UNIT

18

라즈베리 파이 블루투스

로봇SW 교육원

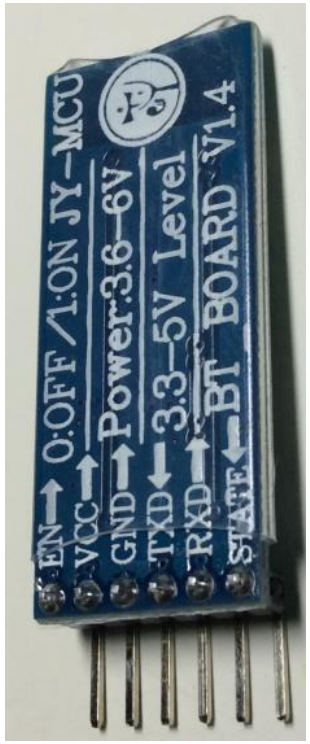
최상훈(shchoi82@gmail.com)

학습 목표

- 블루투스 모듈 제어
- 블루투스 동글 제어

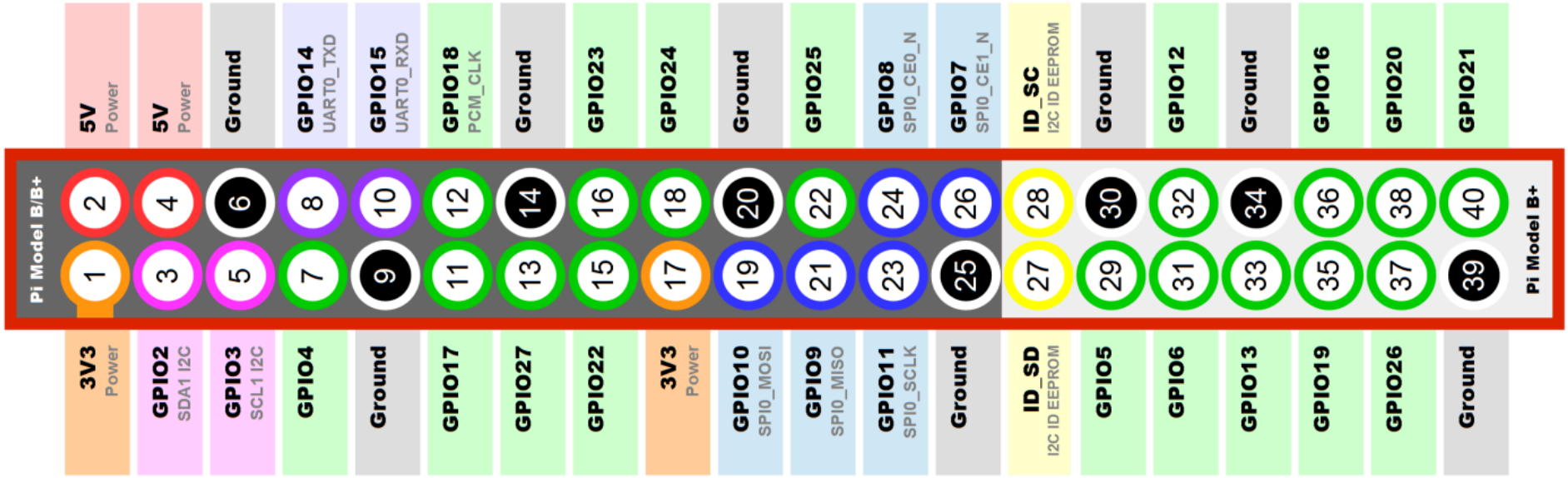
Bluetooth Module

- Bluetooth Slave UART Board
  - UART**인터페이스용 블루투스모듈**
  - slave/device mode



라즈베리 파이 GPIO

<라즈베리 파이 B+의 P1 헤더 핀 GPIO 배치도>



www.raspberrypi-spy.co.uk

wiringPi 라이브러리

- **라즈베리파이** GPIO **라이브러리**
- *GPIO Interface library for the Raspberry Pi*
- *http://wiringpi.com/*

실습 1 : wiringPi 라이브러리 설치(1/3)

- **라즈베리 파이 업데이트 및 업그레이드**

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

- wiringpPi **라이브러리 다운로드**

```
$ sudo apt-get install git-core
```

```
$ git clone git://git.drogon.net/wiringPi
```

- wiringpPi **라이브러리 빌드 및 설치**

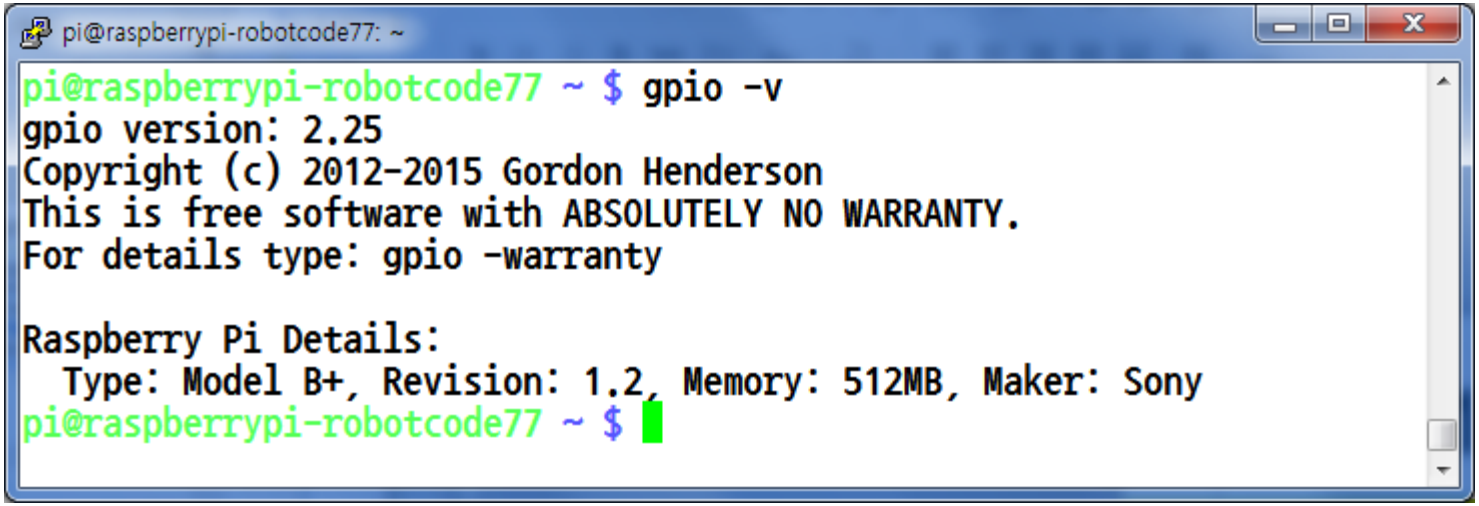
```
$ cd wiringPi
```

```
$ ./build
```

실습 1 : wiringPi 라이브러리 설치(2/3)

- 설치 확인

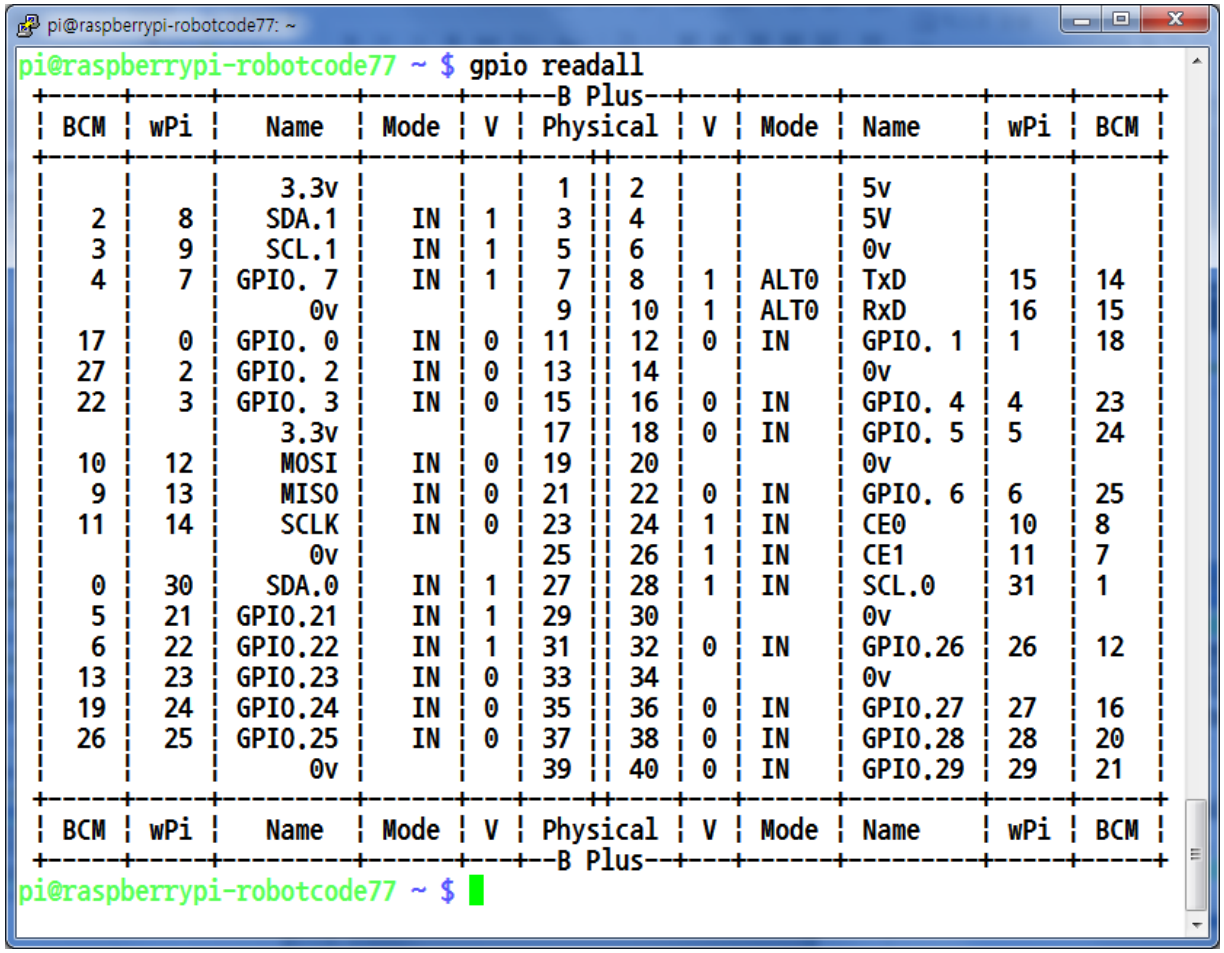
```
$ gpio -v
```



실습 1 : wiringPi 라이브러리 설치(3/3)

- GPIO 핀 정보 확인

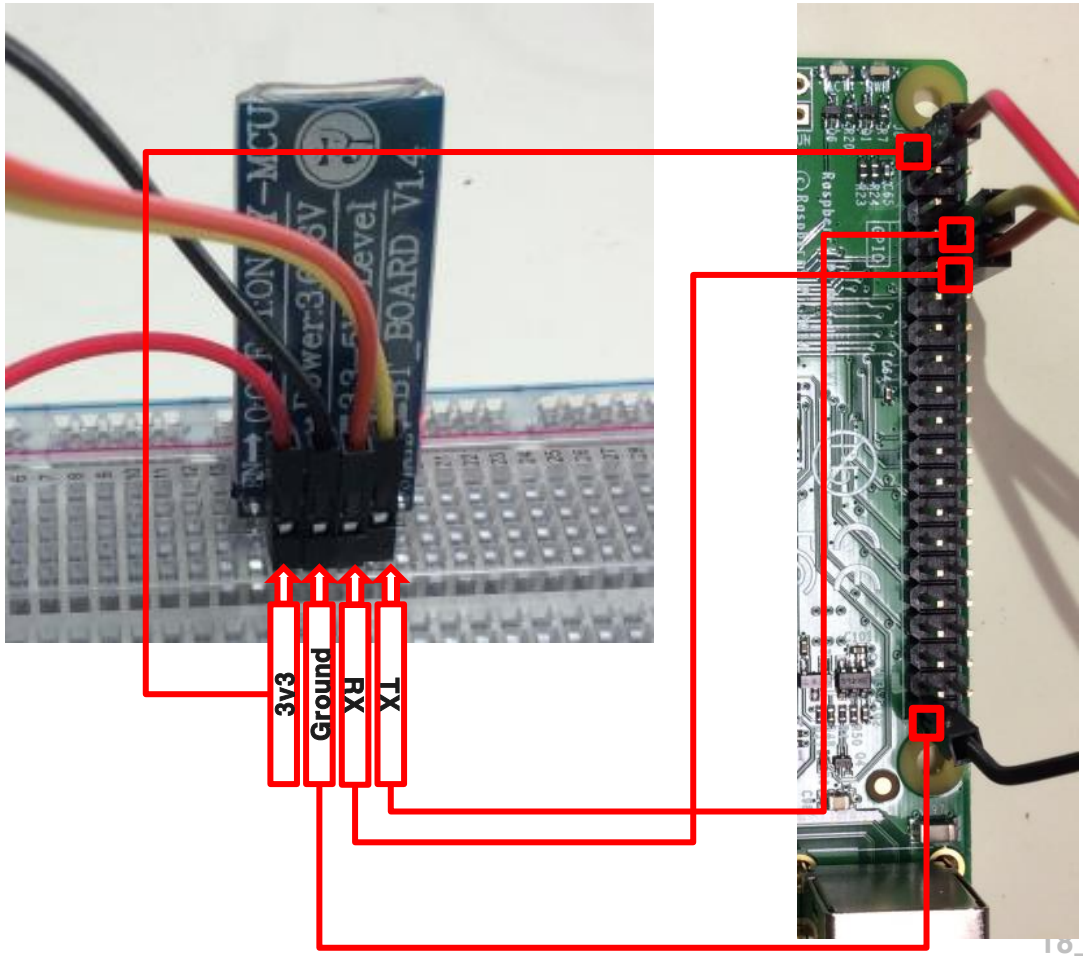
```
$ gpio readall
```



```
pi@raspberrypi-robotcode77: ~
pi@raspberrypi-robotcode77 ~ $ gpio readall
 +-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 |   IN | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 |   IN | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 1 | ALT0 | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT0 | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |   IN | 0 | 11 || 12 | 0 | IN   | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI |   IN | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO |   IN | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK |   IN | 0 | 23 || 24 | 1 | IN   | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | IN   | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+--B Plus--+---+------+---------+-----+-----+
pi@raspberrypi-robotcode77 ~ $
```

실습 2 : Bluetooth 모듈 설정(1/12)

- wiringPi **라이브러리를 이용한 시리얼 통신**
- **구성**
  - Bluetooth Slave UART Board



실습 2 : Bluetooth 모듈 설정(2/12)

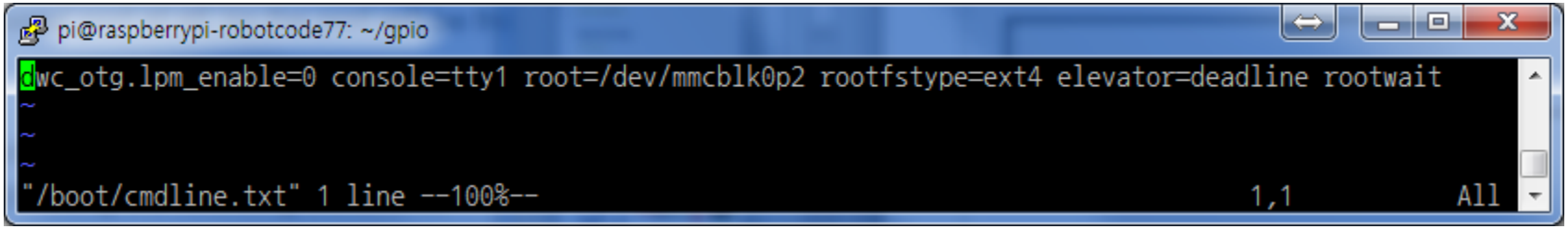
- /boot/cmdline.txt **파일 수정**
  - console, kgdboc **의** ttyAMA0 **부분 삭제**
  - **변경 전**

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200
console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline
rootwait
```

  - **변경 후**

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

- **$ sudo vim /boot/cmdline.txt**



실습 2 : Bluetooth 모듈 설정(3/12)

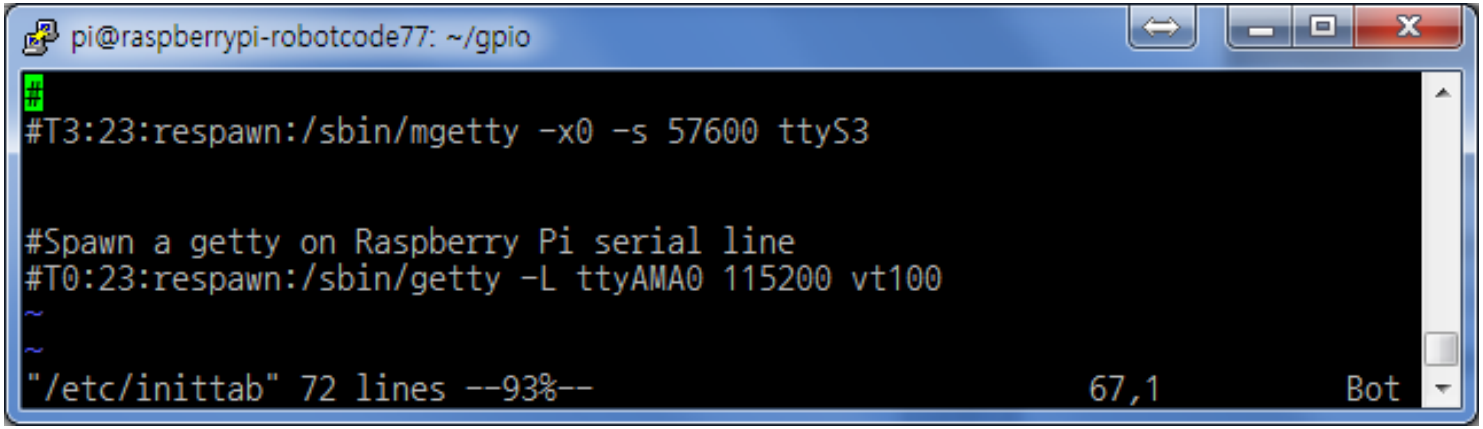
- /etc/inittab **파일 수정**
  - **변경 전**

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

  - **변경 후**

```
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

- **$ sudo vim /etc/inittab**



실습 2 : Bluetooth 모듈 설정(4/12)

- **재부팅**
  - sudo reboot

- **SSH 재접속**

- **minicom 설치**

```
$ sudo apt-get install minicom
```

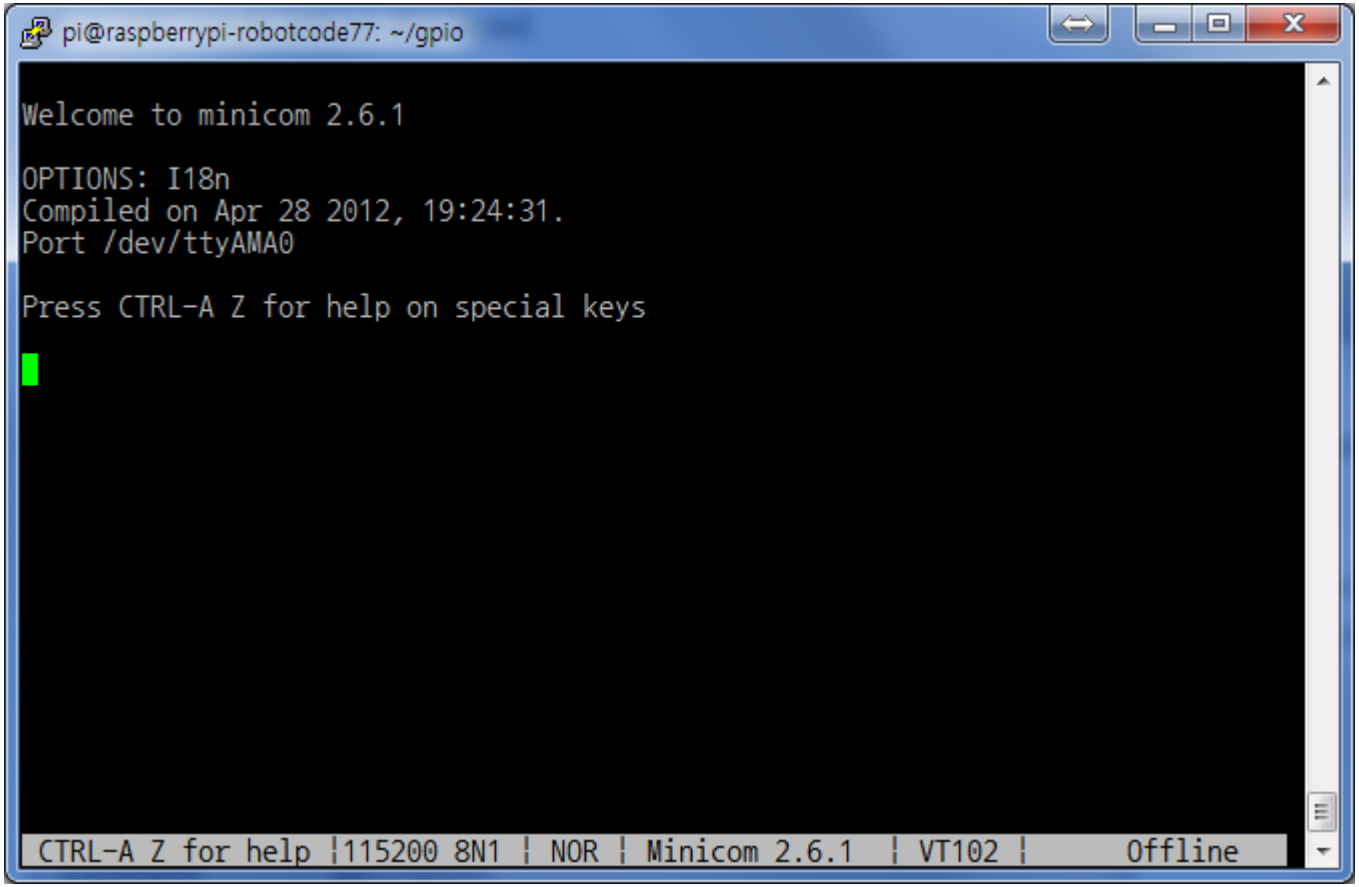
- **Bluetooth 접속**

```
$ minicom -b 9600 -o -D /dev/ttyAMA0
```

