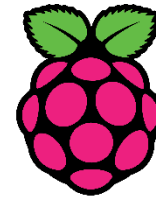


Sensor Programming

센서 프로그래밍



RaspberryPi



RASPBIAN

블루투스 기반 하드웨어 제어

Sensor Programming

센서 프로그래밍

블루투스 기반 하드웨어 제어
- 블루투스 소개



RaspberryPi



RASPBIAN

블루투스 소개

■ 블루투스 소개

- 근거리 무선통신을 위한 산업표준
- 1999년 5월 20일에 공식적으로 발표 by Jim Kardach
- 1998년 'Bluetooth Special Interest Group' 블루투스에 관한 동맹 결성
 - 참여기업 : 에릭슨, 인텔, 레노버, 마이크로소프트, 애플, 노키아, 도시바, IBM
- 전 세계적으로 이용할 수 있는 무선 주파수 (ISM 대역인 2.45GHz를 사용)
- 저렴한 칩 가격과 저전력 구현



블루투스 소개

■ 버전별 특징

- 블루투스 1.1
 - 가장 초기에 나온 1.0부터 1.1을 거쳐 1.2까지 개선됨, 전송속도 723kbps
- 블루투스 2.0 + EDR
 - EDR (Enhanced Data Rate) , 전송속도 3Mbps
- 블루투스 3.0 + HS
 - 2009년 4월 21일에 발표
 - 전송 속도 최대 속도 24Mbps로 향상
- 블루투스 4.0
 - 블루투스 저전력 (Bluetooth Low Energy : BLE) 기술을 지원 (4.0)
 - 전력소모를 최소화하고 배터리 수명을 연장하는데 중점
- 블루투스 4.1
 - 공존성(Coexistence) 향상 (LTE 무선통신과의 간섭 현상 최소화)
 - Better Connections (이동 중인 연결 장치 간에 유효 거리 범위 내 자동 연결)
 - 사물인터넷(The Internet of Things)을 위한 새로운 IPv6 사용 표준 포함

