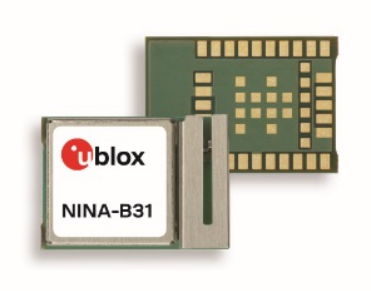
S

# NINA-B31 series

## Stand-alone Bluetooth 5 low energy modules

Getting Started




**Abstract**

This document describes how to set up and use the NINA-B31 series Bluetooth 5 low energy module with the u-blox connectivity software. It also provides a technical overview of the NINA-B31 series.

**www.u-blox.com**

UBX-18022394 - R02

# Document Information

| | |
|---|---|
| **Title** | **NINA-B31 series** |
| **Subtitle** | Stand-alone Bluetooth 5 low energy modules |
| **Document type** | Getting Started |
| **Document number** | UBX-18022394 |
| **Revision and date** | R02 13-Sep-2018 |
| **Disclosure Restriction** | |

This document applies to the following products:

| Product name | Type number | Firmware version | PCN reference |
|---|---|---|---|
| NINA-B311 | NINA-B311-00B-00 | 1.0.0 | N/A |
| NINA-B312 | NINA-B312-00B-00 | 1.0.0 | N/A |

# Contents

# 1 Introduction

The NINA-B31 series are small stand-alone Bluetooth 5 low energy modules featuring full Bluetooth 5 support, a powerful Arm® Cortex®-M4 with FPU, and state-of-the-art power performance. The embedded low power crystal improves power consumption by enabling optimal power save modes.

The NINA-B3 series has two sub-series, NINA-B30 series and NINA-B31 series. This document describes how to set up and use the NINA-B31 series modules with u-blox connectivity software.

## 1.1 Key features

One of the key features of NINA-B31 is the Wireless serial cable replacement functionality. The basic functionality is to transfer data between the serial port and a wireless link. It is possible to configure the NINA-B31 module to automatically setup a connection and/or accept an incoming connection using AT commands.

For a host, this means that an existing serial cable can be replaced by a wireless solution; in many cases without the need to modify the host.

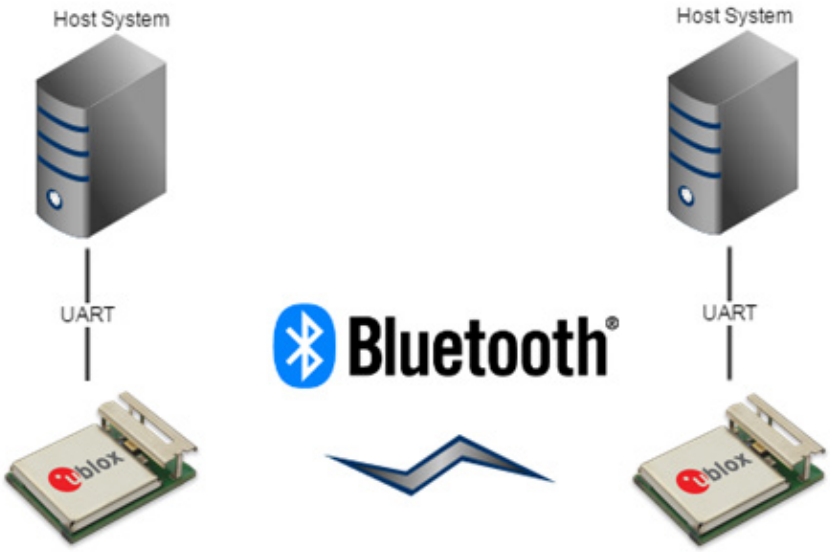



**Figure 1: NINA-B31 series Bluetooth setup**

## 1.2 Modes of operation

The NINA-B31 series module can operate in the following three different modes:

- Command mode
- Data mode
- Extended Data mode

In command mode, the module is configured using AT commands. For information regarding the available AT commands, see u-blox Short Range Modules AT Commands Manual [1].

In data mode, the module transmits data transparently between the serial UART and wireless connection(s).

The extended data mode (EDM) allows individual control of each active link. While sending and/or receiving data, AT commands can simultaneously be sent to the module from the host. For more information about EDM, see section 2.6 and u-blox Extended Data Mode Protocol Specification [7]).

## 1.3 u-blox Low Energy Serial Port Service (SPS)

In Bluetooth BR/EDR, there is a Serial Port Profile (SPP) that emulates a serial port over the air. For Bluetooth low energy, there is no such profile and thereby no standardized way of transferring generic data over the air.

The u-blox Low Energy Serial Port Service is a non-standard profile developed for the u-blox serial port adapter concept. The specification is open and may be implemented in any Bluetooth Low Energy device to enable generic data transmission.

The serial port service is implemented on top of the Generic Attribute Profile (GATT). It contains the following characteristics:

- FIFO for reading and writing data
- Credits to simulate the Bluetooth BR/EDR credit-based flow control

To connect, the central/client side sets up the ACL (Asynchronous Connection-Less) link and enables indications for both the FIFO and credits characteristics. Then the server side sends the number of credits to the client using the credit indication, and the client writes the number of credits to the server. The credits correspond to the number of packets the remote side may write before receiving new credits. The received credits are added to the remaining ones (if any).

When the connection has been established and credits have been received, data transmission starts. The client writes to the FIFO and the server sends indications or notifications with the FIFO data. Connections without credits are also supported. Connections without credits are faster to establish and ideal for scenarios where a small amount of data shall be transferred.

For more information, see the u-blox Low Energy Serial Port Service Protocol Specification [10].

## 1.4 NFC

The NINA-B31 series includes NFC (Near Field Communication), which can be used to “touch” NINA-B31 to an NFC reader device to perform actions. NFC can simplify the connection process to a module and reduce the number of steps when provisioning an NINA-B31 module. NFC can also be used to launch apps on smartphones and store a web link or a text.

An NFC antenna is included in the NINA-B31 evaluation kit.

# 2 Configuration

You can configure the NINA-B31 module according to your specific requirements by means of AT commands (see u-blox Short Range Modules AT Commands Manual [1]). The easiest way to get started is to use the u-blox s-center, which is a graphical user interface for sending the AT-commands (see s-center Product Summary [4]). The u-blox s-center enables easy configuration for the most common AT commands.

