



라즈베리 파이 블루투스

로봇SW 교육원

최상훈(shchoi82@gmail.com)

학습 목표

2

- 블루투스 모듈 제어
- 블루투스 Dongle 제어

Bluetooth Module

3

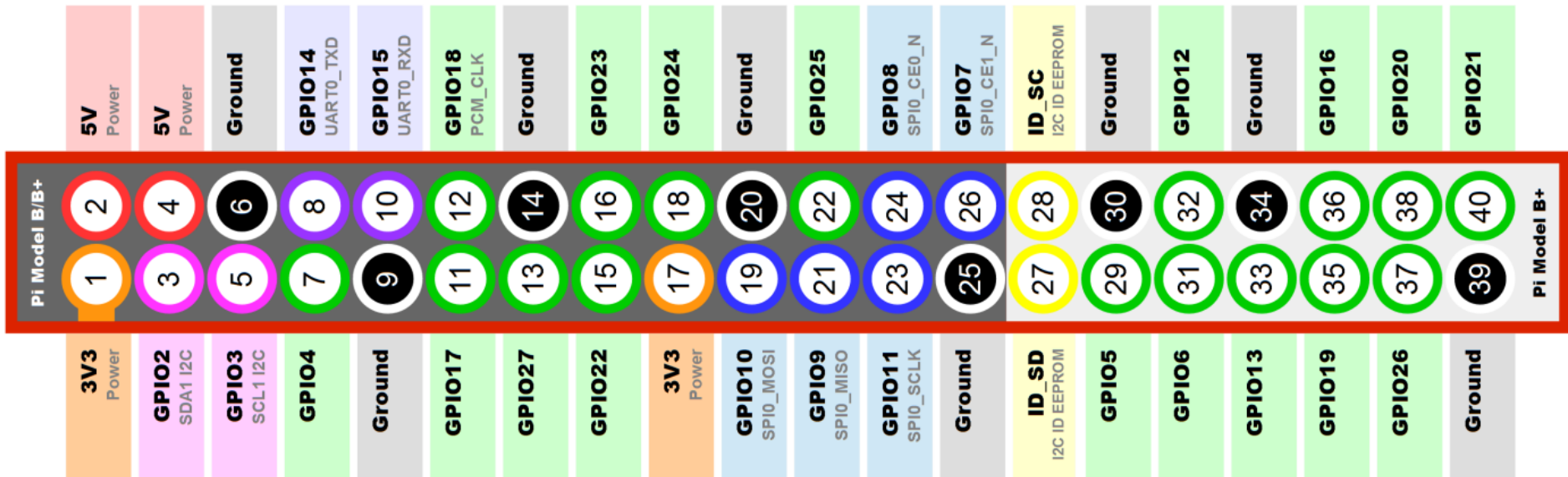
- Bluetooth Slave UART Board
 - UART인터페이스용 블루투스모듈
 - slave/device mode



라즈베리 파이 GPIO

4

〈라즈베리 파이 B+의 P1 헤더 핀 GPIO 배치도〉



wiringPi 라이브러리

5

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



실습 1 : wiringPi 라이브러리 설치(1/3)

6

- 라즈베리 파이 업데이트 및 업그레이드

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

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

- wiringPi 라이브러리 다운로드

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

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

- wiringPi 라이브러리 빌드 및 설치

```
$ cd wiringPi
```

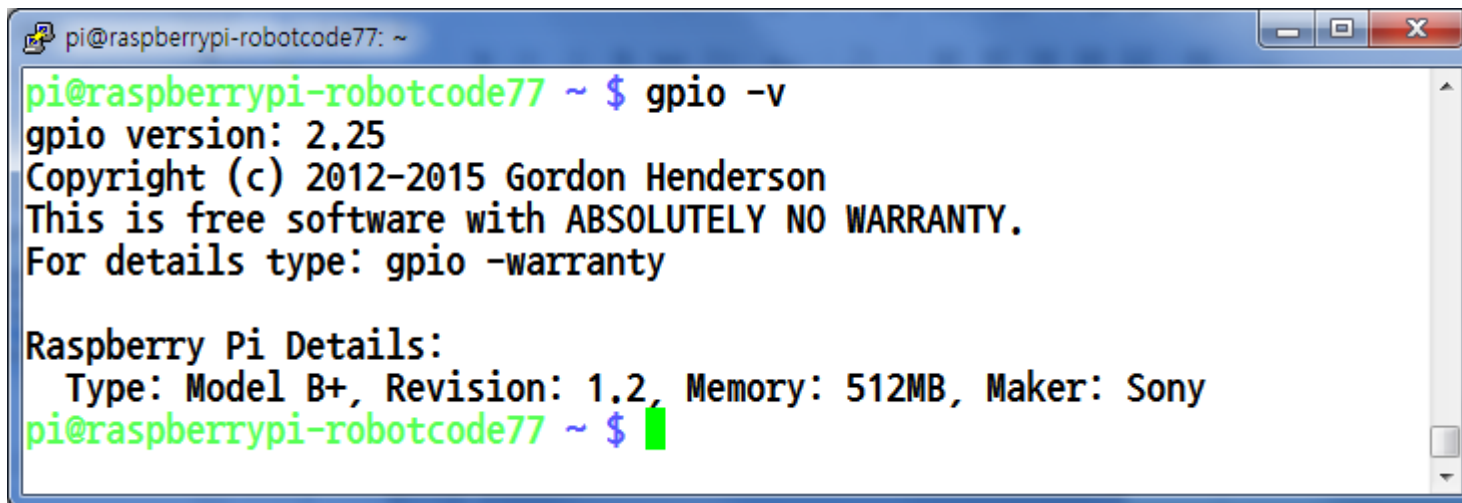
```
$ ./build
```

실습 1 : wiringPi 라이브러리 설치(2/3)

7

- 설치 확인

```
$ gpio -v
```



```
pi@raspberrypi-robotcode77: ~  
pi@raspberrypi-robotcode77 ~ $ gpio -v  
gpio version: 2.25  
Copyright (c) 2012-2015 Gordon Henderson  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type: gpio -warranty  
  
Raspberry Pi Details:  
Type: Model B+, Revision: 1.2, Memory: 512MB, Maker: Sony  
pi@raspberrypi-robotcode77 ~ $
```

실습 1 : wiringPi 라이브러리 설치(3/3)

8

• GPIO 핀 정보 확인

```
$ gpio readall
```

pi@raspberrypi-robotcode77: ~

```
pi@raspberrypi-robotcode77 ~$ gpio readall
```

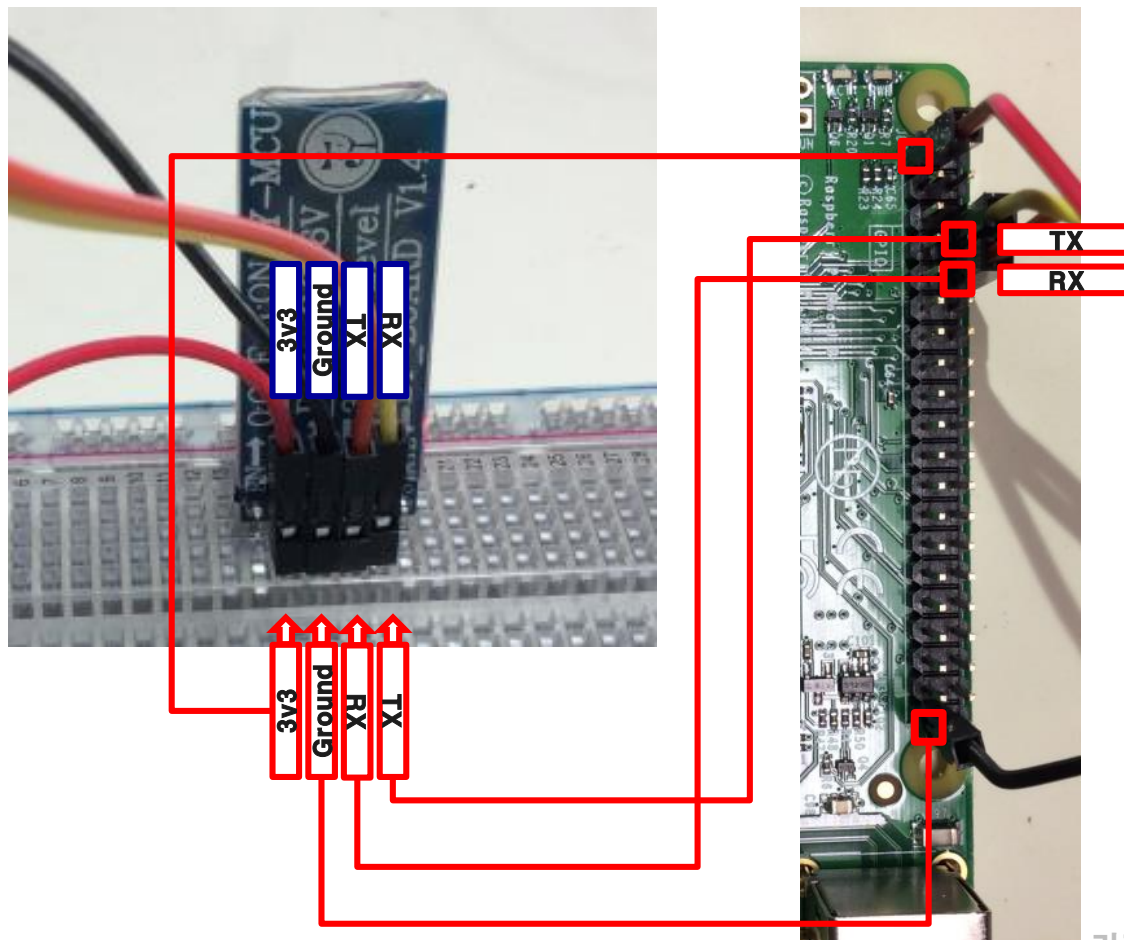
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1	2		5v		
2	8	SDA.1	IN	1	3	4		5V		
3	9	SCL.1	IN	1	5	6		0v		
4	7	GPIO. 7	IN	1	7	8	1	ALT0 TxD	15	14
		0v			9	10	1	ALT0 RxD	16	15
17	0	GPIO. 0	IN	0	11	12	0	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13	14		0v		
22	3	GPIO. 3	IN	0	15	16	0	GPIO. 4	4	23
		3.3v			17	18	0	GPIO. 5	5	24
10	12	MOSI	IN	0	19	20		0v		
9	13	MISO	IN	0	21	22	0	GPIO. 6	6	25
11	14	SCLK	IN	0	23	24	1	CE0	10	8
		0v			25	26	1	CE1	11	7
0	30	SDA.0	IN	1	27	28	1	SCL.0	31	1
5	21	GPIO.21	IN	1	29	30		0v		
6	22	GPIO.22	IN	1	31	32	0	GPIO.26	26	12
13	23	GPIO.23	IN	0	33	34		0v		
19	24	GPIO.24	IN	0	35	36	0	GPIO.27	27	16
26	25	GPIO.25	IN	0	37	38	0	GPIO.28	28	20
		0v			39	40	0	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)

9

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



실습 2 : Bluetooth 모듈 설정(2/12)

10

- /boot/cmdline.txt 파일 수정

- console=serial0,115200 부분 삭제

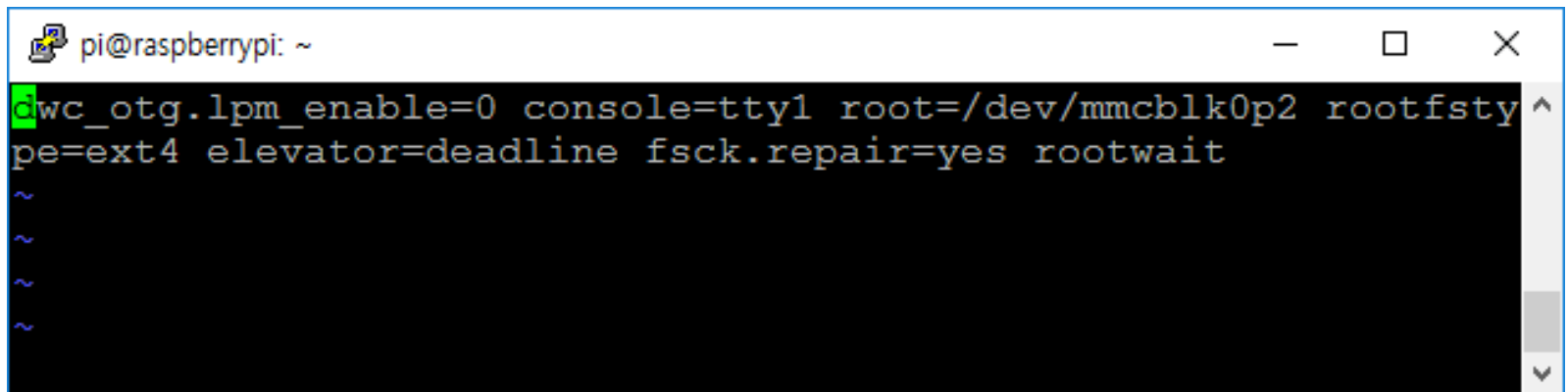
- 변경 전

```
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1  
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes  
rootwait
```

- 변경 후

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

- \$ sudo vim /boot/cmdline.txt



The screenshot shows a terminal window titled 'pi@raspberrypi: ~'. The command prompt is '\$'. The terminal displays the contents of the file /boot/cmdline.txt, which has been edited to remove the 'console=serial0,115200' part. The visible text is: 'dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait'. The first line is highlighted in green. There are four tilde '~' characters on the following lines, indicating the end of the file. The terminal window has standard window controls (minimize, maximize, close) in the top right corner.

실습 2 : Bluetooth 모듈 설정(4/12)

11

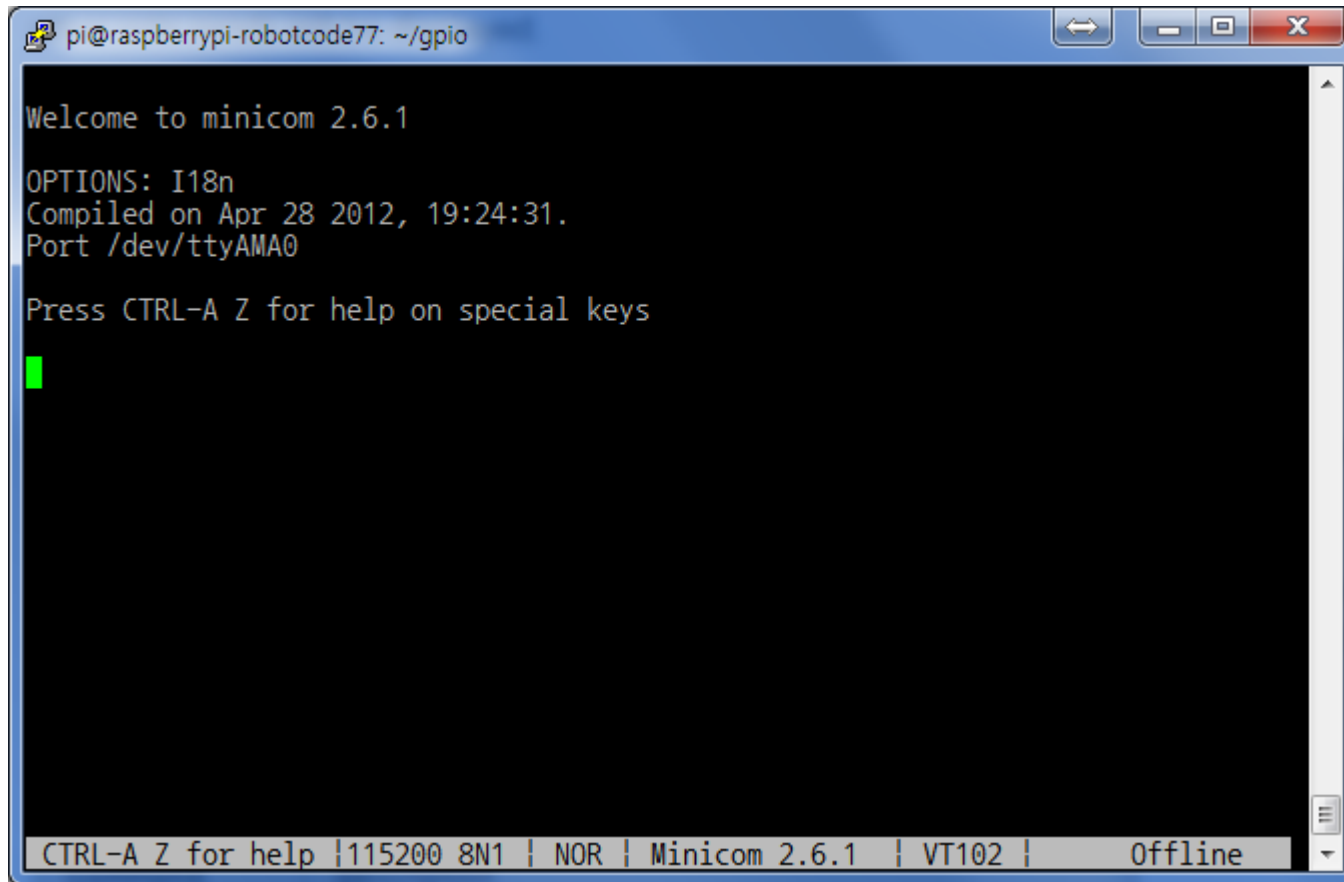
- 재부팅
 - `sudo reboot`
- SSH 재접속
- minicom 설치
 - `$ sudo apt-get install minicom`
- Bluetooth 접속
 - `$ minicom -b 9600 -o -D /dev/ttyAMA0`

실습 2 : Bluetooth 모듈 설정(5/12)

12

- minicom 실행화면

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



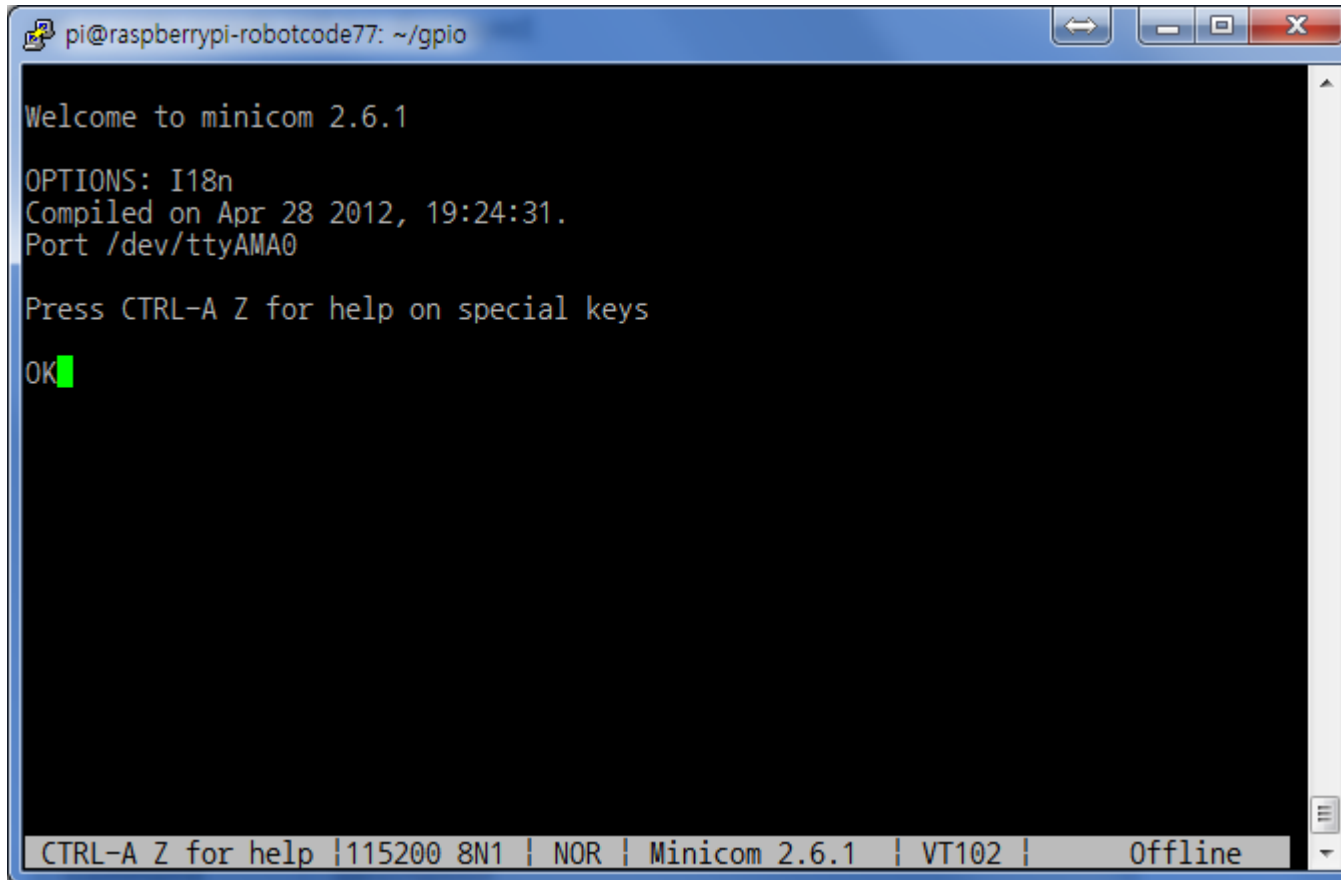
```
pi@raspberrypi-robotcode77: ~/gpio
Welcome to minicom 2.6.1
OPTIONS: I18n
Compiled on Apr 28 2012, 19:24:31.
Port /dev/ttyAMA0
Press CTRL-A Z for help on special keys
█
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.6.1 | VT102 | Offline
```

