*n*Blue™ *Bluetooth*® 4.0
*n*Blue™ Beacon Configuration Tool
User's Guide

AT HOME. AT WORK. ON THE ROAD. USING BLUETOOTH WIRELESS TECHNOLOGY MEANS TOTAL FREEDOM FROM THE CONSTRAINTS AND CLUTTER OF WIRES IN YOUR LIFE.

Table of Contents

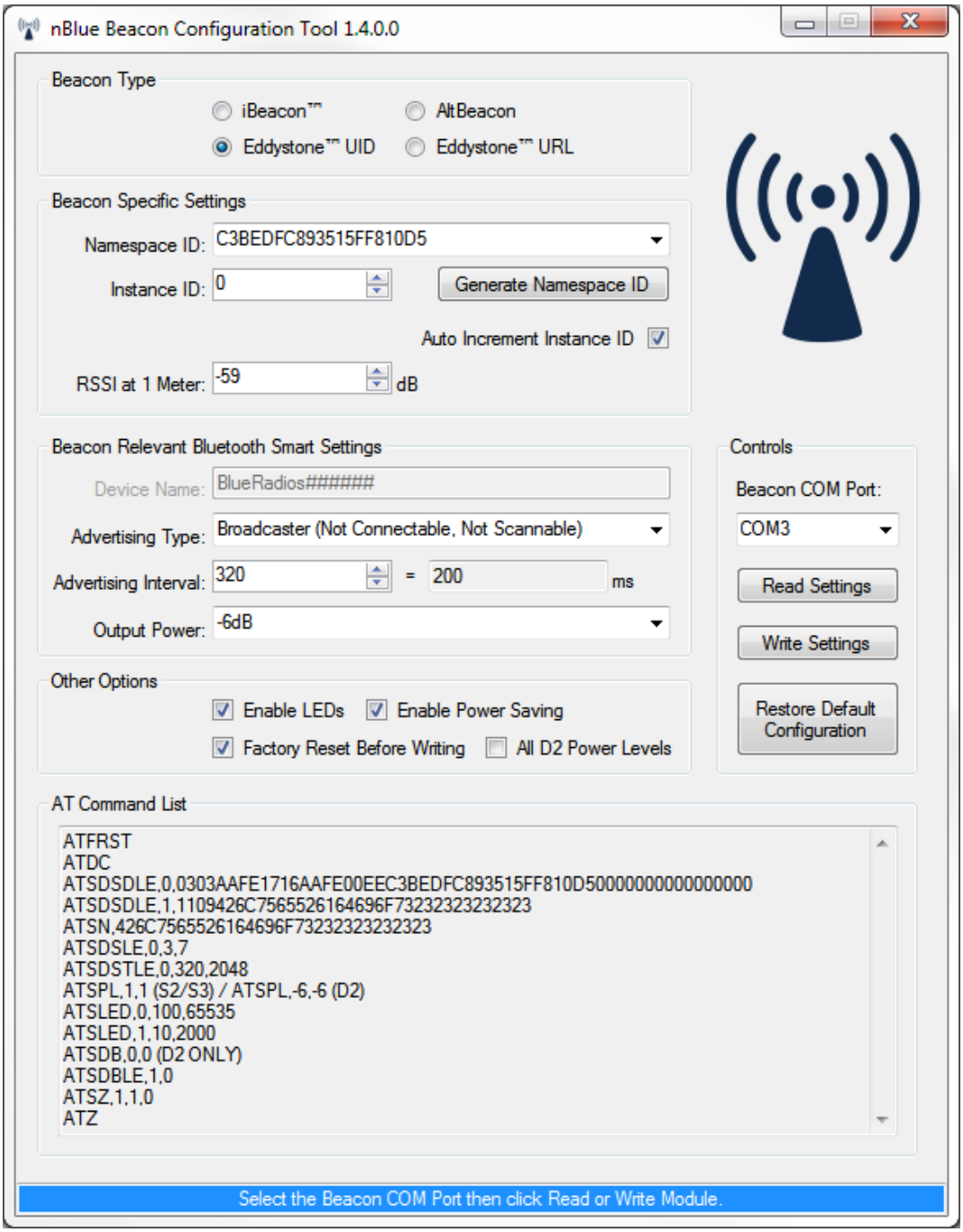
Revision History

Rev #	Date	Description
1.0.0.0	2/7/2014	**Initial Release**
1.1.0.0	2/14/2014	- Added Restore Default Configuration Button - Updated ToolTips - Minor bugfixes and UI changes
1.2.0.0	12/31/2014	- Added support for AltBeacon protocol - Set default output power to -6dBm to better match default -59dB measured power / reference RSSI - Minor bug fixes and UI changes - Renamed application nBlue Beacon Configuration Tool - Renamed executable nBBCT.exe
1.3.0.0	3/2/2015	- Added support for D2 modules - Minor bug fixes and UI changes
1.4.0.0	8/6/2015	- Added support for Eddystone™, an open beacon format from Google that supports Android and iOS devices - Minor bug fixes and UI changes

1 Introduction

***n*Blue™ Beacon Configuration Tool** (nBBCT) is a Windows application that allows ***Blue*Radios® *n*Blue™** *Bluetooth* 4.0 modules to be configured as a beacon. It requires the .NET Framework 4, which can be downloaded from: http://www.microsoft.com/en-us/download/details.aspx?id=17851.

Since beacons can be used for many different applications, nBBCT is only used for configuring beacons. A separate application is required to use the beacon for its intended purpose.



2 Background

***Blue*Radios® *n*Blue™** *Bluetooth* 4.0 modules can be configured using iBeacon™ location and proximity detection technology as well as either of the open AltBeacon or Eddystone™ beacon standards. iBeacon is a proprietary specification developed by Apple Inc. intended for use only with Apple products, whereas AltBeacon and Eddystone are open specifications which can be used with any remote device. All three specifications provide the same basic functionality, with the exception of Eddystone which extends the functionality to also allow a beacon to advertise a URL.

2.1 iBeacon

