BluEpyc BLE Devices

BLE Gateway and EchoBeacon operating modes

Summary

BluEpyc BLE Gateway

The Gateway has a Bluetooth Low Energy (BLE) interface with which it receives (observer) advertising packages from Beacons in its field of action and a TCP/IP (Ethernet or Wi-Fi) interface that allows the communication with the host.

The Gateway then allows you to send data with BLE source to a TCP/IP network; the IP address of the destination host is set via the Gateway parameters.

The BLE Gateway uses a binary communication protocol, which is documented in the related system manual.

Gateway data-set

The package sent from the Gateway to the host consists of a series of information, which represent the data set.

You can configure the data set using the configuration software or web interface, that allow you to specify the relevant information you wish to receive.

The information that constitute the data set is the following:
- **Address** of the Beacon
- **Advertising data**: contains the entire advertising package sent by the Beacon detected (maximum 31 bytes)
- **RSSI**: indicates the intensity with which the Beacon signal is received by the Gateway.
- **Date and Time** of detection
- **Antenna number**: antenna that has detected the Beacon
- **Digital Input**: State/Variation of digital inputs

Data-set configuration via BluEpyc Start software

The data set can be configured using the following configuration mask:
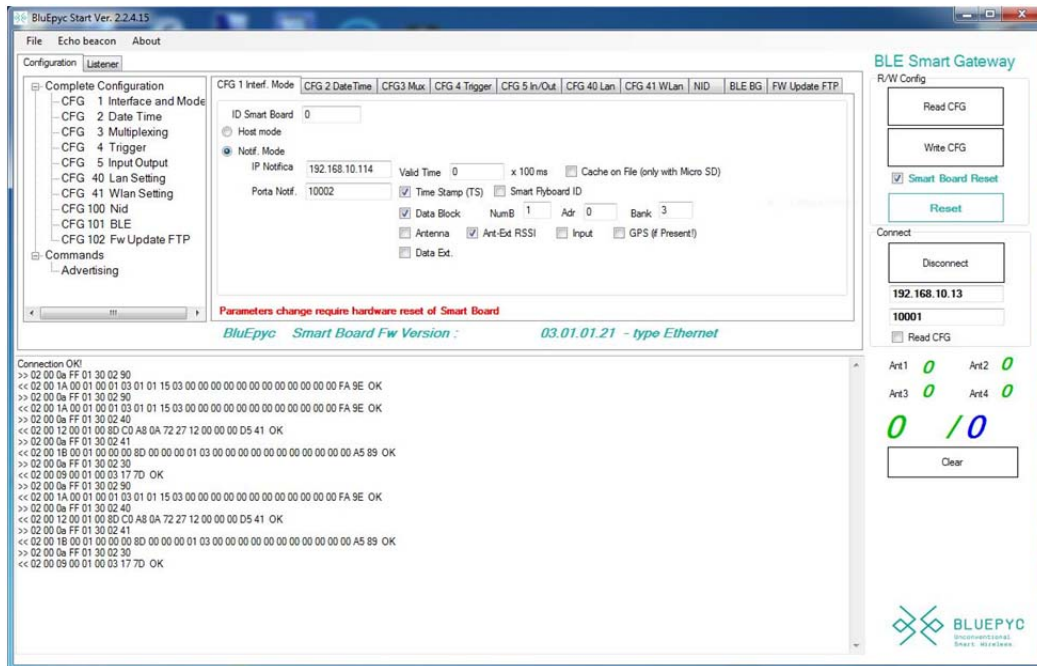
The most important information that allows you to identify a Beacon is:
- Beacon address
- advertising

This information, once set up in the data-set, allows the identification of the Beacon that has been detected (via its address) and to know the content that has been transmitted in the advertising.

Once received from the Gateway, the information is made available on the TCP/IP network.