실습 2 : Bluetooth 모듈 설정(6/12)

13

- 테스트

- AT 키보드 입력후 OK 확인



```
pi@raspberrypi-robotcode77: ~/gpio
Welcome to minicom 2.6.1
OPTIONS: I18n
Compiled on Apr 28 2012, 19:24:31.
Port /dev/ttyAMA0
Press CTRL-A Z for help on special keys
OK
```

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.6.1 | VT102 | Offline

실습 2 : Bluetooth 모듈 설정(7/12)

14

• 명령어 종류

Command	Description
AT	Bluetooth module 테스트
AT+VERSION	모듈 버전 확인
AT+NAME 이름	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)

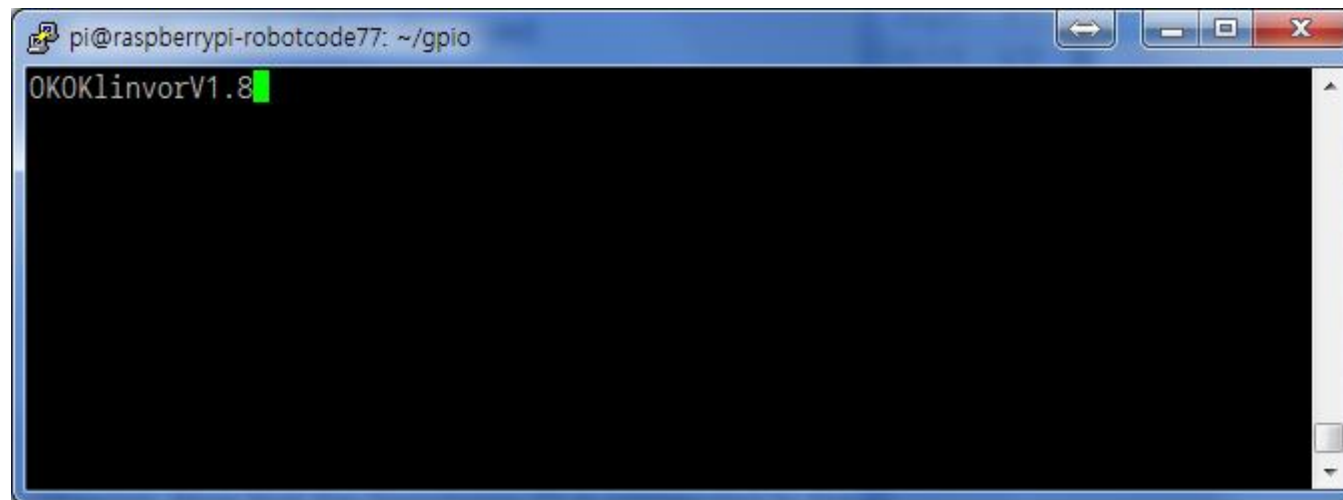
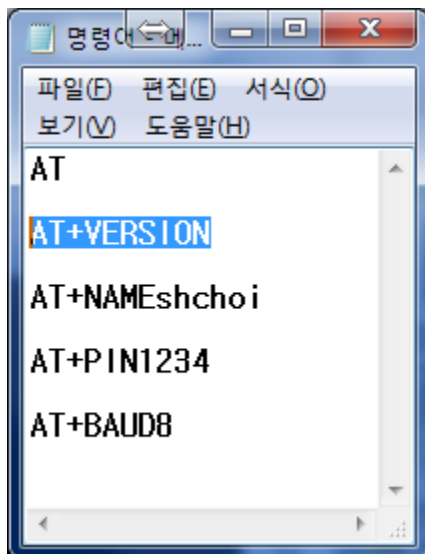
15

• 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)

16

- 이름 변경

- AT+NAMEshchoi-bt



A terminal window titled 'pi@raspberrypi-robotcode77: ~/gpio' with a blue header bar. The window contains the text 'OKOKlinvorV1.80Ksetname' followed by a green cursor. The window has standard Linux window controls (back, forward, close, maximize, minimize) in the top right corner.

- baud rate 변경

- AT+BAUD8

(minicom 재접속 필요)

실습 2 : Bluetooth 모듈 설정(10/12)

17

- minicom 종료
 - Ctrl + a 입력
 - z 입력

```

pi@raspberrypi-robotcode77: ~/gpio

Minicom Command Summary

Commands can be called by CTRL-A <key>

Main Functions                                Other Functions
Dialing directory..D  run script (Go)....G  Clear Screen.....C
Send files.....S    Receive files.....R  cOnfigure Minicom..O
comm Parameters....P  Add linefeed.....A  Suspend minicom....J
Capture on/off....L  Hangup.....H        eXit and reset.....X
send break.....F    initialize Modem...M  Quit with no reset..Q
Terminal settings..T  run Kermit.....K    Cursor key mode....I
lineWrap on/off....W  local Echo on/off..E  Help screen.....Z
Paste file.....Y     Timestamp toggle...N  scroll Back.....B

Select function or press Enter for none.█

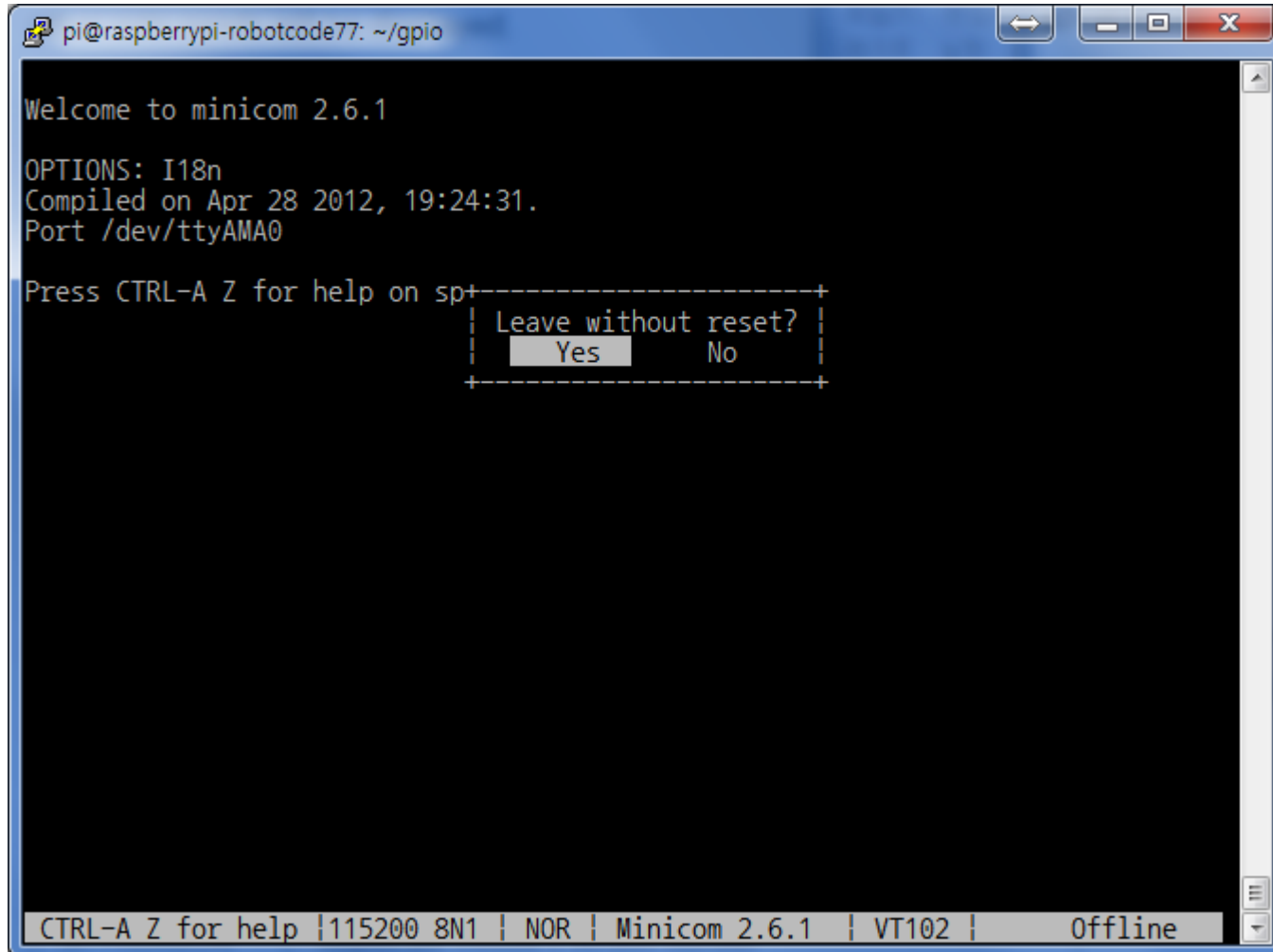
Written by Miquel van Smoorenburg 1991-1995
Some additions by Jukka Lahtinen 1997-2000
i18n by Arnaldo Carvalho de Melo 1998

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.6.1 | VT102 | Offline
  
```

실습 2 : Bluetooth 모듈 설정(11/12)

18

- q 입력
- Yes Enter



```
pi@raspberrypi-robotcode77: ~/gpio
Welcome to minicom 2.6.1
OPTIONS: I18n
Compiled on Apr 28 2012, 19:24:31.
Port /dev/ttyAMA0
Press CTRL-A Z for help on sp
Leave without reset?
  Yes      No
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.6.1 | VT102 | Offline
```

실습 2 : Bluetooth 모듈 설정(12/12)

19

- minicom 실행

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

- Pin 번호변경

- AT+PIN5216

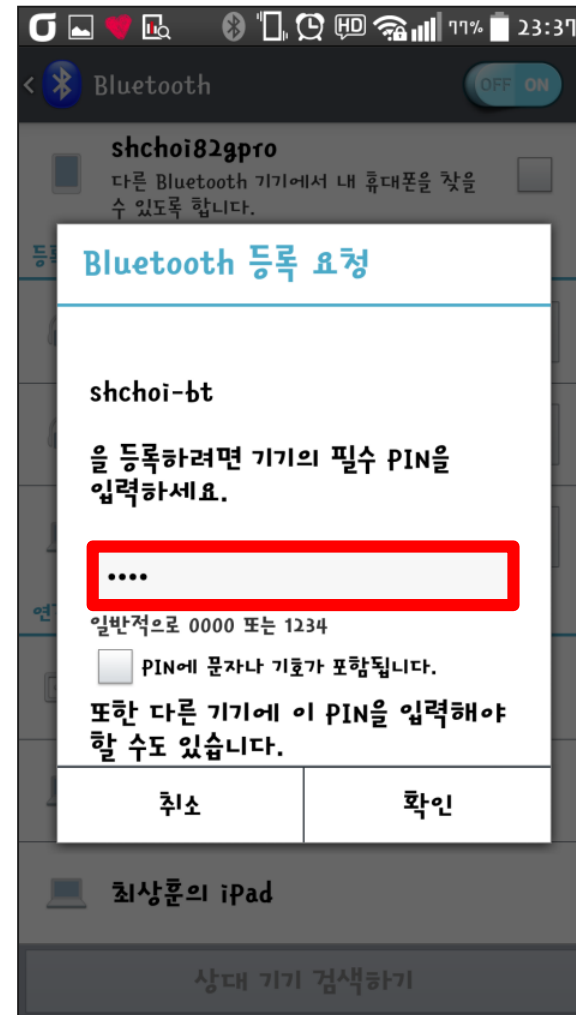


- minicom 종료

실습 3 : Bluetooth 페어링(1/2)

20

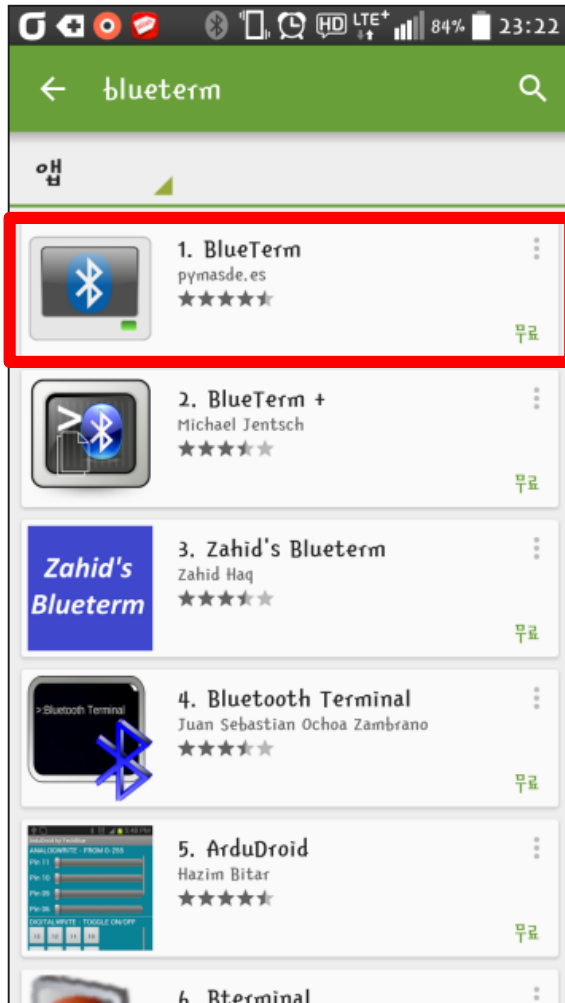
- 안드로이드 bluetooth 기기 등록



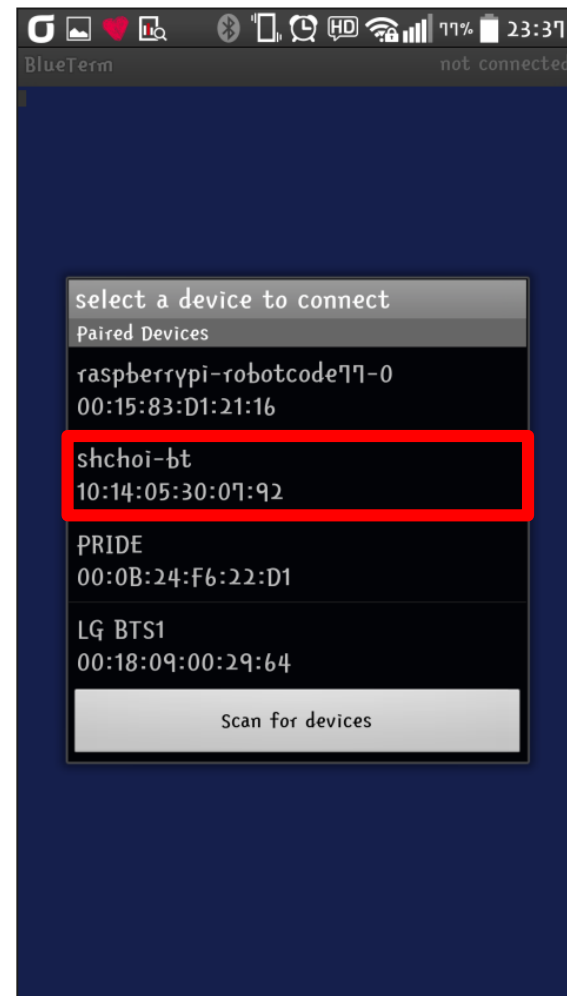
실습 3 : Bluetooth 페어링(2/2)

21

- 안드로이드 bluetooth 터미널 앱 설치
 - blueterm 설치



- 연결



실습 4 : Bluetooth 통신(1/4)

22

- 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)

23

```
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**

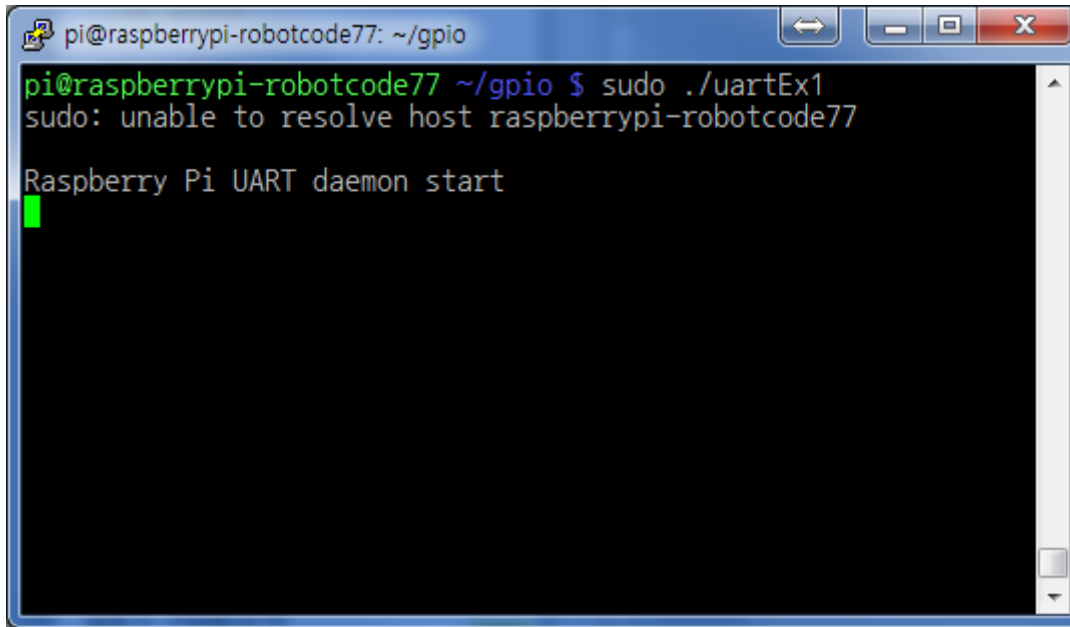
```
$ gcc -Wall -W -lwiringPi uartEx1.c -o uartEx1
```

- **실행**

```
$ sudo ./uartEx1
```

실습 4 : Bluetooth 통신(3/4)

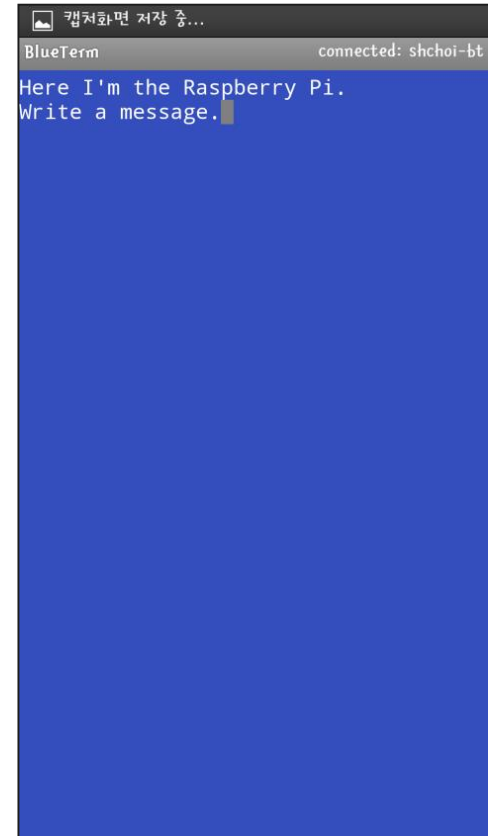
24



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx1
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
```

