

라즈베리파이2와 BEACON의 만남

IoT, Bluetooth, BLE, iBeacon, ...

Ian Y. Choi (Occupation: HURAY)

KrDAG 회원인데 네트워크 엔지니어 아니고 개발자...

목차

1. IoT와 네트워크
2. 블루투스과 비콘
3. 비콘의 활용 예시: 실내 측위
4. 라즈베리파이2+비콘을 통한 구현 테스트
5. 마무리

IoT (Internet of Things)

IoT의 정의에 대해 잠깐 살펴봅시다

센서가 부착된 사물을 유무선 통신망으로 연결, 이를 통해 발생하는 실시간 데이터를 사람 개입 없이 인터넷으로 주고 받는 기술이나 환경 (NIA, 2015)

ICT를 기반으로 다양한 물리적 및 가상의 사물들을 연결하여 진보된 서비스를 제공하기 위한 글로벌 서비스 인프라 (ITU-T)

구분	M2M (Machine-to-Machine)	IoT (Internet of Things)
통신주체	Machine	Things/환경
연결관계	사물-사물 사물-사람	사람-Things 사람-사람
연결방식	유/무선 네트워크	Internet

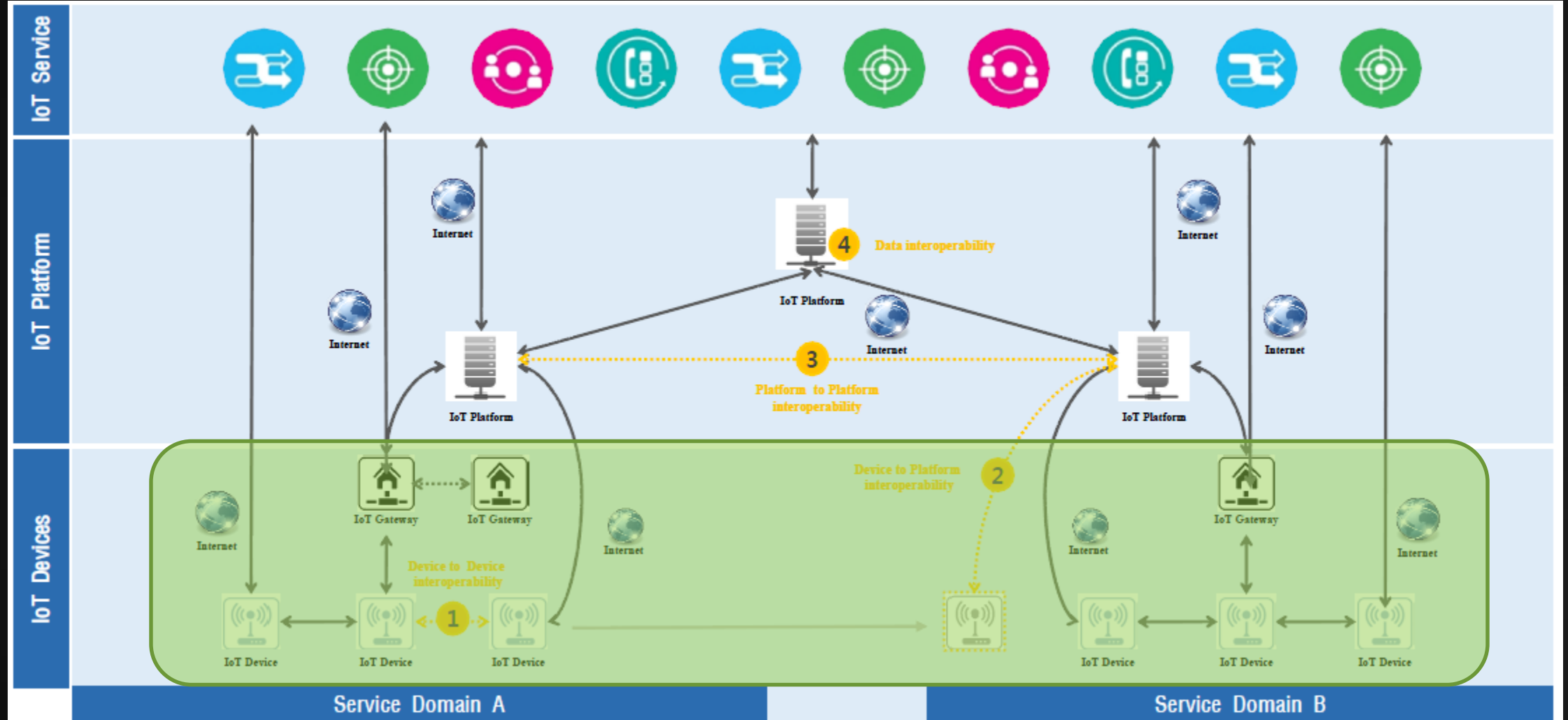
IoT 플랫폼

여러 분야를 중심으로 하여 IoT 플랫폼이 발전하고 있음



IoT 플랫폼과 네트워크

결국 Internet이 연결되어야 하는데, 장치쪽 네트워크는 어떻게?



IoT 장치를 위한 네트워크 프로토콜

고려 사항: 수용 범위, 전송 속도, 저전력, 지원 장치 대수, ...

무엇을 위한 IoT인지도 중요하고, IoT를 위해 수집하는 센서 및 정보도 중요한데...
어떤 장치를 어떤 네트워크로 사용해 IoT 플랫폼을 구성해야 할까 역시 중요함!

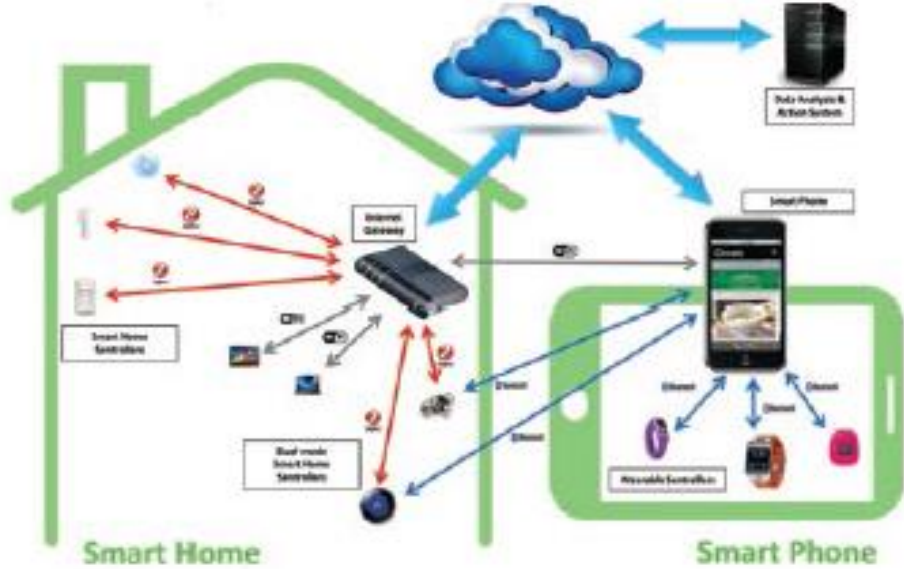
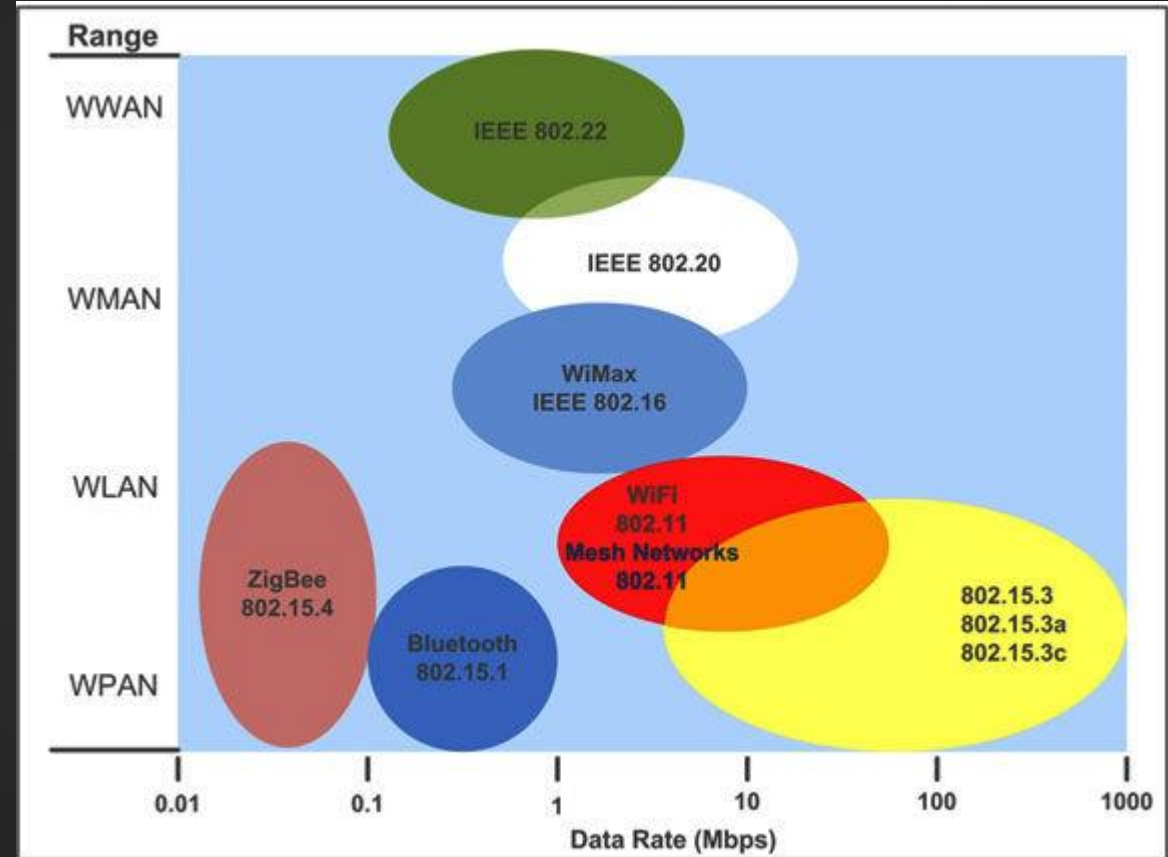


Figure 2: WiFi, Bluetooth (Smart) and ZigBee in the Smart Home and the Smartphone






#2. 블루투스 와 비콘

파란 이빨과 신호등은 무슨 관계?!