Beacon detection using BluEpyc Start software

To verify that the BLE devices and settings are correct, it is possible to carry out detection tests using the BluEpyc Start software:
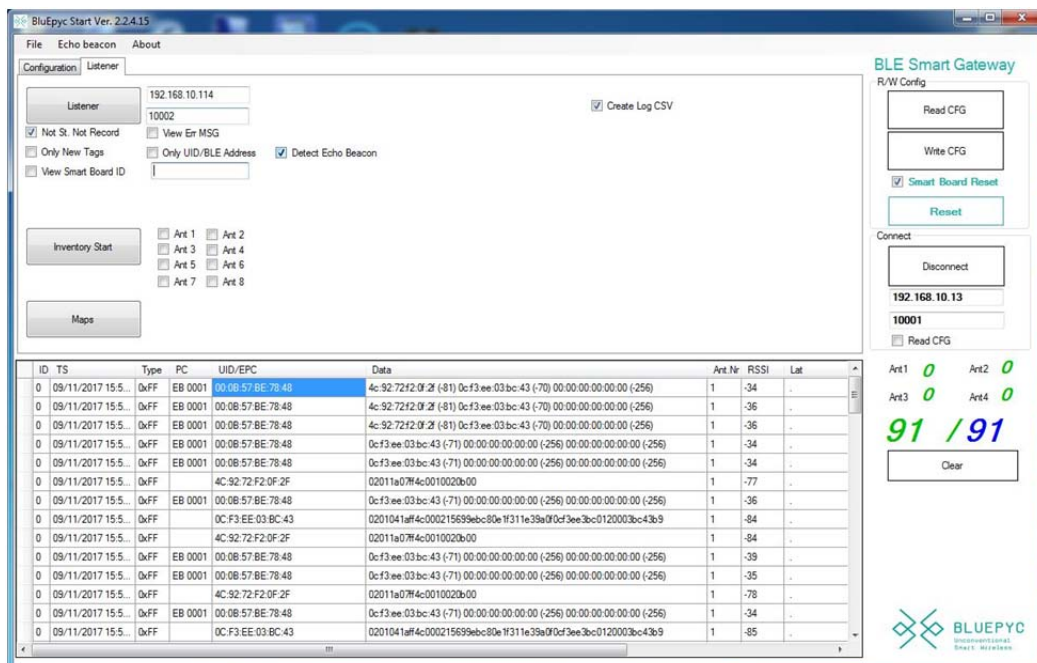
Available SDKs

In order to receive data that the Gateway sends to the destination IP address, you can:

- Directly process the data received by extrapolating it from the communication protocol
- Use the available libraries for Windows (. Net), Linux, RasPI, Android, MacOSX.
- Use the Windows service

Protocol development

You can listen to the reception port by extracting data from the communication protocol; this protocol is documented in the Gateway's reference manual .

This approach is a bit laborious and is generally useful mainly in cases where the host does not allow the use of high level development languages, such as PLCs, microcontrollers, etc.

Using libraries

The use of the libraries makes it easy to receive data that is sent to the host via asynchronous events. These data are represented in the form of intuitive and easily manageable objects, which can be processed quickly in subsequent operations, such as entering them into a database.

Windows service

The Windows service allows you to receive data from the Gateways by performing an initial time-based filter, which is used to eliminate double readings.

Once the data has been received, the service allows you to send them to one of the following destinations:

- SQL server database
- File archive CSV format
- Web service

In the case of SQL server database storage, predefined fields are filled in a SQL server table; the fields filled in basically reflect the set data set. For example, if the date-set is set to receive the Beacon address, its advertising and the date and time of detection, the corresponding fields will be filled in in the SQL server table.

Similarly, the fields corresponding to the selected data set will be filled in for the CSV file.

In the case of sending to a web service, the fields decided in the configuration of the data set are entered in a Json format and sent via http to the web server of destination. An example web server is provided, which exposes the web interface of reception in php.

Indoor or outdoor RTLS (Real-time locating systems)

The EchoBeacon is a device that has a Bluetooth Low Energy interface, like the Beacon. The difference lies in the fact that while Beacons are only able to broadcast, EchoBeacon is able to receive advertising packages (from the Beacons) and to retransmit them (with different power) to a listening Gateway.