< Raspberry Pi >



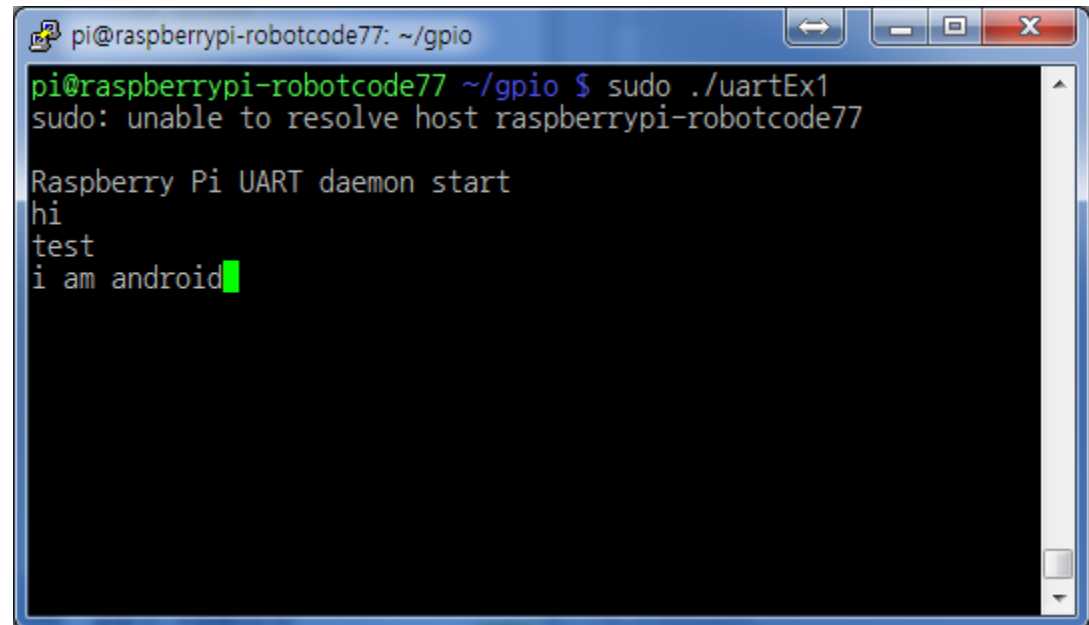
```
캡처화면 저장 중...
BlueTerm                                connected: shchoi-bt
Here I'm the Raspberry Pi.
Write a message.
```


실습 4 : Bluetooth 통신(4/4)

25

• 통신 테스트

- 안드로이드의 터미널에 텍스트를 입력하면
- RaspberryPi의 터미널창 그대로 출력됨



< Raspberry Pi >

실습 5 : Bluetooth 통신 (1/4)

26

- 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)

27

```
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

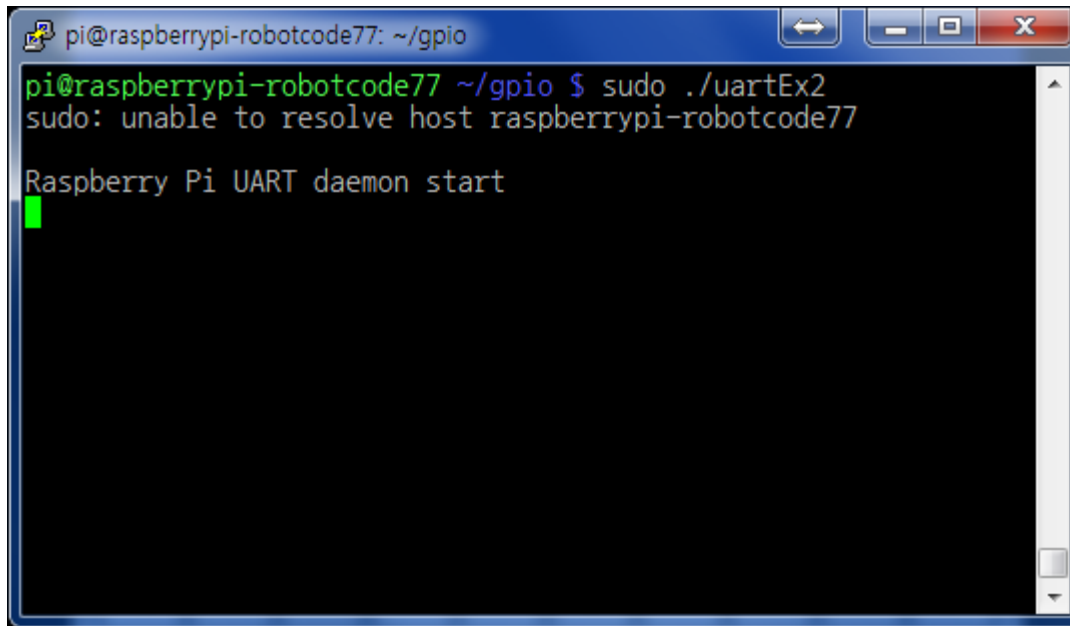
```
$ gcc -Wall -W -lwiringPi uartEx2.c -o uartEx2
```

- 실행

```
$ sudo ./uartEx2
```

실습 5 : Bluetooth 통신 (3/4)

28



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx2
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
```

< Raspberry Pi >



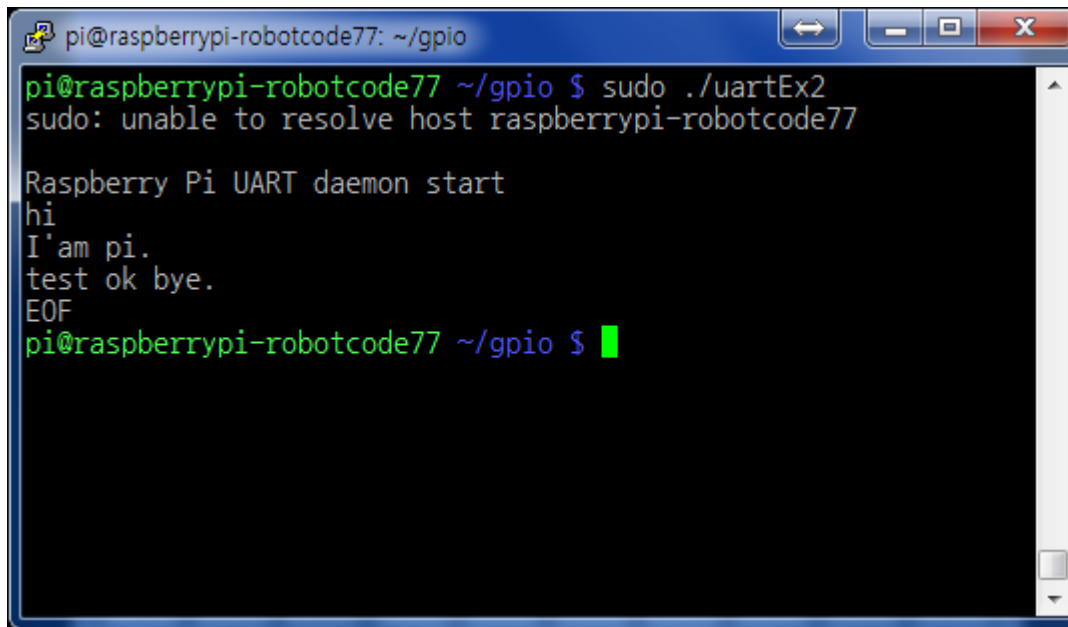
```
BlueTerm connected: shchoi-bt
Here I'm the Raspberry Pi.
I'll send a message.
```

실습 5 : Bluetooth 통신 (4/4)

29

• 통신 테스트

- RaspberryPi의 터미널에 텍스트 입력하면
- 안드로이드 터미널창에 그대로 출력됨



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx2
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
hi
I'am pi.
test ok bye.
EOF
pi@raspberrypi-robotcode77 ~/gpio $
```

< Raspberry Pi >



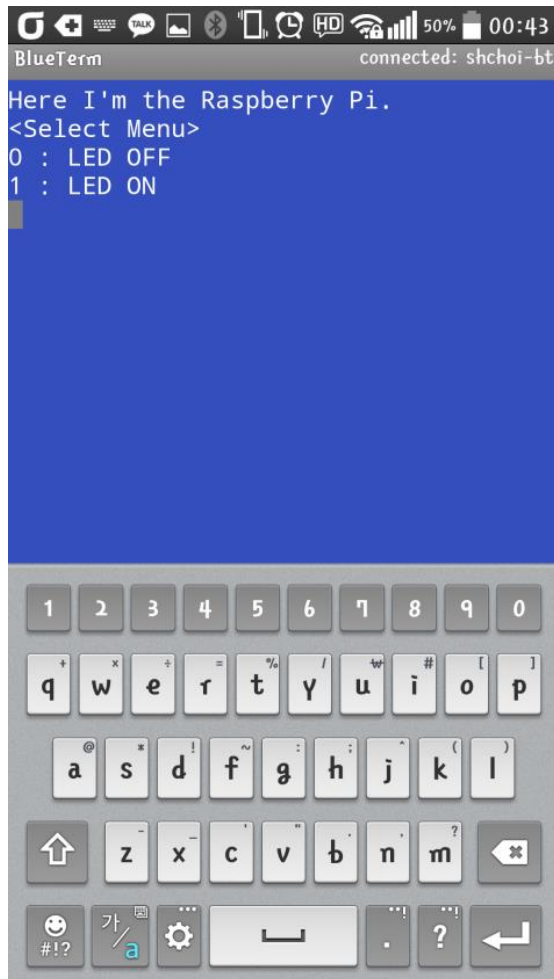
```
BlueTerm
connected: shchoi-bt

Here I'm the Raspberry Pi.
I'll send a message.
hi
I'am pi.
test ok bye.
```

미션 1 : LED 제어

30

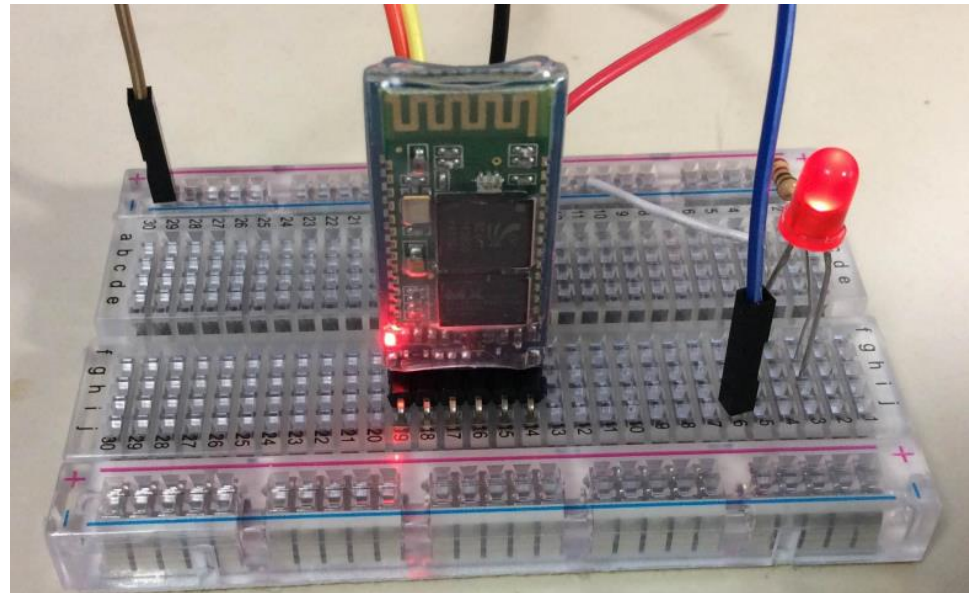
- Bluetooth 통신을 통한 Raspberry Pi GPIO 제어하기
 - 메뉴출력



미션 1 : LED 제어

31

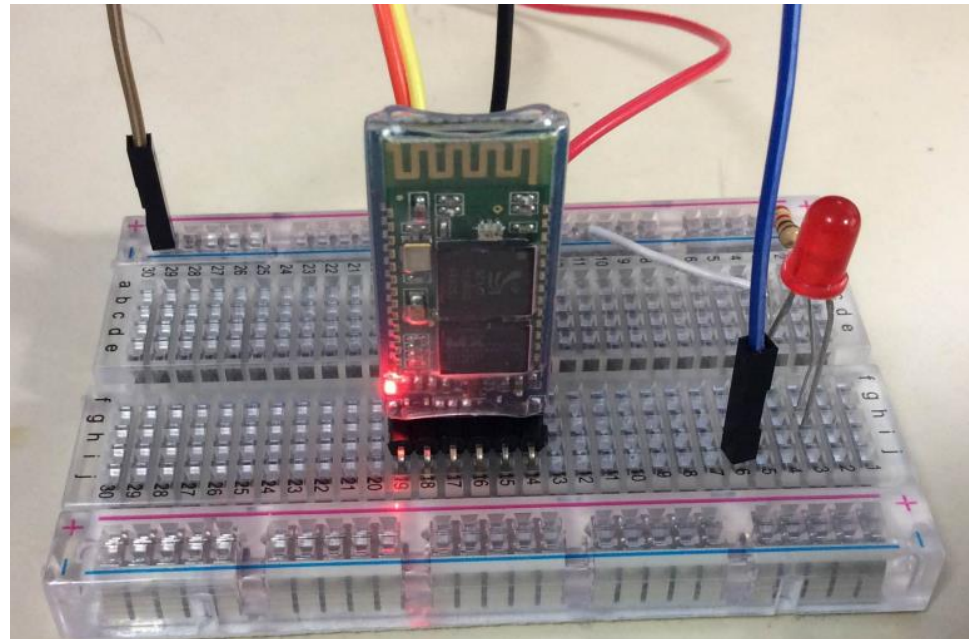
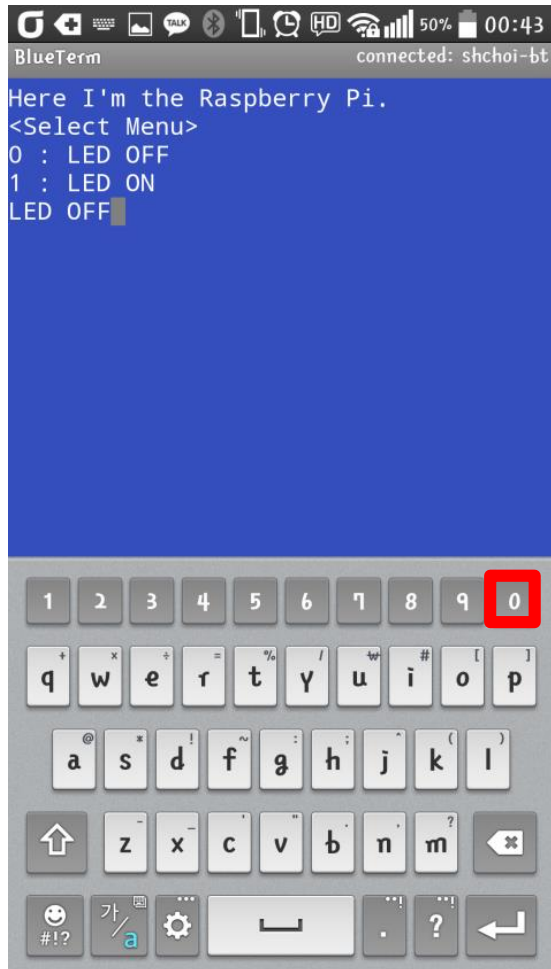
- Raspberry Pi GPIO 제어
 - GPIO에 연결된 LED를 ON



미션 1 : LED 제어

32

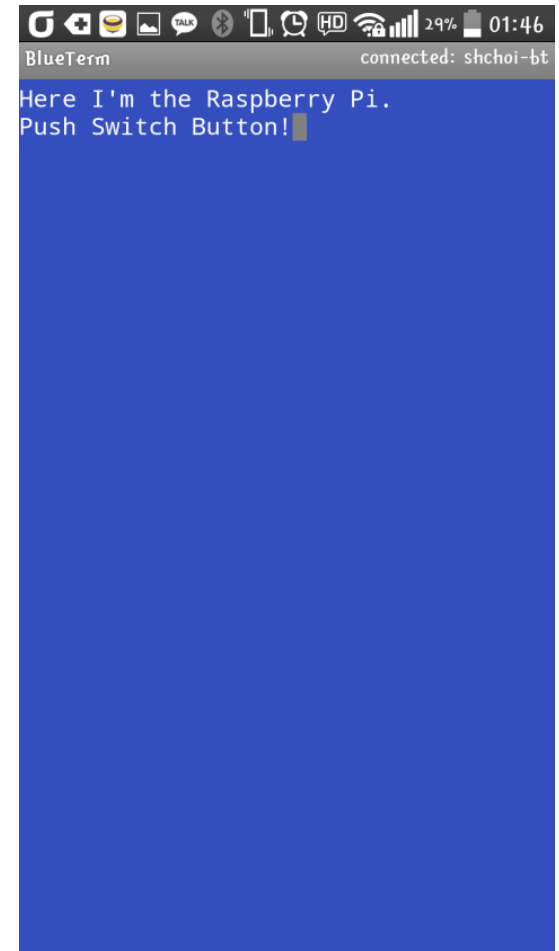
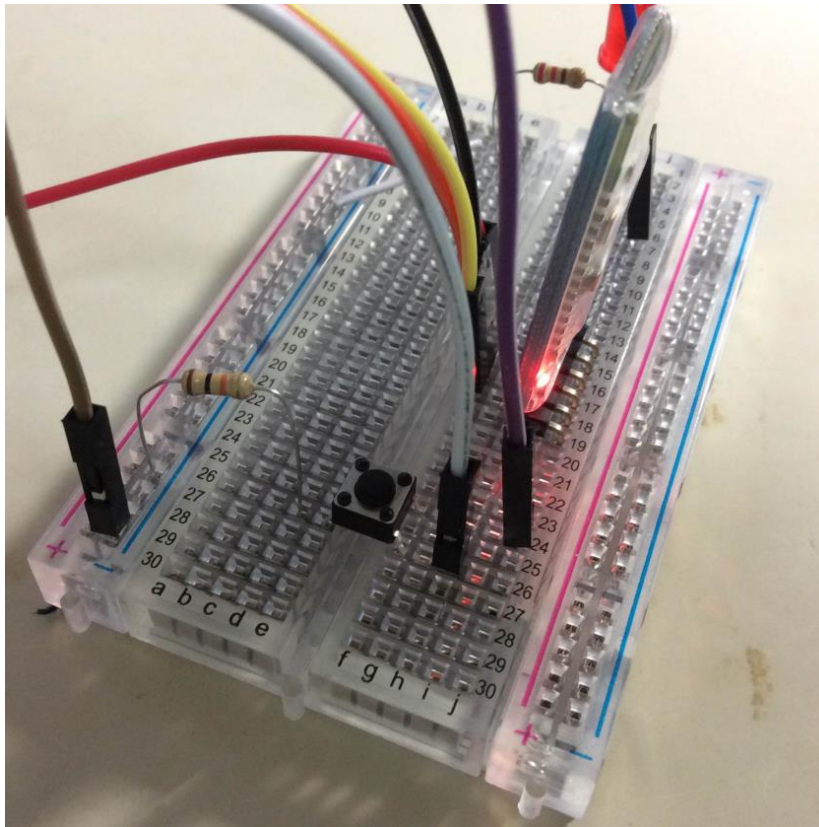
- Raspberry Pi GPIO 제어
 - GPIO에 연결된 LED를 OFF



미션 2 : Swtich 모니터링

33

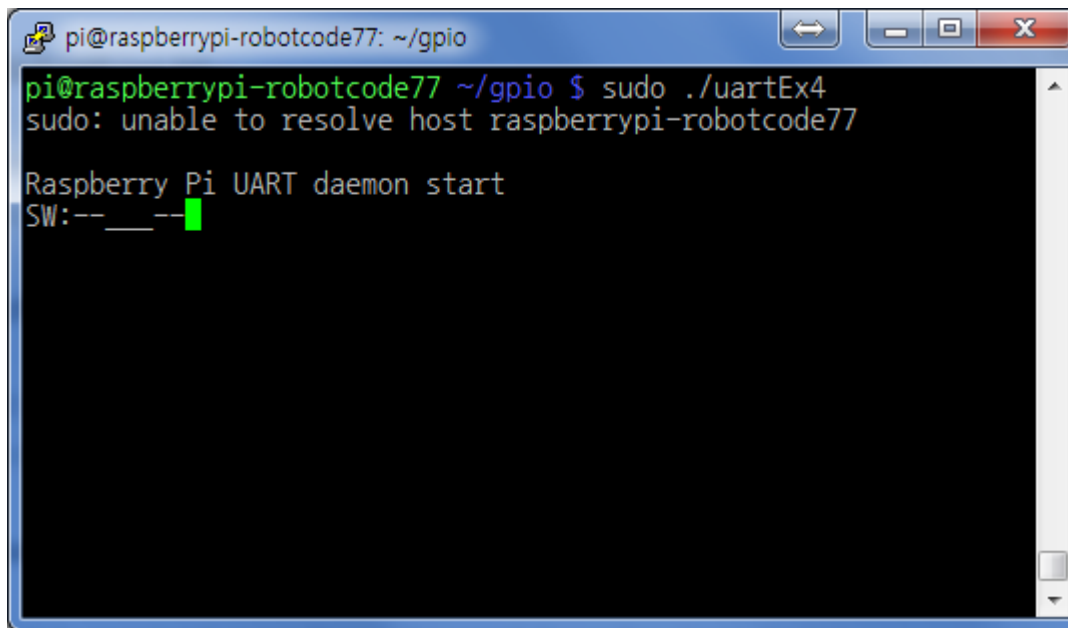
- Raspberry Pi GPIO 제어
 - GPIO에 연결된 Swtich 상태 모니터링



미션 2 : Swtich 모니터링

34

- Raspberry Pi GPIO 제어
 - GPIO에 연결된 Swtich 상태 모니터링



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx4
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
SW:--__--
```