Bluetooth, BLE, Beacon, iBeacon 순으로 알아보시다

beacon 미국·영국 [ˈbiːkən]  영국식  ★

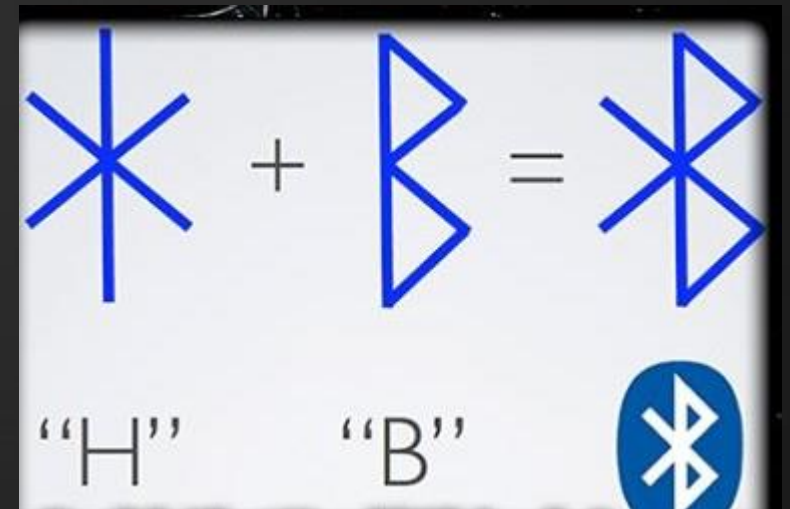
1. (안전 운행을 유도하는) 신호등
2. (배·비행기의 위치 확인을 돕는) 무선 송신소
3. (과거 신호용으로 피워 올리던) 봉화

Bluetooth란

이름의 유래부터 살펴봅시다

10세기 스칸디나비아 반도를 통일한 바이킹 헤럴드 블루투스 (Harald Bluetooth)처럼 여러 디지털 기기에서 사용하는 무선 통신 규격을 통일한다는 상징적 의미를 담고자 하였음

저렴한 가격, 저전력, 소형 크기 지원을 강조하는 Bluetooth는 1998년 Bluetooth SIG (Special Interest Group)이 결성되고, 1999년 7월 26일 1.0 규격이 탄생하여, 현재 4.0 버전대 규격까지 나와있음



Bluetooth 역사

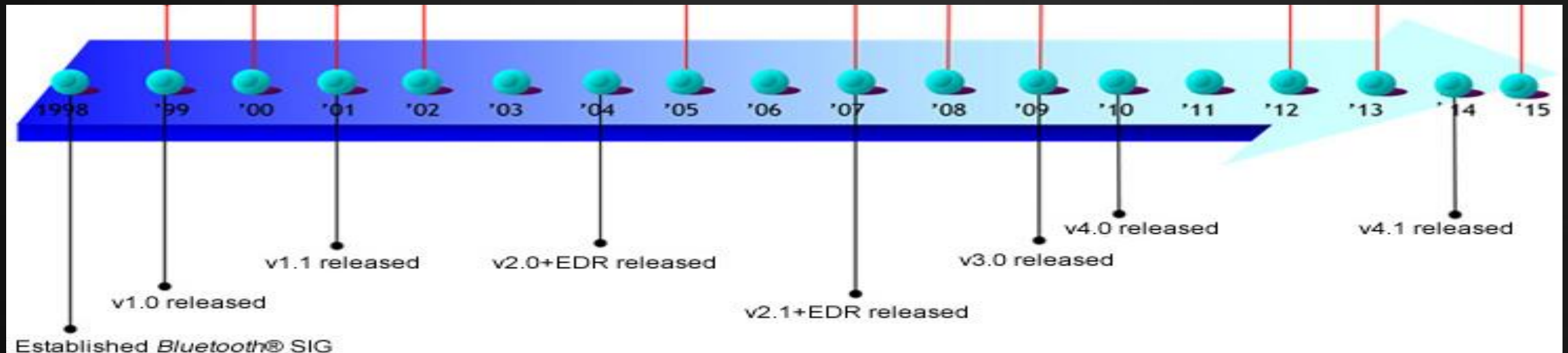
1.0부터 4.0 버전대까지의 시간 순서

현재 버전: Bluetooth v4.2 (2014년 12월 2일)

Version	Data rate	Max. application throughput
1.2	1 Mbit/s	>80 kbit/s
2.0 + EDR	3 Mbit/s	>80 kbit/s
3.0 + HS	24 Mbit/s	See Version 3.0 + HS
4.0	24 Mbit/s	See Version 4.0 LE

참고 - EDR: Enhanced Data Rate, HS: High Speed

4.0 버전대에 포함된 것 중 하나로, 바로 저전력 (Low Energy)가 있음 → BLE라고 함



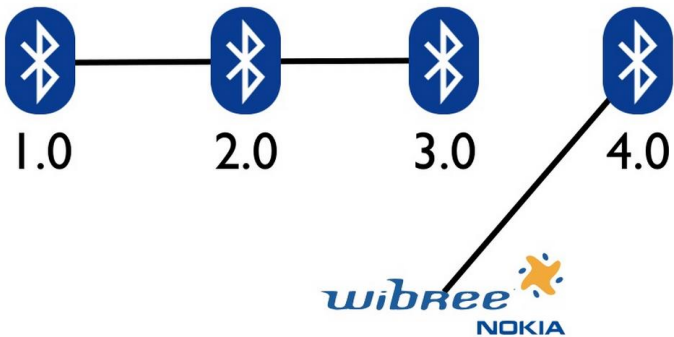
BLE (Bluetooth Low Energy)

저전력을 강조하는 Bluetooth 4.0에 포함된 서브셋 BLE

Bluetooth 버전 4.0을 발표하면서, 지원 전송 속도가 감소되지만 저전력을 강조하는 BLE를 Bluetooth 4.0의 서브셋으로 발표

버전 4.1: Bluetooth SIG가 IoT 시장에서 주요 플레이어가 되고자 함

BT4.0 Not Backwards Compatible



BluetoothTM

4.0

Low Energy

	Bluetooth V2.1	Bluetooth Low Energy
Standardization Body	Bluetooth SIG	Bluetooth SIG
Range	~30 m (class 2)	~50 m
Frequency	2.4–2.5 GHz	2.4–2.5 GHz
Bit Rate	1–3 Mbit/s	~200 kbit/s
Set-Up Time	<6 s	<0.003 s
Voice Capable?	Yes	No
Max Output Power	+20 dBm	+10 dBm
Modulation Scheme	GFSK	GFSK
Modulation Index	0.35	0.5
Number of Channels	79	40
Channel Bandwidth	1 MHz	2 MHz

비콘 (Beacon)

















본래 무선에서 내보내는 주기적인 신호를 비콘이라고 이야기하나,
여기서 이야기하는 비콘은 “블루투스 비콘” 을 말함


작고, 배터리를 갖고 있는 무선송신기

BLE 안테나를 내장하며 신호를
브로드캐스팅함

구성 요소
: 보드 (Board), 블루투스 칩셋 (BLE
Chipset), 배터리



 Estimate estimate Stylish beacons with colorful silicon case, by Y-combinator alum startup Estimate.	 Chipset Nordic	 Battery Large (1000 mAh)	 Life 21.4 Months
 Gimbal series 10 GIMBAL™ Slick and thin beacons by Qualcomm spinoff, Gimbal.	 Chipset Gimbal	 Battery Small (240 mAh)	 Life 0.9 Months
 Gimbal series 21 GIMBAL™ Enterprise beacon offering by Qualcomm spinoff, Gimbal.	 Chipset Gimbal	 Battery 4x AA (2000 mAh)	 Life 16.4 Months
 iNow Beacon by iNow, with Nordic nRF51822 chipset and a AAA battery	 Chipset Nordic	 Battery 2x AAA (1000 mAh)	 Life 12 Months

 Redbear Labs RedBearLab Larger sized beacon with two alkaline AA batteries and Texas chipset.	 Chipset Texas Instruments	 Battery 2x AAA (1000 mAh)	 Life 21.0 Months
 roximity ROXIMITY Beacons by American company Roximity with Nordic Semiconductors chipset and a small-sized button cell.	 Chipset Nordic	 Battery Small (240 mAh)	 Life 6.0 Months
 Radius Network Radius Networks Battery-powered beacon by American company Radius Networks, the maker of open-source beacon-related utilities.	 Chipset Nordic	 Battery Small (240 mAh)	 Life 5.0 Months
 MPact MPact beacons are developed by Motorola Solutions, now owned by Zebra Technologies.	 Chipset Texas Instruments	 Battery Medium (640 mAh)	 Life 2.1 Months
 RECO RECO Beacon by Reco, with Nordic Semiconductors chipset and a medium-sized battery.	 Chipset Nordic	 Battery Medium (640 mAh)	 Life 23.0 Months

iBeacon

2013년 Apple 개발자 컨퍼런스에서 발표되면서 대중적으로 확대

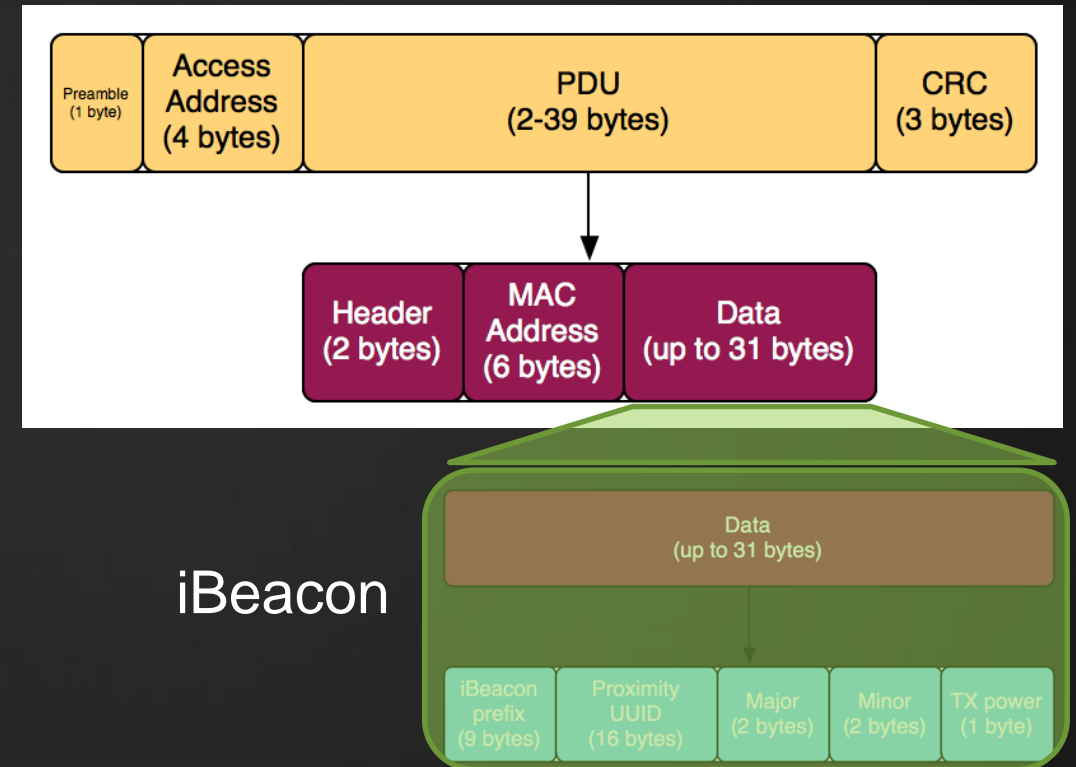
BLE에는 Advertising과 Connecting 2가지 모드가 있는데, 비콘에서는 Advertising만 사용하여 주기적으로 신호를 브로드캐스팅하는 것이 일반적임

iBeacon에서 BLE Advertising로 보내는 데이터: UUID, Major, Minor, TX power

iBeacons

- Core Location monitors Bluetooth LE beacon signals
- Various hardware can be a beacon
 - Third-party Bluetooth LE emitters
 - iOS devices
- Advantages
 - Accuracy and range awareness
 - One beacon ID can cover multiple locations

