User's Guide

BluetopiaPM Stack Build Guide for Linux

ABSTRACT

TI provides Bluetopia Platform Manager (BluetopiaPM) *Bluetooth*® stack for Linux hosts to work with TI WL18xx and CC2564X Dual Mode BT controllers. This document is intended to serve as a user guide for integrating BluetopiaPM stack release into Arm® Linux®-based platform. This document details stack architecture, core components, configuration files, build procedure and testing of Bluetopia stack release. The document also serves as a user's guide for verifying the basic Bluetooth functionalities and provides brief overview for debugging and FAQs. The examples provided are using the PROCESSOR-SDK-LINUX-AM335X 06_00_00_07. But the same methodology is applicable to other SDKs as well that are based on Arm Cortex®-A architectures.

Table of Contents

Trademarks

Sitara™ is a trademark of Texas Instruments.
Bluetooth® is a registered trademark of Bluetooth SIG, Inc.
Cortex® is a registered trademark of Arm Limited.
All trademarks are the property of their respective owners.

Copyright © 2021 Texas Instruments Incorporated

1 Introduction

This document explains how to get started with TI Bluetooth Protocol Stack on a target device running ARM Linux. It provides a basic description of the package, and walks through the download and installation procedures. Finally, it shows how to run a sample application BluetopiaPM Demo applications and the BluetopiaPM installation comes with pre-built binaries with AM335x. However, for users who wish to modify sample applications or the build environment, visit the 'Build BluetopiaPM' section below for details.

2 Bluetopia

Bluetopia is a low-level Bluetooth Protocol Stack. Bluetopia is designed to be small in code space and RAM so that it can be ran on small embedded platforms where memory is limited. At any given point in time only one application has access to the Bluetopia stack. This can make development difficult and complicated if multiple profiles are using the Bluetooth Controller concurrently.

3 Bluetopia Platform Manager (BluetopiaPM)

Bluetopia Platform Manager (BluetopiaPM) is a service that allows multiple applications to access the Bluetooth link concurrently. This allows developers to create multiple independent applications that each use the Bluetooth link. Take for example a Head Set running Linux that uses the Hands-Free Profile and also supports Advanced Audio Distribution Profile (A2DP) Sink audio playback. Using Bluetopia a development team would have to write one application that handles both Hands-Free Events and A2DP Events. With BluetopiaPM the team could write two independent applications: one to handle Hands-Free Events and one to handle A2DP Events.

BluetopiaPM also provides features that simplify the management of device connections:

- It stores pairing and link key information for each device connection in persistent memory.
- It automatically caches discovered device information such as the device name and class of the device.
- It automatically caches remote device service discovery information for quick access to a device's services without having to retrieve them again from the remote device.

A convenient API is provided to access cached device information.

TI Bluetooth Stack (based on Bluetopia) is provided for WL18xx and CC256x devices. It is composed of Bluetooth protocol stack, Bluetooth profiles stack and Platform Manager.

4 Hardware Requirements

- 1x AM335x/AM43x/AM57x Evaluation Module or AM335x Beagle Bone
- For Wilink8: WL1837MODCOM8I or WL1835MODCOM8B
- For CC256x: CC2564MODNEM or CC2564MODNEM or CC2564C QFN
- 1x RS-232 to USB adapter cable to access CLI interface on the platform from PC
- Bluetooth remote devices (such as a phone or another EVM) that should support the above profiles, preferably with the opposite role for some cases, for demonstration purposes.

Note

When using a CC256x controller you need to perform the following Hardware Modifications in order to make it compatible to the Sitara™ platforms. For more information, see Appendix A.

5 Software Requirements

- AM335x Linux SDK or image for flashing onto an SD card.
- Wilink 8 Bluetooth Add-On
- TI EVM (AM335x, AM437x, AM57x) Bluetooth setup script can be downloaded from the following Wilink 8 Bluetooth Add-On
- PC Terminal software (for example, Putty or TeraTerm).
- Most recently updated BTS script from WL18xx-BT-SP (see note below).

Copyright © 2021 Texas Instruments Incorporated

Note

Download the latest WL18xx Bluetooth Firmware Script (BTS script) from: WL18xx-BT-SP. Read the license agreement and follow the installation instructions in the previous link. The script will be automatically be downloaded to the WL18xx device whenever the stack power up the device.

6 Stack Supported Features

Following is the major list of features supported by BluetopiaPM stack.

- Dual mode Classic Bluetooth (BR/EDR) and Bluetooth low energy (BLE) Stack
- Bluetooth Core Specification v5.1
- Platform Manager framework to enable multiple client applications to use the stack
- Debug option to capture BT snoop logs for Debug/Analysis.
- Support for Multiple BT profiles:
  - Advanced Audio Distribution Profile (A2DP)
  - Audio/Video Remote Control Profile (AVRCP)
  - Generic Access Profile (GAP)
  - Generic Audio/Video Distribution Profile (GAVDP)
  - Headset Profile (HSP)
  - Hands Free Profile (HFP)
  - Human Interface Device Profile (HID)
  - Message Access Profile (MAP)
  - Phonebook Access Profile (PBAP)
  - Serial Port Profile (SPP)
  - Generic Access Profile Services (GAPS)
  - Generic Attribute Profile (GATT)

7 TI BluetopiaPM Stack Linux Architecture

BluetopiaPM stack is user space based and uses standard Linux tty driver interface to communicate with the DualMode BT controllers on a UART interface. The stack supports H4 (4wire) HCI protocol with hardware flow control. Figure 7-1 shows the high-level driver partitioning and architecture.

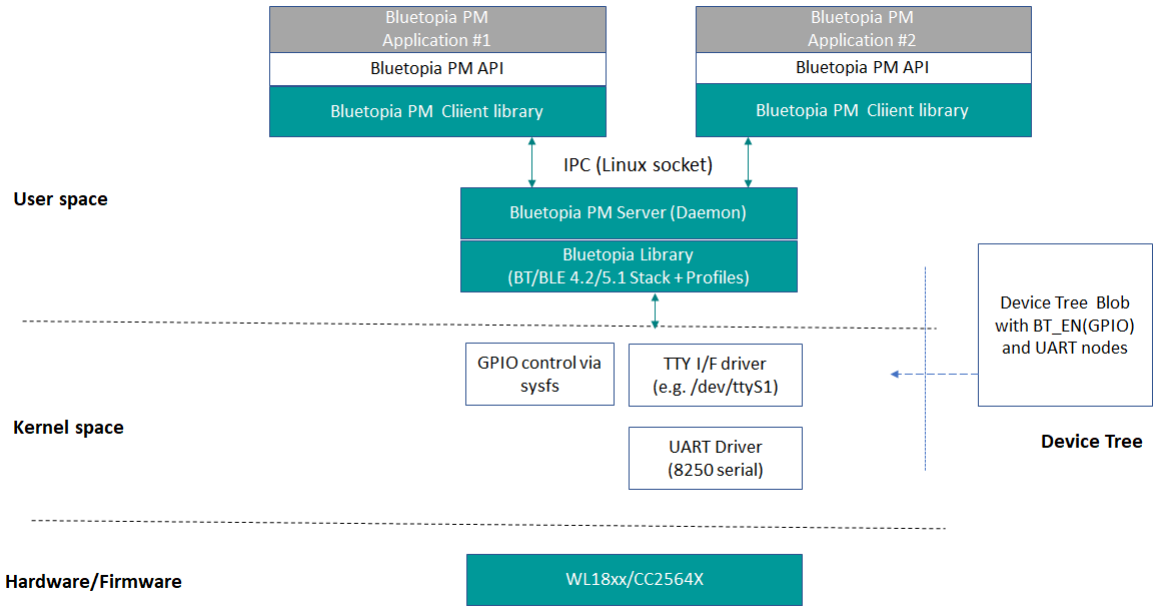


Figure 7-1. BluetopiaPM Stack Architecture

Copyright © 2021 Texas Instruments Incorporated

BluetopiaPM and Bluez stacks are mutually exclusive. This means, when using BluetopiaPM stack, one should not use BlueZ stack and ensure that the BlueZ driver in the kernel is not initialized. The ‘bluetooth’ property for the HCI universal asynchronous receiver/transmitter (UART) node in the device tree should not be enabled for using BluetopiaPM stack. For example, ‘bluetooth’ property in below uart3 node should be removed for using BluetopiaPM stack.

```
&uart3 {
    pinctrl-names = "default";
    pinctrl-0 = <&uart3_pins &bt_pins>;
    status = "okay";

    bluetooth {
        compatible = "ti,wl1835-st";
        enable-gpios = <&gpio0 28 GPIO_ACTIVE_HIGH>;
    };
};
```

The section below briefs the high-level components in the Stack and their functionality.