블루투스 소개

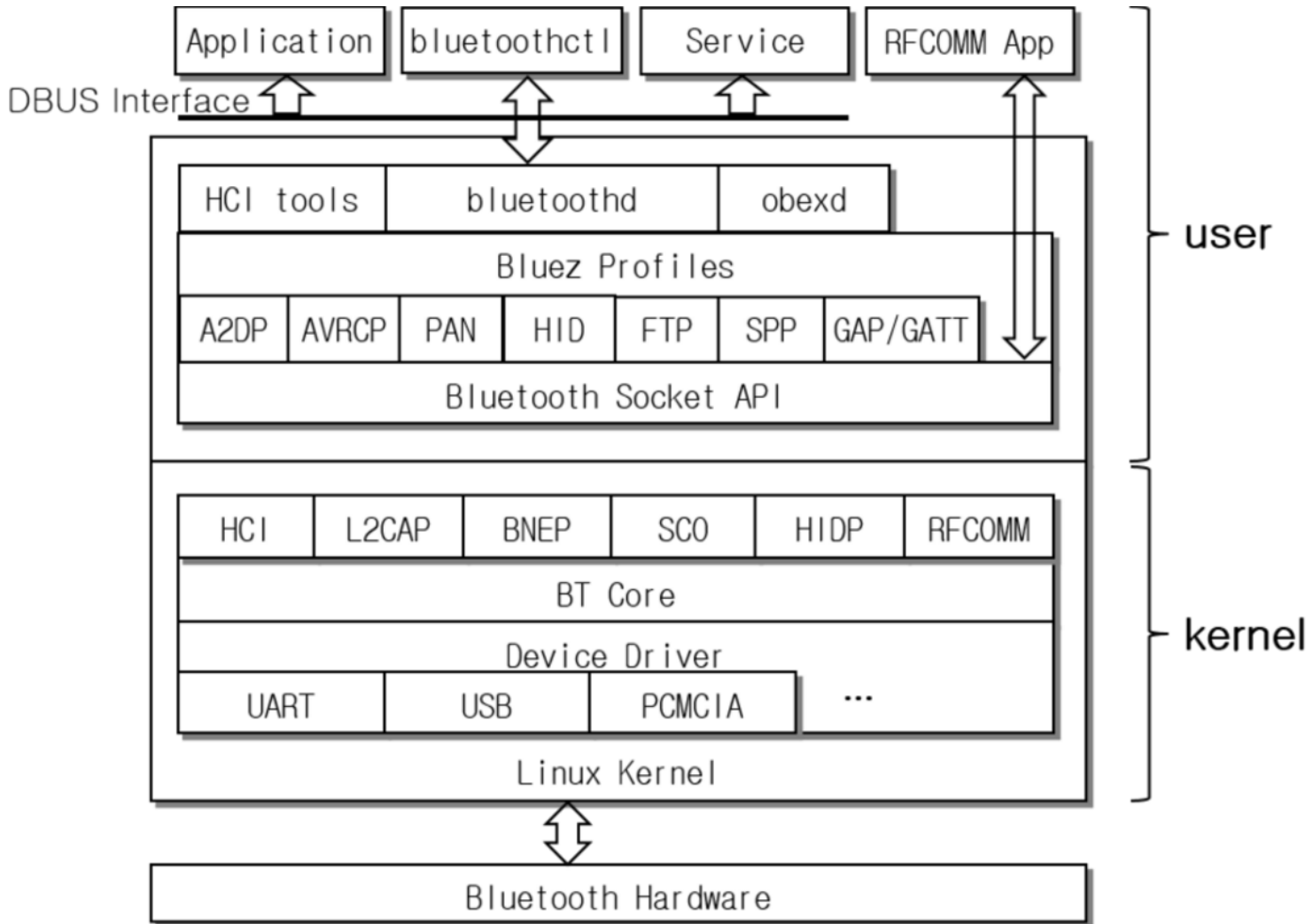
■ 블루투스 프로파일

- 장치간의 통신을 위해 논리적 특성들을 취합해 분류한 응용 계층의 S/W
 - 오디오 전송, 네트워크 사용, 파일 전송, 심박수 전송
 - 직렬 포트 데이터 전송, 프로토콜을 사용하기 위한 특성 등등.

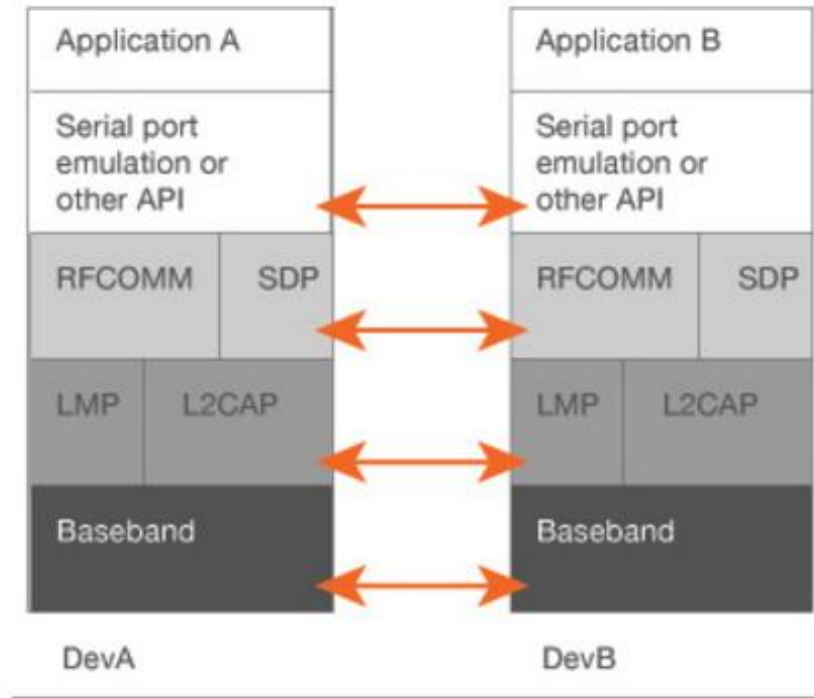
■ 블루투스 프로토콜

- 데이터를 송수신하기 위한 약속으로 성능을 결정하는 핵심 역할
- 연결된 디바이스에서 어떤 서비스가 가능한지 서비스의 특징에 관한 정보를 교환
- 베이스 밴드 소프트웨어를 추상화 하기 위한 인터페이스를 제공 (HCI : 커널에 구현)
- 상위 계층(프로파일)과의 인터페이스를 이용하여 HCI를 이용하여 연결 및 해제를 수행
- 베이스밴드 소프트웨어
 - 블루투스 모듈과 관련된 펌웨어 형태로 칩 제조회사에서 제공