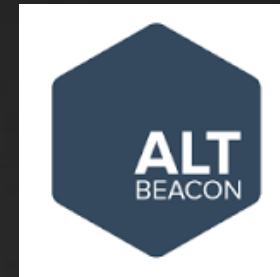
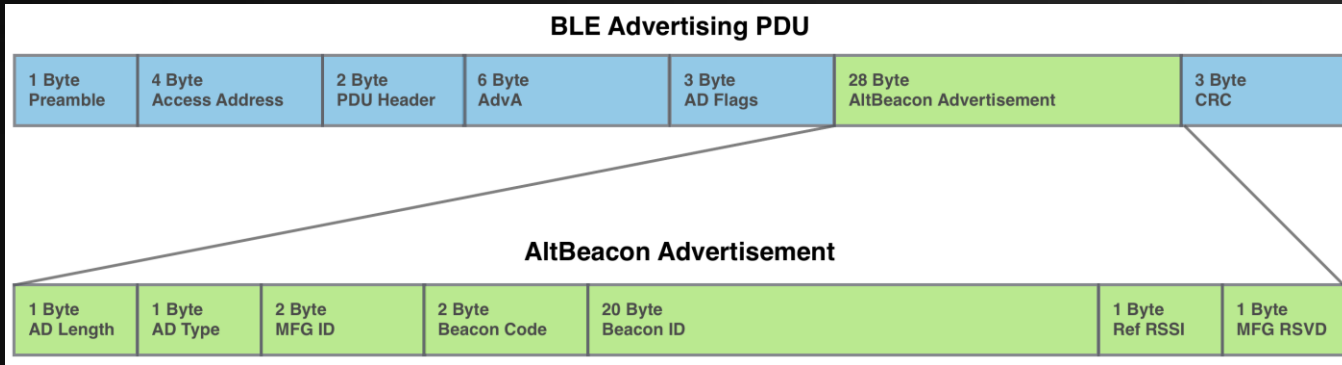




질문1: iBeacon이 표준인가?

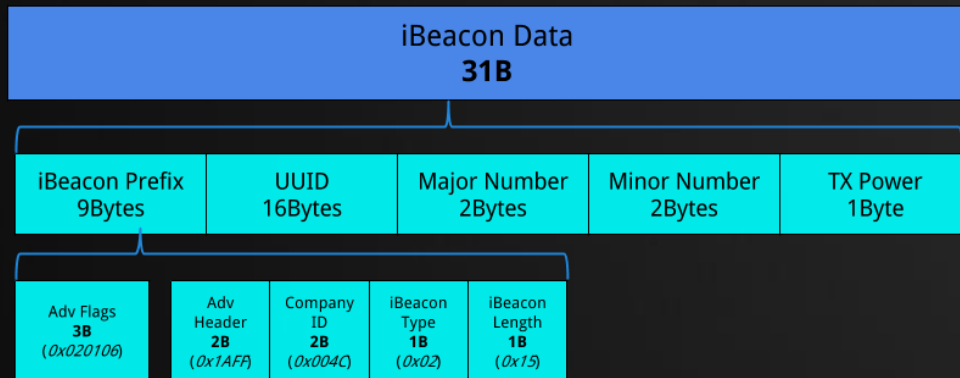
iBeacon이 표준은 아니나, 거의 모든 태그들이 iBeacon을 지원합니다.

BLE에서 보내는 데이터에 대한 여러 프로토콜: iBeacon, AltBeacon, URIBeacon, ...

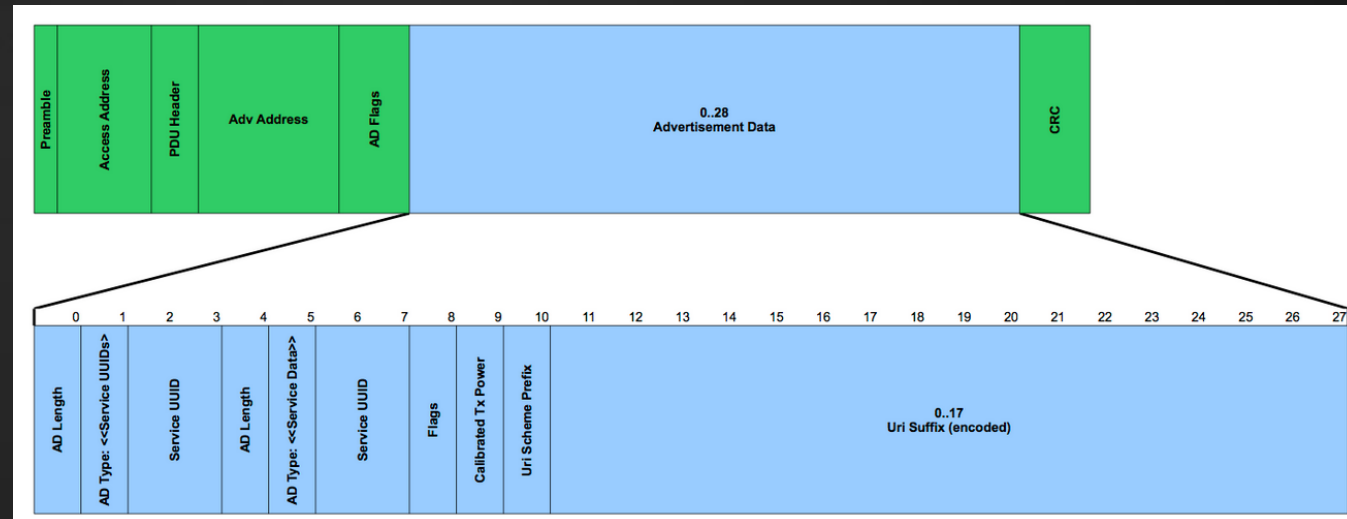


AltBeacon

URIBeacon



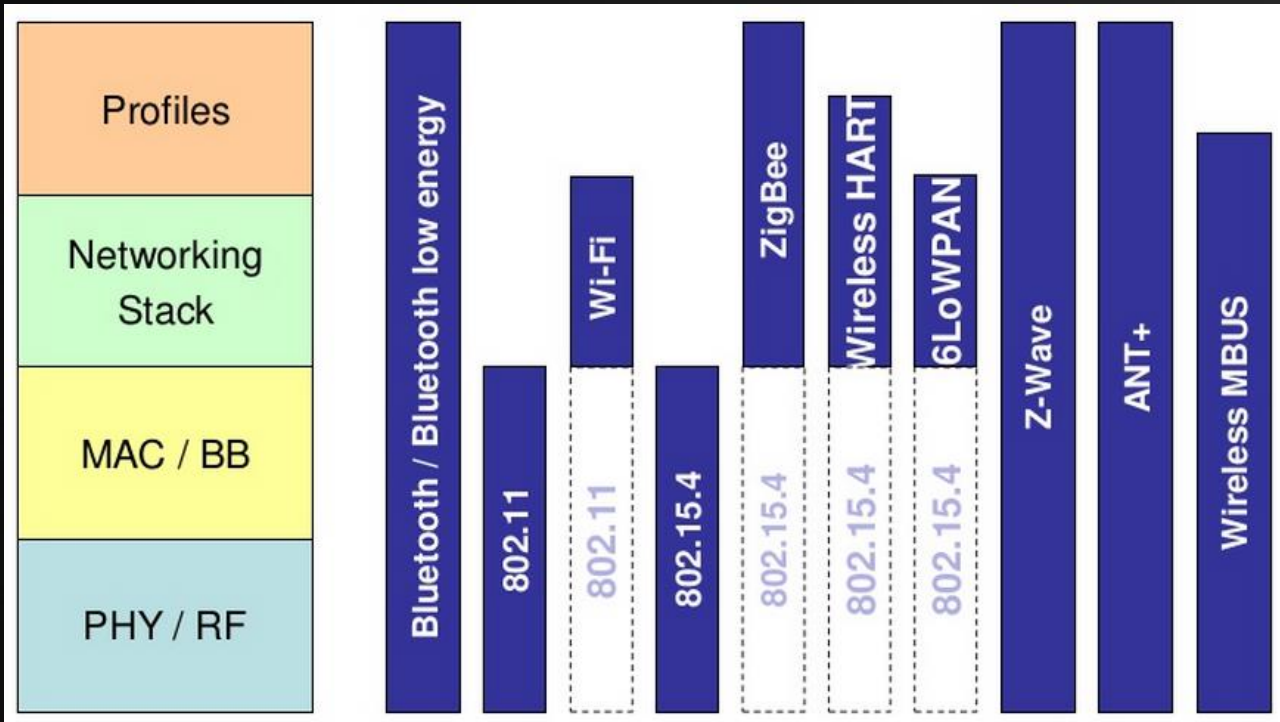
iBeacon



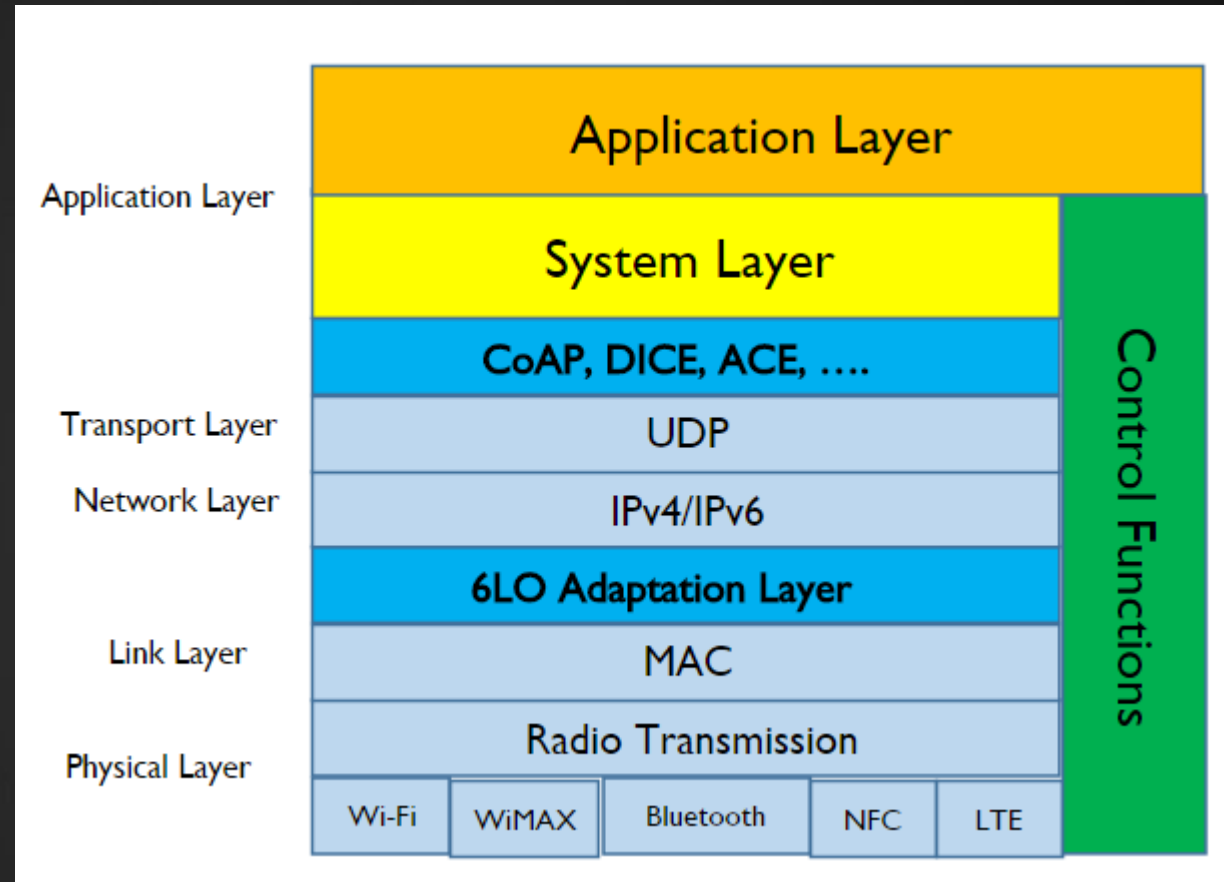
질문2: IoT에서는 BLE만이 답인가?

