

物聯網與微處理機系統設計

Internet of Things and Microprocessor System Design

Lecture 11 - Beacon Application

Lecturer: 陳彥安 Chen, Yan-Ann
YZU CSE

Outline

- BLE Beacon Application
- Eddystone Protocol
- iBeacon Protocol
- BLE Scanner

Outline

- BLE Beacon Application
- Eddystone Protocol
- iBeacon Protocol
- BLE Scanner

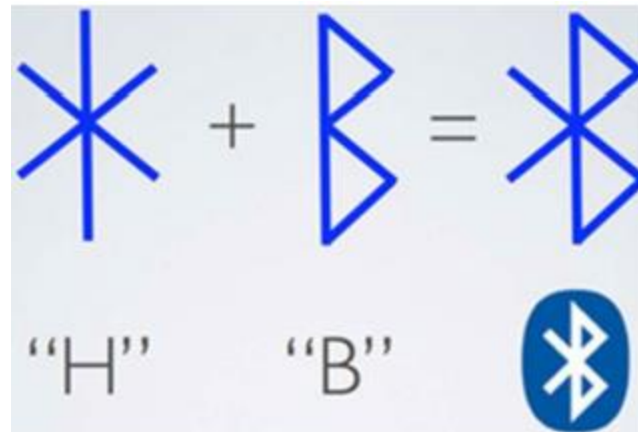
Bluetooth

- Objective
 - Communication between computer and electronic devices
- Features
 - Short-range wireless technology (10-100m)
 - Using 2.4 ~ 2.485 GHz [ISM Band](#)
- Bluetooth classic: 802.15.1
- Bluetooth 4.0 Low Energy (BLE): [Bluetooth SIG](#)



Bluetooth Origin

- Harald "Bluetooth" Gormsson was a king of Denmark and Norway.
- The development of the "short-link" radio technology, later named Bluetooth, was initiated in 1989 by Nils Rydbeck, CTO at [Ericsson Mobile](#) in Lund, Sweden.



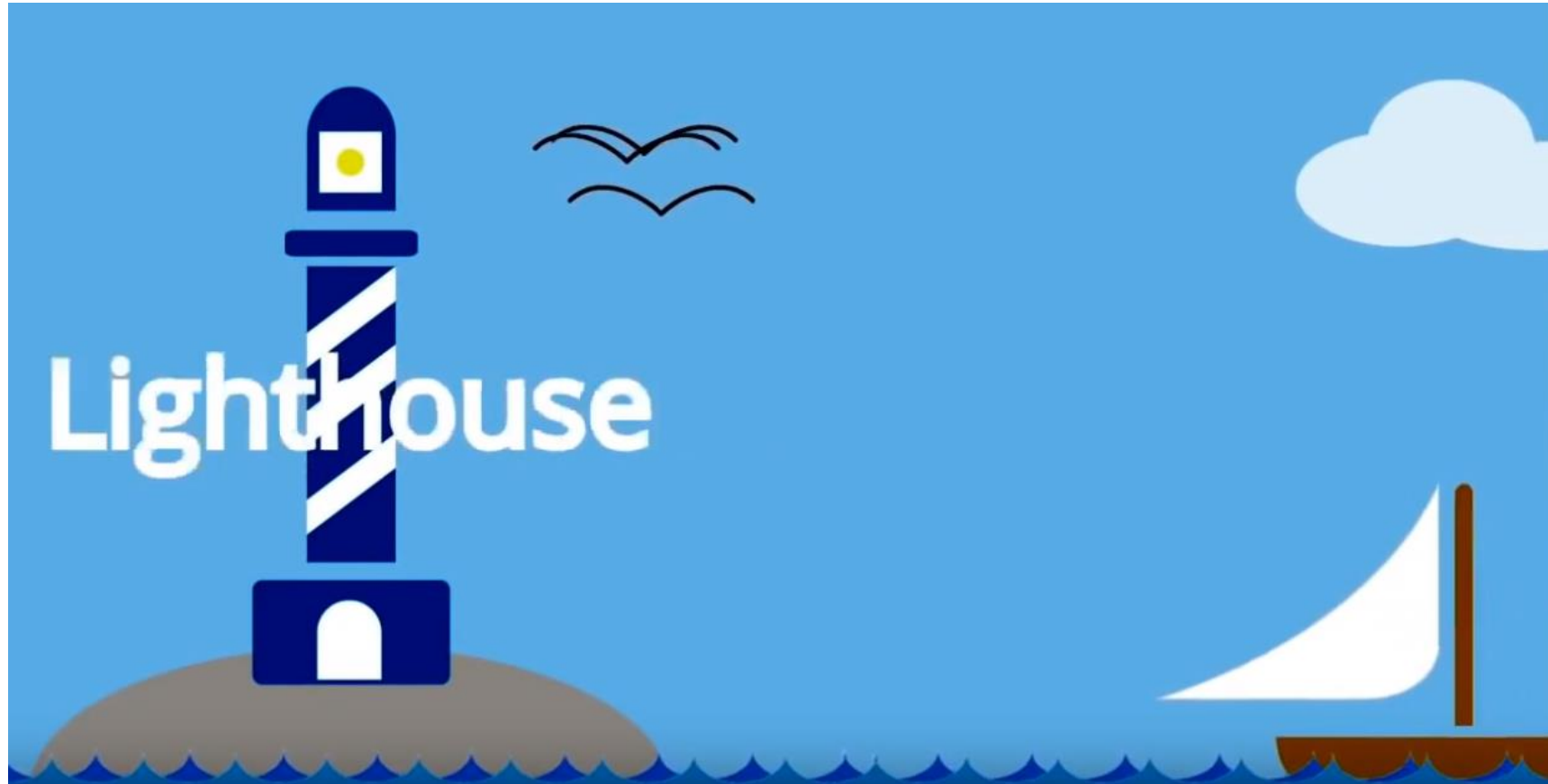
Bluetooth Low Energy (BLE)

- Bluetooth Low Energy is intended to provide considerably **reduced power consumption and cost** while maintaining a similar communication range.
- Aimed at novel applications in the healthcare, fitness, beacons, security, and home entertainment industries.
- Evolution: Classic (BR/EDR), High Speed (HS), Low Energy

| | Classic | BLE |
|-------------------|------------|--------------|
| Throughput | 2 ~ 3 Mbps | 0.2 Mbps |
| Range | 50 ~ 300 m | 10 ~ 30 m |
| Power consumption | 1 W | 0.01 ~ 0.5 W |
| Connection time | 5 s | 0.1 s |



What Is Beacon Technology ?



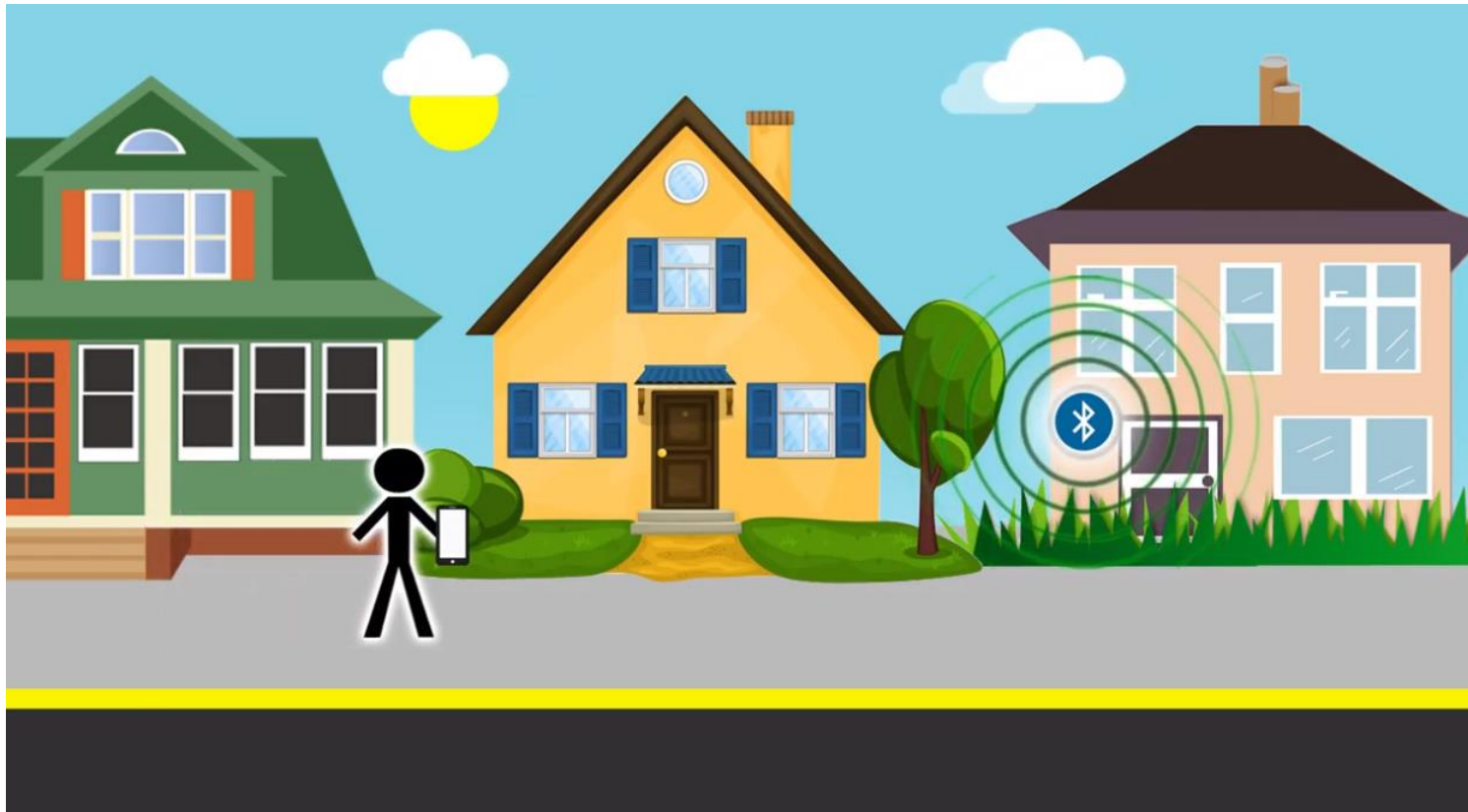
Ref: <https://www.youtube.com/watch?v=cvQO-kUMIVo>

What Is Beacon Technology ?