- BluetopiaPM/Bluetopia:
  - Dual mode Classic Bluetooth (BR/EDR) and Bluetooth Low Energy (BLE) protocol stack and profiles.
  - Supports Bluetooth Core Specification v5.1.
  - Supports WL18xx and CC256X devices.
  - BluetopiaPM (Platform Manager) is a client-server add-on that enables multiple applications to simultaneously communicate with the Bluetopia stack.
  - Each application will be linked to BluetopiaPM Client library that handles the communications to a single BluetopiaPM server daemon that hosts the Bluetopia stack.

For more details on the BluetopiaPM stack, see https://www.ti.com/tool/tibluetoothstack-sdk

8 Installing Bluetopia Platform Manager

TI Bluetooth Stack (based on Bluetopia) is provided with the WL18xx or CC256x devices. It is composed of Bluetooth protocol stack, Bluetooth profiles stack and a Platform Manager. The Bluetooth Add-On Packages can be re-built and used with any ARM Linux for WL18xx/ CC256x platforms. But for convenience ready to use pre-built binaries built with TISDK processor tool chain are released with the BluetopiaPM stack. The stack is not part of TI AM335x/AM47x/AM57x platform SDK but is available as an add-on that can be downloaded from the Software Requirements section.

The package is downloaded as a tar ball and should be extracted (in a Linux machine).

- https://www.ti.com/tool/TI-BT-4-2-STACK-LINUX-ADDON

The tar ball includes two Linux installers: for 32bit and 64bit release. Please use the right one according to your Host Development platform type. The following demonstrates the 64bit case:

```
$ chmod +x ./AM335xBluetopiaLinuxProduction-4.2.1.1.1.0-linux-x64-installer.run
$ ./AM335xBluetopiaLinuxProduction-4.2.1.1.1.0-linux-x64-installer.run
```

When the installer completes, the installation directory should include Bluetopia Platform Manager (BluetopiaPM) Binaries, Sources and Documentation.

- Binaries - includes pre-compiled sample applications, BluetopiaPM server daemon and BluetopiaPM Client libraries.
- Sources - platform specific files (to enable recompiling for different platforms) and sample applications code.
- Documentation - License docs, users guides and API specifications.
- The installation directory can be updated during the installer execution. The default is

```
~/AM335xBluetopiaLinuxProduction-4.2.1.1.1.0/
```


Copyright © 2021 Texas Instruments Incorporated

9 BluetopiaPM Architecture

9.1 Installation Directory Structure

[install-dir]:	1. The BluetopiaPM Distribution directory 2. The Build directory 3. The Documentation directory
[install-dir]/BluetopiaPM:	1. The BluetopiaPM Distribution source code 2. The BluetopiaPM Distribution libraries A Bluetopia distribution is also included inside of this directory because Bluetopia is required to build and re-configure BluetopiaPM.
[install-dir]/build:	• Makefiles that can be used to compile the environment and build the sample applications.
[install-dir]/Documentation:	• Documentation relevant to the release.

9.2 BluetopiaPM Structure

The directory structure of **[install-dir]/BluetopiaPM** is as follows:

bin:	The BluetopiaPM bin directory contains pre-compiled executables for the PM Server and for Sample Client Applications that demonstrate using the PM Modules.
btpmerr:	The BluetopiaPM Error directory contains includes and libraries used to convert BluetopiaPM Error Codes to User-Friendly Text Strings.
btpmipc:	The btpmipc directory contains includes source and header files for BluetopiaPM for the BluetopiaPM Inteprocess communication abstraction layer.
btpmmodc:	The BluetopiaPM Module Configuration directory contains code and makefiles that can be used to customize the PM Server Executable and to customize which modules are initialized.
btpmvs:	The BluetopiaPM btpmvs directory contains Vendor specific functions/definitions/constants used to define a set of vendor specific functions, supported for CC256x and WL18x.
include:	The BluetopiaPM Include directory contains header files that should be included by an application.
lib:	The BluetopiaPM Library directory contains the BluetopiaPM Client Library. Client applications can link to this library.
modules:	The BluetopiaPM Modules directory contains includes and libraries that can be use when re-building the PM Server or PM Client Applications.
sample:	The BluetopiaPM Samples directory contains source code. These are samples are provided as a reference to demonstrate how to use the modules and get started with development in BluetopiaPM.
ss1btpm/server:	The Stonestreet One BluetopiaPM Server directory contains make files that can be used to customize the PM Server library. The PM Server Executable links to the server binary when the PM Server is re-built. Note that source code is provided for the PM Server so that it can be customized if necessary.

Copyright © 2021 Texas Instruments Incorporated

10 The BluetopiaPM Server – Pairing Mode

SS1BTPM is compiled to support secure simple pairing as the default configuration. If legacy pairing is needed instead, for example so that the user is forced to enter a fixed pin code, then it is possible to configure and recompile SS1BTPM to only support legacy pairing. To do this open <INSTALLATION_DIR>/BluetopiaPM/ss1btpm/server/BTPMSRVR.c in a text editor. Near the bottom of the file modify the assignment:

```
DefaultInitializationData.DefaultPairabilityModeSSP = TRUE;
```

and change TRUE to FALSE.

```
DefaultInitializationData.DefaultPairabilityModeSSP = FALSE;
```

After making this change next rebuild SS1BTPM.

SS1BTPM is now compiled to support only legacy pairing. You will notice that the next time a device pairs with the target that the demo application will prompt the user for a pin code response. When this occurs respond with the **PINCodeResponse** command, for example **PINCodeResponse 0000**. The pin code used, **0000** in this example, must be the same on both devices.

Note

Keep in mind that Legacy Pairing is generally not as secure as Secure Simple Pairing. If security is a concern for your application then legacy pairing is not recommended. For more details on the security of Bluetooth pairing modes refer to the core specification that can be found on the Bluetooth SIG's website. Also note that, in the default configuration where secure simple pairing is enabled, legacy pairing is still supported. If a remote device attempts to connect to the target and the remote device only supports legacy pairing, the target will fall back to legacy pairing.

11 Platform Integration

The following section details the integration of the BluetopiaPM stack to the Linux SDK platform. The references and instructions provided are applicable to any platform using ARM Linux operating system on a Cortex-A processor/SoC.

The following build procedures are for systems with SD card boot. The image creation for other boot methods may vary. Please refer to corresponding SDK build inductions for details on image creation for other boot methods.

The generic steps to integrate the BluetopiaPM stack manually to any ARM Linux platform are provided below. These steps were tried with TI processor SDK release for AM335x/AM437x/AM57x EVMs. Note that the following steps assume Linux host environment up and running. For more information on setting up your Linux host PC, follow the instructions provided in Processor SDK Getting Started Guide.

