Application Note

How to Use BLE with the FOX3-3G-BLE Device

FOX3-3G-BLE and BLE Modes
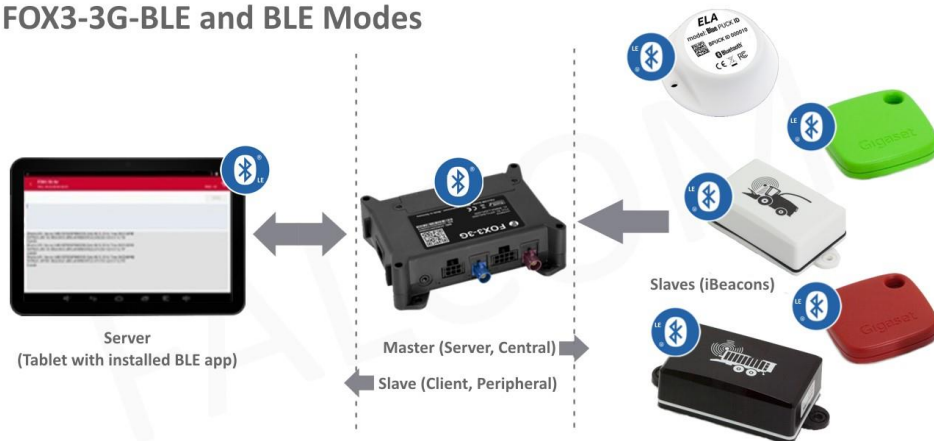
Server
(Tablet with installed BLE app)

Master (Server, Central)

Slave (Client, Peripheral)

Slaves (iBeacons)

Intellectual Property

Contacts

Lantronix, Inc.
7535 Irvine Center Drive, Suite 100
Irvine, CA 92618, USA
Toll Free: 800-526-8766
Phone: 949-453-3990
Fax: 949-453-3995

Technical Support
Online: www.lantronix.com/support

Sales Offices
For a current list of our domestic and international sales offices, go to the Lantronix web site at www.lantronix.com/about/contact

Disclaimer

Revision History

Date	Rev.	Comments
December 2016	1.0.0	Initial version.
May 2017	1.0.1	New document design Added new commands $PFAL,Sys.Ble.Send,<protocol>,"<text+dynamic entries>" and $PFAL,Msg.Send.Ble,<protocol>,"<text+ dynamic entries>" Added new configuration parameter "DEVICE.BLE.STARTUP=<on|off>"
May 2017	1.0.2	Renamed the app for android to "SPP Terminal Demo"
February 2019	1.0.3	Document throughout updated. Added sections 1.2 and 1.3, and 1.4.1 to 1.4.4. Added new commands/events/dynamic variables – see table in section 1.5 Extended section 1.7
June 2018	1.0.4	Added new dynamic entries for the released iBeacons - see table in section 1.5 Added new configuration parameter BLE.SCANDURATION=<ScanDuration> - see table in section 1.5
October 2019	A	Initial Lantronix document. Added Lantronix document part number, logo, contact information, and links.

For the latest revision of this product document, please check our online documentation at www.lantronix.com/support/documentation.

Table of Contents

1 INTRODUCTION

This guide shows how to configure your FOX3-3G-BLE device, connect it to the evaluation Android app and monitor device activities via the BLE interface or receive the data advertised from iBeacon sensors and forward their data to a remote server for analyzing.

1.1 General

Bluetooth® Low Energy (BLE) (also called Bluetooth® Smart or Version 4.0+ of the Bluetooth® specification) is the power- and application-friendly version of Bluetooth® that was built for the Internet-of-Things (IoT). The FOX3-3G-BLE is a 3G vehicle telematics device that features Bluetooth Low Energy connectivity for in-car access, company fleets, car-sharing, and car-rental applications to connect, share or distribute information in real time. FOX3-3G-BLE can operate either as a slave (peripheral) device or a master (central) device depending on user required functionality of the device. FOX3-3G-BLE (Bluetooth Low Energy) has a Line Of Sight (LOS) range of up to 50* meters.

As a slave, the FOX3-3G-BLE device connects to another BLE enabled device with an installed BLE App (e.g. on BLE smartphone or tablet) and exchanges data. This mode can be used, for example, for **HoS (**Hours of Service**) solutions** to review driving history on the app HoS on tablet and submit electronically HOS records to a remote server.

The Android app (**for evaluation purposes only**) called "**BLE Terminal Demo**", available by request from technical support, can be used to test and evaluate the BLE functions of the FOX3-3G-BLE on your smartphone or tablet. This demo application runs under **Android 4.4** and higher.

As a master, the FOX3-3G-BLE device receives the information sent out from other BLE iBeacons (peripherals, slaves), collects their data and sends this information over TCP to a remote server. This mode can be used for example for **IoT solution**s in transportation and logistics to connect iBeacon sensors attached to objects and identify them on a remote platform server. The combination of FOX3-3G-BLE and iBeacons allows fleet operators to build solution that can track the transport route of containers, trailers or swap bodies via the tractor, including timestamps, GPS tracking, route tracking, geofence sector monitoring and GPS logbook.

