Application note

Bluetooth low energy beacons with Eddystone

Introduction

The beacon application in the X-CUBE-BLE1 software expansion for STM32Cube is an implementation of the Google Eddystone beacon profile, built on the STM32Cube software platform.

The package comes with code examples for the X-NUCLEO-IDB05A2 when connected to a NUCLEO-L053R8, NUCLEO-L476RG or NUCLEO-F401RE development board.

The application features:

- BLE profile running on a Google Eddystone beacon platform
- Support for the UID and URL frame types
- Portability across different STM32 device families thanks to STM32Cube

Note: The beacon application is also available when connecting an STM32 Nucleo development board to the X-NUCLEO-IDB05A1 expansion board.

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
ACI	Application controller interface
ATT	Attribute protocol
BLE	Bluetooth low energy
BSP	Board support package
BT	Bluetooth
GAP	Generic access profile
GATT	Generic attribute profile
HAL	Hardware abstraction layer
HCI	Host controller interface
IDE	Integrated development environment
MCU	Micro controller unit
PCI	Profile command interface
UUID	Universally unique identifier

2 BLE Eddystone beacon overview

The BLE Eddystone beacon uses the following hardware and software components available for download at www.st.com:

- NUCLEO-L053R8: STM32 Nucleo-64 development board with STM32L053R8 MCU, supports Arduino and ST morpho connectivity
- NUCLEO-L476RG: ultra-low-power with FPU ARM Cortex-M4 MCU 80 MHz with 1 Mbyte Flash, LCD, USB OTG
- NUCLEO-F401RE: STM32 dynamic efficiency MCU, ARM Cortex-M4 core with DSP and FPU, up to 512 Kbytes Flash, 84 MHz CPU, Art accelerator
- X-NUCLEO-IDB05A2: BLE expansion based on the BlueNRG-M0 network processor module for STM32 Nucleo
- STM32CubeL0 HAL support package
- STM32CubeL4 HAL support package
- STM32CubeF4 HAL support package
- X-CUBE-BLE1 driver package, BLE software expansion for STM32Cube
- Custom Eddystone compliant profile supporting UID and URL frame types

RELATED LINKS

Visit the STM32Cube ecosystem web page on www.st.com for further information

3 Eddystone beacon demonstration application

The software development kit contains a BlueNRG-MS/BlueNRG-M0 configuration example which advertises specific service data and allows another BLE device to recognize if it is in the range of the BlueNRG-MS/ BlueNRG-M0 beacon device.

3.1 Initialization

To correctly configure a BlueNRG-MS/BlueNRG-M0 device to be used as an Eddystone beacon device, you have to:

- initialize the GATT (general attribute profile) server in the device (`ACI_GATT_INIT`);
- initialize the GAP (general access profile) in the device in peripheral mode (`ACI_GAP_INIT: peripheral`).

3.2 Advertising service data

The BLE Eddystone beacon application advertises the service data listed in the table below.

Table 2. BlueNRG-MS/BlueNRG-M0 Eddystone beacon advertising service data

Mode	Data field	Description	Notes
UID	Tx Power	Calibrated Tx power at 0 m	The best way to determine this value is to measure the beacon actual output at 1 meter and add 41 dBm (signal loss over 1 meter).
	Namespace ID	10-byte ID Namespace	Unique self-assigned beacon namespace.
	Beacon ID	6-byte ID Instance	Unique ID within the namespace.
URL	Tx power	Calibrated Tx power at 0 m	The best way to determine this value is to measure the beacon actual output at 1 meter and add 41 dBm (signal loss over 1 meter).
	URL scheme	Encoded Scheme Prefix	Refer to the Eddystone github for details.
	Encoded URL	Encoded URL (max 17 char).	The URL scheme is defined by RFC-1738. It is recommended to use a URL shortening service if the desired URL is longer than 17 characters.

3.3 Entering non-connectable mode

To set a static MAC address, the device uses the ACI HAL to write the desired MAC address in the `BlueNRG-MS_Init()` function, where `SERVER_BDADDR` is the 6-byte MAC address.

```
aci_hal_write_config_data(CONFIG_DATA_PUBADDR_OFFSET,
 CONFIG_DATA_PUBADDR_LEN, SERVER_BDADDR)
```

The BLE beacon device uses the GAP ACI command to enter non-connectable, undirected mode.

```
aci_gap_set_discoverable(ADV_NONCONN_IND, /*< Advertise as non-connectable, undirected. */
AdvertisingInterval, AdvertisingInterval, /*< Set the advertising interval min and max (0.625
 us increment). */
PUBLIC_ADDR, /*< Use the public address. */
NO_WHITE_LIST_USE, /*< Do not set any connection white list. */
0, NULL, /*< Do not use a local name. */
0, NULL, /*< Do not include the service UUID list. */
0, 0); /*< Do not set a slave connection interval. */
```