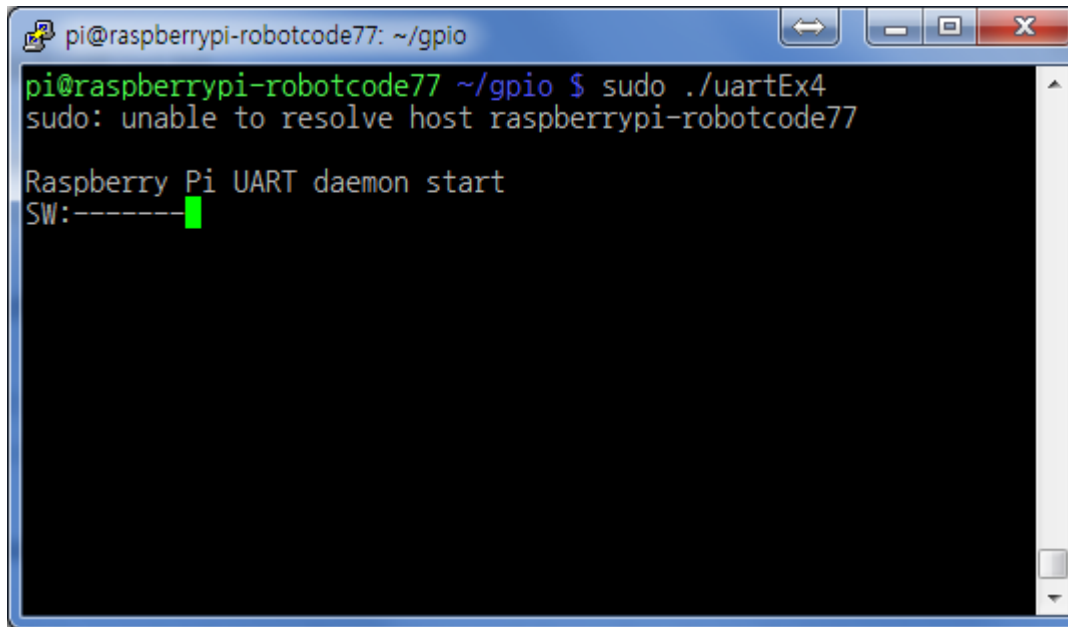


```
BlueTerm connected: shchoi-bt
Here I'm the Raspberry Pi.
Push Switch Button!
SW:--__--
```

미션 2 : Swtich 모니터링

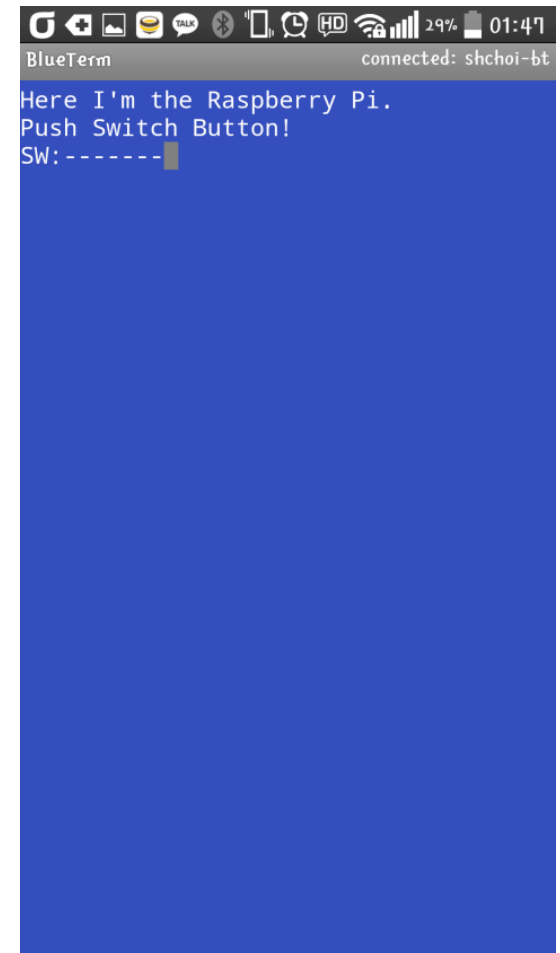
35

- Raspberry Pi GPIO 제어
 - GPIO에 연결된 Swtich 상태 모니터링



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx4
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
SW:-----█
```



```
BlueTerm connected: shchoi-bt
Here I'm the Raspberry Pi.
Push Switch Button!
SW:-----█
```

Bluetooth Dongle

로봇SW 교육원

최상훈(shchoi82@gmail.com)

Bluetooth Dongle

37

- Bluetooth Dongle
 - Bluetooth CSR 4.0 Harmony



실습1 : 블루투스 dongle 연결

38

- 블루투스 dongle 연결



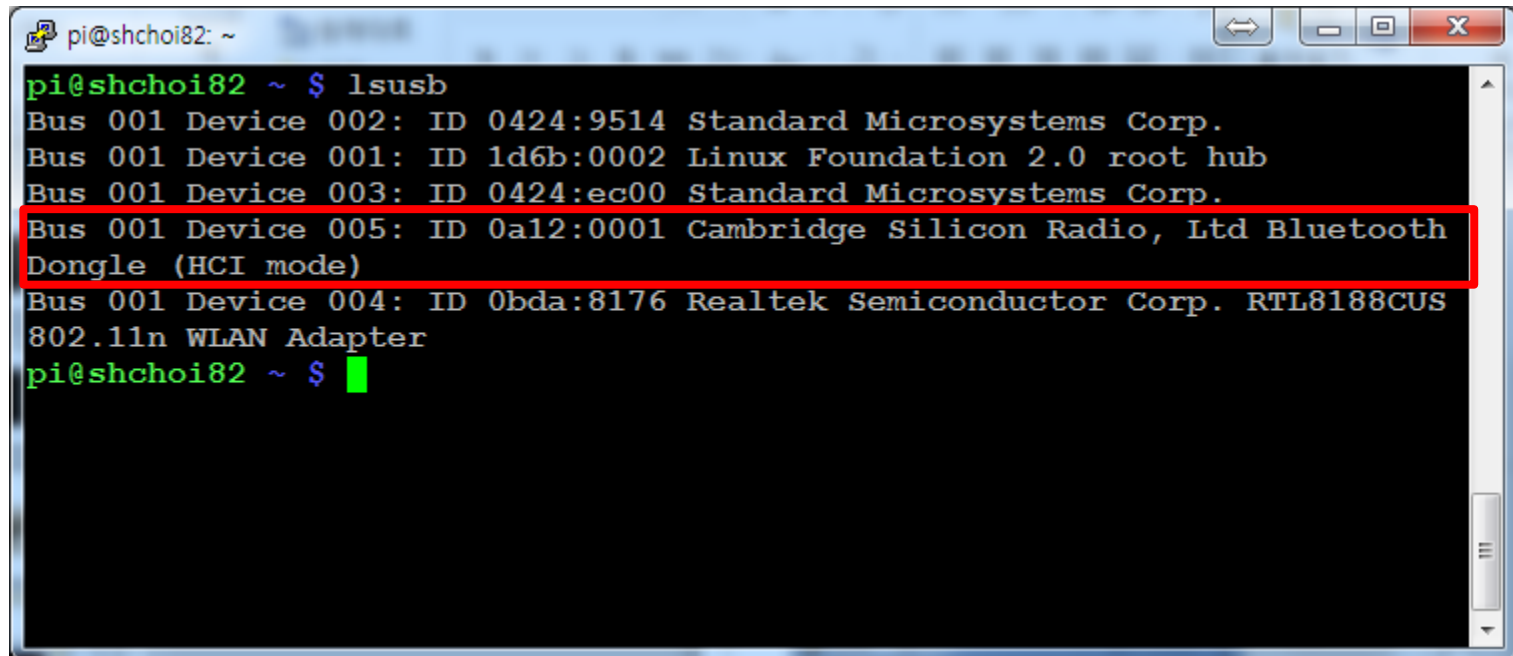
실습2-1 : 블루투스 Dongle 설정

39

- 블루투스 Dongle 연결 확인

```
$ lsusb
```

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



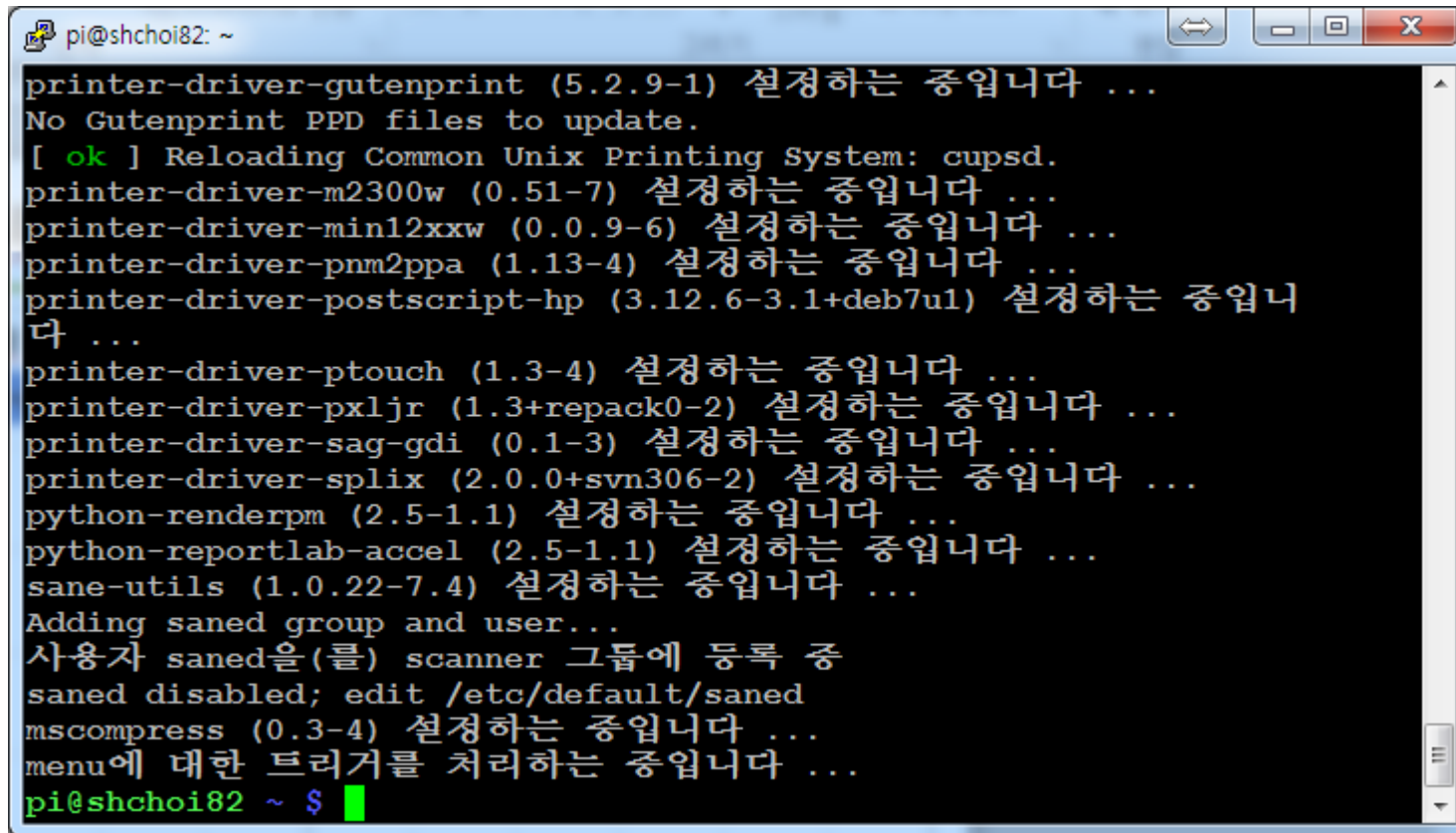
```
pi@shchoi82: ~  
pi@shchoi82 ~ $ lsusb  
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.  
Bus 001 Device 005: ID 0a12:0001 Cambridge Silicon Radio, Ltd Bluetooth  
Dongle (HCI mode)  
Bus 001 Device 004: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS  
802.11n WLAN Adapter  
pi@shchoi82 ~ $
```

실습2-2 : 블루투스 동글 설정

40

- bluez bluez-util minicom 설치

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



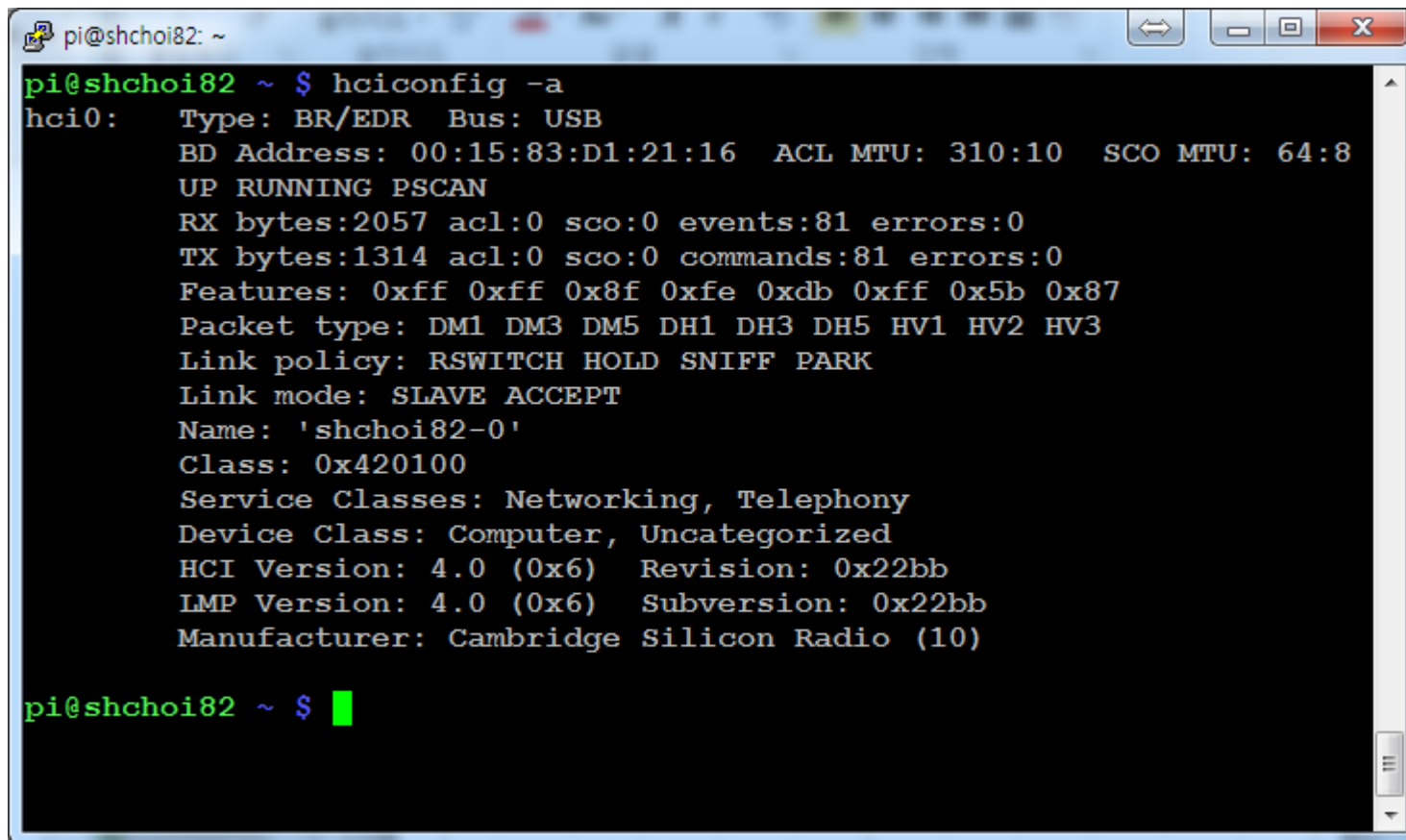
```
pi@shchoi82: ~  
printer-driver-gutenprint (5.2.9-1) 설정하는 중입니다 ...  
No Gutenprint PPD files to update.  
[ ok ] Reloading Common Unix Printing System: cupsd.  
printer-driver-m2300w (0.51-7) 설정하는 중입니다 ...  
printer-driver-min12xxw (0.0.9-6) 설정하는 중입니다 ...  
printer-driver-pnm2ppa (1.13-4) 설정하는 중입니다 ...  
printer-driver-postscript-hp (3.12.6-3.1+deb7u1) 설정하는 중입니  
다 ...  
printer-driver-ptouch (1.3-4) 설정하는 중입니다 ...  
printer-driver-pxljr (1.3+repack0-2) 설정하는 중입니다 ...  
printer-driver-sag-gdi (0.1-3) 설정하는 중입니다 ...  
printer-driver-splix (2.0.0+svn306-2) 설정하는 중입니다 ...  
python-renderpm (2.5-1.1) 설정하는 중입니다 ...  
python-reportlab-accel (2.5-1.1) 설정하는 중입니다 ...  
sane-utils (1.0.22-7.4) 설정하는 중입니다 ...  
Adding saned group and user...  
사용자 saned을(를) scanner 그룹에 등록 중  
saned disabled; edit /etc/default/saned  
mscompress (0.3-4) 설정하는 중입니다 ...  
menu에 대한 트리거를 처리하는 중입니다 ...  
pi@shchoi82 ~ $
```


실습2-3 : 블루투스 동글 설정

41

- 블루투스 장치 정보 확인

