Ⅱ. ARTIK 기술 교육

5. Wi-Fi

Wi-Fi

Wi-Fi(Wireless Fidelity)

- Where a wireless access point is installed , A wireless local area network (WLAN) capable of wireless Internet access within a certain distance using radio waves or infrared transmission
- In 1997, the Institute of Electrical and Electronics Engineers (IEEE) standardized Wi-Fi as IEEE 802.11

The advantages and disadvantages of Wi-Fi

advantages

- Compatibility is high due to the generalization of Wi-Fi
- High Data transmission speed
- AP installation is easy and installation cost is low
- Various versions such as a, b, g, n (IEEE 802.11)

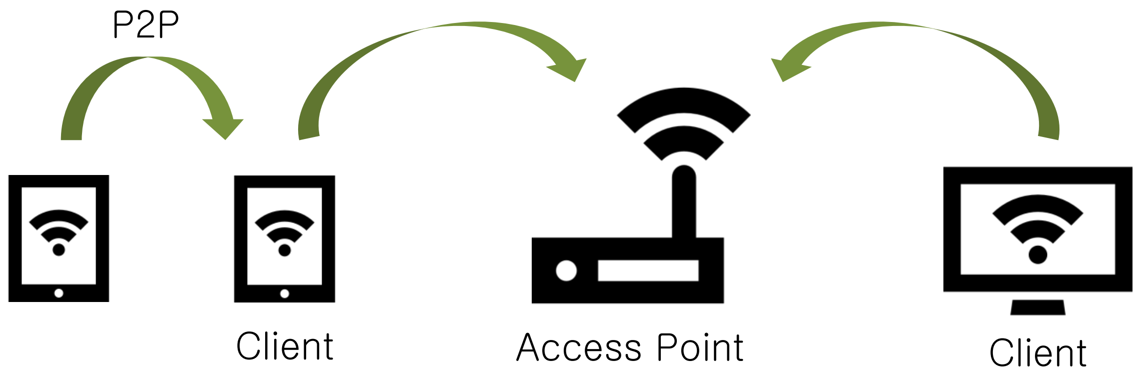
disadvantages

- Security risk due to simultaneous access of multiple devices to one AP
- Wireless Internet is available only near AP
- Relatively low communication range
- Interference between AP radio waves

Wi-Fi in ARTIK

■ Wi-Fi

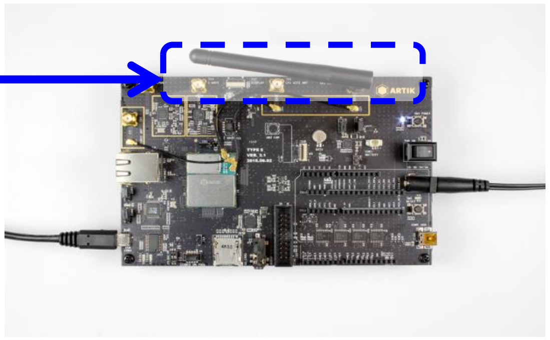
- Client : Mode to connect to an Internet-connected AP
  Required for most development work.
- Access Point : Mode to connect to Internet through Ethernet LAN in ARTIK.
  Enables wireless connection of other devices
- Direct or Wi-Fi Point-to-Point (P2P) : Mode to connect to other Wi-Fi client devices



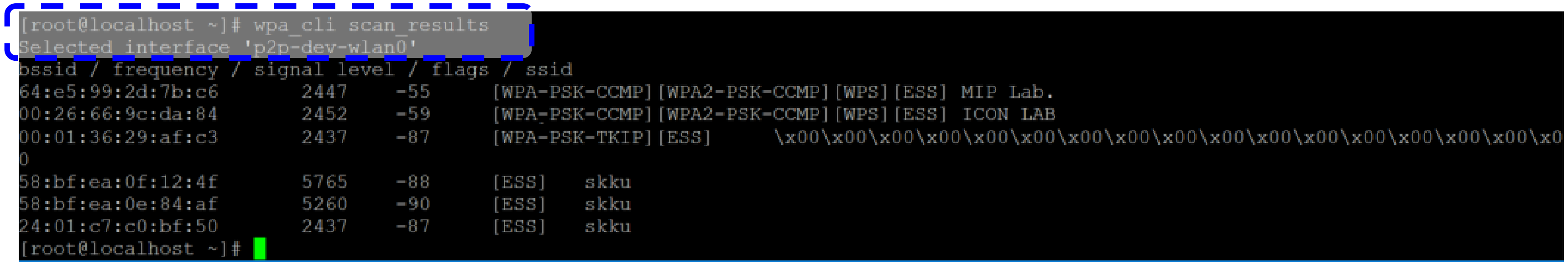
Using Wi-Fi on ARTIK 5

How to connect to WiFi

- Step 1. Attach a antenna

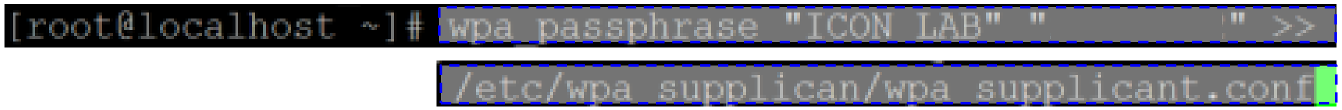


- Step 2. Scan for wireless access points
  - *# wpa_cli scan_results*



```
[root@localhost ~]# wpa_cli scan_results
Selected interface 'p2p-dev-wlan0'
bssid / frequency / signal level / flags / ssid
64:e5:99:2d:7b:c6       2447    -55     [WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS] MIP Lab.
00:26:66:9c:da:84       2452    -59     [WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS] ICON LAB
00:01:36:29:af:c3       2437    -87     [WPA-PSK-TKIP][ESS]     \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0
58:bf:ea:0f:12:4f       5765    -88     [ESS]   skku
58:bf:ea:0e:84:af       5260    -90     [ESS]   skku
24:01:c7:c0:bf:50       2437    -87     [ESS]   skku
[root@localhost ~]#
```

- Step 3. Configure wpa_supplicant.conf
  - *# wpa_passphrase "SSID" "PASSWORD" >> /etc/wpa_supplicant/wpa_supplicant.conf*



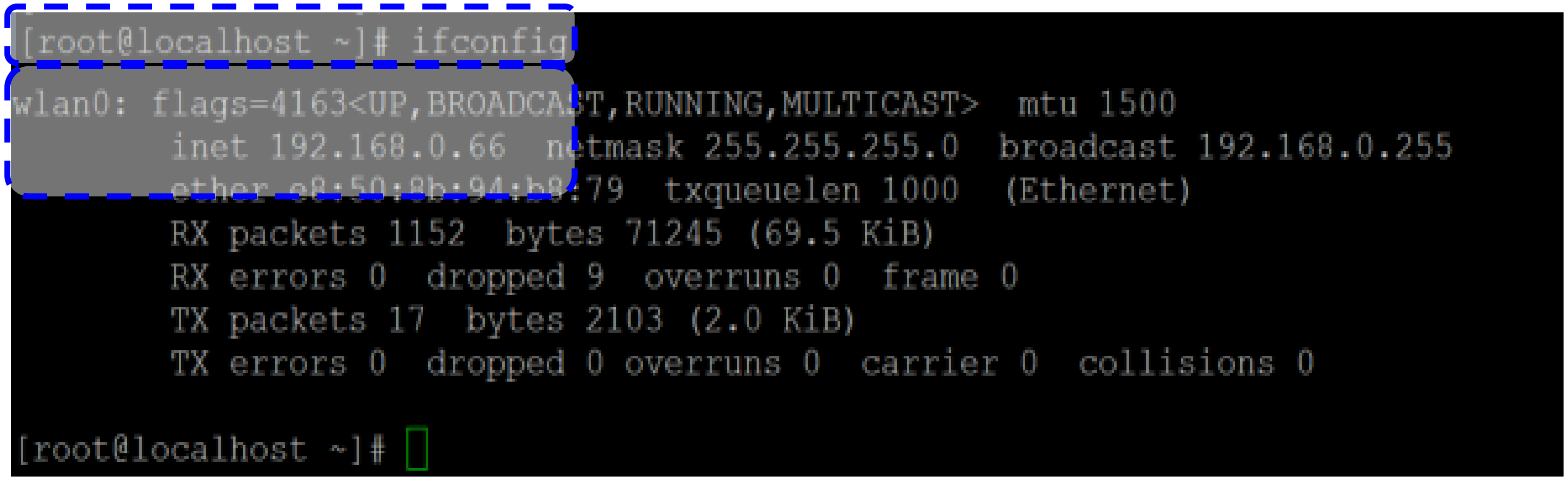
```
[root@localhost ~]# wpa_passphrase "ICON LAB" "          " >>
                    /etc/wpa_supplican/wpa_supplicant.conf
```

Using Wi-Fi on ARTIK 5

How to connect to WiFi

- Step 4. Restart wpa_supplicant
  - *# systemctl restart wpa_supplicant*

- Step 5. Get an IP address
  - *# dhclient wlan0*
  - *# ifconfig*

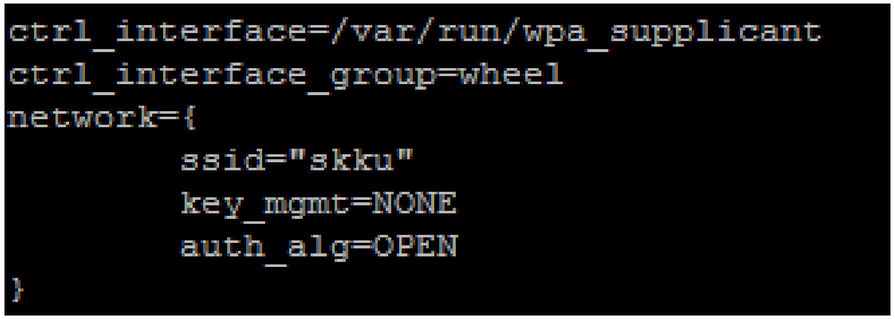


```
[root@localhost ~]# ifconfig
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.66  netmask 255.255.255.0  broadcast 192.168.0.255
        ether e8:50:8b:94:b8:79  txqueuelen 1000  (Ethernet)
        RX packets 1152  bytes 71245 (69.5 KiB)
        RX errors 0  dropped 9  overruns 0  frame 0
        TX packets 17  bytes 2103 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]#
```

Using Wi-Fi on ARTIK 5

If you are connecting to unsecured Wi-Fi

- Step 1. Configure wpa_supplicant.conf
  - *# wpa_passphrase "SSID" "Any 8 digits" >> /etc/wpa_supplicant/wpa_supplicant.conf*
- Step 2. Reconfigure wpa_supplicant.conf
  - *# vi /etc/wpa_supplicant/wpa_supplicant.conf*
- Step 3. Delete both psk lines and use the following command



```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=wheel
network={
        ssid="skku"
        key_mgmt=NONE
        auth_alg=OPEN
}
```