상황에 따라서는 다른 네트워크 방식이 효율적일 수도 있습니다.

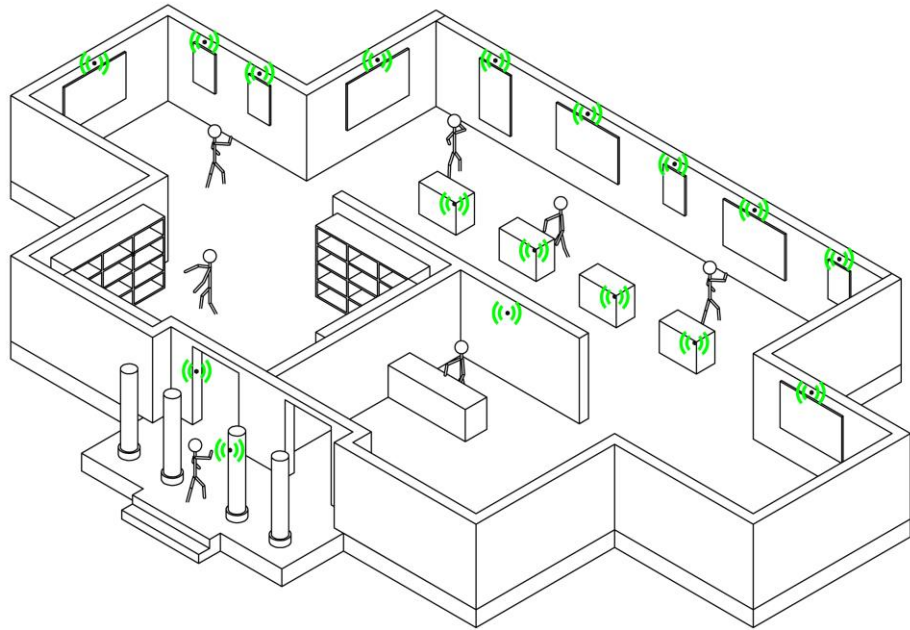
다양한 무선 통신 표준의 장단점이 존재하며 필요에 따라 상위 계층 고려도 있어야 함



다양한 무선 통신 표준



다양한 IoT 네트워크 프로토콜



#3. 비콘 활용 예시

실내 측위 (Indoor Positioning)

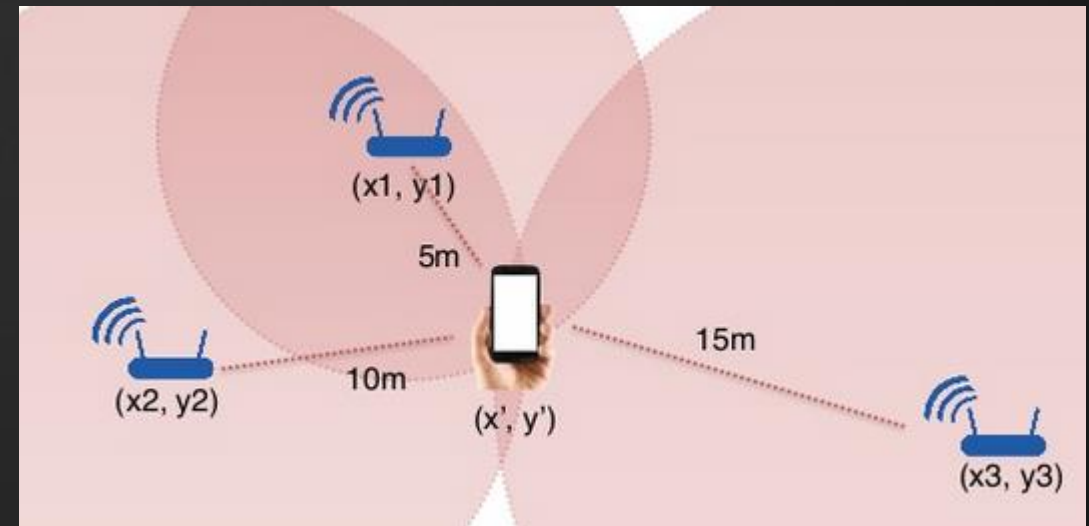
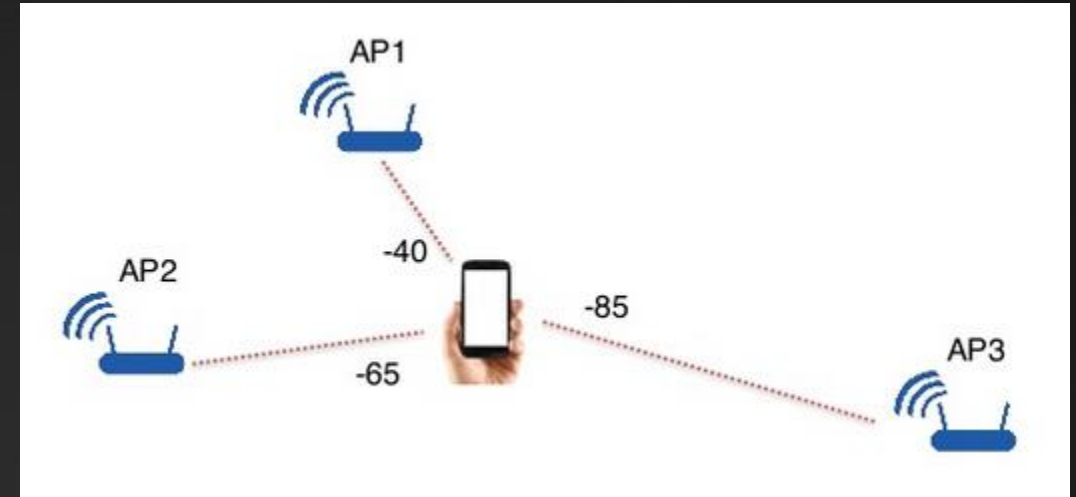
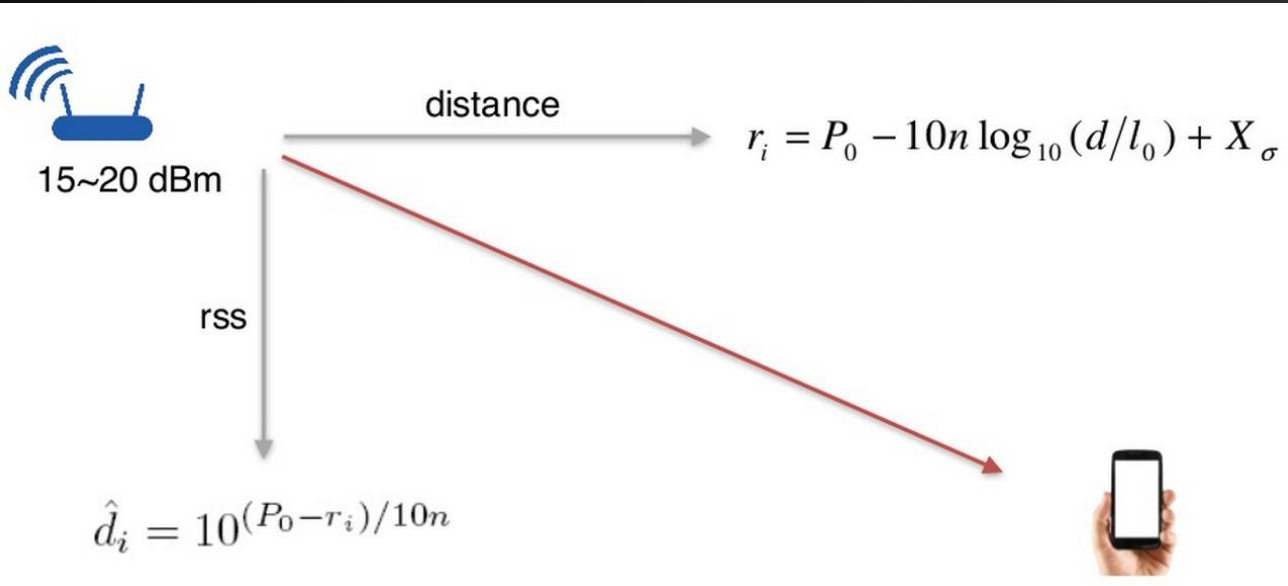
비콘 신호 데이터를 이용해 위치를 파악한다고?

무선 통신을 활용한 실내 측위

무선 신호의 특징을 활용하면 (이론적으로) 위치 파악이 가능합니다.

RSSI (Received signal strength indication, 수신된 신호 강도 지표)로 거리를 대략적으로 파악할 수 있다고 합니다.

WiFi의 경우 예시: 삼각측량법



비콘으로 실내 측위가 가능할까?

정확도가 매우 높지는 않으나 실내/외, ~미터 정도로는 가능하다고 합니다.