> ** The effective signal range will be reduced depending on the installation location of the device and the material and nature of the obstructions.*

1.2 What is needed to get started with FOX3-3G-BLE (as Slave)?

If you are going to use the device as a slave device, which means to exchange data with another master device, the following cables and apps should be available to operate with FOX3-3G-BLE device:

1. **FOX3-3G-BLE** with firmware version 3.x.x and its control-box (included in the promotion kit) are needed to connect the device to a PC and perform device configuration for the first time.
2. **CA31 for in-vehicle installation:** Main port extended installation cable with 2x4pin connector and 6 x open ends as well as 2x2pin UCOM connector can be used to power the device, activate low power modes with wake up reason "IGN", monitor the state of input/output, connect to the CAN-FMS gateway of the vehicle, connect a serial device (e.g. a passive or active RFID reader) to the 1st RS-232 serial (8pin connector) port of the FOX3-3G-BLE.
3. Smartphone or tablet with Android 4.4 or higher and installed "**BLE Terminal Demo**" Android app, provided for test purposes.
4. **Workbench software** to load the configuration into the FOX3-3G-BLE device.
5. Power source from +12 to +32 VDC (e.g. 12 VDC in your vehicle)

1.3 What is needed to get started with FOX3-3G-BLE (as Master)?

If you are going to use the device as a master device, which means to collect data from iBeacons and send this information over TCP to a remote server, the following cables should be available to operate with FOX3-3G-BLE device:

1. **FOX3-3G-BLE** with firmware version 3.x.x, its control-box (included in the promotion kit) that is needed to connect the device to a PC and perform device configuration for the first time
2. **CA70** or **CA31 for in-vehicle installation: (CA70) -** Main port basic installation cable, with 2x4pin connector and 3 x open ended wires, can be used to power the device and activate low power modes with wake up reason "IGN"; **(CA31)**- Main port extended installation cable with 2x4pin connector and 6 x open ends as well as 2x2pin UCOM connector can be used to power the device, activate low power modes with wake up reason "IGN", monitor the state of inputs/outputs, connect a serial device (e.g. a passive or active RFID reader) to the 1st RS-232 serial (8pin connector) port of the FOX3-3G-BLE.
3. iBeacon sensors (e.g. ELA iBeacon sensors)
4. **Workbench software** to load the configuration into the FOX3 device
5. Power source from +12 to +32 VDC (e.g. 12 VDC in your vehicle)

1.4 Hardware setup

This chapter explains how to connect and get started with FOX3-3G-BLE. Follow this four step process to get started with BLE:

1. Connect FOX3-3G-BLE device to your PC with the help of Control-box or USB cable.
2. Install the Workbench software and load the configuration into a FOX3-3G-BLE device.
3. Start testing the device configuration for BLE applications.

1.4.1 Connecting a FOX3-3G-BLE device to your PC

To connect the FOX3-3G-BLE device to your PC either use the Control-box (included in the promotion kit) or a USB cable* *(USB 2.0 Cable A-Male to Mini-B-5pin-Male)*, connect it to the USB port on the device and install the USB driver. The USB driver can be downloaded from our website:

  https://www.lantronix.com/support/.

* Please note that, when using the USB cable for device configuration you have to apply extra power to the FOX3-2G/3G/4G series.

If you are going to use the Control-box for connection of FOX3-3G-BLE device to a PC, go to our website https://www.lantronix.com/products/fox3-series/ and download the document "FOX3/-3G PROMOTION-KIT Getting Started" – then refer to the chapter 3.1.2 in that document.
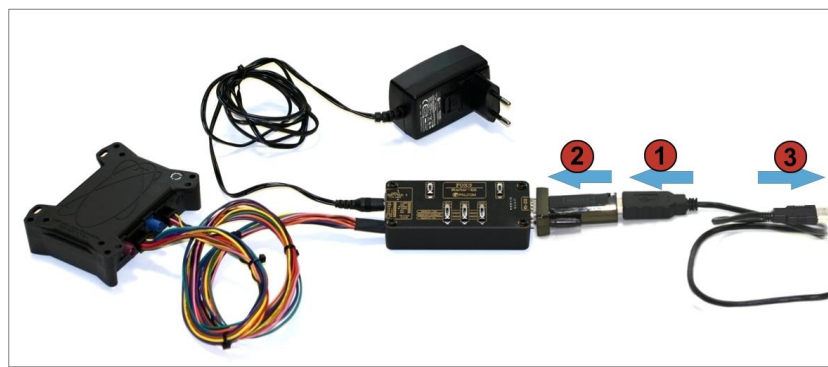

Figure 1:   Use the Control-box for connection of a FOX3-2G/3G/4G series device to a PC

1.4.2 Installing the Workbench software and loading the configuration into the device

To be able to communicate with the FOX3-3G-BLE device and load a configuration onto the device, you need to install the **_Workbench_** software. Go to the following link to download it: https://www.lantronix.com/products/workbench/

System requirements for using the Workbench software:

(a) PC with 700 megahertz or higher processor Intel Pentium II or compatible processor recommended,
(b) 512 megabytes (MB) of RAM or higher recommended (512MB minimum supported; may limit performance and some features),
(c) 90 megabytes (MB) of hard disk space available (recommended 2 gigabytes),
(d) Keyboard and Mouse.