To get started, download a ready-to-use distribution (WiLink 8 Linux Bluetooth Add-on) from:

WiLink 8 Linux Add-On Packages

Note

The same distribution can be used to build projects for CC2564X platform as well.

The packages found in the link above were cross-compiled with Linaro GCC toolchain that is included in Texas Instrument's AM335x Linux SDK or AM437x Linux SDK

If you are using a different cross-compiler or a different version of Texas Instrument's SDK then you may need to contact Texas Instrument's to determine if the packages above are binary-compatible with your development environment.

Note

If using the WL183x there are pre-built applications that are ready to run out of the box.

Copyright © 2021 Texas Instruments Incorporated

11.1 Build BluetopiaPM

Browse to the installation folder, enter build folder and follow the instructions in Build Instructions file to build sample applications in one's build environment.

1. Make sure you're in the build directory (<INSTALLATION_DIR>/build) as so. For example:

```
cd ~/AM335xBluetopiaLinuxProduction-4.2.1.1.1.0/build
```

2. Make a copy of the setup-bt.sample file. For example:

```
cp setup-bt.sample setup-bt
```

3. Update the TOOLCHAIN_PATH and KERNEL_PATH variables in setup-bt for your environment. For example:

```
# Path to local toolchain installation
 export TOOLCHAIN_PATH=~/ am335x/tisdk6.0/linux-development/linux-devkit/sysroots/

# Path to SDK compiled kernel
 export KERNEL_PATH=~/ am335x/tisdk6.0/linux-development/board-support/
linux-4.19.38+gitAUTOINC+4dae378bbe-g4dae378bbe/
```

4. Enable or disable the builds for certain projects and additional configuration settings in the 'proj/proj-defs.mak.

   For example to update the DEVICE variable in the proj/proj-defs.mak file to build for either the CC256x or Wilink8 platform (default), by commenting out the unwanted platform.

```
DEVICE = wl18xx
#DEVICE = cc256x
```

5. Update the TOOLCHAIN_PATH_NATIVE and TOOLCHAIN_PATH_TARGET variables in the **proj/toochain-defs.mak** file. For example:

```
TOOLCHAIN_PATH_NATIVE=$(TOOLCHAIN_PATH)/x86_64-arago-linux/
TOOLCHAIN_PATH_TARGET=$(TOOLCHAIN_PATH)/armv7ahf-neon-linux-gnueabi/
```

6. Run the following command to configure the environment from above settings (from the build directory).

```
source configure.sh
```

11.1.1 Modification of the PM Build Environment

When making changes in the PM environment, such as, adding files or when wanting to remove sample applications from the distribution, you will need to change the following files: In <INSTALLATION_DIR>/build/proj.

- plat-rules.mak - This file compile the sample apps and the libraries, you'll need to add/remove the section that you want.

Note

Some of the sample applications have generic build process so they are being held in variable in the platform.mak file.

```
$(PMBINDIR)/%:  $(OBJDIR)/%_C.o $(PMLIBDIR)/libBTPM_C.a
$(PMLIBDIR)/libBTPM_HAL_C.a
    @echo LN $(^F) -o $(@F)
    @$(CC)  -L$(LIBDIR) $^ $(LDFLAGS) -o $@
```

when using %, the % will be replaced with the sample apps from the variable.

Copyright © 2021 Texas Instruments Incorporated

a) example for deleting sample application:

```
$(PMBINDIR)/LinuxAUDM:  $(OBJDIR)/LinuxAUDM_C.o
$(OBJDIR)/AudioDecoderPM_C.o $(OBJDIR)/AudioEncoderPM_C.o
$(PMLIBDIR)/libBTPM_C.a $(PMLIBDIR)/libBTPM_HAL_C.a
    @echo LN $(^F) -o $(@F)
    @$(CC)  -L$(LIBDIR) $^ $(LDFLAGS) -o $@
```

removing this section will disable the compilation of the LinuxAUDM sample application.

b) example for adding sample application:

```
$(PMBINDIR)/LinuxYOURAPP:  $(OBJDIR)/LinuxYOURAPP_C.o
$(PMLIBDIR)/libBTPM_C.a   $(PMLIBDIR)/libBTPM_HAL_C.a
    @echo LN $(^F) -o $(@F)
    @$(CC)  -L$(LIBDIR) $^ $(LDFLAGS) -o $@
```

be sure to add all the libs and object that you need to compile your application.

Note

Make sure not to remove libBTPS* files.

- platform.mak - This file creates some variables that hold the sample apps in order to compile them in the plat-rules.mak file, you'll need to add/remove the sample that you want.

Build

- Run make command to build the environment in the ~/build directory

12 Adding BluetopiaPM to the SD Card

The target root can be a Network File System (NFS) to the target (root NFS) or a mounted SD card path (for example, /media/rootfs).

12.1 Sample Applications

- Copy the BluetopiaPM server (SS1BTPM) and the provided Sample applications that were built:

```
sudo mkdir [target-root]/home/root/BluetopiaPM
sudo cp -rf <INSTALLATION_DIR>/BluetopiaPM/bin
[target-root]/home/root/BluetopiaPM/
```

12.2 Gstreamer Plugin

- BluetopiaPM include a GStreamer Plugin that should be copied to the target rootfs, the GStreamer is being used in Audio (AUD) Demo application:

```
sudo cp <INSTALLATION_DIR>/BluetopiaPM/lib/libgstss1bluetooth.so [target-root]/usr/lib/gstreamer-1.0/
```

12.3 VNET Plugin

- BluetopiaPM include a VNET Plugin that should be copied to the target rootfs, The VNET is being used in PAN Demo application:

```
sudo mkdir -p [target-root]/home/root/BluetopiaPM/Bluetopia/VNET/driver
sudo cp -rf <INSTALLATION_DIR>/BluetopiaPM/Bluetopia/VNET/driver/VNETInst [target-root]/home/root/
BluetopiaPM/Bluetopia/VNET/driver/
sudo cp -rf <INSTALLATION_DIR>/BluetopiaPM/Bluetopia/VNET/*.ko [target-root]/home/root/BluetopiaPM/
Bluetopia/VNET/
```

12.3.1 Voice Over BLE Python Script

The **VoLE Demo** application utilizes a python script in order to decode, format and analyze the audio data.

```
sudo cp <INSTALLATION_DIR>/BluetopiaPM/sample/LinuxVoLE/audio_frame_linux.py [target-root]/home/root/
BluetopiaPM/bin
```


Copyright © 2021 Texas Instruments Incorporated

12.4 Init-scripts

Before the stack can be used make sure to have an updated Bluetooth Script (BTS) on your target platform. The latest BTS file and installation instructions can be found in WL18xx-BT-SP and can be copied directly to the SD card's firmware directory as shown below:

```
sudo cp -rf TIInit_X.Y.Z.bts [target-root]/lib/firmware/
```

In order to add support for **AVPR**, **BLE** or **BR/EDR** for the CC256x devices, see Appendix D.

12.5 Bluetooth Hardware Configuration Script

The bt-en.sh script described below creates a custom configuration file based on the board you're using.