Ref: <https://www.youtube.com/watch?v=cvQO-kUMIVo>

Beacon In Real Estate



Ref: <https://www.youtube.com/watch?v=oQW1CQy3GUw>

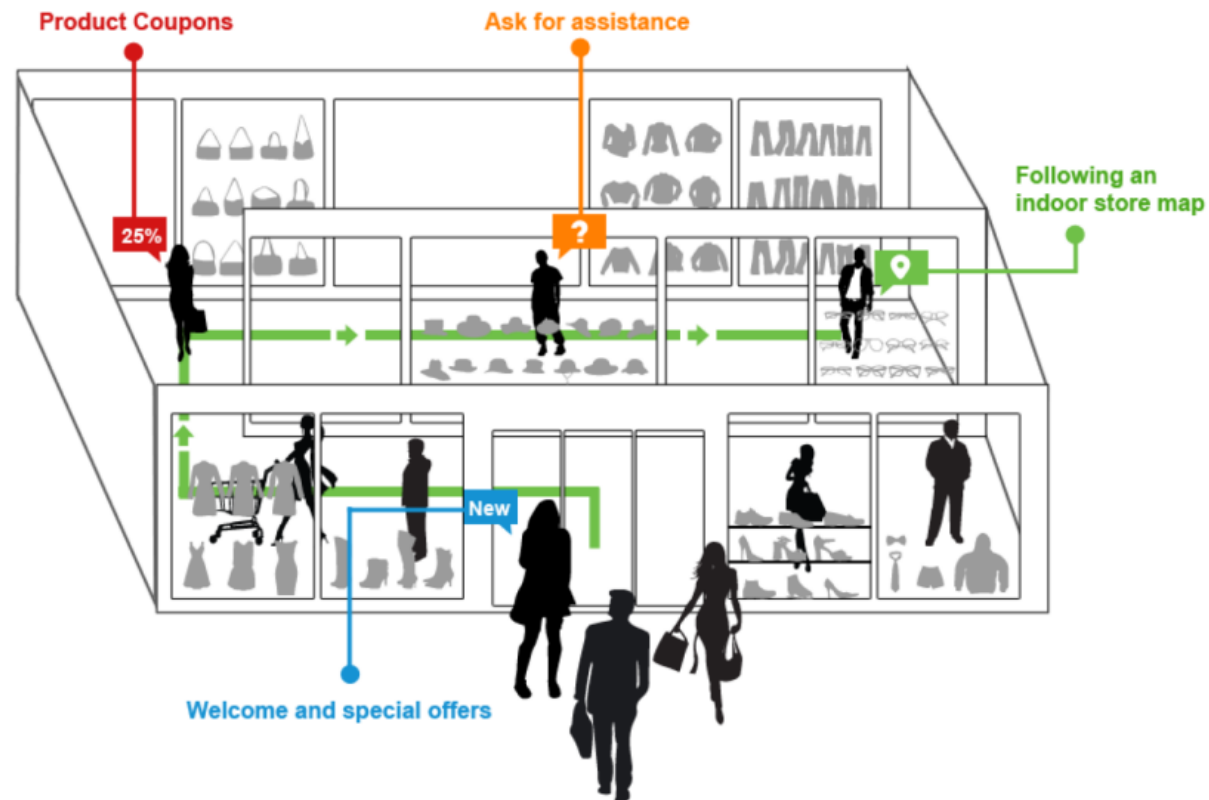
Beacon In Real Estate



Ref: <https://www.youtube.com/watch?v=oQW1CQy3GUw>

BLE Application

- Location-based service (LBS)
- Push message



Beacon x Taipei Main Station



台北車站通

ASKEY Computer Corporation 地圖與導航

★★★★★ 193

3+

含廣告內容

這個應用程式與你的部分裝置相容。

加入願望清單

安裝



Line Beacon x Taipei MRT



LINE Beacon是一個藍牙發射裝置，可以將優惠、特價資訊，或是特別情報發到您的手機中。對您來說，只要走到**LINE Beacon**的發送範圍，並且開啟手機藍牙，就可以收到**LINE Beacon**發給你的訊號囉！**LINE Beacon**可以結合許多時髦有趣的手機應用💕，在台灣人常去的日本，可是一點也不陌生！

偷偷預告一下，**LINE** 即將要在台北捷運玩一個超狂的踩點遊戲！（當然是有獎品的那種）屆時會透過**LINE Beacon**跟大家玩，所以現在我們要先做暖身練習，這樣到時候才不會漏接**LINE Beacon**發給你的訊號喔！能在連外國旅客都說讚的台北捷運玩，讓大台北捷運族優先試行，可是破天荒第一次啊！（**LINE Beacon**本人也表示感動🙏）



Beacon x Tainan City Hall



要到臺南市政府永華行政中心洽公的市民，現在透過洽公智慧導航系統的幫助，如要向市府申請補助，可以先下載「臺南洽公小幫手」的App，就可以搜尋衛生醫療、補助資訊、教育與求職相關的便民服務，這裡提供了洽辦單位的聯絡電話，並顯示了在行政大樓內的位置，還能透過室內導航，引導民眾直接前往。

Ref: <https://www.ithome.com.tw/news/116008>

Anti-Lost Alarm

藍牙追蹤 秒尋所有

防止家人及貴物品丟失



分享: [Facebook](#) [Twitter](#) [Google+](#) [Pinterest](#) [YouTube](#)

喜歡 (381)

商城 【限量優惠】智慧防丟器防丟貼片手機遺失報警寵物狗牌兒童尋物器自拍器自拍器防丟報器【國家認證】【C1-00049】

4.8 ★★★★★ 127 評價 431 已售出

\$380 **\$99** 2.6折

賣場折價券 現折\$20 現折\$30 現折\$40 現折\$60

蝦幣 下單現賺0蝦幣

運送 免運費 滿\$990, 免運費

運費 \$60 - \$90

規格 白色 黑色 藍色

數量 - 1 + 還剩2997件

加入購物車

直接購買

15天鑑賞期

退貨無負擔

假一賠二



B-Tag

B-Door



Ref: <https://www.youtube.com/watch?v=c5cF5TRP7kg>

AirTag

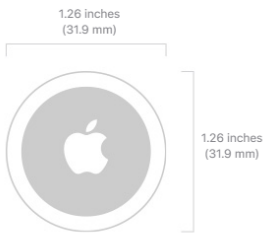
How does it work?

Your AirTag sends out a secure Bluetooth signal that can be detected by nearby devices in the Find My network. These devices send the location of your AirTag to iCloud — then you can go to the Find My app and see it on a map. The whole process is anonymous and encrypted to protect your privacy. And it's efficient, so there's no need to worry about battery life or data usage.



AirTag Tech Specs

Size and Weight



Size
Diameter: 1.26 inches (31.9 mm)
Height: 0.31 inch (8.0 mm)

Weight
0.39 ounce (11 grams)

Splash, Water, and Dust Resistance

Rated IP67 (maximum depth of 1 meter up to 30 minutes) under IEC standard 60529

Connectivity

Bluetooth for proximity finding
Apple-designed U1 chip for Ultra Wideband and Precision Finding
NFC tap for Lost Mode

Speaker

Built-in speaker

Battery

User-replaceable CR2032 coin cell battery

Sensor

Accelerometer

Accessibility

The Find My app is compatible with these iPhone accessibility features:

- VoiceOver
- Invert Colors
- Larger Text
- Compatibility with braille displays

System Requirements and Compatibility

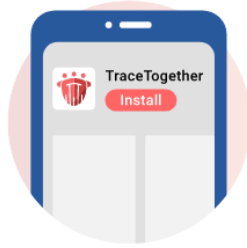
Apple ID
iPhone and iPod touch models with iOS 14.5 or later
iPad models with iPadOS 14.5 or later

Environmental Requirements

Operating ambient temperature: -4° to 140° F (-20° to 60° C)

TraceTogether

How to get TraceTogether?



Download the TraceTogether mobile app

Set up your profile, and help those around you to set up too.

or



Collect a physical TraceTogether Token

For those may not own or prefer not to use a mobile phone



Research

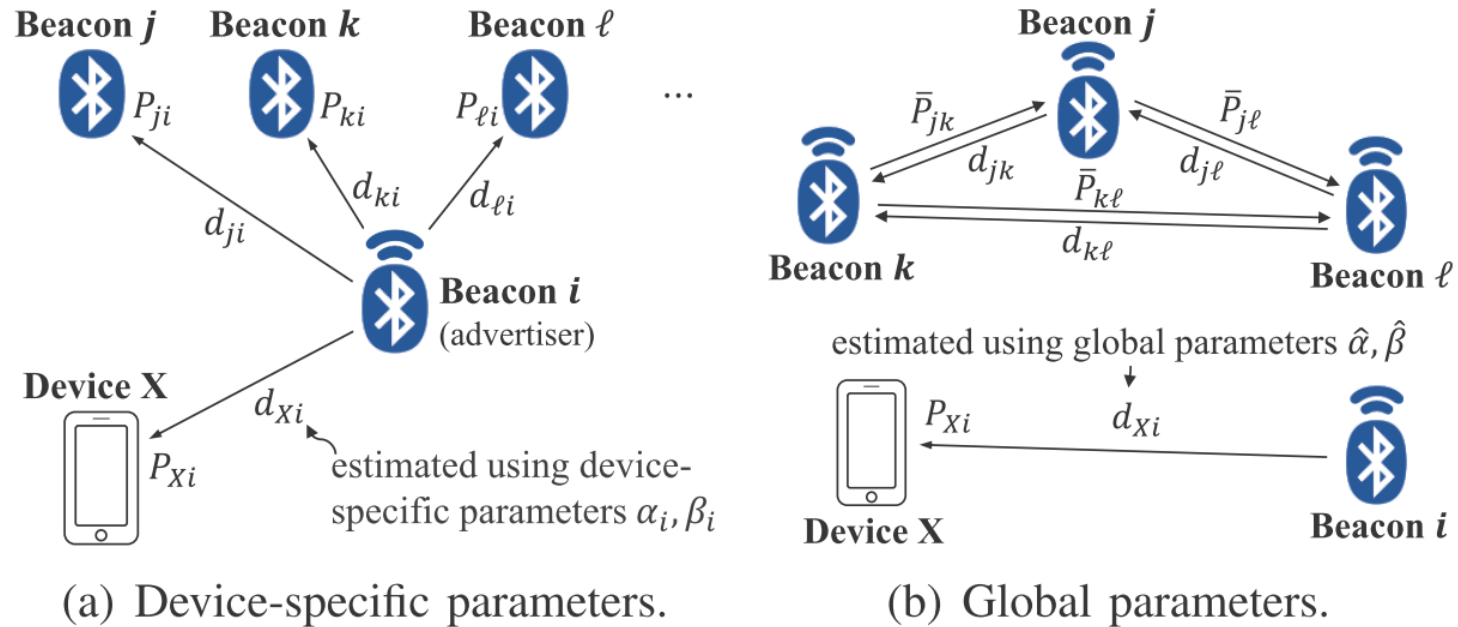


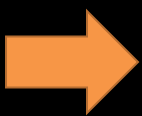
Fig. 2. Two types of regression parameters for adaptive ranging.