This approach, by making appropriate transmission power and reception filter settings, allows the EchoBeacon to detect Beacons only within a limited area.

The detection and re-transmission procedures take place at the same time: the EchoBeacon is able to receive the signal coming from the Beacons and at the same time retransmit it.

Once the EchoBeacon has detected some Beacons within its competence area, it is able to retransmit their collected information to the Gateway.

The packets transmitted from the EchoBeacon to a Gateway contain all the data necessary to identify both the EchoBeacon (and thus the area) and the Beacons detected:

- **EchoBeacon address**
- **Address of the Beacon (s) detected**
- **RSSI (signal intensity) with which the Beacon has been detected**
- **Battery level and temperature detected**
  - This information is transmitted only when the related feature is enabled (Eddystone TLM packets detection enabled) and when surrounding beacons are transmitting Eddystone TLM packets.
- **Keybeacon status**
  - This information is transmitted only when the related feature is enabled and the Beacon detected is BluEpyc KeyBeacon type
  - The information contains the status of the button pressed on the KeyBeacon and the related output (relay) activated on the EchoBeacon side (this feature must be enabled on the EchoBeacon first).
- **Advertising Bytes from beacons**
  - This feature must be enabled on the EchoBeacon first. The maximum number of bytes coming from beacons that the EchoBeacon is able to forward in its packet is five.

Bu using more EchoBeacons linked to different areas, it is possible o know at any time the position of the detected Beacons. This structure allows the realization of an indoor or outdoor RTLS **(Real-time locating systems)** with the **Zone-Method**.

Echo packet sent from EchoBeacon

One of the packets sent by EchoBeacon is the Echo packet (with 32 Byte payload). This packet contains the Beacon detected information and has the following format (Bluetooth standard):

BLE Flag	BLE Manufacturer Specific Data	Source	Echo Beacon ID	Type	Packet Count	Beacon Address	Beacon RSSI	Packet Payload	Eddystone TLM Volt Temperature	Key Beacon state	Adv Source Info
3 Byte	4 Byte	1 Byte	2 Byte	1 Byte	1 Byte	6 Byte	1 Byte	1 Byte	4 Byte	1 Byte	6 Byte
BLE Information		Packet Information				Beacon Information		Advanced Beacon Information			

- ❈ **BLE flag**: in this field there are data BLE standard-compliant used to define the type of packet sent.
- ❈ **BLE Manufacturer Specific Data**: in this field there are data BLE standard-compliant which include the Company ID for the following custom data. BluEpyc Company ID is 0x05, 0xBE.
- ❈ **Source:** Identifier byte to identify the device that has sent the packet. In case of EchoBeacon packets this value is referred with 2
- ❈ **EchoBeacon ID**: EchoBeacon identification number
  - Range: from 1 to 65535
  - EchoBeacon can be identified with this ID or with its own BLE MAC Address. The BLE MAC Address is received from Gateway together with this packet.
- ❈ **Type:** packet- type; for Echo packet, this value is 3.
- ❈ **Packet Count:** counter value, increased for any packet transmitted
- ❈ **Beacon Address**: detected beacon address (6 Bytes)
- ❈ **RSSI:** Signal Strength with which the beacon has been detected
- ❈ **Packet Payload:** bit mask to identify the information present in the next Bytes (Eddystone TLM, Keybeacon State, etc.)
- ❈ **Eddystone TLM:** in order to be included in the packet, this feature must be enabled first; In case it is enabled this field contains the temperature (°C) and battery level (V) transmitted from the beacons detected. If the beacon detected does not communicate these information, this field is set to 0x00000000.
- ❈ **KeyBeacon state:** in order to be included in the packet, this feature must be enabled first; In case it is enabled and the beacon detected is KeyBeacon-type, this field contains the state of the button pressed or the related Output that must be activated on the Echobeacon.
  In case the beacon detected is not KeyBeacon-type, this field is set to 0x00.
- ❈ **Advertising Source info:** if this feature is enabled, this field contains a part of the advertising data coming from the detected beacon. This feature must be set first using the byte start byte position and length of the data that are meant to be included.

