

# CC256x Service Pack User's Guide

---

Return to the CC256x Main Wiki (<http://processors.wiki.ti.com/index.php/CC256x>)

## Contents

---

### Overview

CC256x Service Pack Download links

### Using The CC256x Service Pack with Linux (TI-BT-4-2-STACK-LINUX-ADDON)

BLE Addon

AVPR Addon

## Overview

This wiki page describes all of the required steps to setup/update the CC256x Service Packs for the TI Dual-mode Bluetooth Stack running on a Linux platform.

 **Note:** Please note that the following instructions are useful to edit the service pack for use with TI Dual-mode Bluetooth Stack for Linux. For other Bluetooth stacks like BlueZ, refer to the service-pack on this git : <https://git.ti.com/ti-bt/service-packs/trees/master/initscripts>

The HCITester application, used for editing BTS scripts, can be downloaded [here](http://www.ti.com/tool/WILINK-BT_WIFI-WIRELESS_TOOLS) ([http://www.ti.com/tool/WILINK-BT\\_WIFI-WIRELESS\\_TOOLS](http://www.ti.com/tool/WILINK-BT_WIFI-WIRELESS_TOOLS)) as part of the Wilink tools package. The full User's guide with initial setup instructions may be found [here](http://www.ti.com/lit/pdf/swru136) (<http://www.ti.com/lit/pdf/swru136>).

## CC256x Service Pack Download links

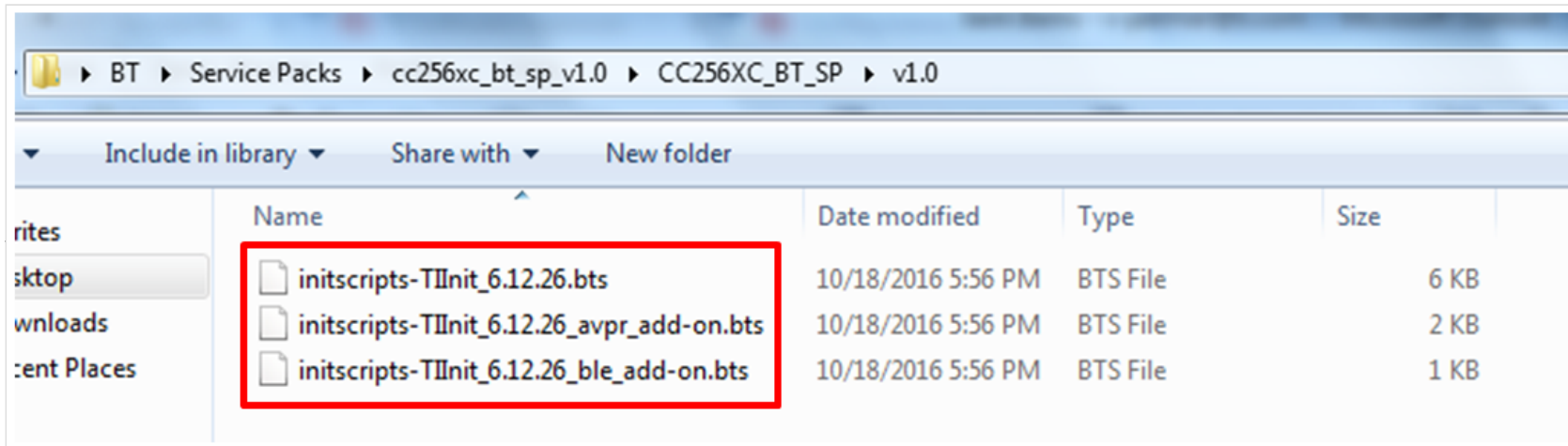
---

[CC2564xC Init Script Download \(http://www.ti.com/tool/cc256xc-bt-sp\)](http://www.ti.com/tool/cc256xc-bt-sp)


[CC2564xB Init Script Download \(http://www.ti.com/tool/cc256xb-bt-sp\)](http://www.ti.com/tool/cc256xb-bt-sp)