After downloading the Workbench software, double click on the file to install it. An install shield will guide you through the installation. Integrated online help can be found once the installation is completed.
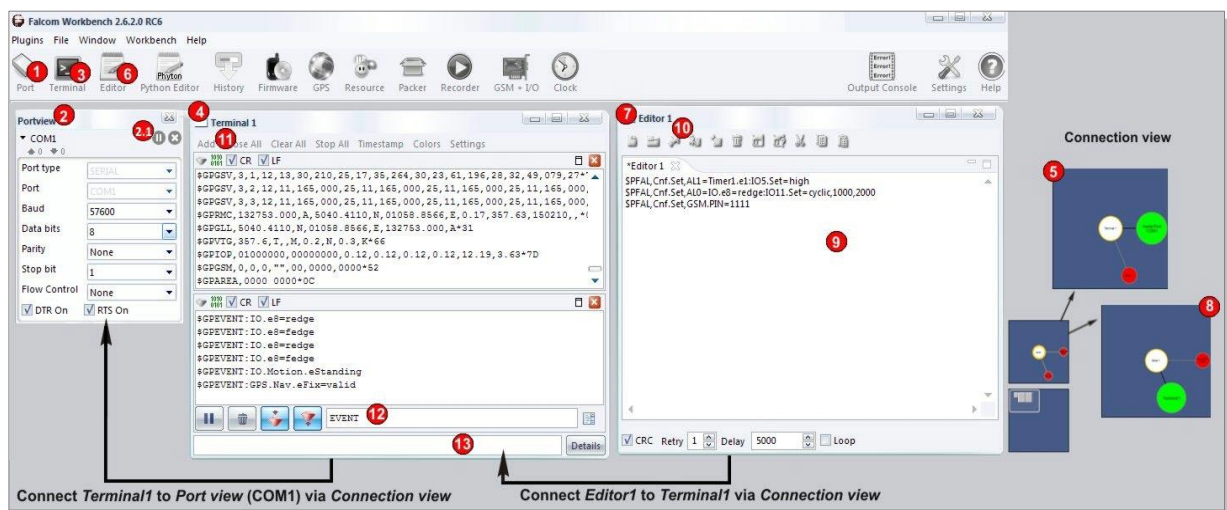

Figure 2:   Workbench software

1. After completing the installation, double-click on **Workbench.exe** to start it. Open a new **COM Port** from the Toolbar **(1)**, on the **COM Port view (2)**, choose the port where FOX3-2G/3G/4G is connected (go to Device Manager > Ports > USB Serial Port (COMxx), where **xx** is the COM port to use), define the port settings (**115200** bps, **8** Data bits, **No** Parity bit, **1** Stop bit, **None** Flow control) and finally, click the connect (play) icon **(2.1)** to open that COM port. You can also get an online help in HTML format if you click the "**Help**" button on the upper-right hand corner.
2. Open a new **Console** from the Toolbar **(3)**, click on the **Console1 (4)**, then go to **Connection view** and click on **COMPort (5)**.
3. Open a new **Editor** from the Toolbar **(6)**, click on the **Editor1 (7)**, then go to **Connection view** and click on **Console1 (8)**.
4. To load the configuration or send commands to the FOX3-3G-BLE device, type them in the **Editor (9)**, then click on **Start sending configuration (10)** or double-click with the left mouse key on each configuration line individually, in the **Editor (9)**.
5. Now, copy the configuration lines from one of the tables in chapter 1.4.4.2 or 1.4.4.3 below and past them in the **Editor (9)** and perform the step 4 above.

1.4.3 Powering up the FOX3-3G-BLE with the help of either CA70 or CA69

After the device has been successfully configured, apply power to the device using either cable CA70 or CA31. To apply power to the device, plug in the 8pin connector of one of these cables to the 8pin connector of the FOX3-3G-BLE device. Finally, connect the power pins (wires at the end of the used cable - IN+, GND - refer to the package insert of the device) of the used cable to an external power source ranging from +12 to +32 VDC (e.g. 12 VDC in your vehicle).

1.4.4 Testing the BLE modes of the FOX3-3G-BLE device

This section describes how to test the configuration examples given in tables in the sections below. After loading the configuration examples in the sections below into the FOX3-3G-BLE device and applying power, the device is ready for testing its BLE modes. For more details about PFAL commands related to the BLE, refer to section below 1.4.4.1.

1.4.4.1 PFAL commands implemented for BLE in the AVL firmware 3.0.x and 3.1.x

The table below gives you the list of PFAL commands/events/dynamic entries which are implemented in AVL firmware 3.x.x, related to use with the BLE functionality.