Watch Dog packet

If EchoBeacon does not detect any Beacons for a certain period of time (configurable), EchoBeacon still sends an advertising packet, called "Watch Dog", containing its identifier.
In this way it is possible to periodically check the correct operation of the EchoBeacon. The packet has the following format:

BLE Flag	BLE Manifacturer Specific Data	Source	Echo Beacon ID	Type	Packet Count					
3 byte	4 byte	1 byte	2 byte	1 byte	1 byte	19 byte				
BLE Information		Packet Information				RFU				

The packet format is very similar to the Echo packet one.
Since this packet is transmitted only when no beacons in the surrounding environment are detected, it does not contains beacons identifiers and data;
The packet-**Type** value for Watch Dog packet is referred with 2.

EchoBeacon parameters

Is possible to set the Echobeacn parameter through BLE exposed service.
Available Echobeacon parametrs are:

❖ **ID EchoBeacon**
- EchoBeacon identification number

❖ **Beacon Interval**
- Time distance between two transmitted packets (transmission frequency)

❖ **Watch Dog Interval**
- Time distance between two transmitted Watch Dog packet in case no packets are detected

❖ **Watch Dog Frame Number**
- Number of Watch Dog packet sent when the Watch Dog Interval elapses. When the Watch Dog Interval elapses if the EchoBeacon does not detect any beacon, it sends a number of Watch Dog packet equal to this value.

❖ **Advertising Frame Number**
- Number of Echo packets sent for every beacon detected. For every beacon detected, the Echobeacon sends a number of equal packets defined from this parameters.

❖ **RSSI filter**
- It is possible to set a filter on the signal strength with which packets from the Beacons are received. By setting appropriate values, is possible to increase or decrease the reception range, which determines the area monitored by the EchoBeacon.
- Only the advertising packages with signal strength value higher then filter value will be processed. All others packets will be discarded.

❖ **Advertising Mask**
- With this parameter is possible to apply a filter mask on the data that are currently transmitted from the detected beacons. All the beacons that transmit a data that fulfil the Mask will be processed, discarded all the others.

❖ **Valid Time**
- This parameter defines how long a single Beacon, after being transmitted from the EchoBeacon, should no longer be retransmitted. For example: With a 5-second Valid Time set in the EchoBeacon and a Beacon that transmits with a ping rate equal to 200 ms, the Echo packets related to the detected beacon will be transmitted first by the EchoBeacon and no

longer transmitted until 5 seconds are elapsed (even if the Beacon is detected several times in 5 seconds). This feature avoids overloading of transmissions with Beacons stationary near the EchoBeacon.

❋ **Transmission Power**
- This is the transmission power with which advertising packages are transmitted from EchoBeacon to the gateway
- By setting this value it is possible to retransmit the packets of the locally detected Beacons, at distances up to 100 meters.

❋ **Relays activation**
- In EchoBeacon Wall Mount version, two relays are implemented, which can be activated separately by using a specific parameter. If one or both relays are enabled through parameters, when a beacon that meets the filters conditions described above is detected, the corresponding relay is activated (closed) for a configurable time value.
- As example, through the relay it is possible to trigger a lock creating an access control system, or it is possible to activate an alarm light signal or any other kind of alarm.

❋ **Beacon List**
- It is possible to upload a Beacon list (Beacon MAC-address) on the EchoBeacon up to a maximum of 180 MAC-address. Using Beacon List functional parameters, it is possible to manage the behaviour of the EchoBeacon, in case Beacon detection take place; the Echobeacon working mode for this kind of scenario, is described in the following table:

Beacon	*Echo Advertising enable*	*Relay 1 activation*	*Relay 2 activation*
Present in Beacon List	Yes/No	Yes /No	Yes /No
NOT present in Beacon List	Yes /No	Yes /No	Yes /No

Using the beacon list and an appropriate parameter setting, it is possible to realize an automatic access control.
- For beacon present in the beacon list:
  - Echo advertising activation(with detected beacon information included)
  - Relay1 Activation enabled to provide a visual bright feedback for granted access
  - Relay2 Activation enabled for door opening
- For beacon **NOT** present in the beacon list:
  - No action.