To advertise the specific selected service data, the BLE beacon application uses the GAP ACIs in `EddystoneUID_Init()` or `EddystoneURL_Init()` functions.

```
/* Remove TX power level field from the advertising data: it may be necessary to have enough
space for the beacon service data */
ret = aci_gap_delete_ad_type(AD_TYPE_TX_POWER_LEVEL);/*
 Define the beacon service payload for UID data */
uint8_t service_data[] =
{
 23, /*< Length. */
AD_TYPE_SERVICE_DATA, /*< Service Data data type value. */
0xAA, 0xFE, /*< 16-bit Eddystone UUID. */
0x00, /*< UID frame type. */
EddystoneUID_Init->CalibratedTxPower, /*< Ranging data. */
EddystoneUID_Init->NamespaceID[0], /*< 10-byte ID Namespace. */
EddystoneUID_Init->NamespaceID[1],
EddystoneUID_Init->NamespaceID[2],
EddystoneUID_Init->NamespaceID[3],
EddystoneUID_Init->NamespaceID[4],
EddystoneUID_Init->NamespaceID[5],
EddystoneUID_Init->NamespaceID[6],
EddystoneUID_Init->NamespaceID[7],
EddystoneUID_Init->NamespaceID[8],
EddystoneUID_Init->NamespaceID[9],
EddystoneUID_Init->BeaconID[0], /*< 6-byte ID Instance. */
EddystoneUID_Init->BeaconID[1],
EddystoneUID_Init->BeaconID[2],
EddystoneUID_Init->BeaconID[3],
EddystoneUID_Init->BeaconID[4],
EddystoneUID_Init->BeaconID[5],
0x00, /*< Reserved. */
0x00 /*< Reserved. */
};
/* Set the beacon service data on the advertising packet */
ret = aci_gap_update_adv_data(sizeof(service_data), service_data);
/* Define the beacon service uuid list */
uint8_t service_uuid_list[] =
{
 3, /*< Length. */
AD_TYPE_16_BIT_SERV_UUID_CMPLT_LIST, /*< Complete list of 16-bit Service UUIDs data type valu
e. */
0xAA, 0xFE /*< 16-bit Eddystone UUID. */
};
/* Set the beacon service data on the advertising packet */
ret = aci_gap_update_adv_data(sizeof(service_uuid_list), service_uuid_list);
```

3.4 Modifying eddystone_beacon.h

Beacon configuration can be performed easily by modifying the relevant fields within eddystone_beacon.h.

```
#define MAC_ADDRESS 0x12, 0x34, 0x00, 0xE1, 0x80, 0x03
#define EDDYSTONE_UID_BEACON_TYPE (0x01u)
#define EDDYSTONE_URL_BEACON_TYPE (0x02u)
#define ADVERTISING_INTERVAL_IN_MS (10000)
#define CALIBRATED_TX_POWER_AT_0_M ((uint8_t) (-22))
#define NAMESPACE_ID 'w', 'w', 'w', '.','s','t', '.', 'c', 'o', 'm'
#define BEACON_ID 0, 0, 0, 0, 0, 1
#define URL_PREFIX HTTP
#define PHYSICAL_WEB_URL "goo.gl/viVrdi"
```

The `MAC_ADDRESS` field must be modified with the desired MAC address, in MAC-48 format. The ordering is in LSB.

`ADVERTISING_INTERVAL_IN_MS` is a common field for all beacon types and must be specified.

`CALIBRATED_TX_POWER_AT_0_M` can be determined by measuring the transmission power (in dBm) at 1 m and adding 41 dBm, which is the standard loss over 1 m. This field is required for UID and URL beacons.

NAMESPACE_ID and BEACON_ID are specific to the Eddystone UID beacon (refer to Table 2. BlueNRG-MS/ BlueNRG-M0 Eddystone beacon advertising service data for details).

URL_PREFIX specifies the prefix of the desired URL:

- HTTP, if the address begins with “http://”
- HTTPS, if the address begins with “https://
- HTTP_WWW, if the address begins with “http://www.”
- HTTPS_WWW, if the address begins with “https://www.”

PHYSICAL_WEB_URL is the remainder of the URL after the prefix.

Note: There is a 17-character limit to this URL.

4 Limitations and known issues

Currently, multi-beacons are not supported: only a single Eddystone beacon frame type can be exposed at any given point in time.

Eddystone advertising interval must be less than 40959 milliseconds.

5 References

1. Google beacons
2. UM1873: Getting started with the X-CUBE-BLE1 Bluetooth Low Energy software expansion for STM32Cube
3. AN4642: Overview of the BLE Profiles application for X-CUBE-BLE1

Revision history

Table 3. **Document revision history**

Date	Version	Changes
14-Dec-2016	1	Initial release.
28-Apr-2020	2	Added BlueNRG-M0 module and X-NUCLEO-IDB05A2 expansion board compatibility information. Minor text changes.

Contents

List of tables