실습 2 : Bluetooth 모듈 설정(5/12)

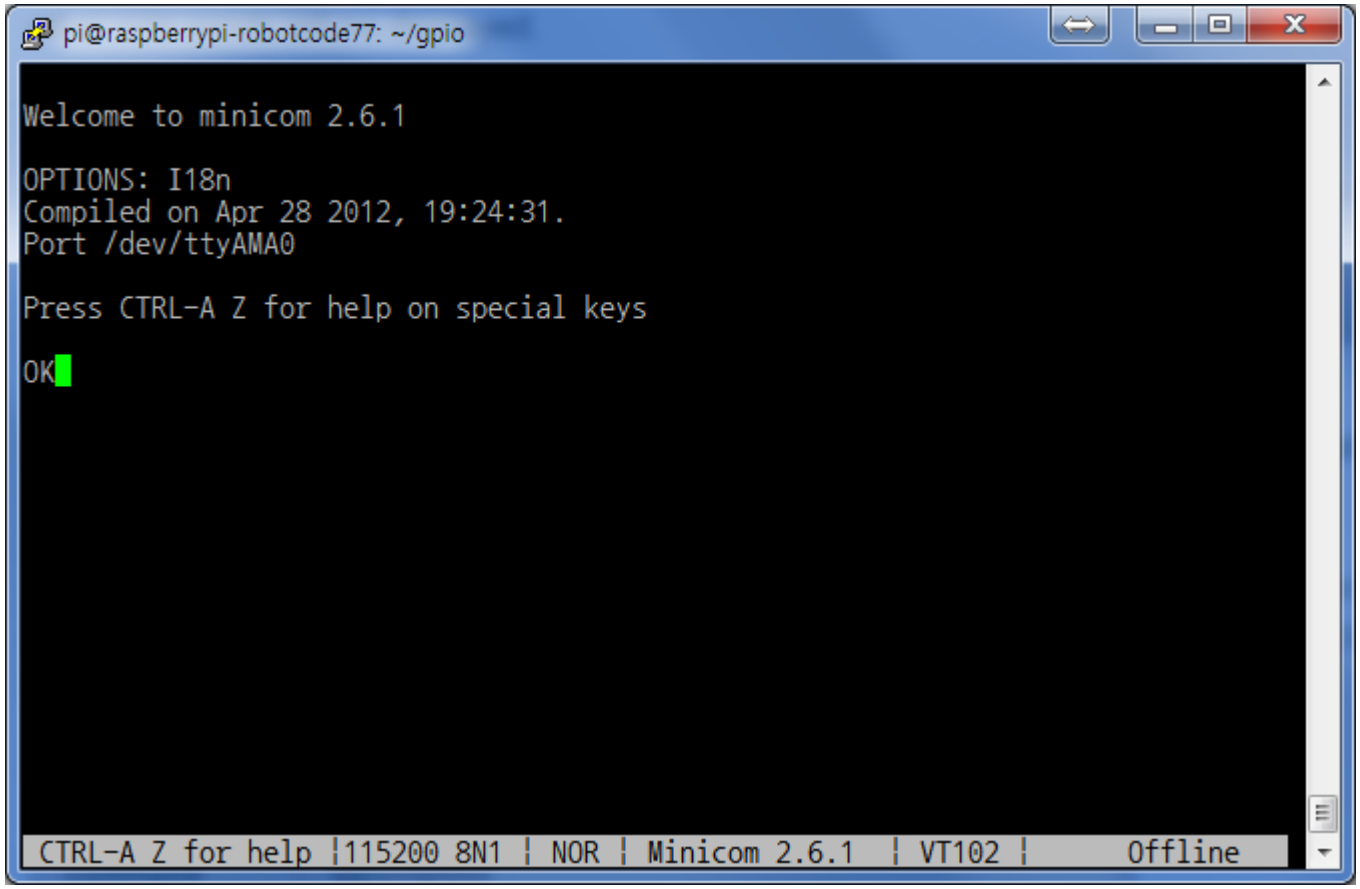
- **minicom 실행화면**

```
$ minicom -b 9600 -o -D /dev/ttyAMA0
```



실습 2 : Bluetooth 모듈 설정(6/12)

- **테스트**
  - **AT 키보드 입력후 OK 확인**



실습 2 : Bluetooth 모듈 설정(7/12)

- **명령어 종류**

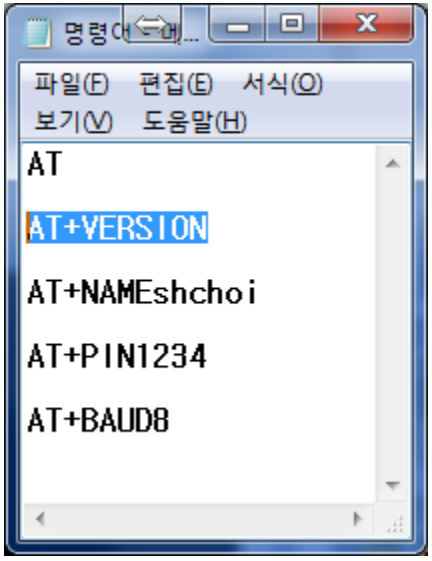
Command	Description
AT	Bluetooth module 테스트
AT+VERSION	모듈 버전 확인
AT+NAME이름	Bluetooth ID(이름) 설정(최대 20자)
AT+PINnnnn	핀번호(nnnn) 설정
AT+BAUDn	baud rate(n) 설정 2: 2400bps 3: 4800bps 4: 9600bps 5: 19200bps 6: 38400bps 7: 57600bps 8: 115200bps

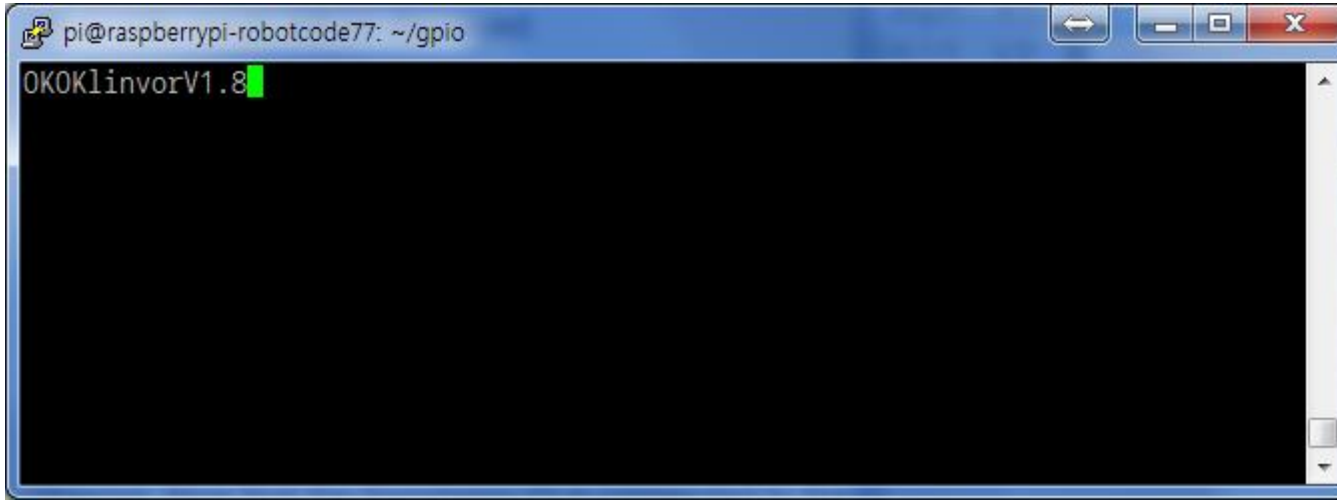
- **기본설정 상태**
  - Baud rate: 9600
  - PIN: 1234

실습 2 : Bluetooth 모듈 설정(8/12)

- **명령어 입력방법**
  - **클립보드를 이용해 붙여넣음**
  - **Ctrl + c**
  - **Shift + Insert 또는 마우스 우클릭**

- **예)**
  1. **AT+VERSION 클립보드에 복사(Ctrl+c)**
  2. **터미널 창에서 Shift + Insert 또는 마우스 우클릭**





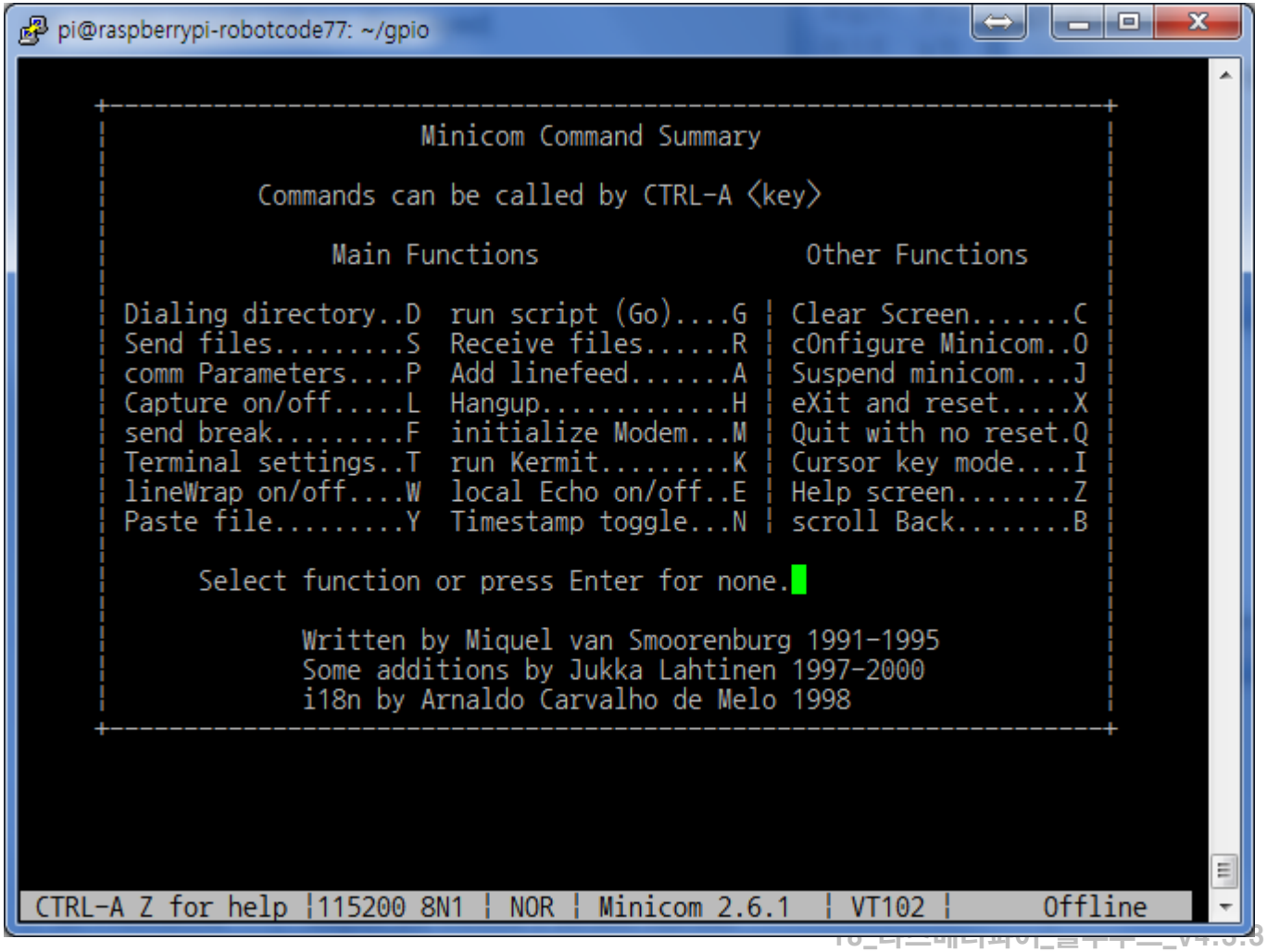
실습 2 : Bluetooth 모듈 설정(9/12)

- **이름 변경**
  - **AT+NAMEshchoi-bt**

- **baud rate 변경**
  - **AT+BAUD8**
  - **minicom 재접속 필요**

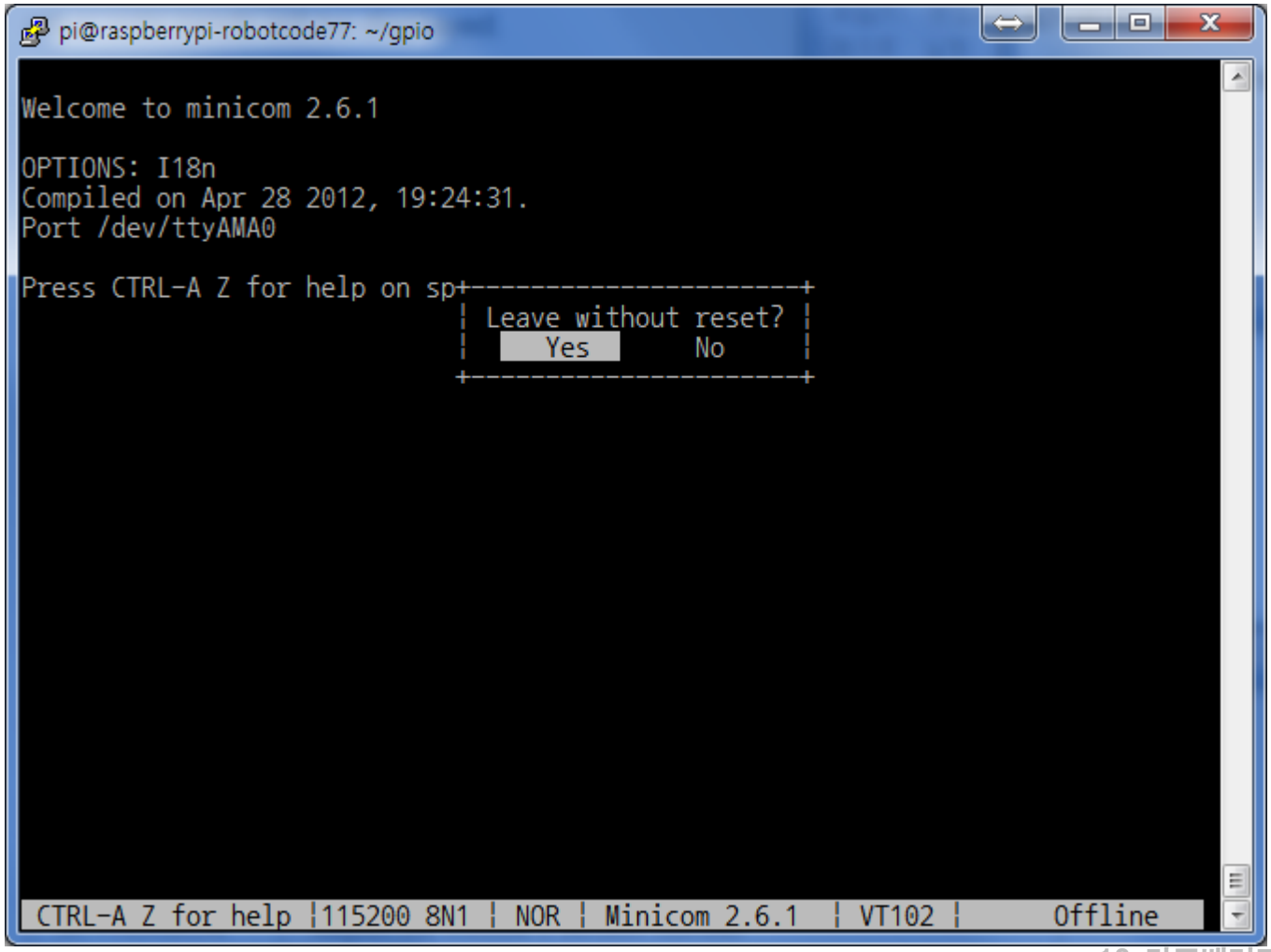
실습 2 : Bluetooth 모듈 설정(10/12)

- minicom 종료
  - Ctrl + a 입력
  - z 입력



실습 2 : Bluetooth 모듈 설정(11/12)

- q 입력
- Yes확인 후 Enter



실습 2 : Bluetooth 모듈 설정(12/12)

- **minicom 실행**

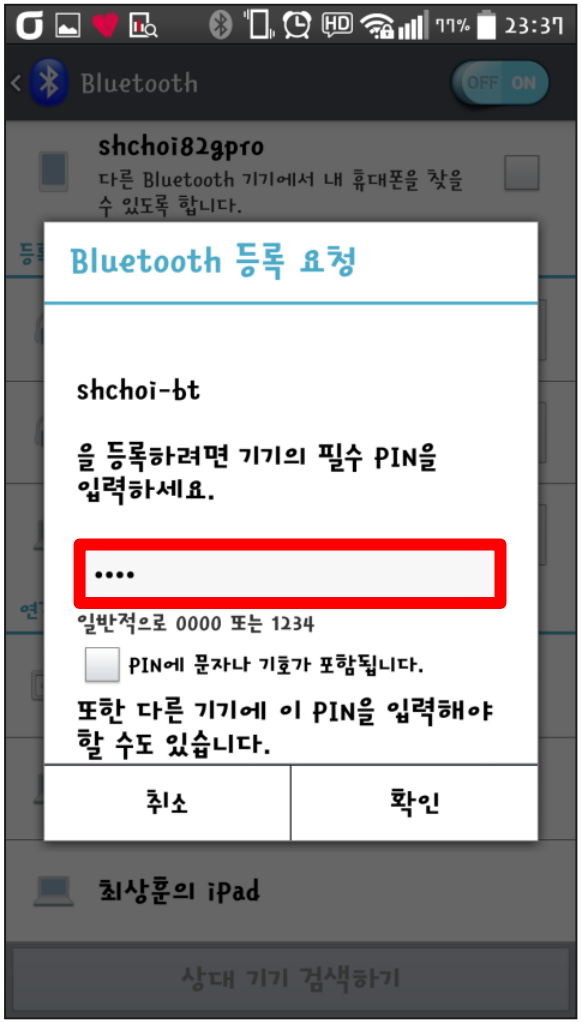
```
$ minicom -b 115200 -o -D /dev/ttyAMA0
```

- **Pin 번호변경**
  - AT+PIN5216

- **minicom 종료**

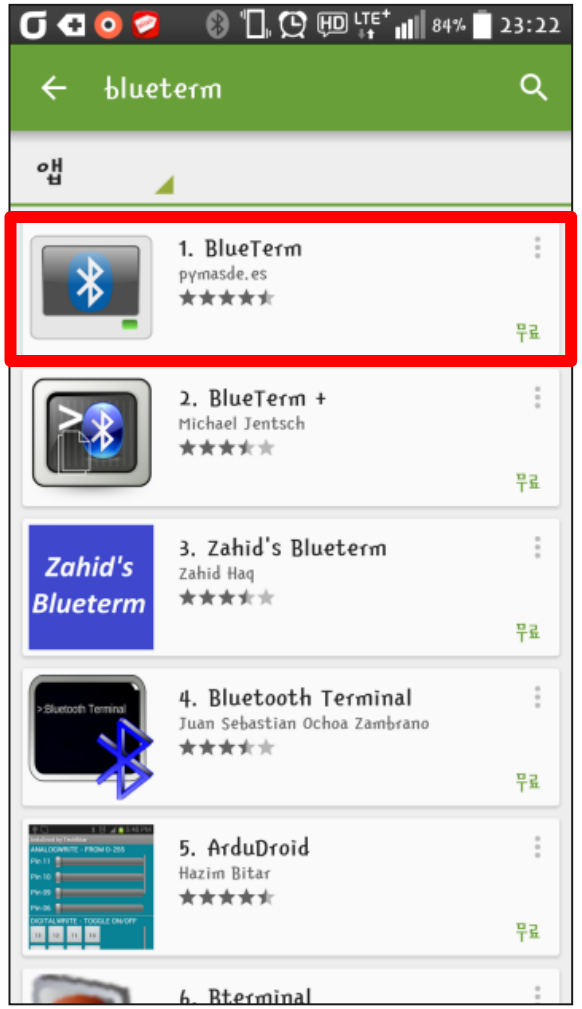
실습 3 : Bluetooth 페어링(1/2)