Another use could be the creating of an access control for protected areas: in this scenario, the beacon list is the list of beacons that have the authorization to enter in the area. A possible parameter setting could be the following:

- For beacon present in the beacon list:
  - Echo advertising disabled
  - Relay1 Activation enabled to provide a visual bright feedback for granted access

- For beacon NOT present in the beacon list:
  - Echo advertising enabled
  - Relay1 Activation enabled to provide a visual bright/acoustic feedback for denied access

❖ **Advanced Beacon Information**

- You can enter additional information about the detected Beacon in the Echo packet using these parameters (see the Echo packet template in this document). This additional information can be activated individually and/or simultaneously and are:

  ▪ **KeyBeacon Status**

  If enabled, the status of the current Beacon detected (beacon must be BluEpyc KeyBeacon type) will be insterted in the Echo packet.
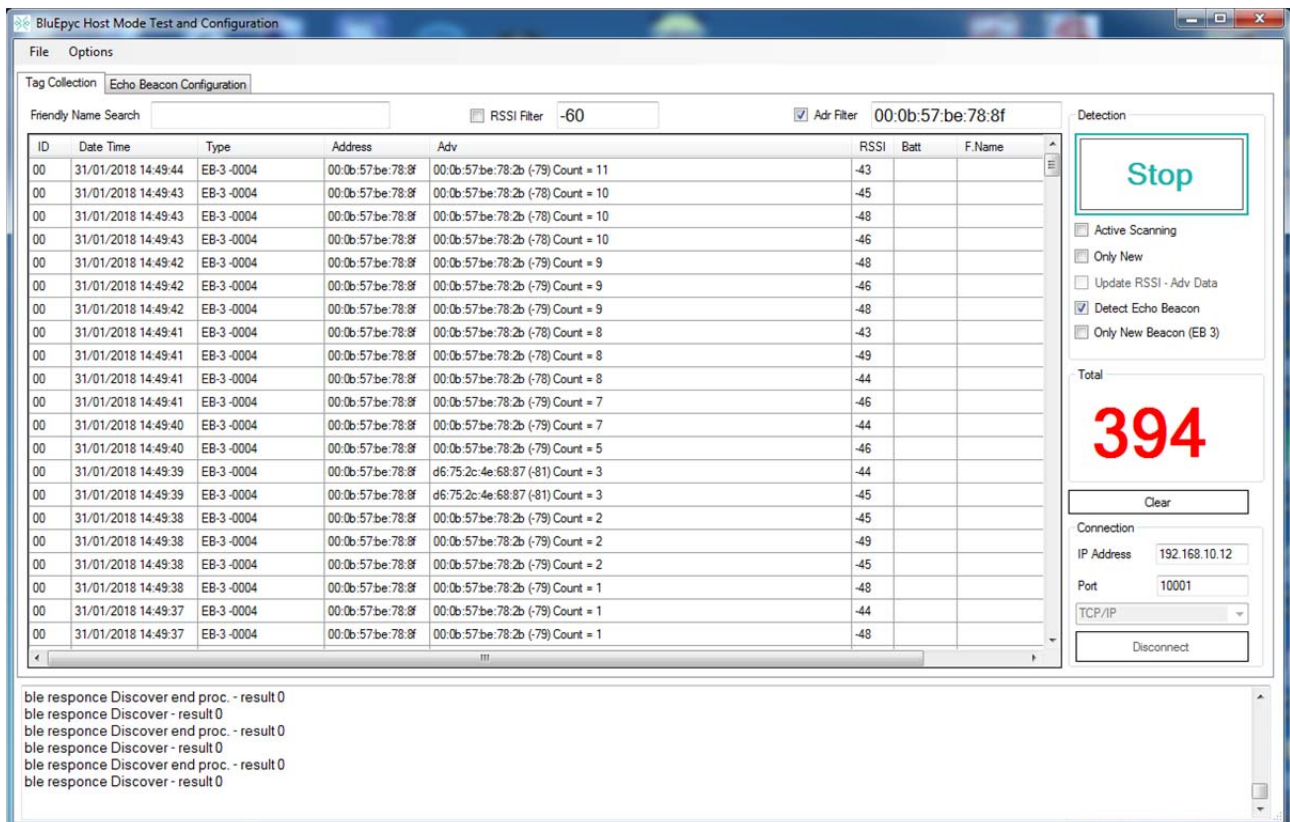
  ▪ **Eddystone TLM**