Note

If using a Beaglebone, see the page on patching the device tree and skip the bt-en.sh setup script steps below:

1. Unzip the file Downloaded file and then copy it to the target platform:

   ```
   sudo cp Downloads/bt-en.sh [target-root]/home/root/BluetopiaPM/
   ```

2. Navigate to the correct directory on the target platform:

   ```
   cd /home/root/BluetopiaPM/
   ```

3. Update the scripts permissions:

   ```
   chmod +x bt-en.sh
   ```

4. Run the script:

   ```
   ./bt-en.sh
   ```

When using a custom platform, you'll need to create a similar script with your platform specific settings. As an example, you'll have to update the following variables:

- tty

  and
- gpio

Copyright © 2021 Texas Instruments Incorporated

Note, this includes the following statement at the start of the script. For example::

```
echo gpio
```

1. Create a new script file:

```
vi bt-en.sh
```

2. Copy the following into the empty file:

```
echo 16 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio16/direction
echo 1 > /sys/class/gpio/gpio16/value
echo "Done enabling BT"

gpio="nshutdown_gpio=16"
tty="tty=/dev/ttyS3"
flow="flow_cntrl=1"
baud_rate="baud_rate=3000000"
mkdir /home/root/tibt
echo $gpio > /home/root/tibt/config
echo $tty >> /home/root/tibt/config
echo $flow >> /home/root/tibt/config
echo $baud_rate >> /home/root/tibt/config
```

3. Update the gpio and tty variables to match your platform, in this example we're using GPIO 16 and tty S3.
4. Update the scripts permissions:

```
chmod +x bt-en.sh
```

5. Run the script:

```
./bt-en.sh
```

13 Terminal Connection

Plug in the RS-232 0 port to a pc via NULL modem and then look at the Device manager for Communications Port (COM x) under Ports (COM & LPT). Attach a terminal program like PuTTY to the serial port (COM x) for the board, x means which COM is open for Communications Port in Device Manager. The serial parameter to use is 115200 Baud rate. Once connected, turn on the device Power button and you should see the am335x getting initialized on the terminal.

```
Arago Project http://arago-project.org am335x-evm /dev/ttyS0
Arago 2015.05 am335x-evm /dev/ttyS0
am335x-evm login:
```

When the initialization is over, type root in order to use the file system.

Note

The serial port may differ for different platforms (for example, AM437x will be /dev/ttyS1).

```
am335x-evm login: root
root@am335x-evm:~#
```

14 Running a Sample Application

Now that you have installed Platform Manager to your host platform and copied relevant binaries (for this example at least SS1BTPM and LinuxSPPM should be copied) and init-scripts to your target, let's demonstrate running a Platform Manager application. The following should be done on the target (using USB or Ethernet connection):

```
# cd /home/root/BluetopiaPM/bin
# ./SS1BTPM &
# ./LinuxSPPM
```


Copyright © 2021 Texas Instruments Incorporated

You should now see LinuxSPPM's menu displayed in the terminal. Now let's try to initialize the BluetopiaPM Server in LinuxSPPM and turn on the Bluetooth Controller.

```
SPPM>Initialize 1
SPPM>SetDevicePower 1
```

If all went well you should see a "Device Powered On" message in the terminal. Congratulations, you have just run your first Platform Manager application! For more detailed information on using the LinuxSPPM Demo and for other Demo Guides,see the Bluetopia Platform Manager Demo Guide (located where?). The basics of Bluetopia Platform Manager's sample applications are discussed in the next section.

15 BluetopiaPM Samples

Samples are provided with the Bluetopia Platform Manager distribution to demonstrate how to use the BluetopiaPM modules. Each sample provides a command-line interface which can be used to send commands to the PM Server. The sample application commands encapsulate Bluetopia API commands and demonstrate how to use common API functions. The samples also display events and event data when an event is received from the PM Server.

The samples are provided to help the end user get started with application development in PM. The samples are not ready-to-use production applications, as an application like this depends on the project-specific requirements. However, the samples can be used as a starting point for application development. Re-building a sample that helps with getting started with application development in PM is discussed next.

Copyright © 2021 Texas Instruments Incorporated

A C256X EVM Modifications for COM8 Connector

For the CC2564MODNEM or CC2564C QFN boards

- On the EM board perform the following steps in order to connect it to the SDMMC board.

1. Desolder the headers **RF1** and **RF2**
2. Desolder the regulators **U2**, **U3**, and **U4**
3. Short resistor R2 **for the QFN-EM board**

Note

On the CC2564MODNEM board – Remove J4 jumper, and connect J3 (Changing VBAT_MCU to VBAT_EDGE).

Copyright © 2021 Texas Instruments Incorporated

B Bluetopia Build Log for Reference

```
make

Latitude-E6430:/media/work/BluetopiaPM-421110/AM335xBluetopiaLinuxProduction-4.2.1.1.1.0/build$ make
CC_S BTPMSRVR.c -o BTPMSRVR_S.o
LN BTPMSRVR_S.o libBTPM_S.a libBTPM_HAL_S.a libSS1BTAUD.a libSS1BTGAV.a libSS1BTAVC.a libSS1BTAVR.a
libSS1SBC.a libSS1BTFTP.a libSS1BTHDP.a libSS1BTHDS.a libSS1BTHIDH.a libSS1BTHID.a libSS1BTPAN.a
libBTPS_VNET.a libSS1BTPBA.a libSS1BTHFR.a libSS1BTMAP.a libSS1BTOPP.a libSS1BTCTS.a libSS1BTNDCS.a
libSS1BTRTUS.a libSS1BTDIS.a libSS1BTGAT.a libSS1BTGAPS.a libSS1BTCSCS.a libSS1BTHIDS.a
libSS1BTNDC.a libSS1BTRTU.a libSS1BTIAS.a libSS1BTHRS.a libSS1BTSCP.a libSS1BTLLS.a libSS1BTBAS.a
libSS1BTTPS.a libBTPS.a libBTPS_PLA.a libBTPS_OSA.a -o SS1BTPM
CC-C LinuxAUDM.c -o LinuxAUDM_C.o
CC-C AudioDecoderPM.c -o AudioDecoderPM_C.o
CC-C AudioEncoderPM.c -o AudioEncoderPM_C.o
CC_C BTPMMODC.c -o BTPMMODC_C.o
CC_C BTPMIPC.c -o BTPMIPC_C.o
CC-C BTPMVS.c -o BTPMVS_C.o
AR BTPMMODC_C.o BTPMIPC_C.o BTPMVS_C.o -o libBTPM_HAL_C.a
arm-linux-gnueabihf-ar: creating ../BluetopiaPM/lib/libBTPM_HAL_C.a
LN LinuxAUDM_C.o AudioDecoderPM_C.o AudioEncoderPM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxAUDM
CC-C LinuxDEVM.c -o LinuxDEVM_C.o
LN LinuxDEVM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxDEVM
CC-C LinuxHDPM.c -o LinuxHDPM_C.o
LN LinuxHDPM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHDPM
CC-C LinuxHDPM_11073.c -o LinuxHDPM_11073_C.o
CC-C IEEE_11073.c -o IEEE_11073_C.o
LN LinuxHDPM_11073_C.o IEEE_11073_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHDPM_11073
CC-C LinuxVoLE.c -o LinuxVoLE_C.o
LN LinuxVoLE_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxVoLE
CC-C LinuxHIDM.c -o LinuxHIDM_C.o
LN LinuxHIDM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHIDM
CC-C LinuxHRPM.c -o LinuxHRPM_C.o
LN LinuxHRPM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHRPM
CC-C LinuxOPPM.c -o LinuxOPPM_C.o
LN LinuxOPPM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxOPPM
CC-C LinuxPBAM.c -o LinuxPBAM_C.o
LN LinuxPBAM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxPBAM
CC-C LinuxSPPLE_PM.c -o LinuxSPPLE_PM_C.o
LN LinuxSPPLE_PM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxSPPLE_PM
CC-C LinuxCSCM.c -o LinuxCSCM_C.o
LN LinuxCSCM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxCSCM
CC-C LinuxIBeacon.c -o LinuxIBeacon_C.o
LN LinuxIBeacon_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxIBeacon
CC-C LinuxBASM.c -o LinuxBASM_C.o
LN LinuxBASM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxBASM
CC-C LinuxHOGM.c -o LinuxHOGM_C.o
LN LinuxHOGM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHOGM
CC-C LinuxPANM.c -o LinuxPANM_C.o
LN LinuxPANM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxPANM
CC-C LinuxSPPM.c -o LinuxSPPM_C.o
LN LinuxSPPM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxSPPM
CC-C SS1Tool.c -o SS1Tool_C.o
LN SS1Tool_C.o libBTPM_C.a libBTPM_HAL_C.a -o SS1Tool
CC-C LinuxFTPM_SRV.c -o LinuxFTPM_SRV_C.o
LN LinuxFTPM_SRV_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxFTPM_SRV
CC-C LinuxFTPM_CLT.c -o LinuxFTPM_CLT_C.o
LN LinuxFTPM_CLT_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxFTPM_CLT
CC-C LinuxFMPM_TAR.c -o LinuxFMPM_TAR_C.o
LN LinuxFMPM_TAR_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxFMPM_TAR
CC-C LinuxGATM_SRV.c -o LinuxGATM_SRV_C.o
LN LinuxGATM_SRV_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxGATM_SRV
CC-C LinuxGATM_CLT.c -o LinuxGATM_CLT_C.o
LN LinuxGATM_CLT_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxGATM_CLT
CC-C LinuxHFRM_AG.c -o LinuxHFRM_AG_C.o
LN LinuxHFRM_AG_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHFRM_AG
CC-C LinuxHFRM_HF.c -o LinuxHFRM_HF_C.o
LN LinuxHFRM_HF_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHFRM_HF
CC-C LinuxHDSM_AG.c -o LinuxHDSM_AG_C.o
LN LinuxHDSM_AG_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHDSM_AG
CC-C LinuxHDSM_HS.c -o LinuxHDSM_HS_C.o
LN LinuxHDSM_HS_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxHDSM_HS
CC-C LinuxMAPM_MCE.c -o LinuxMAPM_MCE_C.o
CC-C MsgStorePM.c -o MsgStorePM_C.o
LN LinuxMAPM_MCE_C.o MsgStorePM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxMAPM_MCE
CC-C LinuxMAPM_MSE.c -o LinuxMAPM_MSE_C.o
LN LinuxMAPM_MSE_C.o MsgStorePM_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxMAPM_MSE
CC-C LinuxPXPM_MON.c -o LinuxPXPM_MON_C.o
```