J.-W. Qiu, C.-P. Lin, and Y.-C. Tseng, "BLE-based Collaborative Indoor Localization with Adaptive Multi-iteration and Mobile Encountering", IEEE Wireless Communications and Networking Conference (WCNC), 2016.

Bluetooth on RPi

- Does your BT device support BLE? (“LE support”)

\$ hciconfig -a hci0 features



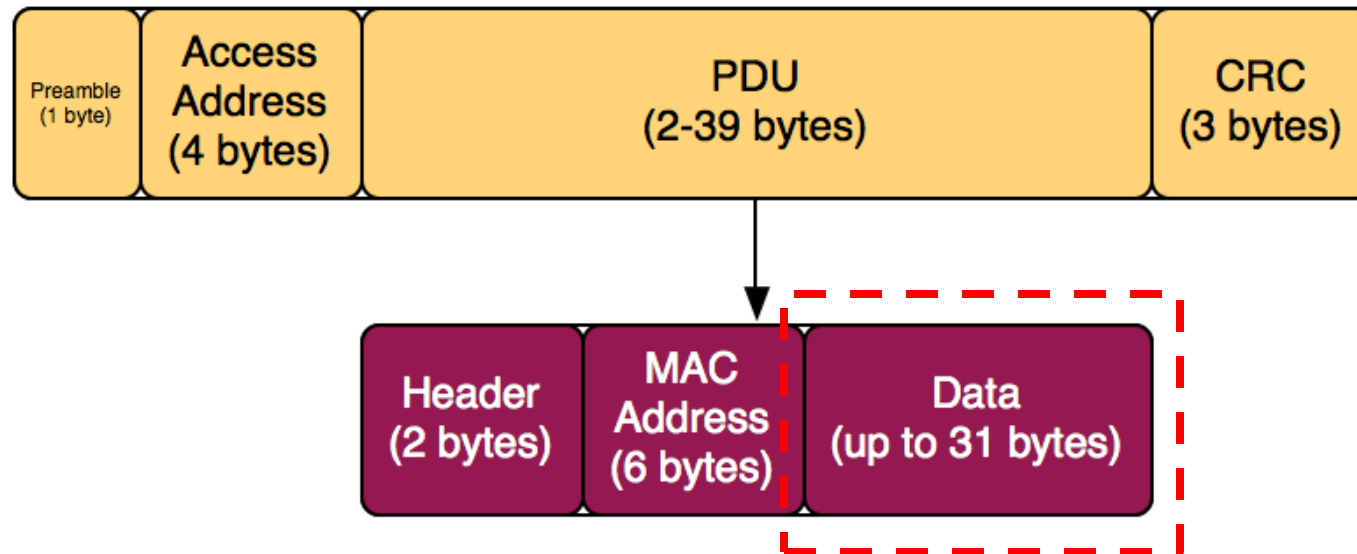
```
pi@rpi4-A00:~ $ hciconfig -a hci0 features
hci0:  Type: Primary  Bus: UART
      BD Address: DC:A6:32:9D:D8:92  ACL MTU: 1021:8  SCO MTU: 64:1
      Features page 0: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
          <3-slot packets> <5-slot packets> <encryption> <slot offset>
          <timing accuracy> <role switch> <sniff mode> <RSSI>
          <channel quality> <SCO link> <HV2 packets> <HV3 packets>
          <u-law log> <A-law log> <CVSD> <paging scheme> <power control>
          <transparent SCO> <broadcast encrypt> <EDR ACL 2 Mbps>
          <EDR ACL 3 Mbps> <enhanced iscan> <interlaced iscan>
          <interlaced pscan> <inquiry with RSSI> <extended SCO>
          <EV4 packets> <EV5 packets> <AFH cap. slave>
          <AFH class. slave> <LE support> <3-slot EDR ACL>
          <5-slot EDR ACL> <sniff subrating> <pause encryption>
          <AFH cap. master> <AFH class. master> <EDR eSCO 2 Mbps>
          <EDR eSCO 3 Mbps> <3-slot EDR eSCO> <extended inquiry>
          <LE and BR/EDR> <simple pairing> <encapsulated PDU>
          <err. data report> <non-flush flag> <LSTO> <inquiry TX power>
          <EPC> <extended features>
      Features page 1: 0x0b 0x00 0x00 0x00 0x00 0x00 0x00 0x00
      Features page 2: 0x7f 0x0b 0x00 0x00 0x00 0x00 0x00 0x00
```

Bluetooth Tools

- bluetoothctl - bluetooth control tool
- hciconfig - configure Bluetooth devices
- hcitool - configure Bluetooth connections
- l2ping - Send L2CAP echo request and receive answer
- btmon - Bluetooth monitor
- gatttool - GATT tool

BLE Frame Format

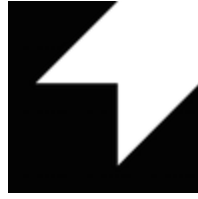
- 1 byte preamble
- 4 byte access address
- 2-39 bytes advertising channel PDU
- 3 bytes CRC



Outline

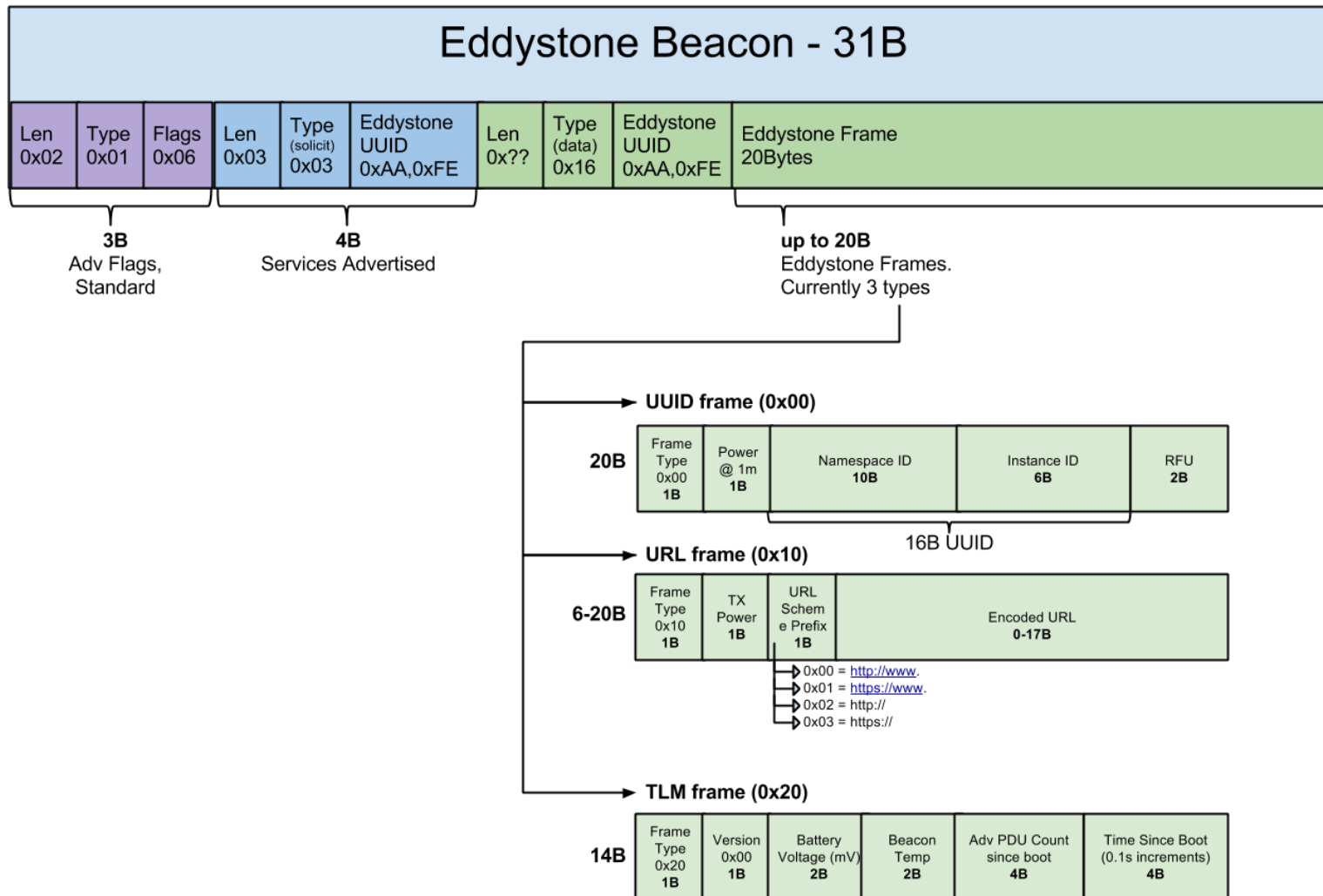
- BLE Beacon Application
- Eddystone Protocol
- iBeacon Protocol
- BLE Scanner