- Step 4. The following procedure is the same as before

Ⅱ. ARTIK 기술 교육

6. Socket Communication

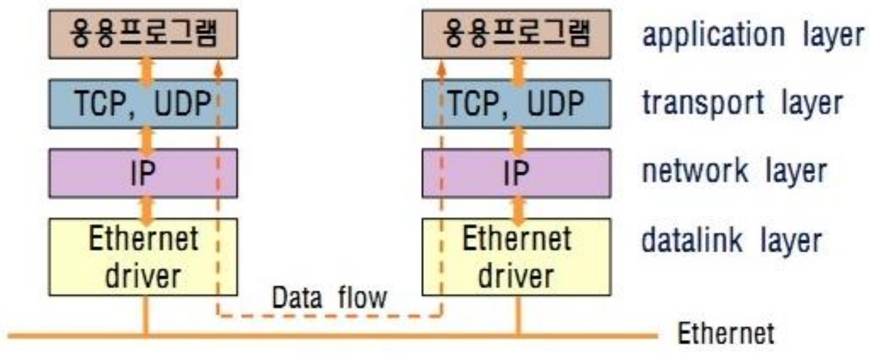
TCP/IP 통신

■ Protocol

- 종단 시스템 간 데이터 교환 통신 규약

■ TCP/IP (Transmission Control Protocol/Internet Protocol)

- 호스트들의 상호 통신을 위한 표준화된 프로토콜
- IP
  - 규약/규칙에 따라 부여된 주소 : IP Address (ex. 256.152.10.100)
- Port
  - IP로 접근 후 내부의 어떤 프로그램과 통신을 할지에 대해 구분할때 사용
  - 통신을 하기 위한 출입구
  - 0~65535까지 사용가능(0~1023까지는 시스템에서 사용, 1024~65536사이 사용 가능)



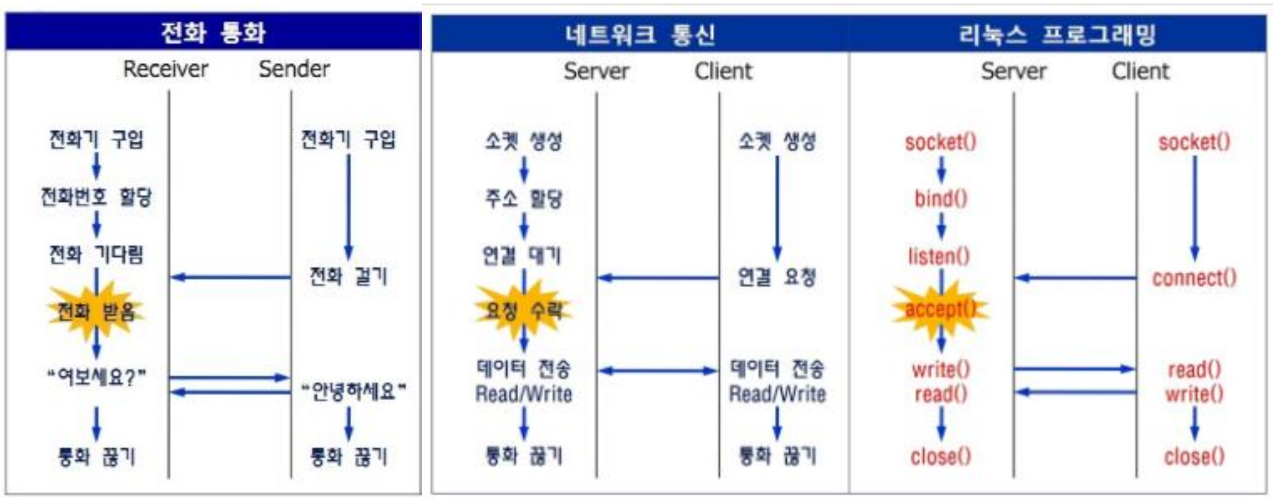
소켓(socket) 통신

■ 소켓 통신

- TCP/IP통신을 하기 위해 사용되는 소프트웨어적인 장치
- 소켓은 각 포트를 사용하여 통신을 수행하는 도구
- 포트는 출입구, 소켓은 출입구를 통하여 데이터를 직접 송수신하는 매개체

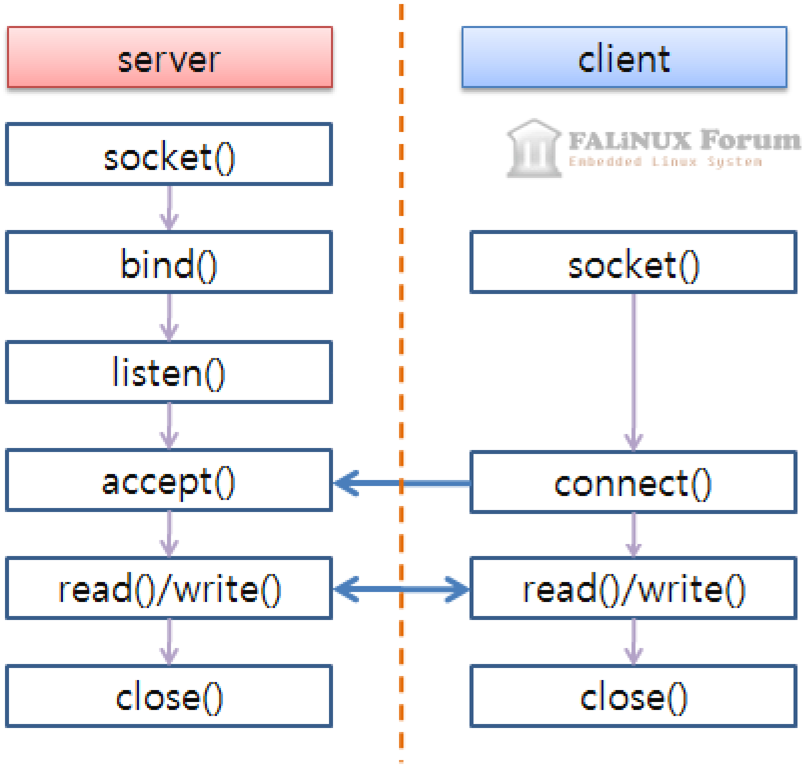
■ 소켓 동작 모양

- 서버 소켓과 클라이언트 소켓으로 나뉨
- 서버 소켓 : 클라이언트 소켓의 연결 요청 대기, 연결 요청시 클라이언트 소켓 생성 후 통신 가능
- 클라이언트 소켓 : 대기 없이 바로 사용 가능, 실제 데이터 송수신이 일어나는 소켓



Source code of Wi-Fi Communication

■ Wi-Fi Socket Communication



Source code of Wi-Fi Communication

Source Code(Server-1)

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include <assert.h>

#define MYPORT 3490
#define BACKLOG 10
#define MAXDATASIZE 100

int main() {
    int sockfd, new_fd;
    struct sockaddr_in my_addr;
    struct sockaddr_in their_addr;
    int sin_size;
    int on = 1;
    char *recv_data;

    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socket");
        exit(1);
    }

    my_addr.sin_family = AF_INET;
    my_addr.sin_port = htons(MYPORT);
    my_addr.sin_addr.s_addr = INADDR_ANY;
    bzero(&(my_addr.sin_zero), 8);
```

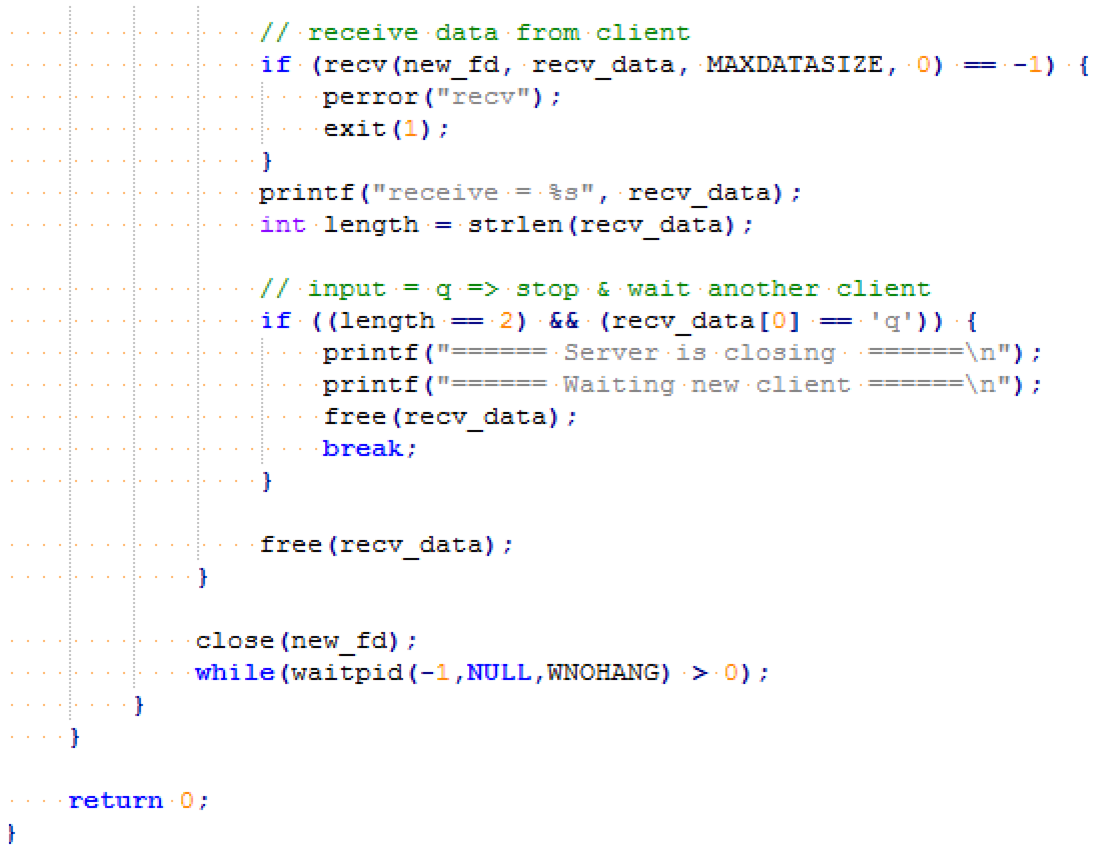
Source code of Wi-Fi Communication

Source Code(Server-2)

```
    setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR, &on, sizeof(on));
    if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == -1) {
        perror("bind");
        exit(1);
    }

    if (listen(sockfd, BACKLOG) == -1) {
        perror("listen");
        exit(1);
    }

    printf("========= [PORT] : %d =========\n", MYPORT);
    printf("Server : waiting client\n");

    while(1) {
        // wait until connecting with client
        sin_size = sizeof(struct sockaddr_in);
        if ((new_fd = accept(sockfd, (struct sockaddr *)&their_addr,&sin_size)) == -1) {
            perror("accept");
            continue;
        }

        if (!fork()) {
            while(1){
                // memory allocation for reading buffer
                recv_data = (char *)malloc(MAXDATASIZE*sizeof(char));
                assert(recv_data);
```

Source code of Wi-Fi Communication

Source Code(Server-3)



```
                // receive data from client
                if (recv(new_fd, recv_data, MAXDATASIZE, 0) == -1) {
                    perror("recv");
                    exit(1);
                }
                printf("receive = %s", recv_data);
                int length = strlen(recv_data);