Copyright © 2021 Texas Instruments Incorporated

```
LN LinuxPXPM_MON_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxPXPM_MON
CC-C LinuxBle_SRV.c -o LinuxBle_SRV_C.o
LN LinuxBle_SRV_C.o libBTPM_C.a libBTPM_HAL_C.a -o LinuxBle_SRV
################################################################
# Build is Complete!
################################################################
```


Copyright © 2021 Texas Instruments Incorporated

C Sample Run

```
am335x-evm login:
root

root@am335x-
evm:~#

root@am335x-evm:~# ls -l ~/tibt/
config
-rw-r--r--    1 root     root            62 Nov 13 03:38 /home/root/tibt/
config
root@am335x-evm:~# cat ~/tibt/
config
nshutdown_gpio=3

tty=/dev/
ttyS1

flow_cntrl=1

baud_rate=115200

root@am335x-
evm:~#

root@am335x-evm:~# cd
BluetopiaPM

root@am335x-evm:~/
BluetopiaPM#

root@am335x-evm:~/
BluetopiaPM#

root@am335x-evm:~/BluetopiaPM# ./SS1BTPM
&
[1]
867

root@am335x-evm:~/BluetopiaPM# ./
LinuxDEVM


******************************************************************

* Command Options: 1) Initialize
*
*                  2) Cleanup
*
*                  3) QueryDebugZoneMask
*
*                  4) SetDebugZoneMask
*
*                  5) SetDebugZoneMaskPID
*
*                  6) ShutdownService
*
*                  7) RegisterEventCallback,
*
*                  8) UnRegisterEventCallback,
*
*                  9) QueryDevicePower
*
*                  10)SetDevicePower
*
*                  11)QueryLocalDeviceProperties
*
*                  12)SetLocalDeviceName
*
*                  13)SetLocalClassOfDevice
*
*                  14)SetDiscoverable
*
*                  15)SetConnectable
*
*                  16)SetPairable
*
```


Copyright © 2021 Texas Instruments Incorporated

```
*                   17)StartDeviceDiscovery
*
*                   18)StopDeviceDiscovery
*
*                   19)QueryRemoteDeviceList
*
*                   20)QueryRemoteDeviceProperties
*
*                   21)AddRemoteDevice
*
*                   22)DeleteRemoteDevice
*
*                   23)UpdateRemoteDeviceAppData
*
*                   24)DeleteRemoteDevices
*
*                   25)PairWithRemoteDevice
*
*                   26)CancelPairWithRemoteDevice
*
*                   27)UnPairRemoteDevice
*
*                   28)QueryRemoteDeviceServices
*
*                   29)QueryRemoteDeviceServiceSupported
*
*                   30)QueryRemoteDevicesForService
*
*                   31)QueryRemoteDeviceServiceClasses
*
*                   32)AuthenticateRemoteDevice
*
*                   33)EncryptRemoteDevice
*
*                   34)ConnectWithRemoteDevice
*
*                   35)DisconnectRemoteDevice
*
*                   36)SetRemoteDeviceLinkSniff
*
*                   37)SetRemoteDeviceLinkActive
*
*                   38)CreateSDPRecord
*
*                   39)DeleteSDPRecord
*
*                   40)AddSDPAttribute
*
*                   41)DeleteSDPAttribute
*
*                   42)EnableBluetoothDebug
*
*                   43)RegisterAuthentication
*
*                   44)UnRegisterAuthentication
*
*                   45)PINCodeResponse
*
*                   46)PassKeyResponse
*
*                   47)UserConfirmationResponse
*
*                   48)ChangeSimplePairingParameters
*
*                   49)EnableFeature
*
*                   50)DisableFeature
*
*                   51)QueryActiveFeatures
*
*                   52)SendRawHCICommand
*
*                   53)SendHCIVSCommand (Example)
*
*                   Help, Quit.
*
*****************************************************************

DEVM>1
```


Copyright © 2021 Texas Instruments Incorporated

```
1



BTPM_Initialize() Success:
0.

DEVM_RegisterEventCallback() Success:
5.
DEVM>10
1



echo 3 > /sys/class/gpio/
export

echo out > /sys/class/gpio/gpio3/
direction
echo 0 > /sys/class/gpio/gpio3/
value
echo 1 > /sys/class/gpio/gpio3/
value
echo 0 > /sys/class/gpio/gpio3/
value
echo 1 > /sys/class/gpio/gpio3/
value
[  127.268181] NET: Registered protocol family
15
[  127.409250] Initializing XFRM netlink
socket
BT COMM PORT (/dev/ttyS1):
1

Status: Executing BTS Script /lib/firmware/
TIInit_11.8.32.bts.
Status: BTS Script successfully
executed.
DEVM_PowerOnDevice() Success:
0.
DEVM>

Device Powered
On.

DEVM>44



DEVM_UnRegisterAuthentication()
Success.
DEVM>StartDeviceDiscovery



Usage: StartDeviceDiscovery
[Duration].
Function
Error.

DEVM>StartDeviceDiscovery
100



Attempting to Start Discovery (100
Seconds).


Local Device Properties
Changed.

DEVM>

Device Discovery
Started.

DEVM>DEVM_StartDeviceDiscovery() Success:
0.
```