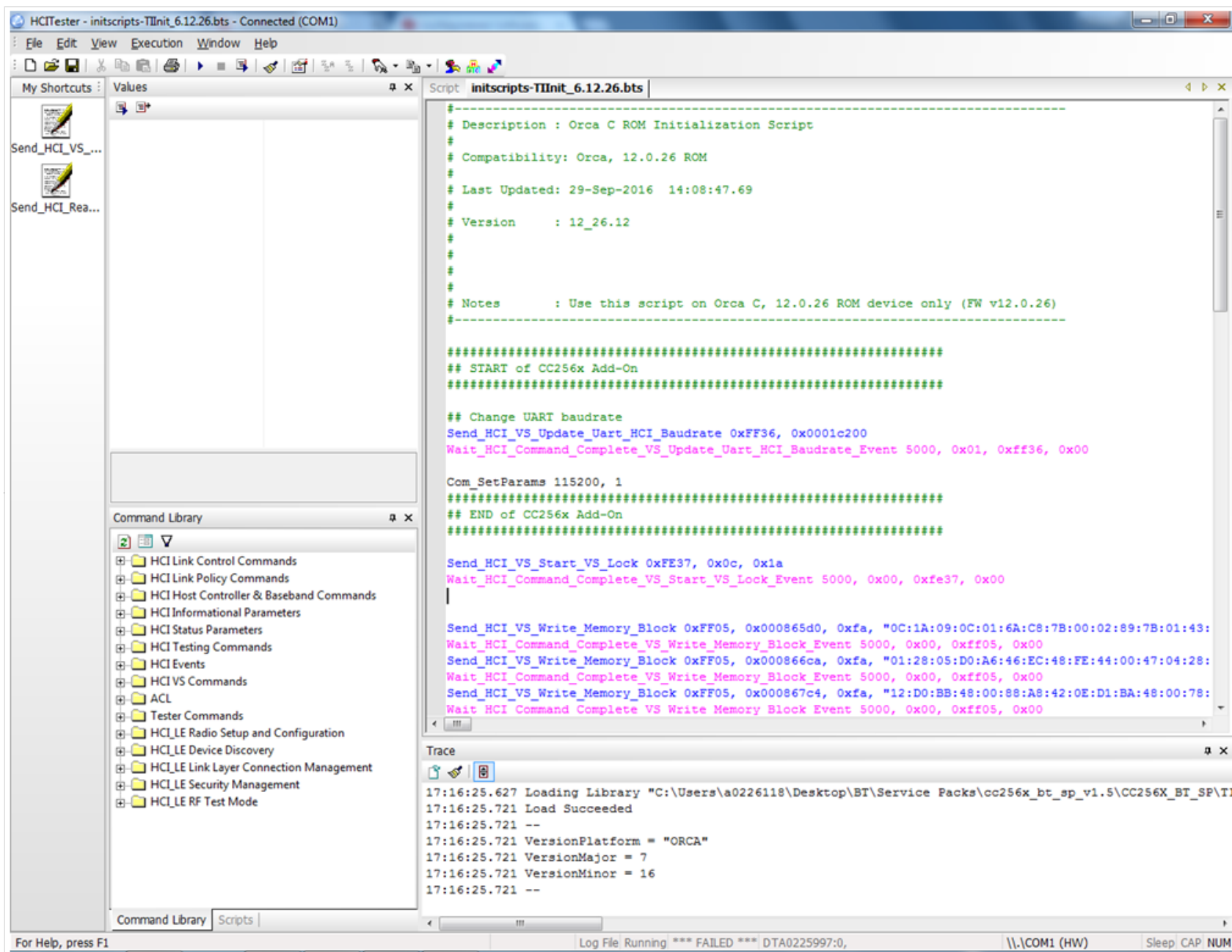
## Using The CC256x Service Pack with Linux (TI-BT-4-2-STACK-LINUX-ADDON)

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.

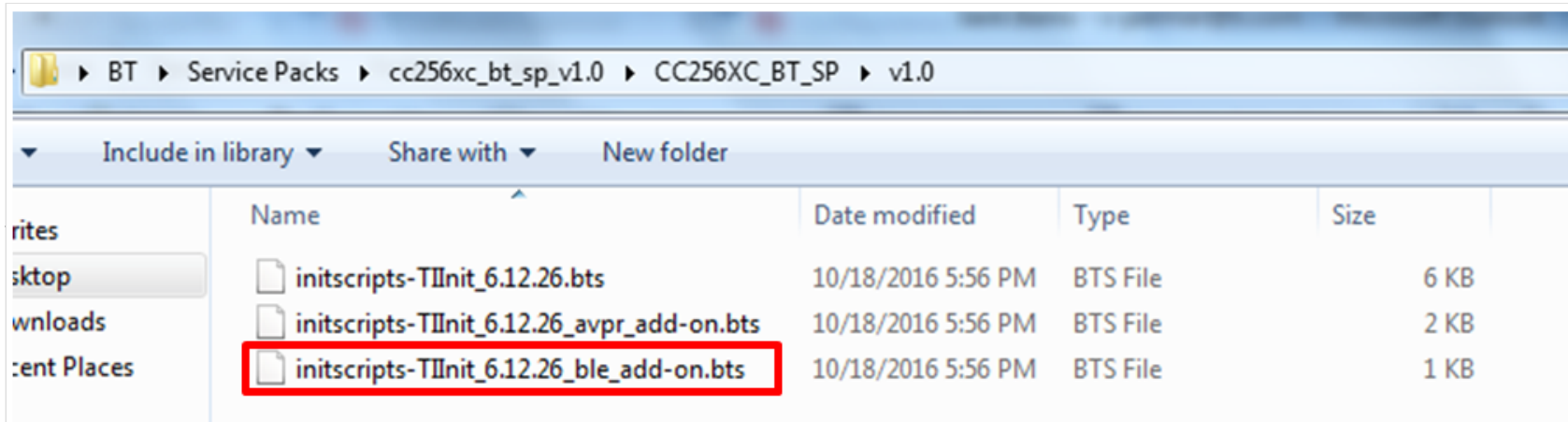
 Note: **Include the TIinit\_6.12.26.xml included in the CC256XC-BT-SP download with HCITester if not already included. (For more details visit the HCI Tester User's Guide [here \(http://www.ti.com/lit/pdf/swru136\)](http://www.ti.com/lit/pdf/swru136)).**



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.