iBeacon is a positioning system that Apple Inc. calls "a new class of low-powered, low-cost transmitters that can notify nearby iOS devices of their presence." The technology enables an iOS device or other hardware to send push notifications to iOS devices in close proximity.

In a real life scenario it would be more of a location-aware, context-aware, pervasive small wireless sensor beacon that could pinpoint your location in a store: iBeacons could send you notifications of items around you that are on sale or items you may be looking for, and it could enable payments at the point of sale (POS) where you don't need to remove your wallet or card to make a payment.

It works by using *Bluetooth* low energy proximity sensing to transmit a universally unique identifier picked up by a compatible app or operating system that can be turned into a physical location or trigger an action on the device such as a Check-in on social media.

2.2 AltBeacon

From http://altbeacon.org/:

AltBeacon is a protocol specification that defines a message format for proximity beacon advertisements. AltBeacon proximity beacon advertisements are transmitted by devices for the purpose of signaling their proximity to nearby receivers. The contents of the emitted message contain information that the receiving device can use to identify the beacon and to compute its relative distance to the beacon. The receiving device may use this information as a contextual trigger to execute procedures and implement behaviors that are relevant to being in proximity to the transmitting beacon.

Example use cases for proximity beacons include but are not limited to:

- Notifying users of special offers as they visit areas within a department store
- Presenting opportunities to explore additional information about an exhibit to a museum visitor
- Automatically checking in with a restaurant's reservation system as the customer arrives

Use of the AltBeacon format will require updating the module to firmware version 3.5.3.0 or newer due to an issue in previous versions. The module can be updated using *n*Blue Programmer.