블루투스 소개 : 리눅스 블루투스 프로토콜 스택



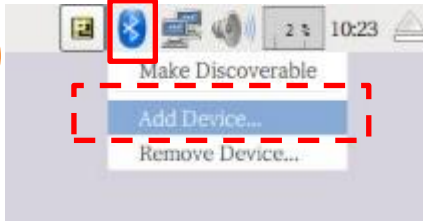
SPP 프로파일



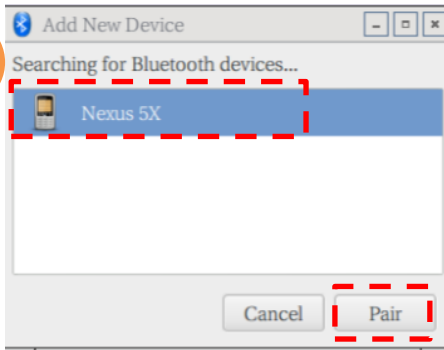
- SPP(Serial Port Profile)
 - SPP는 두 블루투스 장치를 직렬 포트 방식으로 연결
 - 연결을 기다리는 서버 장치는 SDP를 통해 SPP 서비스를 알림
 - 이를 감지한 클라이언트가 접속을 시도하면 연결됨
 - 이후 RFCOMM 프로토콜을 통해 데이터 전송