## 2.1 Basic settings

There are some basic commands for controlling the general Bluetooth behavior of the module as listed below.

- Connection Mode - AT+UBTCM: Configures the module for incoming connections.
- Discover Mode - AT+UBTDM: Configures the visibility for remote devices making inquiries.
- Pairing Mode - AT+UBTPM: Configures the ability to pair (authenticate) to remote devices.
- Bluetooth name - AT+UBTLN: Sets the name presented to remote devices making inquiries or name requests.

Once pairing has been done with a remote device, it is recommended to disable both discover and pairing mode for security reasons and performance.

## 2.2 Peers

A connection consists of a sender and a receiver of data. Every sender and receiver in a setup is referred to as a peer. Thus, a peer is capable of either receiving or sending data.

There are two kinds of peer classes in the serial port adapter:

- Local peer
- Remote peer

The local peer is synonymous with the UART. In contrast to the local peer, the remote peer is another device. Several remote peers can be defined if a multidrop scenario is needed.

A remote peer is addressed using a Uniform Resource Locator, URL. These locators are strings representing the nodes on internet or on a local net. This is the same addressing technology used in case of a web browser. For more information about URLs, see http://www.rfc-base.org/txt/rfc-1738.txt.

In general, URLs are written as follows:

- <scheme>:<scheme-specific-part>

Where <scheme> is the scheme or protocol used when communicating and <scheme-specific-part> is normally the address and port number of the remote node. For example, a web server on the internet can have the following address:

http://www.u-blox.com/
This tells the browser to use the HTTP protocol and connect to the node at address http://www.u-blox.com/. A similar addressing scheme is used by NINA-B31 to pinpoint the remote peer. The scheme is not "http", but the node addressing is identical.

Available schemes:

- sps, which is for the Bluetooth low energy u-blox Serial Port Service

Syntax:

- sps://bluetooth_address/

Example

- sps://0012f3000001

A peer can be set up using either the default remote peer command **AT+UDDRP** or dynamically created using the connect peer command **AT+UDCP**. A connection is closed using **AT+UDCPC**.

☞ The Bluetooth low energy SPS service is enabled by default in NINA-B31 and it accepts incoming connections by default.

## 2.3 Client and server

A client initiates a connection and the server accepts an incoming connection.

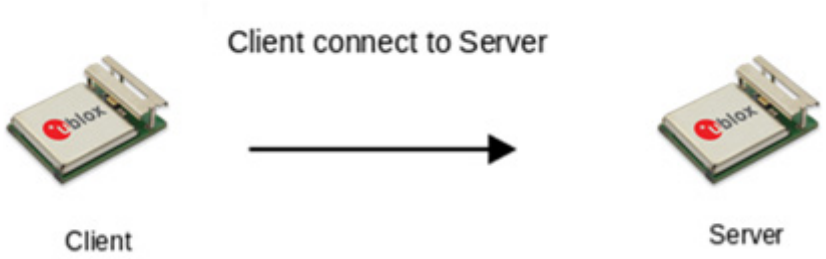



**Figure 2: NINA-B31 as a client connecting to the NINA-B31 server**

To configure a client to automatically set up a connection to a server, configure the correct Bluetooth service and the remote peer. The Bluetooth service controls the type of connection that is requested (see section 2.4) and the preferred server to connect is defined by the remote peer.

- **Default Remote Peer:** **AT+UDDRP** configures server (profile and address) to connect and when to initiate the connection. Peer is enumerated starting with id 0.
- **Server Configuration:** **AT+UDSC** only accepts incoming connection attempts for the configured server profile. NINA-B31 has the SPS enabled by default.

## 2.4 2 Mbps physical layer (PHY)

In Bluetooth 5, a new physical layer has been introduced. This physical layer doubles the modulation rate as compared to Bluetooth 4.2. The change from 1 Mbps to 2 Mbps is done by doubling the symbol rate; because of this, the bandwidth becomes twice as wide. The Bluetooth low energy already has 2 MHz channel centers so the increased bandwidth does not reduce the number of channels.

- **Preferred Transmitter/Receiver PHY:** **AT+UBTLECFG=27/28** sets the preferred PHYs to be used as response for peer-initiated PHY update requests.
- **Low Energy PHY Request:** **AT+UBTLEPHYR** requests a change of PHY for the specified connection.

## 2.5 Bluetooth services

The Generic Attributes (GATT) is used when Bluetooth LE devices exchange data. There are two roles defined in GATT – the client and server. For more information about GATT, see [8].

The GATT server stores data received from the GATT client. The GATT server sends responses to requests and sends indications and notifications to the GATT client when specified events occur on the GATT server.

Official as well as user-defined GATT Services can be set up with AT commands. The Serial Port Service (SPS) is a u-blox proprietary serial cable emulation service, which is enabled by default.

## 2.6 Command mode, data mode, and extended data mode

By default, NINA-B31 enters command mode and has to be reconfigured to start up in data mode or extended data mode. From the data mode or extended data mode, it is possible to enter the command mode by transmitting escape sequence to the module. By default, the escape sequence is:

1. Silence 1 second
2. +++
3. Silence 1 second

☞ The +++ must be sent within 200 ms, which means that it is difficult or impossible to enter the escape sequence manually using a terminal window though the characters can also be typically pasted instead. The module leaves the command mode and enters data mode using the **ATO1/ATO2** command. It is also possible to toggle the UART DTR pin from High to Low to enter the command mode.