- **안드로이드** bluetooth **기기 등록**

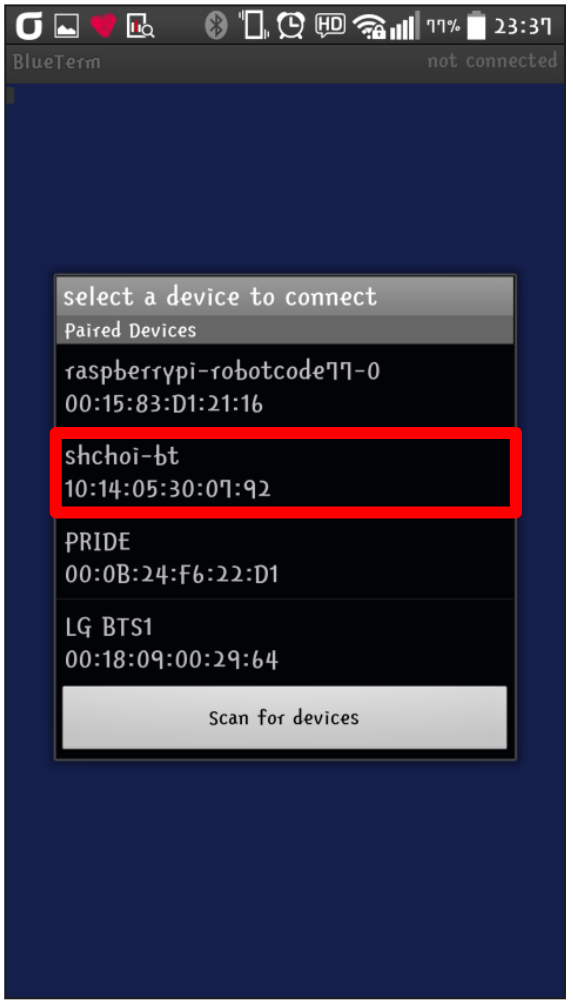


실습 3 : Bluetooth 페어링(2/2)

- **안드로이드** bluetooth **터미널 앱 설치**
  - blueterm **설치**



  - **연결(페어링)**



실습 4 : Bluetooth 통신(1/4)

- **안드로이드에서 RaspberryPi로 데이터 전송**
  - uartEx1.c

```
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include <wiringPi.h>
#include <wiringSerial.h>

#define EXIT_SUCC 0
#define EXIT_FAIL 1

int main()
{
    int fd;
    int data;

    setbuf(stdout, NULL);

    if(wiringPiSetupGpio() == -1){
        fprintf(stdout, "Unable to start wiringPi : %s\n", strerror(errno));
        return EXIT_FAIL;
    }

    // 시리얼 통신 초기화 및 속도 설정
    if((fd = serialOpen("/dev/ttyAMA0", 115200)) < 0)
    {
        fprintf(stderr, "Unable to open serial device : %s\n", strerror(errno));
        return EXIT_FAIL;
    }
```

실습 4 : Bluetooth 통신(2/4)


```
    printf("\nRaspberry Pi UART daemon start\n");
    serialPuts(fd, "Here I'm the Raspberry Pi.\r\n");   // to serial
    serialPuts(fd, "Write a message.\r\n");             // to serial

    while(1){
        data = serialGetchar(fd);               // from serial
        printf("%c",data);
    }
    return EXIT_SUCC;
}
```

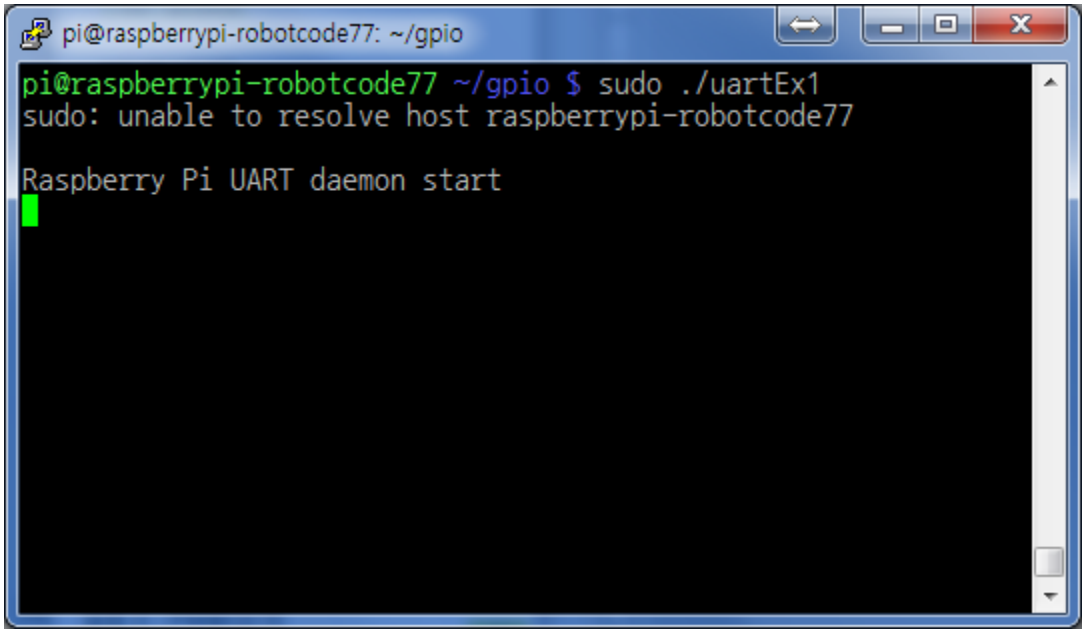
- **컴파일**

```
$ gcc -Wall -W -lwiringPi uartEx1.c -o uartEx1
```

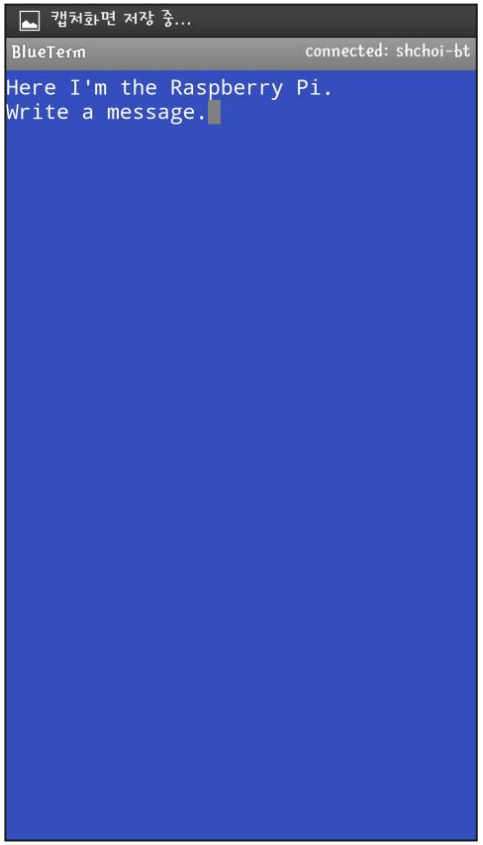
- **실행**

```
$ sudo ./uartEx1
```

실습 4 : Bluetooth 통신(3/4)

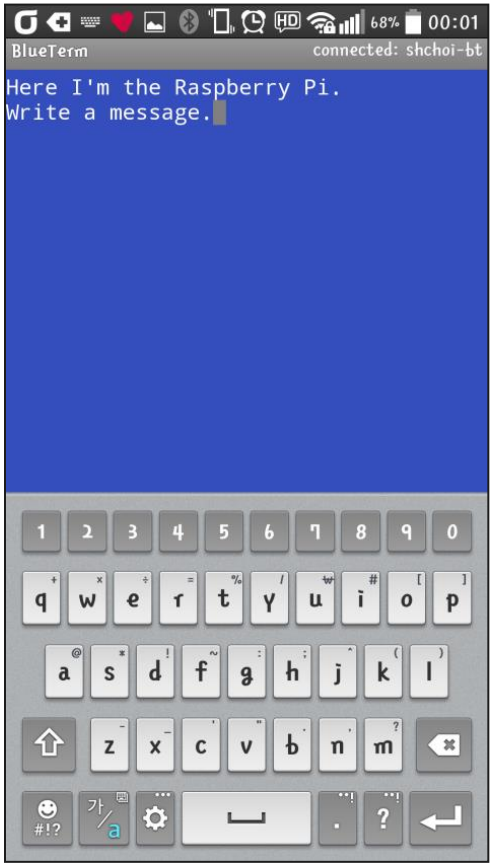


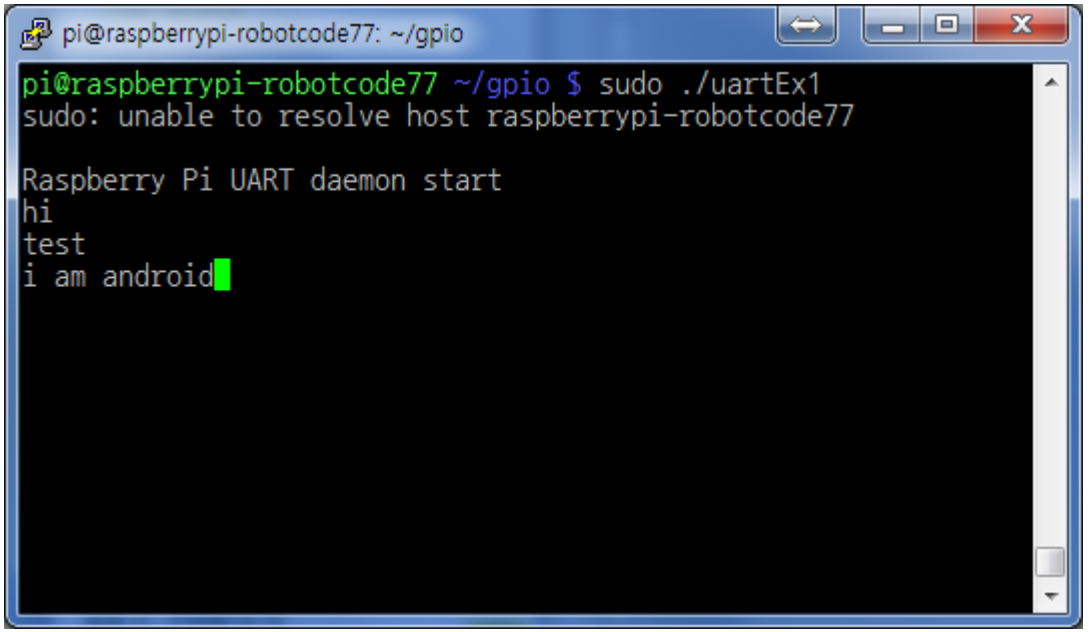
< Raspberry Pi >



실습 4 : Bluetooth 통신(4/4)

- 통신 테스트
  - 안드로이드의 터미널에 텍스트를 입력하면
  - RaspberryPi의 터미널창 그대로 출력됨





< Raspberry Pi >

실습 5 : Bluetooth 통신 (1/4)

- RaspberryPi에서 안드로이드로 데이터 전송
  - uartEx2.c

```
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include <wiringPi.h>
#include <wiringSerial.h>

#define EXIT_SUCC 0
#define EXIT_FAIL 1

int main()
{
    int fd;
    int data;

    setbuf(stdout, NULL);
    setbuf(stdin, NULL);

    if(wiringPiSetupGpio() == -1){
        fprintf(stdout, "Unable to start wiringPi : %s\n", strerror(errno));
        return EXIT_FAIL;
    }

    // 시리얼 통신 초기화 및 속도 설정
    if((fd = serialOpen("/dev/ttyAMA0", 115200)) < 0)
    {
        fprintf(stderr, "Unable to open serial device : %s\n", strerror(errno));
        return EXIT_FAIL;
    }
```

실습 5 : Bluetooth 통신 (2/4)


```
    printf("\nRaspberry Pi UART daemon start\n");
    serialPuts(fd, "Here I'm the Raspberry Pi.\r\n");   // to serial
    serialPuts(fd, "Write a message.\r\n");             // to serial

    while(1){
        if((data = fgetc(stdin)) == EOF){
            printf("EOF\n");
            break;
        }
        if(data == '\n'){
            serialPutchar(fd, '\r');                    // to serial
        }
        serialPutchar(fd, data);                        // to serial
    }
    return EXIT_SUCC;
}
```

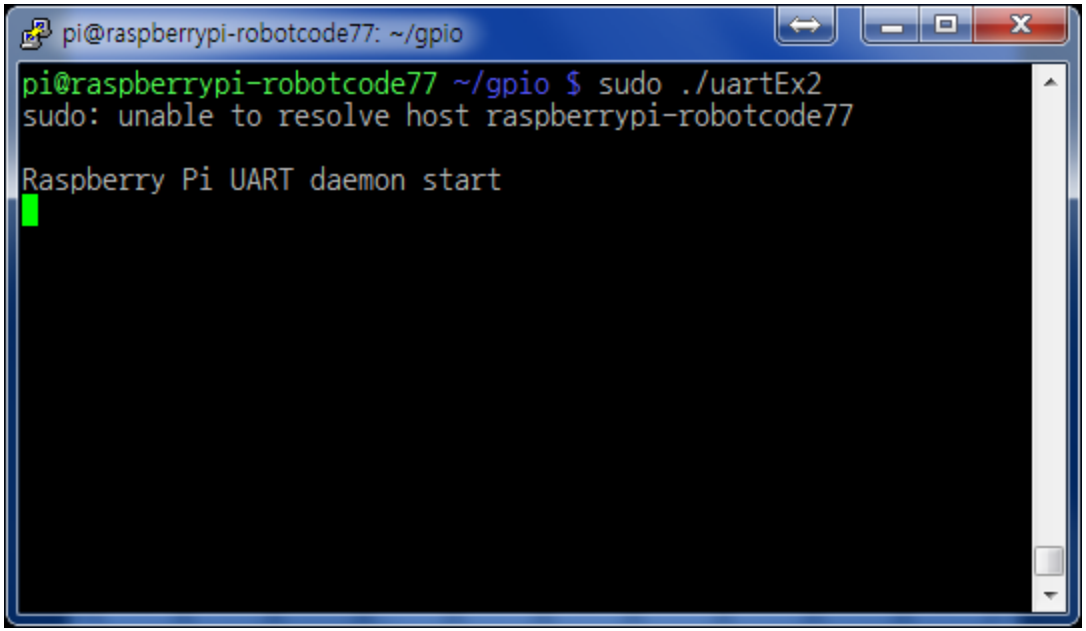
- **컴파일**

```
$ gcc -Wall -W -lwiringPi uartEx2.c -o uartEx2
```

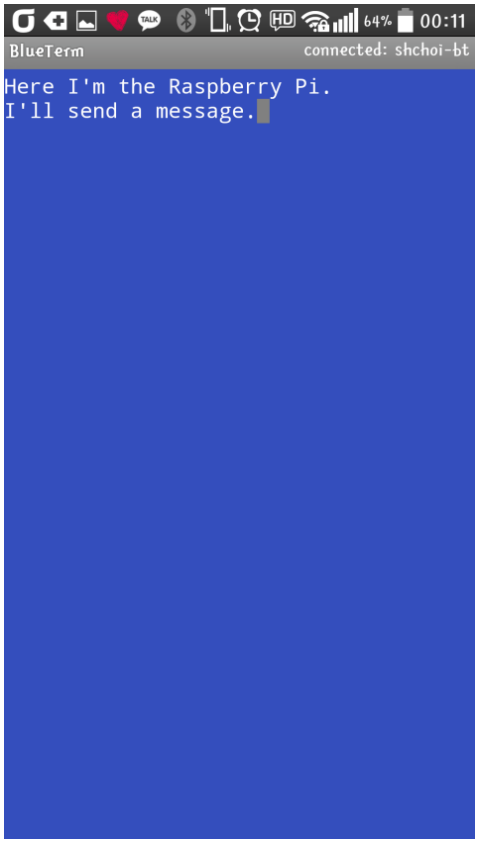
- **실행**

```
$ sudo ./uartEx2
```

실습 5 : Bluetooth 통신 (3/4)

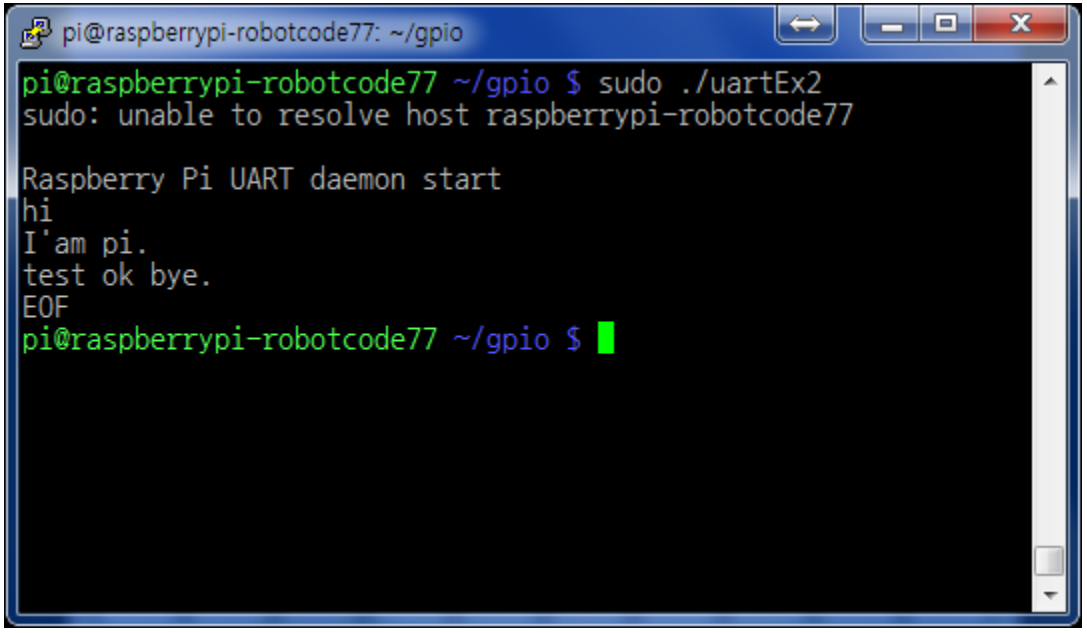


< Raspberry Pi >

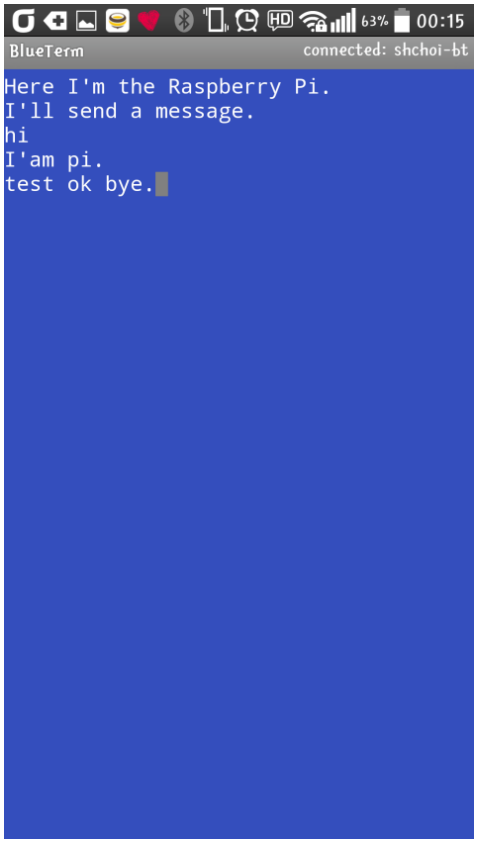


실습 5 : Bluetooth 통신 (4/4)