- *nBlue is a trademark of BlueRadios, Inc.*
- *iBeacon is a trademark of Apple Inc.*
- *Eddystone is a trademark of Google Inc.*

2.3 Eddystone™

From https://github.com/google/eddystone:

Eddystone is a protocol specification that defines a Bluetooth low energy (BLE) message format for proximity beacon messages. It describes several different frame types that may be used individually or in combinations to create beacons that can be used for a variety of applications.

The design of Eddystone has been driven by several key goals:

- Works well with Android and iOS Bluetooth developer APIs
- Straightforward implementation on a wide range of existing BLE devices
- Flexible architecture permitting development of new frame types
- Fully compliant with the Bluetooth Core Specification

Eddystone defines 3 different advertising frame types: Eddystone-UID, Eddystone-URL and Eddystone-TLM. Only Eddystone-UID and Eddystone-URL are supported by nBBCT, Eddystone-TLM is not supported.

2.3.1 Eddystone-UID

The Eddystone-UID frame broadcasts an opaque, unique 16-byte Beacon ID composed of a 10-byte namespace ID and a 6-byte instance ID. The Beacon ID may be useful in mapping a device to a record in external storage. The namespace ID may be used to group a particular set of beacons, while the instance ID identifies individual devices in the group. The division of namespace and instance IDs may also be used to optimize BLE scanning strategies, e.g. by filtering only on the namespace.

2.3.2 Eddystone-URL

The Eddystone-URL frame broadcasts a URL using a compressed encoding format in order to fit more within the limited advertisement packet.Once decoded, the URL can be used by any client with access to the internet. For example, if an Eddystone-URL beacon were to broadcast the URL https://goo.gl/Aq18zF, then any client that received this packet could choose to visit that url. The Eddystone-URL frame forms the backbone of the Physical Web, an effort to enable frictionless discovery of web content relating to one's surroundings.

3 Advertising Data Format

3.1 iBeacon

iBeacon advertising packets consist of the following fields:

3.1.1 Proximity UUID

A 16 byte UUID (Universally Unique Identifier) used to identify a set of beacons in a certain location or region.

3.1.2 Major ID

A 2 byte value 0-65535 used to identify a group of related beacons.

3.1.3 Minor ID

A 2 byte value 0-65535 used to identify a specific beacon within its Major group.

3.1.4 Measured Power

A 1 byte signed value containing the average RSSI measured at 1m from the beacon. This is used by iOS to estimate the distance to the beacon, so it should be measured and set for each beacon in its intended installation location.

3.2 AltBeacon

AltBeacon advertising packets consist of the following fields:

3.2.1 Beacon ID

A 20-byte value uniquely identifying the beacon, broken up into the following 3 fields:

3.2.1.1 UUID (ID1)

A 16-byte UUID (Universally Unique Identifier) used to identify the advertiser's organizational unit.

3.2.1.2 ID2

A 2-byte value 0-65535 that may be assigned as needed for the use case.

3.2.1.3 ID3

A 2-byte value 0-65535 that may be assigned as needed for the use case.

3.2.2 Reference RSSI

A 1 byte signed value containing the average RSSI measured at 1m from the beacon. This is used to estimate the distance to the beacon, so it should be measured and set for each beacon in its intended installation location.

Use of the AltBeacon format will require updating the module to firmware version 3.5.3.0 or newer due to an issue in previous versions. The module can be updated using *n*Blue Programmer.

3.3 Eddystone-UID

Eddystone-UID advertising packets consist of the following fields:

3.3.1 Beacon ID

A 16 byte value uniquely identifying the beacon, broken up into 2 fields: the namespace ID and the instance ID. The namespace ID may be used to group a particular set of beacons, while the instance ID identifies individual devices in the group.

3.3.1.1 Namespace ID

A 10 byte value used to group a particular set of beacons.

3.3.1.2 Instance ID

A 6 byte value used to identify an individual beacon within a namespace.

3.3.1.3 TX Power

A 1 byte signed value containing the average RSSI measured at **0m** from the beacon (different from the 1m distance used by iBeacon and AltBeacon). This is used to estimate the distance to the beacon, so it should be measured and set for each beacon in its intended installation location.

