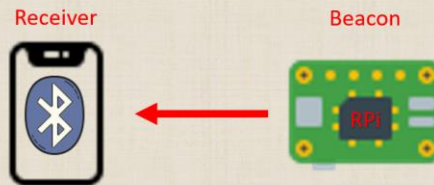


Q1: 用 `advertise-url` 指令產生 Eddystone 的廣告訊息(請參考 PPT 第 19 頁)

- 廣告訊息格式為 <http://www.nycu.edu.tw/xx>
  - xx = 自己的學號後兩碼
- 再利用手機 App 抓取 Eddystone 廣告訊息並截圖附在結報內



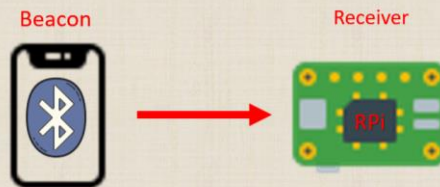
距離: 0.53m

RSSI: -56dBm

URL: <http://www.nycu.edu.tw/62>

Q2: 由手機擔任 Beacon，RPi 利用接收到的 RSSI、TX power 計算出傳輸距離

- 根據助教提供的 **Lab1.py**，並自行完成
- 接收到 RSSI 跟 TX power 後計算出距離，並按照公式輸出距離在 terminal (PPT 第 22 頁)
  - UUID = 00000000000000000000000000000000xxxxxxxx
  - 請將 **xxxxxxxx** 部分改成自己學號，UUID 共有 32 碼
- 使用 AltBeacon 中 Nexus 4 model 的參數 (PPT 第 23 頁)



※請用 `sudo python Lab1.py` 來執行 Q2 程式，提供權限讓 python 能使用藍芽模組

```
-----
('raw_uuid', '00000000000000000000000000000811562')
('uuid:', '00000000-0000-0000-0000-0000000811562')
('major:', '0', ', minor:', '0', ', txpower:', '-59')
('rssi', '-42')
('distance (m)', 0.03341741003670675)
-----
('raw_uuid', '00000000000000000000000000000811562')
('uuid:', '00000000-0000-0000-0000-0000000811562')
('major:', '0', ', minor:', '0', ', txpower:', '-59')
('rssi', '-42')
('distance (m)', 0.03341741003670675)
-----
('raw_uuid', '00000000000000000000000000000811562')
('uuid:', '00000000-0000-0000-0000-0000000811562')
('major:', '0', ', minor:', '0', ', txpower:', '-59')
('rssi', '-58')
('distance (m)', 0.842868491626227)
-----
('raw_uuid', '00000000000000000000000000000811562')
('uuid:', '00000000-0000-0000-0000-0000000811562')
('major:', '0', ', minor:', '0', ', txpower:', '-59')
('rssi', '-70')
('distance (m)', 1.9304364502555003)
-----
('raw_uuid', '00000000000000000000000000000811562')
('uuid:', '00000000-0000-0000-0000-0000000811562')
('major:', '0', ', minor:', '0', ', txpower:', '-59')
('rssi', '-59')
('distance (m)', 0.97085)
```



### 3. 本次實驗心得，你學到了什麼東西？

```
def encodeurl(url):
    i = 0
    data = []

    for s in range(len(schemes)):
        scheme = schemes[s]
        if url.startswith(scheme):
            data.append(s)
            i += len(scheme)
            break
    else:
        raise Exception("Invalid url scheme")

    while i < len(url):
        if url[i] == '.':
            for e in range(len(extensions)):
                expansion = extensions[e]
                if url.startswith(expansion, i):
                    data.append(e)
                    i += len(expansion)
                    break
            else:
                data.append(0x2E)
                i += 1
        else:
            data.append(ord(url[i]))
            i += 1

    return data

def encodeMessage(url):
    encodedurl = encodeurl(url)
    encodedurlLength = len(encodedurl)

    verboseOutput("Encoded url length: " + str(encodedurlLength))

    if encodedurlLength > 18:
        raise Exception("Encoded url too long (max 18 bytes)")

    message = [
        0x02, # Flags length
        0x01, # Flags data type value
        0x1a, # Flags data

        0x03, # Service UUID length
        0x03, # Service UUID data type value
        0xaa, # 16-bit Eddystone UUID
        0xfe, # 16-bit Eddystone UUID

        5 + len(encodedurl), # Service Data length
        0x16, # Service Data data type value
        0xaa, # 16-bit Eddystone UUID
        0xfe, # 16-bit Eddystone UUID

        0x10, # Eddystone-url frame type
        0xed, # txpower
    ]

    message += encodedurl

    return message
```

```
schemes = [
    "http://www.",
    "https://www.",
    "http://",
    "https://",
]

extensions = [
    ".com/", ".org/", ".edu/", ".net/", ".info/", ".biz/", ".gov/",
    ".com", ".org", ".edu", ".net", ".info", ".biz", ".gov",
]
```

很快的 trace 了一下 advertise-url 的 code，首先先將網址中比較特殊的字串，ex: “http://www.”， “.edu”做編碼，剩餘的文字再透過 ASCII 轉成 16 進位。因此對於我這次使用的 url 編碼後的結果為:

Encode string	Value in hexadecimal
http://www.	0x00
nycu	0x61 0x79 0x63 0x75
.edu	0x09
.	0x2e
tw/	0x74 0x77 0x2f
62	0x36 0x32

接著再使用 Eddystone Protocol Specification 的規則將其放入正確的位置，整個過程不難，而且寫的很簡潔，很值得學習!這次的實驗還蠻好玩的，也非常簡單，沒想到可以透過這麼簡單的操作及設定就能夠讓兩台裝置連接傳輸訊息，非常有趣!

Reference: <https://github.com/google/eddystone/tree/master/eddystone-url>