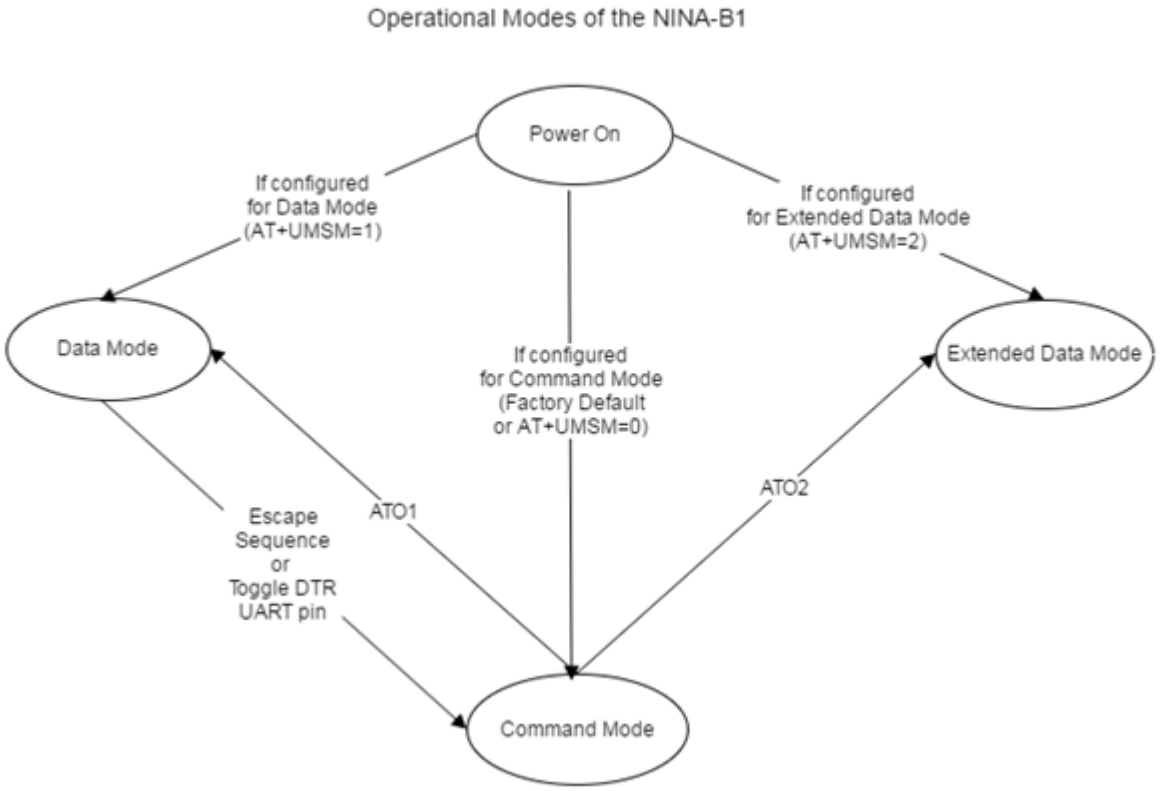



**Figure 3: Operational modes**

In data mode, the data transmitted on the UART to the module is transmitted (and broadcast for all connections) over the air to all wireless connections. Data received from the wireless connections may be interleaved upon reception and it may be difficult or impossible to figure out from which remote device the data is received. This "multipoint strategy" is known as Wireless Multidrop. It is suitable mainly for master/slave polled protocols such as Modbus and point-to-point communication.

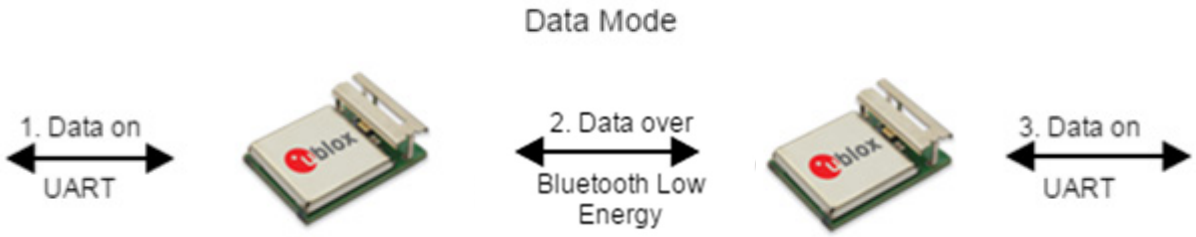



**Figure 4: Data mode**

The extended data mode is a simple protocol for the UART, which enables transmission of data to one specific remote device and also informs the remote device from where the data originated. It is also possible to execute AT commands as part of the extended data mode protocol. Hence, it is not necessary to enter the command mode from the extended data mode.

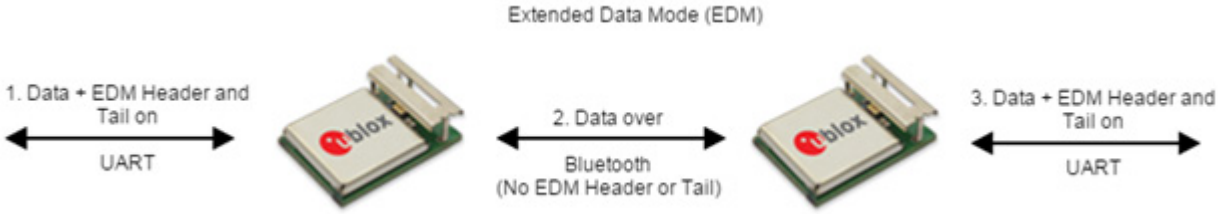



**Figure 5: Extended data mode**

☞ No extended data mode protocol data is available over the air. Only "raw" data is transmitted in the same way as in the data mode over air. Hence, it is possible for one side to be configured for extended data mode and the other side for the standard data mode.

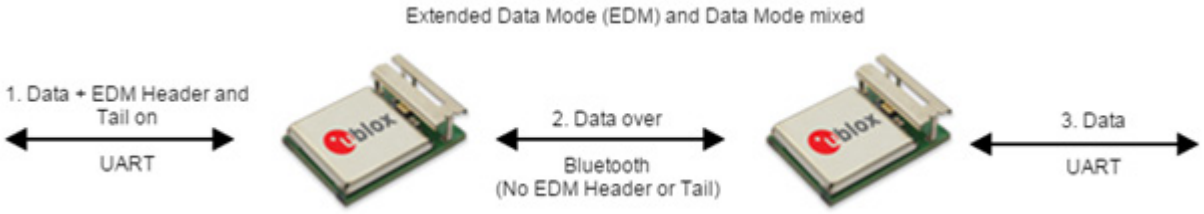



**Figure 6: Data mode and extended data mode**

## 2.7 Multipoint

It is possible to configure a module to support up to 8 parallel Bluetooth connections. These connections can be to 1 central and 7 peripherals in a multirole configuration. The **AT+UBTCFG** command is used to set the maximum number of Bluetooth connections.

In data mode, data is transmitted over air to all the connected devices and the data received from remote devices is interleaved.