  If enabled, the battery level or the temperature of the current Beacon detected (beacon must transmit Eddystone TLM packet) will be insterted in the Echo packet.

  ▪ **Advertising Source Info**

  If enabled, a part of the advertising data transmitted from to the detected beacon will be inserted in the Echo packet

Data collection from EchoBeacon

The Gateway with the BluEpyc BLE Collector software allows you to receive and analyze data coming from both EchoBeacon and beacon.

EchoBeacon parameters configuration

The parameters of the EchoBeacon can be set on the BLE interface via the Android app, as shown in the example below:
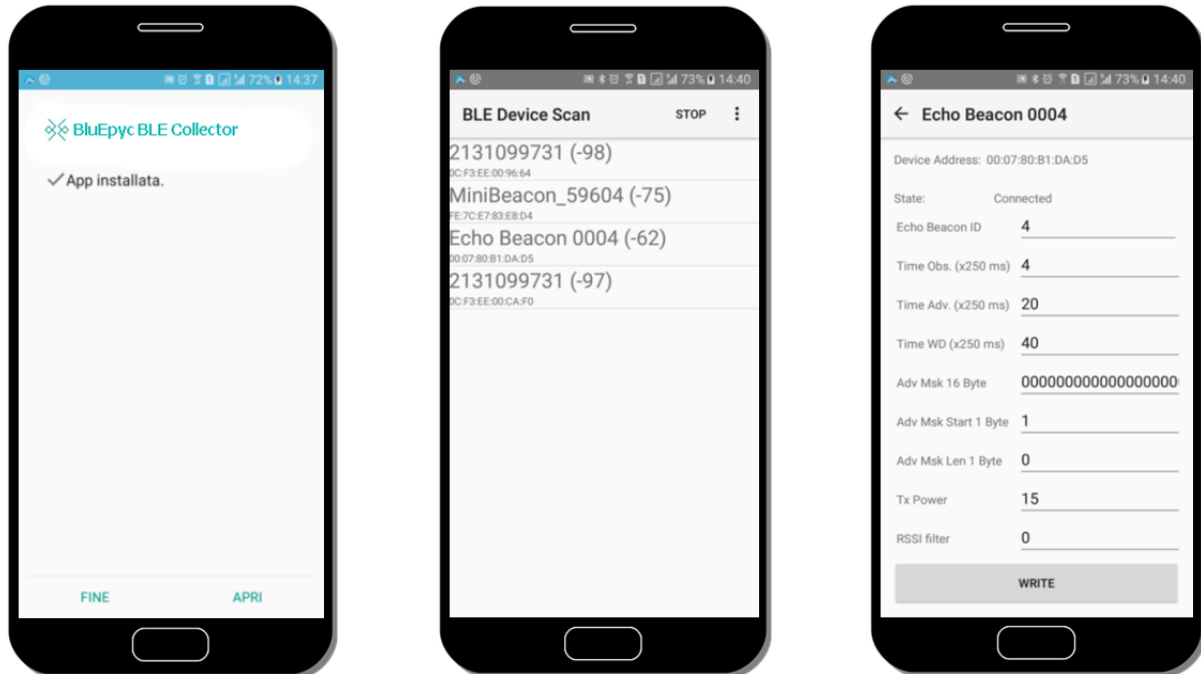
Alternatively, it is possible to perform the same operation via the Gateway, through the appropriate configuration software:

Advanced parameter setting and Beacon List