Configuration Parameter Settings	Description
CNF.Set,DEVICE.BLE.STARTUP=<on|off>	Sets up the configuration of the BLE module in FOX3-3G-BLE devices. Please do not set it "off".
CNF.Set,Ble.AdvName=<Advertized_friendly_name>	Defines the BLE device name. Default = FOX3-3G-AVL e.g. $PFAL,CNF.Set,Ble.AdvName=FOX3-BLE
BLE.WHITELIST=<None|Public|Name|MAC>	Sets the whitelist mode after scanning iBeacons. **None**: Disables checking the attributes of iBeacon sensors to be inserted into the whitelist **Public:** Lists only iBeacon sensors advertising their name(s), after scan. **Name:** Lists all iBeacon sensors that advertise and don't advertise their names, after scan. iBeacon Sensors that do not advertise their names are shown as "unnamed". **MAC:** Lists only iBeacon sensors advertising their MAC, after scan.
BLE.SCANDURATION=<ScanDuration>	This configuration command defines how long the FOX3-3G-BLE is searching for discoverable iBeacon devices. <ScanDuration> ranges from 5 to 60 seconds.
Events Syntax	**Description**
SYS.BLE.eConnected	Occurs when the connection to a host is established (e.g. tablet or smart-phone)
SYS.BLE.eDisconnected	Occurs when the connection to a host is lost (e.g. App on the tablet)
SYS.BLE.eRegister="<BLE-BeaconID/Name>"	Occurs when a BLE beacon is inside the range of the FOX3-3G-BLE.
SYS.BLE.eRegister=whitelist	Occurs when the incoming ID or name matches one of the entries available in the Whitelist. Entries in the Whitelist can be set with *$PFAL,Sys.Whitelist.Set*
SYS.BLE.eRelease="<BLE-BeaconID/Name>"	Occurs when an already registered BLE beacon is out of range of the FOX3-3G-BLE.
SYS.eBleData="<userTxt>"	Occurs when the device receives a user text from the connected host (e.g. tablet or smart-phone)
SYS.BLE.sConnected	True, as long as the BLE connection state of the FOX3-3G-BLE (slave) to a BLE master (Smartphone/Tablet/PC) has changed from disconnected to connected.

SYS.BLE.sDisconnected	True, as long as the BLE connection state of the FOX3-3G-BLE (slave) to a BLE master (Smartphone/Tablet/PC) has changed from connected to disconnected. It may happen that after the master device has disconnected an established connection, the connection of FOX3-3G-BLE (slave) to a master is lost.
Commands syntax	**Description**
Sys.Ble.Scan	Starts scanning for new BLE devices in your near environment. The FOX3-3G-BLE scans for 12 seconds long for around iBeacons. This value of 12 seconds is predefined in the firmware and cann't be customized by the users.
Sys.Ble.List	Shows all available BLE devices listed by names. Depending on the length of the advertised friendly names of the BLE iBeacons, the device can store up to 28 names, if the name for each iBeacon has a length of max. 24 bytes.
SYS.BLE.ClearList	Clears the list of BLE sensors saved in the dynamic variables &(BLE.List) and &(BLE.List2) during the last device scan.
SYS.BLE.Select,<index>	Selects the user specified index from the list saved in the dynamic variable &(BLE.List) and &(BLE.List2) during the last device scan.
SYS.BLE.Show,<index>,<"text">	Reports the received attributes of iBeacon which is in the selected list. <"text"> can include attributes of the selected iBeacon using dynamic entries.
Sys.Ble.Send,<protocol>,"<text+dynamic entries>"	Sends the specified protocols and/or user text to the BLE connected devices.
Msg.Send.Ble,<protocol>,"<text+dynamic entries>"	Sends the specified protocols and/or user text to the BLE connected devices
Dynamic Entries Syntax	**Entries Description**
&(BLE.Name)	Used to report the name of last registered BLE device.
&(BLE.RSSI)	Used to report the RSSI value in dbm from last registered BLE device.
&(BLE.Released)	Used to report the name of last released BLE device.
&(BLE.List)	Used to report the devices found at last scan.
&(BLE.List2)	Reports list of beacon MACs which were successfully scanned.
&(BLE.MAC)	Reports the MAC address of the registered iBeacon.
&(BLE.UUID)	Reports the universally unique identifier (UUID) of the registered iBeacon. This is a 16-byte value that identifies the manufacturing organization. For example, all iBeacons from the same organization will have the same UUID. This field is a mandatory part in the advertisement packet.
&(BLE.Major)	Reports the Major value of the registered iBeacon sensor. This is an optional field in the advertisement packet of iBeacons and not mandatory to advertise every time and some iBeacons do not advertise it at all. This is a 2 byte field that specifies a certain group under an UUID. If the beacons from same UUID are deployed in different cities then each city might a have unique major ID. It is set to 0 if not available.
&(BLE.Minor)	Reports the Minor value of the registered iBeacon sensor. This 2-byte field identifies a particular beacon and is unique for each beacon. This is also an optional field and some iBeacons do not advertise it at all. When the FOX3-3G-BLE detects iBeacons, the above mentioned IDs will inform the master which beacons are in the vicinity. The RSSI values give additional information to localize mobile assets using these identification parameters and RSSI. It is set to 0 if not available.
&(BLE.relName)	Reports the name of the last released beacon.
&(BLE.relMAC)	Reports the MAC address of the last released beacon.
&(BLE.relUUID)	Reports the UUID of the last released beacon.
&(BLE.relMajor)	Reports the major value of the last released beacon.
&(BLE.relMinor)	Reports the minor value of the last released beacon.

