/4G Base HAT 을 선택 합니다.
6 (Enter)

```
Please choose your Sixfab Shield/HAT:  
1: GSM/GPRS Shield  
2: 3G, 4G/LTE Base Shield  
3: Cellular IoT App Shield  
4: Cellular IoT HAT  
5: Tracker HAT  
6: 3G/4G Base HAT  
6 
```


4. CAT.M1 PPP install

5. APN 을 입력 합니다.

유플러스 유심 사용시 : m2m-catm1.default.lguplus.co.kr

텔레노어 유심 사용시 : internet.lte.cxn

```
pi@raspberrypi: ~  
File Edit Tabs Help  
Need to get 436 kB of archives.  
After this operation, 1,107 kB of additional disk space will be used.  
Get:1 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf libpcap0.8 a  
rmhf 1.8.1-6 [124 kB]  
Get:2 http://ftp.harukasan.org/raspbian/raspbian buster/main armhf ppp armhf 2.  
4.7-2+4.1+deb10u1 [312 kB]  
Fetched 436 kB in 7s (60.7 kB/s)  
Selecting previously unselected package libpcap0.8:armhf.  
(Reading database ... 95606 files and directories currently installed.)  
Preparing to unpack .../libpcap0.8_1.8.1-6_armhf.deb ...  
Unpacking libpcap0.8:armhf (1.8.1-6) ...  
Selecting previously unselected package ppp.  
Preparing to unpack .../ppp_2.4.7-2+4.1+deb10u1_armhf.deb ...  
Unpacking ppp (2.4.7-2+4.1+deb10u1) ...  
Setting up libpcap0.8:armhf (1.8.1-6) ...  
Setting up ppp (2.4.7-2+4.1+deb10u1) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/pppd-dns.service →  
/lib/systemd/system/pppd-dns.service.  
Processing triggers for systemd (241-7~deb10u4+rpi1) ...  
Processing triggers for man-db (2.8.5-2) ...  
Processing triggers for libc-bin (2.28-10+rpi1) ...  
What is your carrier APN?  
[ ]
```

What is your carrier APN?
m2m-catm1.default.lguplus.co.kr []

What is your carrier APN?
internet.lte.cxn []

4. CAT.M1 PPP install

6. 기타 설정을 입력 합니다.

1) username and password : n

2) device communication port : ttyUSB3

3) activate auto connect/reconnect service at R.Pi boot up : y

```
Does your carrier need username and password? [Y/n]
```

```
n
```

```
What is your device communication PORT? (ttyS0/ttyUSB3/etc.)
```

```
ttyUSB3
```

```
Do you want to activate auto connect/reconnect service at R.Pi boot up? [Y/n]
```

```
y
```

4. CAT.M1 PPP install

7. ENTER key 를 누르고 라즈베리파이를 재실행 시키면 PPP 서비스가 적용됩니다.

```
--2020-09-01 05:37:39-- https://raw.githubusercontent.com/sixfab/Sixfab_PPP_Installer/master/ppp_installer/reconnect_basehat
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.76.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.76.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 314 [text/plain]
Saving to: 'reconnect.sh'

reconnect.sh      100%[=====>]          314  --.-KB/s    in 0s

2020-09-01 05:37:39 (4.50 MB/s) - 'reconnect.sh' saved [314/314]

Created symlink /etc/systemd/system/multi-user.target.wants/reconnect.service → /etc/systemd/system/reconnect.service.
Press ENTER key to reboot
```