                // input = q => stop & wait another client
                if ((length == 2) && (recv_data[0] == 'q')) {
                    printf("====== Server is closing  ======\n");
                    printf("====== Waiting new client ======\n");
                    free(recv_data);
                    break;
                }

                free(recv_data);
            }

            close(new_fd);
            while(waitpid(-1,NULL,WNOHANG) > 0);
        }
    }

    return 0;
}
```

Source code of Wi-Fi Communication

■ Source Code(Client-1)

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <assert.h>

#define PORT 3490
#define MAXDATASIZE 100

int main(int argc, char *argv[])
{
    int sockfd, numbytes;
    char *send_data;
    struct hostent *he;
    struct sockaddr_in their_addr;

    if (argc != 2) {
        fprintf(stderr, "usage: ./client <host-name>\n");
        exit(1);
    }

    if ((he=gethostbyname(argv[1])) == NULL) {
        herror("gethostbyname");
        exit(1);
    }
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socket");
        exit(1);
    }
```

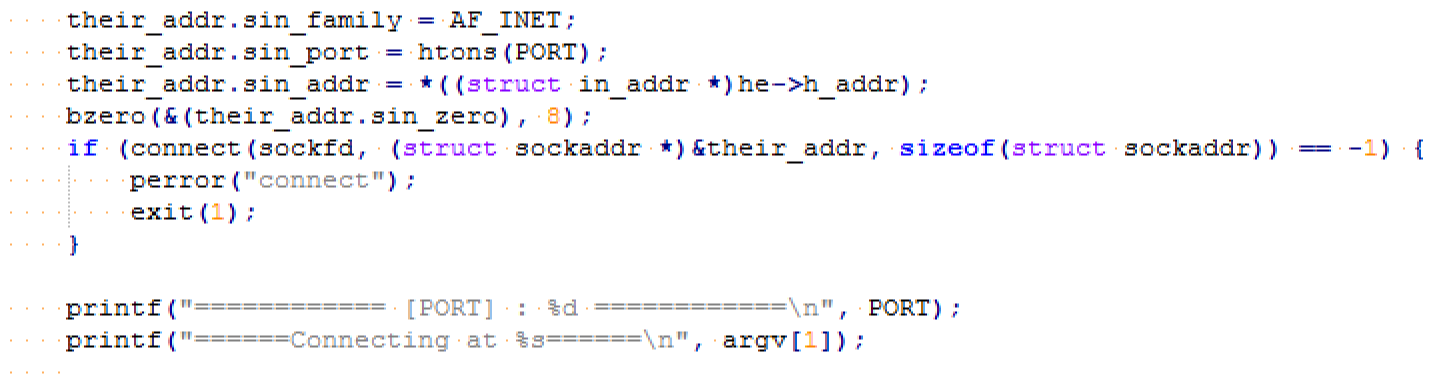
Source code of Wi-Fi Communication

Source Code(Client-1)

```c
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <assert.h>

#define PORT 3490
#define MAXDATASIZE 100

int main(int argc, char *argv[])
{
    int sockfd, numbytes;
    char *send_data;
    struct hostent *he;
    struct sockaddr_in their_addr;

    if (argc != 2) {
        fprintf(stderr, "usage: ./client <host-name>\n");
        exit(1);
    }

    if ((he=gethostbyname(argv[1])) == NULL) {
        herror("gethostbyname");
        exit(1);
    }
    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("socket");
        exit(1);
    }
```

Source code of Wi-Fi Communication

■ Source Code(Client-2)



```
    their_addr.sin_family = AF_INET;
    their_addr.sin_port = htons(PORT);
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    bzero(&(their_addr.sin_zero), 8);
    if (connect(sockfd, (struct sockaddr *)&their_addr, sizeof(struct sockaddr)) == -1) {
        perror("connect");
        exit(1);
    }

    printf("============ [PORT] : %d ============\n", PORT);
    printf("======Connecting at %s======\n", argv[1]);
    
```

Source code of Wi-Fi Communication

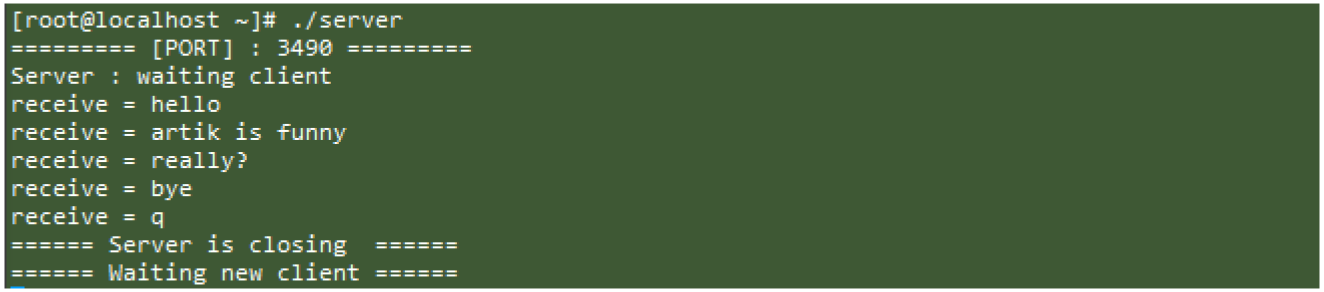
■ Source Code(Client-3)

```
    while(1){
        // memory allocation for reading buffer
        send_data = (char *)malloc(MAXDATASIZE*sizeof(char));
        assert(send_data);

        printf("send[q : exit] : ");
        fgets(send_data, MAXDATASIZE, stdin);
        int length = strlen(send_data);

        // send message => if find error, stop
        if (send(sockfd, send_data, length+1, 0) == -1) {
            perror("send");
            free(send_data);
            break;
        }
        // input = q => stop
        if ((length == 2) && (send_data[0] == 'q')) {
            printf("========= Exiting client !! =========\n");
            free(send_data);
            break;
        }
        free(send_data);
    }

    close(sockfd);

    return 0;
}
```

Wi-Fi ex. result

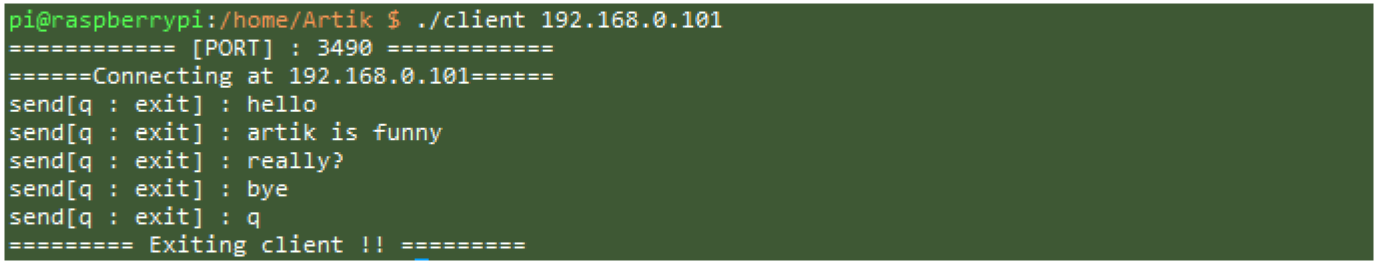
Execution result

- Server ARTIK



```
[root@localhost ~]# ./server
========= [PORT] : 3490 =========
Server : waiting client
receive = hello
receive = artik is funny
receive = really?
receive = bye
receive = q
====== Server is closing  ======
====== Waiting new client ======
```

- Client ARTIK

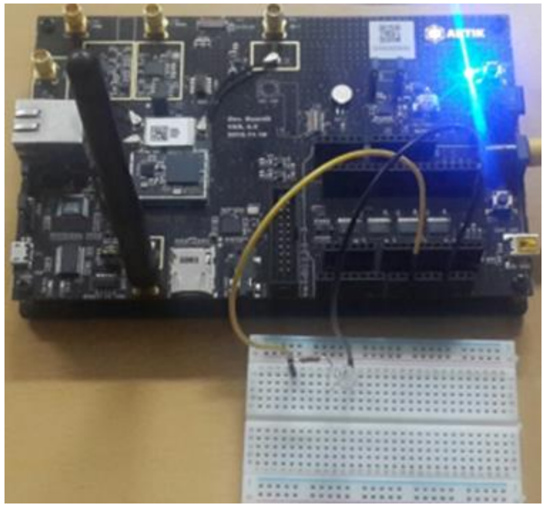


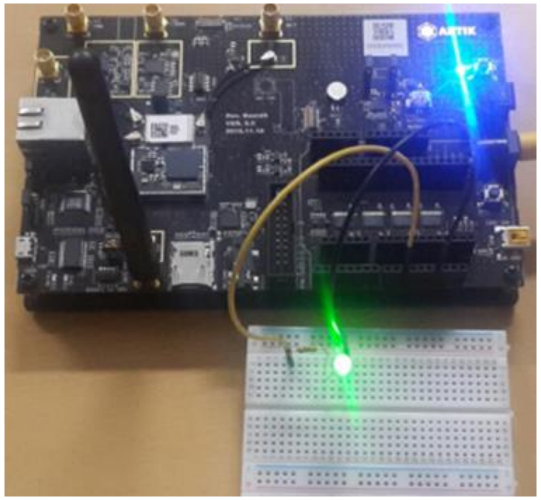
```
pi@raspberrypi:/home/Artik $ ./client 192.168.0.101
============ [PORT] : 3490 ============
======Connecting at 192.168.0.101======
send[q : exit] : hello
send[q : exit] : artik is funny
send[q : exit] : really?
send[q : exit] : bye
send[q : exit] : q
========= Exiting client !! =========
```

Wi-Fi Quiz

Execution result

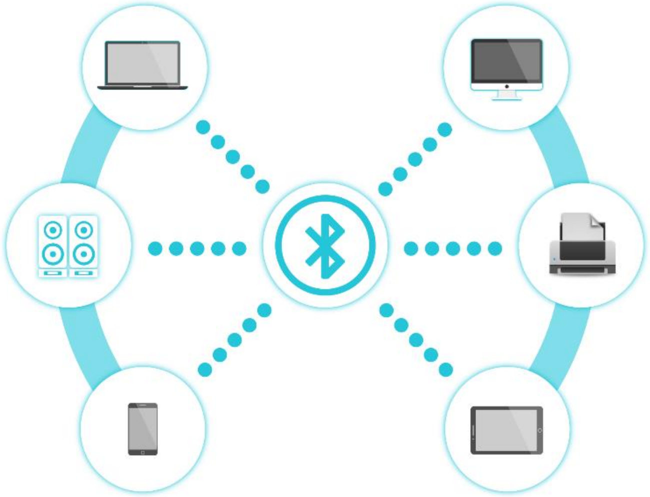
- Using Wi-Fi Socket Communication, control LED


Before


After

Ⅱ. ARTIK 기술 교육

7. Bluetooth



Bluetooth

Bluetooth

- Bluetooth is a small, low-cost, low-power
- Bluetooth is a standard for wireless connectivity in small areas (10 m to 100 m), including smartphones, laptops, and other peripheral device
- In recent years, the Bluetooth Research Group(SIG) released Bluetooth 4.1, which better supports Internet (IoT) and Wearable Device and can coexist with 4G

The advantages and disadvantages of Bluetooth

advantages

- Low cost and low power(100mW)
- High security due to separate transmission
- Communicate even if there is obstacle between devices
- No signal attenuation due to angle