- **통신 테스트**
  - **RaspberryPi의 터미널에 텍스트 입력하면**
  - **안드로이드 터미널창에 그대로 출력됨**



< Raspberry Pi >

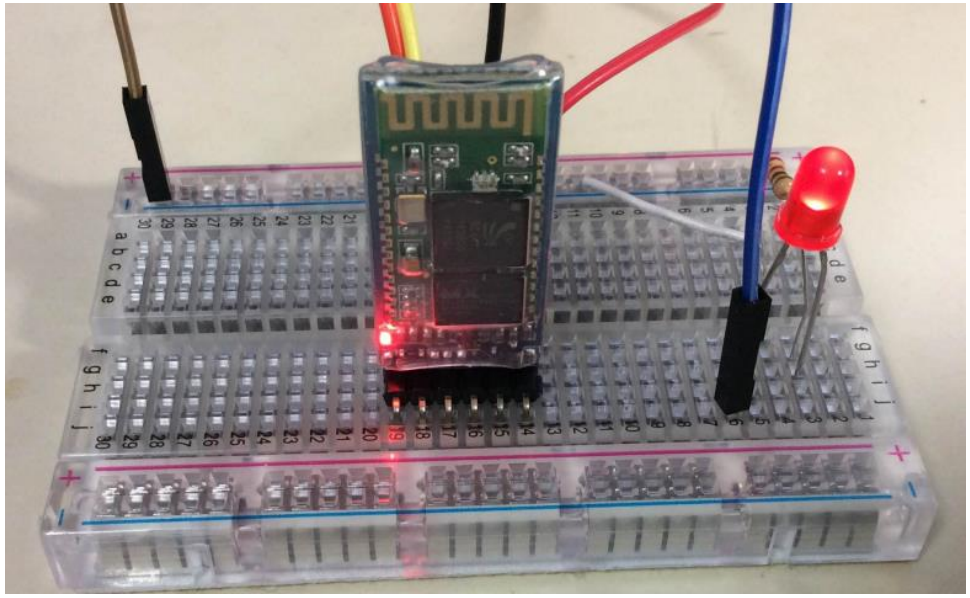


미션 1 : LED 제어

- Bluetooth 통신을 통한 Raspberry Pi GPIO 제어하기
  - 메뉴출력

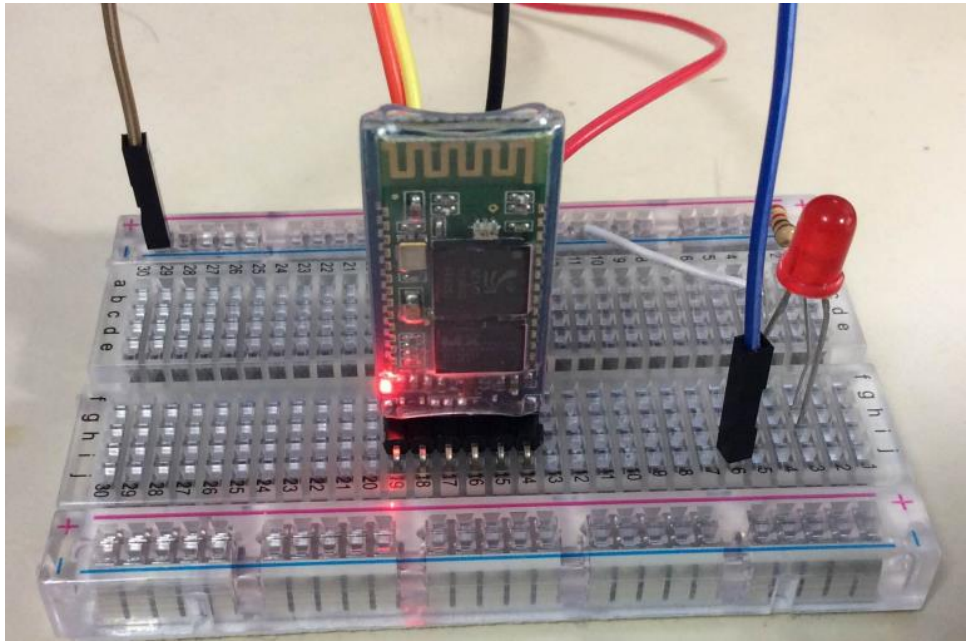
미션 1 : LED 제어

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 LED를 ON



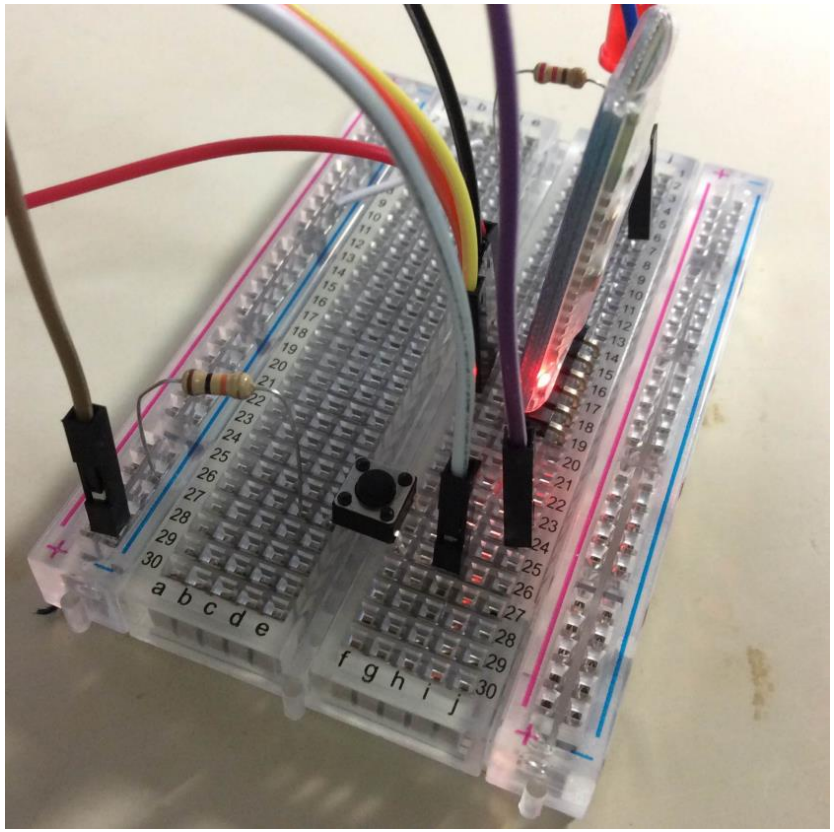
미션 1 : LED 제어

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 LED를 OFF



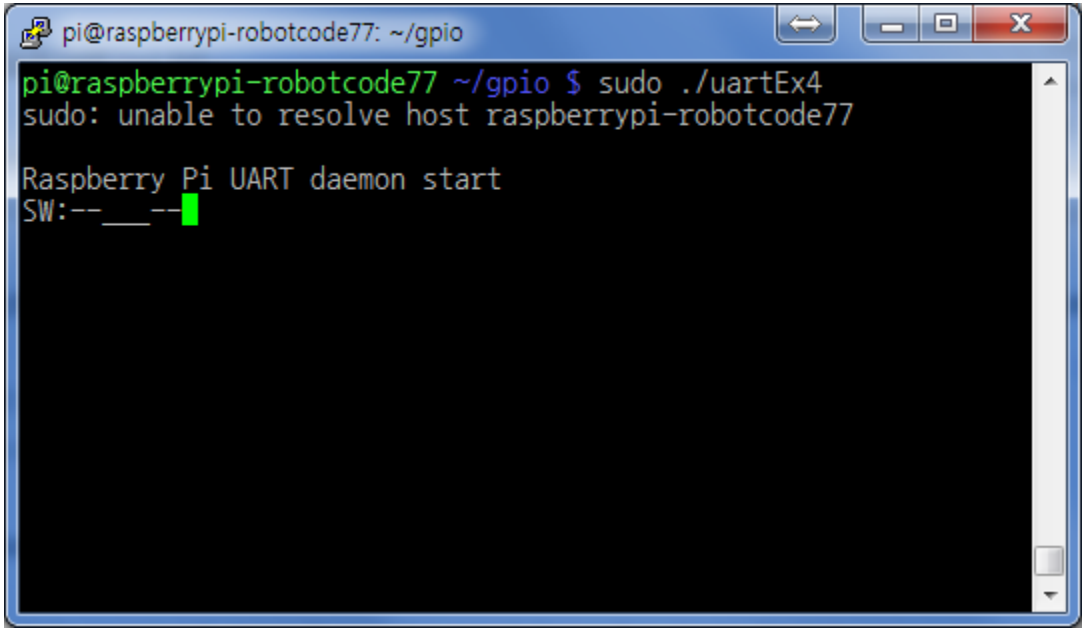
미션 2 : Swtich 모니터링

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 Swtich 상태 모니터링



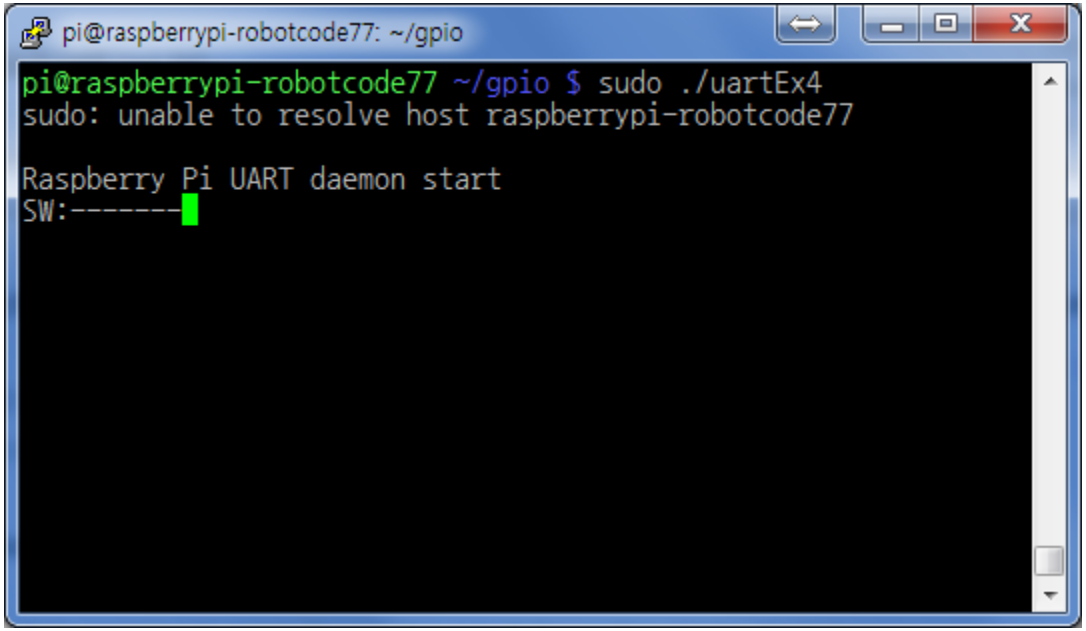
미션 2 : Swtich 모니터링

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 Swtich 상태 모니터링



미션 2 : Swtich 모니터링

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 Swtich 상태 모니터링



Bluetooth Dongle

로봇SW 교육원

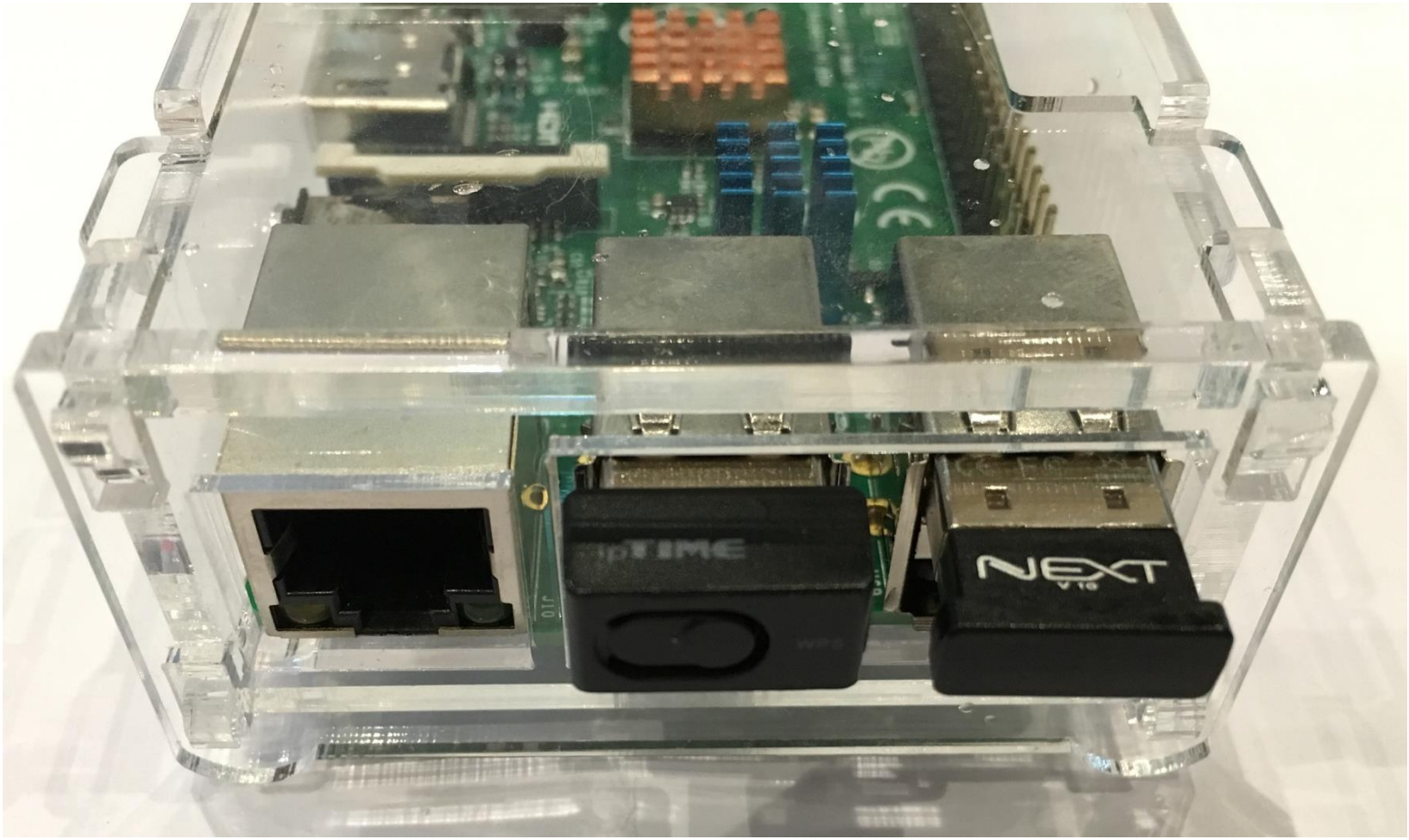
최상훈(shchoi82@gmail.com)

Bluetooth Dongle

- Bluetooth Dongle
  - Bluetooth CSR 4.0 Harmony

실습1 : 블루투스 동글 연결

- 블루투스 동글 연결

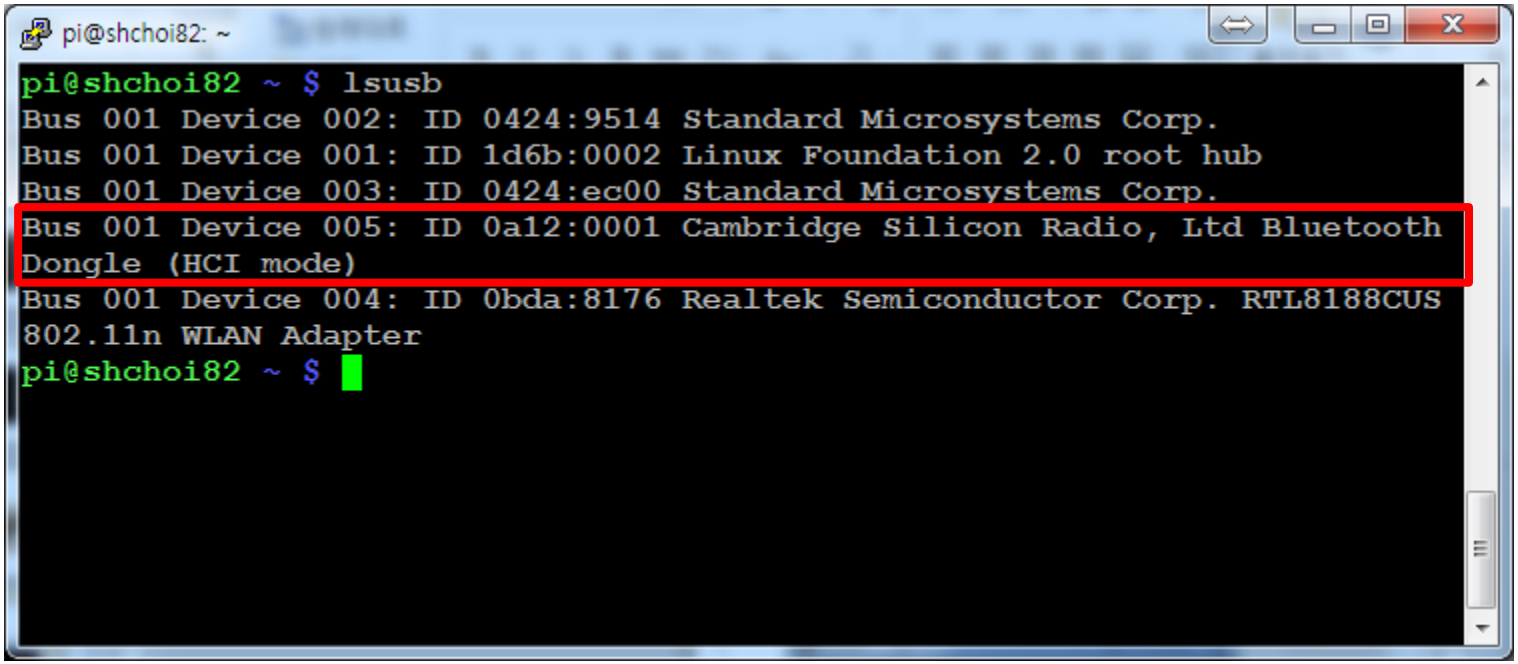


실습2-1 : 블루투스 동글 설정

- 블루투스 동글 연결 확인

```
$ lsusb
```

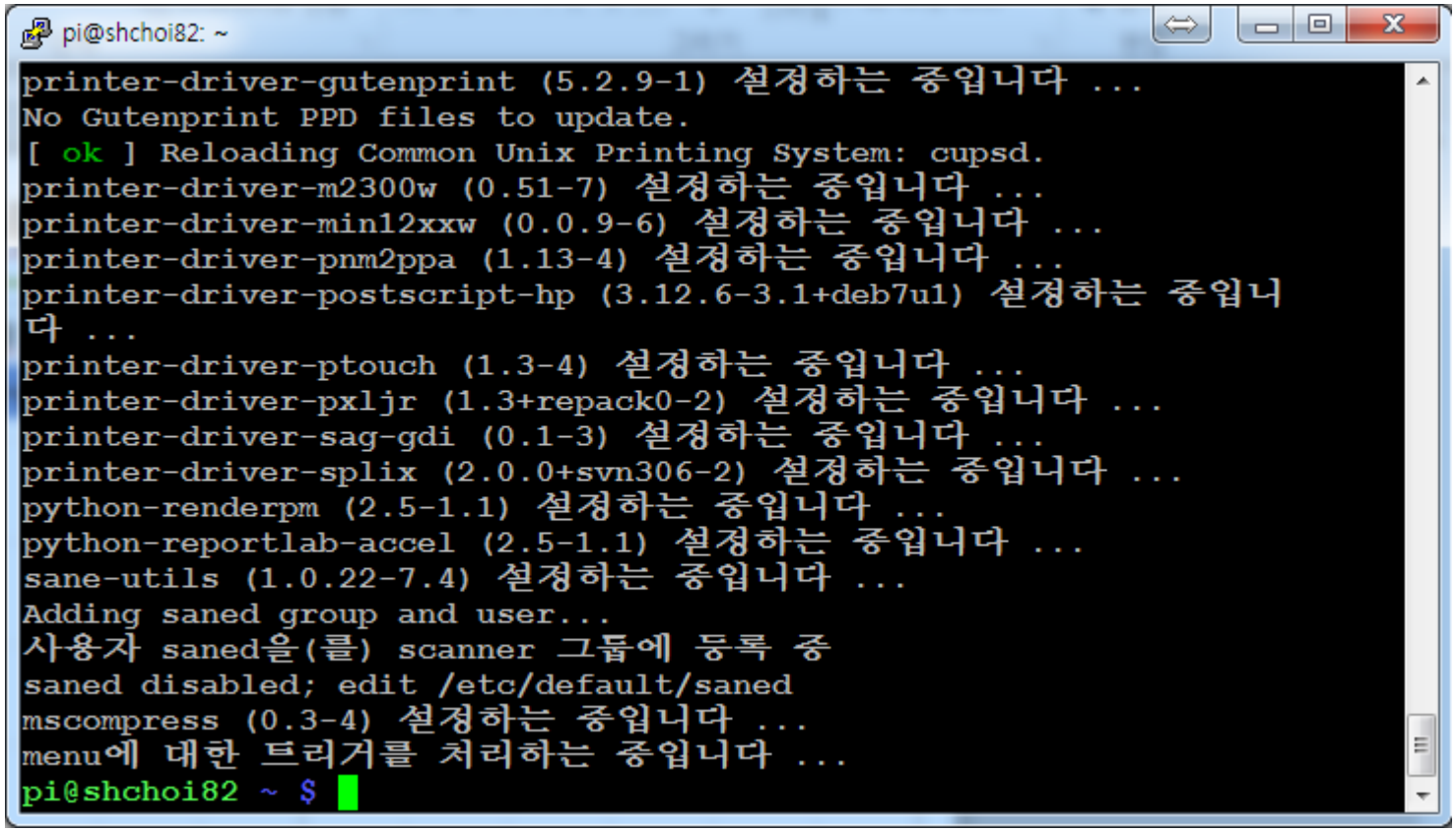
Bus 001 Device 005: ID 0a12:0001 Cambridge Silicon Radio, Ltd Bluetooth Dongle (HCI mode)