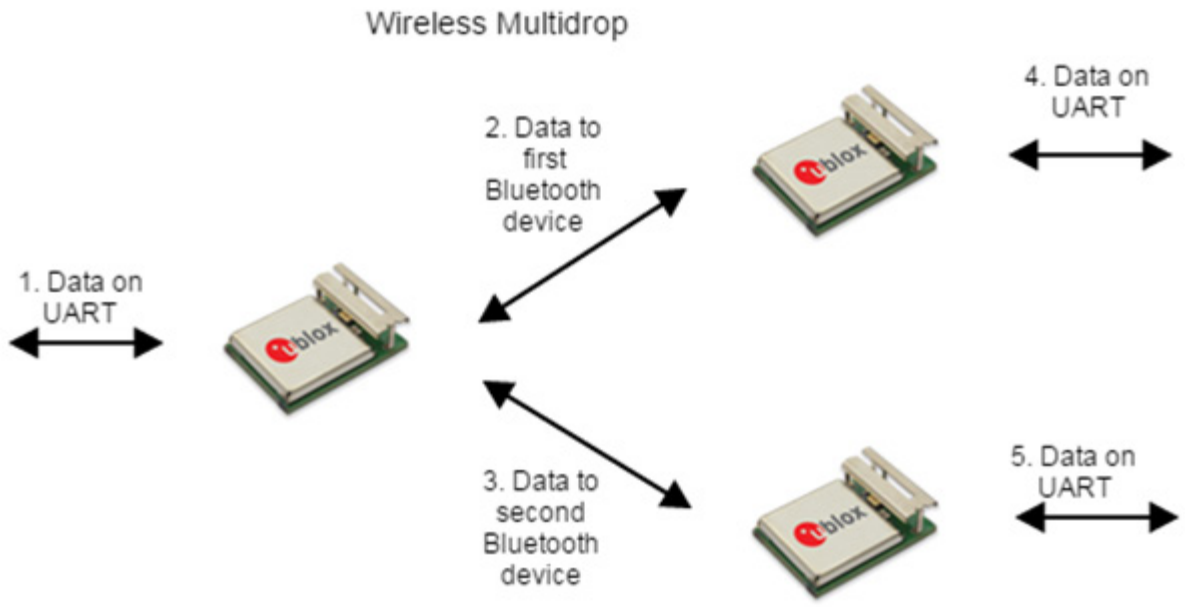



**Figure 7: Wireless Multidrop**

In extended data mode, it is possible to transmit data to a specific remote device and to know from which remote device data is received (see u-blox Extended Data Mode Protocol Specification [7]).

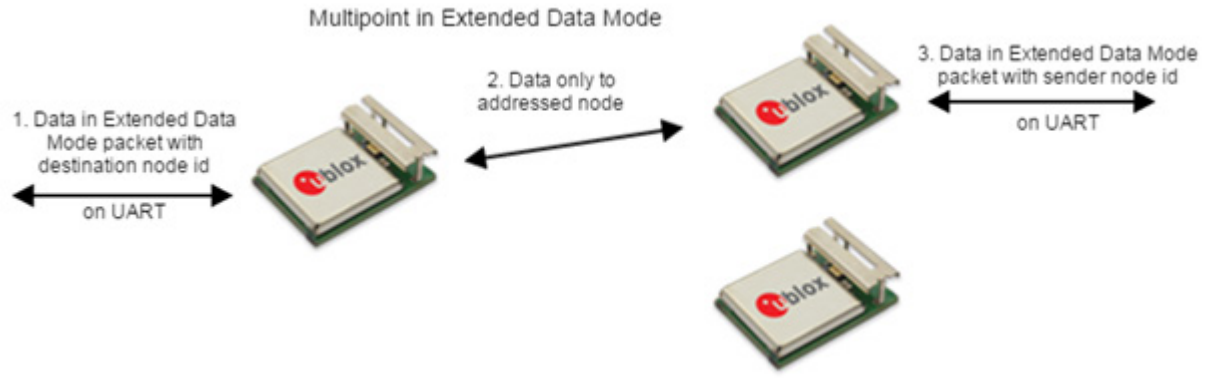



**Figure 8: Multidrop in extended data mode**

## 2.8 Bluetooth security

There are six different security modes that support all kinds of use cases regarding the pairing procedure.

All security modes use encryption. The encryption algorithm is a 128-bit cipher called E0.

Security mode 1 and 2 are implemented to keep the same or similar behavior as for previous versions of u-blox Bluetooth products.

Secure Simple Pairing (SSP) was introduced in Bluetooth 2.1 to improve security and simplify the pairing process. SSP uses a form of public key cryptography and the just works, numeric comparison, passkey entry, and out of band authentication mechanisms.

- Just works: No user interaction is needed, except that the user may need to confirm the pairing process. The method is typically used by devices with very limited IO capabilities.
- Numeric comparison: A 6-digit numeric code is displayed on each device where the user compares the numbers to ensure they are identical.
- Passkey entry: Either a 6-digit numeric code is displayed on one device that the user enters on the other device, or the same 6-digit numbers are entered on both devices.
- Out of band: This security mode is used to connect to a remote device when the temporary key has been exchanged out of band.

Use the security mode 1 (Bluetooth 2.1 auto accept security) **AT+UBTSM=1** to start with. It is highly recommended to analyze the security and if appropriate, use a more secured solution, for example security mode 2 (Bluetooth 2.1 Just Works) **AT+UBTSM=2**.

## 2.9 Low power modes

The Standby mode is one of the power saving modes in the NINA-B31 module that essentially powers down the module but keeps the system RAM and allows incoming connection to the module. The Sleep mode is the deepest power saving mode of NINA-B31 modules, and does not allow any connection to wake up the module. The low power modes in NINA-B31 are set using the AT&D command.

### 2.9.1 Standby mode

The standby mode in NINA-B31, enabled with **AT&D3**, is used to disable the UART. The radio is still active and the UART can be activated again either on a logical 0 to logical 1 signal transition on the DTR line, or by an incoming Bluetooth connection establishment.

For more information about Standby mode, see NINA-B3 series Data Sheet [2].

