# Table of Contents

**1. Zephyr Beacon example** ..... 2
***1.1. Tasks to do during the laboratory*** ..... 4

# 1. Zephyr Beacon example

It's time to deal with some wireless transmission using nRF52840 SoC. At the beginning, we will use the example of the Bluetooth Low Energy Beacon which is included in the Zephyr installation in PlatformIO. This example advertises the device as the “Test beacon” with details provided in Eddystone [1)] format.

After importing it to the Visual Studio Code we should make some configuration: remove the unwanted build target from platformio.ini, and if we want to receive messages via serial port enable it within prj.conf. Notice that to use BLE in the example it is enabled in the prj.conf file.

```
CONFIG_BT=y
CONFIG_BT_DEBUG_LOG=y
CONFIG_BT_DEVICE_NAME="Test beacon"
```

This example starts the Bluetooth interface with the function bt_enable() that passes the pointer to the bt_ready() which is the callback function to notify that initialization has finished.

```
err = bt_enable(bt_ready);
```

Inside this bt_ready() function, we call bt_le_adv_start() function to start our device to function as the server (broadcaster) and begin advertising.

```
bt_le_adv_start(BT_LE_ADV_NCONN, ad, ARRAY_SIZE(ad), sd, ARRAY_SIZE(sd));
```

The argument “BT_LE_ADV_NCONN” specifies the advertising mode without the possibility of accepting connections from a client (observer).

The “ad” structure holds the content of the advertising frame data.

```
static const struct bt_data ad[] = {
    BT_DATA_BYTES(BT_DATA_FLAGS, BT_LE_AD_NO_BREDR),
    BT_DATA_BYTES(BT_DATA_UUID16_ALL, 0xaa, 0xfe),
    BT_DATA_BYTES(BT_DATA_SVC_DATA16,
              0xaa, 0xfe, /* Eddystone UUID */
              0x10, /* Eddystone-URL frame type */
              0x00, /* Calibrated Tx power at 0m */
              0x00, /* URL Scheme Prefix http://www. */
              'z', 'e', 'p', 'h', 'y', 'r',
              'p', 'r', 'o', 'j', 'e', 'c', 't',
              0x08) /* .org */
};
```

It contains three elements of specific types. The complete list of types with their identifiers you can find in the official Bluetooth SiG website [2)].

Tab. 1: List of elements of the advertising frame data

| Type ID | Type name (as in BLE specification) | Content |
|---|---|---|
| 0x01 | Flags | specifies if BR/EDR is supported |
| 0x03 | Complete List of 16-bit Service Class UUIDs | Eddystone UUID |
| 0x16 | Service Data - 16-bit UUID | UUID and content of Eddystone data |

The “sd” argument is the structure that holds scan response data. This can be read by the observer with a scan request. It has the type of the name of the device and contains the text “Test beacon”.

Tab. 2: List of elements of the advertising frame scan response

| Type ID | Type name (as in BLE specification) | Content |
|---|---|---|
| 0x09 | Complete Local Name | name of the device |

You can observe the details in the nRF Connect application on your mobile phone.

The beacon example is very simple because the beacon device is intended to advertise only so, there is no further functionality implemented. This example shows the so-called Eddystone format [3] of Bluetooth Beacons that send simple information to nearby devices. To reduce the number of bytes sent in the advertisement frame the Eddystone format defines the list of prefixes and suffixes that can be used in the URL encoded as values of non-printable characters[4]. For example, web prefixes can be encoded as in the table 3.

Tab. 3: Eddystone web prefixes

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

Web suffixes defined in Eddystone format are listed in the table 4.

Tab. 4: Eddystone web suffixes

| Decimal | Hex | Suffix |
|---|---|---|
| 0 | 0x00 | .com/ |
| 1 | 0x01 | .org/ |
| 2 | 0x02 | .edu/ |
| 3 | 0x03 | .net/ |
| 4 | 0x04 | .info/ |
| 5 | 0x05 | .biz/ |
| 6 | 0x06 | .gov/ |
| 7 | 0x07 | .com |
| 8 | 0x08 | .org |
| 9 | 0x09 | .edu |
| 10 | 0x0a | .net |
| 11 | 0x0b | .info |
| 12 | 0x0c | .biz |
| 13 | 0x0d | .gov |

# 1.1. Tasks to do during the laboratory

1. Import the Zephyr BLE Beacon example, compile it, and run.
2. Use the nRF Connect application (For Android mobile) and find the “Test beacon” device.
3. Modify the software obtaining different prefixes and suffixes for the beacon.
4. Modify the name of the device with “Polsl Beacon”, and the advertised web address with "https://www.polsl.pl"

[1] [3]
,
https://en.wikipedia.org/wiki/Eddystone_(Google)
[2]
https://bitbucket.org/bluetooth-SIG/public/src/main/assigned_numbers/core/ad_types.yaml
[4]
https://github.com/google/eddystone/tree/master/eddystone-url

From:
https://dokuwiki.zmitac.aei.polsl.pl/ - **RAU6 Dokuwiki**

Permanent link:
**https://dokuwiki.zmitac.aei.polsl.pl/en:edu:zmitac:s2:aih:lab:nrf52zephyrblebeacon?rev=1702143516**

Last update: **2023/12/09 18:38**