

# UART HCI Bluetooth Module for Linux

## BT830

Application Note

v1.2

### INTRODUCTION

Laird's BT830 is a UART HCI Bluetooth module. As there is no EEPROM on board, it is required to load module-specific settings such as MAC address and Crystal trim at the module initialization stage. This application note describes how to use the BlueZ BCCMD tool to *patch* these settings on the Linux platform. BlueZ and BlueZ-Utils packages are required for this operation.

### REQUIREMENTS

- BT830 development board
- BlueZ – Official Linux Bluetooth protocol stack
- BT830-specific configuration files

**Notes:** BT830 development board uses the FTDI USB-UART chip. The testing platform used in this application note has the driver installed automatically. The name of the serial port is `/dev/ttyUSB0`.

Fedora 20 is used as the testing platform. The BlueZ stack is included.

Laird provides a configuration file (BT830.Linux.psr). You can modify this file to select baud rate, Bluetooth MAC address, and so on.

### PATCHING WITH BCCMD

#### Preparation

Before plugging the BT830 development board to the computer, it is important to check if there is any existing Bluetooth radio by typing "hciconfig". If one is found, please close it by entering "hciconfig hciX down". Usually it is hci0 as the first Bluetooth device on the computer. SU login right should be required.

```
[test@localhost ~]$ su
Password:
[root@localhost test]# hciconfig
hci0:  Type: BR/EDR  Bus: USB
       BD Address: 00:80:98:0C:62:99  ACL MTU: 384:8  SCO MTU: 64:8
       UP RUNNING PSCAN
       RX bytes:489 acl:0 sco:0 events:21 errors:0
       TX bytes:337 acl:0 sco:0 commands:21 errors:0

[root@localhost test]# hciconfig hci0 down
[root@localhost test]# hciconfig
hci0:  Type: BR/EDR  Bus: USB
       BD Address: 00:80:98:0C:62:99  ACL MTU: 384:8  SCO MTU: 64:8
       DOWN
       RX bytes:489 acl:0 sco:0 events:21 errors:0
       TX bytes:337 acl:0 sco:0 commands:21 errors:0

[root@localhost test]# █
```

Figure 1: Disable existing computer existing Bluetooth device

After plugging the BT830 development board to the computer, please locate the USB UART port by typing “dmesg | grep FTDI”. The development board uses the FTDI USB-UART chip.

```
[root@localhost test]# dmesg | grep FTDI
[ 2304.441156] usb 2-2: Manufacturer: FTDI
[ 2304.528099] usbserial: USB Serial support registered for FTDI USB Serial Device
[ 2304.528244] ftdi_sio 2-2:1.0: FTDI USB Serial Device converter detected
[ 2304.531635] usb 2-2: FTDI USB Serial Device converter now attached to ttyUSB0
[root@localhost test]# █
```

Figure 2: Locate the FTDI USB-UART port

## Modifying the Configuration File

Laird has provided a configuration file (BT830.Linux.psr) which includes all required settings. You can modify this file to change the UART configurations, BT/WIFI coexistence scheme, Bluetooth MAC address, and so on. The default baud rate in the configure file is 921600 bps.

```
&01ea = 0001 c200
&0001 = 0001 0000 00A4 0016
&01fe = 6590
&01f6 = 000c
```

Figure 3: &01EA is the pskey for baud rate. It is changeable. 0001 C2000 represents 115200 bps

```
&01ea = 000E 1000
&0001 = 0001 0000 00A4 0016
&01fe = 6590
&01f6 = 000c
```

Figure 4: 921600 bps UART baud rate (Default). 0000E 1000 represents 921600 bps.

```
&01ea = 0001 c200
&0001 = 0001 0000 00A4 0016
&01fe = 6590
&01f6 = 000c
```

Figure 5: Changeable BT MAC address - &0001 is Bluetooth MAC scheme

```
&01fe = 6590
&01f6 = 000c
&2480 = 0000
&2483 = 0001 0001
&2484 = 0009 0001
&2485 = 0004 0001
&0017 = 0008|
&0021 = 0008
```

Figure 6: Coexistence scheme - &2480, &2483, &2484, &2485

### Loading the Configuration File

BlueZ's BCCMD is used to load the configuration. New settings are effective once the command is executed. The command syntax is `bccmd -t BCSP -d /dev/ttyUSB0 psload --reset BT830_Linux.psr`. The `hciattach` command attaches the UART HCI module and the syntax is `hciattach /dev/ttyUSB0 bcsp 921600`. The baud rate value (921600) must be included, as this is the value inside the configuration file.

```
File Edit View Search Terminal Help
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# rpm -qi bluez
Name       : bluez
Version    : 5.12
Release    : 1.fc20
Architecture: x86_64
Install Date: Wed 11 Dec 2013 07:11:22 PM EST
Group      : Applications/System
Size       : 3447624
License    : GPLv2+
Signature  : RSA/SHA256, Wed 11 Dec 2013 09:36:05 AM EST, Key ID 2eb161fa246110
c1
Source RPM : bluez-5.12-1.fc20.src.rpm
Build Date : Wed 11 Dec 2013 06:50:17 AM EST
Build Host : buildvm-19.phx2.fedoraproject.org
Relocations: (not relocatable)
Packager   : Fedora Project
Vendor     : Fedora Project
URL        : http://www.bluez.org/
Summary    : Bluetooth utilities
Description:
Utilities for use in Bluetooth applications:
- hcitool
- hciattach
- hciconfig
- bluetoothd
- l2ping
- rfcomm
- sdptool
- bccmd
- bluetoothctl
- btmon
- hcidump
- l2test
- rctest
- start scripts (Red Hat)
- pcmcia configuration files

The BLUETOOTH trademarks are owned by Bluetooth SIG, Inc., U.S.A.
[root@localhost ~]#
```