Eddystone

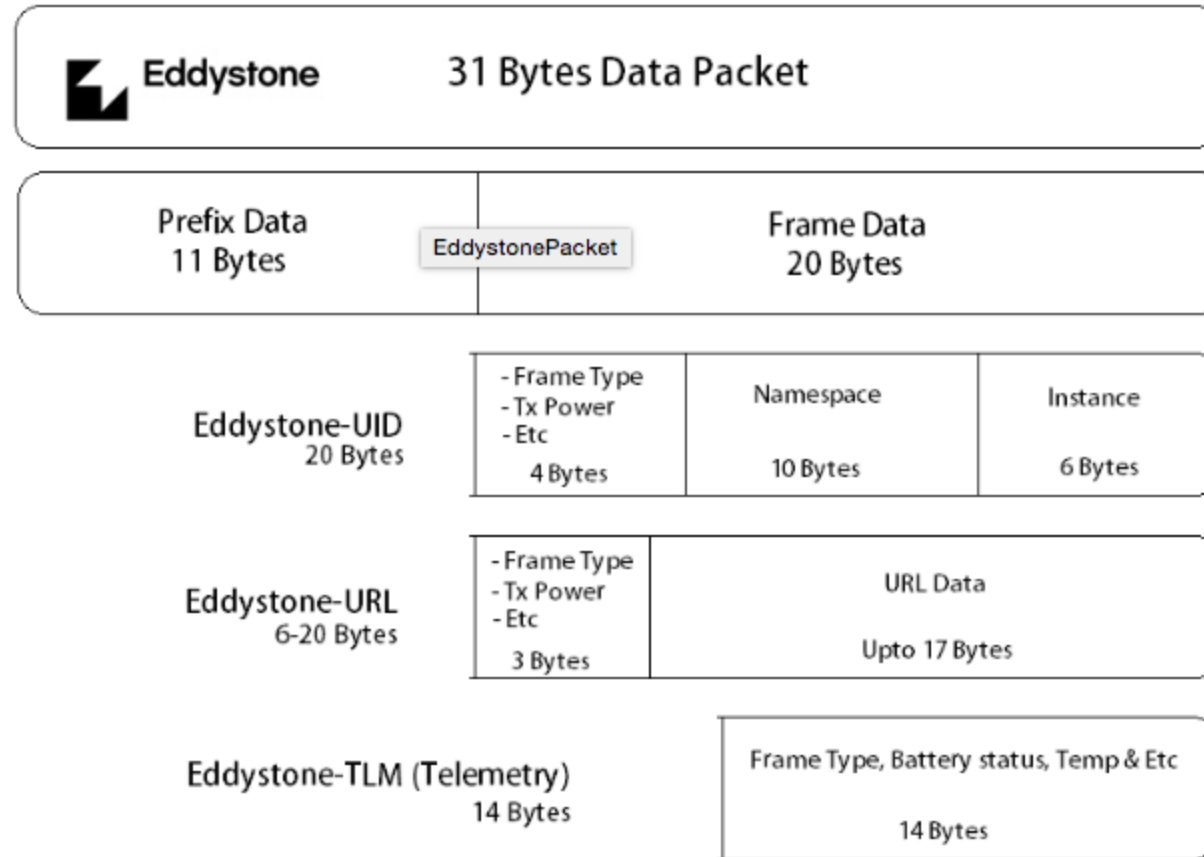


- Eddystone is a protocol specification that defines a **Bluetooth low energy (BLE) message format** for proximity beacon messages.
 - An open beacon format developed by **Google**.
- Design Goals
 - Works well with Android and iOS Bluetooth developer APIs
 - Straightforward implementation on a wide range of **existing BLE devices**
 - Flexible architecture permitting development of new frame types
 - Fully compliant with the Bluetooth Core Specification
- <https://github.com/google/eddystone>

Eddystone Frame Format

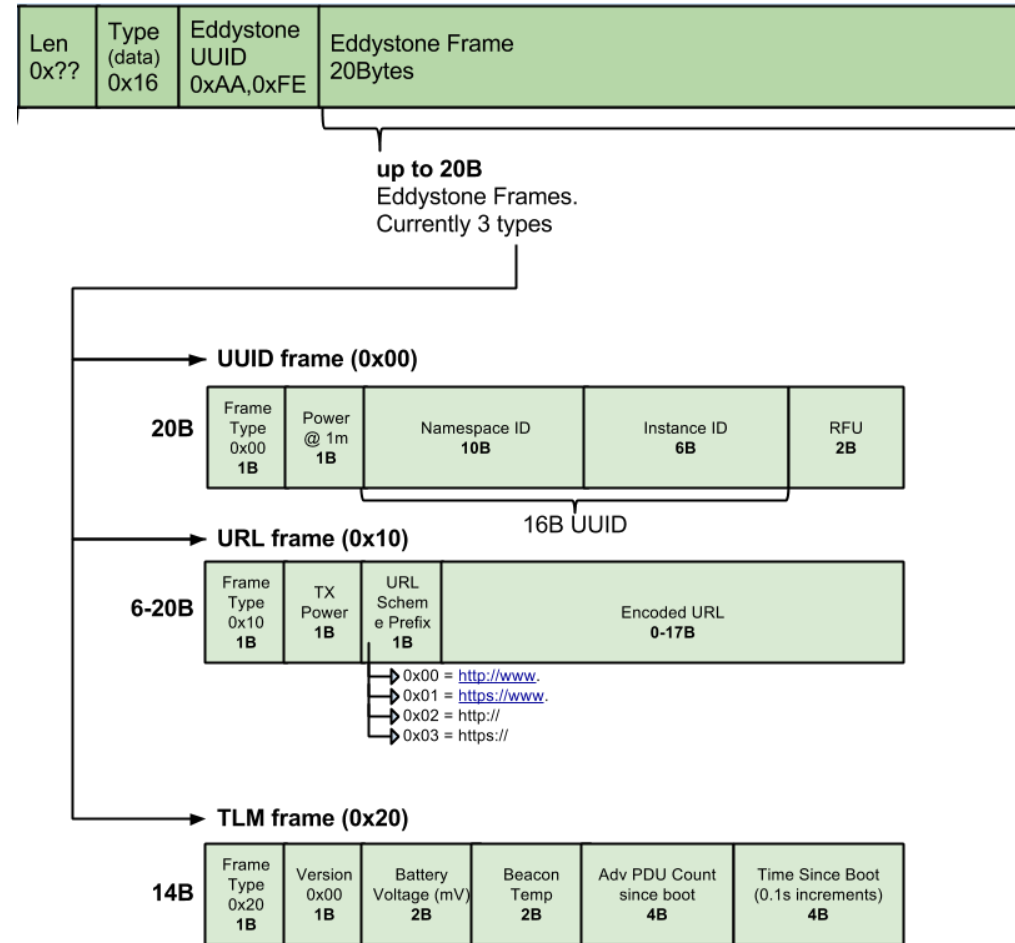


Eddystone Frame Format



Eddystone Frame Format

| Frame Type | High-Order 4 bits | Byte Value |
|------------|-------------------|------------|
| UID | 0000 | 0x00 |
| URL | 0001 | 0x10 |
| TLM | 0010 | 0x20 |
| EID | 0011 | 0x30 |
| RESERVED | 0100 | 0x40 |



<https://github.com/google/eddystone/blob/master/protocol-specification.md>

Eddystone

\$ wget https://raw.githubusercontent.com/google/eddystone/master/eddystone-url/implementations/linux/advertise-url

- advertise URL

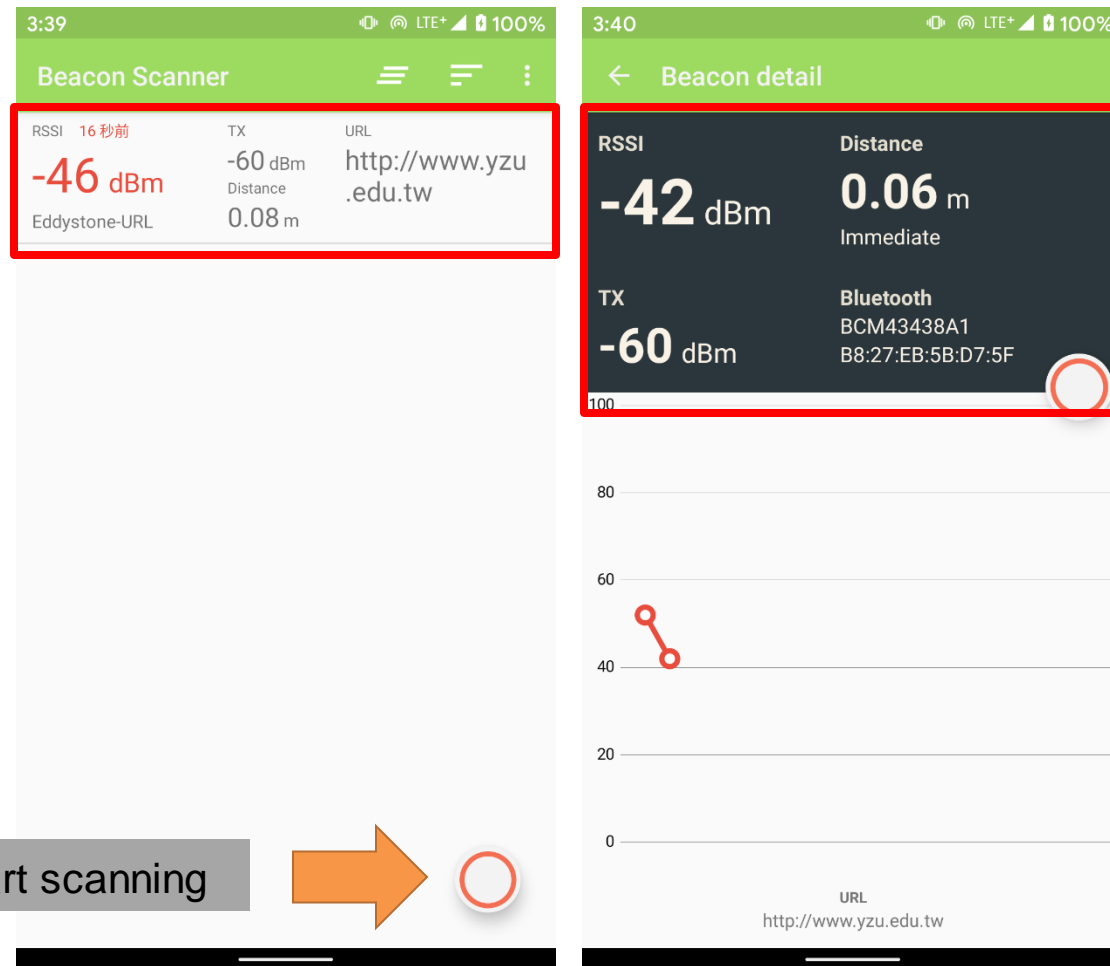
\$ chmod +x advertise-url

\$ sudo ./advertise-url -u http://www.yzu.edu.tw

```
pi@rpi4-A00:~/iot/lec12 $ sudo ./advertise-url -u http://www.yzu.edu.tw
Advertising: http://www.yzu.edu.tw
```

Eddystone Client

- Use Beacon scanner app to check the broadcast message.
 - <https://play.google.com/store/apps/details?id=de.flurp.beaconscanner.app>