“위치기반 기술의 또 다른 혁신, 비콘”

비콘 기술 적용 사례 - 국내

- 국내에서도 비콘 활용은 점점 늘어나는 추세로, 온·오프라인 유통업체들 사이에서 가장 적극적으로 이뤄지고 있음
- 커피전문점 스타벅스는 세계에서 유일하게 국내에서만 비콘을 활용한 서비스를 도입하여 모바일로 주문과 결제를 동시에 진행 할 수 있는 서비스를 제공하고 있으며, SK플래닛은 비콘과 자체 모바일 지갑 서비스 앱인 '시럽(Syrup)'을 연계하여 비콘 사업에 진출
- 온라인 쇼핑몰 G마켓은 3호선 고속터미널역 사각 기둥 광고에 광고물과 비콘을 설치하여 근처 승객을 대상으로 한 무료 음료 쿠폰 이벤트를 진행

비콘 기술 적용 사례

[스타벅스 앱 '사이렌오더' 기능]



- 스타벅스코리아의 커피 주문 앱인 '사이렌오더'는 이용자의 스마트폰에 순차적인 음료제조 과정을 팝업 메시지와 진동으로 전달받을 수 있음
- '나만의 음료'라는 항목을 통해서도 맞춤형 음료도 주문 가능

['시럽(Syrup)'-비콘 연계 앱]



- 시럽 앱을 설치한 유저는 비콘이 설치된 시럽 제휴 매장에서 비콘에서 전송된 쿠폰과 구매정보를 받아볼 수 있음
- 연말까지 전국에 약 5만개 이상 비콘을 설치해 주요 상권마다 '시럽존'을 구축할 예정

[지마켓 고속터미널역 캠페인]



- 3호선 고속터미널역 플랫폼에 진입하여 G마켓 사각 기둥 광고 근처에 접근하면 G마켓 앱을 통해 푸시메시지가 전송되고, 해당 메시지 확인 시 G마켓 앱으로 이동하여 쿠폰 다운 가능



#4. 비콘 with 라즈베리파이2

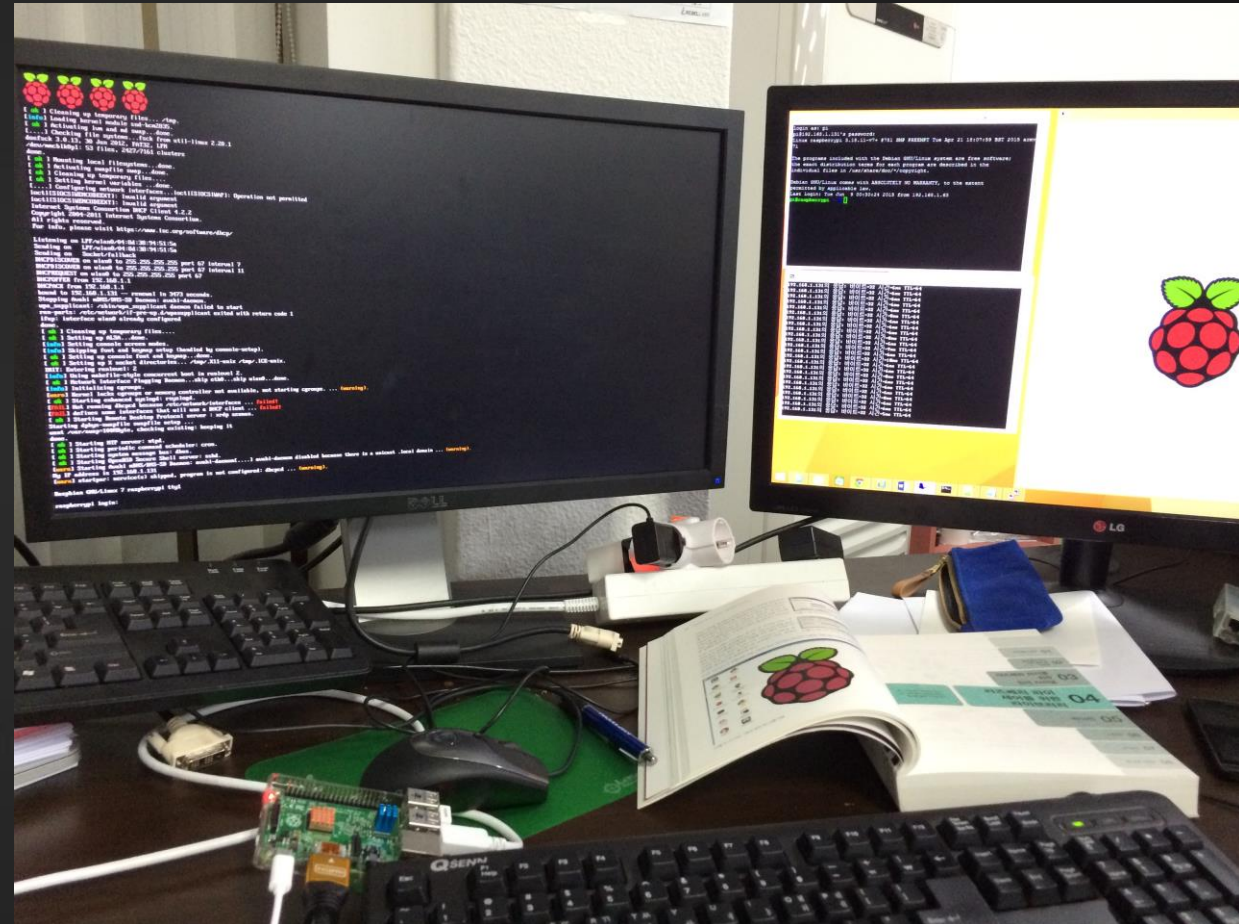
비콘 신호를 보내거나 수신 가능

(프로그래밍 이야기는 죄끔만 넣을게요...)

라즈베리파이2

2015년 상반기, KrDAG IoT 소모임을 통해 배운 라즈베리파이2

재미있었던 소수 정예의 소모임 활동 ☺
(자세한 내용은 소모임 멤버분들께...!)



테스트를 위해 필요한 것들

라즈베리파이2로 비콘을 활용하려면?

라즈베리파이2 + 블루투스 4.0 dongle

해당 조합을 통해 비콘 신호를 브로드캐스팅할 수도 있으며 (송신자), 브로드캐스트 신호를 받을 수도 있음 (수신자)

Estimote 태그를 송신자로 하고, 라즈베리파이2+블루투스 4.0 dongle을 수신자로 테스트



Estimote 비콘 태그



(수신 및 데이터 확인)

테스트 내용

라즈베리파이2로 iBeacon 수신하기 위해 했던 내용 및 주요 과정

블루투스 4.0 동글 인식 확인: “sudo lsusb”

블루투스 장치 up: “sudo hciconfig hci0 up”

Bluez 라이브러리 설치: 프로그래밍을 통해 블루투스 데이터 확인 가능

PyBluez 설치: Python으로 Bluez 사용 가능

샘플 코드: <https://github.com/switchdoclabs/iBeacon-Scanner/blob/master/blescan.py>

샘플 코드 실행 후 결과 확인

테스트 결과

샘플 코드 다운로드 & 실행

```
pi@raspberrypi: ~/iBeacon-Scanner-
File Edit Tabs Help
pi@raspberrypi ~ $ git clone https://github.com/switchdoclabs/iBeacon-Scanner-.git
Cloning into 'iBeacon-Scanner-'...
remote: Counting objects: 5, done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 5
Unpacking objects: 100% (5/5), done.
pi@raspberrypi ~ $ cd iBeacon-Scanner-
pi@raspberrypi ~/iBeacon-Scanner- $ sudo python testblescan.py
ble thread started
-----
b8:78:2e:0f:a1:84,a10f2e78b80f02011a0bffa4c00090603,33984,43008,101,-81
65:92:bc:b9:45:85,0100018545b9bc92,25859,513,26,-61
4c:90:32:18:25:11,0613ffa4c000c0e00a9b8c777aaf5aa58,14598,13514,-58,-67
b8:78:2e:0f:a1:84,a10f2e78b80f02011a0bffa4c00090603,33984,43008,101,-83
65:92:bc:b9:45:85,0100018545b9bc92,25859,513,26,-66
e8:ac:a2:80:ef:a5,b9407f30466eaff925556b57fe6d41315,196,-74,-88
c2:89:a1:63:30:f1,df23eaeaf49746e189069066d9a087d9,12529,41315,-60,-58
b8:78:2e:0f:a1:84,a10f2e78b80f02011a0bffa4c00090603,33984,43008,101,-83
5f:d7:41:4e:9e:bb,1a13ffa4c000c0e00882aaaa09d962154,52973,21443,-82,-84
-----
b8:78:2e:0f:a1:84,a10f2e78b80f02011a0bffa4c00090603,33984,43008,101,-83
c2:89:a1:63:30:f1,df23eaeaf49746e189069066d9a087d9,12529,41315,-60,-52
65:92:bc:b9:45:85,0100018545b9bc92,25859,513,26,-70
5f:d7:41:4e:9e:bb,1a13ffa4c000c0e00882aaaa09d962154,52973,21443,-82,-84
```

MAC
Address

Proximity
UUID

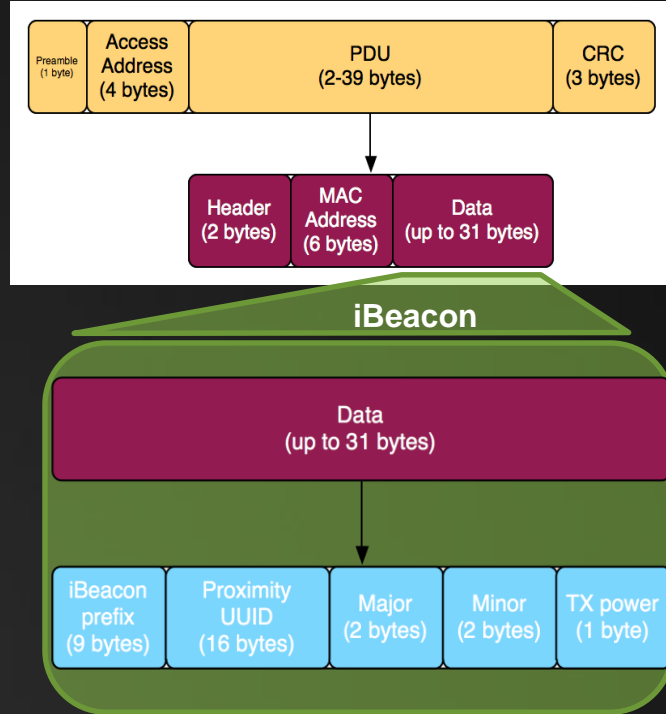
Major

Minor

TX power
@1m
해당 값이
소스에서
틀리게 나옴

```
thon testblescan.py
a087d9,12529,41315,196
00df07,1808,769,45
000000,180,180,197
0090603,33984,43008,101
00df07,1808,769,45
```