### 2.9.2 Sleep mode

The sleep mode in the NINA-B31 module is enabled with the command **AT&D4**, which causes a logical 0 to logical 1 signal transition of the DTR line to completely shut off the module. Upon a logical 1 to logical 0 signal transition on the DTR line, the module is started again. This is the lowest power mode, and NINA-B31 cannot keep or make any connection in this mode.

For more information about Sleep mode, see the NINA-B3 series Data Sheet [2].

# 3 Evaluation board

There are two NINA-B31 evaluation boards:

- EVK-NINA-B311 has a NINA-B311 module with u-blox connectivity software and an antenna connector for connecting to external antennas.
- EVK-NINA-B312 has a NINA-B312 module with u-blox connectivity software and an internal antenna (a unique 2.4 GHz metal sheet antenna soldered on to the module).

## 3.1 LED indications and buttons

There are two operational buttons (SW1 and SW2) and one multi LED (DS9), as shown in Figure 9.

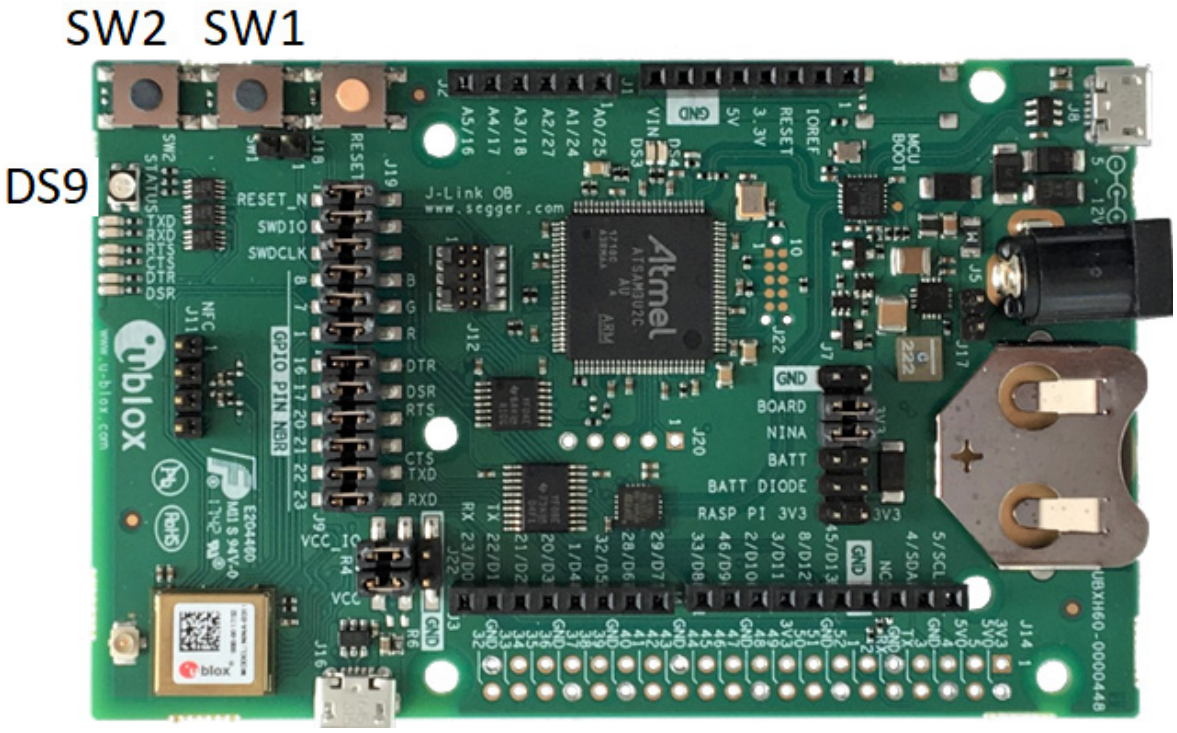



**Figure 9: EVK-NINA-B31 Evaluation board**

The LED indicates what mode is currently active and what activity is currently in progress. The following color indications are used. See EVK-NINA-B3 User Guide [3] for more information about the EVK-NINA-B3.

- Green: The current mode is data mode and no connection attempts are in progress.
- Orange: The current mode is command mode.
- Purple: A connection attempt is in progress.
- Blue: A connection is currently active.
- Blue Blinking: A connection is active and data is transmitted or received over air.
- Red Blinking: Error detected. Typically, this means buffer overflow, parity or framing error detected on the UART.

☞ The LED on the evaluation board is a 3-colour LED, which means that, on the module, it corresponds to three IO pins.

## 3.2 Restore default serial settings

If the SW2 button is pressed during power on, the module resets the serial settings and escape sequence to the default values.

- Default serial settings are 115.2 kbps, 8N1 and HW flow control enabled
- Default escape sequence is +++
- Default escape sequence timing is 1s silence before and after the escape sequence

## 3.3 Restore factory settings

If both the SW1 and SW2 buttons are pressed for 10 seconds during power up, the factory settings are restored. You can also restore to the factory settings using the **AT+UFACTORY** command followed by power off/on.

## 3.4 Enter the bootloader

To enter the bootloader, press the SW1 and SW2 buttons during a reset; this starts the module in the u-blox bootloader. You can also enter the bootloader using the **AT+UFWUPD** command. The bootloader will time out and resume the application after 10 seconds.

## 3.5 Flashing the module

The PC tool s-center can be used to flash NINA-B31. Instructions on flashing the NINA-B31 can be found in the Software section of the NINA-B3 System Integration Manual [11].

Go to the u-blox support webpage to obtain the latest available software.

# 4 Use case examples

☞ Reset your devices to factory mode using the following AT commands before executing all use cases

- Store configuration

  ```
  AT+UFACTORY
  ```

- Restart NINA-B31

  ```
  AT+CPWROFF
  ```

## 4.1 Connect two NINA-B31 using Bluetooth low energy

Connect two NINA-B31 modules using Bluetooth low energy Serial Port Service (SPS), which connects automatically.