라즈베리파이 3 스마트폰 블루투스 패어링

2



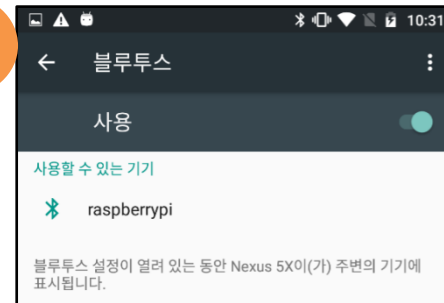
3



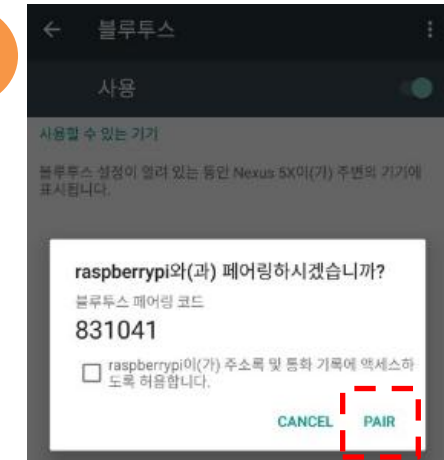
5



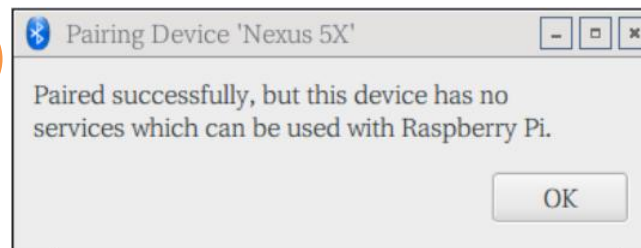
1



4

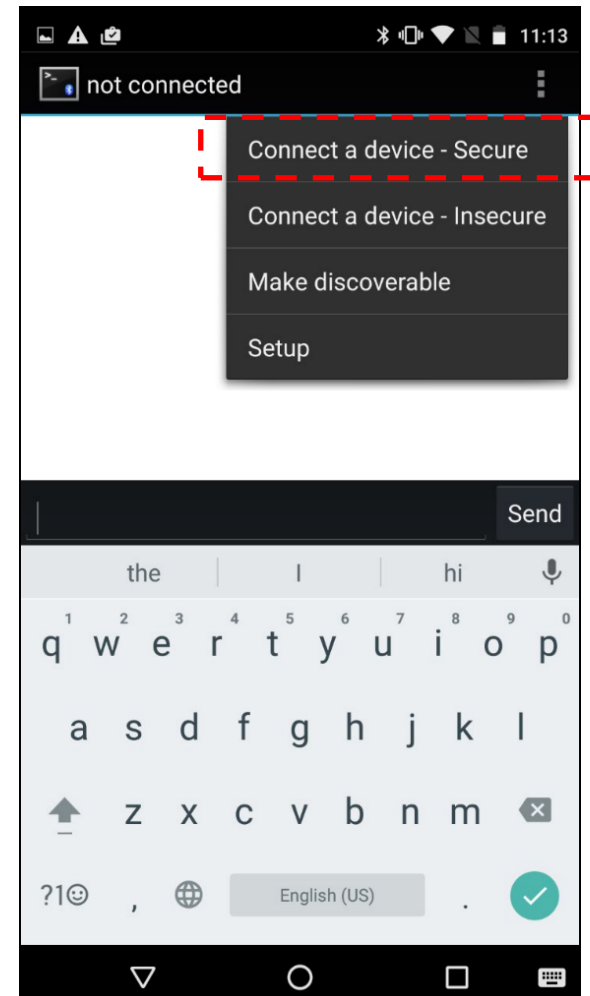
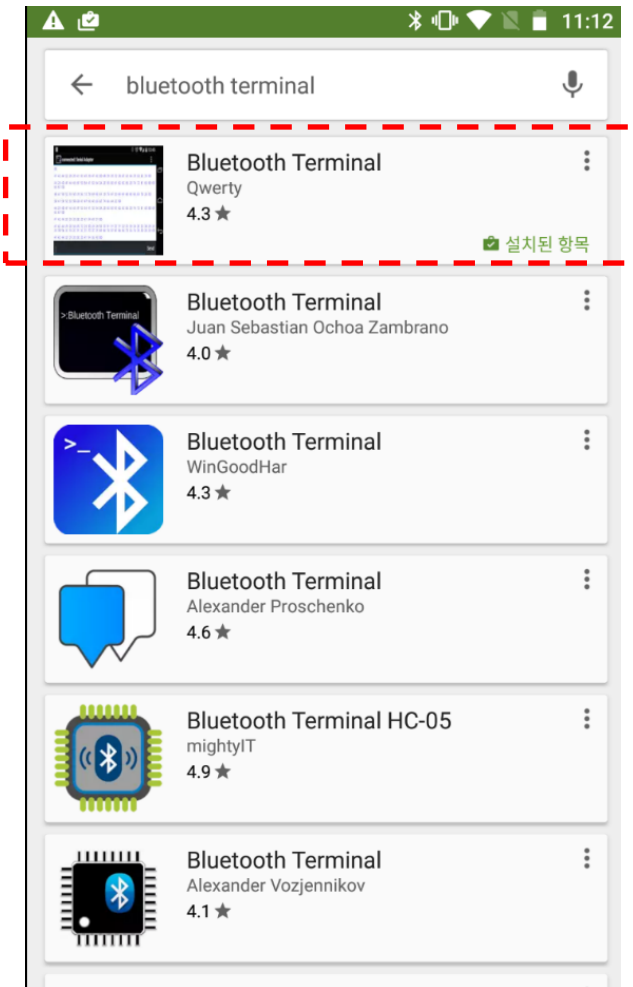