```
$ hciconfig -a
```



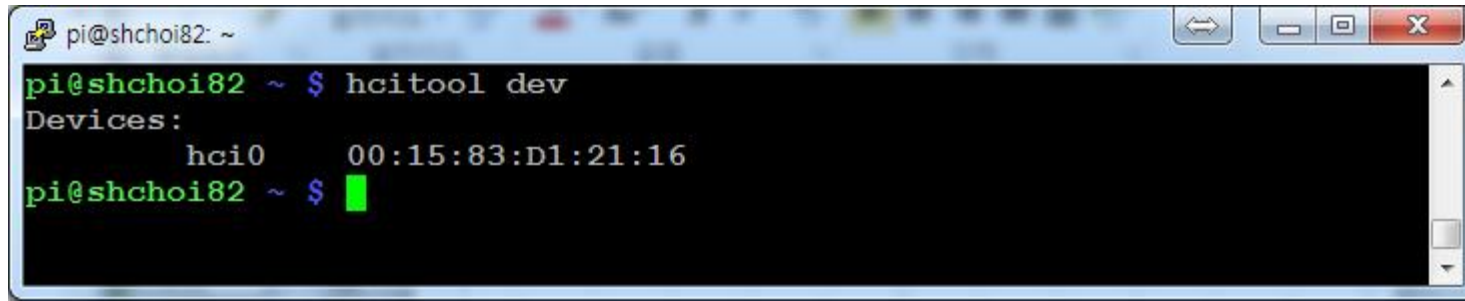
```
pi@shchoi82: ~  
pi@shchoi82 ~ $ hciconfig -a  
hci0:    Type: BR/EDR  Bus: USB  
        BD Address: 00:15:83:D1:21:16  ACL MTU: 310:10  SCO MTU: 64:8  
        UP RUNNING PSCAN  
        RX bytes:2057 acl:0 sco:0 events:81 errors:0  
        TX bytes:1314 acl:0 sco:0 commands:81 errors:0  
        Features: 0xff 0xff 0x8f 0xfe 0xdb 0xff 0x5b 0x87  
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3  
        Link policy: RSWITCH HOLD SNIFF PARK  
        Link mode: SLAVE ACCEPT  
        Name: 'shchoi82-0'  
        Class: 0x420100  
        Service Classes: Networking, Telephony  
        Device Class: Computer, Uncategorized  
        HCI Version: 4.0 (0x6)  Revision: 0x22bb  
        LMP Version: 4.0 (0x6)  Subversion: 0x22bb  
        Manufacturer: Cambridge Silicon Radio (10)  
  
pi@shchoi82 ~ $
```

실습2-4 : 블루투스 동글 설정

42

- 블루투스 MAC 주소 확인

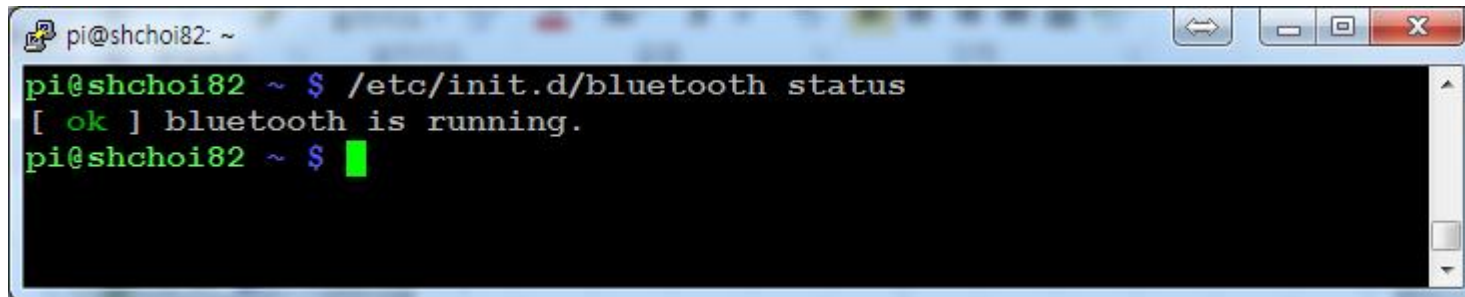
```
$ hcitool dev
```



```
pi@shchoi82: ~  
pi@shchoi82 ~ $ hcitool dev  
Devices:  
    hci0    00:15:83:D1:21:16  
pi@shchoi82 ~ $
```

- 블루투스 동작 확인

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



```
pi@shchoi82: ~  
pi@shchoi82 ~ $ /etc/init.d/bluetooth status  
[ ok ] bluetooth is running.  
pi@shchoi82 ~ $
```

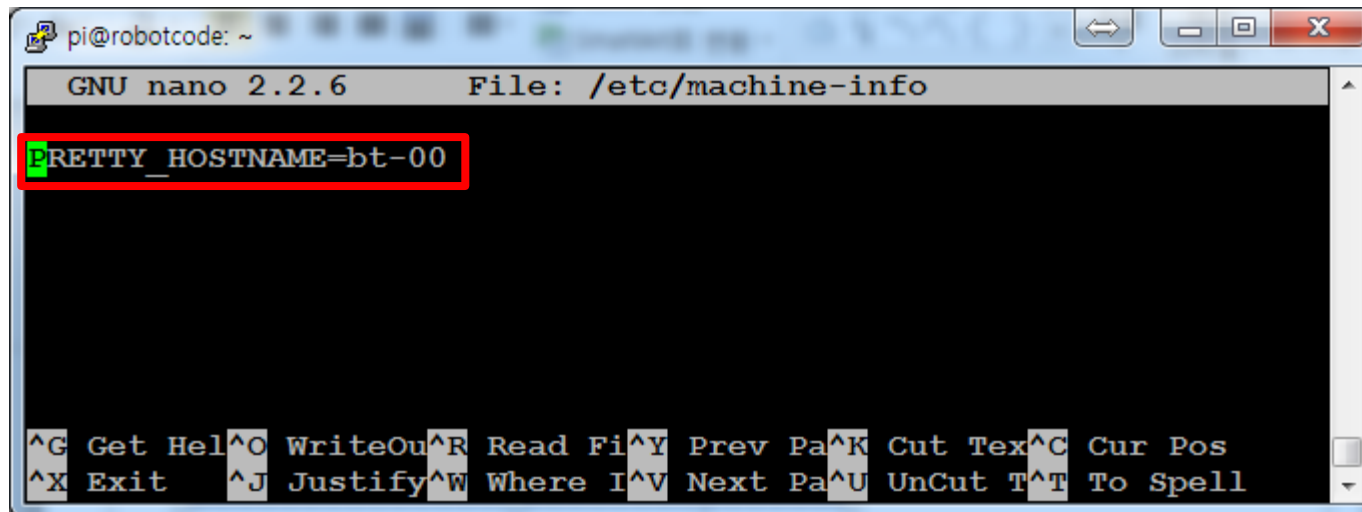
실습2-5 : 블루투스 동글 설정

43

- Name 변경

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

PRETTY_HOSTNAME=bluetooth 이름



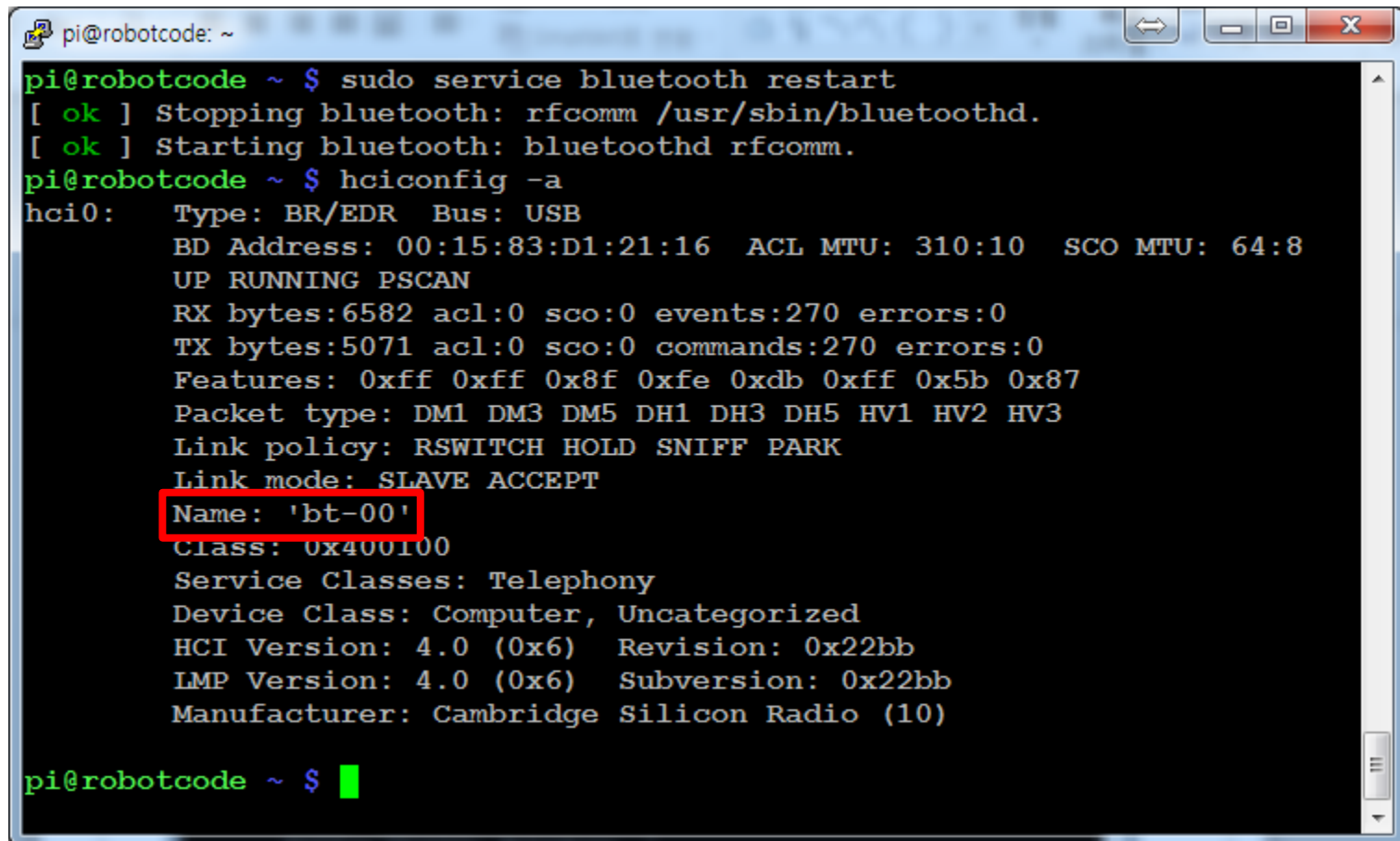
```
pi@robotcode: ~  
GNU nano 2.2.6 File: /etc/machine-info  
PRETTY_HOSTNAME=bt-00  
  
^G Get Hel ^O WriteOu ^R Read Fi ^Y Prev Pa ^K Cut Tex ^C Cur Pos  
^X Exit    ^J Justify ^W Where I ^V Next Pa ^U UnCut T ^T To Spell
```

실습2-6 : 블루투스 동글 설정

44

- Name 변경

```
$ sudo service bluetooth restart
```



```
pi@robotcode: ~  
pi@robotcode ~ $ sudo service bluetooth restart  
[ ok ] Stopping bluetooth: rfcomm /usr/sbin/bluetoothd.  
[ ok ] Starting bluetooth: bluetoothd rfcomm.  
pi@robotcode ~ $ hciconfig -a  
hci0:    Type: BR/EDR  Bus: USB  
        BD Address: 00:15:83:D1:21:16  ACL MTU: 310:10  SCO MTU: 64:8  
        UP RUNNING PSCAN  
        RX bytes:6582 acl:0 sco:0 events:270 errors:0  
        TX bytes:5071 acl:0 sco:0 commands:270 errors:0  
        Features: 0xff 0xff 0x8f 0xfe 0xdb 0xff 0x5b 0x87  
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3  
        Link policy: RSWITCH HOLD SNIFF PARK  
        Link mode: SLAVE ACCEPT  
        Name: 'bt-00'  
        Class: 0x400100  
        Service Classes: Telephony  
        Device Class: Computer, Uncategorized  
        HCI Version: 4.0 (0x6)  Revision: 0x22bb  
        LMP Version: 4.0 (0x6)  Subversion: 0x22bb  
        Manufacturer: Cambridge Silicon Radio (10)  
  
pi@robotcode ~ $
```

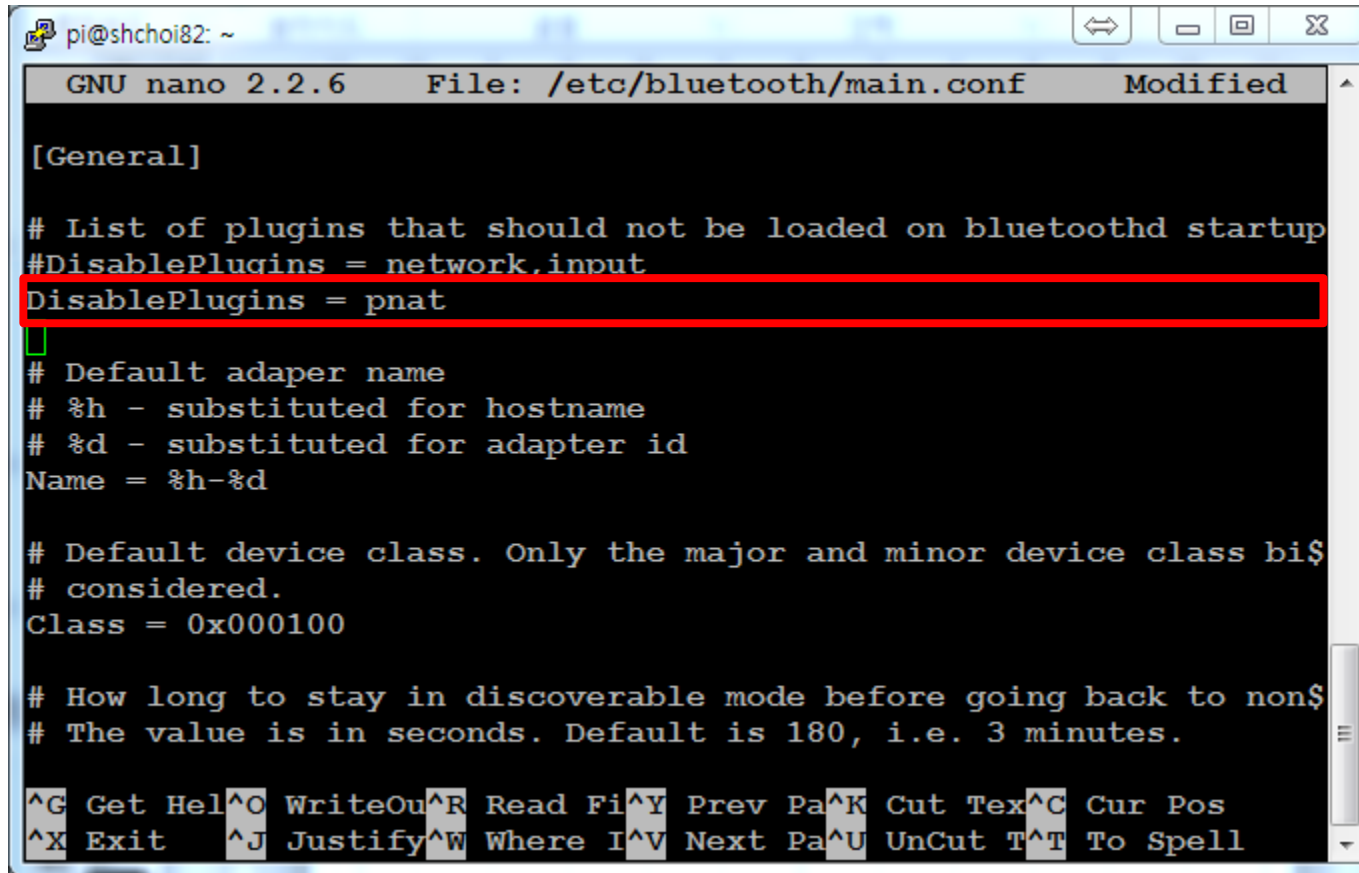
실습2-7 : 블루투스 동글 설정

45

- Bluez PNAT 플러그인 비활성화

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

DisablePlugins = pnat



```
pi@shchoi82: ~  
GNU nano 2.2.6 File: /etc/bluetooth/main.conf Modified  
[General]  
# List of plugins that should not be loaded on bluetoothd startup  
#DisablePlugins = network,input  
DisablePlugins = pnat  
#  
# Default adapter name  
# %h - substituted for hostname  
# %d - substituted for adapter id  
Name = %h-%d  
  
# Default device class. Only the major and minor device class bits  
# considered.  
Class = 0x000100  
  
# How long to stay in discoverable mode before going back to non-discoverable  
# The value is in seconds. Default is 180, i.e. 3 minutes.  
  
^G Get Help ^O Write Out ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where I Am ^V Next Page ^U UnCut Text ^T To Spell
```

실습2-8 : 블루투스 Dongle 설정

46

- 재부팅

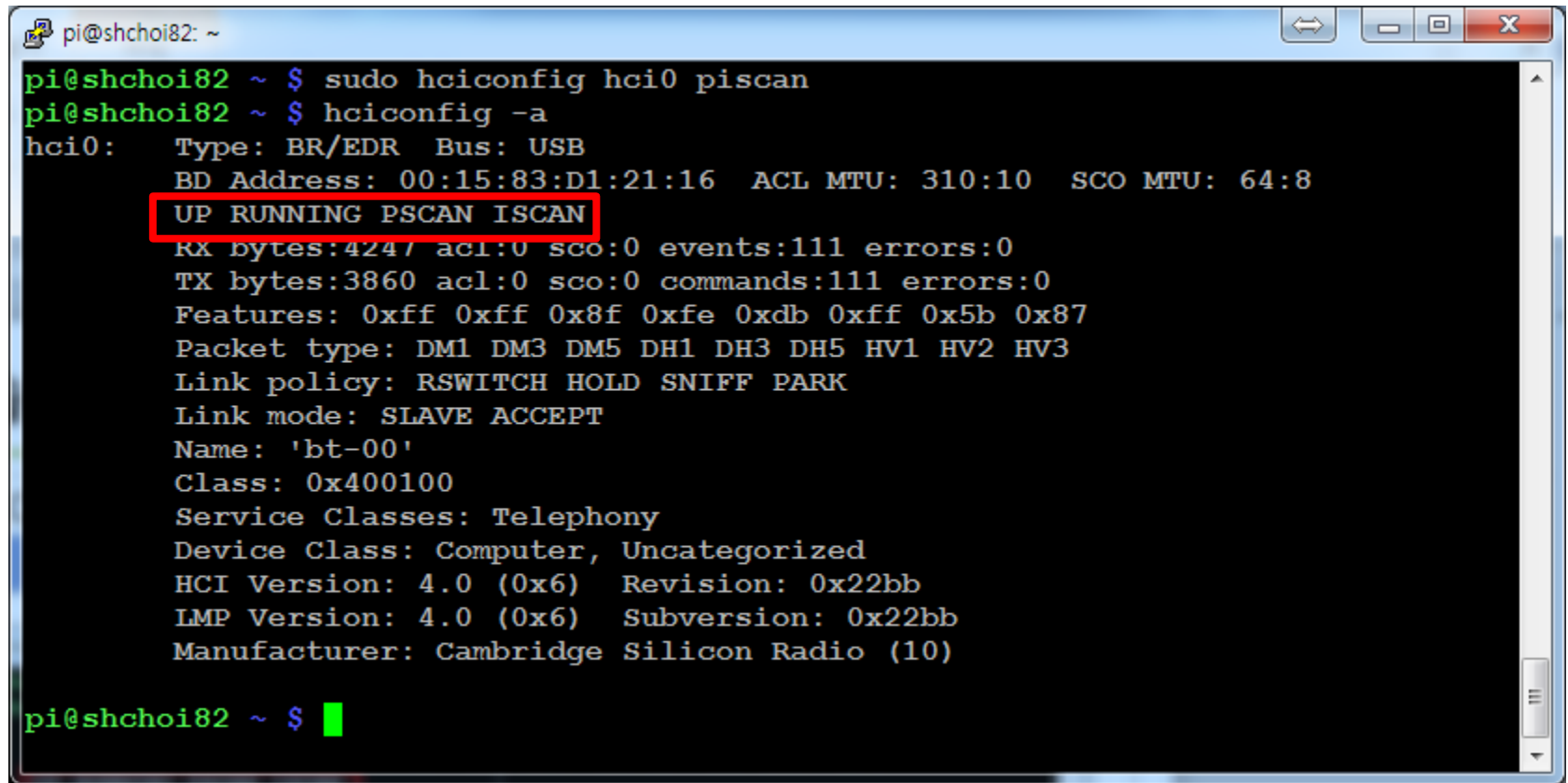
```
$ sudo reboot
```

실습3-1 : 블루투스 등록

47

- PSCAN ISCAN 활성화

```
$ sudo hciconfig hci0 piscan
```

A terminal window titled 'pi@shchoi82: ~' with standard window controls. It shows the execution of 'sudo hciconfig hci0 piscan' followed by 'hciconfig -a'. The output lists various Bluetooth interface details. The line 'UP RUNNING PSCAN ISCAN' is highlighted with a red rectangle. The terminal ends with a green cursor on the prompt.