Eddystone

- Stop advertising

```
$ sudo ./advertise-url -s
```

Eddystone

- Translate URL to Eddystone message
 - Ex: `http://ptt.cc`
 - [ASCII table](https://zh.wikipedia.org/wiki/ASCII) `https://zh.wikipedia.org/wiki/ASCII`

| Value (hex) | URL |
|-------------|---------|
| 02 | http:// |
| 70 | p |
| 74 | t |
| 74 | t |
| 2e | . |
| 63 | c |
| 63 | c |

| Decimal | Hex | Expansion |
|---------|------|--------------|
| 0 | 0x00 | http://www. |
| 1 | 0x01 | https://www. |
| 2 | 0x02 | http:// |
| 3 | 0x03 | https:// |

Eddystone

- LE advertising mode

```
$ sudo hciconfig hci0 leadv 3
```

- Disable scanning

```
$ sudo hciconfig hci0 noscan
```

<http://ptt.cc>



```
$ sudo hcitool -i hci0 cmd 0x08 0x0008 14 02 01 1a 03 03 aa fe 0c 16 aa fe 10 ed 02 70 74 74  
2e 63 63 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
pi@rpi4-A00:~/iot/lec12 $ sudo hcitool -i hci0 cmd 0x08 0x0008 14 02 01 1a 03 03  
aa fe 0c 16 aa fe 10 ed 02 70 74 74 2e 63 63 00 00 00 00 00 00 00 00 00 00 00  
< HCI Command: ogf 0x08, ocf 0x0008, plen 32  
14 02 01 1A 03 03 AA FE 0C 16 AA FE 10 ED 02 70 74 74 2E 63  
63 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
> HCI Event: 0x0e plen 4  
01 08 20 00
```

- Stop advertsing

```
$ sudo hciconfig hci0 noleadv
```

Eddystone by hcitool

```
sudo hcitool -i hci0 cmd 0x08 0x0008 14 02 01 1a 03 03 aa fe 0c 16 aa fe 10 ed 02 70 74 74 2e 63 63 00 00 00 00 00 00 00 00 00 00
```

- 0x08 0x0008: set the ad package
 - #OGF = Operation Group Field = Bluetooth Command Group = 0x08
 - #OCF = Operation Command Field = HCI_LE_Set_Advertising_Data = 0x0008
- 14: the ENTIRE following data packet in bytes
- 02 01 1a: Eddystone Adv Flags
 - 0x06 - The device is BLE only. The full Bluetooth stack is not supported.
 - 0x1A - The device can be used as BLE as well as full Bluetooth Controller/Host simultaneously.
- 03 03 aa fe: Eddystone service adv
- 0c: length (12 byte)
- 16: type (data)
- aa fe: Eddystone UUID
- 10: URL frame type
- ed: TX power
- 02 70 74 74 2e 63 63 : <http://ptt.cc>
- 00 00 00 00 00 00 00 00 00 00 00 00

Lab 11-1

- Advertise an URL with your student ID.
 - Ex: yzu.5566

| Beacon Scanner | | | |
|----------------|-----------|----------|-----------------|
| RSSI | Immediate | TX | URL |
| -45 | dBm | -60 dBm | http://yzu.5566 |
| Eddystone-URL | | Distance | 0.11 m |

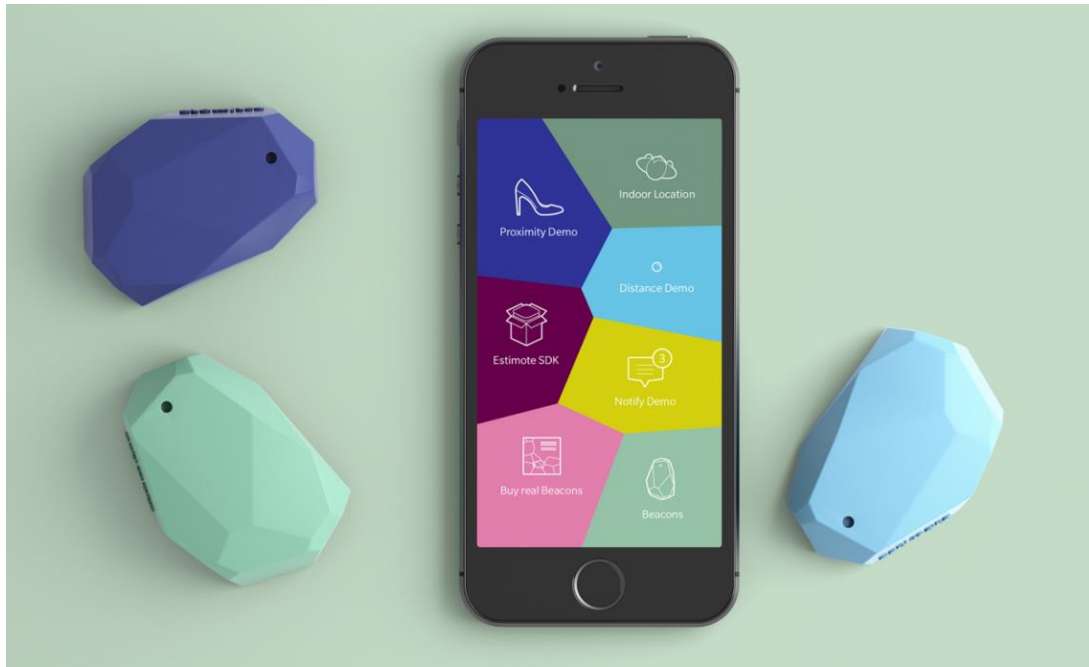
- Demonstrate your result to the TA.

Outline

- BLE Beacon Application
- Eddystone Protocol
- iBeacon Protocol
- BLE Scanner

iBeacon

- <https://developer.apple.com/ibeacon/>

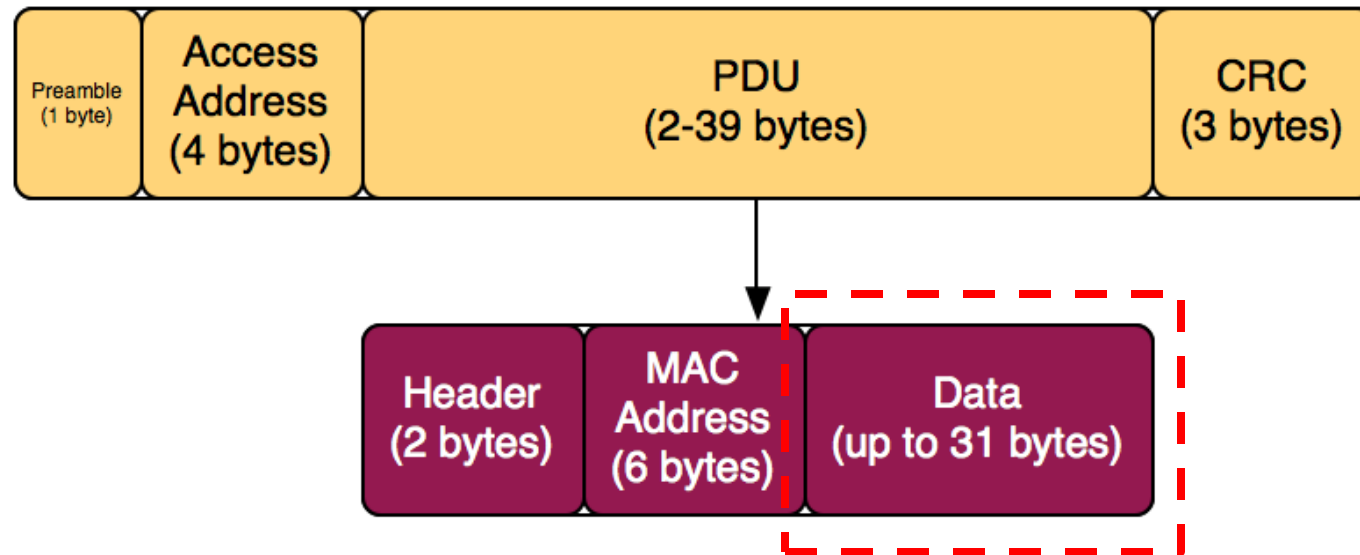


iBeacon

From welcoming people as they arrive at a sporting event to providing information about a nearby museum exhibit, iBeacon opens a new world of possibilities for location awareness, and countless opportunities for interactivity between iOS devices and iBeacon hardware.

BLE Frame Format

- 1 byte preamble
- 4 byte access address
- 2-39 bytes advertising channel PDU
- 3 bytes CRC

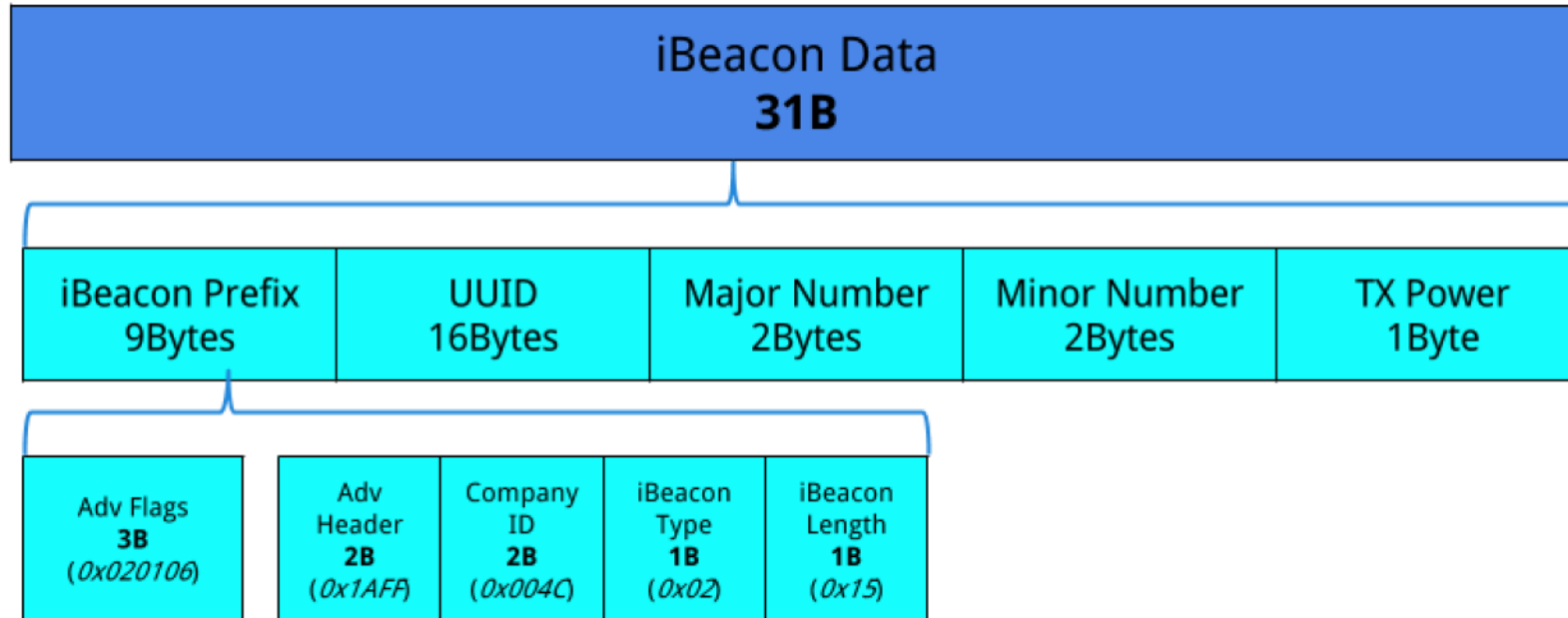


Ref: <http://www.warski.org/blog/2014/01/how-ibeacons-work/>

iBeacon Format



iBeacon



iBeacon Tool

- <https://github.com/dburr/linux-ibeacon>

- Dependencies

```
$ sudo apt-get install bluetooth bluez blueman
```

```
$ git clone https://github.com/dburr/linux-ibeacon
```

```
$ cd linux-ibeacon/
```

```
$ chmod +x ibeacon
```