6

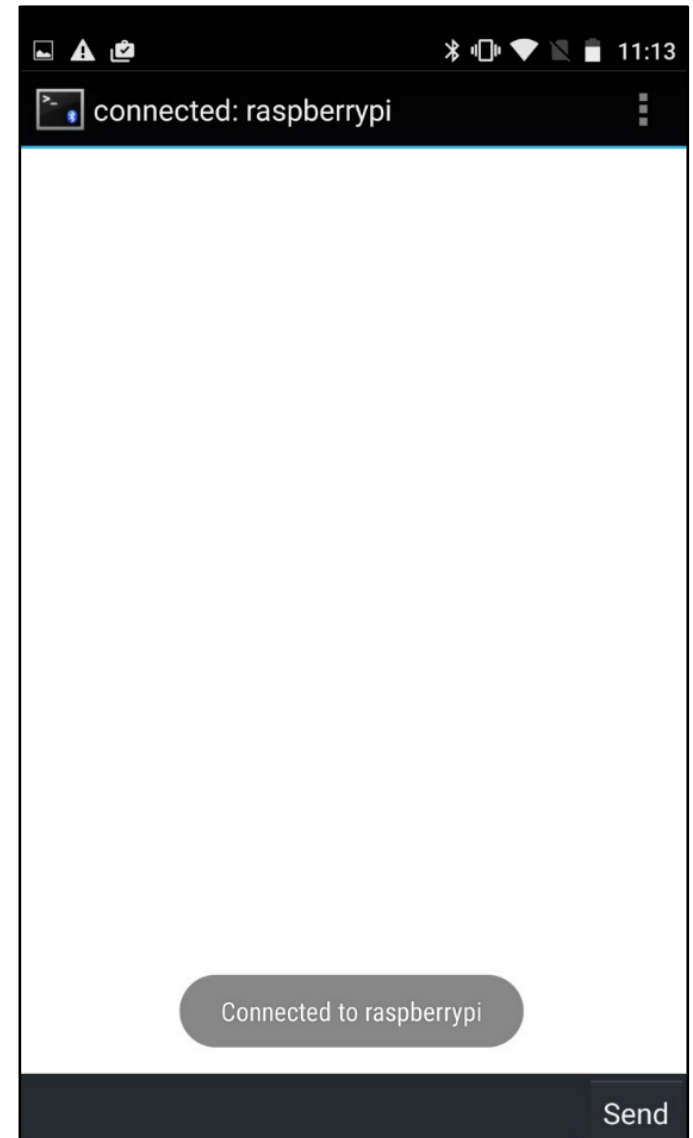
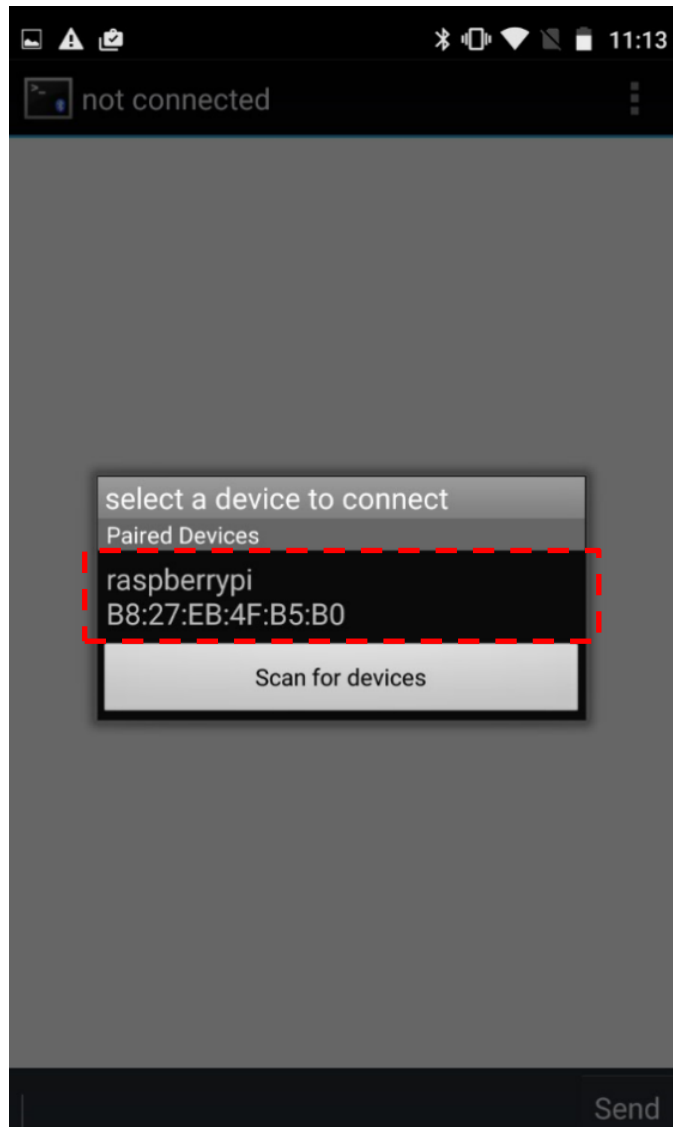