1.4.4.2   Setup your FOX-3G-BLE device as a slave to send specific data to a master/server

Below you will find an example, how to setup and establish the Bluetooth® connection between FOX3-3G-BLE and the host periphery (example App on the tablet with Android 4.4 and higher) and how they communicate between each other.

The demo application "**app-debug.apk**" for Android 4.4 and higher called "**BLE Terminal Demo**" can communicate with the FOX3-3G-BLE over Bluetooth® Low Energy.
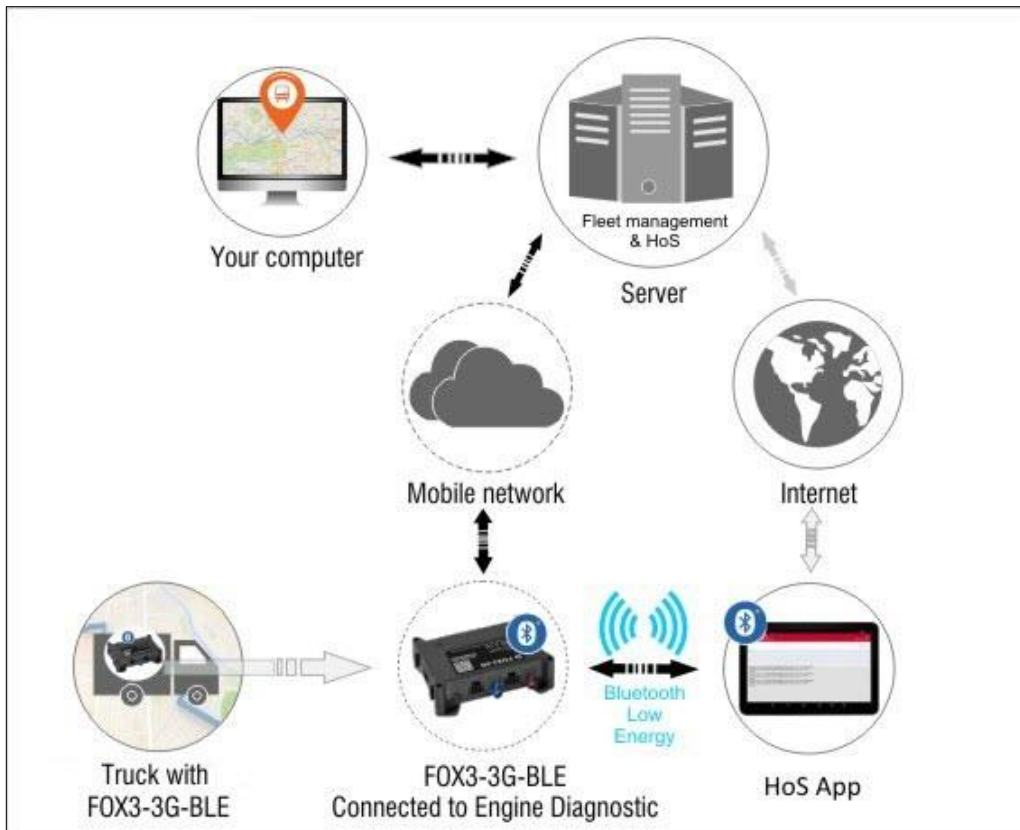

Figure 1:   HoS solutions when using FOX3-3G-BLE as a Slave combined with an Android app

The file "**app-debug.apk**" can be sent to the customers upon request.

1.   Turn on the FOX3-3G-BLE and configure it with such alarm configuration lines.

$PFAL,CNF.Set,**AL1**=SYS.BLE.eConnected:Sys.Timer0.start=cyclic,5000
AL1 starts a cyclic timer with a timeout of 5000 ms, once the device is connected to the BLE server (tablet).
$PFAL,CNF.Set,**AL2**=Sys.Timer.e0:Msg.Send.Ble,8,"Name:&(Devicename); IMEI:&(IMEI); Date: &(Date); Time:&(Time)"
AL2 sends the name, IMEI, date and time of the device to the BLE server (tablet) every 5 seconds (on timer event) during an existing connection.
$PFAL,CNF.Set,**AL3**=SYS.BLE.eDisconnected:Sys.Timer0.stop
AL3 stops timer (occurrence of timer event) after disconnecting from the BLE server (tablet).
$PFAL,Sys.BLE.Enable
This command activates the BLE function inside the device.