Measure the RSSI at 1m and enter this into the RSSI at 1 Meter box and nBBCT will add 41 to this value to convert it to RSSI at 0m (as recommended by the Eddystone specification). 41dBm is the signal loss that occurs over 1 meter.

3.3.2 Eddystone-URL

Eddystone-URL advertising packets consist of the following fields:

3.3.2.1 URL Scheme Prefix

The following URL Scheme Prefixes can be used and will be compressed down to 1 byte:
"http://www.", "https://www.", "http://", "https://"

3.3.2.2 URL

The URL string to advertise. The URL can be up to 17 bytes in length not counting the prefix. The following domain strings will be compressed down to 1 byte to save space so they will only count as 1 byte for the length:
".com", ".com/", ".org", ".org/", ".edu", ".edu/", ".net", ".net/", ".info", ".info/", ".biz", ".biz/", ".gov", ".gov/"

3.3.2.3 TX Power

A 1 byte signed value containing the average RSSI measured at **0m** from the beacon (different from the 1m distance used by iBeacon and AltBeacon). This is used to estimate the distance to the beacon, so it should be measured and set for each beacon in its intended installation location.

Measure the RSSI at 1m and enter this into the RSSI at 1 Meter box and nBBCT will add 41 to this value to convert it to RSSI at 0m (as recommended by the Eddystone specification). 41dBm is the signal loss that occurs over 1 meter.

3.4 iBeacon House Example

In this example we will define a set of beacons for a house.

1. First a Proximity UUID would be generated to group all of the beacons in the house, for example:

   F5DA8F6E-CDEA-4B93-84F6-992103233A65

2. Then Major ID's could be assigned to group the iBeacons on each level of the house:

   0=Basement, 1=Main Floor, 2=Second Floor.

3. Next Minor ID's could be assigned to individual beacons for each room in the house:

   Basement: 0=Media Room, 1=Utility Room.
   Main Floor: 0=Garage,1=Kitchen,2=Family Room.
   Second Floor: 0 = Master Bedroom, 1=Guest Bedroom, 2=Office.

A beacon for the Family Room would then have a Proximity UUID of F5DA8F6E-CDEA-4B93-84F6-992103233A65, Major ID = 1, and Minor ID = 2. If the average RSSI of the beacon at 1m was measured to be -55dB, the Measured Power would be set to -55.

A beacon for the Master Bedroom would then have a Proximity UUID of F5DA8F6E-CDEA-4B93-84F6-992103233A65, Major ID = 2, and Minor ID = 0. If the average RSSI of the beacon at 1m was measured to be -57dB, the Measured Power would be set to -57.

4 Using nBlue Beacon Configuration Tool

4.1 Usage Overview

Using nBlue Beacon Configuration Tool is simple:

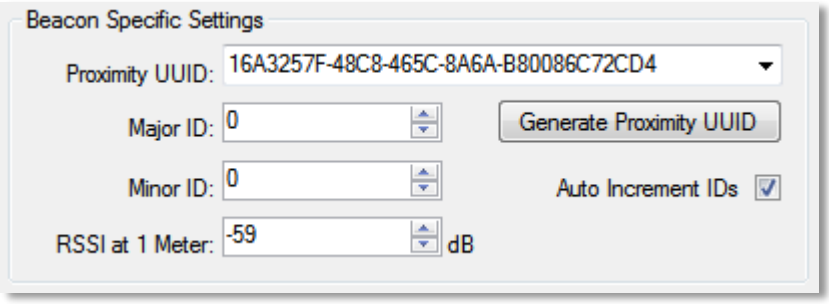
1. Open nBBCT.exe.
2. Connect your module (USB Dongle) to your PC and the module's COM port will automatically be selected.
3. Configure your settings.
4. Click Write settings to configure your beacon.

Your settings will be retained by nBBCT upon closing the app.