```
pi@shchoi82 ~ $ sudo hciconfig hci0 piscan
pi@shchoi82 ~ $ hciconfig -a
hci0:    Type: BR/EDR  Bus: USB
        BD Address: 00:15:83:D1:21:16  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING PSCAN ISCAN
        RX bytes:4247 acl:0 sco:0 events:111 errors:0
        TX bytes:3860 acl:0 sco:0 commands:111 errors:0
        Features: 0xff 0xff 0x8f 0xfe 0xdb 0xff 0x5b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
        Name: 'bt-00'
        Class: 0x400100
        Service Classes: Telephony
        Device Class: Computer, Uncategorized
        HCI Version: 4.0 (0x6)  Revision: 0x22bb
        LMP Version: 4.0 (0x6)  Subversion: 0x22bb
        Manufacturer: Cambridge Silicon Radio (10)

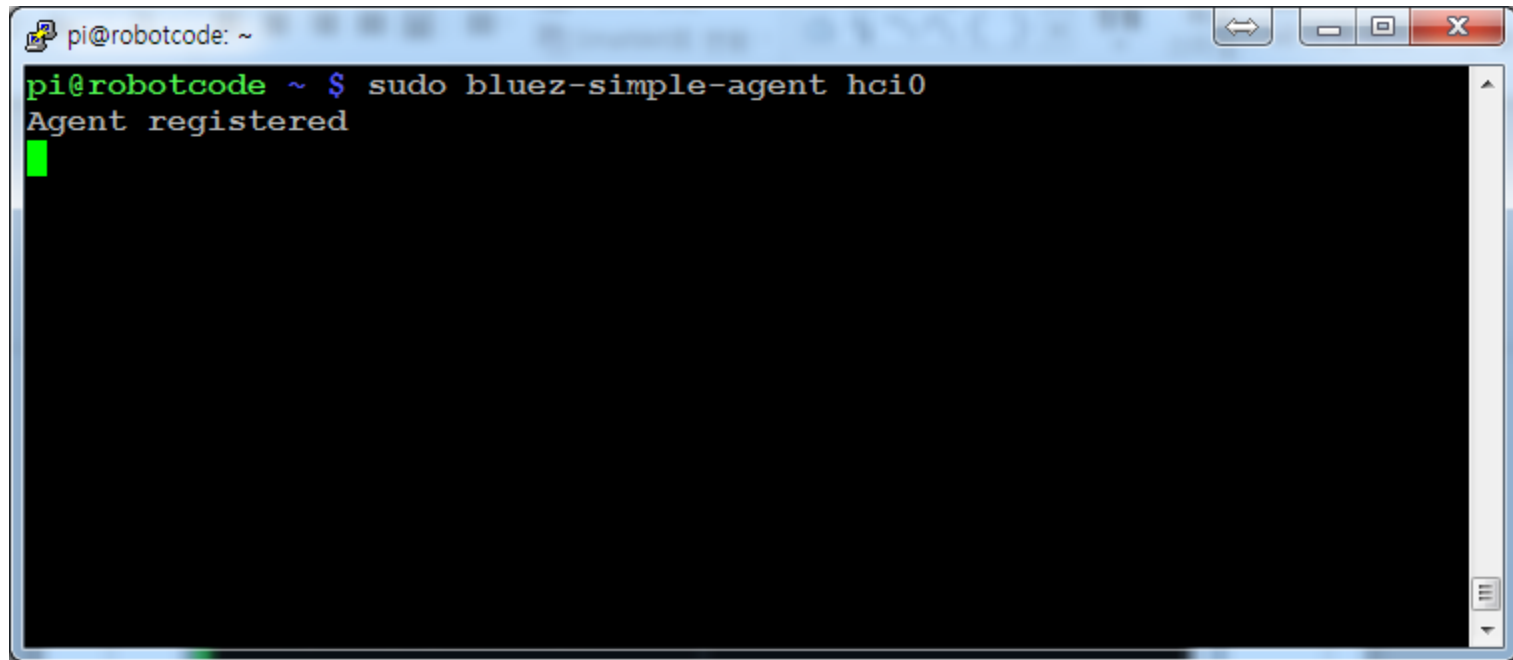
pi@shchoi82 ~ $
```

실습3-2 : 블루투스 등록

48

- bluez-simple-agent 실행

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



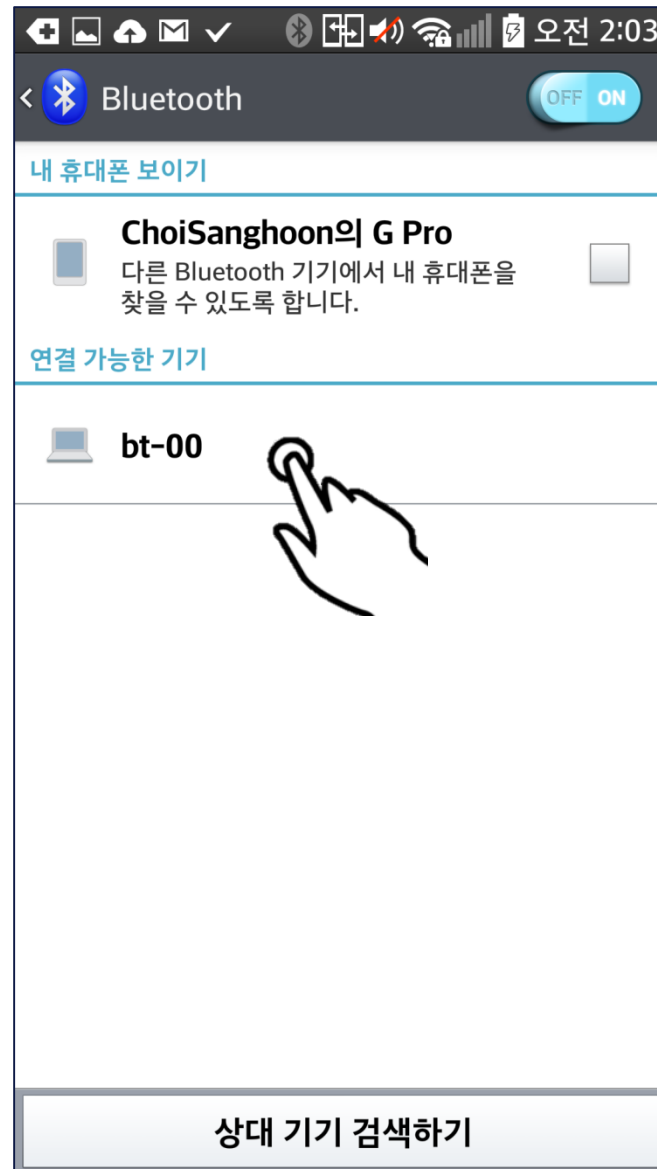
A terminal window titled 'pi@robotcode: ~' with standard window controls. The command `pi@robotcode ~ $ sudo bluez-simple-agent hci0` has been entered and executed. The output is `Agent registered`, followed by a green cursor on a new line.

```
pi@robotcode ~ $ sudo bluez-simple-agent hci0
Agent registered
```


실습3-3 : 블루투스 등록

49

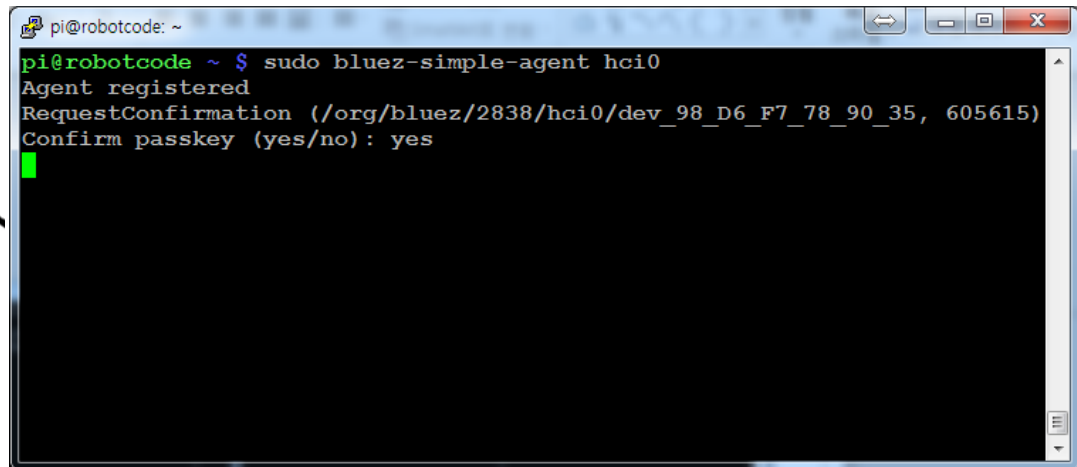
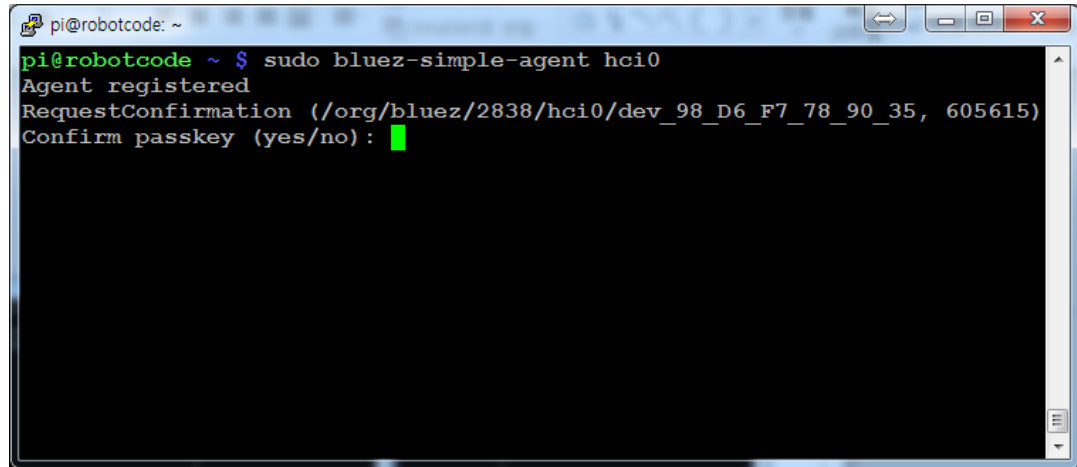
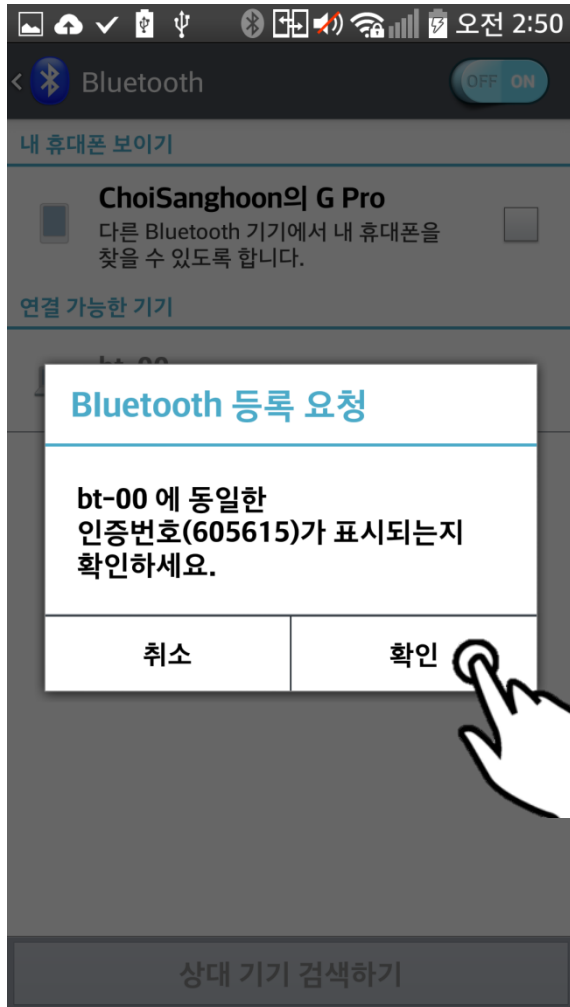
- 안드로이드
 - 블루투스 기기 검색



실습3-4 : 블루투스 등록

50

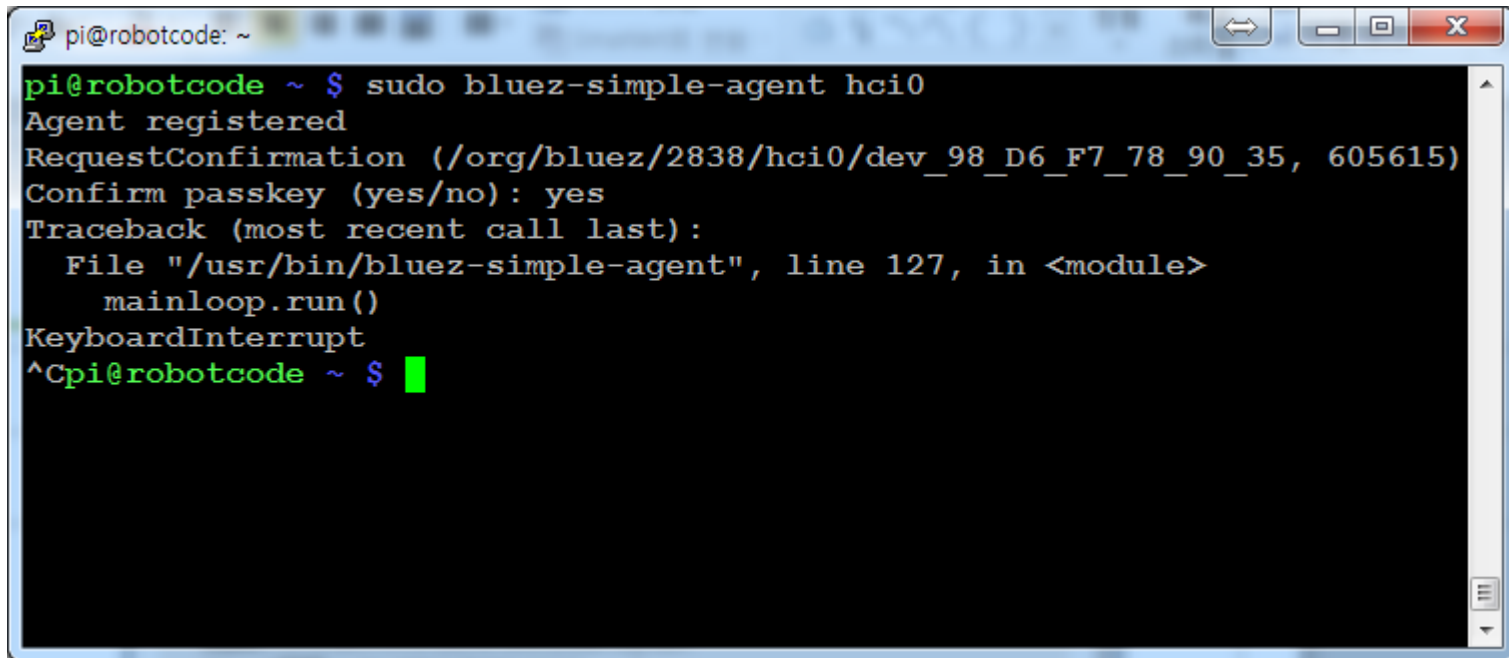
- 안드로이드
 - 등록



실습3-5 : 블루투스 등록

51

- Ctrl + c 빠져나옴



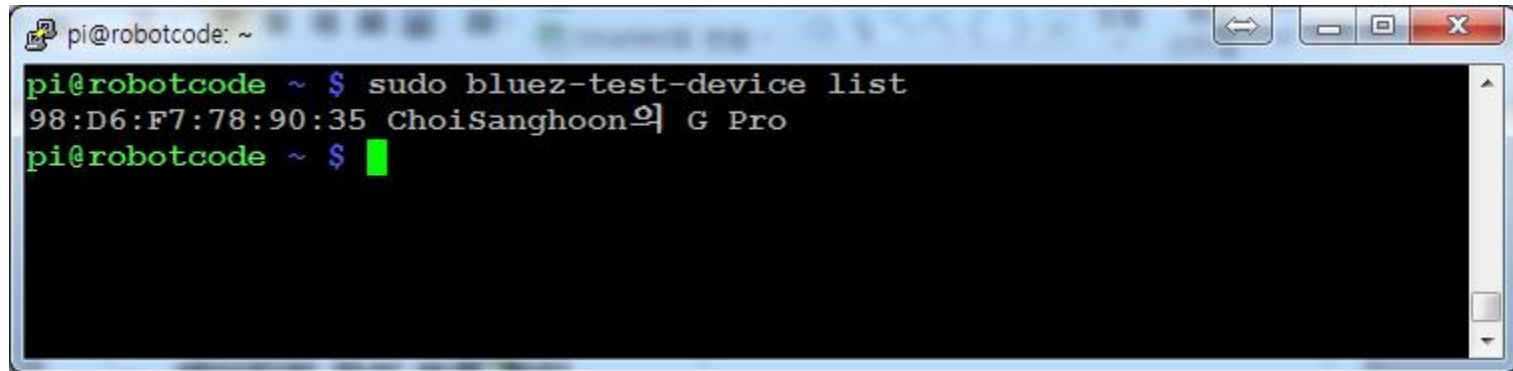
```
pi@robotcode: ~  
pi@robotcode ~ $ sudo bluez-simple-agent hci0  
Agent registered  
RequestConfirmation (/org/bluez/2838/hci0/dev_98_D6_F7_78_90_35, 605615)  
Confirm passkey (yes/no): yes  
Traceback (most recent call last):  
  File "/usr/bin/bluez-simple-agent", line 127, in <module>  
    mainloop.run()  
KeyboardInterrupt  
^Cpi@robotcode ~ $
```

실습3-6 : 블루투스 등록

52

- 등록된 장치 목록 확인

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



```
pi@robotcode: ~  
pi@robotcode ~ $ sudo bluez-test-device list  
98:D6:F7:78:90:35 ChoiSanghoon의 G Pro  
pi@robotcode ~ $
```

- 참고

- 한번 등록되면 재부팅 후에도 정보가 남아있음
- 등록된 장치 삭제

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

실습4-1 : 블루투스 연결

53

- PSCAN ISCAN 활성화

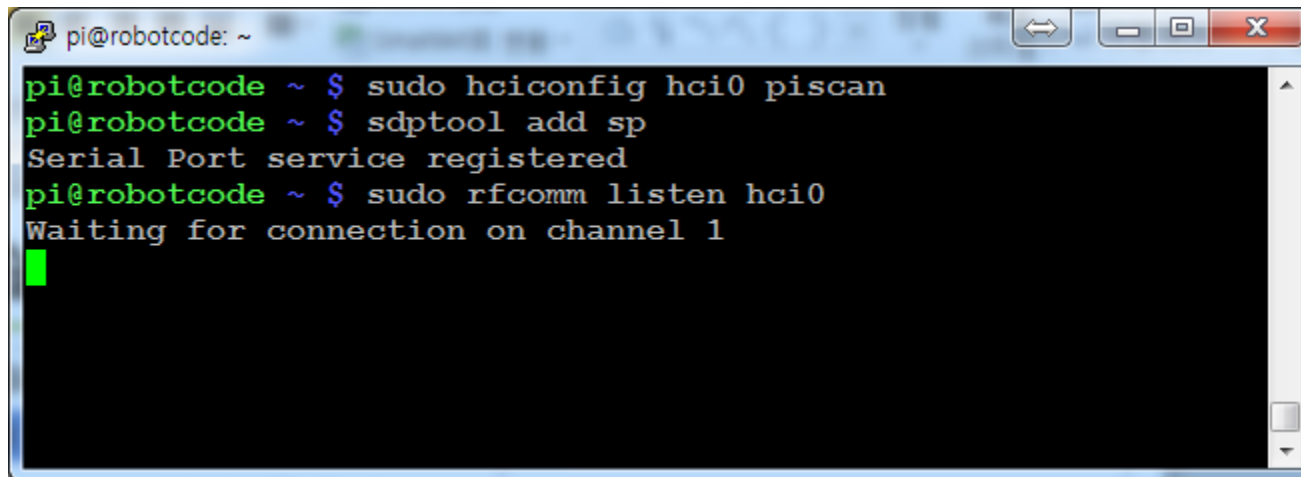
```
$ sudo hciconfig hci0 piscan
```

- Serial Port 프로토콜 추가