Copyright © 2021 Texas Instruments Incorporated

```
DEVM>

Remote Device
Found.

BD_ADDR:
34F39ACA601A

COD:
0x0C012A

Device Name:
LTA0868345B

Device Flags:
0x80000601


NAME_KNOWN


EIR_DATA_KNOWN

RSSI:
-48

Trans. Power:
7

Friendly
Name:

App. Info:   :
00000000

Paired State :
FALSE

Connect State:
FALSE

Encrypt State:
FALSE

Sniff State  :
FALSE

Serv. Known  :
FALSE

DEVM>

Remote Device
Found.

BD_ADDR:
4CEB426EE54B

COD:
0x0C012A

Device Name:   RAJANI-
PC

Device Flags:
0x80000601


NAME_KNOWN


EIR_DATA_KNOWN

RSSI:
-80

Trans. Power:
4
```

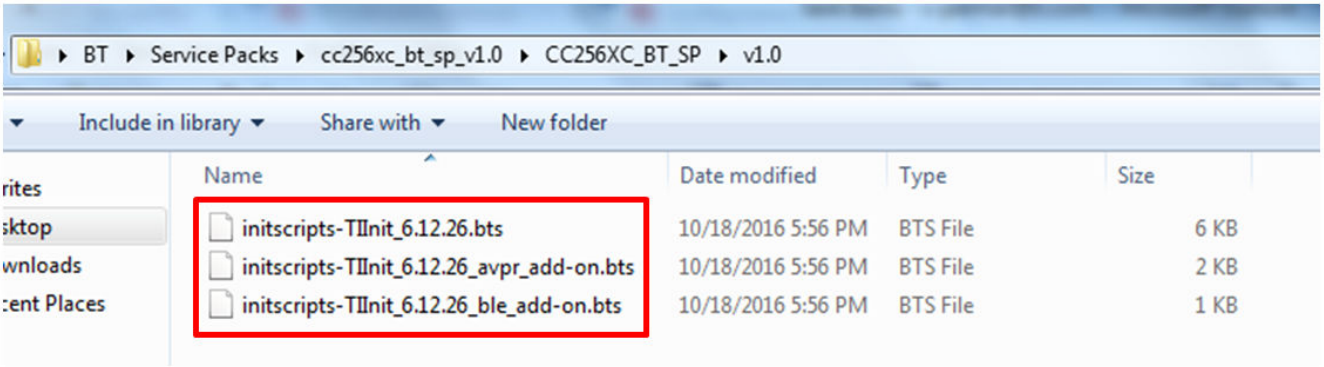

Copyright © 2021 Texas Instruments Incorporated

```
Friendly
Name:

App. Info:    :
00000000

Paired State :
FALSE

Connect State:
FALSE

Encrypt State:
FALSE

Sniff State  :
FALSE

Serv. Known  : FALSE
```


Copyright © 2021 Texas Instruments Incorporated

D Using the CC256x Service Pack With Linux (TI-BT-4-2-STACK-LINUX-ADDON)

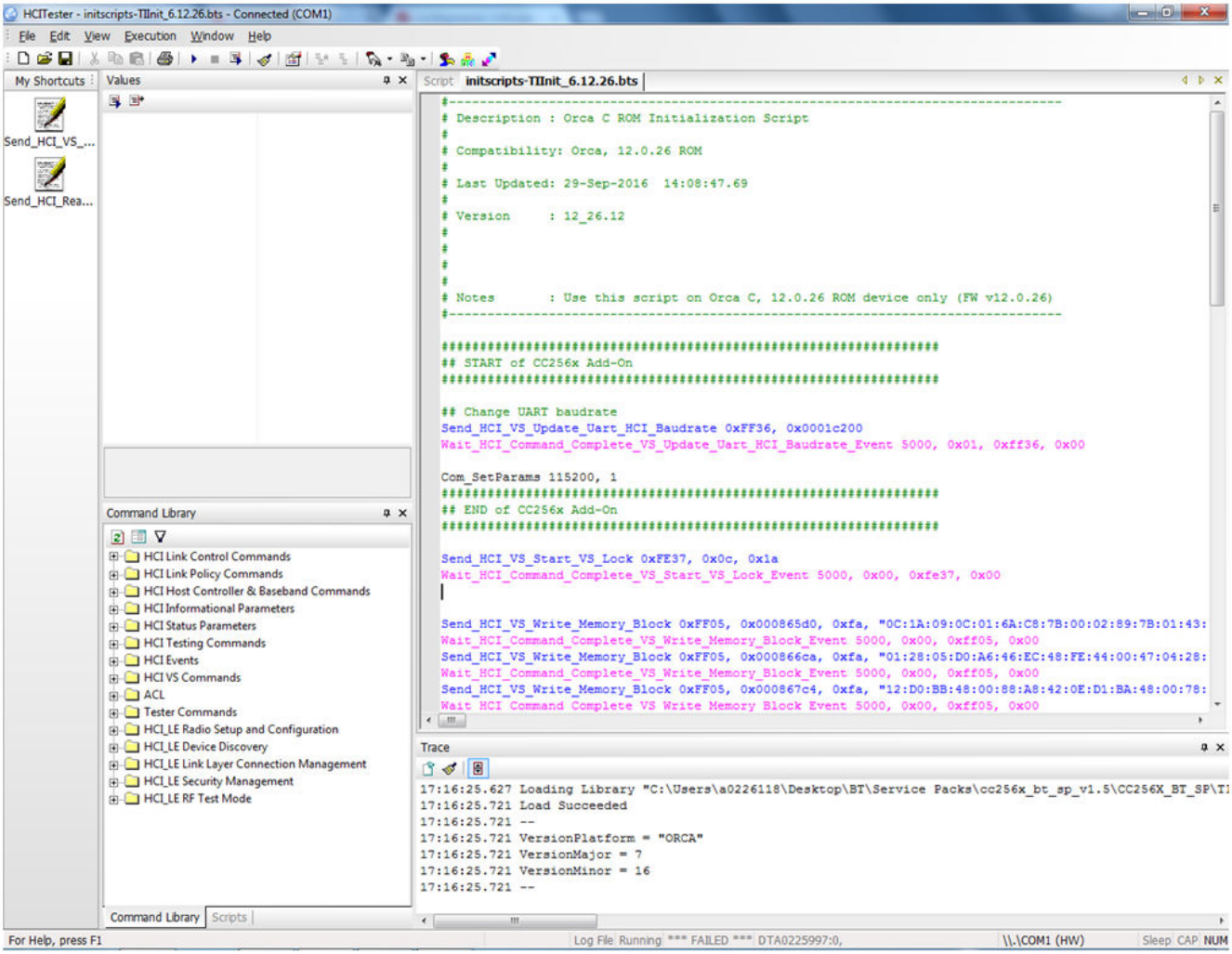
1. Download the latest Service Pack from the CC256XC-BT-SP product page. Links are located in the **CC256x Service Pack Downloads** section.