4.2 Usage Details

4.2.1 Beacon Specific Settings

4.2.1.1 iBeacon Settings



4.2.1.1.1 Proximity UUID

The Proximity UUID can be entered manually in the standard xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxx format or a new UUID can be generated by clicking the Generate UUID button. Once a UUID has been used (by clicking Write Settings) it will be stored in the Proximity UUID drop down menu for future access.

4.2.1.1.2 Major ID

The Major ID can be a value between 0 and 65535.

4.2.1.1.3 Minor ID

The Minor ID can be a value between 0 and 65535.

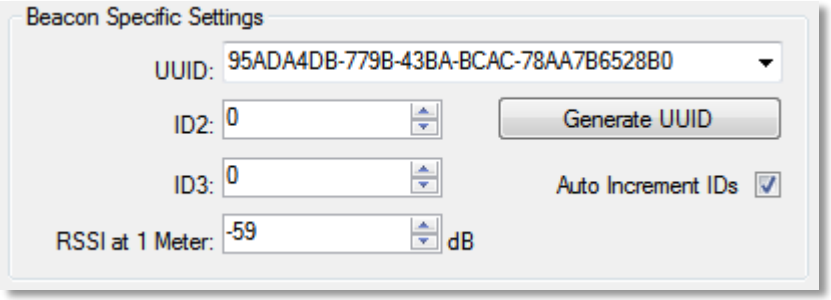
4.2.1.1.4 RSSI at 1 Meter:

The RSSI at 1 Meter can be set from -100 to 20dB.

4.2.1.1.5 Auto Increment IDs

If this box is checked the Minor ID will automatically increment each time a beacon is configured. The Major ID will automatically increment if the Minor ID rolls over from 65535 to 0.

4.2.1.2 AltBeacon Settings



Use of the AltBeacon format will require updating the module to firmware version 3.5.3.0 or newer due to an issue in previous versions. The module can be updated using *n*Blue Programmer.

4.2.1.2.1 UUID (ID1)

The UUID can be entered manually in the standard xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxx format or a new UUID can be generated by clicking the Generate UUID button. Once a UUID has been used (by clicking Write Settings) it will be stored in the UUID drop down menu for future access.

4.2.1.2.2 ID2

ID2 can be a value between 0 and 65535.

4.2.1.2.3 ID3

ID3 can be a value between 0 and 65535.

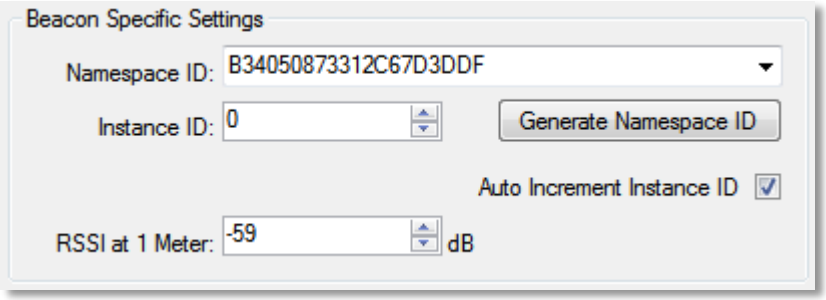
4.2.1.2.4 RSSI at 1 Meter:

The RSSI at 1 Meter can be set from -100 to 20dB.

4.2.1.2.5 Auto Increment IDs

If this box is checked ID2 will automatically increment each time a beacon is configured. ID3 will automatically increment if ID2 rolls over from 65535 to 0.

4.2.1.3 Eddystone-UID Settings



4.2.1.3.1 Namespace ID

The Namespace ID can be entered manually or a new Namespace ID can be generated by clicking the Generate Namespace ID button. Once a Namespace ID has been used (by clicking Write Settings) it will be stored in the Namespace ID drop down menu for future access.