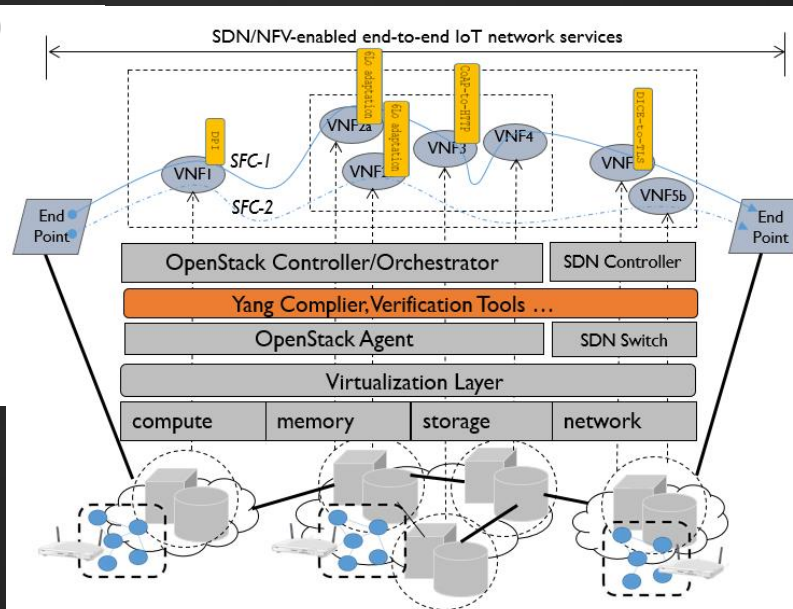
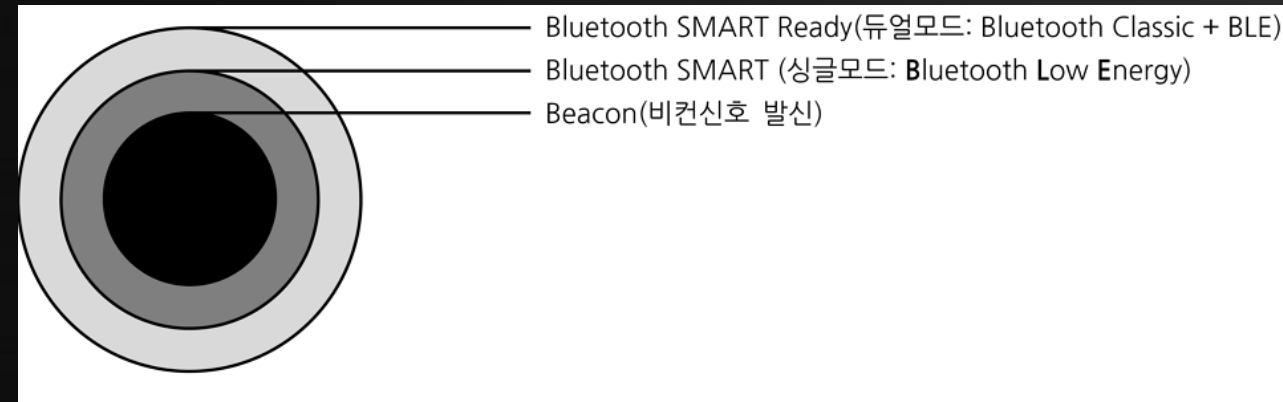
수정된 TX
power
@1m 값 결과



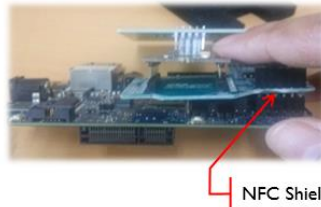
#5. 마무리

마무리

- ✓ IoT 세상에서 인터넷 연결을 위해 네트워크는 필수
- ✓ 그 중, Bluetooth→BLE→Beacon→iBeacon 순으로 살펴봄
- ✓ 라즈베리파이2를 사용해 비콘 테스트를 쉽고 빠르게 할 수 있음
- ✓ 비콘은 실내 측위를 비롯해 다양한 활용 가능 → IoT 세상이 오긴 오는 듯?!
- ✓ 다양한 네트워크 프로토콜로 인한 IoT 게이트웨이 → SDN / NFV ?!



IPv6 over NFC functions
<draft-hong-6lo-ipv6-over-nfc>



Q&A

#. Appendix

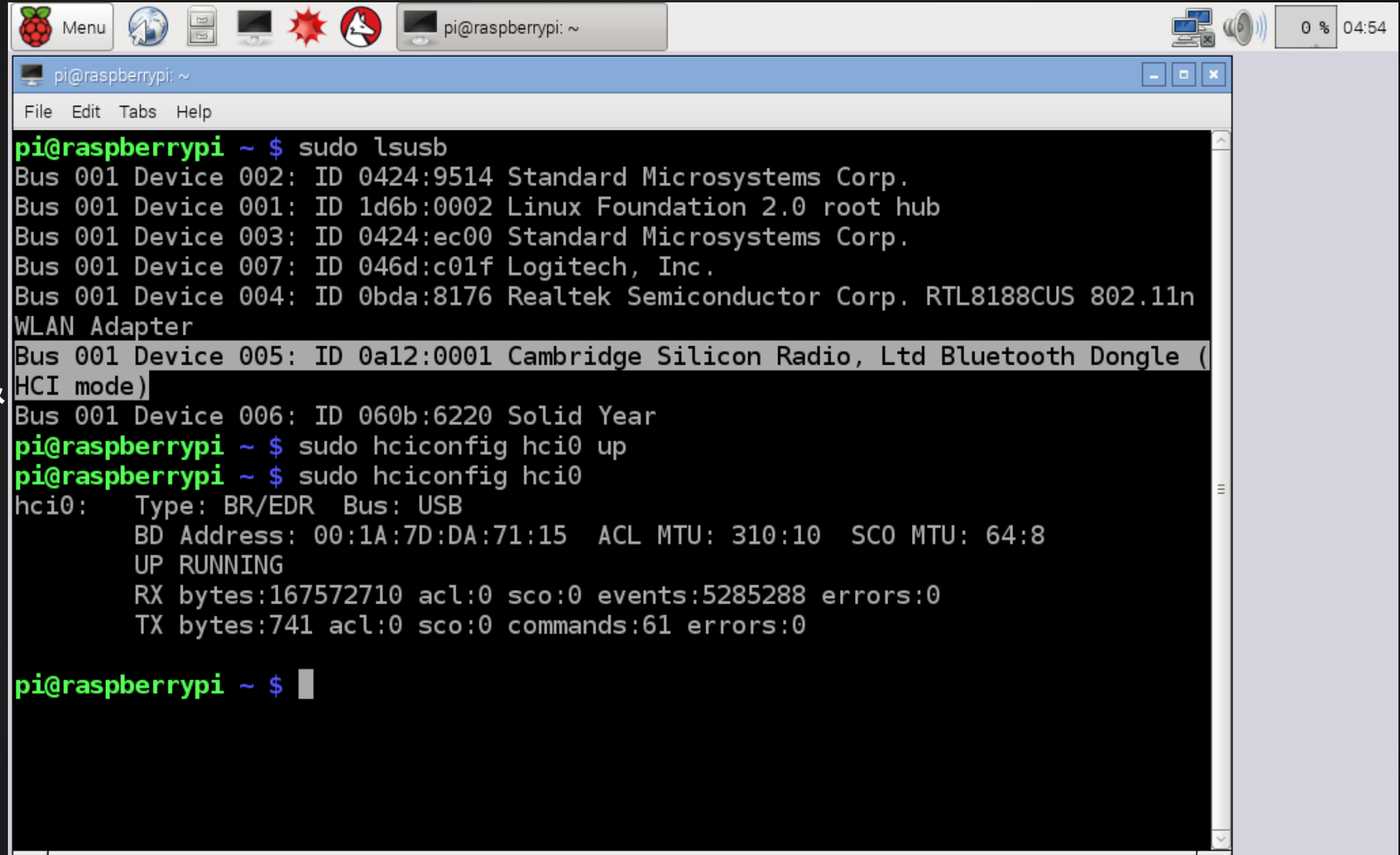
설치 과정: Details

블루투스 4.0 동글 확인 & 장치 UP & 확인

sudo: root (관리자)
권한으로 실행

lsusb: USB 상태 보기

hciconfig: 블루투스
호스트 컨트롤러
인터페이스 보기 &
UP/DOWN 설정

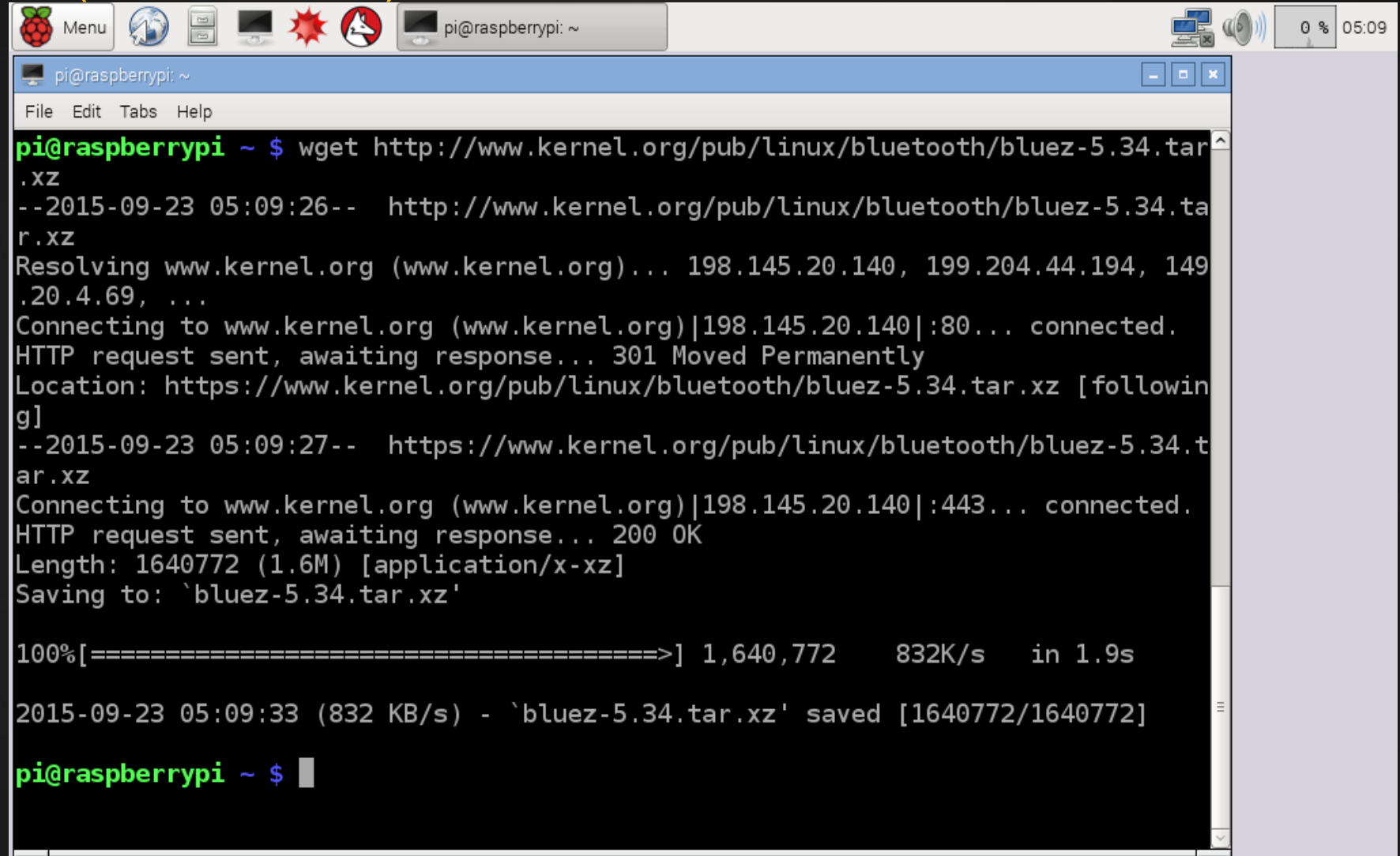


```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ sudo lsusb  
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.  
Bus 001 Device 007: ID 046d:c01f Logitech, Inc.  
Bus 001 Device 004: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n  
WLAN Adapter  
Bus 001 Device 005: ID 0a12:0001 Cambridge Silicon Radio, Ltd Bluetooth Dongle (HCI mode)  
Bus 001 Device 006: ID 060b:6220 Solid Year  
pi@raspberrypi ~ $ sudo hciconfig hci0 up  
pi@raspberrypi ~ $ sudo hciconfig hci0  
hci0: Type: BR/EDR Bus: USB  
BD Address: 00:1A:7D:DA:71:15 ACL MTU: 310:10 SCO MTU: 64:8  
UP RUNNING  
RX bytes:167572710 acl:0 sco:0 events:5285288 errors:0  
TX bytes:741 acl:0 sco:0 commands:61 errors:0  
  
pi@raspberrypi ~ $
```