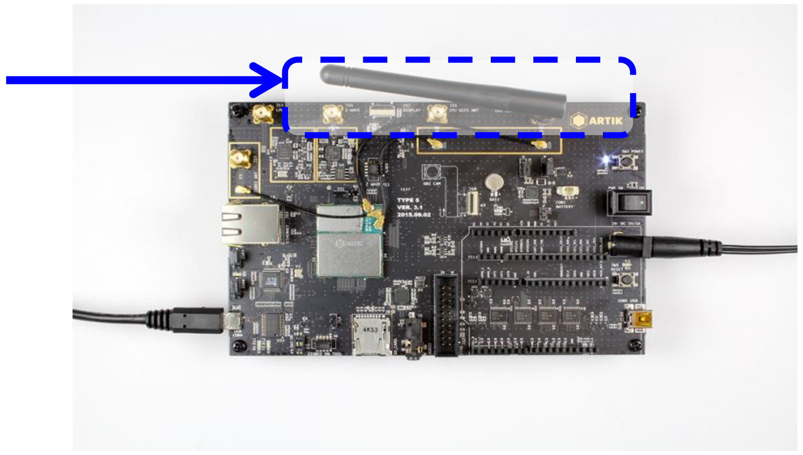
disadvantages

- Bluetooth module needs its own power supply
- Crosstalk during data transmission

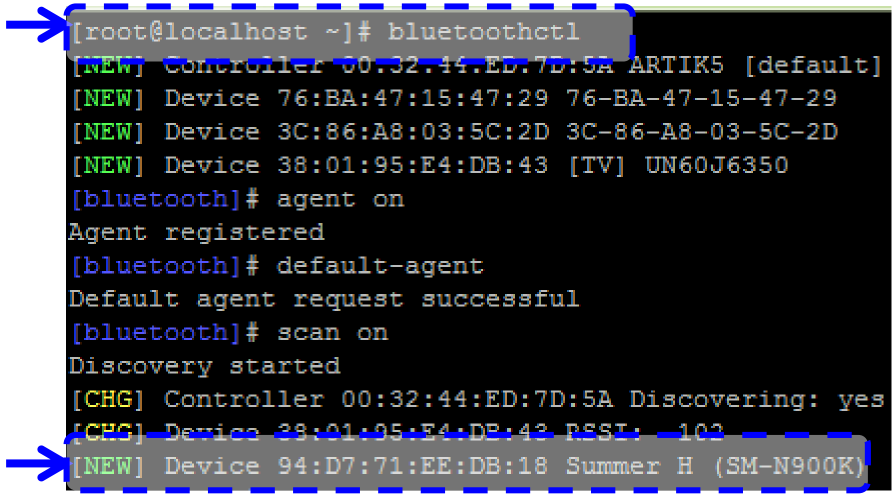
Using Bluetooth on ARTIK 5

How to pair between ARTIK 5 and other device

- Step 1. Attach a antenna



- Step 2. Input commands
  - *# bluetoothctl*
  - *# agent on*
  - *# default-agent*
  - *# scan on*
- Step 3. Check device

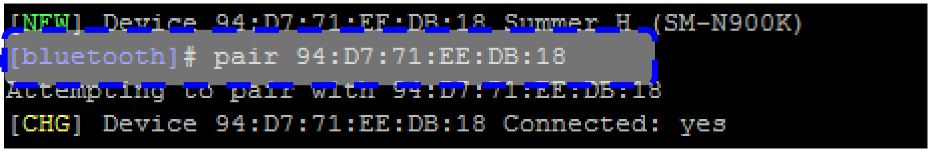


```
[root@localhost ~]# bluetoothctl
[NEW] Controller 00:32:44:ED:7D:5A ARTIK5 [default]
[NEW] Device 76:BA:47:15:47:29 76-BA-47-15-47-29
[NEW] Device 3C:86:A8:03:5C:2D 3C-86-A8-03-5C-2D
[NEW] Device 38:01:95:E4:DB:43 [TV] UN60J6350
[bluetooth]# agent on
Agent registered
[bluetooth]# default-agent
Default agent request successful
[bluetooth]# scan on
Discovery started
[CHG] Controller 00:32:44:ED:7D:5A Discovering: yes
[CHG] Device 38:01:95:E4:DB:43 RSSI: -102
[NEW] Device 94:D7:71:EE:DB:18 Summer H (SM-N900K)
```

Using Bluetooth on ARTIK 5

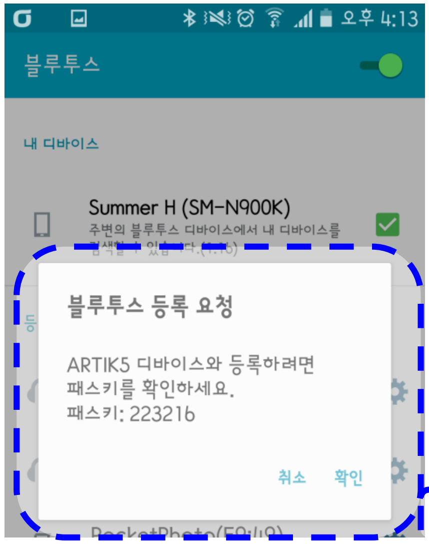
How to pair between ARTIK 5 and other device

- Step 4. Input command
  - *# pair xx : xx : xx : xx : xx : xx*



```
[NEW] Device 94:D7:71:EE:DB:18 Summer H (SM-N900K)
[bluetooth]# pair 94:D7:71:EE:DB:18
Attempting to pair with 94:D7:71:EE:DB:18
[CHG] Device 94:D7:71:EE:DB:18 Connected: yes
```

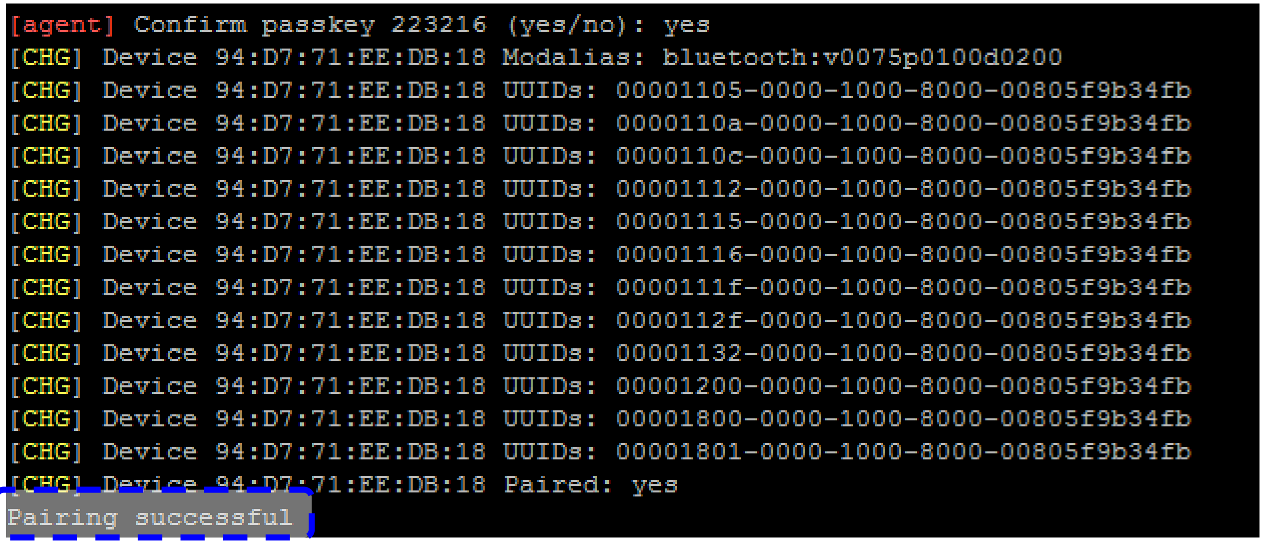
- Step 5. Pairing check
  - check ARTIK 5 and device



오후 4:13
블루투스
내 디바이스
Summer H (SM-N900K)
주변의 블루투스 디바이스에서 내 디바이스를 검색할 수 있습니다. (1:16)

블루투스 등록 요청

ARTIK5 디바이스와 등록하려면
패스키를 확인하세요.
패스키: 223216



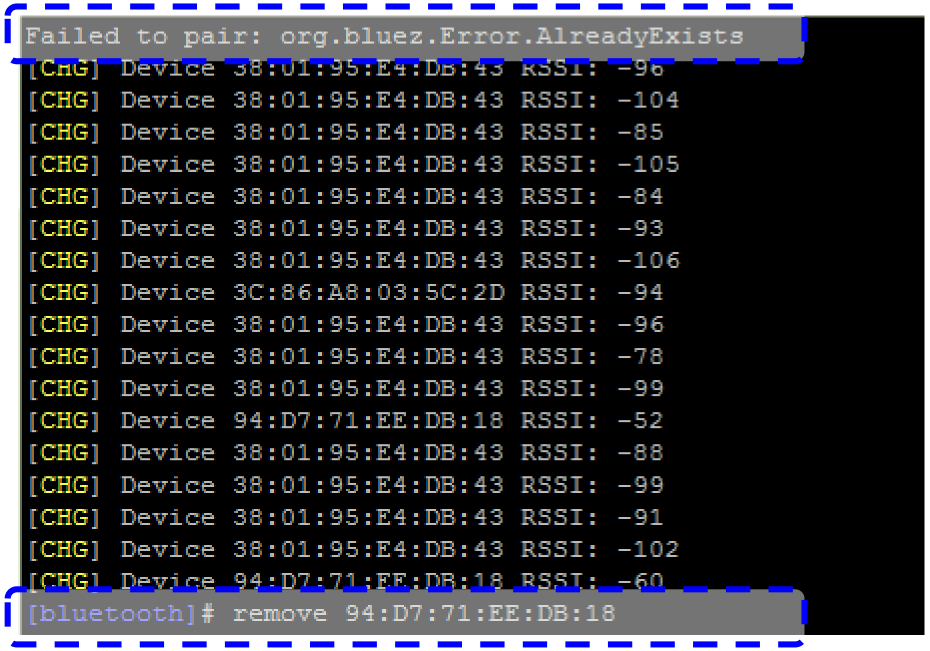
```
[agent] Confirm passkey 223216 (yes/no): yes
[CHG] Device 94:D7:71:EE:DB:18 Modalias: bluetooth:v0075p0100d0200
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 Paired: yes
Pairing successful
```

Using Bluetooth on ARTIK 5

Note

- Case of the pairing fail
  - Once you reconnect after pairing Bluetooth
  - ARTIK is paired to another device

- Solution
  - *# remove xx : xx : xx : xx : xx : xx*
  - Return to Step 2



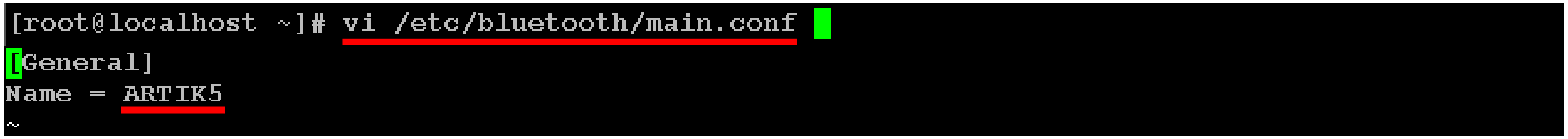
```
Failed to pair: org.bluez.Error.AlreadyExists
[CHG] Device 38:01:95:E4:DB:43 RSSI: -96
[CHG] Device 38:01:95:E4:DB:43 RSSI: -104
[CHG] Device 38:01:95:E4:DB:43 RSSI: -85
[CHG] Device 38:01:95:E4:DB:43 RSSI: -105
[CHG] Device 38:01:95:E4:DB:43 RSSI: -84
[CHG] Device 38:01:95:E4:DB:43 RSSI: -93
[CHG] Device 38:01:95:E4:DB:43 RSSI: -106
[CHG] Device 3C:86:A8:03:5C:2D RSSI: -94
[CHG] Device 38:01:95:E4:DB:43 RSSI: -96
[CHG] Device 38:01:95:E4:DB:43 RSSI: -78
[CHG] Device 38:01:95:E4:DB:43 RSSI: -99
[CHG] Device 94:D7:71:EE:DB:18 RSSI: -52
[CHG] Device 38:01:95:E4:DB:43 RSSI: -88
[CHG] Device 38:01:95:E4:DB:43 RSSI: -99
[CHG] Device 38:01:95:E4:DB:43 RSSI: -91
[CHG] Device 38:01:95:E4:DB:43 RSSI: -102
[CHG] Device 94:D7:71:EE:DB:18 RSSI: -60
[bluetooth]# remove 94:D7:71:EE:DB:18
```