**Set up first NINA-B31 as a Central (Device A)**

- Enable Central Role

  ```
  AT+UBTLE=1
  ```

- Default peer to Serial Port Service and always connected (use the address of Device B, if not known execute `AT+UMLA=1` on Device B to find the local address)

  ```
  AT+UDDRP=0,sps://112233445566,2
  ```

- Startup NINA-B31 in data mode

  ```
  AT+UMSM=1
  ```

- Store configuration

  ```
  AT&W
  ```

- Restart NINA-B31

  ```
  AT+CPWROFF
  ```

**Set up second NINA-B31 as a Peripheral (Device B)**

NINA-B31 has the u-blox Serial Port Service enabled by default:

- Startup in data mode

  ```
  AT+UMSM=1
  ```

- Store configuration

  ```
  AT&W
  ```

- Restart

  ```
  AT+CPWROFF
  ```

Now the blue LED on both devices should indicate that they are connected via SPS. You can test by typing characters on one console and see them appear on the other console.

## 4.2 Set up iBeacon on NINA-B31

**Set up NINA-B31 to advertise Apple iBeacon packets:**

☞ This is only an example use case. Apple requires iBeacon license agreement if it is used in a product.

iBeacon is a Manufacturer-specific event, and contains an 128 bit UUID such as D9B9EC1F-3925-43D0-80A9-1E39D4CEA95C. Note that the byte order for the AT command use the reverse byte

order. Together with the Major (2 bytes), Minor (2 bytes) and the Tx power (1 byte), it builds up the iBeacon advertise packet.

Apple recommends the Local Name in the Scan Response packet. See *[5]* for more information about iBeacon.

- To change advertise packet on NINA-B31 to include iBeacon:

```
AT+UBTAD=1AFF4C000215EBEFD08370A247C89837E7B5634DF52400010001C5
```

- To change Local Name to “Bluetooth Device”:

```
AT+UBTSD=1109426C7565746F6F746820446576696365
```

## 4.3 Set up Eddystone on NINA-B31

**Setup NINA-B31 to advertise Google Eddystone packets:**

☞ This example shows how to send an URL inside the advertisement packet; by nature, the advertisement packet is limited in size and the information is also limited.

The payload normally contains a short version of an URL, for example `https://goo.gl/Aq18zF`. The link is encoded to save space, but most smartphones can use this information to navigate to the advertised URL. See *[6]* for more information about Eddystone.

- To change advertise packet on NINA-B31 to include the Eddystone URL (in this example the URL will point to the u-blox home page) :

```
AT+UBTAD=0303AAFE0D16AAFE10F801752D626C6F7807
```

- To change Local Name to “Bluetooth Device”:

```
AT+UBTSD=1109426C7565746F6F746820446576696365
```

## 4.4 Set up Eddystone with Extended advertising on NINA-B31

**Setup NINA-B31 to advertise Google Eddystone packets with Extended advertising:**

This example shows how to send an URL inside the advertisement packet. Extended advertising is a Bluetooth 5 feature that allows bigger advertising packets, and in this example we will advertise a full URL.

- Enable Extended advertising on NINA-B31. This command requires a restart.

```
AT+UBTLECFG=29,1
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

- To change advertise packet on NINA-B31 to include the URL (in this example, the URL will point to the u-blox NINA-B3 home page) :

```
AT+UBTAD=0303AAFE2616AAFE10F801752D626C6F7800656E2F70726F647563742F6E696E612D62332D73657269657300
```

- To change Local Name to “Bluetooth Device”:

```
AT+UBTSD=1109426C7565746F6F746820446576696365
```

## 4.5 Change the NINA-B31 name

**NINA-B31 (Device A) is a Peripheral**

- Set Local Name on NINA-B31 to

```
AT+UBTLN="u-blox NINA-B31"
```

**NINA-B31 (Device B) as a Central**

- Set Central Role on NINA-B31

```
AT+UBTLE=1
```

- Store configuration

```
AT&W
```

- Restart NINA-B31

```
AT+CPWROFF
```

- Make a Device Discover to find Device A

```
AT+UBTD
+UBTD:112233445566p,-70,"u-blox NINA-B31",1,0F09752D626C6F78204E494E412D...
```

## 4.6 Set up GATT Server on NINA-B31

The following example shows how to set up a GATT Server on NINA-B31. The GATT Server value can be read by a remote device see the use case in section 4.7.

**Set up a GATT Server with a pre-defined service on NINA-B31**

- Define a Heart Rate service - A list of all available pre-defined services and its characteristics can be found at [8] and [9]

```
AT+UBTGSER=180D
+UBTGSER:30
```

- Add a Heart Rate measurement characteristic with notification support

```
AT+UBTGCHA=2A37,10,1,1
+UBTGCHA:32,33
```

- Optionally change the local name to a name indicating the devices new status:

```
AT+UBTLN="NINA-B3 GATT Server"
OK
```

## 4.7 Set up GATT Server / Client on NINA-B31

The following example shows how to set up a GATT Server on one NINA-B31 and receive notifications of updated values at a GATT Client on a second NINA-B31.

**Set up a GATT Server with a pre-defined service on NINA-B31 (Device A)**

See section 4.6 for a guide on how to set up Device A.

**Use the GATT Client on NINA-B31 (Device B) to receive Heart rate measurement values:**

- Enable Central Role

```
AT+UBTLE=1
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

- Find the other device

```
AT+UBTD=4,1
+UBTD:112233445566p,-59,"NINA-B3 GATT Server",1,1109426C756574...
```

- Create an ACL connection

```
AT+UBTACLC=112233445566
+UUBTACLC:0,0,112233445566p
```

- Use the connection handle **0** from the **+UUBTACLC** and Discover Services

```
AT+UBTGDP=0
+UBTGDP:0,1,9,1800
+UBTGDP:0,10,13,1801
+UBTGDP:0,14,22,180A
+UBTGDP:0,23,29,01D7E9014FF344E7838FE226B9E15624
+UBTGDP:0,30,65535,180D
OK
```

- After finding the "Heart Rate service" (180 D), use the start handle **30** and end handle **65535** from the **+UBTGDP**, and Discover all characteristics of service

```
AT+UBTGDCS=0,30,65535
+UBTGDCS:0,31,10,32,2A37
OK
```

- After finding the "Heart Rate Measurement characteristics" (2A37), use the value handle **32** from the **+UBTGDCS** and end handle **65535** from the **+UBTGDP**, and Discover all characteristic descriptors

```
AT+UBTGDCD=0,32,65535
+UBTGDCD:0,32,33,2902
OK
```

- After finding the "Client Characteristic Configuration descriptor" (2902), use the descriptor handle **33** from the **+UBTGDCD** and subscribe to notifications of Heart Rate measurement value changes

```
AT+UBTGWC=0,33,1
OK
```