```
$ sdptool add sp
```

- rfcomm 리스닝 시작

```
$ sudo rfcomm listen hci0
```



```
pi@robotcode: ~  
pi@robotcode ~ $ sudo hciconfig hci0 piscan  
pi@robotcode ~ $ sdptool add sp  
Serial Port service registered  
pi@robotcode ~ $ sudo rfcomm listen hci0  
Waiting for connection on channel 1  
█
```

실습4-2 : 블루투스 연결

54

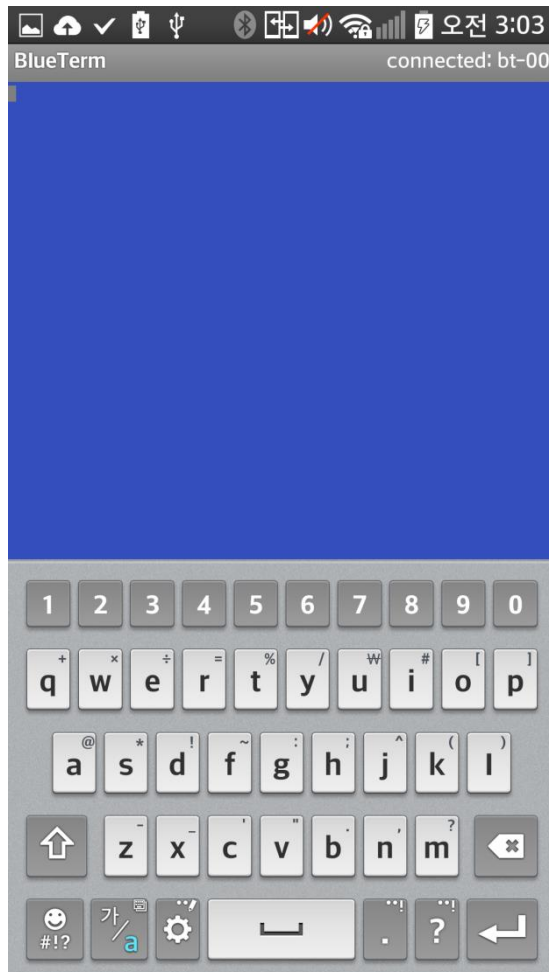
- Bluetooth terminal 연결



실습4-3 : 블루투스 연결

55

- Bluetooth terminal 연결

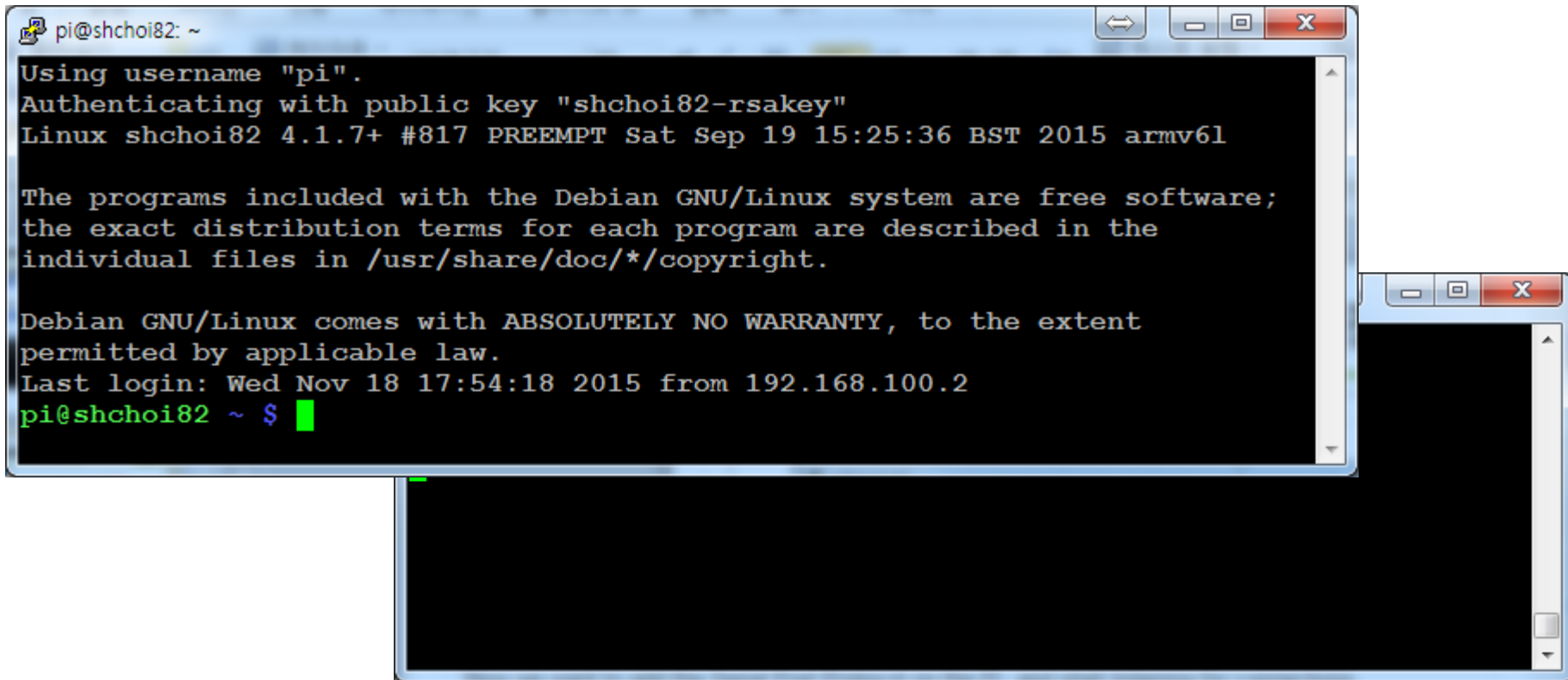


```
pi@robotcode: ~  
pi@robotcode ~ $ sudo hciconfig hci0 piscan  
pi@robotcode ~ $ sdptool add sp  
Serial Port service registered  
pi@robotcode ~ $ sudo rfcomm listen hci0  
Waiting for connection on channel 1  
Connection from 98:D6:F7:78:90:35 to /dev/rfcomm0  
Press CTRL-C for hangup
```

실습4-4 : 블루투스 연결

56

- 새로운 터미널 오픈



```
pi@shchoi82: ~
Using username "pi".
Authenticating with public key "shchoi82-rsa-key"
Linux shchoi82 4.1.7+ #817 PREEMPT Sat Sep 19 15:25:36 BST 2015 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Nov 18 17:54:18 2015 from 192.168.100.2
pi@shchoi82 ~ $
```


실습4-5 : 안드로이드 -> Pi

57

- 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

58

```
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

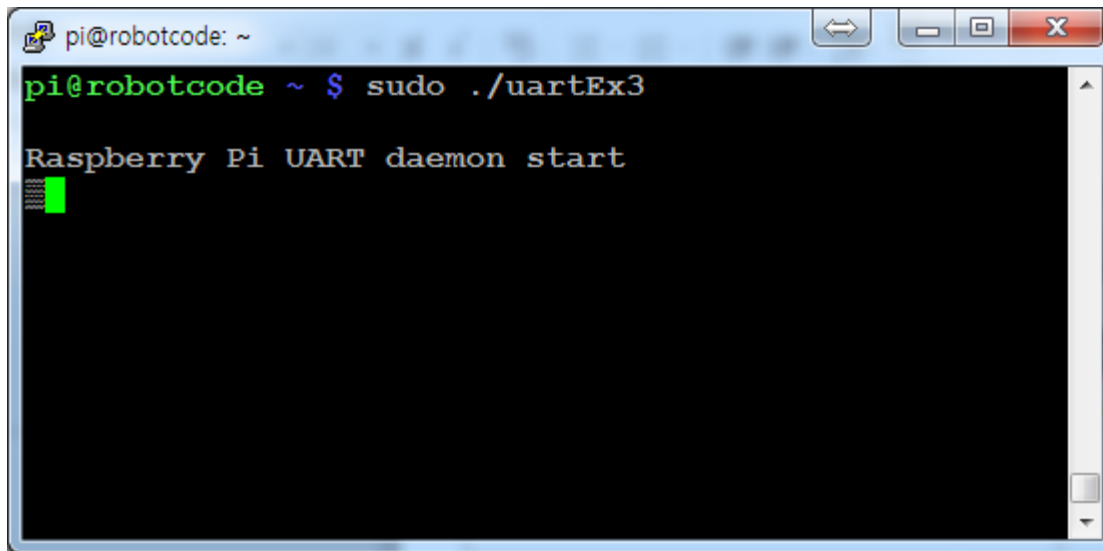
```
$ gcc -Wall -W -lwiringPi uartEx3.c -o uartEx3
```

- 실행

```
$ sudo ./uartEx3
```

실습4-7 : 안드로이드 -> Pi

59



```
pi@robotcode: ~  
pi@robotcode ~ $ sudo ./uartEx3  
Raspberry Pi UART daemon start
```

< Raspberry Pi >



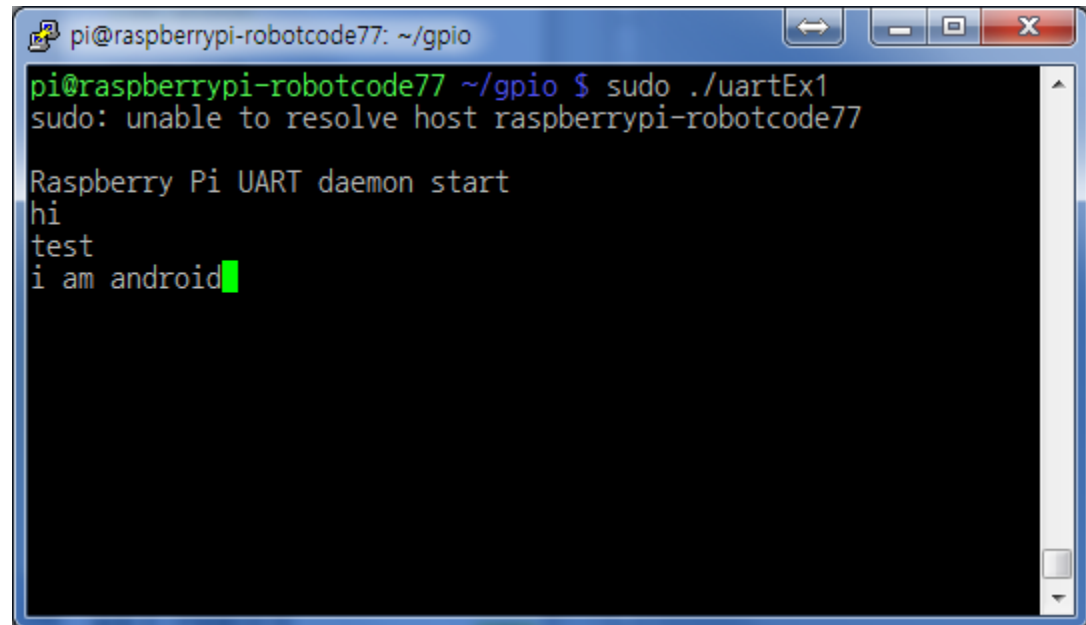
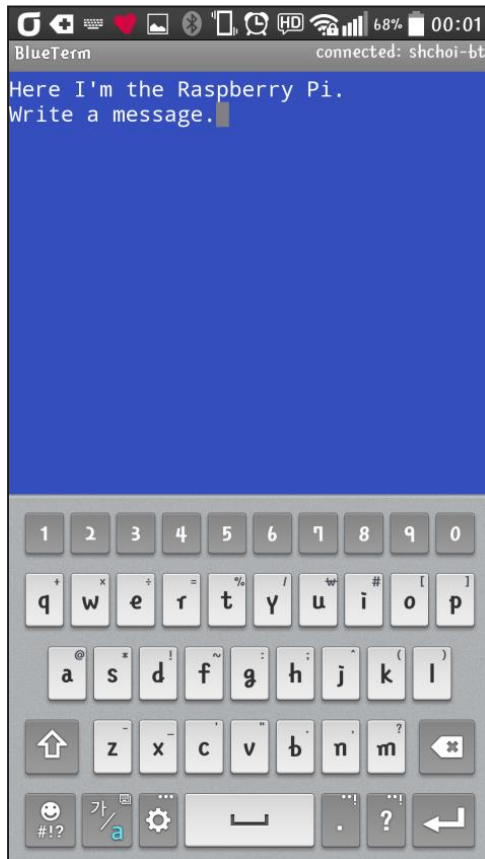
```
BlueTerm  
connected: bt-00  
Here I'm the Raspberry Pi.  
Write a message.
```

실습4-8 : 안드로이드 -> Pi

60

• 통신 테스트

- 안드로이드의 터미널에 텍스트를 입력하면
- RaspberryPi의 터미널창 그대로 출력됨



< Raspberry Pi >

실습 5-1 : Pi -> 안드로이드

61

- 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
    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);
        }
    }
}
```

실습 5-2 : Pi -> 안드로이드

62

```
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

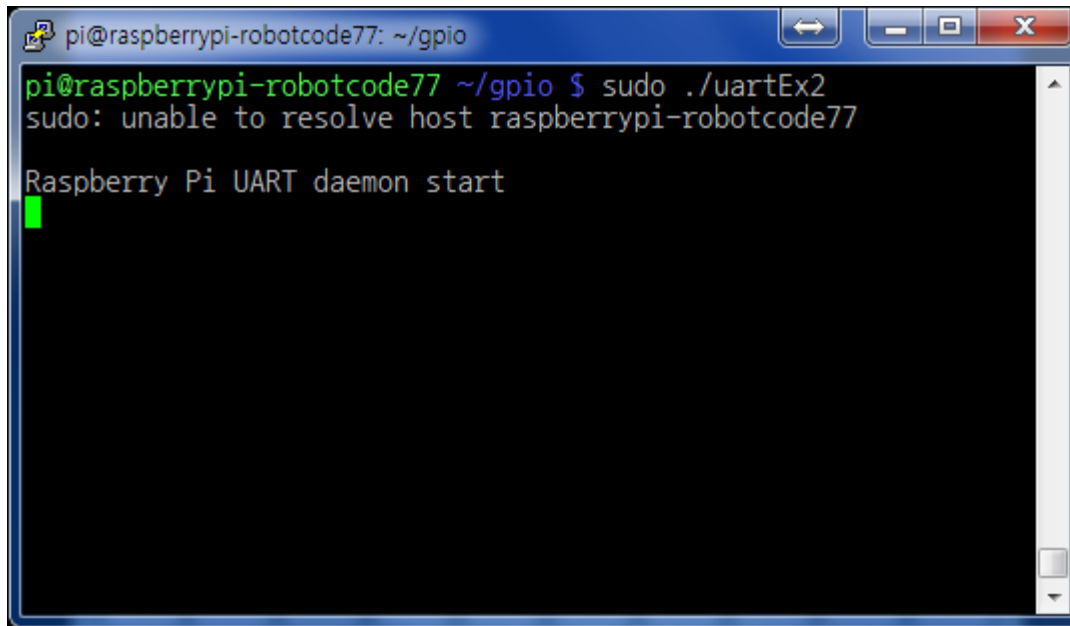
```
$ gcc -Wall -W -lwiringPi uartEx4.c -o uartEx4
```

- 실행

```
$ sudo ./uartEx4
```

실습 5-3 : Pi -> 안드로이드

63



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx2
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
```

< Raspberry Pi >



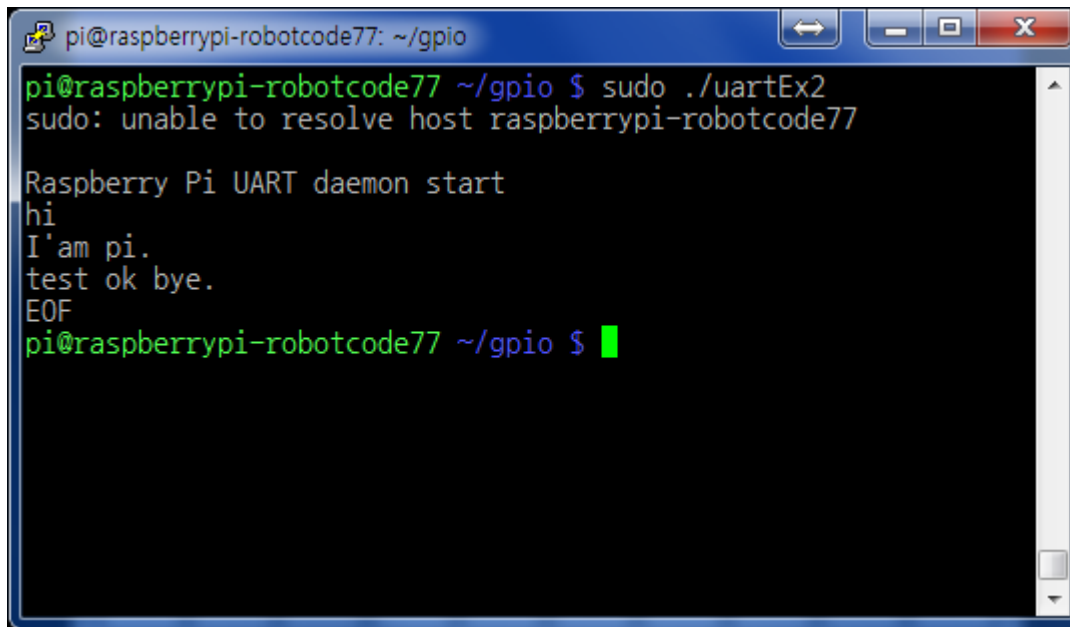
```
BlueTerm connected: shchoi-bt
Here I'm the Raspberry Pi.
I'll send a message.
```

실습 5-4 : Pi -> 안드로이드

64

• 통신 테스트

- RaspberryPi의 터미널에 텍스트 입력하면
- 안드로이드 터미널창에 그대로 출력됨



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx2
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
hi
I'am pi.
test ok bye.
EOF
pi@raspberrypi-robotcode77 ~/gpio $
```

< Raspberry Pi >



```
BlueTerm
connected: shchoi-bt

Here I'm the Raspberry Pi.
I'll send a message.
hi
I'am pi.
test ok bye.
```


미션 1-1 : LED 제어

65

- Bluetooth 통신을 통한 Raspberry Pi GPIO 제어하기
 - 메뉴출력



미션 1-2 : LED 제어

66

- Raspberry Pi GPIO 제어
 - GPIO에 연결된 LED를 ON



미션 1 : LED 제어

67

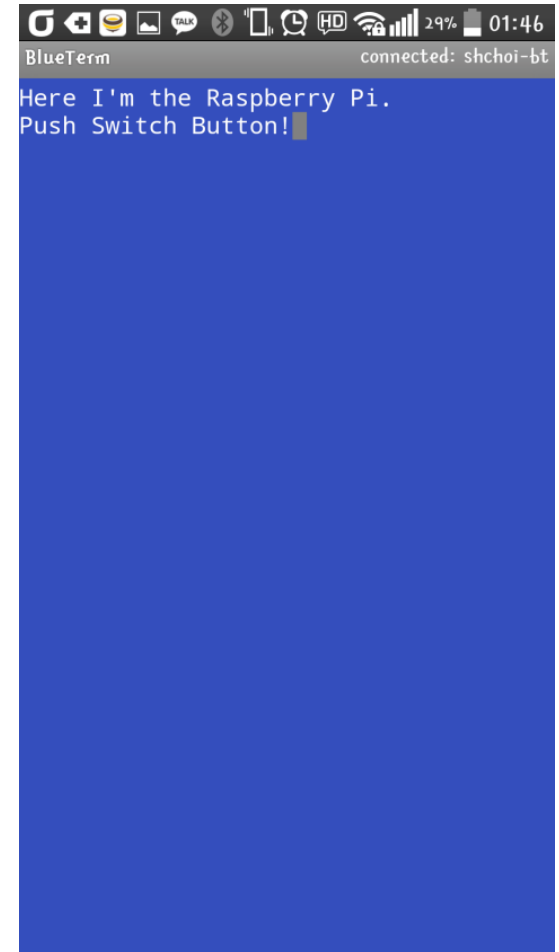
- Raspberry Pi GPIO 제어
 - GPIO에 연결된 LED를 OFF