Using Bluetooth on ARTIK 5

Note

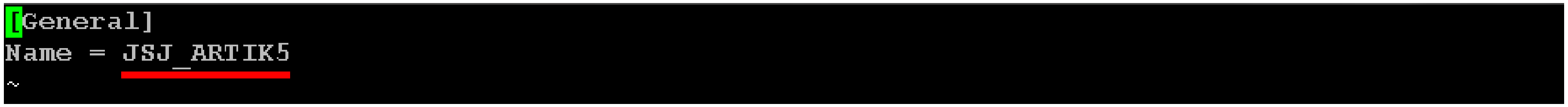
- Rename Bluetooth

  # vi /etc/bluetooth/main.conf



```
[root@localhost ~]# vi /etc/bluetooth/main.conf
[General]
Name = ARTIK5
~
```

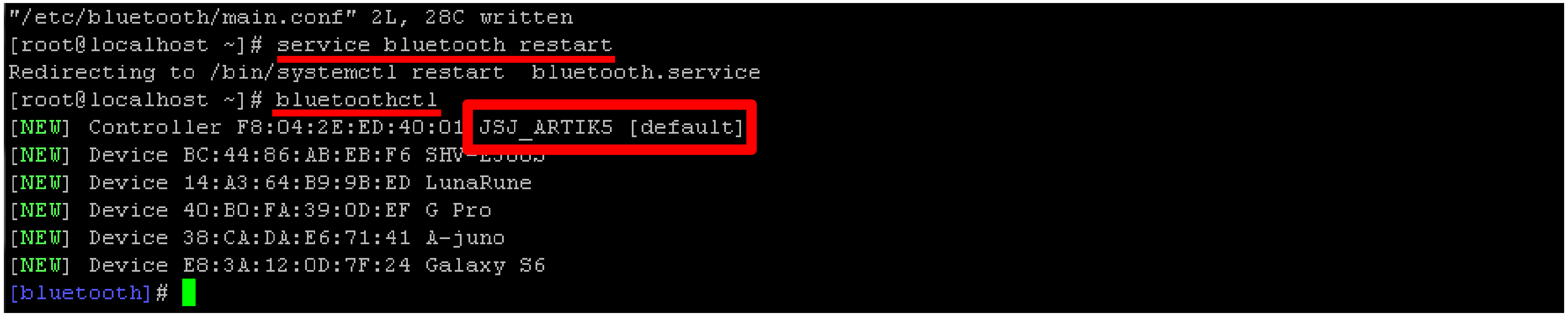
  - *Rename : ARTIK5 ⇒ OOO_ARTIK5 (ex. JSJ_ARTIK5)*



```
[General]
Name = JSJ_ARTIK5
~
```

  # service bluetooth restart

  # bluetoothctl

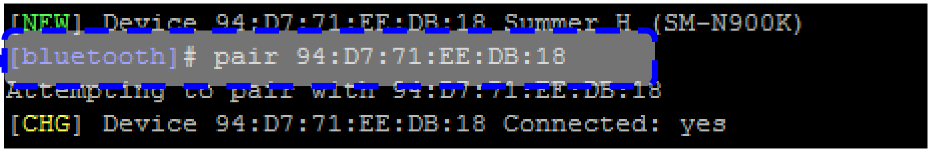


```
"/etc/bluetooth/main.conf" 2L, 28C written
[root@localhost ~]# service bluetooth restart
Redirecting to /bin/systemctl restart  bluetooth.service
[root@localhost ~]# bluetoothctl
[NEW] Controller F8:04:2E:ED:40:01 JSJ_ARTIK5 [default]
[NEW] Device BC:44:86:AB:EB:F6 SHV-E300S
[NEW] Device 14:A3:64:B9:9B:ED LunaRune
[NEW] Device 40:B0:FA:39:0D:EF G Pro
[NEW] Device 38:CA:DA:E6:71:41 A-juno
[NEW] Device E8:3A:12:0D:7F:24 Galaxy S6
[bluetooth]#
```

Using Bluetooth on ARTIK 5

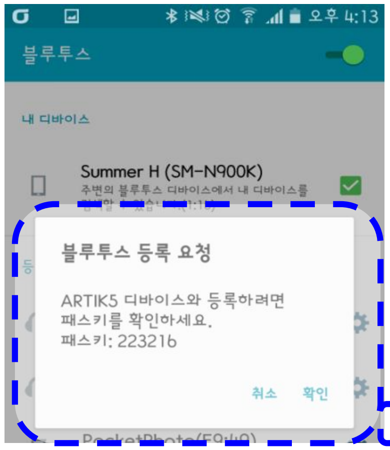
How to pair between ARTIK 5 and other device

- Step 4. Input command
  - *# pair xx : xx : xx : xx : xx : xx*

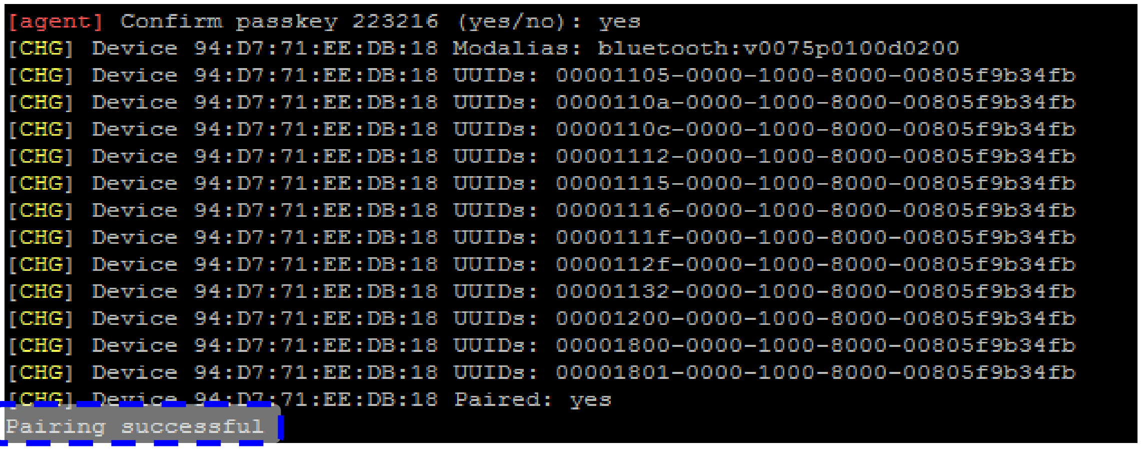


```
[NEW] Device 94:D7:71:EE:DB:18 Summer H (SM-N900K)
[bluetooth]# pair 94:D7:71:EE:DB:18
Attempting to pair with 94:D7:71:EE:DB:18
[CHG] Device 94:D7:71:EE:DB:18 Connected: yes
```

- Step 5. Pairing check and connection
  - check ARTIK 5 and device

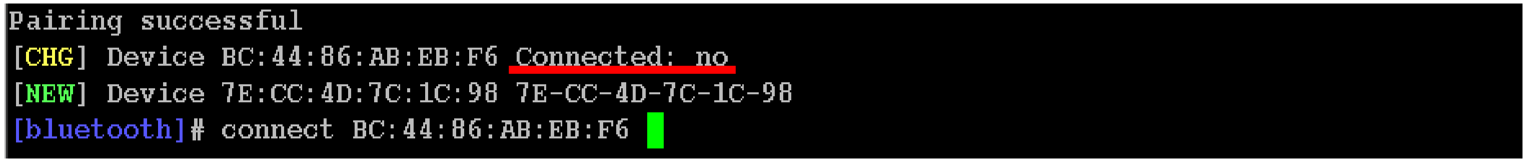


오후 4:13
블루투스
내 디바이스
Summer H (SM-N900K)
주변의 블루투스 디바이스에서 내 디바이스를 검색할 수 있습니다 (1:18)
블루투스 등록 요청
ARTIK5 디바이스와 등록하려면 패스키를 확인하세요.
패스키: 223216
취소
확인
PocketPhoto(F9:49)



```
[agent] Confirm passkey 223216 (yes/no): yes
[CHG] Device 94:D7:71:EE:DB:18 Modalias: bluetooth:v0075p0100d0200
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 94:D7:71:EE:DB:18 Paired: yes
Pairing successful
```

  - *# connect xx : xx : xx : xx : xx : xx*



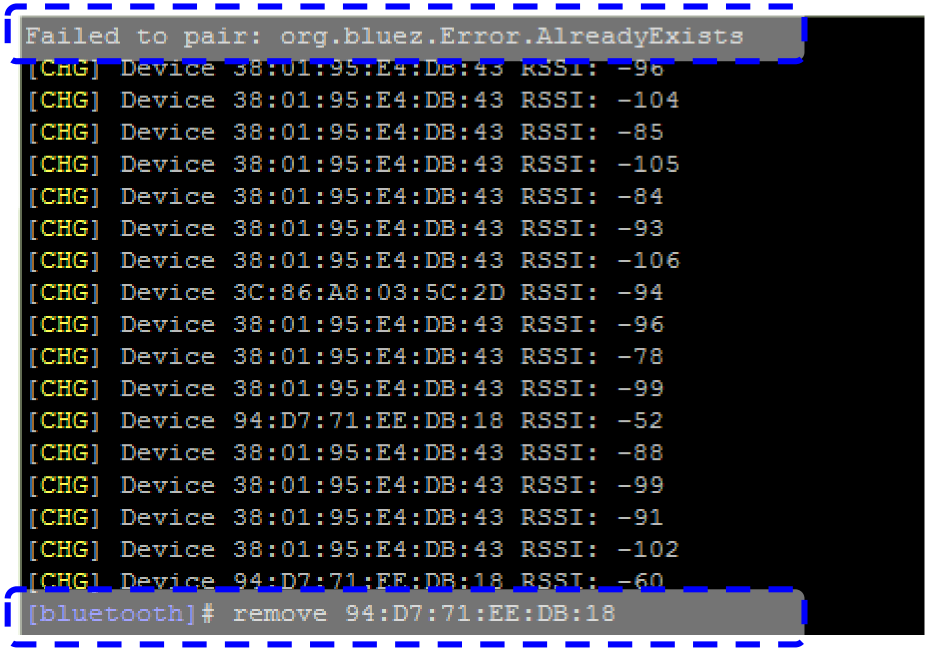
```
Pairing successful
[CHG] Device BC:44:86:AB:EB:F6 Connected: no
[NEW] Device 7E:CC:4D:7C:1C:98 7E-CC-4D-7C-1C-98
[bluetooth]# connect BC:44:86:AB:EB:F6
```