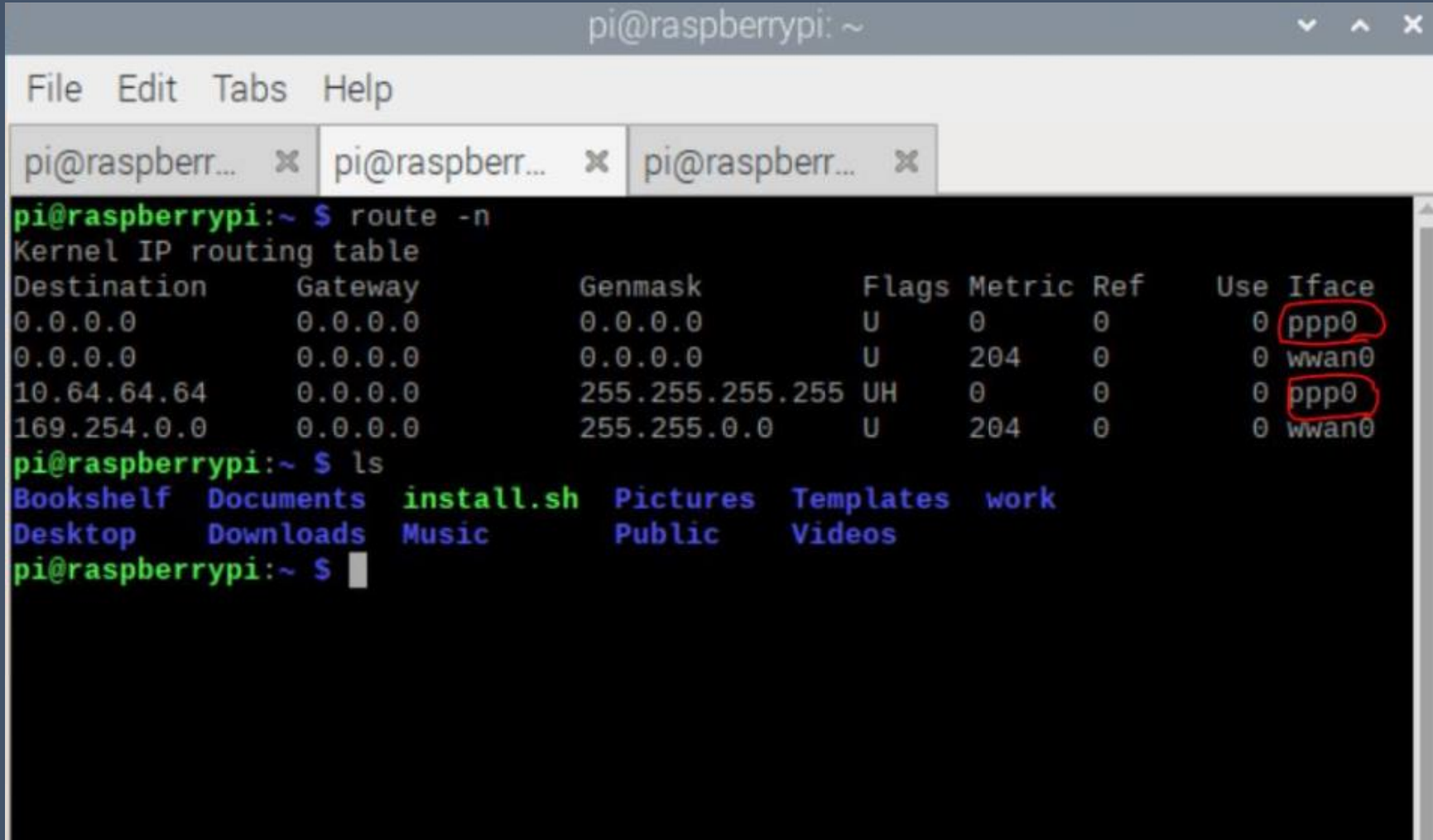
5. CAT.M1 PPP 동작 확인

1. 라즈베리파이 터미널에서 ifconfig 실행, PPP0 디바이스 확인

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberr... x pi@raspberr... x pi@raspberr... x  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500  
inet 10.126.86.39 netmask 255.255.255.255 destination 10.64.64.64  
ppp txqueuelen 3 (Point-to-Point Protocol)  
RX packets 3127 bytes 2947765 (2.8 MiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 3199 bytes 288567 (281.8 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wwan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 169.254.84.213 netmask 255.255.0.0 broadcast 169.254.255.255  
inet6 fe80::ba0a:2726:3666:a87 prefixlen 64 scopeid 0x20<link>  
ether 9e:19:92:00:69:93 txqueuelen 1000 (Ethernet)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 60 dropped 0 overruns 0 carrier 0 collisions 0  
  
pi@raspberrypi:~ $
```

5. CAT.M1 PPP 동작 확인

2. 라즈베리파이 터미널에서 route -n 실행, PPP0 디바이스 확인



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberr... x pi@raspberr... x pi@raspberr... x  
pi@raspberrypi:~ $ route -n  
Kernel IP routing table  
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface  
0.0.0.0          0.0.0.0          0.0.0.0         U        0      0      0 ppp0  
0.0.0.0          0.0.0.0          0.0.0.0         U        204    0      0 wwan0  
10.64.64.64      0.0.0.0          255.255.255.255 UH        0      0      0 ppp0  
169.254.0.0      0.0.0.0          255.255.0.0     U        204    0      0 wwan0  
pi@raspberrypi:~ $ ls  
Bookshelf Documents install.sh Pictures Templates work  
Desktop Downloads Music Public Videos  
pi@raspberrypi:~ $
```

5. CAT.M1 PPP 동작 확인

3. 파이썬 테스트 코드로 통신 테스트

python3 python_echo_client.py

```
pi@raspberrypi: ~/work/python_echo_client
File Edit Tabs Help
pi@raspberr... x pi@raspberr... x pi@raspberr... x

import socket
import sys

#Create a TCP/IP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

#Connect the socket to the port where the server is listening
server_address = ('echo.mbedcloudtesting.com', 7)
print('connecting to {}port {}'.format(*server_address))
sock.connect(server_address)

try:

    #Send data
    message = b'This is the message. It will be repeated.'
    print('sending {!r}'.format(message))
    sock.sendall(message)

    #Look for the response
    amount_received = 0
    amount_expected = len(message)

    while amount_received < amount_expected:
        data = sock.recv(16)
        amount_received += len(data)
        print('received {!r}'.format(data))

finally:
    print('closing socket')
    sock.close()

33,0-1 Bot
```

감사합니다.