iBeacon

- What is UUID?
 - UUID stands for **Universally Unique Identifier**. It contains 32 hexadecimal digits, split into 5 groups, separated by hyphens and should look something like this:
f7826da6-4fa2-4e98-8024-bc5b71e0893e
- Each of the **5 groups** must contain the following number of characters per section:
 - First section: 8
 - Second section: 4
 - Third section: 4
 - Fourth section: 4
 - Fifth section: 12

Start Advertising

\$ sudo ./ibeacon -M 5566 -m 7788

- Major ID = 5566; Minor ID = 7788

```
pi@rpi4-A00:~/iot/lec12/linux-ibeacon $ sudo ./ibeacon -M 5566 -m 7788
Advertising on hci0 with:
  uuid: 0xE20A39F473F54BC4A12F17D1AD07A961
major/minor: 5566/7788 (0x15BE/0x1E6C)
  power: 200 (0xC8)
```

| Beacon Scanner | | | |
|----------------|----------|--------------|-------|
| RSSI 18 秒前 | TX | UUID | |
| -58 dBm | -56 dBm | 17d1ad07a961 | |
| iBeacon | Distance | Major | Minor |
| | 1.09 m | 5566 | 7788 |

\$ sudo ./ibeacon -z

- Stop

```
pi@rpi4-A00:~/iot/lec12/linux-ibeacon $ sudo ./ibeacon -z
Downing iBeacon on hci0
```

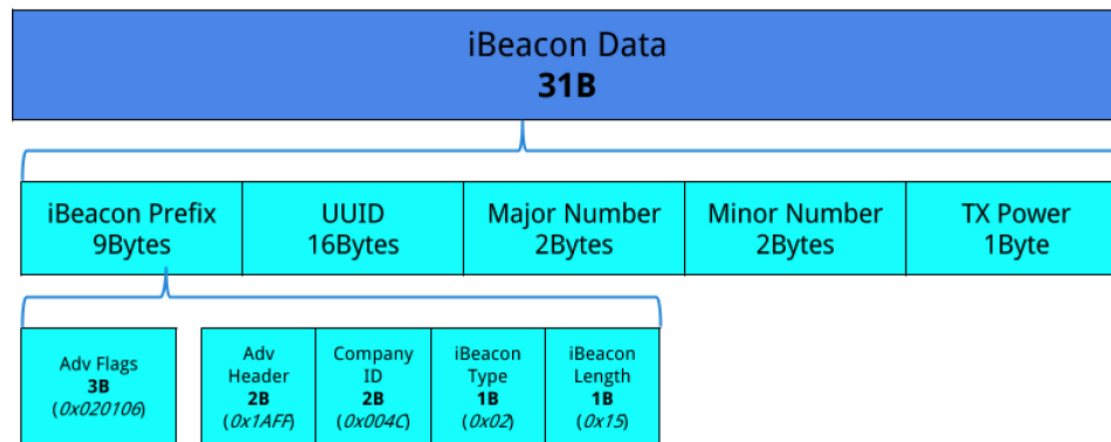

Advertising by hciconfig

```
$ sudo hciconfig hci0 up
```

```
$ sudo hciconfig hci0 leadv 3
```

```
$ sudo hciconfig hci0 noscan
```

```
$ sudo hcidtool -i hci0 cmd 0x08 0x0008 1E 02 01 06 1A FF 00 4C 02 15 E2 0A 39 F4 73 F5 4B  
C4 A1 2F 17 D1 AD 07 A9 61 00 00 00 00 C8 00
```



Ref: <https://blog.gtwang.org/iot/raspberry-pi/diy-build-raspberry-pi-ibeacon-transmitter-tutorial/>

iBeacon by hcitool

```
sudo hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 06 1A FF 00 4C 02 15 E2 0A 39 F4 73 F5 4B C4 A1 2F 17 D1 AD 07 A9 61 00 00 00 00 C8 00
```

- 0x08 0x0008: set the ad package
 - #OGF = Operation Group Field = Bluetooth Command Group = 0x08
 - #OCF = Operation Command Field = HCI_LE_Set_Advertising_Data = 0x0008
- 1E // the ENTIRE following data packet in bytes (31 byte)
- 02 01 06 // set the flags for General Discoverable and BR/EDR not supported
- 1A FF // the length of the Manufacturer specific data field will be 26 bytes
- 00 4C // Company ID
- 02 // iBeacon type, ID
- 15 // length of remaining data in bytes (16B UUID+ 2B major, 2B minor, 1B Txpower)
- E2 0A 39 F4 73 F5 4B C4 A1 2F 17 D1 AD 07 A9 61 // UUID
- 00 00 // Major ID
- 00 00 // Minor ID
- C8 // Tx power

Lab 11-2


- Advertise with your student ID.
 - UUID should contain your student ID
 - Ex: 3345678

| Beacon Scanner | | | | |
|----------------|-----------|----------|--------------|-------|
| RSSI | Immediate | TX | UUID | |
| -48 dBm | | -56 dBm | 334567800000 | |
| | | Distance | Major | Minor |
| iBeacon | | 0.22 m | 0 | 0 |

- Demonstrate your results to the TA.

5:06

LTE 100%

 Beacon detail

RSSI

-48 dBm

Distance

0.22 m

Immediate

TX

-56 dBm

Bluetooth

BCM43438A1

B8:27:EB:5B:D7:5F

100

80

60

40

20

0

UUID

e20a39f4-73f5-4bc4-
a12f-334567800000

Major

0

Minor

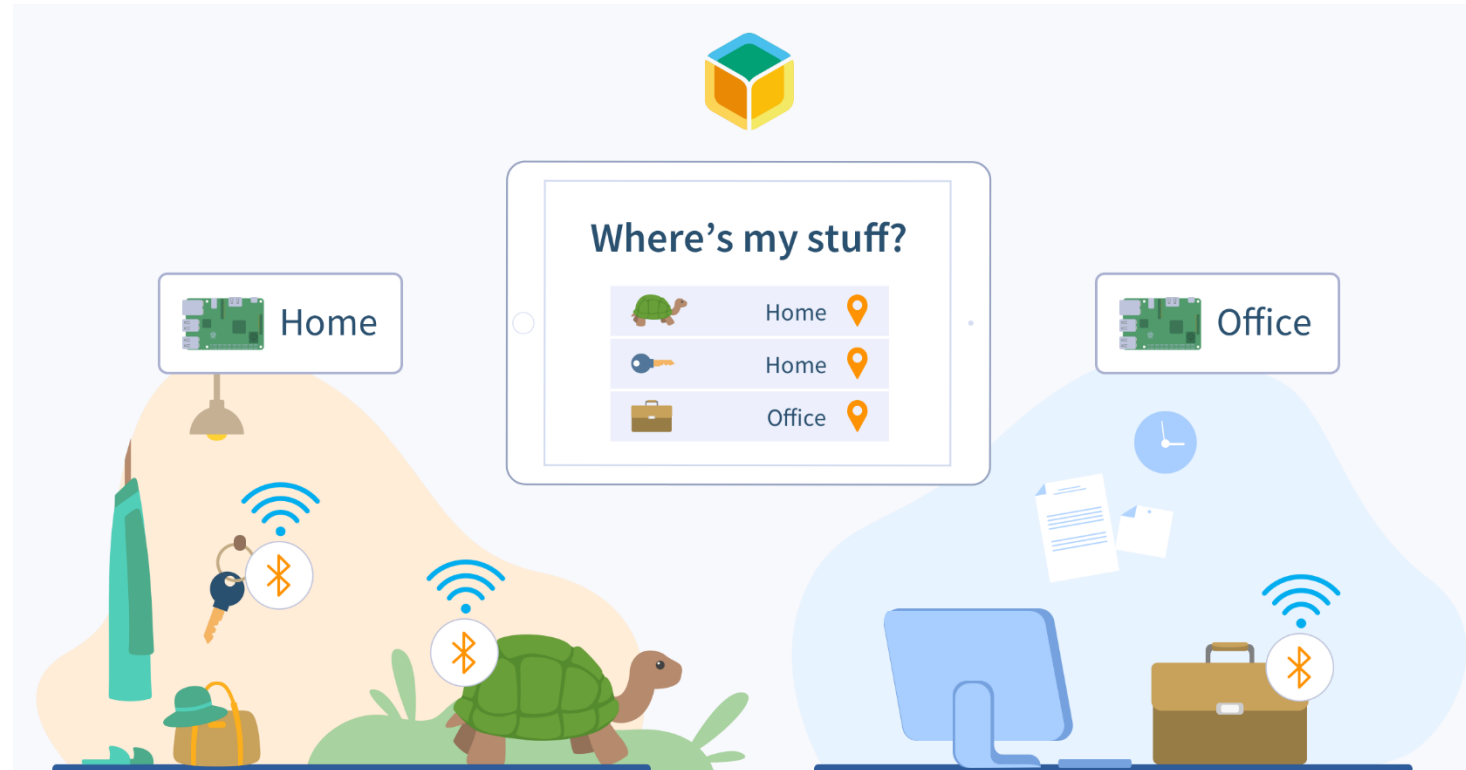
0

Outline

- BLE Beacon Application
- Eddystone Protocol
- iBeacon Protocol
- BLE Scanner

BLE Scanner

- Make RPi scan beacon info.
- Bluetooth LE interface for Python
 - `bluepy` is a Python module which allows communication with Bluetooth Low Energy devices.