Using Bluetooth on ARTIK 5

Note

- Case of the pairing fail
  - Once you reconnect after pairing Bluetooth
  - ARTIK is paired to another device

- Solution
  - *# remove xx : xx : xx : xx : xx : xx*
  - Return to Step 2



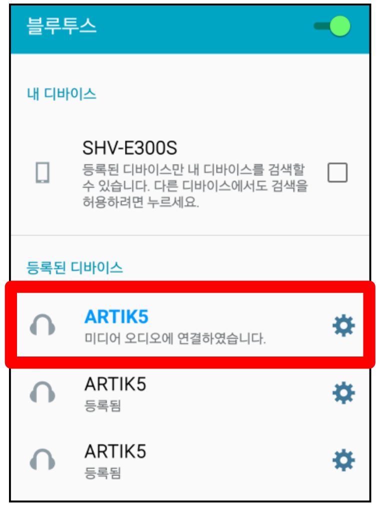
```
Failed to pair: org.bluez.Error.AlreadyExists
[CHG] Device 38:01:95:E4:DB:43 RSSI: -96
[CHG] Device 38:01:95:E4:DB:43 RSSI: -104
[CHG] Device 38:01:95:E4:DB:43 RSSI: -85
[CHG] Device 38:01:95:E4:DB:43 RSSI: -105
[CHG] Device 38:01:95:E4:DB:43 RSSI: -84
[CHG] Device 38:01:95:E4:DB:43 RSSI: -93
[CHG] Device 38:01:95:E4:DB:43 RSSI: -106
[CHG] Device 3C:86:A8:03:5C:2D RSSI: -94
[CHG] Device 38:01:95:E4:DB:43 RSSI: -96
[CHG] Device 38:01:95:E4:DB:43 RSSI: -78
[CHG] Device 38:01:95:E4:DB:43 RSSI: -99
[CHG] Device 94:D7:71:EE:DB:18 RSSI: -52
[CHG] Device 38:01:95:E4:DB:43 RSSI: -88
[CHG] Device 38:01:95:E4:DB:43 RSSI: -99
[CHG] Device 38:01:95:E4:DB:43 RSSI: -91
[CHG] Device 38:01:95:E4:DB:43 RSSI: -102
[CHG] Device 94:D7:71:EE:DB:18 RSSI: -60
[bluetooth]# remove 94:D7:71:EE:DB:18
```

Bluetooth(ARTIK 5 – Smartphone)

How to pair between ARTIK 5 and Bluetooth App

- Step 1. Pairing ARTIK 5 and Smartphone

  # Bluetoothctl
  [bluetooth] *# scan on*
  [bluetooth] *# discoverable on*
  [bluetooth] *# connect xx : xx : xx : xx : xx : xx*
  [bluetooth] *# exit*



블루투스

내 디바이스

SHV-E300S
등록된 디바이스만 내 디바이스를 검색할 수 있습니다. 다른 디바이스에서도 검색을 허용하려면 누르세요.

등록된 디바이스

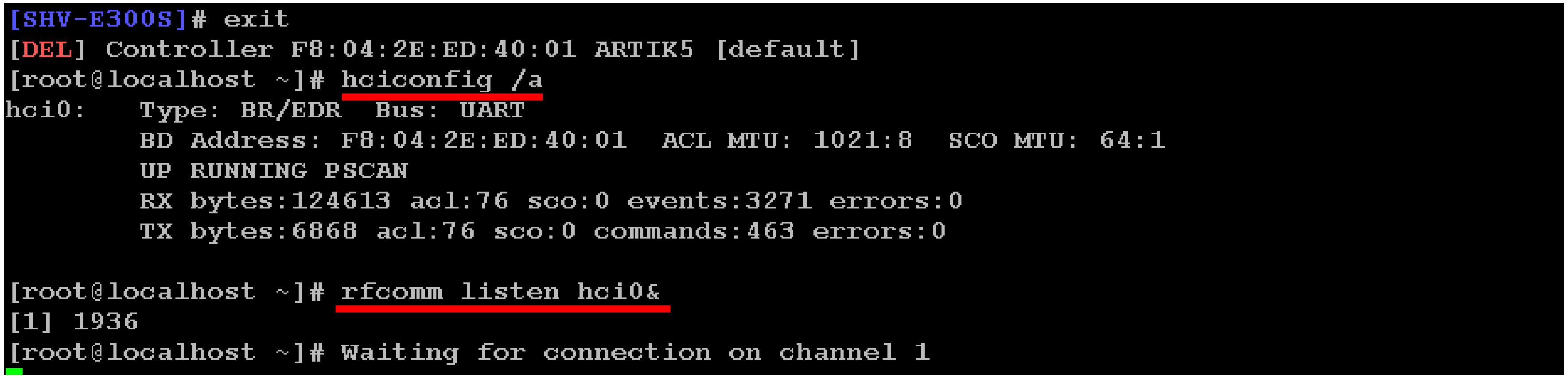
ARTIK5
미디어 오디오에 연결하였습니다.

ARTIK5
등록됨

ARTIK5
등록됨

- Step 2. Bluetooth interface check & Running the server

  # hciconfig /a
  # rfcomm listen hci0&

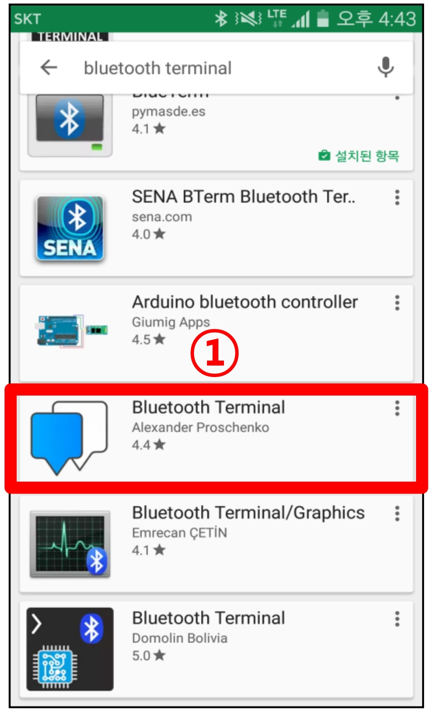


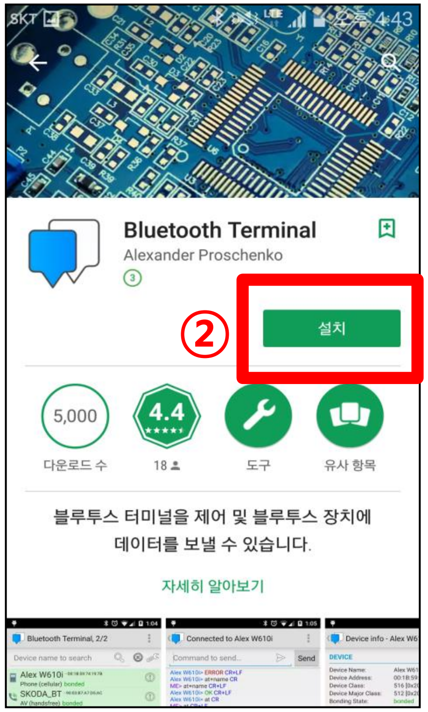
```
[SHV-E300S]# exit
[DEL] Controller F8:04:2E:ED:40:01 ARTIK5 [default]
[root@localhost ~]# hciconfig /a
hci0:   Type: BR/EDR  Bus: UART
        BD Address: F8:04:2E:ED:40:01  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING PSCAN
        RX bytes:124613 acl:76 sco:0 events:3271 errors:0
        TX bytes:6868 acl:76 sco:0 commands:463 errors:0

[root@localhost ~]# rfcomm listen hci0&
[1] 1936
[root@localhost ~]# Waiting for connection on channel 1
```

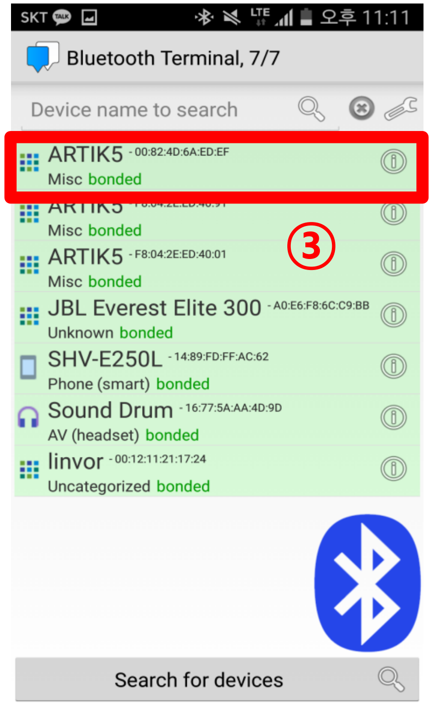
Bluetooth(ARTIK 5 – Smartphone)

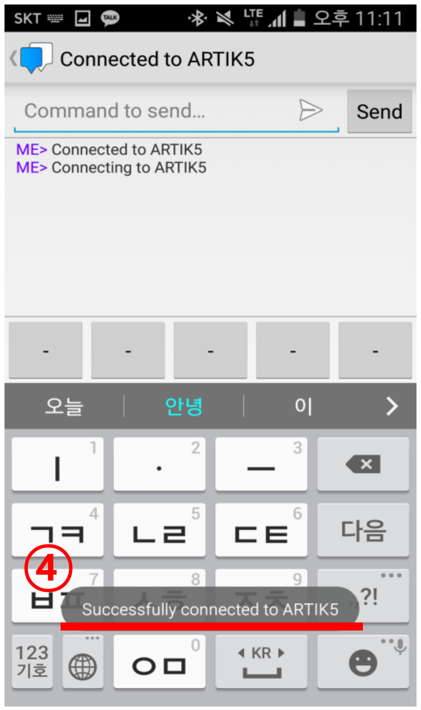
How to pair between ARTIK 5 and Bluetooth App

- Step 3. Bluetooth Terminal app download
  - Development Company : Alexander Proschenko
  - *Start the App*
  - *Click the ARTIK 5*


SKT 오후 4:43
bluetooth terminal
pymasde.es 4.1★ 설치된 항목
SENA BTerm Bluetooth Ter.. sena.com 4.0★
Arduino bluetooth controller Giumig Apps 4.5★
①
Bluetooth Terminal Alexander Proschenko 4.4★
Bluetooth Terminal/Graphics Emrecan ÇETİN 4.1★
Bluetooth Terminal Domolin Bolivia 5.0★


SKT 오후 4:43
Bluetooth Terminal
Alexander Proschenko
②
설치
5,000 다운로드 수 | 4.4 18 | 도구 | 유사 항목
블루투스 터미널을 제어 및 블루투스 장치에
데이터를 보낼 수 있습니다.
자세히 알아보기