미션 2-1 : Swtich 모니터링

68

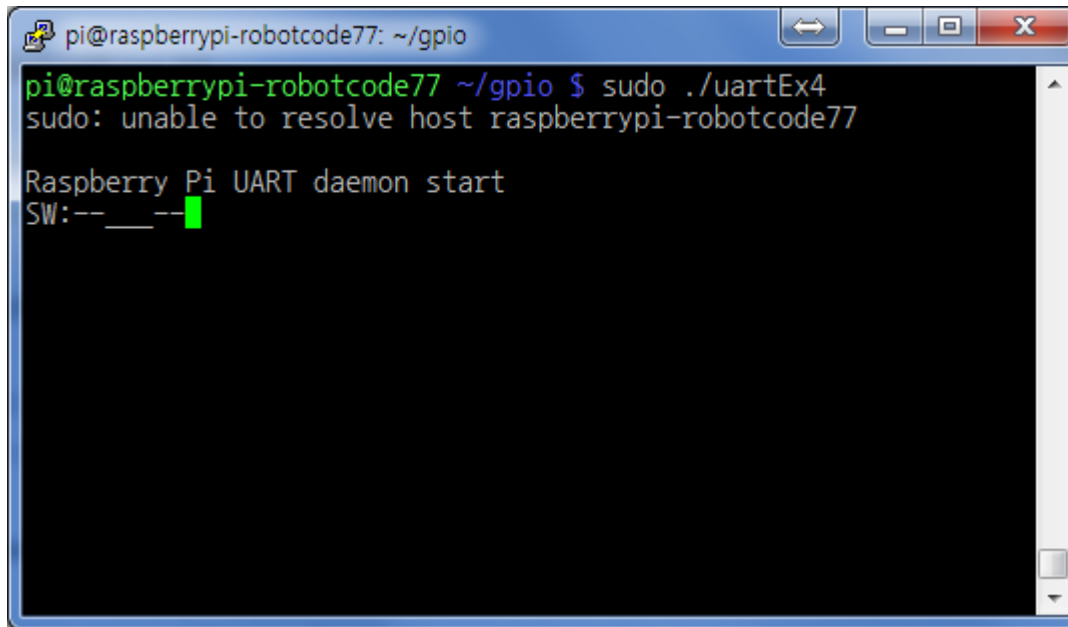
- Raspberry Pi GPIO 제어
 - GPIO에 연결된 Swtich 상태 모니터링



미션 2-2 : Swtich 모니터링

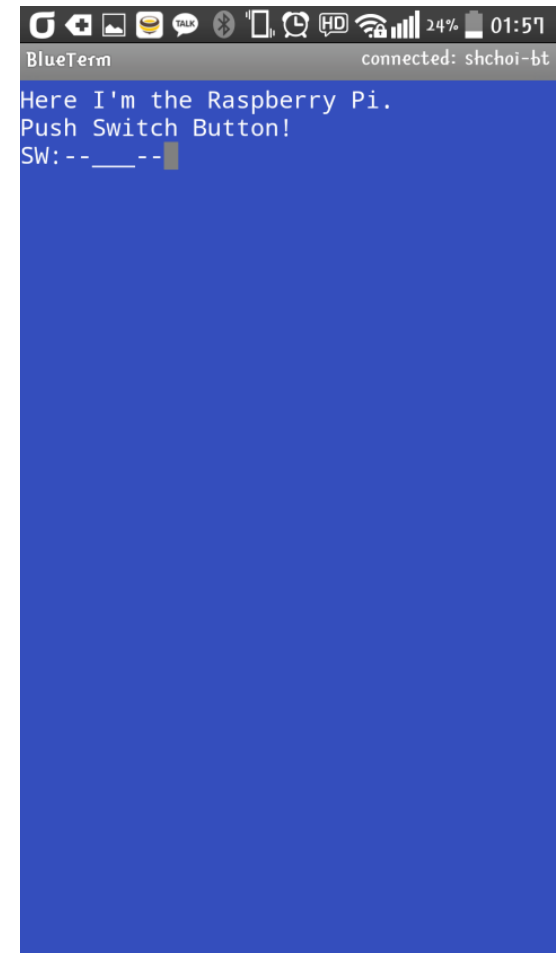
69

- Raspberry Pi GPIO 제어
 - GPIO에 연결된 Swtich 상태 모니터링



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx4
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
SW:--__--
```

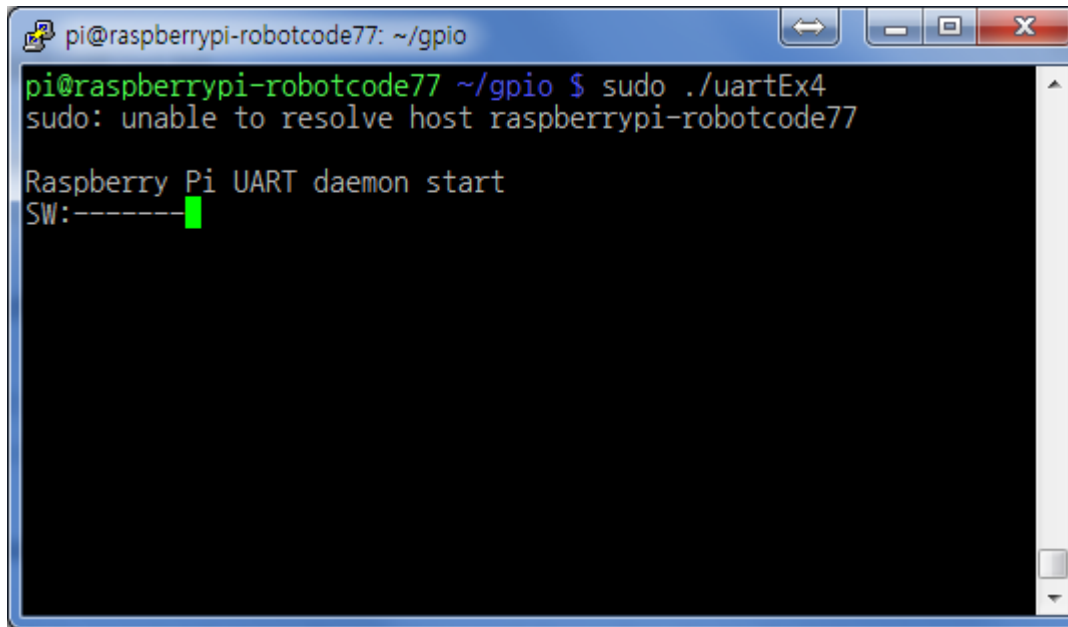


```
BlueTerm connected: shchoi-bt
Here I'm the Raspberry Pi.
Push Switch Button!
SW:--__--
```

미션 2-3 : Swtich 모니터링

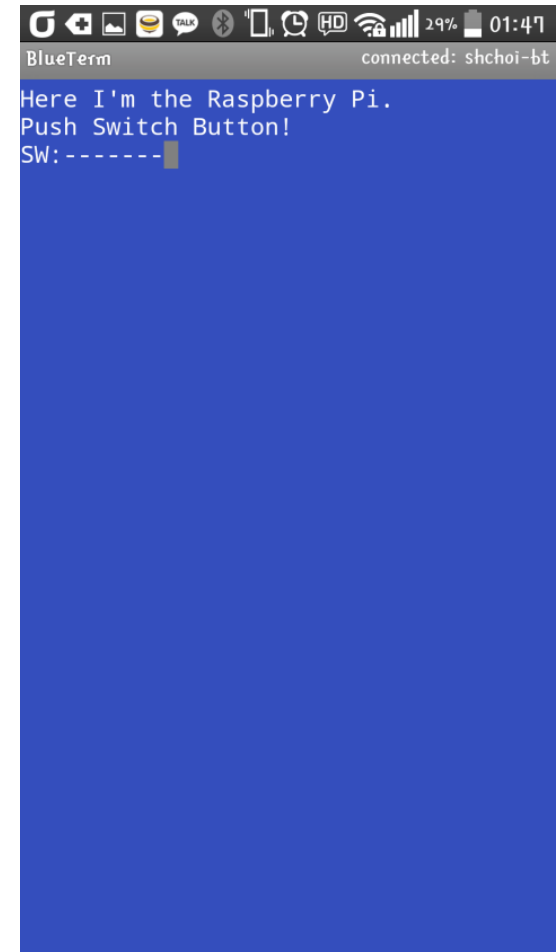
70

- Raspberry Pi GPIO 제어
 - GPIO에 연결된 Swtich 상태 모니터링



```
pi@raspberrypi-robotcode77: ~/gpio
pi@raspberrypi-robotcode77 ~/gpio $ sudo ./uartEx4
sudo: unable to resolve host raspberrypi-robotcode77

Raspberry Pi UART daemon start
SW:-----█
```



```
BlueTerm connected: shchoi-bt
Here I'm the Raspberry Pi.
Push Switch Button!
SW:-----█
```

미션 3

71

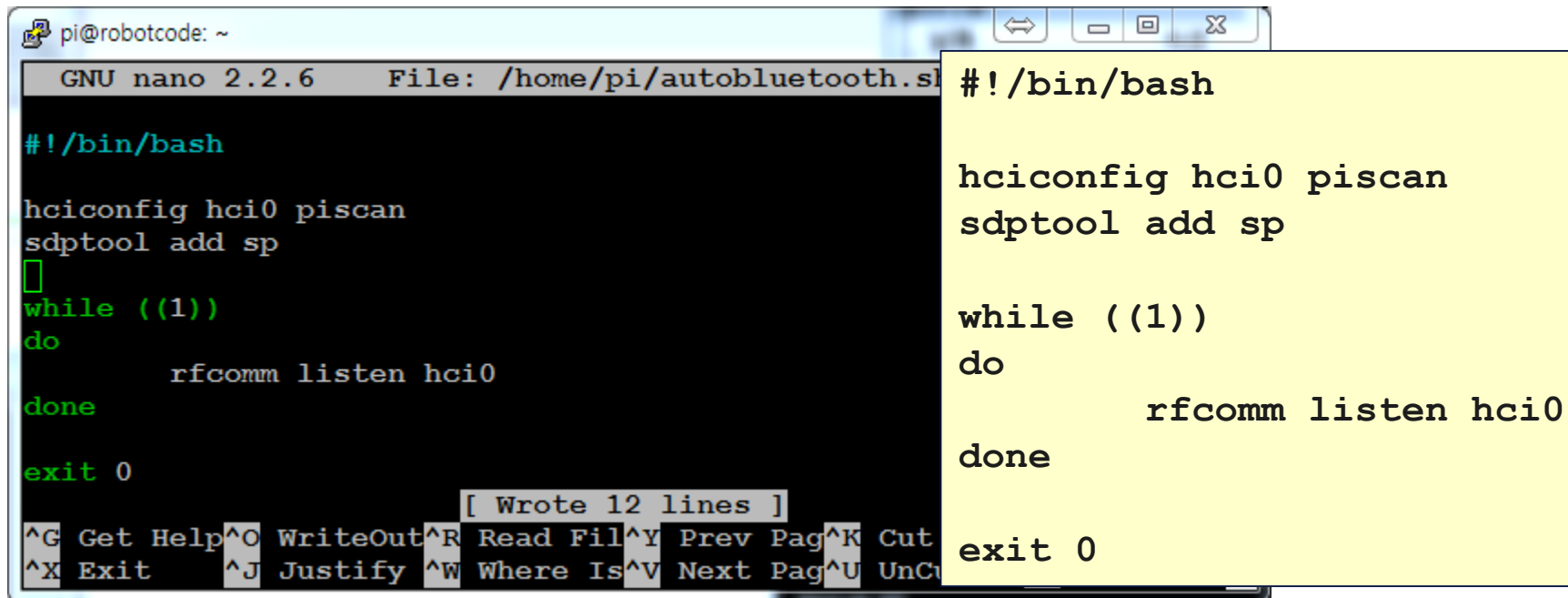
- 현재 작업 디렉토리의 파일 목록을 안드로이드 터미널 화면에 출력하시오.

실습6-1 : 부팅시 자동 설정

72

- 부팅시 자동 설정

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



```
GNU nano 2.2.6 File: /home/pi/autoblueetooth.sh

#!/bin/bash

hciconfig hci0 piscan
sdptool add sp

while ((1))
do
    rfcomm listen hci0
done

exit 0

[ Wrote 12 lines ]

^G Get Help ^O WriteOut ^R Read Fil ^Y Prev Pag ^K Cut
^X Exit ^J Justify ^W Where Is ^V Next Pag ^U UnC
```

```
#!/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/autoblueetooth.sh
```


실습6-2 : 부팅시 자동 설정

73

- 부팅시 자동 설정
 - /etc/rc.local 파일에 추가

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

```

pi@robotcode: ~
GNU nano 2.2.6      File: /etc/rc.local      Modified

# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

# 블루투스 초기화(SP, piscan, listen)
/home/pi/autoblutetooth.sh&

# 자동으로 실행할 프로그램명 지정
/home/pi/autoexec

exit 0

^G Get Help ^O WriteOut ^R Read Fil ^Y Prev Pag ^K Cut Text ^C Cur Pos
^X Exit     ^J Justify  ^W Where Is ^V Next Pag ^U UnCut Te ^T To Spell
  
```

```
/home/pi/autoblutetooth.sh&
```

```
/home/pi/autoexec
```

미션4

74

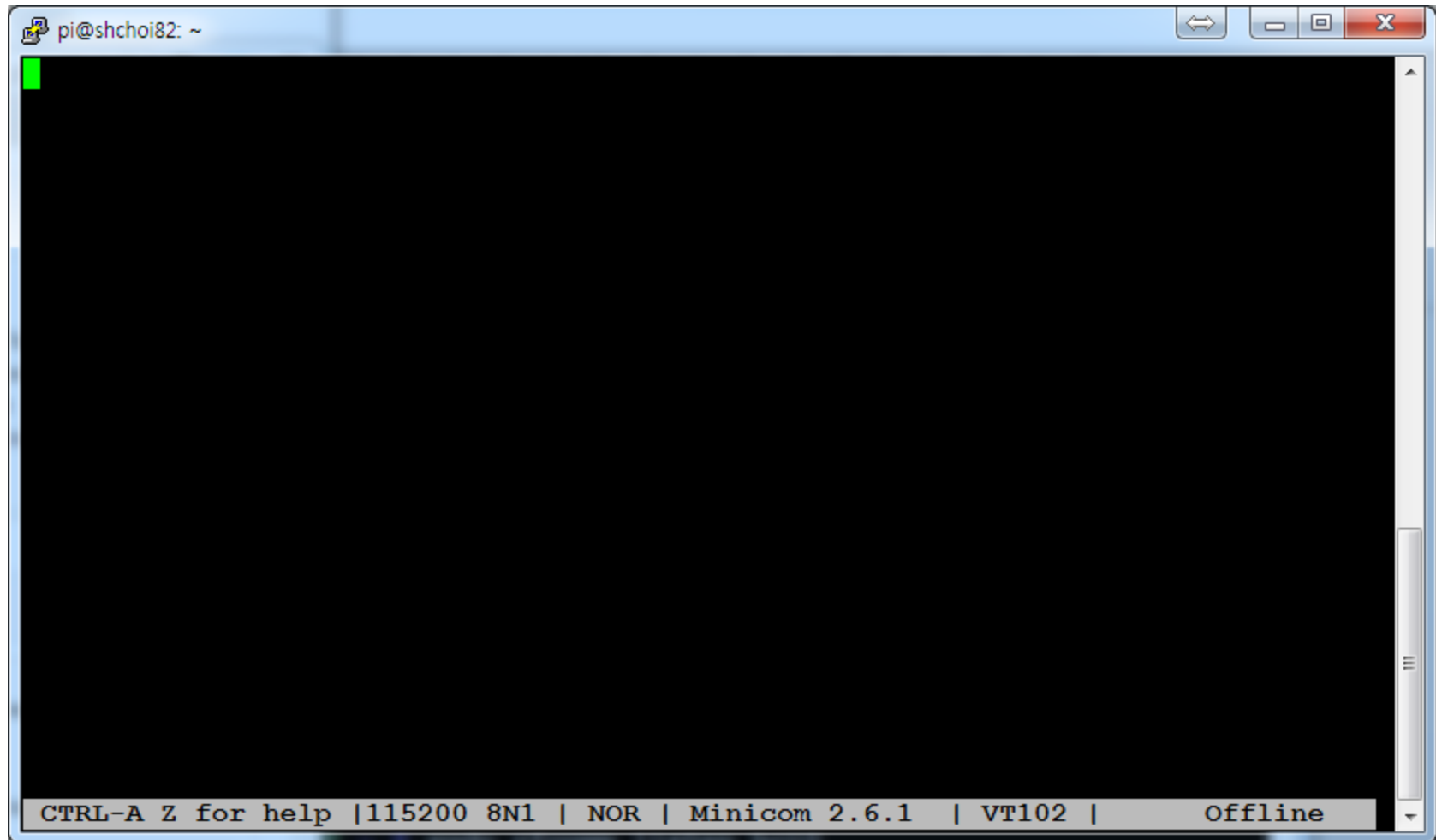
- **미션1에서 작성한 프로그램이 부팅 시 자동으로 실행될 수 있도록 설정 하시오.**

참고 : minicom

75

- 라즈베리파이 minicom 실행

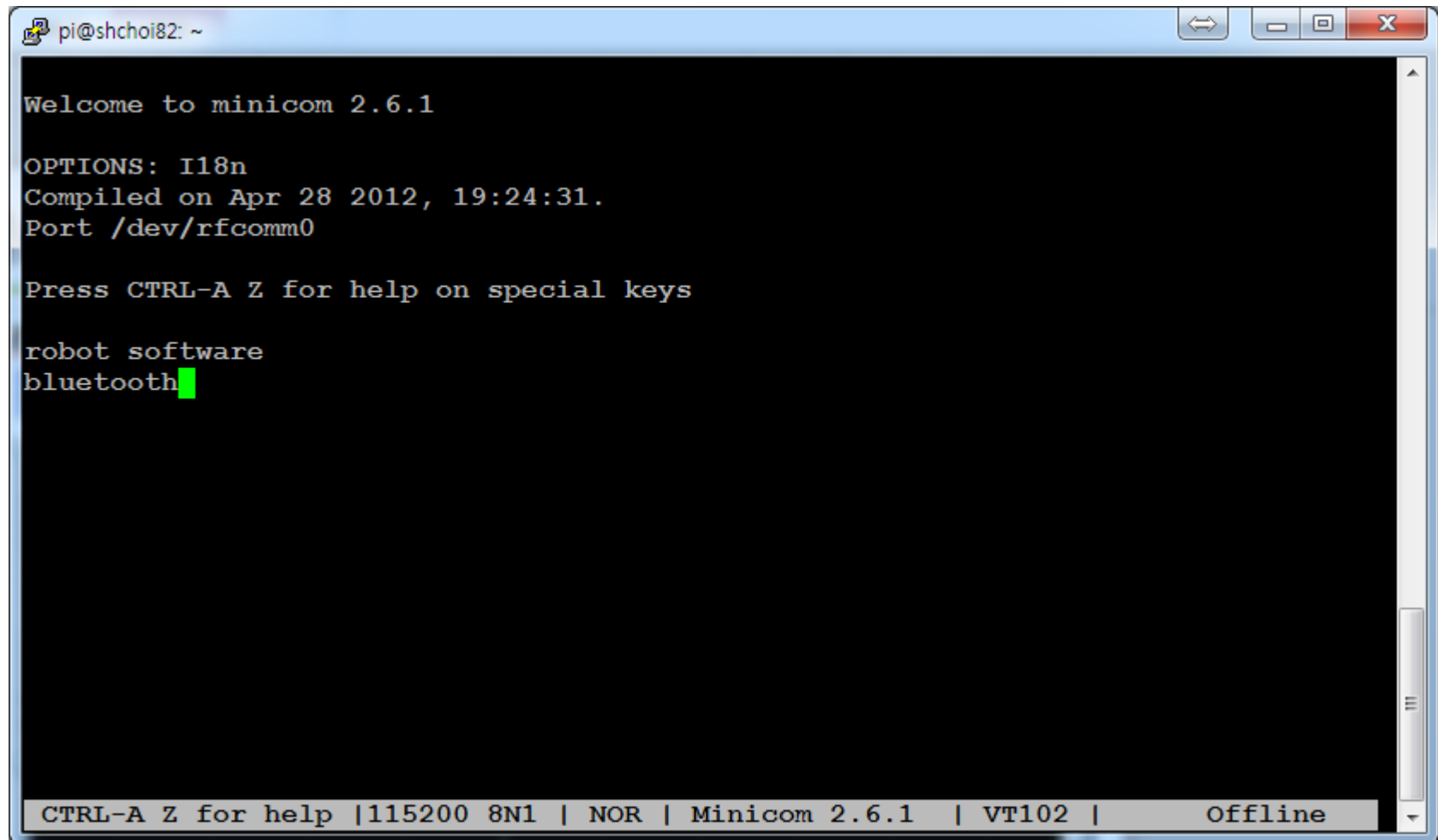
```
$ minicom -D /dev/rfcomm0
```



참고 : minicom

76

- 안드로이드 bluetooth terminal 앱에서 키보드 입력



The image shows a terminal window titled "pi@shchoi82: ~". The terminal output is as follows:

```
Welcome to minicom 2.6.1

OPTIONS: I18n
Compiled on Apr 28 2012, 19:24:31.
Port /dev/rfcomm0

Press CTRL-A Z for help on special keys

robot software
bluetooth█
```

The status bar at the bottom of the terminal window displays: "CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.6.1 | VT102 | Offline".