설치 과정: Details

Bluez 라이브러리 설치 (소스 컴파일): 소스 다운로드

wget: Web으로부터
파일 다운로드



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ wget http://www.kernel.org/pub/linux/bluetooth/bluez-5.34.tar.xz  
--2015-09-23 05:09:26-- http://www.kernel.org/pub/linux/bluetooth/bluez-5.34.tar.xz  
Resolving www.kernel.org (www.kernel.org)... 198.145.20.140, 199.204.44.194, 149.20.4.69, ...  
Connecting to www.kernel.org (www.kernel.org)|198.145.20.140|:80... connected.  
HTTP request sent, awaiting response... 301 Moved Permanently  
Location: https://www.kernel.org/pub/linux/bluetooth/bluez-5.34.tar.xz [following]  
--2015-09-23 05:09:27-- https://www.kernel.org/pub/linux/bluetooth/bluez-5.34.tar.xz  
Connecting to www.kernel.org (www.kernel.org)|198.145.20.140|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 1640772 (1.6M) [application/x-xz]  
Saving to: `bluez-5.34.tar.xz'  
  
100%[=====>] 1,640,772 832K/s in 1.9s  
  
2015-09-23 05:09:33 (832 KB/s) - `bluez-5.34.tar.xz' saved [1640772/1640772]  
  
pi@raspberrypi ~ $
```


설치 과정: Details

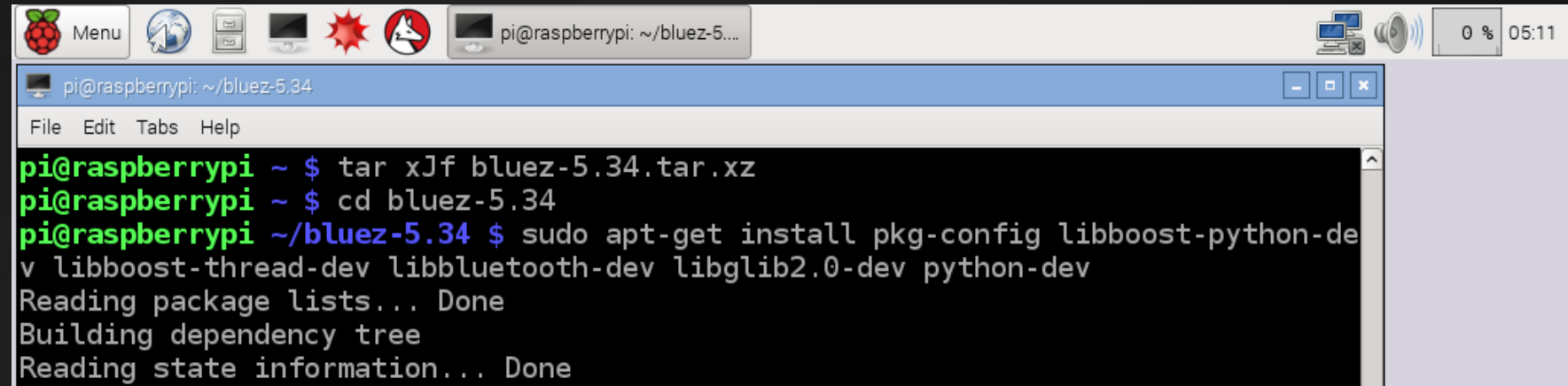
Bluez 라이브러리 설치 (소스 컴파일): 압축 해제 & 의존 라이브러리 설치

tar: (압축) 묶음을
만들거나 해제할 때
사용하는 도구

tar 옵션

- x: 묶음 해제
- J: xz 압축 고려
- f: 파일 이름 지정

apt-get: Debian 계열
리눅스에서 패키지
관리 명령어



```
pi@raspberrypi: ~/bluez-5.34
File Edit Tabs Help

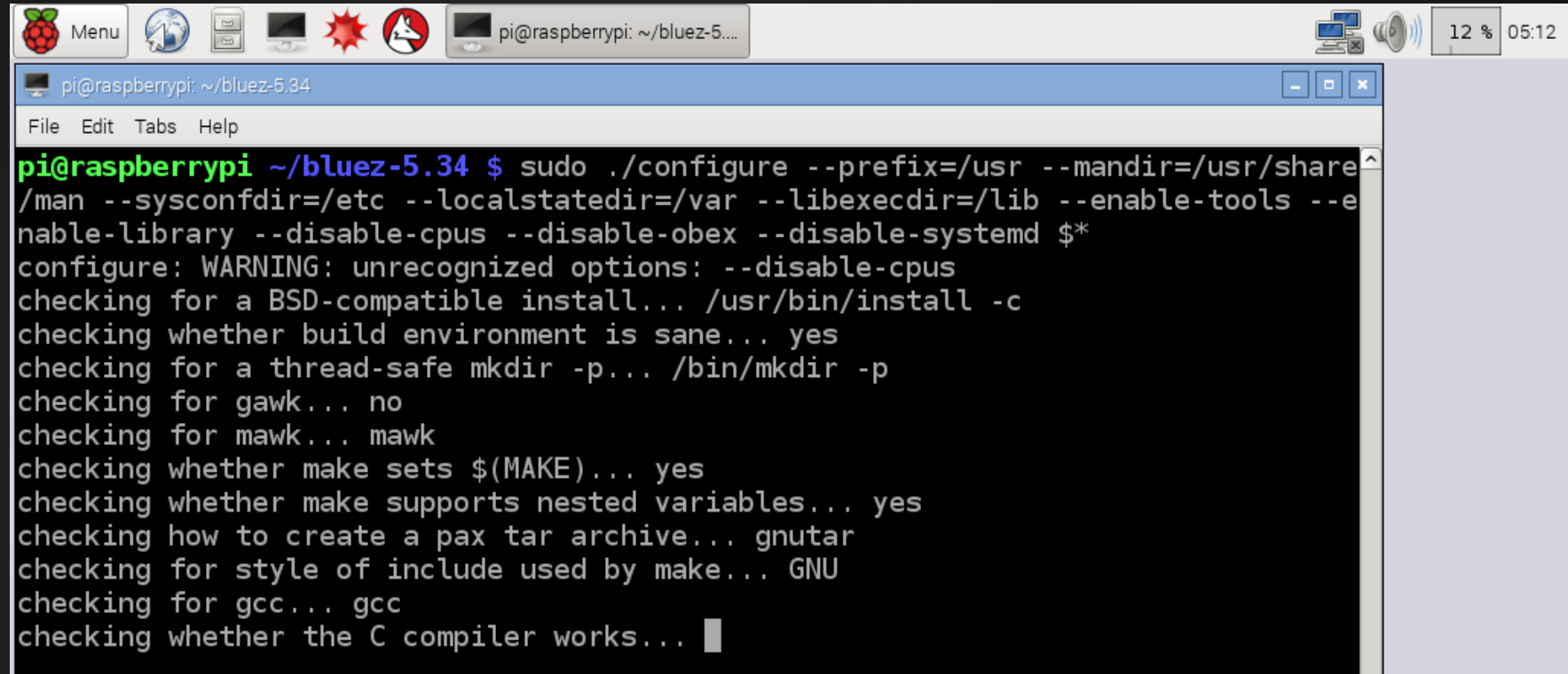
pi@raspberrypi ~ $ tar xJf bluez-5.34.tar.xz
pi@raspberrypi ~ $ cd bluez-5.34
pi@raspberrypi ~/bluez-5.34 $ sudo apt-get install pkg-config libboost-python-dev
libboost-thread-dev libbluetooth-dev libglib2.0-dev python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

설치 과정: Details

Bluez 라이브러리 설치 (소스 컴파일): 소스 컴파일 설정

configure

: 소스 컴파일 작업
이전에 필요한
설정을 지정



```
pi@raspberrypi: ~/bluez-5.34
File Edit Tabs Help

pi@raspberrypi ~/bluez-5.34 $ sudo ./configure --prefix=/usr --mandir=/usr/share
/man --sysconfdir=/etc --localstatedir=/var --libexecdir=/lib --enable-tools --e
nable-library --disable-cpus --disable-obex --disable-systemd $*
configure: WARNING: unrecognized options: --disable-cpus
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking how to create a pax tar archive... gnutar
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... █
```

