



# RN4020 Firmware 1.33BEC Release Note

## 1 Overview

Firmware version 1.33BEC is released for Microchip's RN4020 Bluetooth Low Energy Module. This release adds new the following features:

- Custom Advertising broadcast packets
- GATT Service Definition Enhancement for both public and private services
- Private (random) MAC address for Advertisements

### 1.1 Resolved Issues

Release 1.33BEC contains the fixes from firmware release 1.23.

### 1.2 Known Issues

- Compatibility with previous 1.2x or earlier is not guaranteed, especially if the application relies on SS (set local services) command to create public services. **In firmware 1.33BEC, the SS command only supports battery service and device information.** Firmware 1.33BEC supports creating public services (16-bit UUID) using PS and PC command, in the same manner as defining private services.
- MLDPv2 was introduced in firmware 1.20. MLDPv2 uses unacknowledged GATT writes and notifications to improve data transfer rate across the Bluetooth connection. Therefore, any data transfer over MLDPv2 may have lost bytes if the data is not received by other end. It is highly recommended to implement packet retransmission mechanism for critical data transfers, or use MLDPv1.
- When using server write commands (SHW, SUW) over UART to update GATT characteristics, providing a data payload larger than the defined length may corrupt data. To avoid data corruption, the MCU issuing the write server commands **MUST** ensure the payload does not exceed the defined GATT characteristic size.

### 1.3 New Features

- **GATT Service Enhancements**  
In firmware 1.33BEC, the SS command has been modified to support two public services: battery service and device information. The PS and PC commands are used to support 16-bit UUIDs, allowing more flexibility to define any public or private service.
- **User Defined Advertising Packets**  
The SS command has been modified to support only battery service and device information. The PS and PC commands now support 16-bit UUIDs, allowing more flexibility to define any public or private service.
- **Bluetooth Address Functions**  
Several new commands are added to manage the Bluetooth address of the device. The address can be set to specified MAC or random (private) resolvable address.

## 1.4 Ordering Information

- At the time of this publication, firmware 1.33BEC is available as a DFU firmware update for existing RN4020 customers. Customers who need to order RN4020 modules with the 1.33BEC firmware should contact their Microchip representative.

## 1.5 Firmware Upgrade

RN4020 devices, such as modules and PICTails, can be updated to firmware 1.33BEC using the **DFU Update Utility** which can be downloaded from the [www.microchip.com/RN4020](http://www.microchip.com/RN4020) product page. To perform a firmware update on an RN4020 module or PICTail, execute the utility on a Windows PC with a COM port connected to the RN4020 UART interface. Hardware flow control (CTS/RTS) is required to perform a DFU update. Follow the instructions in the DFU Update Utility.

## 2 New Commands in 1.33BEC

### 2.1 Connectable Advertising Payload

#### NA, <HEX>

This command is used to customize connectable advertisement payload.

By default, a connectable advertisement started by command “A” or auto-advertisement feature has following Advertisement Data (AD) structure format:

Flags   Service UUID List 16-bit   Service UUID List 128-bit   Local Name Complete
--

Due to the limited size of advertisement payload, the following conditions apply:

- When private service or MLDP is enabled, AD structure Service UUID List 16-bit is omitted
- When private service and MLDP are both enabled, only the private service UUID is in the Service UUID List 128-bit.
- In the AD structure Service UUID List 16-bit, common services such as Device Info and Battery services are not listed
- AD structure Local Name Complete is added to advertisement payload only if there is enough space. Therefore, if AD structure Service UUID List 128-bit is included, device name less or equal to 6 bytes could be included in advertisement payload.

Command “NA” provides the capability to modify the default advertisement payload by adding an AD structure one at a time. The input parameter of command “NA” is the raw AD structure format without AD length. See table 1 for AD types. For details specifications of AD structure, please refer to Core Spec Vol3, Part C, Section 11 and Core Spec Supplement v4, Part A.