실습2-2 : 블루투스 동글 설정

- bluez bluez-util mincom **설치**

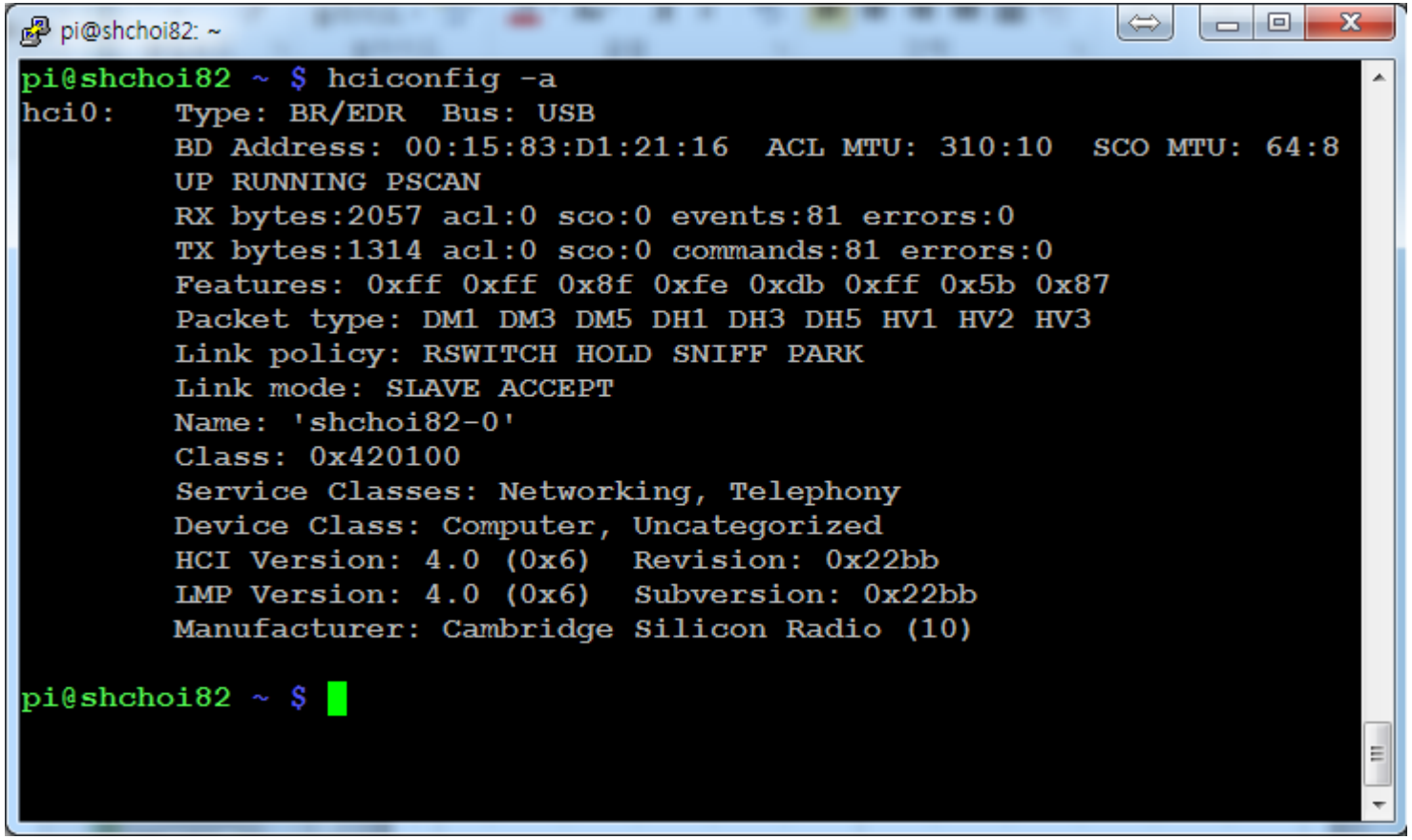
```
$ sudo apt-get install bluez bluez-utils minicom
```