2.   Turn on your tablet (or smart phone) and install the application "**BLE Terminal Demo**" (see **app-debug.apk.** After a successful installation, its icon appears on the display (see Figure below).

Figure 2: icon

3. Activate the Bluetooth® on the tablet.

4. Start the "**BLE Terminal Demo**" App by hitting the icon.

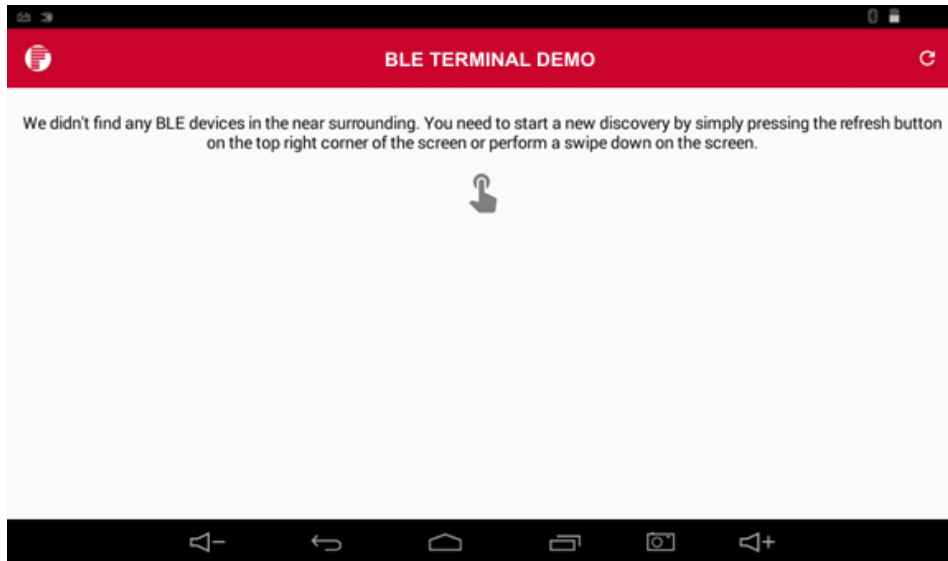5. When the "BLE Terminal Demo" starts up, you will see this window (see Figure below).

Figure 3: Main user interface after starting the "BLE Terminal Demo"

6. Click on the "**scan**" symbol to start scanning the BLE devices near the tablet

7. The App should find "FOX3-3G-AVL". Please note that the maximum length for the FOX3-3G-BLE advertised name is 10 characters.
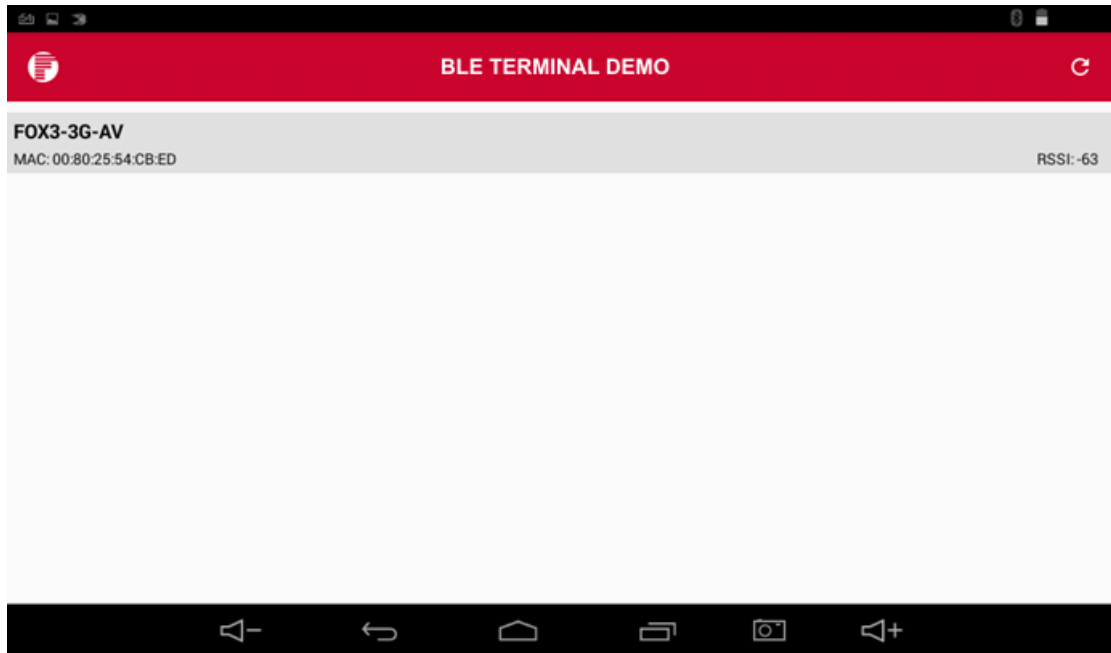
8. Click on the name "FOX3-3G-AVL" (see Figure below)