Command “NA” appends additional AD to the advertisement payload, not replacing previous one. To clear the advertisement payload, use command “NZ”.

The flags field is added by RN4020 module automatically.

For example, to add local device name and TX power level, use following commands:

```
Y                // Stop advertising
NZ              // clear AD payload set by previous NA & NB
NA,0944617665   // AD Type 0x09 Local Name Complete
                // local name complete, and 44617665 are ASCII
                // code “Dave”, the name of device

NA,0A04         // add AD Tx power level, where 0x0A is AD type
                // Tx power level and 04 indicates output power is
                // 4dBm.
A              // start connectable advertisement with custom payload
```

## 2.2 Non-Connectable Advertisement Payload

**NB,<HEX>**

This command is used to customize non-connectable advertisement payload. The use of command “NB” is similar to that of command “NA”. The difference is that command “NB” customizes non-connectable advertisement, while command “NA” customizes connectable advertisement. “NB” can be used to define customer beacons.

The following example defines several types of advertising packets:

```
// Custom ADV packet, MSD ad type (xFF), Microchip vendor ID (0x00CD)
// two byte payload (0xDA1E)
NB,FFCD00DA1E

// AltBeacon Packet (https://github.com/AltBeacon/spec)
// AD Type: (xFF) MSD, see table 1 for AD types
// MFG ID: (x00CD) Microchip MFG ID assigned by BT SIG is used in this
//          in this example. It must be encoded little endian in payload.
// Beacon Code: (xBEAC) fixed value
// Beacon ID: 16 byte UUID: 5b60c10e62f311e59d70feff819cdc9f
//          2 byte user defined: 0xda1e used in this example
// RefRSSI: Signed byte 0 to -127: 0xc9, (-55 dB) used in this example
// Mfg Rsvd: 1 byte reserved for user defined, 0x88 used in this example
NB,FFCD00BEAC560C10E62F311E59D70FEFF819CDC9FDA1EC988
```

## 2.3 Clear Advertisement Payload

**NZ**

This command is used to clear custom advertisement payload **set by previous NA and NB commands**. The default connectable advertisement is enabled after this command is sent.

## 2.4 Assign Address to Local Device

**&,<MAC>**

Command “&” assigns a specific address to local device. It accepts one input parameter, which is 6-byte random address. This random address could be a static or private address. For format of random address, please refer to Bluetooth Core Specification version 4.1, Vol 3, Part C, Section 10.8. If device is currently advertising, the advertising address will be immediately changed to the assigned random address. This command can be invoked directly or automated via a script. **The address is not saved in NVM.**

Example

```
&,6F1234567890          // Set random address to be 6F1234567890
```

## 2.5 Enable Resolvable Random Addresses

**&R**

Command “&R” generates a resolvable random address and assign it as the current random address. If device is currently advertising, the advertising address will be immediately changed to the new resolvable random address. This command can be invoked directly or automated via a script

Example

```
&R // Automatically generate and assign a resolvable
// random address
```

## 2.6 Disable Random Addresses

**&C**

Command “&C” clears the random address and uses local MAC address for advertisement. If device is currently advertising, the advertising address will be immediately changed to the local MAC address.

Example

```
&C // Clears random address and use MAC address
```

## 3 Appendix A: AD Types

Table 1: Bluetooth AD Types for NA and NB commands

AD Type HEX)	Description
01	Flags
02	Incomplete list of 16-bit UUIDs
03	Complete list of 16-bit UUIDs
04	Incomplete list of 32-bit UUIDs
05	Complete list of 32-bit UUIDs
06	Incomplete list of 128-bit UUIDs
07	Complete list of 128-bit UUIDs
08	Shortened local name
09	Complete local name
0A	TX power level
0D	Class of device
0E	Simple pairing hash
0F	Simple pairing randomizer
10	TK value
11	Security OOB flag
12	Slave connection interval range
14	List of 16-bit service UUIDs
15	List of 128-bit service UUIDs
16	Service data
FF	Manufacture Specific Data