If manually entering a Namespace ID, it should be created using one of the following methods:

Truncated Hash of FQDN
Produce a SHA-1 hash of a fully-qualified domain name that you own. If you desire additional obscurity and/or additional namespaces, you may wish to use a random subdomain under this FQDN. Select the first 10 bytes from that hash.

Elided Version 4 UUID
Generate a version 4 UUID then remove bytes 5 - 10 (inclusive). For example: from this UUID: 8b0ca750-*e7a7-4e14-bd99*-095477cb3e77 remove these bytes: e7a7-4e14-bd99. This produces the following namespace: 8b0ca750095477cb3e77. **This is the method used by the Generate Namespace ID button.**

4.2.1.3.2 Instance ID

The Instance ID can be a value between 0 and 281474976710655 (0xFFFFFFFFFFFF).

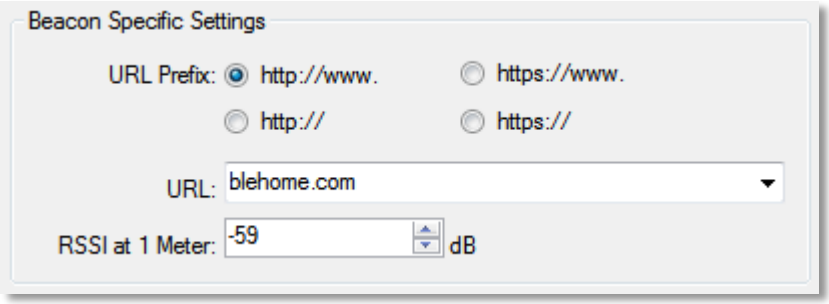
4.2.1.3.3 RSSI at 1 Meter:

The RSSI at 1 Meter can be set from -100 to 20dB. **41 will be added to this value to convert it to RSSI at 0m (as recommended by the Eddystone specification). 41dBm is the signal loss that occurs over 1 meter.**

4.2.1.3.4 Auto Increment Instance ID

If this box is checked the Instance ID will automatically increment each time a beacon is configured.

4.2.1.4 Eddystone-URL Settings



4.2.1.4.1 URL Prefix

Select one of the 4 available URL prefixes.

4.2.1.4.2 URL

Enter the URL string to be advertised (without a prefix). The URL can be up to 17 characters in length not counting the prefix. The following domain strings will be compressed down to 1 byte to save space so they will only count as 1 character towards the length:
".com", ".com/", ".org", ".org/", ".edu", ".edu/", ".net", ".net/", ".info", ".info/", ".biz", ".biz/", ".gov", ".gov/"

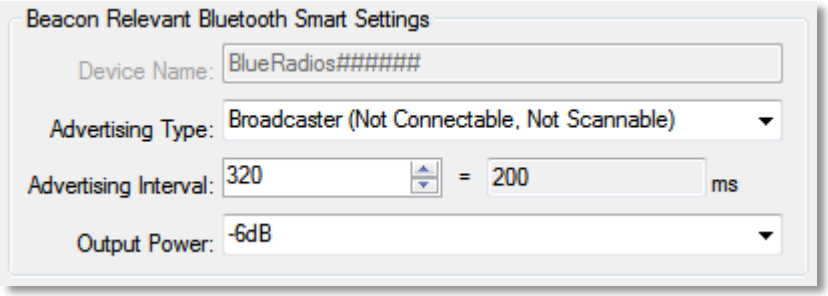
For example "blehome.com" seen above will only take up 8 bytes.

A URL shortening service such as Google URL Shortener can help compress long URLs.

4.2.1.4.3 RSSI at 1 Meter:

The RSSI at 1 Meter can be set from -100 to 20dB. **41 will be added to this value to convert it to RSSI at 0m (as recommended by the Eddystone specification). 41dBm is the signal loss that occurs over 1 meter.**

4.2.2 Beacon Relevant Bluetooth Smart Settings



4.2.2.1 Device Name

The *Bluetooth* Device Name can be up to 20 characters long. '#' characters will be auto populated with the lower part of the *Bluetooth* Device Address to easily create a unique name for each device.