설치 과정: Details

Bluez 라이브러리 설치 (소스 컴파일): 소스 컴파일 진행

make: 소스 컴파일

작업 시작

make install

: 컴파일 완료된

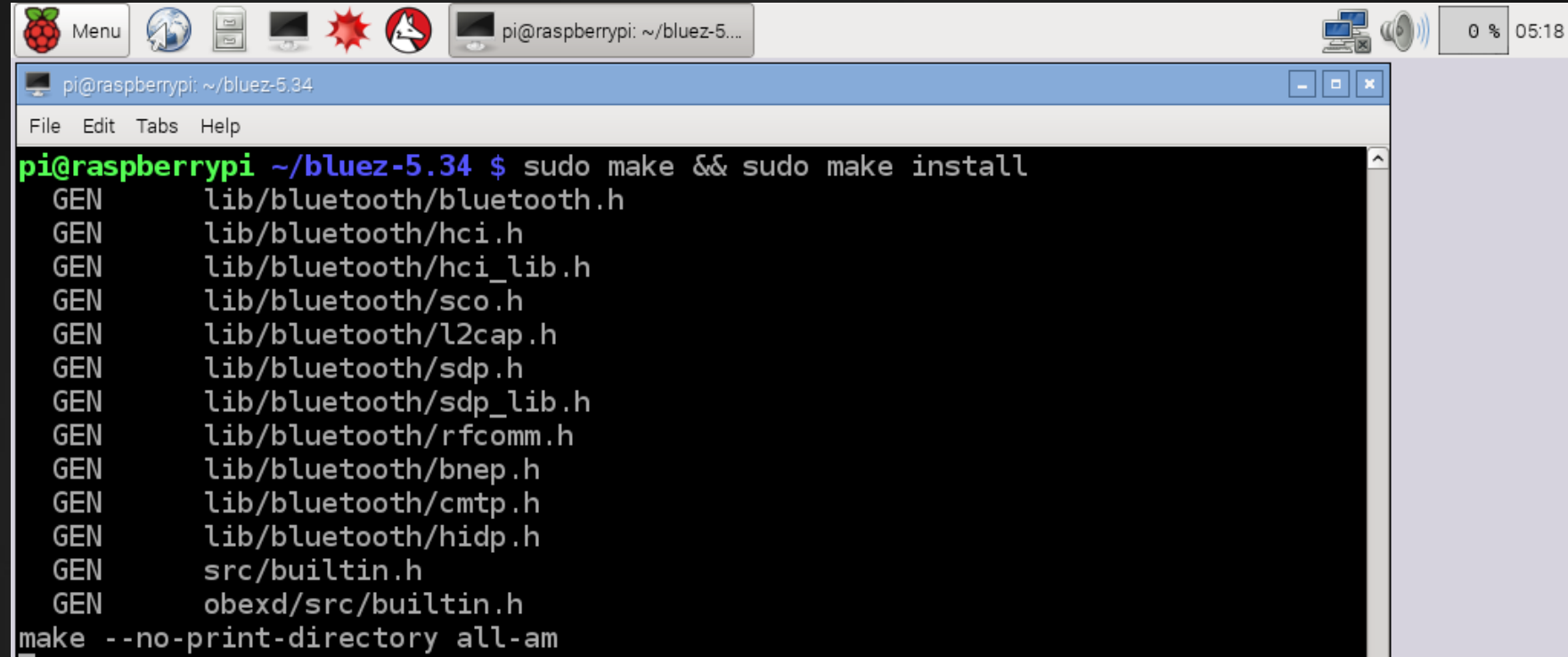
결과를 기반으로

리눅스에 설치

&& : 앞의 작업이

완료된 후 뒤의

작업을 진행

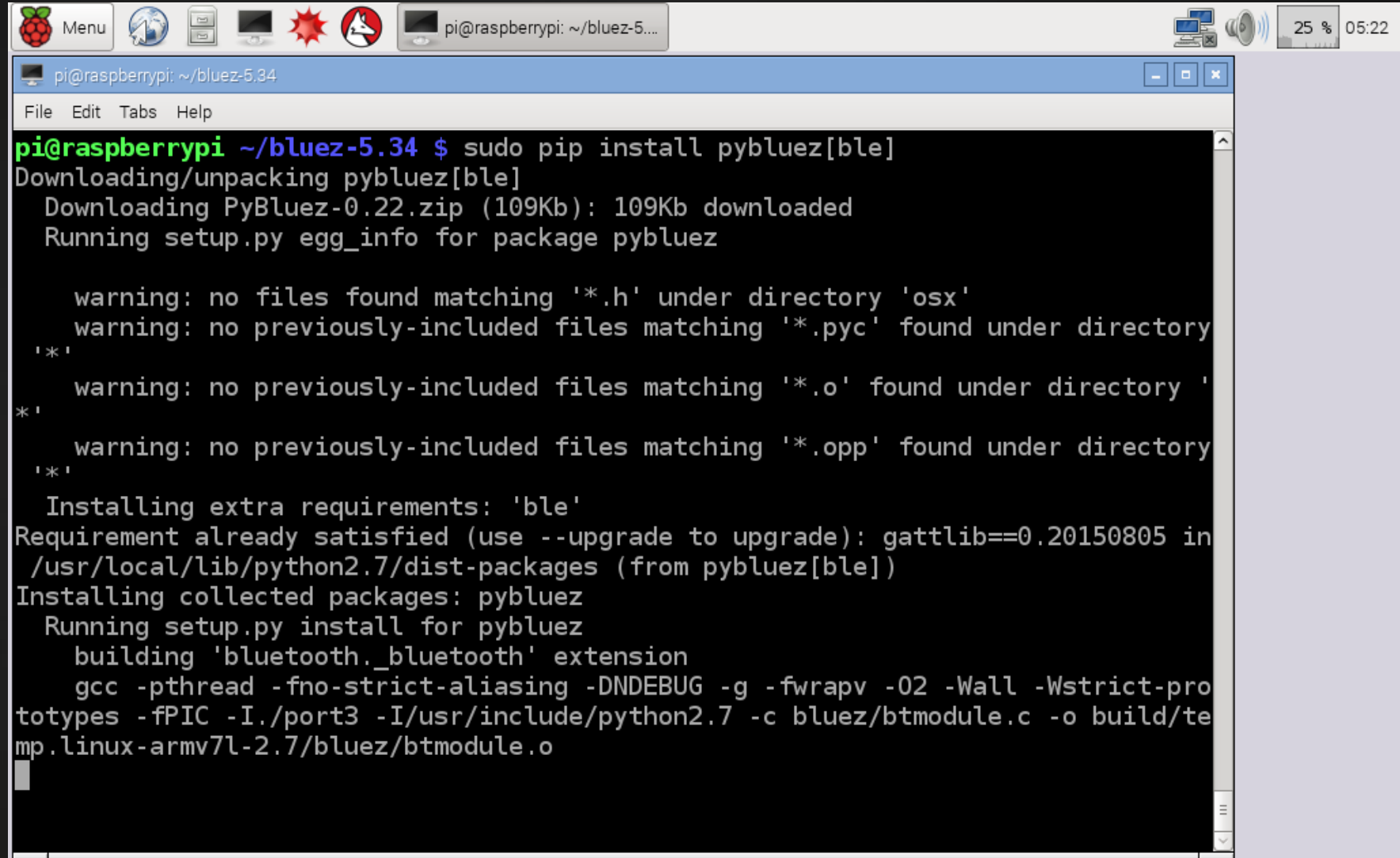


```
pi@raspberrypi: ~/bluez-5.34
File Edit Tabs Help
pi@raspberrypi ~/bluez-5.34 $ sudo make && sudo make install
GEN      lib/bluetooth/bluetooth.h
GEN      lib/bluetooth/hci.h
GEN      lib/bluetooth/hci_lib.h
GEN      lib/bluetooth/sco.h
GEN      lib/bluetooth/l2cap.h
GEN      lib/bluetooth/sdp.h
GEN      lib/bluetooth/sdp_lib.h
GEN      lib/bluetooth/rfcomm.h
GEN      lib/bluetooth/bnep.h
GEN      lib/bluetooth/cmt.h
GEN      lib/bluetooth/hidp.h
GEN      src/builtin.h
GEN      obexd/src/builtin.h
make --no-print-directory all-am
```

테스트 내용: Details

PyBluez 설치: Python에서 BLE 지원을 위한 목적

pip: Python 패키지
설치 명령어



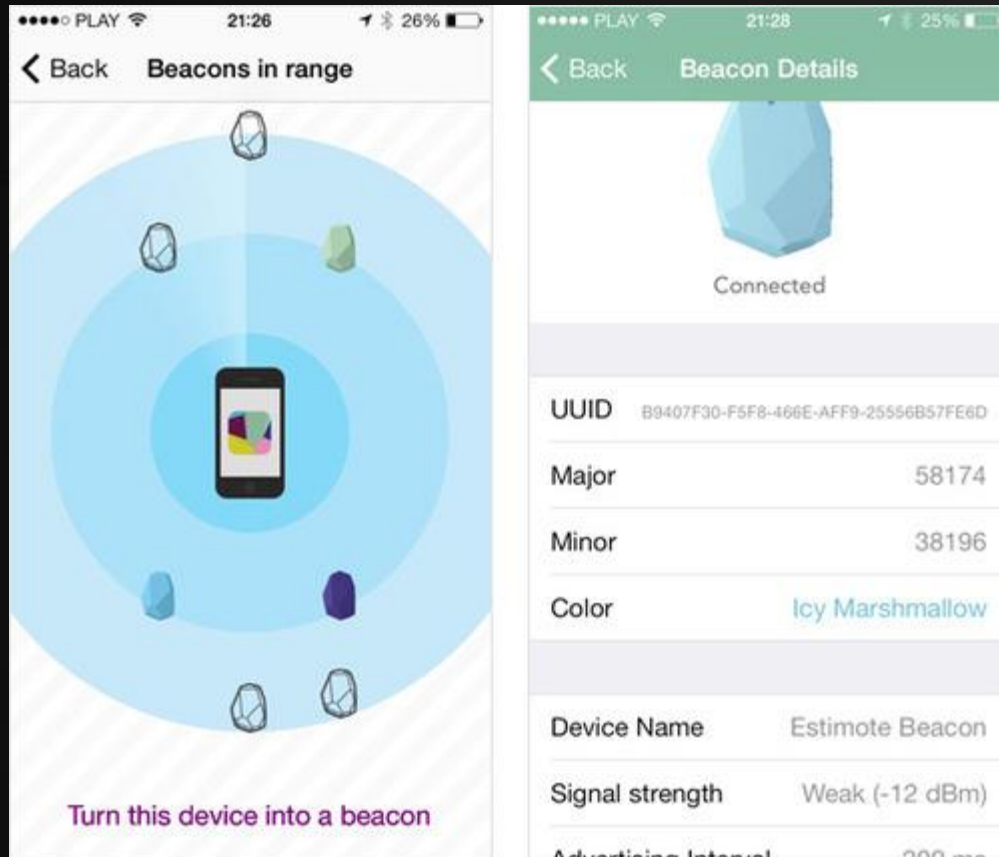
```
pi@raspberrypi: ~/bluez-5.34 $ sudo pip install pybluez[ble]
Downloading/unpacking pybluez[ble]
  Downloading PyBluez-0.22.zip (109Kb): 109Kb downloaded
  Running setup.py egg_info for package pybluez

    warning: no files found matching '*.h' under directory 'osx'
    warning: no previously-included files matching '*.pyc' found under directory
    '*'
    warning: no previously-included files matching '*.o' found under directory '
    *'
    warning: no previously-included files matching '*.opp' found under directory
    '*'

  Installing extra requirements: 'ble'
  Requirement already satisfied (use --upgrade to upgrade): gattlib==0.20150805 in
  /usr/local/lib/python2.7/dist-packages (from pybluez[ble])
  Installing collected packages: pybluez
  Running setup.py install for pybluez
    building 'bluetooth._bluetooth' extension
    gcc -pthread -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-pro
    totypes -fPIC -I./port3 -I/usr/include/python2.7 -c bluez/btmodule.c -o build/te
    mp.linux-armv7l-2.7/bluez/btmodule.o
```

Estimote 클라우드

스마트폰 App을 설치하면 Estimote 태그 정보를 클라우드에 전송



민트1

Edit Settings

Beacon Name	민트1	Broadcasting Power	Strong (4 dBm)
Beacon Color	Mint Cocktail	Maximum Range	~70m / 230 ft
Tags	-	Advertising Interval	950 ms
Broadcasting Scheme	Estimote Default	Remaining Battery Life	24 months
Secure UUID	Off	Basic Battery Saving	Off
Proximity UUID	DF23 <input type="text"/>	Smart Battery Saving	On
Major	1 <input type="text"/> 29	Conditional Advertising	Off
Minor	4 <input type="text"/> 5	Firmware Version	A2.2.0
Mac Address	C2:89:A1 <input type="text"/>	Hardware Version	D3.4