Figure 4: Result of the found BLE devices near the tablet
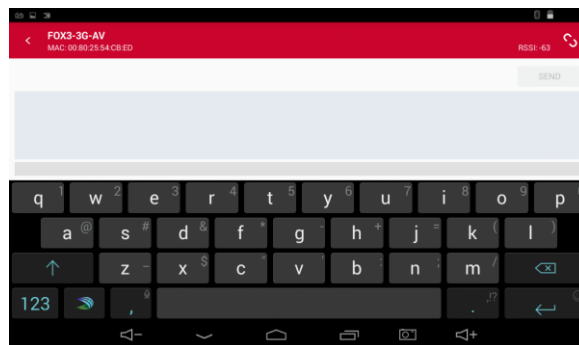
9. A terminal starts (see Figure below)

Figure 5: Terminal interface on the "BLE Spp Demo"

10. Click on the "connect/disconnect" symbol [icon] to establish the BLE connection.

11. Once the BLE connection between the FOX3-3G-BLE and the tablet is established, the FOX3-3G-BLE generates the event: **SYS.BLE.eConnected** and starts sending its data periodically (every 5 seconds), (see Figure below).
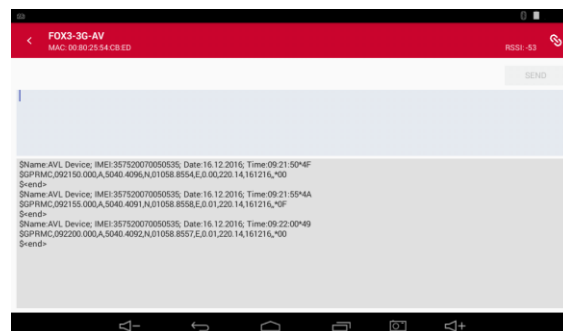

Figure 6: The tablet receives data from the FOX3-3G-BLE

12. To send data e.g. GPRMC protocol and user text to the connected host device, use the command *$PFAL,Sys.Ble.Send,08,"Hello host device"*.

13. While receiving data from the FOX3-3G-BLE you can send texts or PFAL commands from the tablet to the FOX3-3G-BLE. Type **Hello FOX3** in the upper field then hit "**SEND**" (see Figure below)
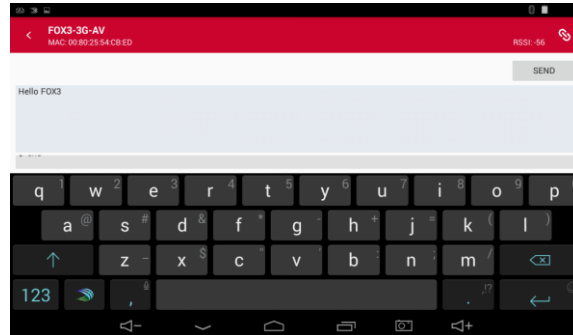

Figure 7: Sending data from the tablet to the FOX3-3G-BLE

14. The FOX3-3G-BLE receives this message and generates the EVENT: **SYS.eBleData="Hello FOX3"**

1.4.4.3 Setup your FOX3-BLE device as a master to scan for BLE iBeacons

For the applications which require BLE beacons detections, you can automate the BLE scan procedure by using the action/command (Sys.Ble.Scan) in PFAL alarm lines. iBeacons can be used in application such as container identification, trailer identification, pallet tracking and more. iBeacons are attached to your pallets, containers, trailers and the FOX3-3G-BLE devices are wired permanently onto trucks.
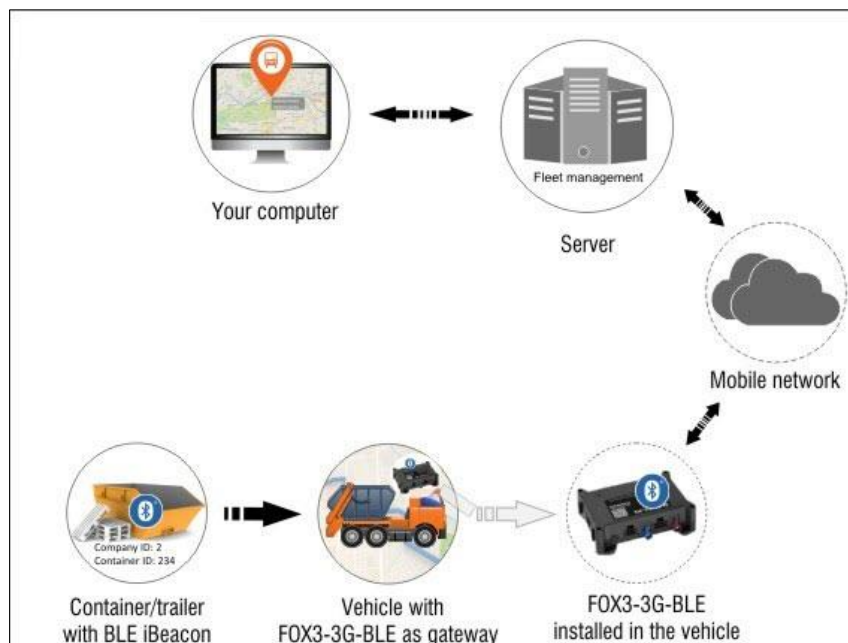