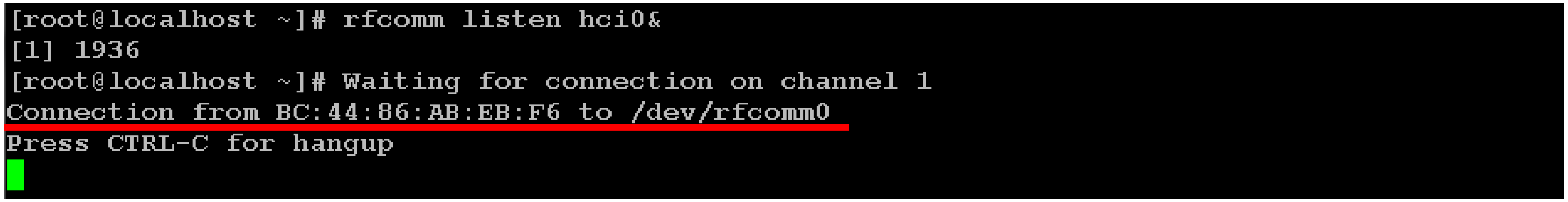

SKT 오후 11:11
Bluetooth Terminal, 7/7
Device name to search
ARTIK5 - 00:82:4D:6A:ED:EF Misc bonded
ARTIK5 Misc bonded
③
ARTIK5 - F8:04:2E:ED:40:01 Misc bonded
JBL Everest Elite 300 - A0:E6:F8:6C:C9:BB Unknown bonded
SHV-E250L - 14:89:FD:FF:AC:62 Phone (smart) bonded
Sound Drum - 16:77:5A:AA:4D:9D AV (headset) bonded
linvor - 00:12:11:21:17:24 Uncategorized bonded
Search for devices


SKT 오후 11:11
Connected to ARTIK5
Command to send... Send
ME> Connected to ARTIK5
ME> Connecting to ARTIK5
④
Successfully connected to ARTIK5

Bluetooth(ARTIK 5 – Smartphone)

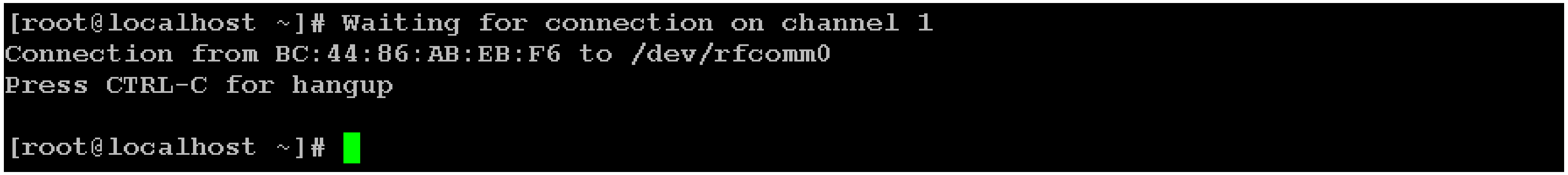
How to pair between ARTIK 5 and Bluetooth App

- Step 4. Confirm ARTIK 5 connection



```
[root@localhost ~]# rfcomm listen hci0&
[1] 1936
[root@localhost ~]# Waiting for connection on channel 1
Connection from BC:44:86:AB:EB:F6 to /dev/rfcomm0
Press CTRL-C for hangup
```

  - *Press 'Enter'*



```
[root@localhost ~]# Waiting for connection on channel 1
Connection from BC:44:86:AB:EB:F6 to /dev/rfcomm0
Press CTRL-C for hangup

[root@localhost ~]#
```

Bluetooth(ARTIK 5 – Smartphone)

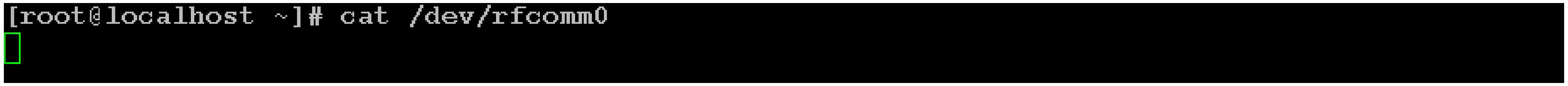
■ How to pair between ARTIK 5 and Bluetooth App

▪ Step 5. Checking the transferred data

<ARTIK 5> *# cat /dev/rfcomm0*

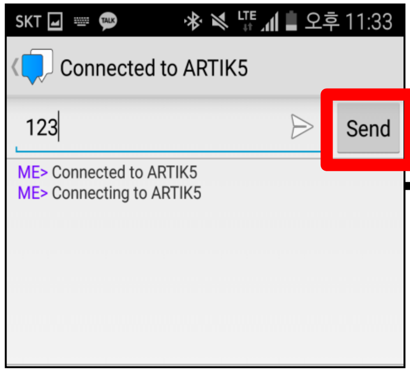
<App> *Send the data (ex. 123)*

①



```
[root@localhost ~]# cat /dev/rfcomm0
```

②



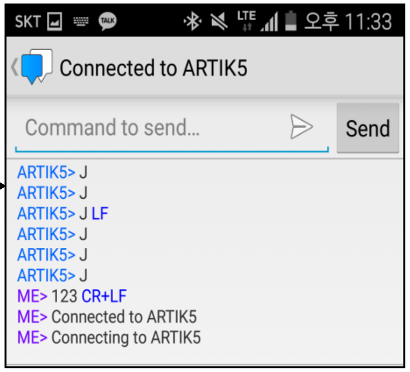
SKT 오후 11:33

Connected to ARTIK5

123

ME> Connected to ARTIK5
ME> Connecting to ARTIK5

③

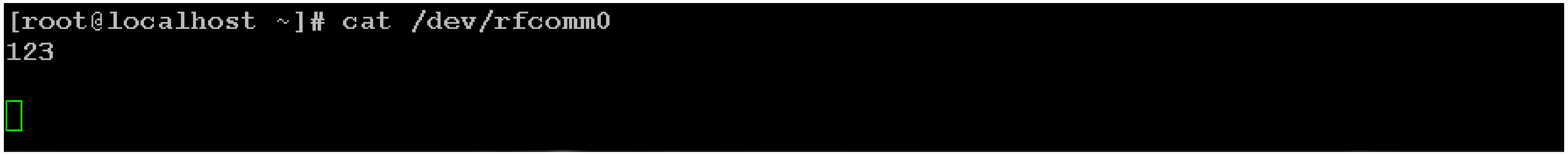


SKT 오후 11:33

Connected to ARTIK5

ARTIK5> J
ARTIK5> J
ARTIK5> J LF
ARTIK5> J
ARTIK5> J
ARTIK5> J
ME> 123 CR+LF
ME> Connected to ARTIK5
ME> Connecting to ARTIK5

④



```
[root@localhost ~]# cat /dev/rfcomm0
123
```

WiFi AP Realization

■ Introduction

Wireless or wired networks are the most important meanings for ARTIK to communicate with other devices. We can configure the network by connecting Ethernet to the wire or connecting to Wi-Fi, where ARTIK can act as a Wi-Fi access point as well as a Wi-Fi client. There have been a detailed SoftAP mode in ARTIK.

Methodology

■ Wi-Fi Access Point setting

Step 1. Connect the WiFi antenna in the lower left (ANT1) of ARTIK

Step 2. dnsmasq(DNS Masquerade server) setting

- Enter vi /etc/dnsmasq.conf to input the following command in the file or uncomment the command, which sets the range of IP addresses for DHCP allocation.

  bind-interfaces

  dhcp-range=192.168.1.2,192.168.1.100

- Interrupt connman(connection manager) service.

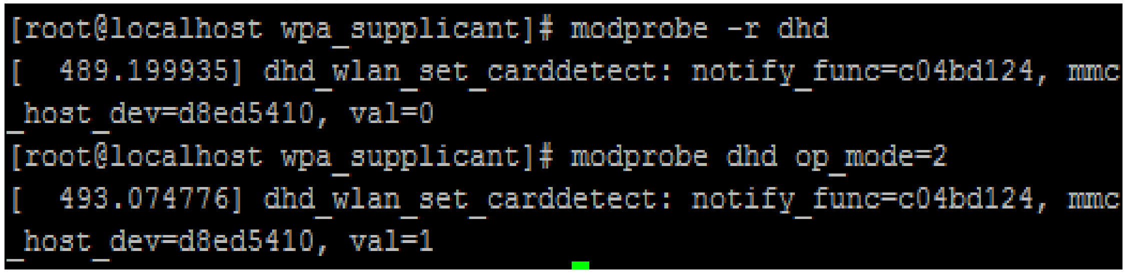
  # systemctl stop connman

- Change the settings of the network driver. If you enter the following command in order, you can see that val = 0 changes to val = 1.

  modprobe : used to add or remove modules

  # modprobe -r dhd

  # modprobe dhd op_mode=2



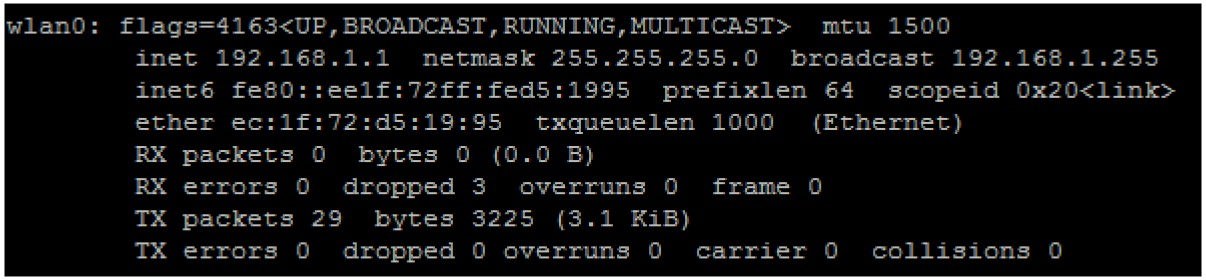
```
[root@localhost wpa_supplicant]# modprobe -r dhd
[  489.199935] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc
_host_dev=d8ed5410, val=0
[root@localhost wpa_supplicant]# modprobe dhd op_mode=2
[  493.074776] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc
_host_dev=d8ed5410, val=1
```

Methodology

■ Wi-Fi Access Point setting

- Set the IP address of Wlan0

  # ifconfig wlan0 192.168.1.1 up



```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.1  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::ee1f:72ff:fed5:1995  prefixlen 64  scopeid 0x20<link>
        ether ec:1f:72:d5:19:95  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 3  overruns 0  frame 0
        TX packets 29  bytes 3225 (3.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

We can see that the IP address you entered is assigned to wlan0 when entering the command.

- Dnsmasq Start

 # dnsmasq -C /etc/dnsmasq.conf

Methodology

■ Wi-Fi Access Point setting

Step 3. iptables setting

- iptables : a kind of firewall, usually software that enables NAT (Network Address Translation) used by Linux.

Enter the following command to change the configuration of iptables

```
sysctl net.ipv4.ip_forward=1
iptables --flush
iptables -t nat --flush
iptables --delete-chain
iptables -t nat --delete-chain
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i wlan0 -j ACCEPT
```

Methodology

■ Wi-Fi Access Point setting

Step 4. hostapd setting

- Enter vi /etc/hostapd/hostapd.conf and then enter the text below. ssid and wpa_passphrase are changed into WiFi ID and password.