라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 클라이언트

- 안드로이드 스마트폰에 play 스토어에서 bluetooth terminal 설치



라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 클라이언트



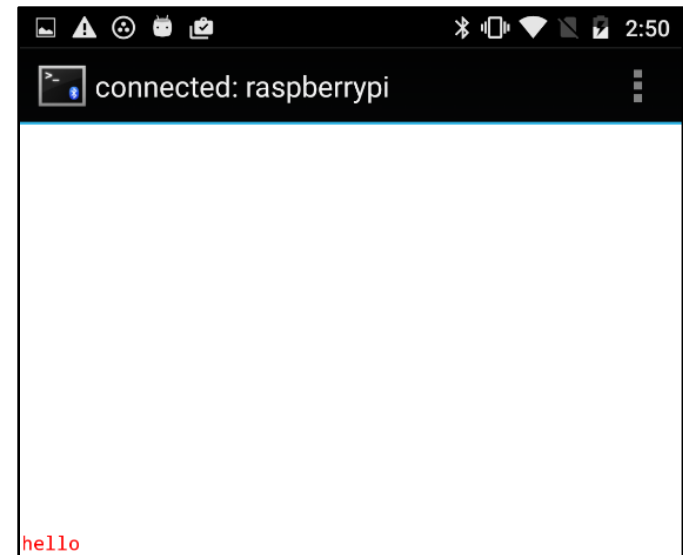
라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 전송

스마트폰과 iCORE-SDP가 SPP로 연결된 상태이므로 iCORE-SDP에는 자동으로 /dev/rfcomm0가 만들어진다.

```
tea@planx: ~  
tea@planx:~ $ ls -l /dev/rfcomm0  
crw-rw---- 1 root dialout 216, 0 7월 17 12:38 /dev/rfcomm0  
tea@planx:~ $ _
```

Pi3에서 아래 명령을 실행한다음
스마트폰에서 데이버를 보내면 아래와
같이 수신 데이터를 볼수 있을 것이다

```
tea@planx:~ $ sudo cat /dev/rfcomm0  
hello
```



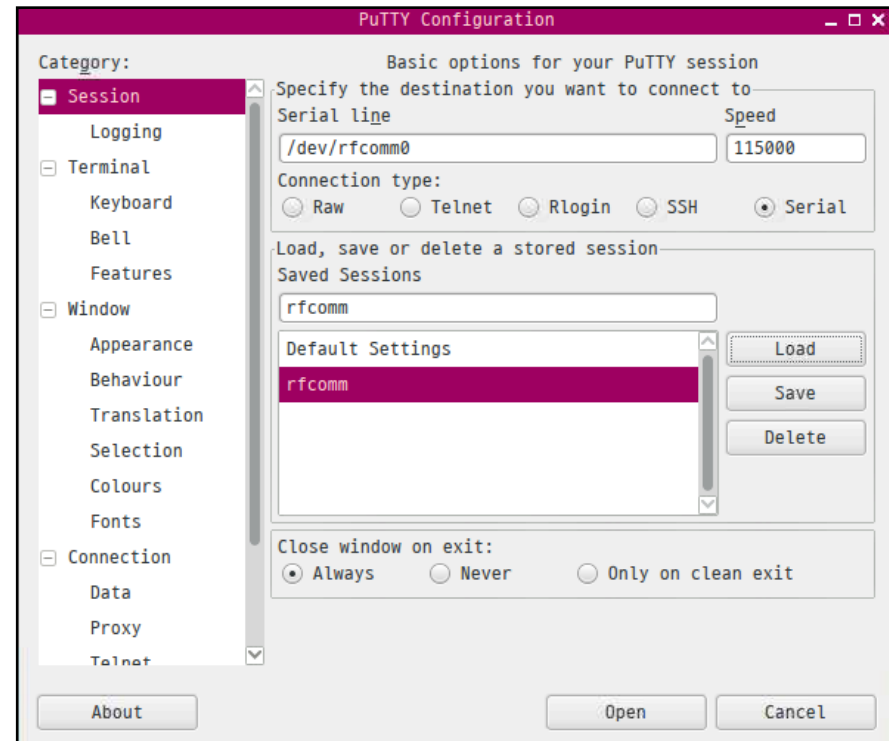
라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 전송