2. Open the initscripts-TIInit_6.12.26.bts with HCITester.

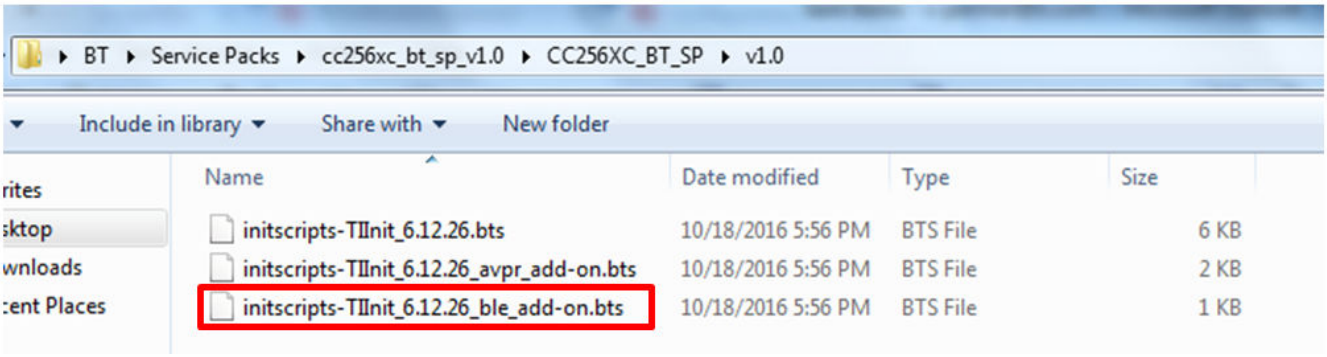
Note

Include the TIInit_6.12.26.xml included in the CC256XC-BT-SP download with HCITester if not already included. (For more details visit the HCI Tester User's Guide).

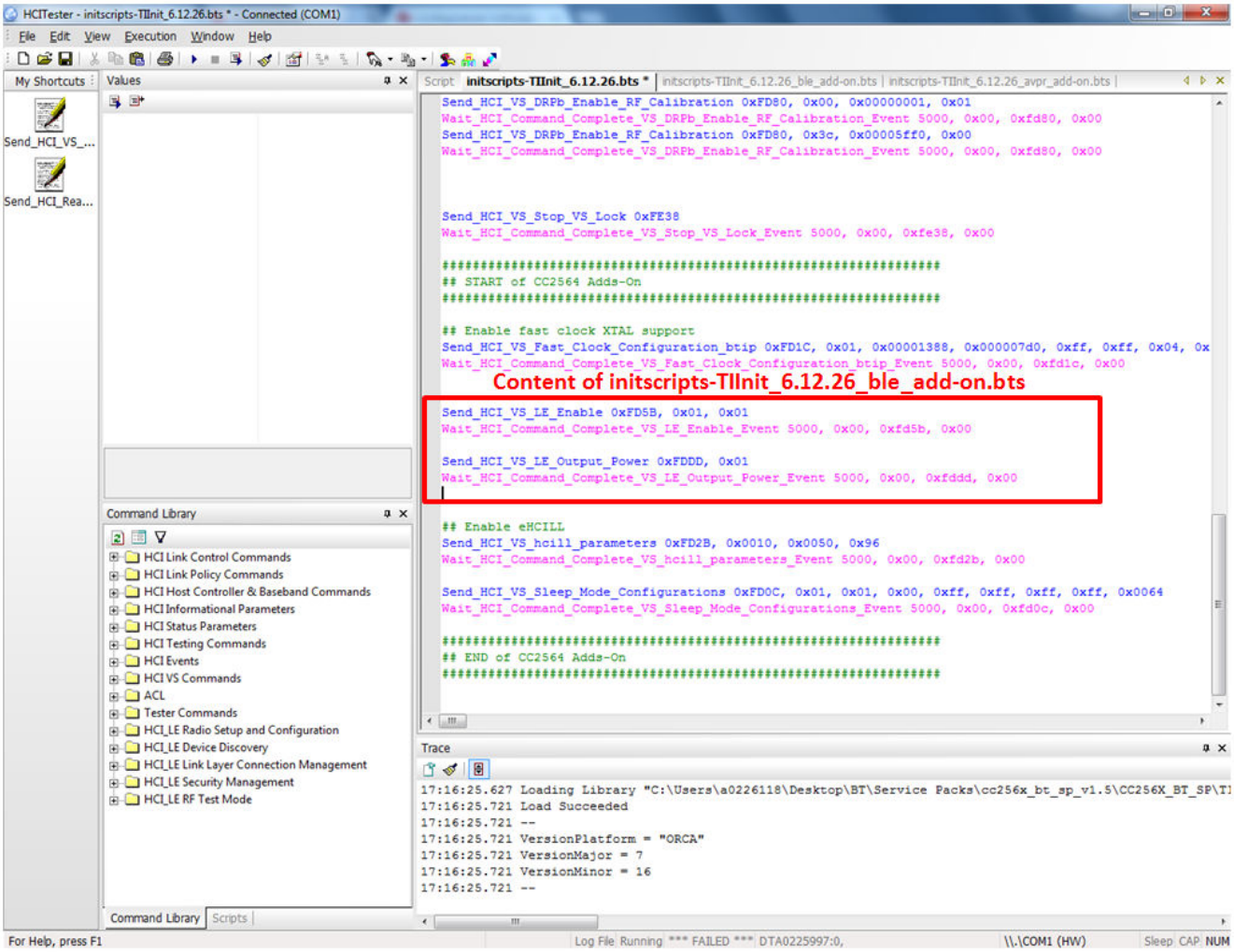


Copyright © 2021 Texas Instruments Incorporated

3. Depending on the use-case, you will need to copy the appropriate add-on (BLE or AVPR) to the main initscripts-TIInit_6.12.26.bts. If using BT only or BT+BLE (non-assisted modes): If unsure if the AVPR is needed or not, use this section and ignore the section b) below.
   a. BLE Addon
      i. Open the initscripts-TIInit_6.12.26_ble_add-on.bts in HCItester.



      ii. Copy the content of the initsctips-TIInit_6.12.26_avpr_add-on.bts at the end of the intsctips-TIInit_6.12.26.bts right before the ## Enable eHCILL section.