**Update and notify new value of Heart Rate measurement (Device A):**

- We can now set a new value (**50**) and send a notification from Device A (the GATT Server)

```
AT+UBTGSN=0,32,50
```

**New value is notified to Device B:**

- An event is received when the remote side (Device A) sends a notification

```
+UUBTGN:0,32,50
```

## 4.8 Configure NINA-B31 over air

**Set up NINA-B31 (Device A) that should be configured over air**

- Enable remote configuration through the SPS server

```
AT+UDSF=0,1
```

- Enter data mode

```
ATO1
```

**Set up NINA-B31 (Device B)**

- Enable Central Role

```
AT+UBTLE=1
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

- Change escape character from default '+' (ASCII 43) to '-' (ASCII 45) to make the escape sequence pass through Device A without getting detected as an escape sequence at Device B

```
ATS2=45
```

- Find the other device (Device A)

```
AT+UBTD=4,1
+UBTD:112233445566p,-59,"NINA-B3 Device A",1,1109426C756574...
```

- Create an SPS connection to Device A

```
AT+UDCP="sps://112233445566"
```

- Enter data mode

```
ATO1
```

- Enter configuration mode by sending escape sequence of Device A. By default, the NINA-B3 escape sequence is:
  1. Silence 1 second
  2. +++
  3. Silence 1 second
- Device B can now send AT-commands to Device A, for example, reading the local Bluetooth address of Device A:

```
AT+UMLA=1
+UMLA:112233445566
```

## 4.9 NFC web link

Store a web link to the NFC tag; this web link will be opened in the browser of your NFC enabled smart phone when you touch the NFC antenna of NINA-B31 with your smart phone.

- Set URI to NFC tag

```
AT+UNFCURI=1,https://www.u-blox.com
```

- Enable NFC URI

```
AT+UNFCEN=2
```

- Touch the NFC antenna of NINA-B31 with the smart phone to go to the u-blox website.

## 4.10 NFC launch app

Store an Android application's package name to the NFC tag; this application will be opened on your NFC enabled smart phone when you touch the NFC antenna of NINA-B31 with your smart phone.

The Android package name is the unique identifier of an application. It can be found by searching for the app and finding the Google Play web page for the app. For example, the web page for Google Calendar is: https://play.google.com/store/apps/details?id=**com.google.android.calendar**

- Set URI to NFC tag

```
AT+UNFCURI=2,com.google.android.calendar
```

- Enable NFC URI

```
AT+UNFCEN=2
```

- Touch the NFC antenna of NINA-B31 with the smart phone to open the Google Calendar app

## 4.11 Pair NINA-B31 to a smart phone using NFC

Use NFC to initiate the pairing between NINA-B31 and an NFC enabled smart phone.

- Enable NFC OOB pairing

```
AT+UNFCEN=1
```

- Touch the NFC antenna of NINA-B31 with the smart phone to initiate the pairing

## 4.12 Pair two NINA-B31 using Out-of-Band security mode

Generate an OOB Temporary Key in NINA-B31. Another NINA-B31 can then use this OOB Temporary Key to authenticate during the pairing procedure.

**Set up NINA-B31 (Device A) as a Peripheral**

- Set OOB security mode

```
AT+UBTSM=6,0
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

- Generate a random OOB Temporary Key

```
AT+UBTOTK=0
```

- Read the random OOB Temporary Key

```
AT+UBTOTK?
+UBTOTK:239FBA31E3A435B10AF28B5FEF47DDC1
```

- Note the local address for pairing from Device B.

```
AT+UMLA=1
+UMLA:D4CA6EB00613
```

**Set up NINA-B31 (Device B) that should initiate the pairing**

- Set Central role

```
AT+UBTLE=1
```

- Set OOB security mode

```
AT+UBTSM=6,0
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

- Input the OOB Temporary Key generated in Device A

```
AT+UBTOTK=1,239FBA31E3A435B10AF28B5FEF47DDC1
```

- Pair to Device A (Bluetooth address D4CA6EB00613)

```
AT+UBTB=D4CA6EB00613p,1
```

## 4.13 Connect two NINA-B31 using 2 Mbps PHY

The default physical layer (PHY) is used when connecting two NINA-B31 modules

**Set preferred PHY on NINA-B31 (Device A)**

- Set 1 Mbps and 2 Mbps as preferred PHYs for Tx

```
AT+UBTLECFG=27,3
```

- and for Rx

```
AT+UBTLECFG=28,3
```

**Connect two NINA-B31 according to the use case in section 4.1.**

**When the connection is established, request a new PHY configuration for the connection (Device B)**

- Go to AT mode by sending the escape sequence:
  1. Silence 1 second
  2. +++
  3. Silence 1 second
- Request 2 Mbps PHY for both Tx and Rx on connection with handle 0

```
AT+UBTLEPHYR=0,2,2
```

- If the remote device (Device A) accepts the request, a successful event will be received

```
+UUBTLEPHYU:0,0,2,2
```

- Go to data mode

```
ATO1
```

## 4.14 Connect two NINA-B31 using Coded PHY

### 4.14.1 Connect using default physical layer

The default physical layer (PHY) is used when connecting two NINA-B31 modules.

**Set preferred PHY on both NINA-B31 (Device A and B)**

- Set Coded PHY as preferred PHYs for Tx

```
AT+UBTLECFG=27,4
```

- and for Rx

```
AT+UBTLECFG=28,4
```

**Configure Device A to be in the Central GAP role**

- Set Central role

```
AT+UBTLE=1
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

**Set up connection from Device A to Device B**

- Create an SPS connection to Device B (use AT+UMLA=1 to find the address of device B)

```
AT+UDCP=sps://223344556677p
```

- Make sure you get a successful connection

```
+UDCP:1
OK

+UUBTACLC:0,0,223344556677p

+UUDPC:1,1,4,223344556677p,20
```

- Request Coded PHY on the connection:

```
AT+UBTLEPHYR=0,4,4
```

- If the remote device (Device B) accepts the request, a successful event will be received

```
+UUBTLEPHYU:0,0,4,4
```

- Go to data mode (on both devices)

```
ATO1
```

### 4.14.2 Define CODED phy using extended advertising and scanning

**Set up Extended advertising and scanning on both devices (Device A + Device B)**

- Define Scanning and Advertising Coded PHY on both devices.

```
AT+UBTLECFG=29,2
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

**Configure Device A to be in the Central GAP role**

- Set Central role

```
AT+UBTLE=1
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

**Set up connection from Device A to Device B**

- Create an SPS connection to Device B (use AT+UMLA=1 to find the address of device B)

```
AT+UDCP=sps://223344556677p
```

- Go to data mode (on both devices)

```
ATO1
```

## 4.15 Change Device Information values of NINA-B31

Change the characteristics values of the Device Information service (UUID 0x180A) to the values of your choice.