Ref: <https://www.balena.io/blog/tracking-bluetooth-beacons-with-a-raspberry-pi/>

Beacon Simulator

- <https://play.google.com/store/apps/details?id=net.alea.beaconsimulator>
- Install this APP to make your phone as a BLE beacon.



Beacon Simulator

Vincent Hiribarren 工具

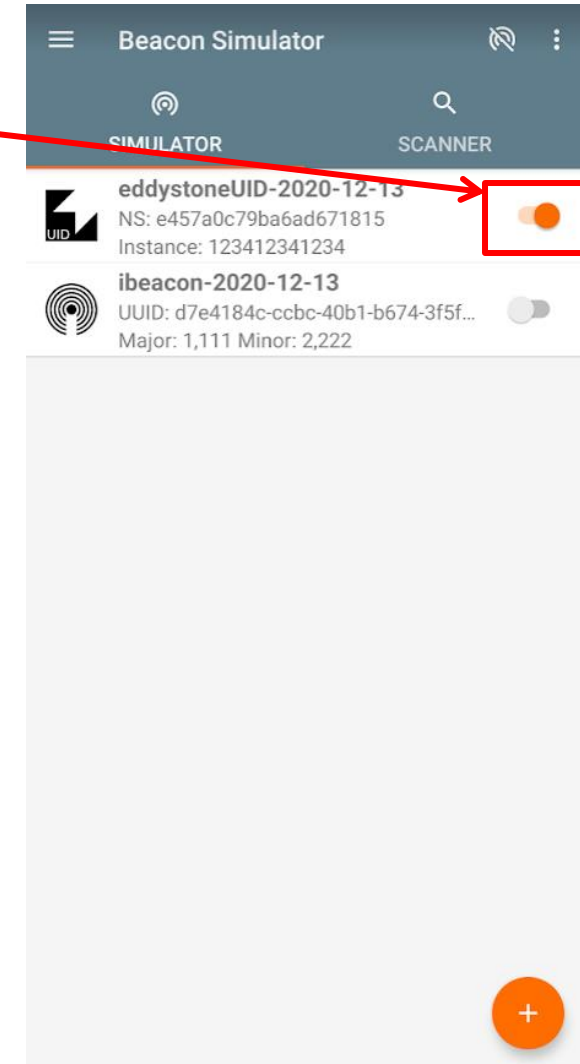
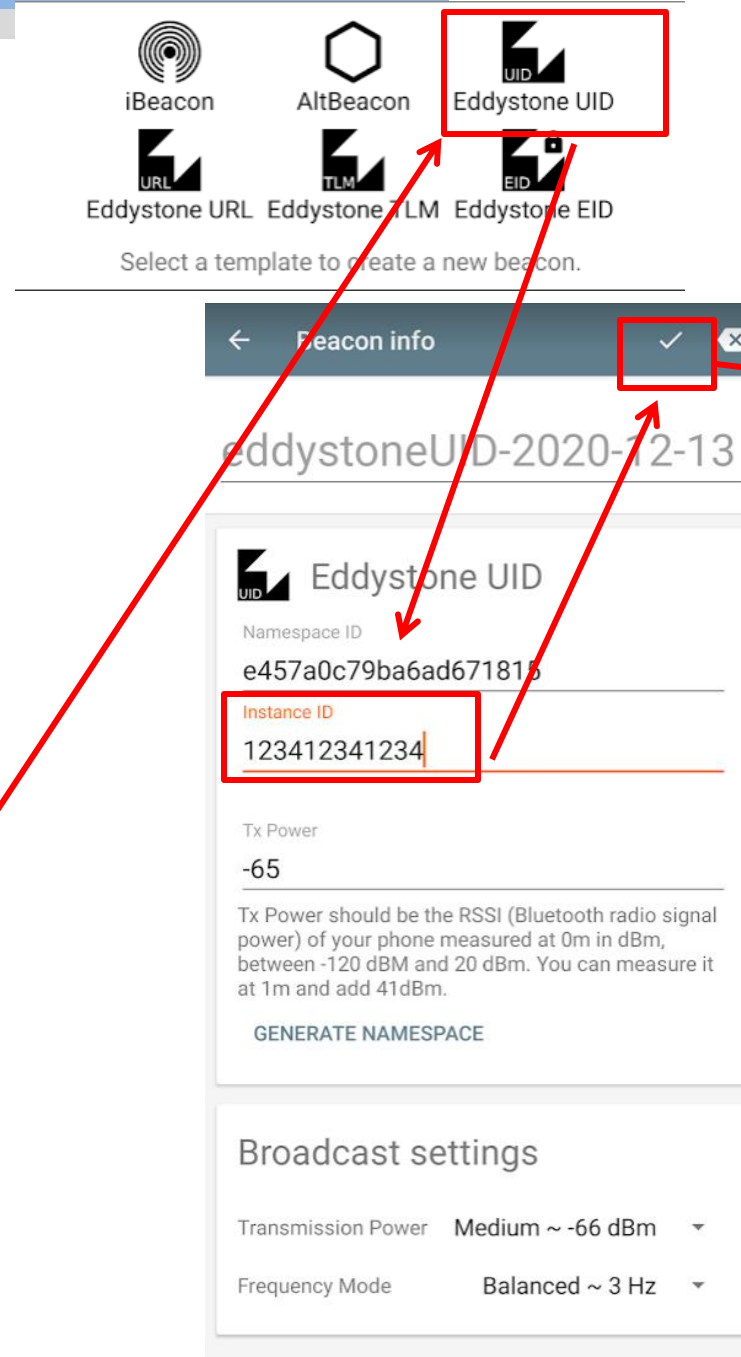
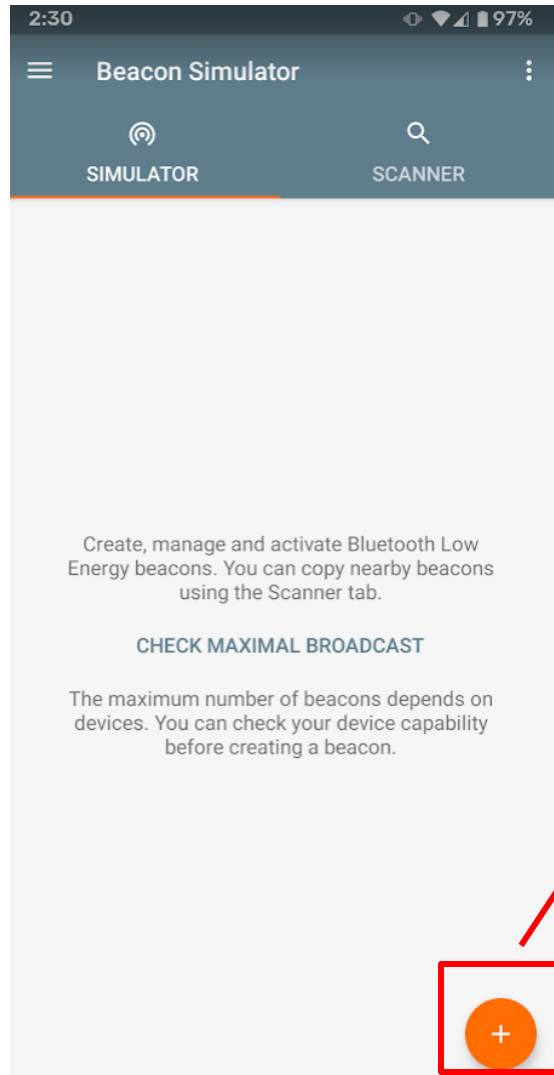
3+

★★★★★ 276 人

加入願望清單

安裝

Eddystone



RPi Scanner

- <http://ianharvey.github.io/bluepy-doc/scanner.html#sample-code>

\$ wget https://raw.githubusercontent.com/yachentw/yzucseiot/main/lec11/scanner.py

- scanner.py

```
from bluepy.btle import Scanner, DefaultDelegate

class ScanDelegate(DefaultDelegate):
    def __init__(self):
        DefaultDelegate.__init__(self)

    def handleDiscovery(self, dev, isNewDev, isNewData):
        if isNewDev:
            print("Discovered device", dev.addr)
        elif isNewData:
            print("Received new data from", dev.addr)

scanner = Scanner().withDelegate(ScanDelegate())
devices = scanner.scan(10.0)

for dev in devices:
    print("Device %s (%s), RSSI=%d dB" % (dev.addr, dev.addrType, dev.rssi))
    for (adtype, desc, value) in dev.getScanData():
        print("  %s = %s" % (desc, value))
```