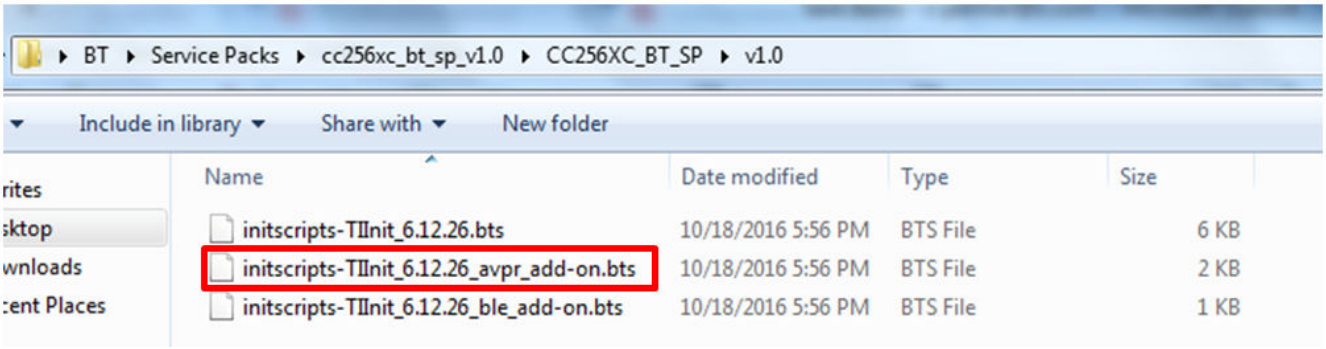


Copyright © 2021 Texas Instruments Incorporated

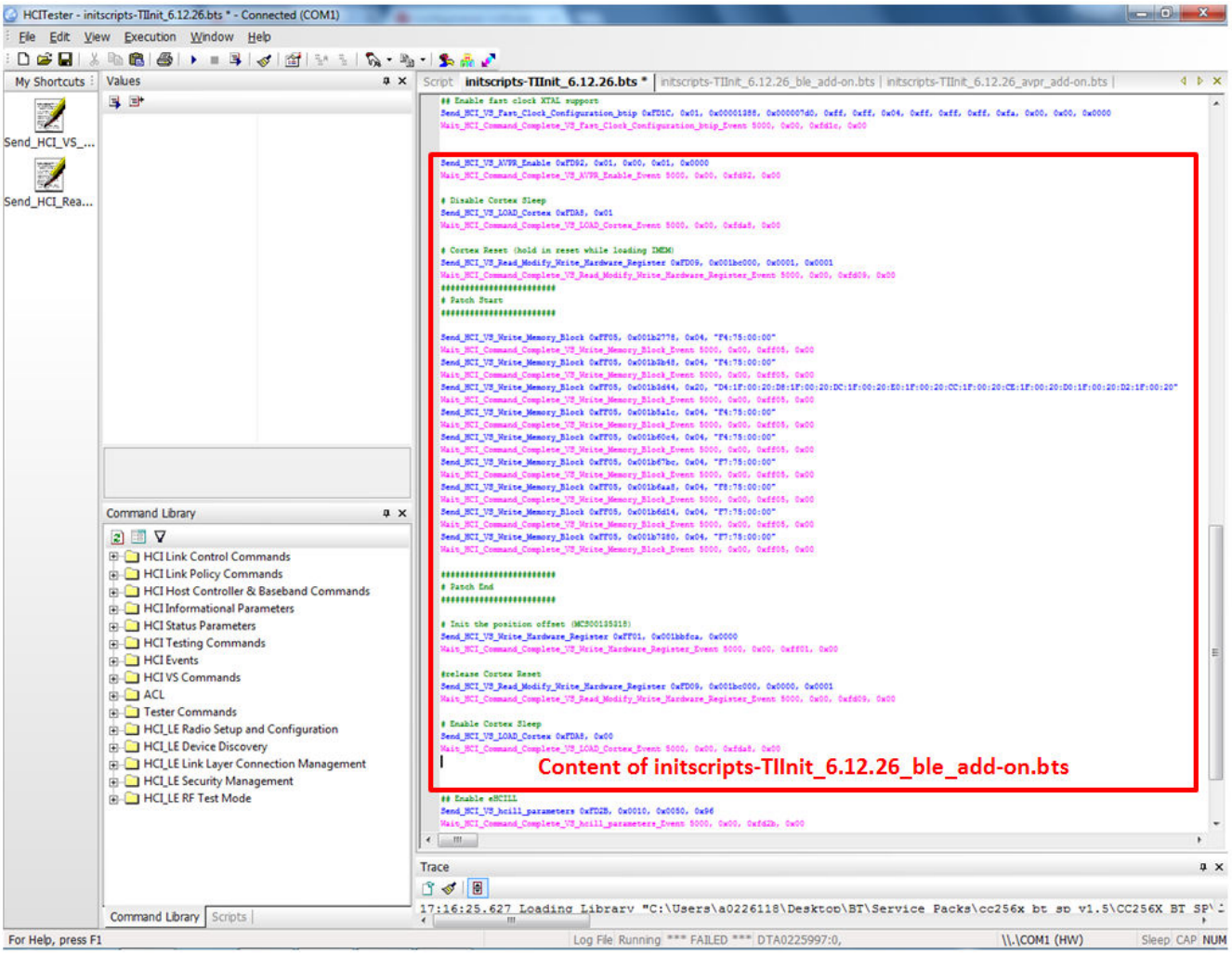
b. AVPR Addon

If using AVPR for Assisted A2DP (A3DP) or Assisted HFP: Use the AVPR patches only is using assisted mode. If unsure, follow the instructions to use the BLE add-on.

i. Open the initscripts-TIInit_6.12.26_avpr_add-on.bts in HCItester.



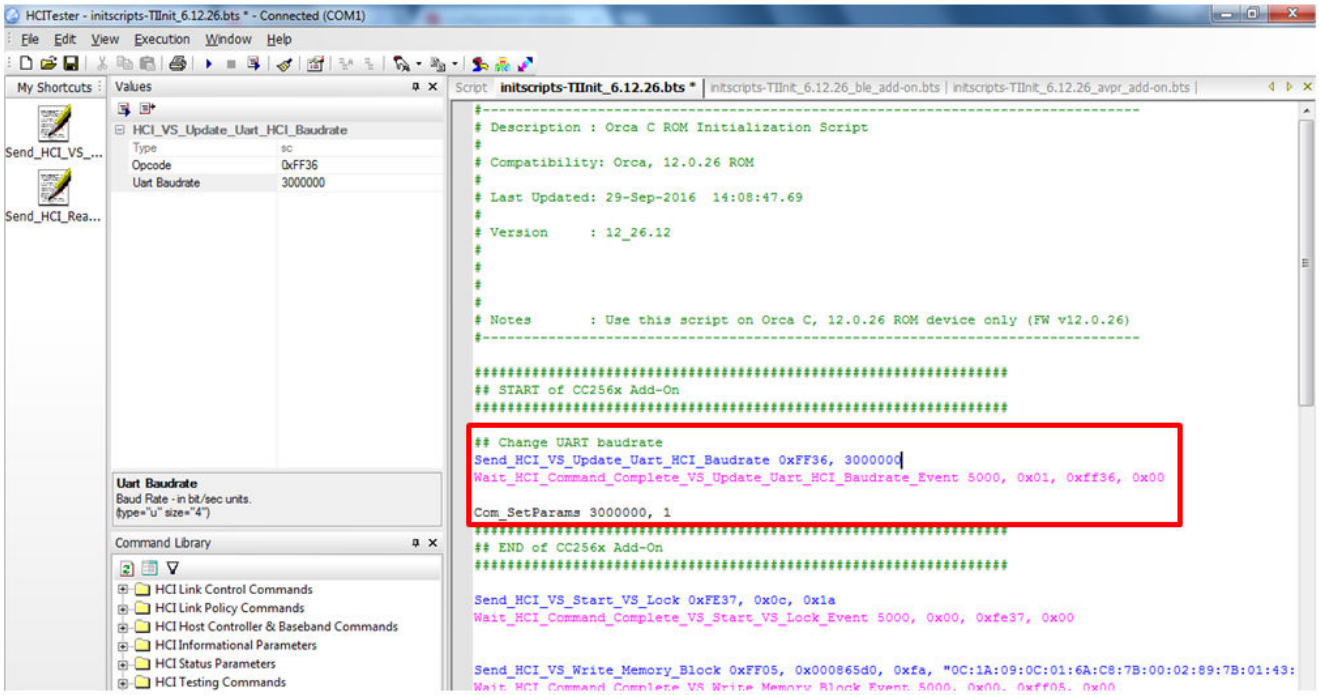
ii. Copy the content of the **initsctips-TIInit_6.12.26_avpr_add-on.bts** at the end of the **intsctips-TIInit_6.12.26.bts** right before the **## Enable eHCILL** section.



Copyright © 2021 Texas Instruments Incorporated

4. Optional: The default HCI UART baud rate is set to 115200 in the initscripts-TIInit_6.12.26.bts. If needed, you can modify the first HCI command in the initscript-TIInit_6.12.26.bts to change the baud rate.

   Example: Changing the baud rate to 3000000)



   The first command and its response between the host processor and the CC256X will be sent at 115200 (chip default) baud rate. After this first command of the initscript, the CC256X will switch to the specified new baud rate value (3000000). The host must also switch to this new baud rate after receiving the HCI command response for the first command.

5. In every event, replace the 0x00 value in the Number of HCI Commands parameter to any by selecting Edit -> Replace -> Replace All and configuring the parameters as follows:

   Find What: "5000, 0x00"

   Replace with: "5000, any"

   Replace all

6. Save the modified .bts file as TIInit_6.12.26.bts and copy this file to the [target-root]/lib/firmware directory of the Linux root.

   ```
   sudo cp TIInit_<VERSION>.bts [target-root]/lib/firmware/
   ```


Copyright © 2021 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (https:www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2021, Texas Instruments Incorporated