- Read the current values of the Device Information service

```
AT+UBTLEDIS?
+UBTLEDIS:"u-blox","NINA-B3","4.0.0","4.0.0"
```

- Set the Manufacturer, Model, Firmware version and Software version to “Company”, “Model A”, “fw1.0” and “sw1.1” respectively.

```
AT+UBTLEDIS="Company","Model A","fw1.0","sw1.1"
```

- Store configuration

```
AT&W
```

- Restart

```
AT+CPWROFF
```

- Connect to NINA-B31 with for example, a smart phone using the nRF Connect app and verify the characteristics values of the Device Information service.

## 4.16 Write NINA-B31 GPIO pin and read NINA-B31 GPIO pin

Configure, write, and read GPIO pin using AT command. See NINA-B3 System Integration Manual [11] for GPIO pin id mapping.

**Preconditions**

- Connect GPIO pin 2 on NINA-B31 (Device A) to GPIO pin 25 on NINA-B31 (Device B)

**Set up NINA-B31 (Device A) GPIO pin**

- Check current configuration status of available GPIOs (result shows that all GPIOs are disabled)

```
AT+UGPIOC?
+UGPIOC:2,255
+UGPIOC:3,255
+UGPIOC:4,255
+UGPIOC:5,255
+UGPIOC:24,255
+UGPIOC:25,255
+UGPIOC:27,255
OK
```

- Set pin 2 as an input, no resistor activated

```
AT+UGPIOC=2,1,0
```

- Read the current value of pin 2 current value is 0 (zero)

```
AT+UGPIOR=2
+UGPIOR:2,0
```

**Set up NINA-B31 (Device B) GPIO pin**

- Check current configuration status of available GPIOs (result shows that all GPIOs are disabled)

```
AT+UGPIOC?
+UGPIOC:2,255
+UGPIOC:3,255
+UGPIOC:4,255
+UGPIOC:5,255
+UGPIOC:24,255
+UGPIOC:25,255
+UGPIOC:27,255
OK
```

- Set pin 25 as an output with initial value set to 0 (zero)

```
AT+UGPIOC=25,0,0
```

- Set pin 25 to 1 (one)

```
AT+UGPIOW=25,1
```

**Read the value of Device A's pin 2**

- Read pin 2 (Device A), the read value is 1 (one)

```
AT+UGPIOR=2
+UGPIOR:2,1
```

# Appendix

# A Glossary

| Abbreviation | Definition |
|---|---|
| **ACL** | Asynchronous Connection-Less |
| **BR** | Basic Rate |
| **DSR** | Data Set Ready |
| **DTR** | Data Terminal Ready |
| **EDM** | Extended Data Mode |
| **EDR** | Enhanced Data Rate |
| **GATT** | Generic Attributes |
| **GPIO** | General Purpose Input/Output |
| **HTTP** | Hypertext Transfer Protocol |
| **LED** | Light-Emitting Diode |
| **NFC** | Near Field Communication |
| **OOB** | Out Of Band |
| **PHY** | Physical Layer |
| **SPS** | Serial Port Service |
| **URL** | Uniform Resource Locator |

**Table 1: Explanation of the abbreviations and terms used**

# Related documents

[1] u-blox Short Range Modules AT Commands Manual, Document No. UBX-14044127
[2] NINA-B3 series Data Sheet, Document No. UBX-17052099
[3] EVK- NINA-B3 User Guide, Document No. UBX-17056481
[4] s-center Product Summary, Document No. UBX-15020022
[5] https://developer.apple.com/ibeacon/
[6] https://developers.google.com/beacons/eddystone
[7] u-blox Extended Data Mode Protocol Specification, Document No. UBX-14044126
[8] https://www.bluetooth.com/specifications/gatt/services
[9] https://www.bluetooth.com/specifications/gatt/characteristics
[10] u-blox Low Energy Serial Port Service Protocol Specification, Document No. UBX-16011192
[11] NINA-B3 System Integration Manual, Document No. UBX-17056748

☞ For regular updates to u-blox documentation and to receive product change notifications, register on our homepage (www.u-blox.com).

# Revision history

| Revision | Date | Name | Comments |
|---|---|---|---|
| R01 | 12-Jun-2018 | fbro, kgom | Initial release. |
| R02 | 13-Sep-2018 | cmag, mape | Included a note related to reset of device in section 4. Included additional use case examples (section 4.4 and 4.14). |

# Contact

For complete contact information, visit us at www.u-blox.com.

**u-blox Offices**

**North, Central and South America**

**u-blox America, Inc.**

Phone: +1 703 483 3180
E-mail: info_us@u-blox.com

**Regional Office West Coast:**

Phone: +1 408 573 3640
E-mail: info_us@u-blox.com

**Technical Support:**

Phone: +1 703 483 3185
E-mail: support@u-blox.com

**Headquarters**
**Europe, Middle East, Africa**

**u-blox AG**

Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support@u-blox.com

**Asia, Australia, Pacific**

**u-blox Singapore Pte. Ltd.**

Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

**Regional Office Australia:**

Phone: +61 2 8448 2016
E-mail: info_anz@u-blox.com
Support: support_ap@u-blox.com

**Regional Office China (Beijing):**

Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Chongqing):**

Phone: +86 23 6815 1588
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Shanghai):**

Phone: +86 21 6090 4832
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Shenzhen):**

Phone: +86 755 8627 1083
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office India:**

Phone: +91 80 405 092 00
E-mail: info_in@u-blox.com
Support: support_in@u-blox.com

**Regional Office Japan (Osaka):**

Phone: +81 6 6941 3660
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

**Regional Office Japan (Tokyo):**

Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

**Regional Office Korea:**

Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

**Regional Office Taiwan:**

Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com