실습2-3 : 블루투스 동글 설정

- 블루투스 장치 정보 확인

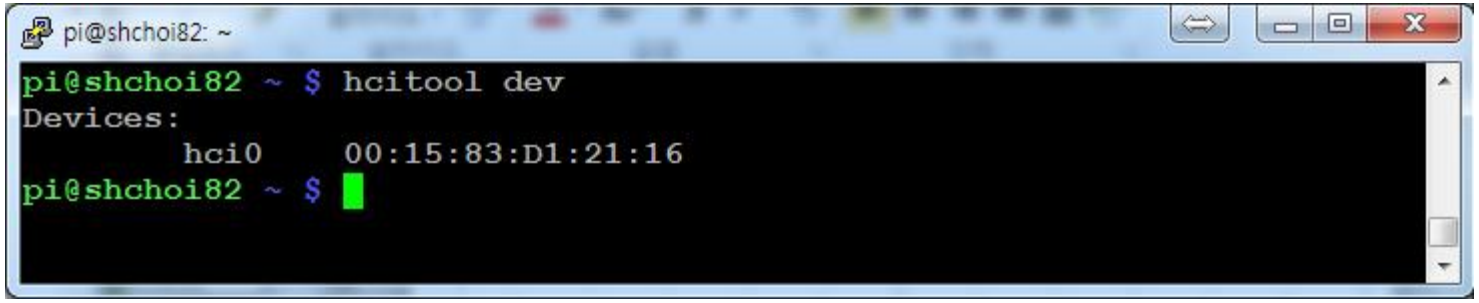
```
$ hciconfig -a
```



실습2-4 : 블루투스 동글 설정

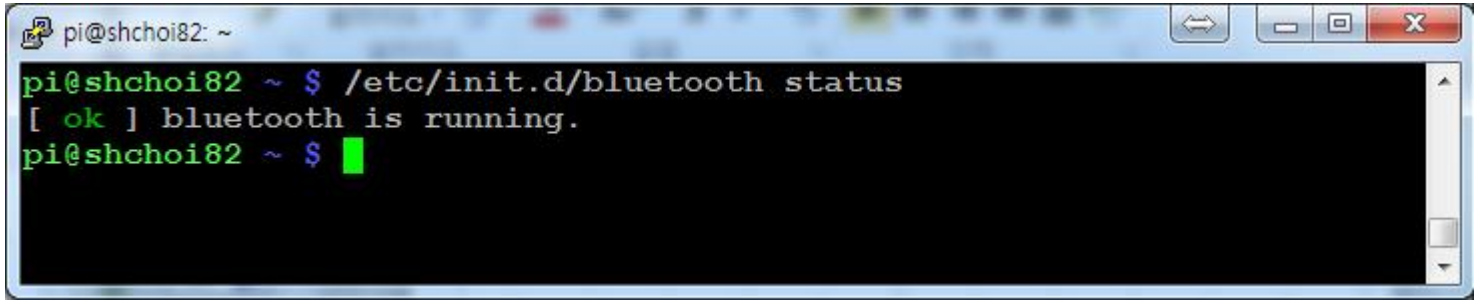
- 블루투스 MAC 주소 확인

```
$ hcitool dev
```



- 블루투스 동작 확인

```
$ /etc/init.d/bluetooth status
```

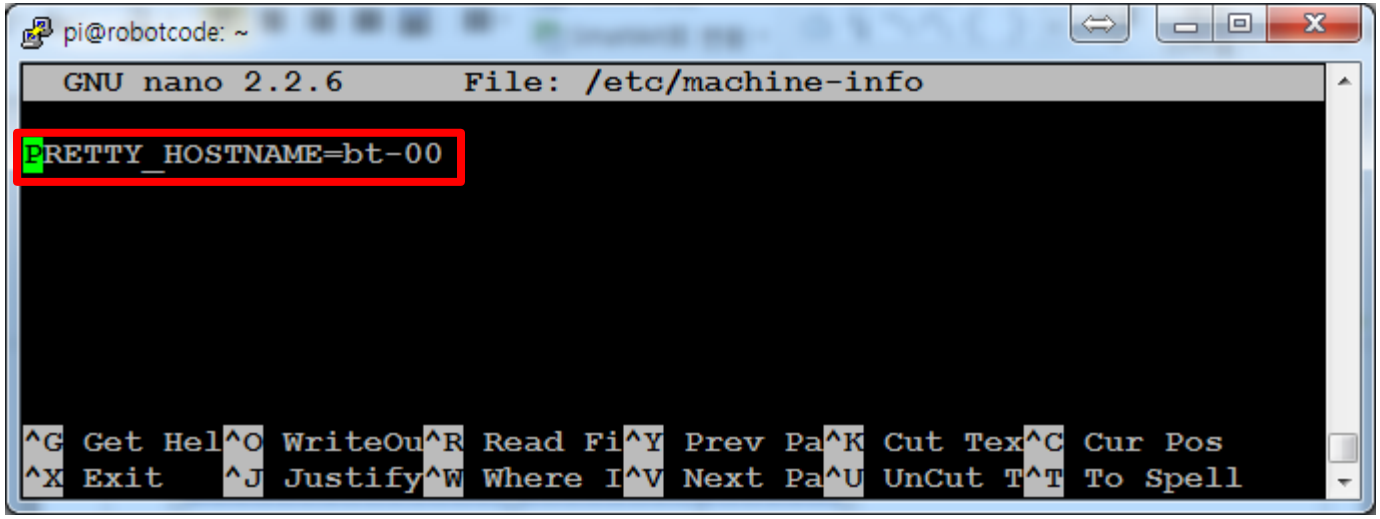


실습2-5 : 블루투스 동글 설정

- Name 변경

```
$ sudo nano /etc/machine-info
```

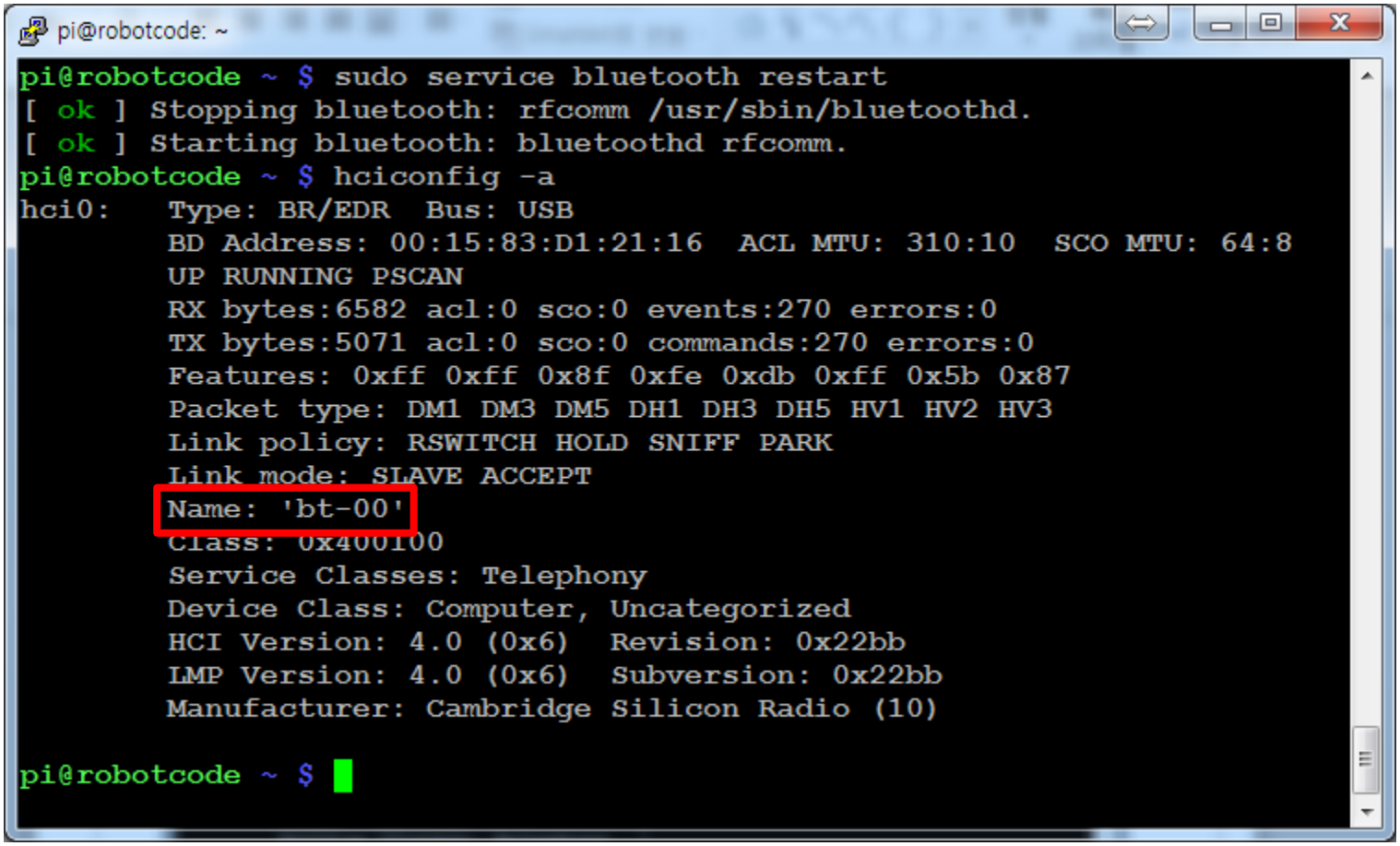
PRETTY_HOSTNAME=bluetooth 이름



실습2-6 : 블루투스 동글 설정

- Name 변경

```
$ sudo service bluetooth restart
```

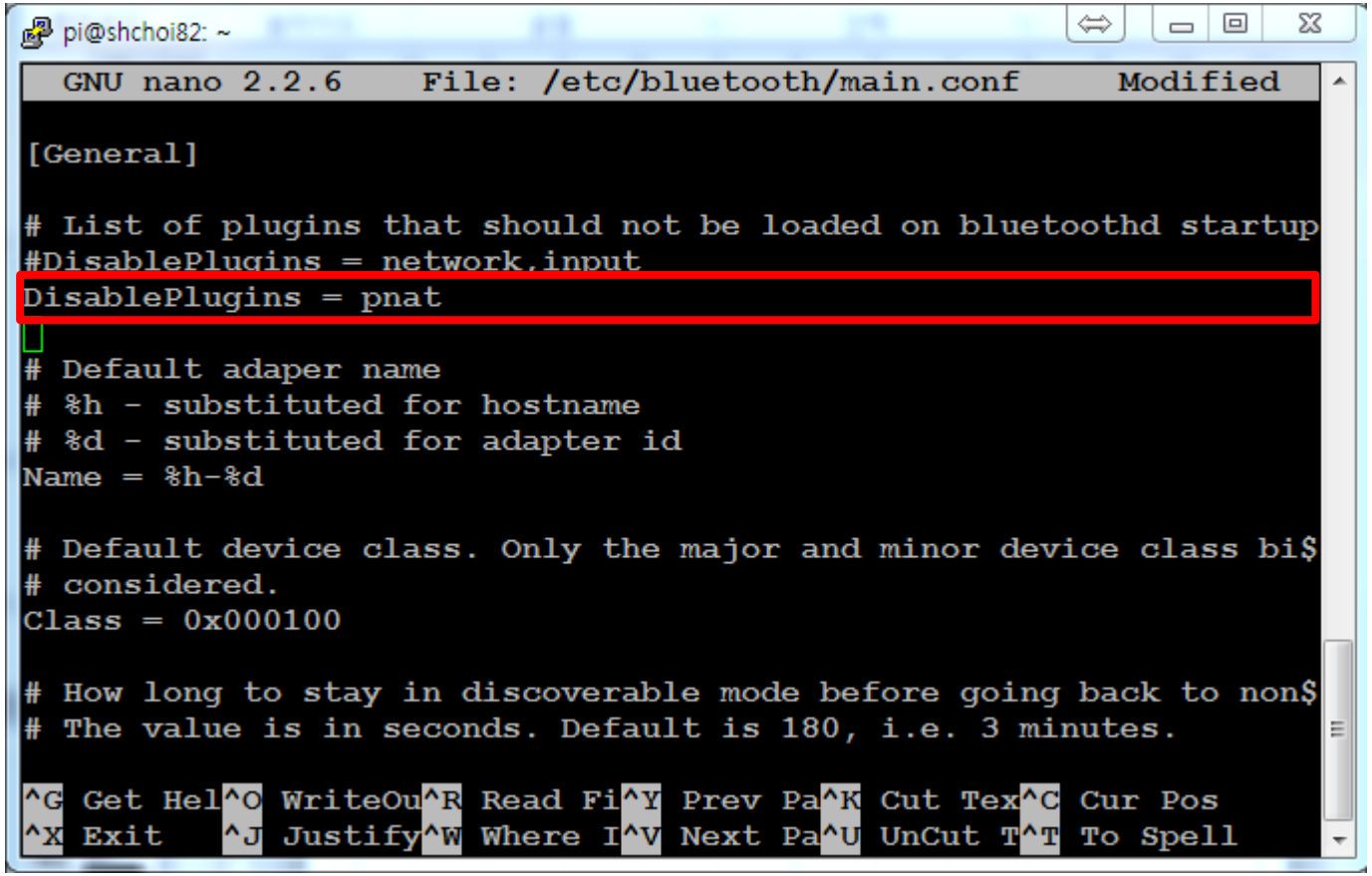


실습2-7 : 블루투스 동글 설정

- Bluez PNAT **플러그인 비활성화**

```
$ sudo nano /etc/bluetooth/main.conf
```

DisablePlugins = pnat



실습2-8 : 블루투스 동글 설정

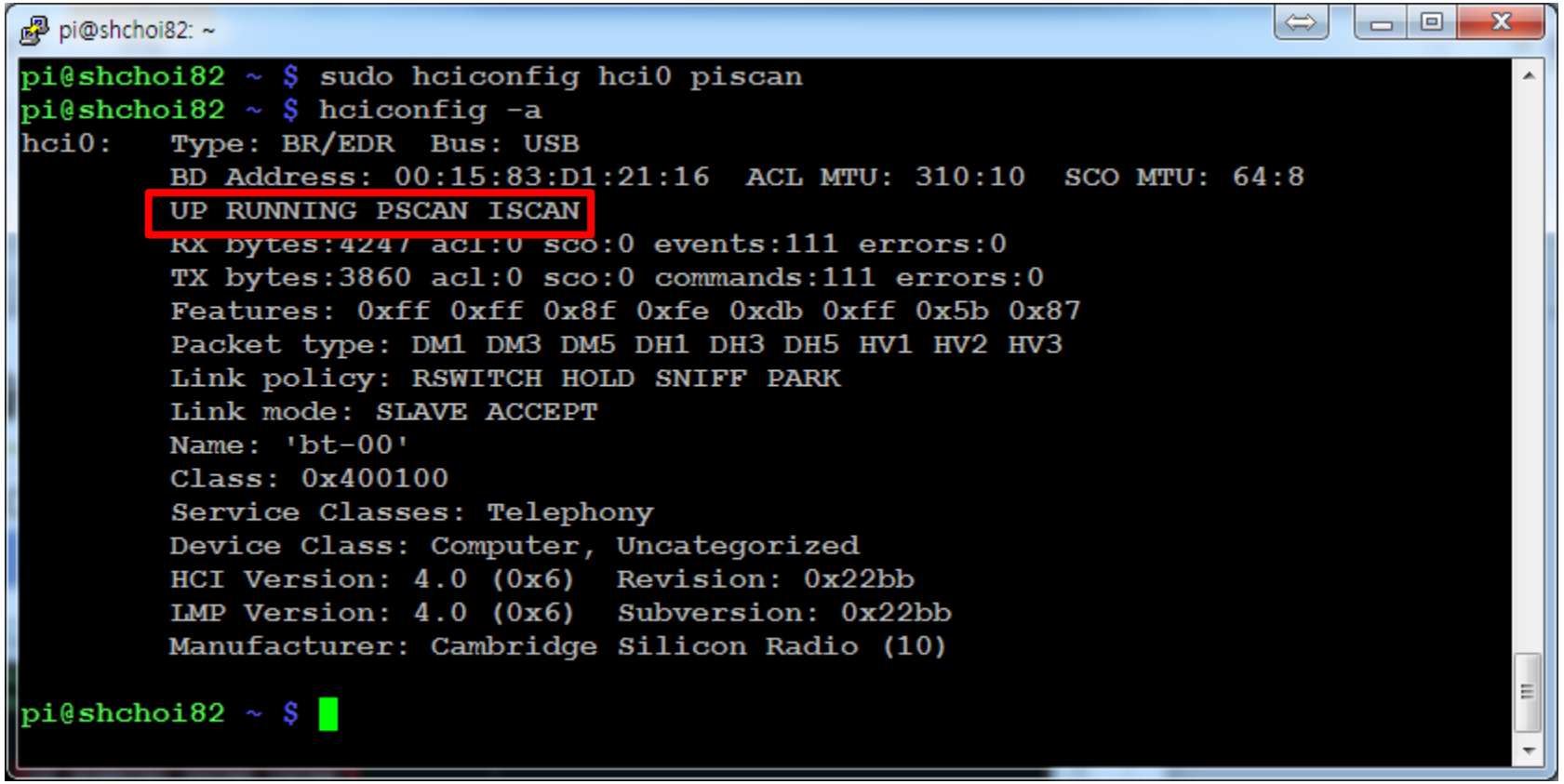
- 재부팅

```
$ sudo reboot
```

실습3-1 : 블루투스 등록

- PSCAN ISCAN **활성화**

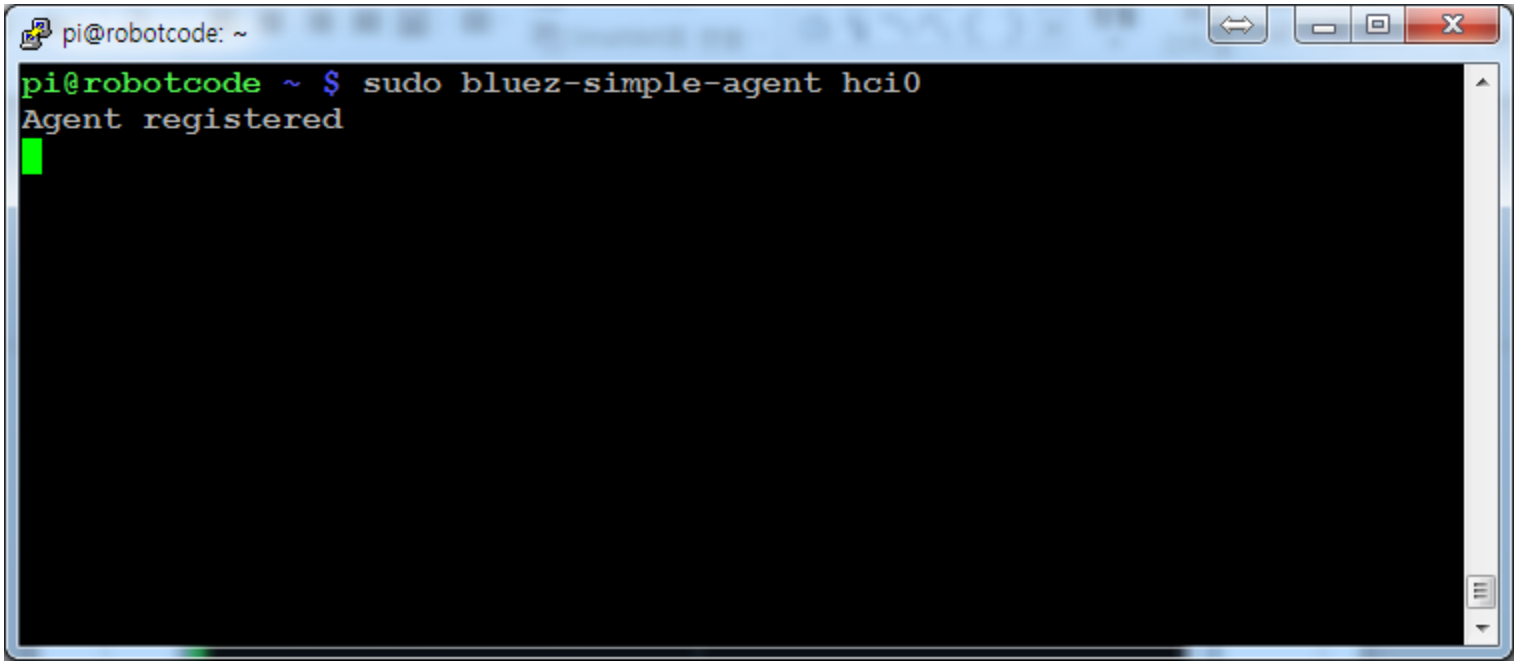
```
$ sudo hciconfig hci0 piscan
```



실습3-2 : 블루투스 등록

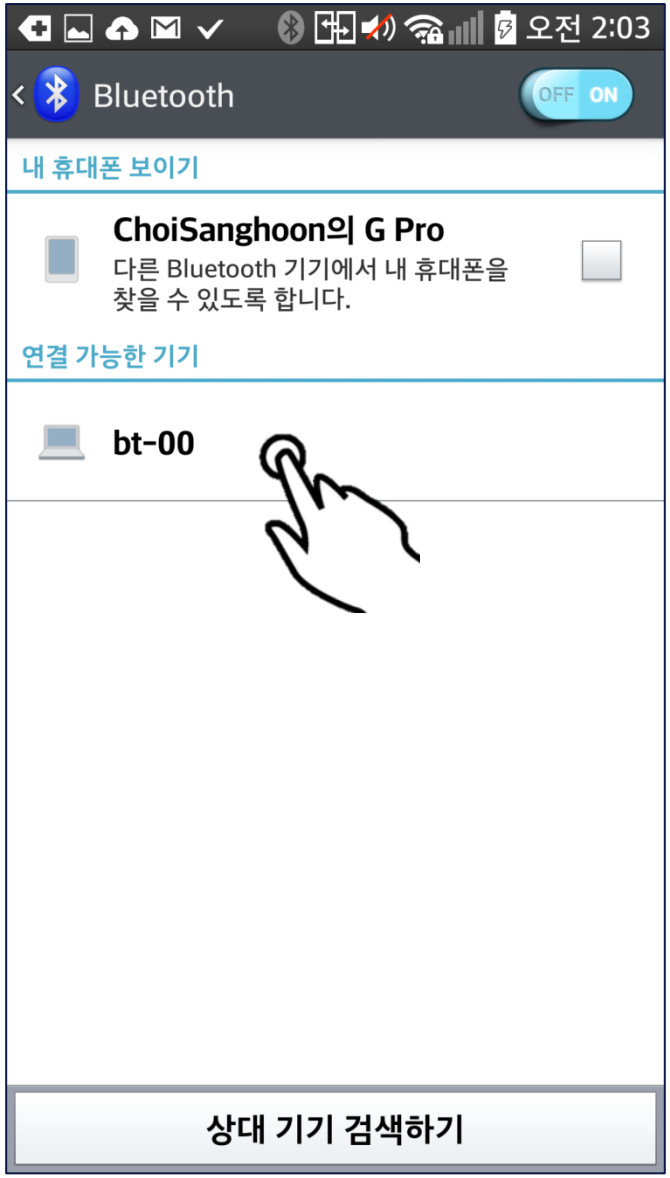
- bluez-simple-agent **실행**

```
$ sudo bluez-simple-agent hci0
```



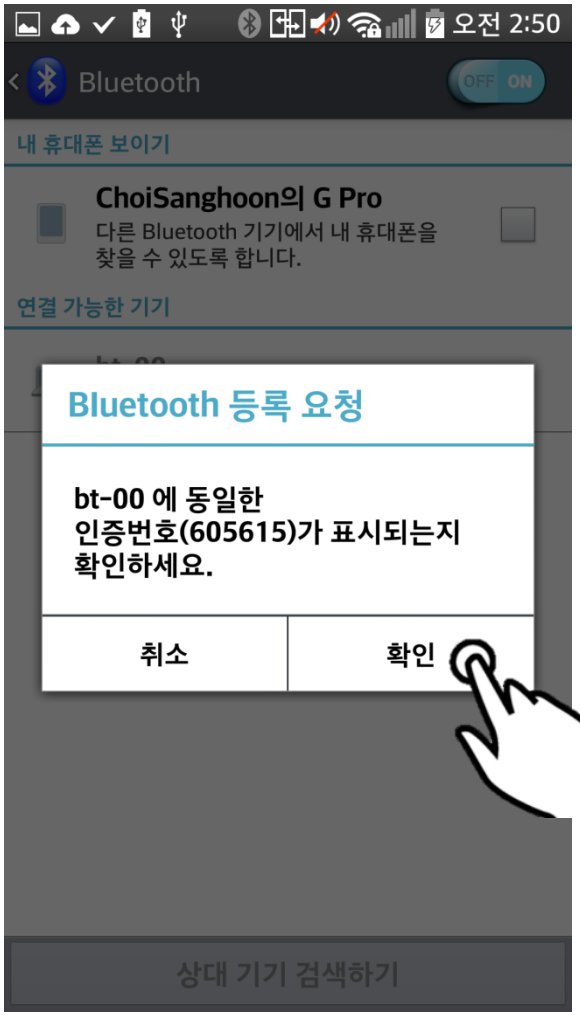
실습3-3 : 블루투스 등록

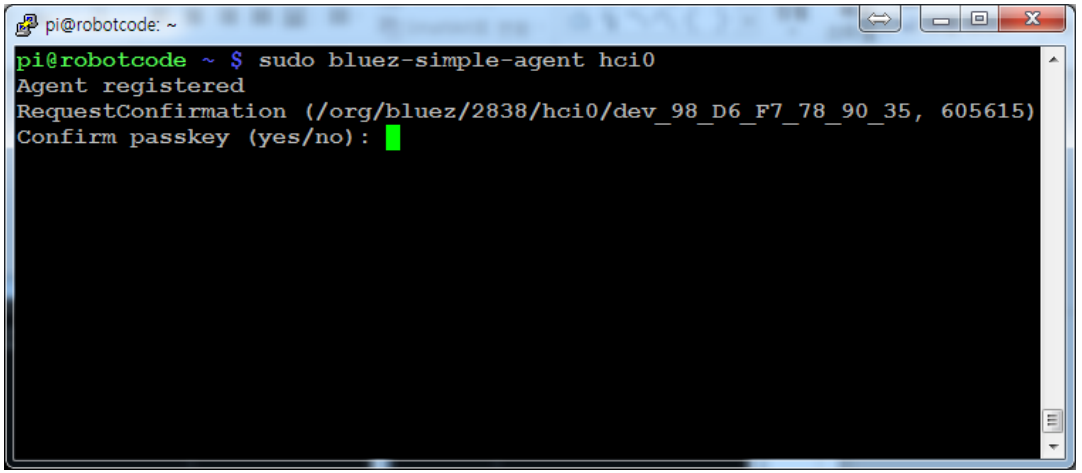
- 안드로이드
  - 블루투스 기기 검색

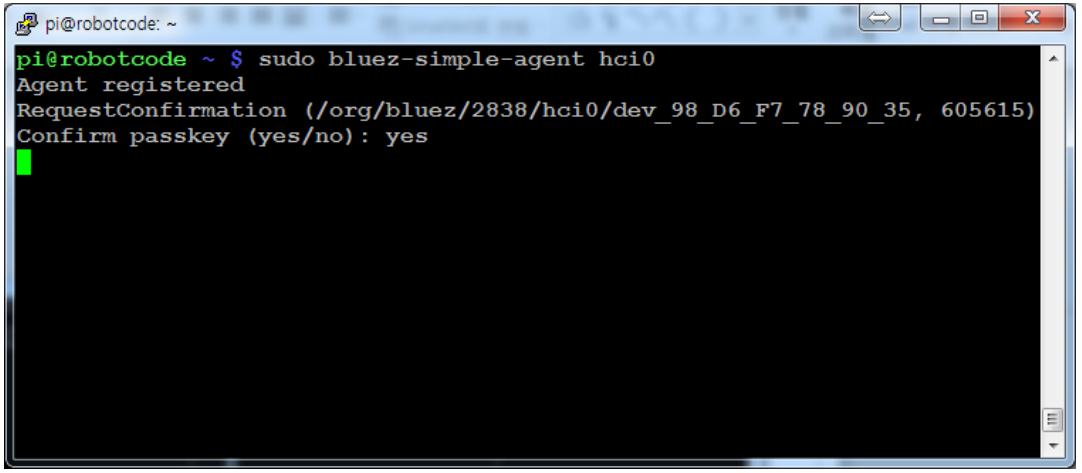


실습3-4 : 블루투스 등록

- 안드로이드
  - 등록

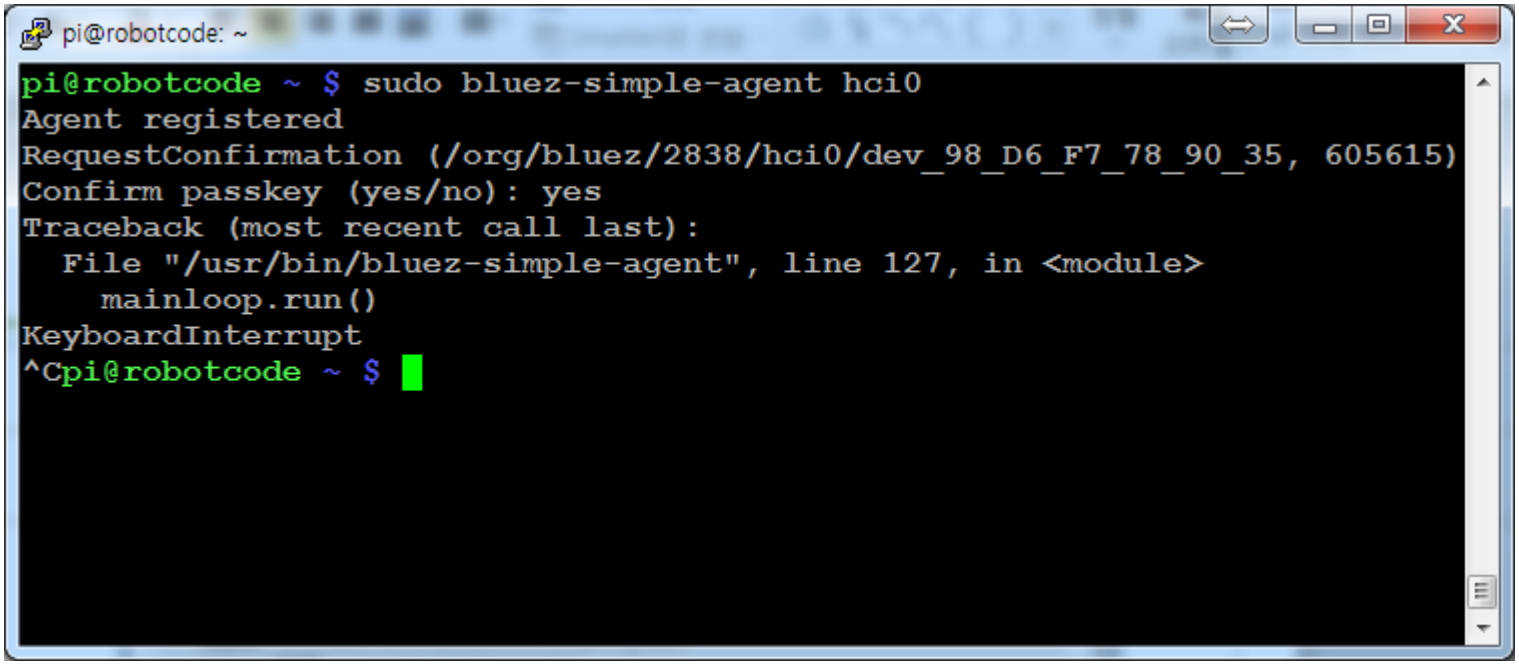






실습3-5 : 블루투스 등록

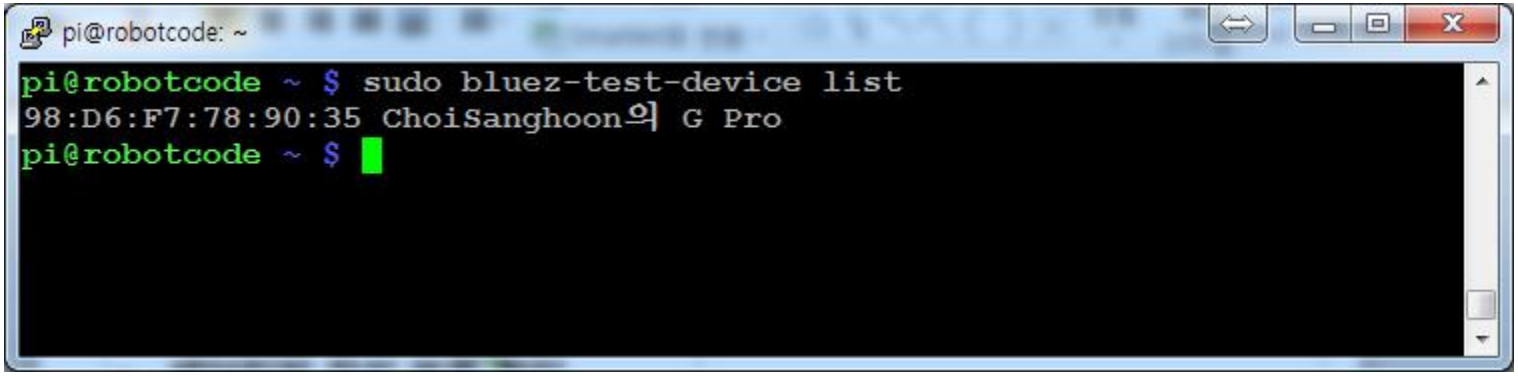
- Ctrl + c **빠져나옴**



실습3-6 : 블루투스 등록

- 등록된 장치 목록 확인

```
$ sudo bluez-test-device list
```



- 참고
  - 한번 등록되면 재부팅 후에도 정보가 남아있음
  - 등록된 장치 삭제

```
$ sudo bluez-test-device remove [MAC주소]
```

실습4-1 : 블루투스 연결

- PSCAN ISCAN **활성화**

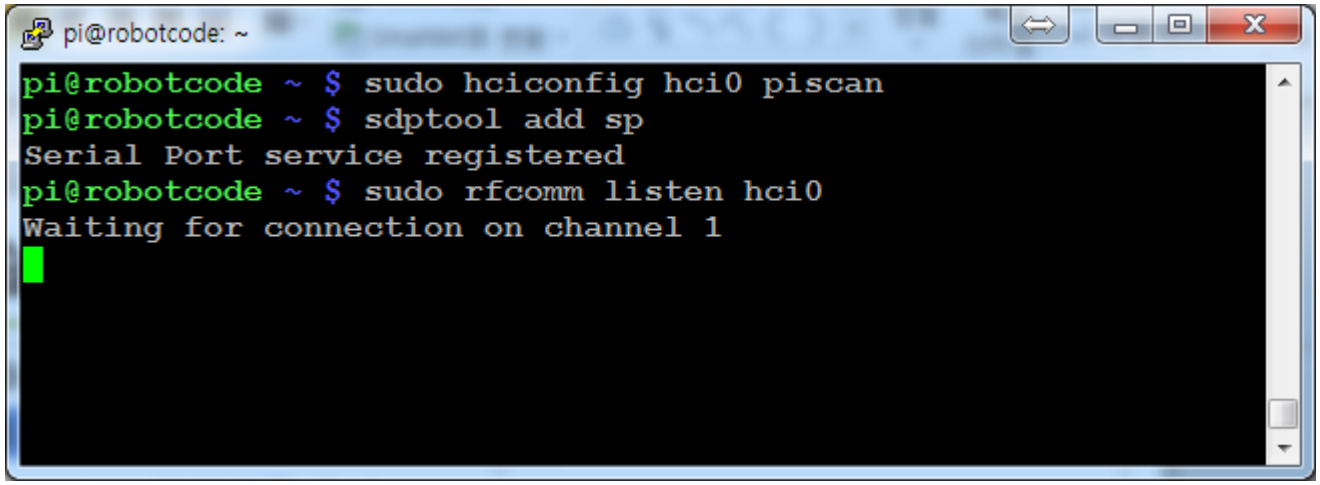
```
$ sudo hciconfig hci0 piscan
```

- Serial Port **프로토콜 추가**

```
$ sdptool add sp
```

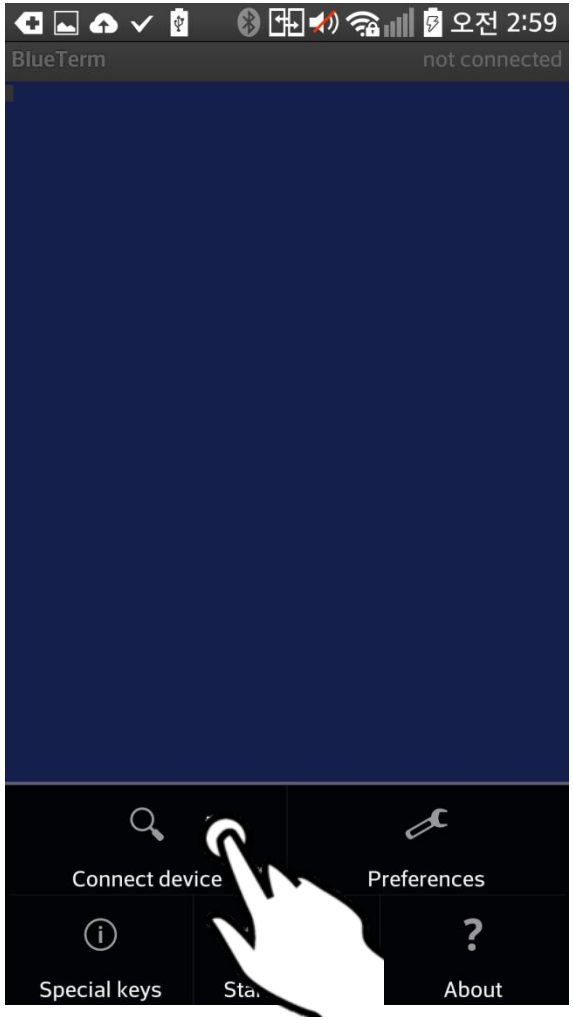
- rfcomm **리스닝 시작**

```
$ sudo rfcomm listen hci0
```