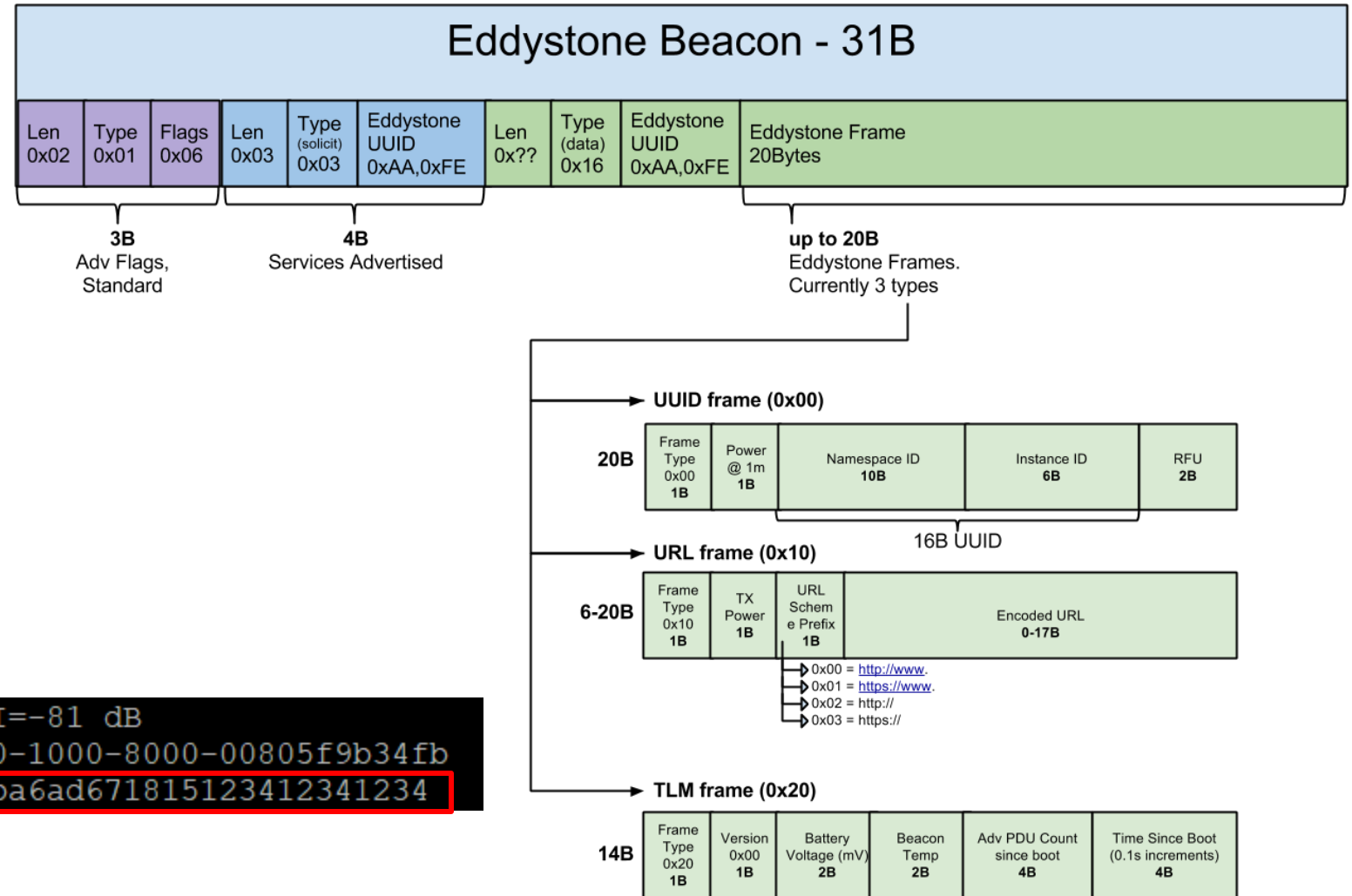
```
$ sudo pip3 install bluepy
```

```
$ sudo reboot
```

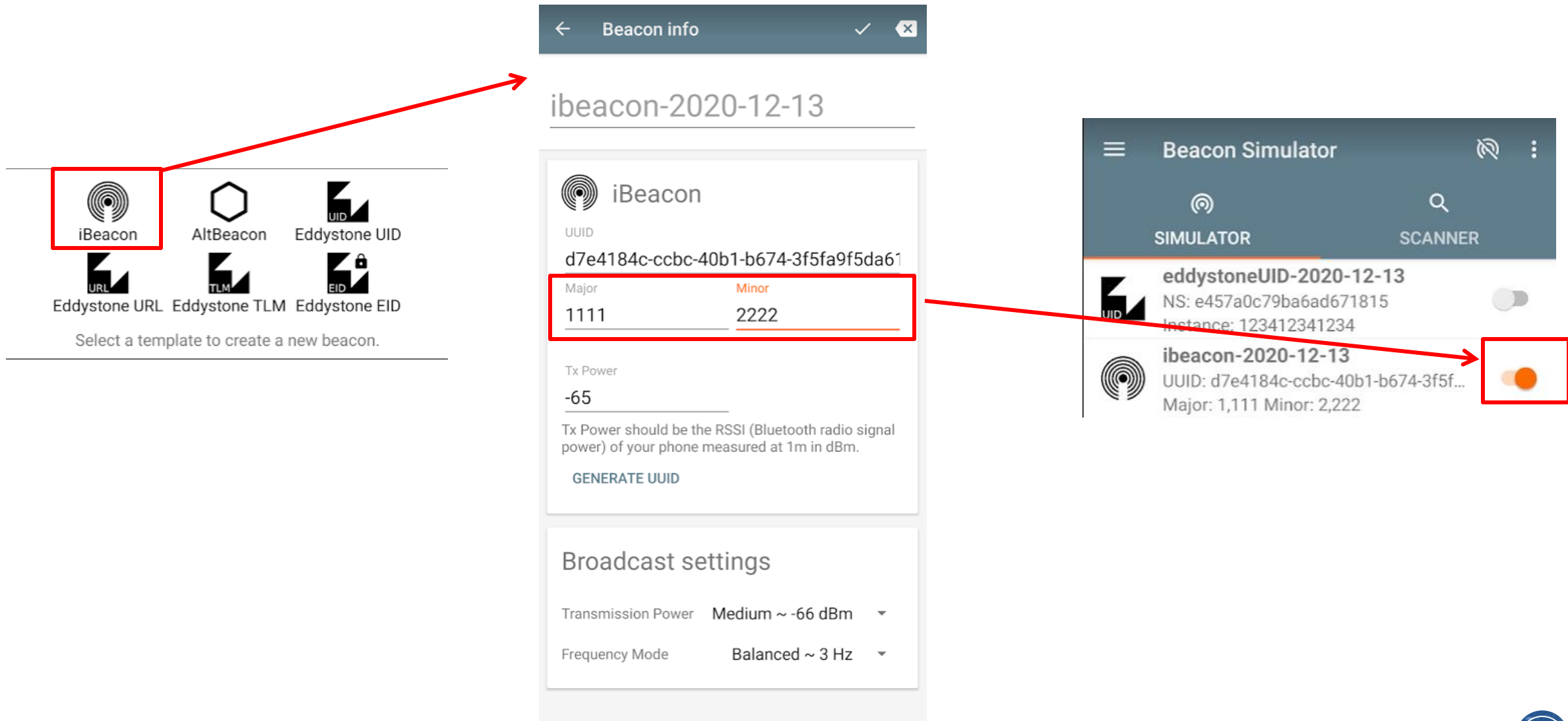
```
$ sudo python3 scanner.py
```

```
Discovered device 28:b8:54:de:4e:6c
Discovered device 42:10:1d:a3:f1:6c
Discovered device 73:ee:37:a7:6d:29
Discovered device 50:d0:d0:a7:cb:26
Discovered device 58:cb:52:50:ee:fb
Discovered device 5d:67:6d:02:37:48
Discovered device f4:fe:fb:bb:4e:d8
```

```
Device 58:cb:52:50:ee:fb (public), RSSI=-81 dB
Complete 16b Services = 0000feaa-0000-1000-8000-00805f9b34fb
16b Service Data = aafe00bfe457a0c79ba6ad671815123412341234
```



iBeacon



The image shows a sequence of three screenshots from the iBeacon app, illustrating the process of creating and simulating a beacon.

Left Screenshot: Template Selection

Select a template to create a new beacon.

- iBeacon (selected)
- AltBeacon
- Eddystone UID
- Eddystone URL
- Eddystone TLM
- Eddystone EID

Middle Screenshot: Beacon info

ibeacon-2020-12-13

iBeacon

UUID
d7e4184c-ccbc-40b1-b674-3f5fa9f5da61

Major: 1111 Minor: 2222

Tx Power: -65

Tx Power should be the RSSI (Bluetooth radio signal power) of your phone measured at 1m in dBm.

GENERATE UUID

Broadcast settings

Transmission Power: Medium ~ -66 dBm

Frequency Mode: Balanced ~ 3 Hz

Right Screenshot: Beacon Simulator

Beacon Simulator

SIMULATOR | SCANNER

eddstoneUID-2020-12-13
NS: e457a0c79ba6ad671815
Instance: 123412341234

ibeacon-2020-12-13
UUID: d7e4184c-ccbc-40b1-b674-3f5f...
Major: 1,111 Minor: 2,222

RPi Scanner

\$ sudo python3 scanner.py

```
Device 58:cb:52:50:ee:fb (public), RSSI=-94 dB
Manufacturer = 4c000215d7e4184ccbc40b1b6743f5fa9f5da61045708aebf
```

← Beacon info ✓ ✕

ibeacon-2020-12-13



iBeacon

UUID

d7e4184c-ccbc-40b1-b674-3f5fa9f5da61

Major

1111

Minor

2222

Tx Power

-65

Tx Power should be the RSSI (Bluetooth radio signal power) of your phone measured at 1m in dBm.

GENERATE UUID

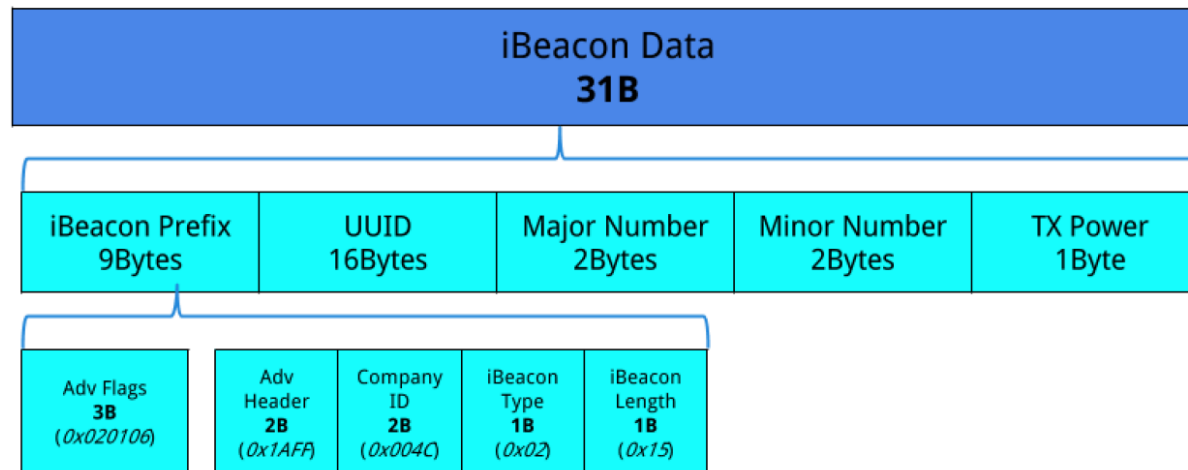
Broadcast settings

Transmission Power Medium ~ -66 dBm ▾

Frequency Mode Balanced ~ 3 Hz ▾

uuid

1111 2222



Lab 11-3

- Modify the code to trace your smartphone only.
- You will get a higher RSSI when the phone is approaching to your RPi.
- Ex:

```
(venv) pi@rpi4-A00:~/iot/lec12 $ sudo python3 target.py
RSSI: -79
RSSI: -84
RSSI: -90
RSSI: -89
RSSI: -76
RSSI: -83
RSSI: -71
RSSI: -59
RSSI: -56
RSSI: -47
RSSI: -45
RSSI: -45
```