The name is irrelevant if using the Broadcaster Advertising Type since it can't be advertised. Change the Advertising Type to Scannable Broadcaster or Peripheral to enable the Device Name textbox.

4.2.2.2 Advertising Type

The Advertising Type determines how the beacon will advertise and how other devices can interact with it. Beacons are intended to be passive broadcasting devices, so connectivity and scannability are not necessary, but the options are provided if needed.

- **Broadcaster** (Not Connectable, Not Scannable) – This is the default and recommended mode for a beacon. In this mode the device is not connectable or scannable and will be the most power efficient.
- **Scannable Broadcaster** (Not Connectable, Scannable) **–** In this mode the beacon is still not connectable, but will be scannable – allowing other devices to read its Device Name.
- **Peripheral** (Connectable, Scannable) **–** In this mode the beacon is connectable and scannable. *Note: This mode is not recommended for beacon deployments.*

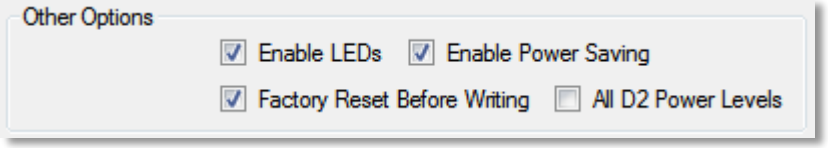
4.2.2.3 Advertising Interval

The Advertising Interval is in units of .625ms and can be set from 32 to 16384 which equals 20 to 10240ms. A lower advertising interval will result in the beacon being discovered faster by iOS devices and faster location updates, but will result in higher power consumption.

4.2.2.4 Output Power

The output power can be adjusted based on the application. A higher output power will provide greater range, but will result in higher power consumption. If the output power is changed the Measured Power should be recalibrated.

4.2.3 Other Options



4.2.3.1 Enable LEDs

If checked PIO_5 (Green LED) will flash once every 2 seconds when the module is advertising and PIO_2 (Blue LED) will turn on when the module has been connected too.

4.2.3.2 Enable Power Saving

If checked sleep mode will be enabled in the module to conserve power in battery powered applications.

4.2.3.3 Factory Reset Before Writing

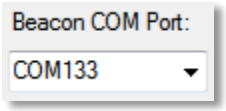
Check this box to automatically factory reset the module prior to writing. In case the module has been previously modified using AT commands, this will guarantee it will be properly configured as a beacon.

4.2.3.4 All D2 Power Levels

Check this box to enable the full range of output power settings for D2 modules. This will change the Output Power combo box to a numeric box that will allow selecting -23 to 10 dB in 1 dB increments. Only check if programming a D2 module.

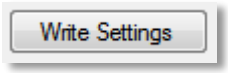
4.2.4 Controls

4.2.4.1 Beacon COM Port



The COM Port that the module to be configured is attached to. If a new COM Port is detected it will automatically be selected, so connect your device after launching nBBCT and it will automatically be selected.

4.2.4.2 Writing Settings

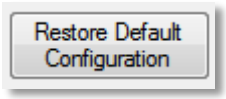


Click this button to write the settings to the module attached to the Beacon COM port.

4.2.4.3 Reading Settings

Click the Read Settings button to read back the settings from a module previously configured as a beacon.

4.2.4.4 Restoring Default Application Configuration



Click the Restore Default Configuration to reset all of the application settings to their default values. This will not write the defaults to your device unless you click Write Settings afterward.

4.2.5 AT Command List

The AT command list displays the set of AT commands that will be sent to the module when Write Settings is clicked.

5 Common Errors

5.1 Failed To Open COM

- Make sure the selected COM port isn't opened in another application.

5.2 Communication Timeout

- Make sure the module is connected to the selected COM port.

5.3 Read Failed

- The module has not been configured as a beacon yet, or the settings have been externally modified, so the settings cannot be read.