iCORE-SDP에서도 테스트를 위해 PuTTY를 실행하는데 /dev/rfcomm0는 root 사용자만 접근할 수 있으므로 터미널에서 sudo putty 명령을 사용한다.

```
tea@planx:~ $ sudo putty
```

PuTTY가 실행되면 'Select a serial line'만 /dev/rfcomm0이고 Speed(baud)와 Flow control은 115200, None으로 설정한 후 Open으로 연결한다

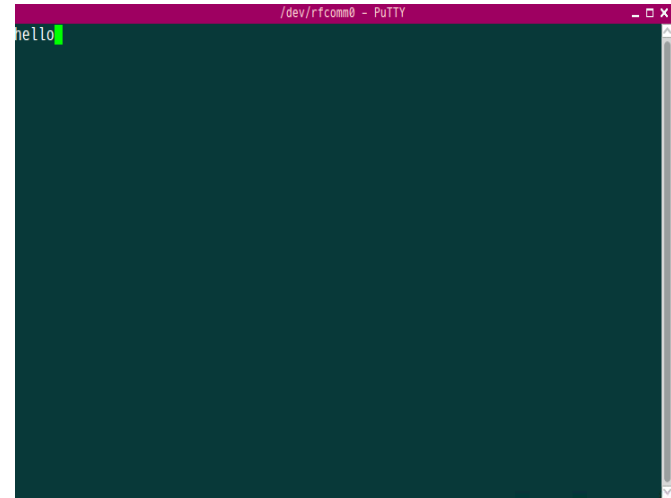
당사 패키지에는 이미 설정되어 있는 rfcomm 을 선택하고 [Load]→[Open] 한다



라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 전송

정상적으로 연결되면 블루투스를 시리얼 장치처럼 사용해 데이터를 교환할 수 있다. PuTTY는 에코 기능을 사용하지 않으므로 자신이 입력한 글자는 자신의 위치에 표시되지 않는다..

주의할 점은 연결이 끊어져도 해당 파일은 자동으로 제거되지 않으며, 다시 외부 장치와 SPP로 연결하려면 반드시 /dev/rfcomm0를 제거해야 한다. systemctl status rfcomm 명령으로 확인 PID를 kill 명령으로 제거하면 /dev/rfcomm0도 함께 제거되고 서비스는 자동으로 재 시작하면서 새로운 연결을 기다리게 된다



```
tea@planx: ~/python_lab/suk_python_ex/Bluetooth
tea@planx:~/python_lab/suk_python_ex/Bluetooth $ systemctl status rfcomm
● rfcomm.service - RFCOMM service
   Loaded: loaded (/lib/systemd/system/rfcomm.service; enabled)
   Active: active (running) since 월 2017-07-17 12:15:50 KST; 51min ago
   Main PID: 1165 (rfcomm)
   CGroup: /system.slice/rfcomm.service
           └─1165 /usr/bin/rfcomm watch hci0
tea@planx:~/python_lab/suk_python_ex/Bluetooth $ sudo kill -9 1165
tea@planx:~/python_lab/suk_python_ex/Bluetooth $ systemctl status rfcomm
● rfcomm.service - RFCOMM service
   Loaded: loaded (/lib/systemd/system/rfcomm.service; enabled)
   Active: active (running) since 월 2017-07-17 13:08:48 KST; 6s ago
```