```
interface=wlan0
driver=nl80211
ssid=ARTIK_AP
auth_algs=1
hw_mode=g
channel=6
wpa=2
wpa_passphrase=artik@iot
wpa_pairwise=TKIP CCMP
rsn_pairwise=CCMP
```

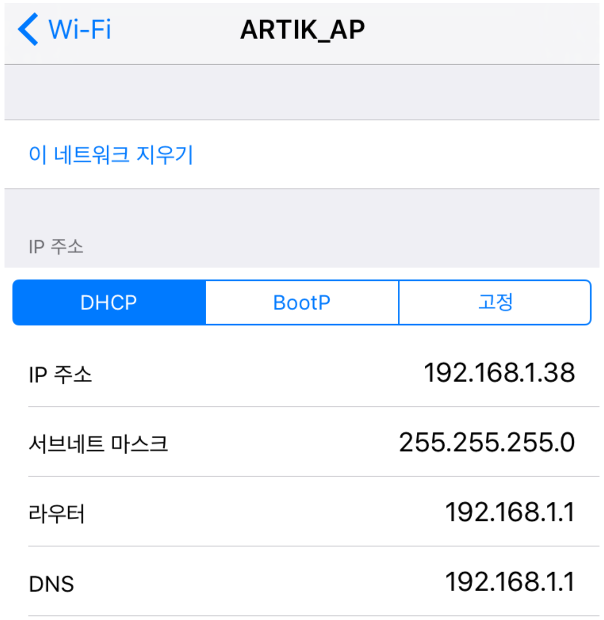
- # hostapd /etc/hostapd/hostapd.conf -B Activate hostapd.

Methodology

■ Wi-Fi Access Point setting

Step 5. WiFi connection

You can connect to ARTIK_AP using mobile phone as the picture on the right.



IP 주소	192.168.1.38
서브네트 마스크	255.255.255.0
라우터	192.168.1.1
DNS	192.168.1.1

Step 6. Exit

- Enter the following command to exit AP mode, .

```
killall hostapd
modprobe -r dhd
modprobe dhd op_mode=0
ifconfig wlan0 up
if [ -f "/usr/lib/systemd/system/wpa_supplicant.service" ]; then
systemctl restart wpa_supplicant
fi
```

Methodology

■ AP mode scripting

- We can confirm that AP mode works well in the part of **Wi-Fi Access Point setting**. However, there are so many commands to input, so let us write a script to turn on and off the AP mode for convenience.

Most of the code consists of commands from the part of **Wi-Fi Access Point setting**.

However, if you have previously execute AP mode and exit without entering a normal exit code, when you attempt to reconnect again, you will see an error message (delete /var/run/hostapd/wlan0 file) and fail to reconnect.

So we added a command (`rm -f var/run/hostapd/wlan0`) to delete the file before the script.

Methodology

AP mode scripting

- Source Code

APmode.sh

```
#! /bin/bash
rm -f /var/run/hostapd/wlan0
echo "[Configuring dnsmasq...]"
sed -i 's/#bind-interfaces/bind-
interfaces/g' /etc/dnsmasq.conf
echo dhcp-
range=192.168.1.2,192.168.1.100>>/etc/dnsmasq.conf
systemctl stop connman
ifconfig eth0 up
dhclient eth0
modprobe -r dhd
modprobe dhd op_mode=2
ifconfig wlan0 192.168.1.1 up
dnsmasq -C /etc/dnsmasq.conf
echo "[Configuring iptables...]"
sysctl net.ipv4.ip_forward=1
iptables --flush
iptables -t nat --flush
iptables --delete-chain
iptables -t nat --delete-chain
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i wlan0 -j ACCEPT
echo "[Configuring hostapd...]"
sed -
i 's/#interface=/interface=wlan0/g' /etc/hostapd/hosta
pd.conf
sed -
i 's/#driver=/driver=nl80211/g' /etc/hostapd/hostapd.conf
sed -
i 's/#ssid=/ssid=ARTIK_AP/g' /etc/hostapd/hostapd.conf
sed -
i 's/#auth_algs/auth_algs=1/g' /etc/hostapd/hostapd.conf
sed -i 's/#hw_mode=/hw_mode=g/g' /etc/hostapd/hostapd.conf
sed -i 's/#channel=/channel=6/g' /etc/hostapd/hostapd.conf
sed -
i 's/#wpa_passphrase=/wpa_passphrase=artik@iot/g' /etc/hos
tapd/hostapd.conf
sed -i 's/#wpa_pairwise=TKIP/wpa_pairwise=TKIP
CCMP/g' /etc/hostapd/hostapd.conf
sed -
i 's/#rsn_pairwise=CCMP/rsn_pairwise=CCMP/g' /etc/hostapd/
hostapd.conf
sed -i 's/#wpa=3/wpa=2/g' /etc/hostapd/hostapd.conf

hostapd /etc/hostapd/hostapd.conf -B
```

APmode_end.sh

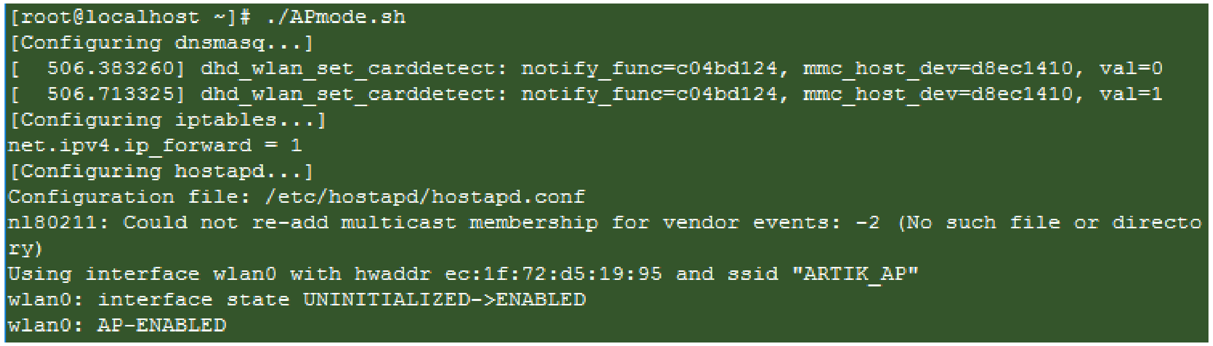
```
#! /bin/bash
killall hostapd
modprobe -r dhd
modprobe dhd op_mode=0
ifconfig wlan0 up
if [ -
f "/usr/lib/systemd/system/wpa_supplicant.service" ]; then
  systemctl restart wpa_supplicant
        fi
```

Result

■ Execution result

Wi-Fi Access Point connects ARTIK to the Internet via Ethernet LAN (eth0), and then supplies wireless network to other devices. Therefore Network Address Translation (NAT) to set up dnsmasq, iptables, and hostapd is necessary. Though it is much easier to input the command for the step and check the normal operation, it is more convenient for us to write the script

- When the script is executed, it is output as below and it is confirmed that a WiFi named ARTIK_AP is created.

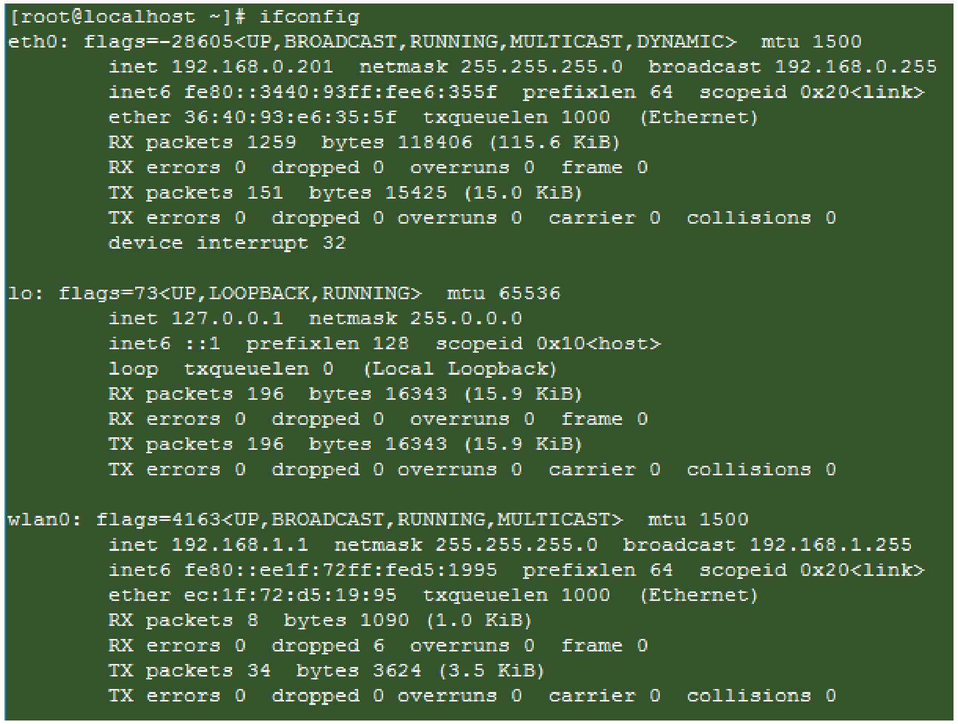


```
[root@localhost ~]# ./APmode.sh
[Configuring dnsmasq...]
[  506.383260] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=0
[  506.713325] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=1
[Configuring iptables...]
net.ipv4.ip_forward = 1
[Configuring hostapd...]
Configuration file: /etc/hostapd/hostapd.conf
nl80211: Could not re-add multicast membership for vendor events: -2 (No such file or directo
ry)
Using interface wlan0 with hwaddr ec:1f:72:d5:19:95 and ssid "ARTIK_AP"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

Result

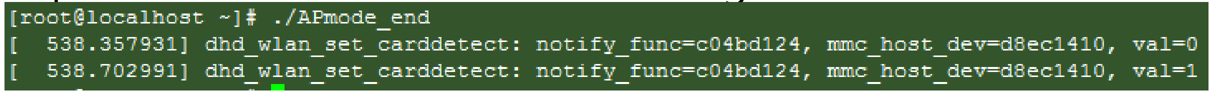
Execution result

- Ifconfig shows that eth0 is automatically assigned an IP address when wired LAN is connected and wlan0 is set to 192.168.1.1 shown as below.



```
[root@localhost ~]# ifconfig
eth0: flags=-28605<UP,BROADCAST,RUNNING,MULTICAST,DYNAMIC>  mtu 1500
        inet 192.168.0.201  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::3440:93ff:fee6:355f  prefixlen 64  scopeid 0x20<link>
        ether 36:40:93:e6:35:5f  txqueuelen 1000  (Ethernet)
        RX packets 1259  bytes 118406 (115.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 151  bytes 15425 (15.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 32

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 0  (Local Loopback)
        RX packets 196  bytes 16343 (15.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 196  bytes 16343 (15.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.1  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::ee1f:72ff:fed5:1995  prefixlen 64  scopeid 0x20<link>
        ether ec:1f:72:d5:19:95  txqueuelen 1000  (Ethernet)
        RX packets 8  bytes 1090 (1.0 KiB)
        RX errors 0  dropped 6  overruns 0  frame 0
        TX packets 34  bytes 3624 (3.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Output is shown as below when exiting.



```
[root@localhost ~]# ./APmode_end
[  538.357931] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=0
[  538.702991] dhd_wlan_set_carddetect: notify_func=c04bd124, mmc_host_dev=d8ec1410, val=1
```

Conclusions

- ARTIK5 has used Access Point Mode, which can supply wireless network to other devices using Wi-Fi.

- Since ARTIK5 has two Wi-Fi antennas at 2.4GHz and 5GHz, the initial goal was to have two AP modes run at the same time.

- In other words, wlan0 and wlan1 were controlled at the same time, but it was not able to proceed because of limitation of one built-in wireless LAN card in hardware.