How to set the Bluetooth device address on BlueNRG-MS

By Salvo Bonina

Main components	
BlueNRG-MS	Upgradable Bluetooth® Low Energy network processor
SPBTLE-RF	Very low power module for Bluetooth® Smart v4.1
SPBTLE-RF0	Very low power module for Bluetooth® Smart v4.1

Purpose and benefits

The BlueNRG-MS is a very low power Bluetooth® Low Energy (BLE) single-mode network processor, compliant with Bluetooth Core Specification v4.1. The entire BLE stack runs on the embedded Cortex M0 core. The device interfaces with an external host microcontroller using the SPI transport layer and a set of APIs composed of standard Host Controller Interface (HCI) commands and vendor-specific Application Command Interface (ACI) commands.

The purpose of this document is to describe how to set the Bluetooth Device address on the BlueNRG-MS.

Note that the content of this document applies to the SPBTLE-RF and SPBTLE-RF0 modules.

Description

All Bluetooth devices must have a Bluetooth Device Address that uniquely identifies the device to another Bluetooth device.

The following device addresses are supported from the BlueNRG-MS device:

- Public address
- Random address: the random device address may be of either of the following two sub-types:
  - Static address
  - Private address

Device addresses may be either public or random. Public device and random device addresses are both 48 bits in length and are denoted as colon-delimited hex values e.g. AA:BB:CC:DD:EE:FF.

The public device address shall be created in accordance with the IEEE 802-2001 standard, using a valid Organizationally Unique Identifier (OUI) obtained from the IEEE Registration Authority. Public device addresses are known as MAC addresses. For additional information about MAC addresses, the reader can refer to publicly available sources such as https://en.wikipedia.org/wiki/MAC_address.

For the details on how a BLE device can generate a random address of either subtype, the reader can refer to the Bluetooth Core Specification v4.1.

BlueNRG-MS doesn't have a valid pre-assigned Public address. If the application would like to use a public address, this address needs to be obtained by the competent authority and then stored in a persistent memory location of the final product (either within the host microcontroller Flash or into an external storage area). During the BlueNRG-MS initialization phase, the application must configure this address into the BlueNRG-MS radio.

The ACI command to set the public address is *tBleStatus aci_hal_write_config_data(uint8_t offset, uint8_t len, const uint8_t *val).* The parameters must be set as follow:

- *Offset: 0x00*

- *Length: 0x06*

- *Value: pointer to the public address value, e.g. 0xaabbccddeeff (6-byte array)*

The host microprocessor should send the command *aci_hal_write_config_data* to the BlueNRG-MS device before starting BLE operations and every time after a power-up or reset, since the command *aci_hal_write_config_data* does not persistently save the data in the BlueNRG-MS Flash.

The following pseudo-code example illustrates how to set a MAC address from the application:

uint8_t bdaddr[] = {0xFF, 0xEE, 0xDD, 0xCC, 0xBB, 0xAA};

ret=aci_hal_write_config_data(0x00, 0x06, bdaddr);

if(ret) { PRINTF("Setting address failed.\n")}

Further, to have the device advertising with the public address, the function

tBleStatus aci_gap_set_discoverable()

must be called with the parameter "*OwnAddType*" set to 0x00 : Public Device Address.

BLE devices can also use random addresses. From the BlueNRG-MS radio stack version 7.1c and onwards, the random address is generated autonomously by the BlueNRG-MS radio stack upon the first call to the API *aci_gap_init()*. This address is stored persistently into the BlueNRG-MS Flash. The address value can be read from the application using the *tBleStatus aci_hal_read_config_data(uint8_t offset, uint16_t data_len, uint8_t *data_len_out_p, uint8_t *data);* command with the parameter *offset* set equal to 0x80.

Alternatively, the application can set from the external host processor a random address using the *int hci_le_set_random_address(tBDAddr bdaddr);* command after each reset. If the random address is not set through the *hci_le_set_random_address* command, then the address generation is handled autonomously by the stack as described above.

Support material

Related design support material
STSW-BLUENRG-DK - Setup for BlueNRG Kits
STEVAL-IDB005V1 – Bluetooth low energy board based on the BlueNRG-MS network processor
STEVAL-IDB006V1 – BlueNRG-MS Evaluation board
STEVAL-IDB006V1M - Bluetooth® Low Energy USB dongle based on the SPBTLE-RF module
Documentation
DS10691: Upgradable Bluetooth® Low Energy network processor
DS11045: SPBTLE-RF Very low power module for Bluetooth® Smart v4.1
UM1865: BlueNRG-MS Bluetooth® LE stack application command interface (ACI)
PM0237: BlueNRG, BlueNRG-MS stacks programming guidelines

Revision history

Date	Version	Changes
22-Aug-2016	1	Initial release
17-Jan-2017	2	Updated to comply with new API functions
19-Dec-2018	3	Added detailed steps to set up a public address on a BlueNRG-MS device. Added reference to the SPBTLE-RF0 module

IMPORTANT NOTICE – PLEASE READ CAREFULLY