Figure 8:   IoT solutions when using FOX3-3G-BLE as a Master combined with iBeacons

The FOX3-3G-BLE scans and reads the NAME, UUID, RSSI, MAC, MAJOR and MINOR from each iBeacon located within range without pairing with the iBeacon. These values are stored in a list in the FOX3-3G-BLE, which are then removed from the FOX3-3G-BLE if iBeacons are not within its range to continuously update the current status of objects. The device transmits the stored data in

the dynamic variables &(NAME), &(UUID), &(RSSI), &(MAC), &(MAJOR) and &(MINOR) over mobile network at predetermined times when a new iBeacons is registered or removed to a remote server. The remote server stores the transmitted list of iBeacons for analysis and report generations.

The table below shows how to configure your FOX3-3G-BLE as a gateway for identifying fixed objects or tracking mobile objects during the transport. If an iBeacon does not advertise its name, the FOX3-3G-BLE will output "unnamed" instead of an empty name as shown in the example below.

$PFAL,Cnf.Set,**AL1**=Sys.Device.eStart:sys.Timer1.Start=cyclic,20000
AL1 starts a cyclic timer with a time out of 20 sec on device start up to scan for iBeacons. With this cyclic timer, you can set how often the FOX3-3G-BLE device scans for iBeacons.
$PFAL,Cnf.Set,**AL2**=Sys.Timer.e1:Sys.Ble.Scan
AL2 starts scanning for iBeacons on timer event. In our example, the device scans every 20 seconds. The FOX3-3G-BLE scans for 12 seconds long for around iBeacons. This value of 12 seconds is predefined in the firmware and cann't be customized by the users.
$PFAL,Cnf.set,**AL3**=Sys.Ble.eRegister:TCP.Client.Send,8,"BLE REGISTER MAC: &(Ble.MAC); UUID: &(BLE.UUID); MAJOR: &(Ble.Major); MINOR: &(Ble.Minor)"
AL3: On new iBeacon registration the device sends the MAC, UUID, Major and Minor of the iBeacons and the RMC protocol of the device to a connected remote server.

Data transferred to the server	$BLE REGISTER MAC: 7C:2F:80:AA:63:82; UUID: 80010215123480AA638CBBC5; MAJOR: 1; MINOR: 10*51 $GPRMC,083157.000,A,5040.4185,N,01058.8433,E,2.42,169.88,250618,,*00
	$BLE REGISTER MAC: 7A:2F:80:A0:63:00; UUID: 4C000102030405060708090A0B0C0D0E0F10; MAJOR: 10; MINOR: 267*65 $GPRMC,083157.000,A,5040.4185,N,01058.8433,E,2.42,169.88,250618,,*00

$PFAL,Cnf.set,**AL4**=Sys.Ble.eRelease:TCP.Client.Send,8, "BLE RELEASE MAC: &(Ble.relMAC); UUID: &(BLE.relUUID); MAJOR: &(Ble.relMajor); MINOR: &(Ble.relMinor)"
AL4 s On iBeacon release the device sends the RMC protocol and the MAC, UUID, Major and Minor of the iBeacons of the last released iBeacons to a connected remote server.

Data transferred to the server	$BLE RELEASE MAC: 7C:2F:80:AA:63:82; UUID: 80010215123480AA638CBBC5; MAJOR: 1; MINOR: 10*51 $GPRMC,083157.000,A,5040.4185,N,01058.8433,E,2.42,169.88,250618,,*00
	$BLE RELEASE MAC: 7A:2F:80:A0:63:00; UUID: 4C000102030405060708090A0B0C0D0E0F10; MAJOR: 10; MINOR: 267*51 $GPRMC,083157.000,A,5040.4185,N,01058.8433,E,2.42,169.88,250618,,*00

$PFAL,Sys.BLE.Enable
This command activates the BLE function inside the device.

NOTES

➢ **Please be sure that the cyclic scan for the BLE beacons is not less than 20 sec.**

➢ **The FOX3-3G-BLE can't be operated in two modes, i.e. BLE connection with a host and scanning BLE beacons simultaneously.**

The iBeacon sensors from ELA are designed to identify and track industrial equipment. They support UUID, RSSI, Major and Minor values and have been tested. Here are some of the iBeacon specifications at a glance:

✓ High receiving range: 500 meters (open field)
✓ Lifecycle: up to 20 years
✓ Robust industrial casing - IP68 waterproof

- ✓ Integrated NFC chipset for programming *(You must change their format from **ID/temp** into **iBeacon**. The **id** and **temp** formats are currently not supported in the FOX3-3G-BLE)*
- ✓ Operating temperature: -40°C to +85 °C
- ✓ 2 mounting ∅ 4mm holes
- ✓ ∅57mm - H: 18mm - W: 36g
- ✓ RoHS compliant