Program Documentation

ble\_beacon

Schule der ansprechenden Künste

*Andreas Erdmann*

[1 Overview 2](#_Toc25407782)

[1.1 Main Features 2](#_Toc25407783)

[1.2 Hardware 2](#_Toc25407784)

[1.3 Configuration 2](#_Toc25407785)

[1.3.1 Event scheduling 2](#_Toc25407786)

[1.3.2 Beacon data 3](#_Toc25407787)

[1.3.3 Sensor data 3](#_Toc25407788)

[1.4 Important Variables in main.c 4](#_Toc25407789)

[1.5 Program overview 4](#_Toc25407790)

[1.6 Work on advertising\_init() 5](#_Toc25407791)

[1.6.1 Bare, almost nothing 5](#_Toc25407792)

[1.6.2 Adding uuids in advdata 6](#_Toc25407793)

[1.6.3 Adding short UUIDs in advdata 7](#_Toc25407794)

[1.6.4 Advdata packet 7](#_Toc25407795)

# Overview

The ble\_beacon is a firmware for Nordic Semiconductor NRF52832 SOC with support for temperature sensor, humidity sensor and Accelerometer, powered with a CR2032 coin cell.

## Main Features

* Adjustable Bluetooth LE Advertising providing sensor data and beacon battery level
* Sensor data buffer up to 2.500 entries (for time, temperature, humidity data) with configurable interval
* Low power consumption (CR2032 running time > 1 year)
* Configurable Name, Major and Minor device ID, transmit power, broadcast interval
* Location of sensors is stored using mobile App
* Keyfinder functionality (“Where’s my key?”)[[1]](#footnote-1)
* Anti-theft protection (Alert if distance between beacon and mobile App is too big) [[2]](#footnote-2)
* Related iPhone and Android app supporting
  + Comfortable Beacon identification and coupling
  + Beacon device configuration
  + Sensor Data viewer
  + Alert function
  + Location function

## Hardware

The firmware is implemented and tested with the following hardware and components:

|  |  |
| --- | --- |
| Sensor and components |  |
| Nordic Semiconductor NRF52832 SOC |  |
| KIONIX KX022 (± 2g / 4g / 8g Tri-axis Digital Accelerometer) | I2C |
| Sensirion SHT3x (Digital temperature and humidity sensor) | I2C |
| LED | GPIO26 |
| Pushbutton (on board) | GPIO25 |
| Pushbutton (on jig) | GPIO09 |
| 32 MHz crystal |  |
| 32.768 kHz crystal |  |
| Coin cell battery CR2032 holder |  |

TODO:

* Add hardware supplier
* Get and add FCC/CE certificate
* Add schematics

## Configuration

### Event scheduling

|  |  |
| --- | --- |
| Event scheduling |  |
| BLE advertising | Every 1 s [[3]](#footnote-3) |
| SADC power level sample | Every 60 s |
| Sensor sample | Every 15 s |
|  |  |
|  |  |

### Beacon data

#### Advertising data

|  |  |  |
| --- | --- | --- |
| uint8\_t m\_beacon\_info[APP\_BEACON\_INFO\_LENGTH] |  |  |
| Device major value | 2 byte | 0..1 |
| Device minor value | 2 byte | 2..3 |
| Temperature | 2 byte | 4..5 |
| Humidity | 2 byte | 6..7 |
| Accel X | 2 byte | 8..9 |
| Accel Y | 2 byte | 10..11 |
| Accel Z | 2 byte | 12..13 |
| Battery | 2 byte | 14..15 |
| APP\_BEACON\_INFO\_LENGTH | 0x10 | 0..15 |

#### Scan response data

|  |  |  |
| --- | --- | --- |
| uint8\_t m\_beacon\_info[APP\_BEACON\_INFO\_LENGTH] |  |  |
| Device major value | 2 byte | 0..1 |
| Device minor value | 2 byte | 2..3 |
| Temperature | 2 byte | 4..5 |
| Humidity | 2 byte | 6..7 |
| Accel X | 2 byte | 8..9 |
| Accel Y | 2 byte | 10..11 |
| Accel Z | 2 byte | 12..13 |
| Battery | 2 byte | 14..15 |
| APP\_BEACON\_INFO\_LENGTH | 0x10 | 0..15 |