Figure 7: `rpm -qi bluez` determines whether or not BlueZ is installed

```
[root@localhost Laird]# bccmd -t BCSP -d /dev/ttyUSB0 psload --reset BT830_Linux.psr
Loading 0x212c ... done
Loading 0x212d ... done
```

Figure 8: `bccmd -t BCSP -d /dev/ttyUSB0 psload --reset BT830_linux.psr` loads the psr file

It is possible to encounter the “Operation timed out” error and a few second delay, as the module goes through a reset cycle. This error can be ignored, and you can continue to the “hciattach”. It is recommended to do the initialization on system boot, thus the delay is not noticeable.

```
Loading 0x22c8 ... done
Loading 0x216d ... done
Loading PSKEY_CLOCK_REQUEST_ENABLE ... done
Operation timed out
[root@localhost Documents]#
```

Figure 9: Operation timed out error as the module goes through a reset cycle

```
[root@localhost Laird]# hciattach /dev/ttyUSB0 bccsp 921600
Device setup complete
```

Figure 10: `hciattach /dev/ttyUSB0 bccsp 921600` attaches the BT830

## Launching the Bluetooth Stack with New Settings

To confirm BT830 has been successfully attached, type “hciconfig” to see all recognized Bluetooth radios . On this computer, there are 2 Bluetooth radios. BT830 has been attached as “hci1”. To enable BT830, please type “hciconfig hci1 up”.

```
[root@localhost Documents]# hciconfig
hci1: Type: BR/EDR Bus: UART
      BD Address: 00:16:A4:01:00:00 ACL MTU: 310:10 SCO MTU: 64:8
      DOWN
      RX bytes:1025 acl:0 sco:0 events:28 errors:0
      TX bytes:836 acl:0 sco:0 commands:28 errors:0

hci0: Type: BR/EDR Bus: USB
      BD Address: 00:80:98:0C:62:99 ACL MTU: 384:8 SCO MTU: 64:8
      DOWN
      RX bytes:489 acl:0 sco:0 events:21 errors:0
      TX bytes:337 acl:0 sco:0 commands:21 errors:0

[root@localhost Documents]# hciconfig hci1 up
[root@localhost Documents]# hciconfig
hci1: Type: BR/EDR Bus: UART
      BD Address: 00:16:A4:01:00:00 ACL MTU: 310:10 SCO MTU: 64:8
      UP RUNNING PSCAN
      RX bytes:2101 acl:0 sco:0 events:59 errors:0
      TX bytes:1994 acl:0 sco:0 commands:60 errors:0

hci0: Type: BR/EDR Bus: USB
      BD Address: 00:80:98:0C:62:99 ACL MTU: 384:8 SCO MTU: 64:8
      DOWN
      RX bytes:489 acl:0 sco:0 events:21 errors:0
      TX bytes:337 acl:0 sco:0 commands:21 errors:0

[root@localhost Documents]#
```

Figure 11: BT830 is recognized and enabled with the MAC address from the configuration file

## CONNECT THE BT830 VIA COMMAND LINE

### Verify BT830 Connection

In Linux, you may configure and test the BT830 via terminal. The Linux utility to configure and identify Bluetooth is `hcitool`. To verify that the BT830 is recognized by the operating system, run `hcitool` and check for devices as follows:

1. Open the command terminal.
2. Enter the following command:

```
hcitool dev
```

This command displays local devices. If it finds one, it returns the following:

```
[root@localhost Laird]# hcitool dev
Devices:
    hci1    00:16:A4:01:00:00
```

This MAC address should correspond to the MAC address of the BT830 module.

**Note:** The `hcitool` command uses the first available Bluetooth device for its operations. If multiple Bluetooth devices are found, all `hcitool` commands must specify which device to use, as follows:

```
hcitool [-i <hciX>] [command [command parameters]]
```

In this example, `<hciX>` must correspond to the HCI device number found using `hcitool dev`.

### Scan with Classic Bluetooth

With the device initialized, you may test Bluetooth functionality from the command prompt. To test scanning, you must have a nearby device (such as a tablet or smartphone) set to be discoverable.

The command to initialize a scan is:

```
hcitool scan
```

When a scan is initialized, the terminal returns found devices in the following format:

```
Scanning ...
    [MAC Address]    Friendly_Name
```

If there are discoverable devices nearby, they appear in this list as they are discovered.

```
[root@localhost Laird]# hcitool scan
Scanning ...
    58:A2:B5:5F:B6:C6    My_G2
[root@localhost Laird]#
```

Figure 9: Discoverable devices

## Scan with Bluetooth Low Energy

The hcitool commands to scan Bluetooth Low Energy are distinct from those used in classic Bluetooth connections. To initiate a BLE scan from the terminal, issue the following command:

```
#hcitool lscan
```

The terminal returns the following:

```
LE Scan ...  
[MAC Address] - [BLE device]
```

```
[root@localhost Laird]# hcitool lscan  
LE Scan ...  
FC:11:11:15:02:85 LT_UPASS
```

Figure 10: BLE MAC address

## REVISION HISTORY

Version	Date	Notes	Approver
1.0	23 July 2014	Initial Release	Jonathan Kaye
1.1	21 Apr 2015	Reviewed updates for BT coexistence setting	Jordan Manser
1.2	01 Apr 2016	Rearrange the loading procedure	Jonathan Kaye