라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 전송

```
from bluetooth import *  
  
server_sock=BluetoothSocket( RFCOMM )  
  
server_sock.bind(("",PORT_ANY))  
  
server_sock.listen(1)  
  
port = server_sock.getsockname()[1]  
  
uuid = "00001800-0000-1000-8000-00805F9B34FB"  
  
advertise_service( server_sock, "raspberrypi",  
                   service_id = uuid,  
                   service_classes = [ uuid, SERIAL_PORT_CLASS ],  
                   profiles = [ SERIAL_PORT_PROFILE ],  
                   )
```

라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 전송

```
print("Waiting for connection on RFCOMM channel %d" % port)
client_sock, client_info = server_sock.accept()
print("Accepted connection from ", client_info)
try:
    while True:
        data = client_sock.recv(1024)
        if len(data) == 0: break
        print("received [%s]" % data)
except IOError:
    pass
print("disconnected")
client_sock.close()
server_sock.close()
print("all done")
```

라즈베리파이 3 스마트폰 블루투스 연결 (SPP) : 전송

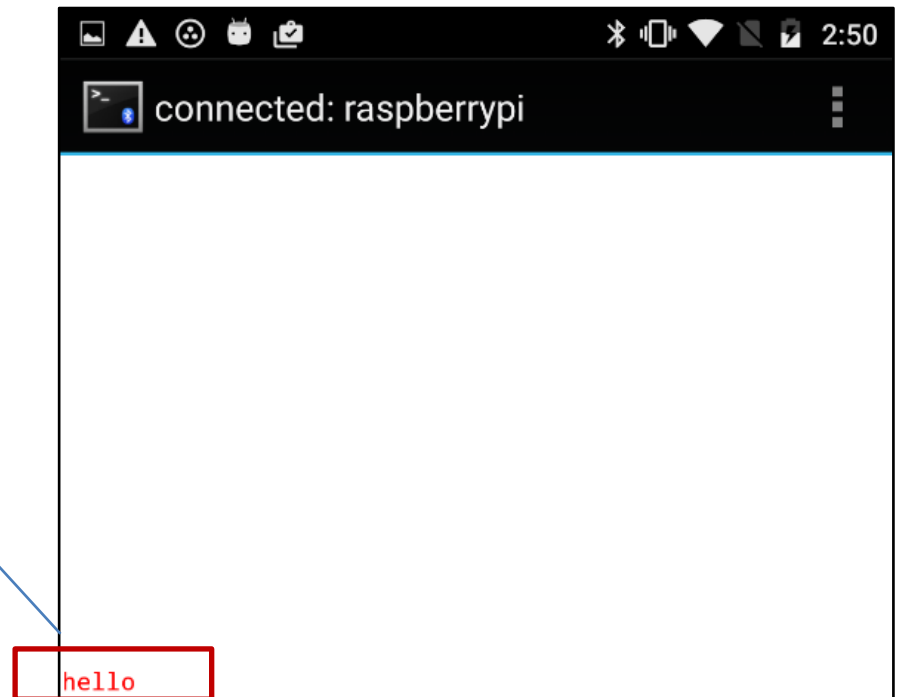
```
pi@raspberrypi:~ $ cd pybluez/examples/simple/
```

```
pi@raspberrypi:~/pybluez/examples/simple $ sudo python rfcomm-server.py
```

```
Waiting for connection on RFCOMM channel 1
```

```
('Accepted connection from ', ('78:F8:82:53:A4:5E', 1))
```

```
received [hello]
```



블루투스 led 실습

```
pin = 14
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(pin, GPIO.OUT)
...
while True:
    bytedata = client_sock.recv(1024)

    data = bytedata.decode()

    print("received [%s]" % data)

    if len(data) == 0:
        break
    elif (data == "1"):
        GPIO.output(pin, GPIO.HIGH)
    elif (data == "0"):
        GPIO.output(pin, GPIO.LOW)
```

Sensor Programming

센서 프로그래밍

Thank you and Question?



RaspberryPi



RASPBIAN