### Sensor data

The sensor values are stored in

#define BUFFER\_SIZE 21 // read buffer from sensors: temp+hum (6=2\*msb,lsb,crc) + xyz (6=3\*lsb,msb) + INT\_REL (5) + INS1 (4)

static uint8\_t m\_buffer[BUFFER\_SIZE];

|  |  |  |
| --- | --- | --- |
| uint8\_t m\_buffer[BUFFER\_SIZE]; |  |  |
| Temperature | 2 byte | 0..1 |
| Temperature CRC (unused) | 1 byte | 2 |
| Humidity | 2 byte | 3..4 |
| Humidity CRC (unused) | 1 byte | 5 |
| Accel X | 2 byte | 6..7 |
| Accel Y | 2 byte | 8..9 |
| Accel Z | 2 byte | 10..11 |
|  |  |  |
|  |  |  |
|  |  |  |

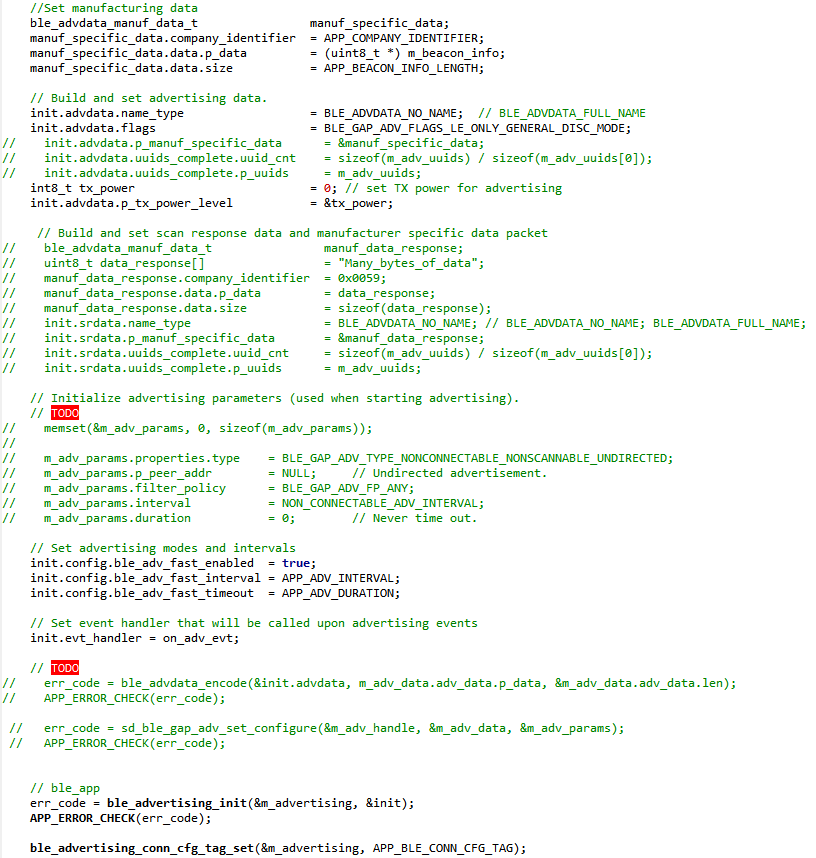
## Important Variables in main.c

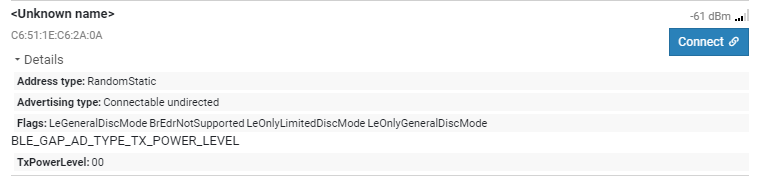
|  |  |
| --- | --- |
| Variables |  |
| uint8\_t m\_buffer[BUFFER\_SIZE] | Holds the current sensor data, will be updated in function read\_all\_sensors() and read/used in process\_all\_data() to update the encoded advertising data. |
| m\_transferbuffer\_counter |  |
| m\_transferbuffer[][] |  |
| ble\_uuid\_t m\_adv\_uuids[] |  |
| NRF\_BLE\_GATT\_DEF(m\_gatt) | GATT module instance |
| NRF\_BLE\_QWR\_DEF(m\_qwr) | Context for Queued Write module |