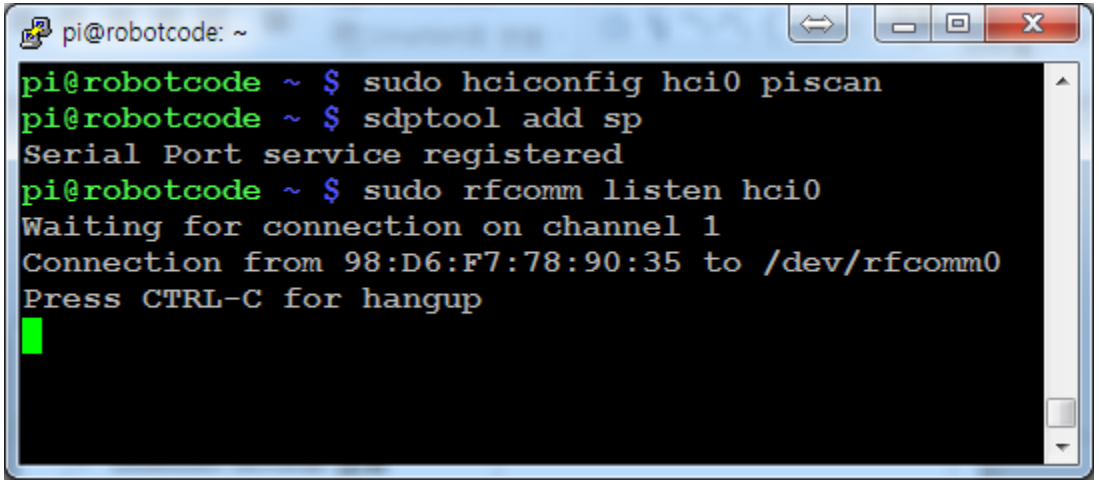
실습4-2 : 블루투스 연결

- Bluetooth terminal **연결**



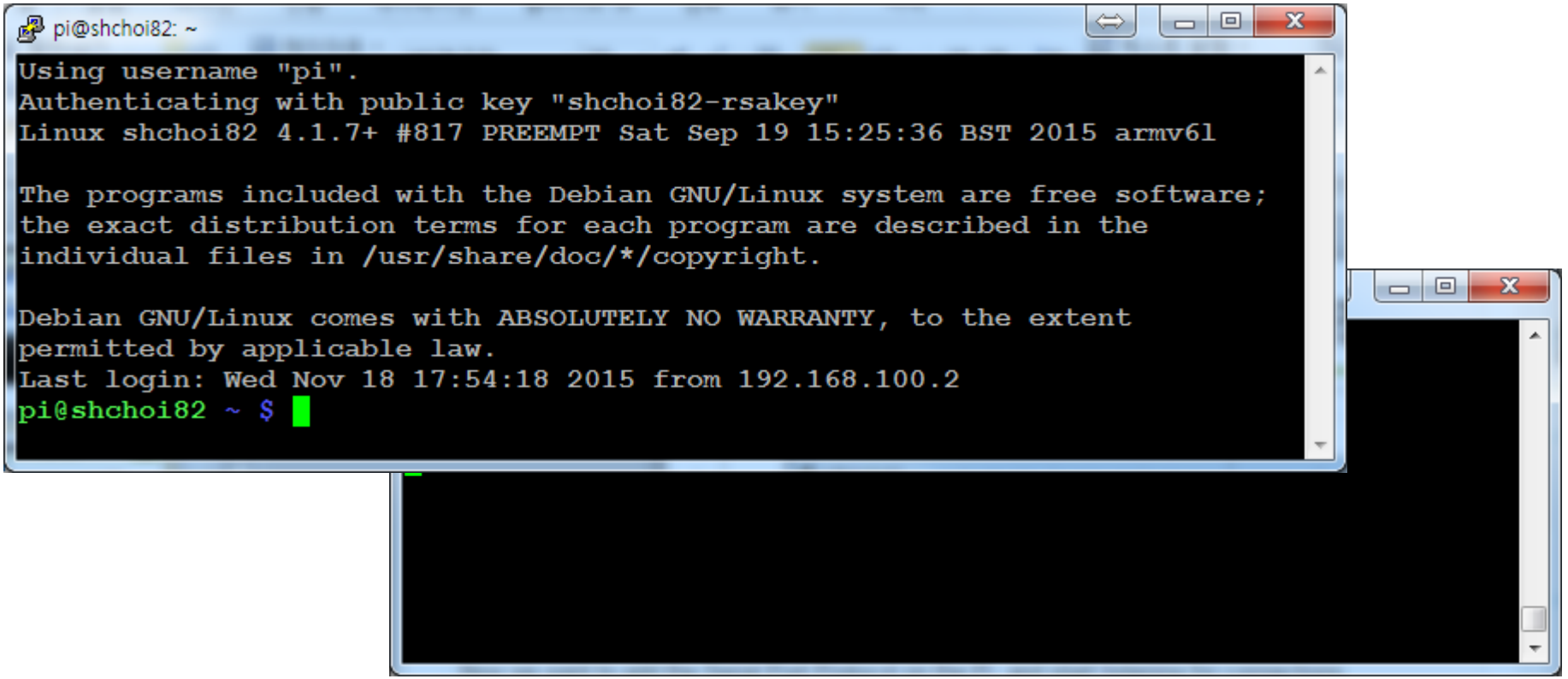
실습4-3 : 블루투스 연결

- Bluetooth terminal **연결**



실습4-4 : 블루투스 연결

- 새로운 터미널 오픈



실습4-5 : 안드로이드 -> Pi

- RaspberryPi에서 안드로이드로 데이터 전송
  - uartEx3.c

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
#include <wiringPi.h>
#include <wiringSerial.h>

#define EXIT_SUCC 0
#define EXIT_FAIL 1

int main()
{
    int fd;
    int data;
    int i;

    setbuf(stdout, NULL);

    if(wiringPiSetupGpio() == -1){
        fprintf(stdout, "Unable to start wiringPi : %s\n", strerror(errno));
        return EXIT_FAIL;
    }

    // 시리얼 통신 초기화 및 속도 설정
    for(i = 10 ; i > 0 ; i--){
        if((fd = serialOpen("/dev/rfcomm0", 115200)) < 0)
        {
            printf("블루투스 연결을 기다립니다...[%d] status : %s\r", i, strerror(errno));
            sleep(1);
        }
    }
```

실습4-6 : 안드로이드 -> Pi


```
    if( fd < 0 ){
        fprintf(stderr, "\n%s\n", strerror(errno));
        return EXIT_FAIL;
    }
    printf("\nRaspberry Pi UART daemon start\n");
    serialPuts(fd, "Here I'm the Raspberry Pi.\r\n");   // to serial
    serialPuts(fd, "Write a message.\r\n");             // to serial

    while(1){
        if(serialDataAvail(fd) == -1){
            fprintf(stderr, "%s\n", strerror(errno));
            return EXIT_FAIL;
        }
        data = serialGetchar(fd);                 // from serial
        if(data == -1)
            continue;
        printf("%c",data);
    }
    return EXIT_SUCC;
}
```

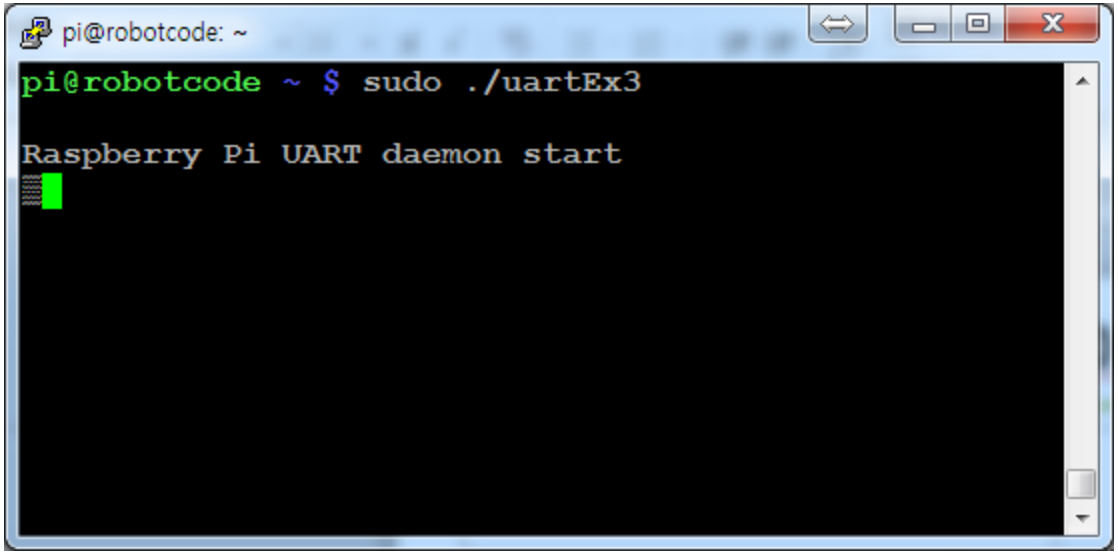
- **컴파일**

```
$ gcc -Wall -W -lwiringPi uartEx3.c -o uartEx3
```

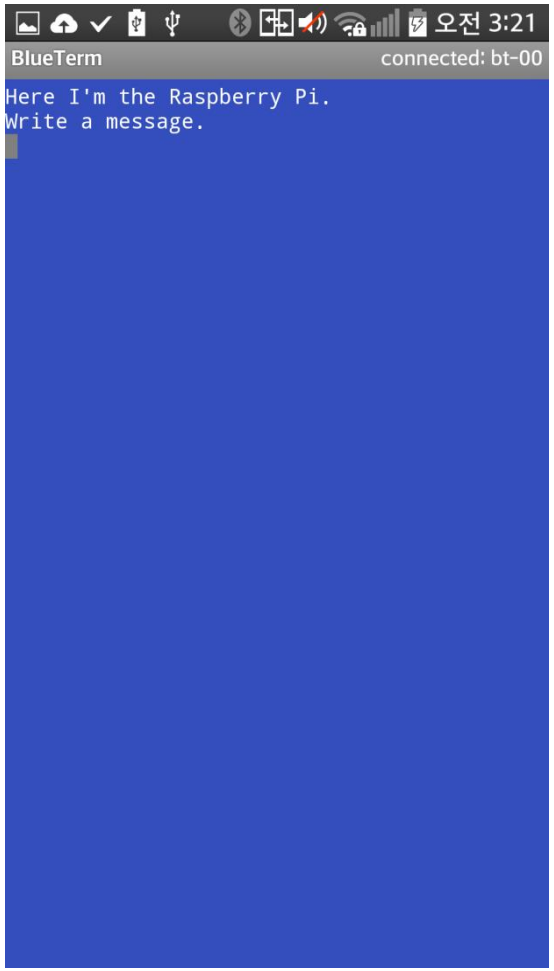
- **실행**

```
$ sudo ./uartEx3
```

실습4-7 : 안드로이드 -> Pi

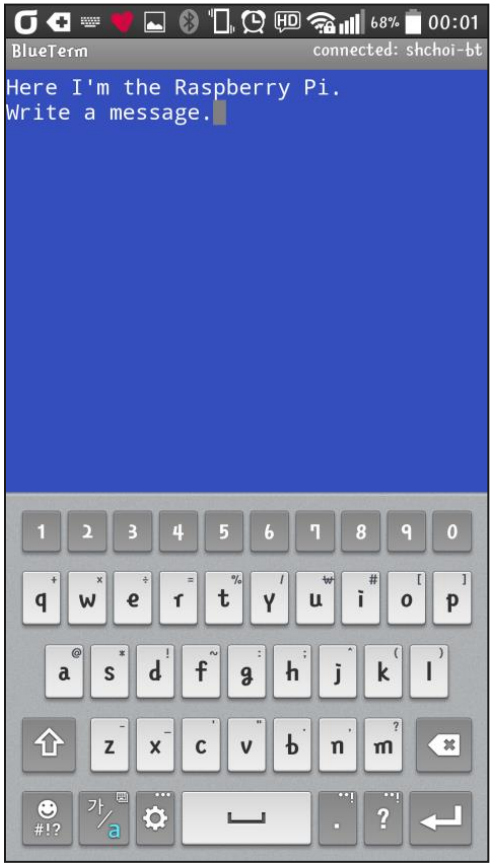


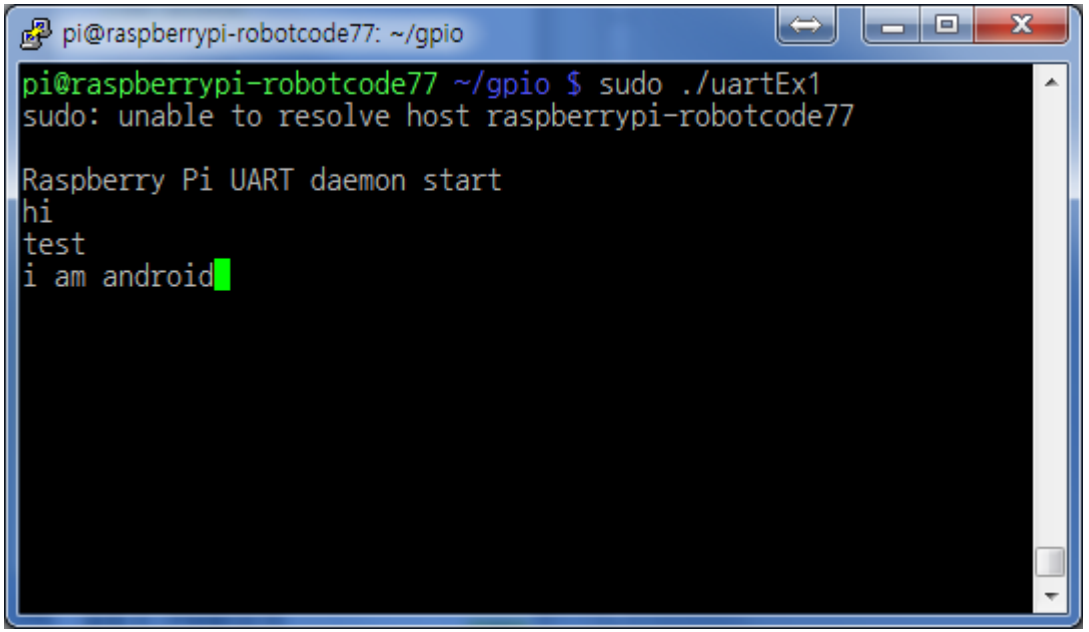
< Raspberry Pi >



실습4-8 : 안드로이드 -> Pi

- 통신 테스트
  - 안드로이드의 터미널에 텍스트를 입력하면
  - RaspberryPi의 터미널창 그대로 출력됨





< Raspberry Pi >

실습 5-1 : Pi -> 안드로이드

- RaspberryPi에서 안드로이드로 데이터 전송
  - uartEx4.c

```
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include <wiringPi.h>
#include <wiringSerial.h>

#define EXIT_SUCC 0
#define EXIT_FAIL 1

int main()
{
    int fd;
    int data;

    setbuf(stdout, NULL);
    setbuf(stdin, NULL);

    if(wiringPiSetupGpio() == -1){
        fprintf(stdout, "Unable to start wiringPi : %s\n", strerror(errno));
        return EXIT_FAIL;
    }

    // 시리얼 통신 초기화 및 속도 설정
    if((fd = serialOpen("/dev/ttyAMA0", 115200)) < 0)
    {
        fprintf(stderr, "Unable to open serial device : %s\n", strerror(errno));
        return EXIT_FAIL;
    }
```

실습 5-2 : Pi -> 안드로이드


```
    printf("\nRaspberry Pi UART daemon start\n");
    serialPuts(fd, "Here I'm the Raspberry Pi.\r\n");   // to serial
    serialPuts(fd, "Write a message.\r\n");             // to serial

    while(1){
        if((data = fgetc(stdin)) == EOF){
            printf("EOF\n");
            break;
        }
        if(data == '\n'){
            serialPutchar(fd, '\r');                    // to serial
        }
        serialPutchar(fd, data);                        // to serial
    }
    return EXIT_SUCC;
}
```

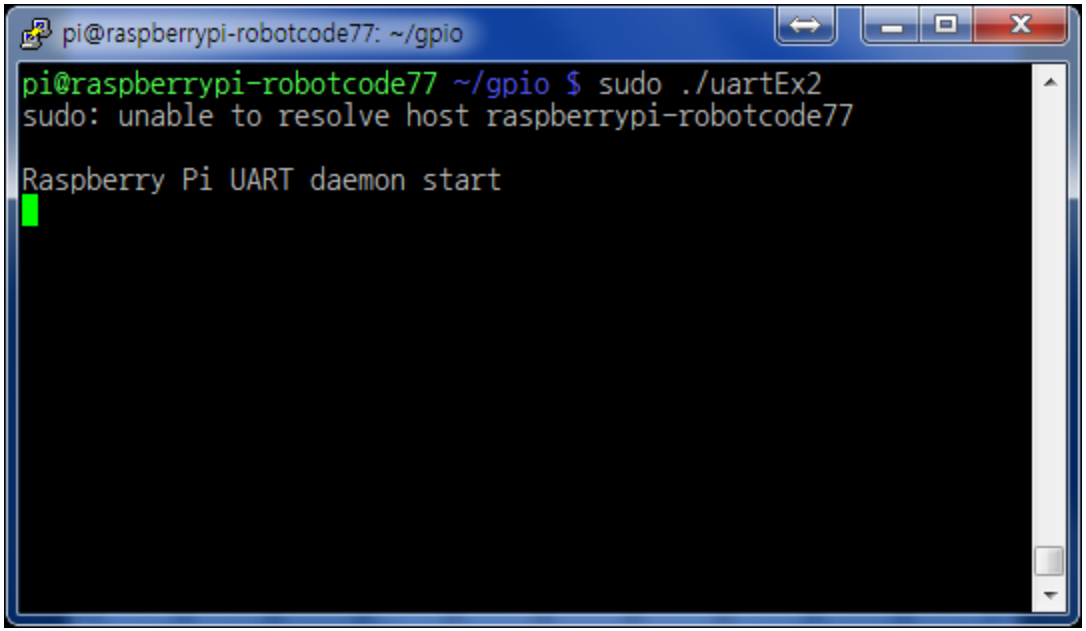
- **컴파일**

```
$ gcc -Wall -W -lwiringPi uartEx4.c -o uartEx4
```

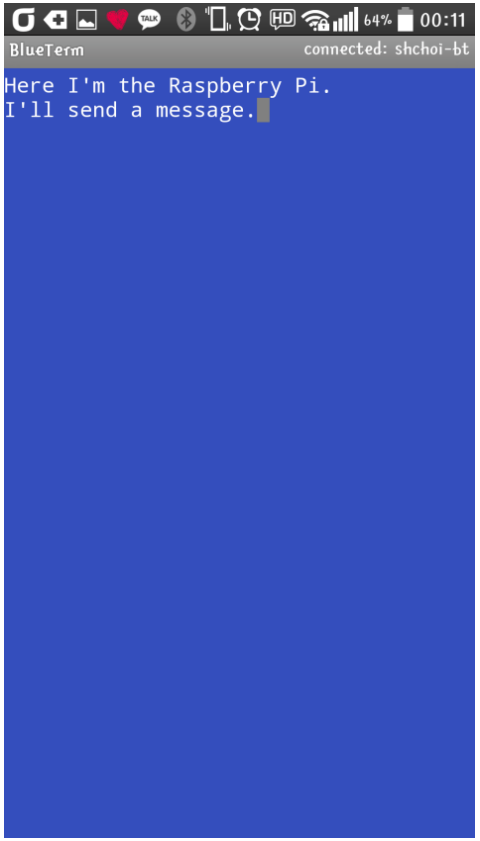
- **실행**