| BLE\_ADVERTISING\_DEF(m\_advertising) | Advertising module instance |
| uint16\_t m\_conn\_handle | Handle of current connection |
| ble\_os\_t m\_our\_service |  |
| uint8\_t m\_beacon\_info[] |  |

## Program overview

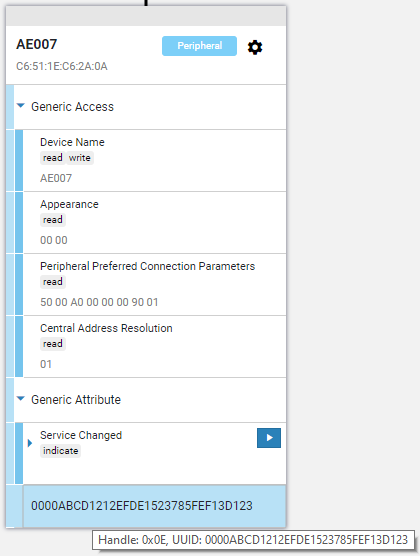
## Work on advertising\_init()

### Bare, almost nothing





Connect:



### Adding uuids in advdata



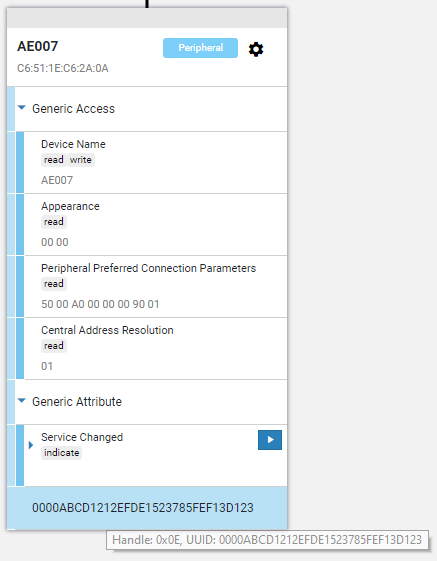
With



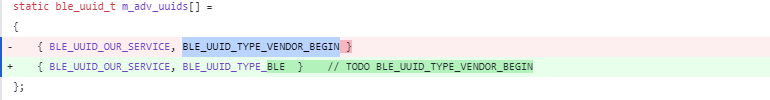




Connect:



### Adding short UUIDs in advdata



Gives



### Advdata packet

The array m\_advertising.enc\_advdata contains:

Flags 0..2 02 01 06

TX Power Level 3..5 02 0A 00

UUIDs 6..9 03 03 CD AB // short UUID „ABCD“

Manuf. Spec. Data 10 13 // length

11 FF // BLE\_GAP\_AD\_TYPE\_MANUFACTURER\_SPECIFIC\_DATA

12..13 59 00 // Company Identifier

14..17 00 07 00 08 // MAJ MIN

18..19 FE FE // Temperature

20..21 FD FD // Humidity

22..23 AA AA // X

24..25 BB BB // Y

26..27 CC CC // Z

28..29 0B B8 // Battery

30 00 // ?





1. Requires iPhone or Android app, GPS of cell phone is used. [↑](#footnote-ref-1)
2. Requires iPhone or Android app. [↑](#footnote-ref-2)
3. Configurable using iPhone/Android app [↑](#footnote-ref-3)