```
$ sudo ./uartEx4
```

실습 5-3 : Pi -> 안드로이드

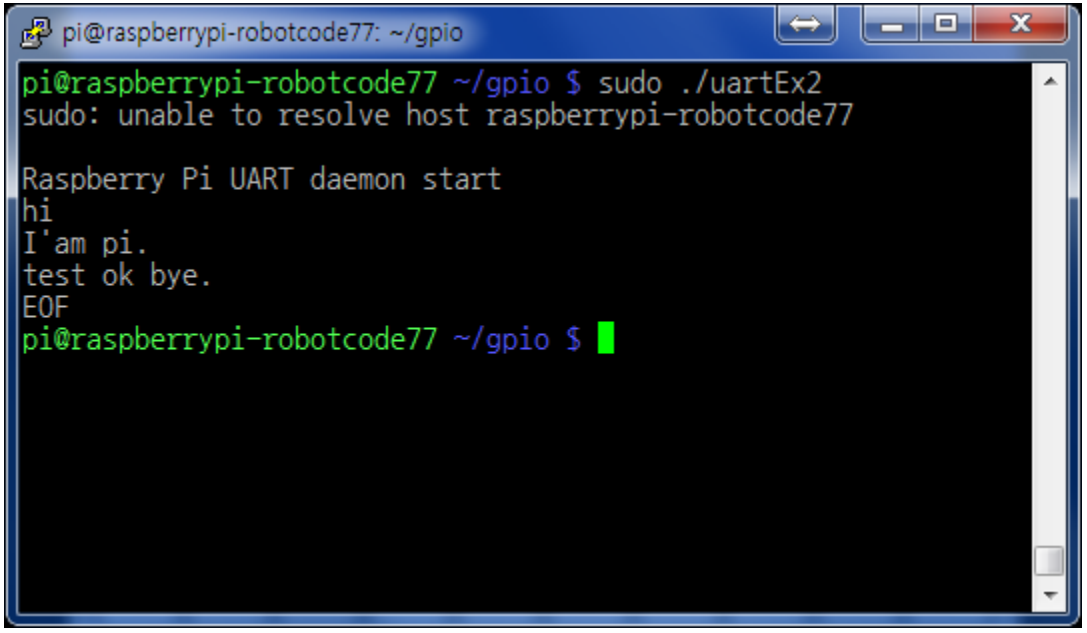


< Raspberry Pi >

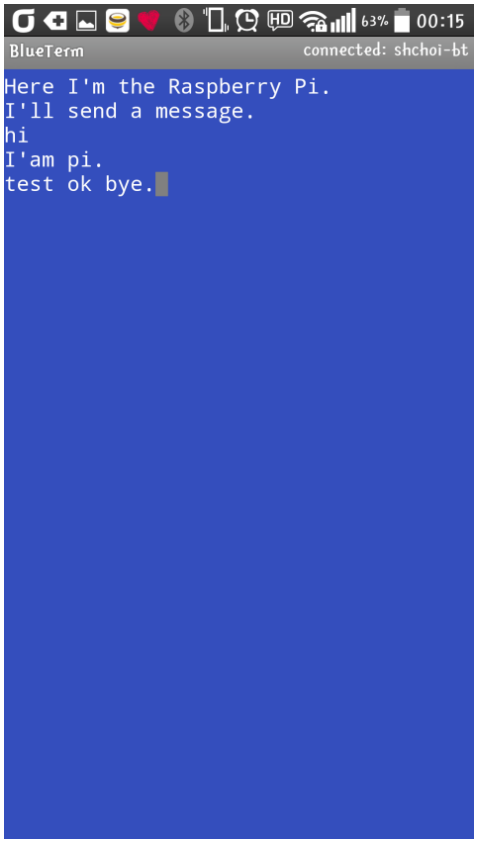


실습 5-4 : Pi -> 안드로이드

- 통신 테스트
  - RaspberryPi의 터미널에 텍스트 입력하면
  - 안드로이드 터미널창에 그대로 출력됨



< Raspberry Pi >



미션 1-1 : LED 제어

- Bluetooth 통신을 통한 Raspberry Pi GPIO 제어하기
  - 메뉴출력

미션 1-2 : LED 제어

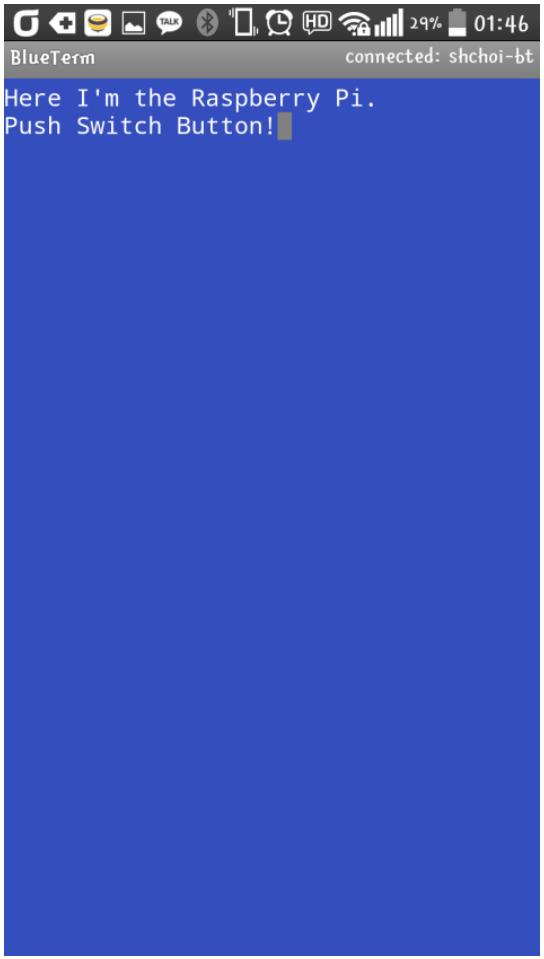
- Raspberry Pi GPIO 제어
  - GPIO에 연결된 LED를 ON

미션 1 : LED 제어

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 LED를 OFF

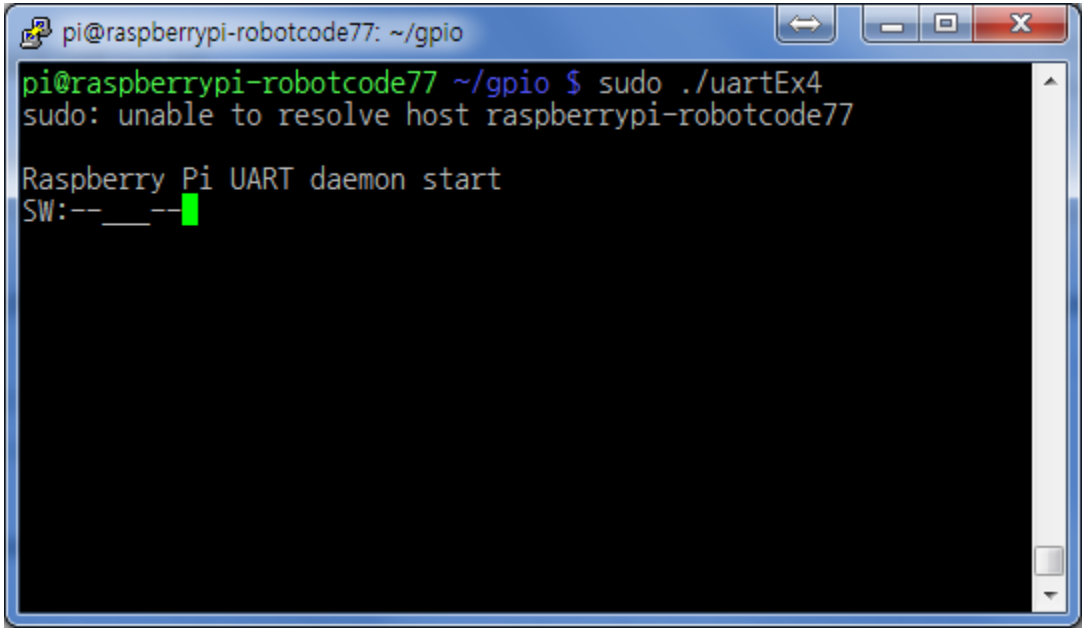
미션 2-1 : Swtich 모니터링

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 Swtich 상태 모니터링



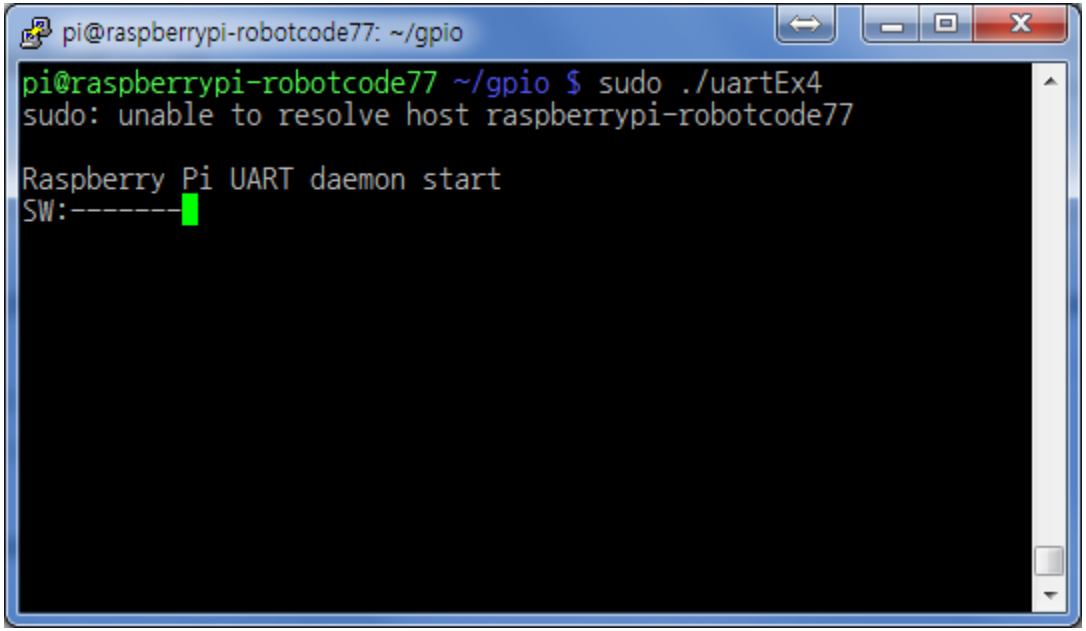
미션 2-2 : Swtich 모니터링

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 Swtich 상태 모니터링



미션 2-3 : Swtich 모니터링

- Raspberry Pi GPIO 제어
  - GPIO에 연결된 Swtich 상태 모니터링



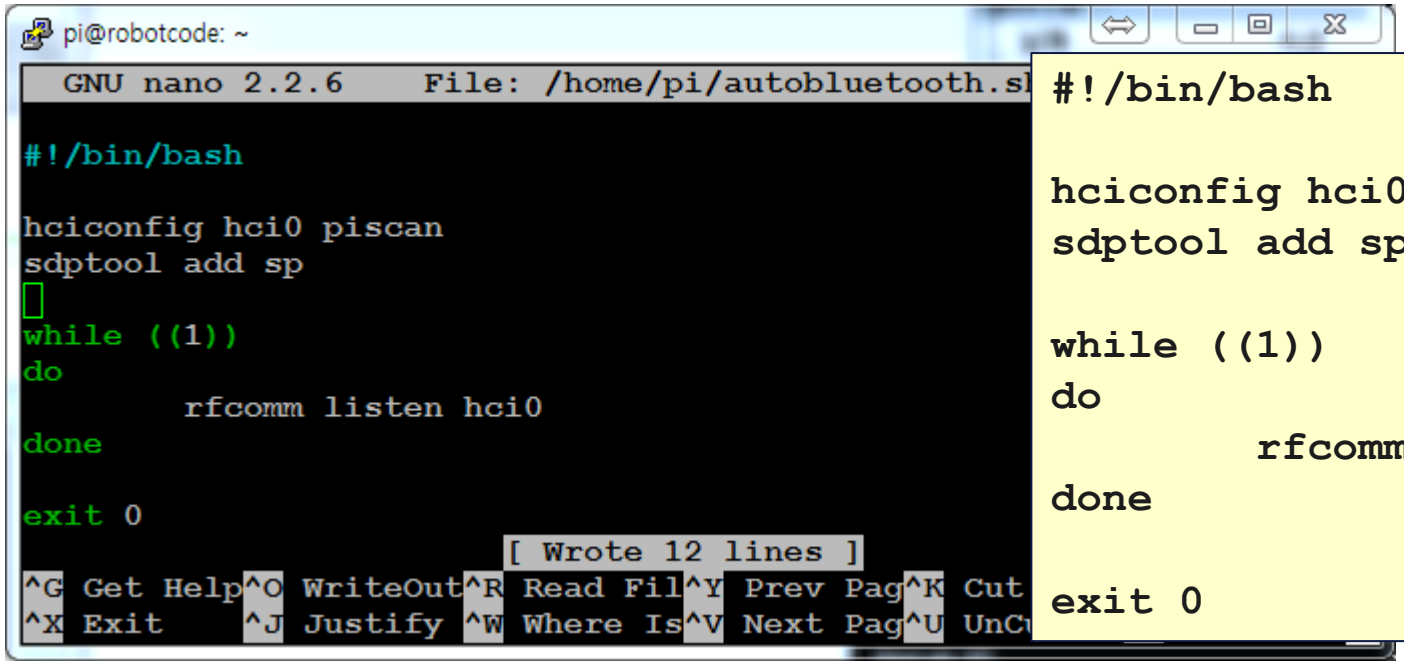
미션 3

- 현재 작업 디렉토리의 파일 목록을 안드로이드 터미널 화면에 출력하시오.

실습6-1 : 부팅시 자동 설정

- 부팅시 자동 설정

```
$ nano /home/pi/autobluetooth.sh
```




```
#!/bin/bash

hciconfig hci0 piscan
sdptool add sp

while ((1))
do
        rfcomm listen hci0
done

exit 0
```

※ 53 slide **참고**

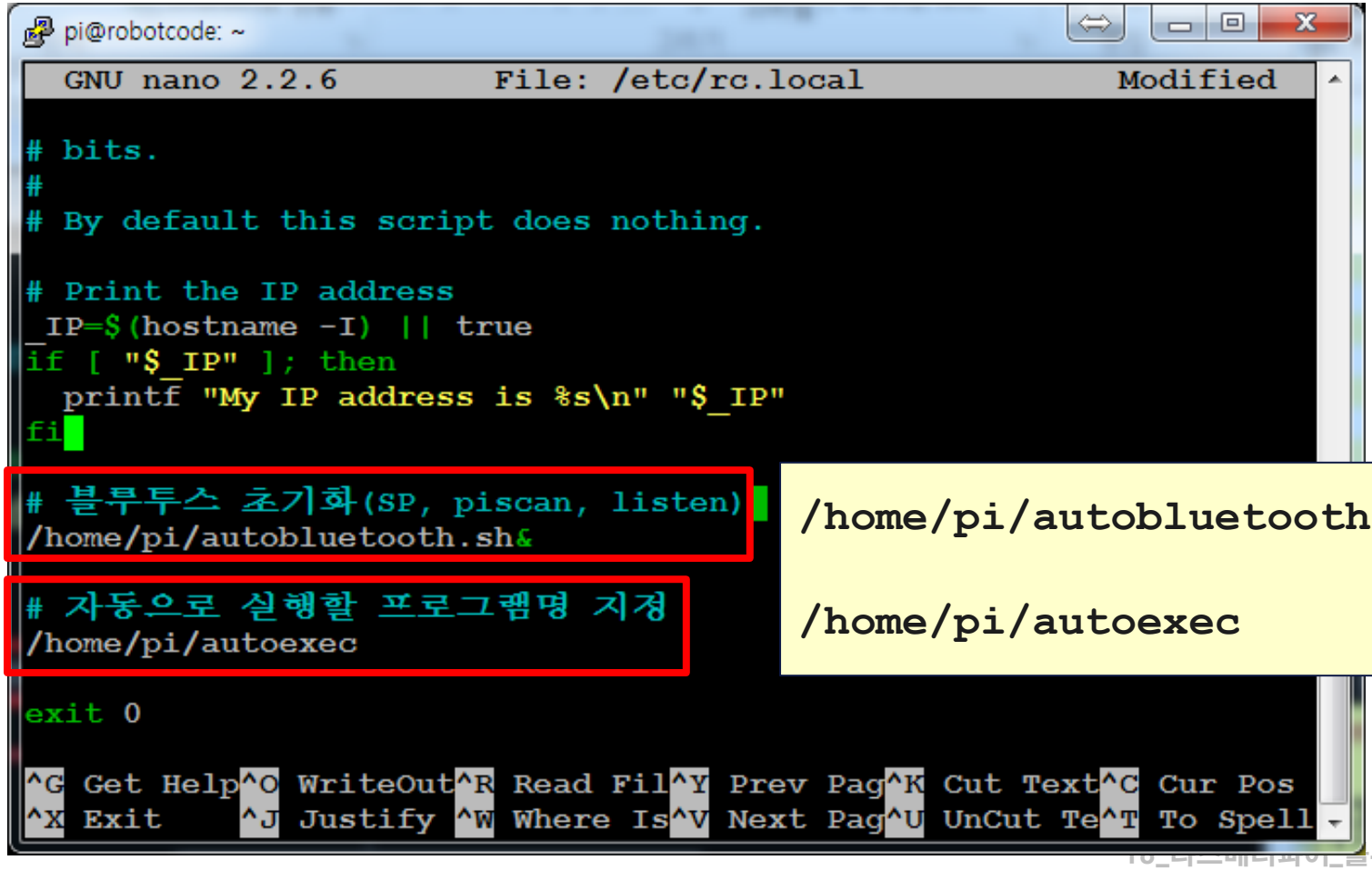
- 실행 권한 설정

```
$ chmod +x /home/pi/autobluetooth.sh
```

실습6-2 : 부팅시 자동 설정

- **부팅시 자동 설정**
  - /etc/rc.local **파일에 추가**

```
$ sudo nano /etc/rc.local
```




```
/home/pi/autobluetooth.sh&

/home/pi/autoexec
```

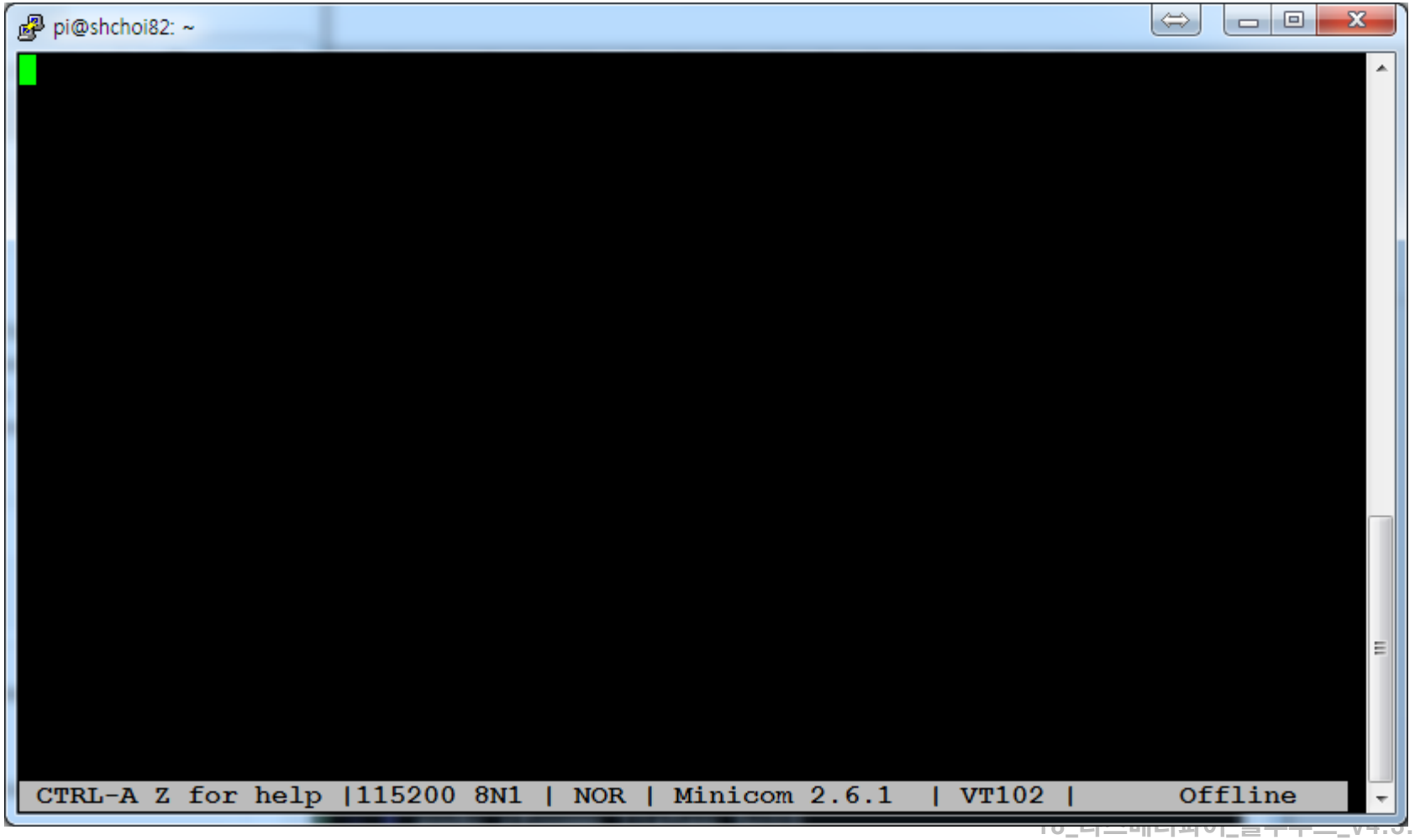
미션4

- 미션1에서 작성한 프로그램이 부팅 시 자동으로 실행될 수 있도록 설정하시오.

참고 : minicom

- **라즈베리파이** minicom **실행**

```
$ minicom -D /dev/rfcomm0
```



참고 : minicom

- **안드로이드** bluetooth terminal **앱에서 키보드 입력**