## BLE Addon

i. Open the **initscripts-TIinit\_6.12.26\_ble\_add-on.bts** in HCItester.



ii. Copy the content of the **initscripts-TIinit\_6.12.26\_ble\_add-on.bts** at the end of the **initscripts-TIinit\_6.12.26.bts** right before the **## Enable eHCILL** section.

The screenshot shows the HCITester application interface. The main window displays the content of the script `initscripts-TIinit_6.12.26_ble_add-on.bts`. The script includes various HCI commands and wait statements for RF calibration, stop vs lock, and fast clock XTAL support. A red box highlights the following section:

```

Send_HCI_VS_LE_Enable 0xFD5B, 0x01, 0x01
Wait_HCI_Command_Complete_VS_LE_Enable_Event 5000, 0x00, 0xfd5b, 0x00

Send_HCI_VS_LE_Output_Power 0xFDDD, 0x01
Wait_HCI_Command_Complete_VS_LE_Output_Power_Event 5000, 0x00, 0xfddd, 0x00

```

The Command Library on the left lists various HCI commands, including HCI Link Control, HCI Link Policy, HCI Host Controller & Baseband, HCI Informational Parameters, HCI Status Parameters, HCI Testing Commands, HCI Events, HCI VS Commands, ACL, Tester Commands, HCI\_LE Radio Setup and Configuration, HCI\_LE Device Discovery, HCI\_LE Link Layer Connection Management, HCI\_LE Security Management, and HCI\_LE RF Test Mode.

The Trace window at the bottom shows the following log entries:

```

17:16:25.627 Loading Library "C:\Users\A0226118\Desktop\BT\Service Packs\cc256x_bt_sp_v1.5\CC256X_BT_SP\I
17:16:25.721 Load Succeeded
17:16:25.721 --
17:16:25.721 VersionPlatform = "ORCA"
17:16:25.721 VersionMajor = 7
17:16:25.721 VersionMinor = 16
17:16:25.721 --

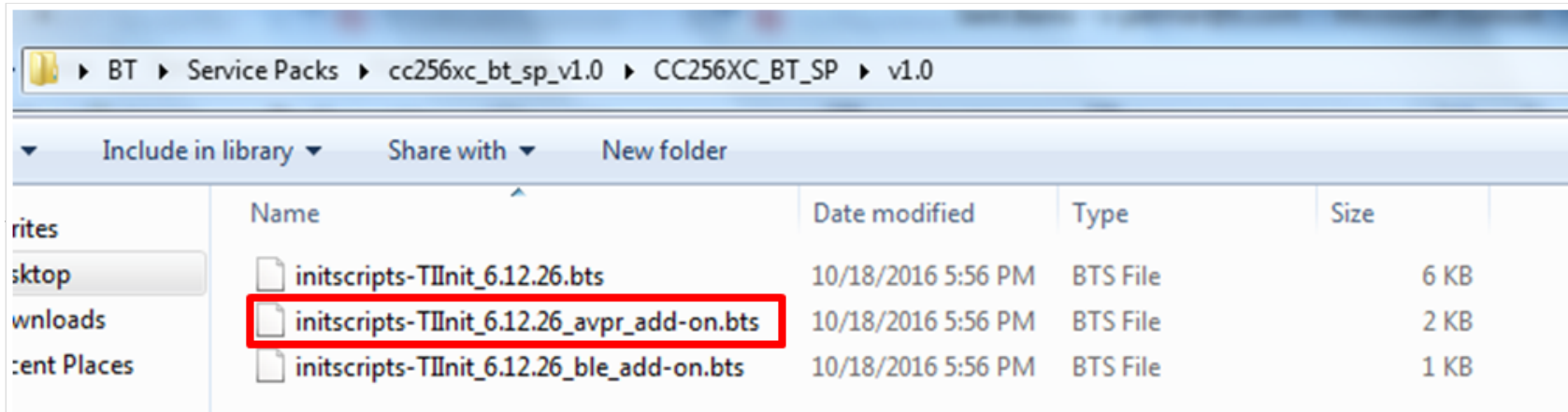
```

The status bar at the bottom indicates "Log File Running \*\*\* FAILED \*\*\* DTA0225997:0, \\.\COM1 (HW) Sleep CAP NUM".

## AVPR Addon

If using AVPR for Assisted A2DP (A3DP) or Assisted HFP: Use the AVPR patches only 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 `initscripts-TIInit_6.12.26_avpr_add-on.bts` at the end of the `initscripts-TIInit_6.12.26.bts` right before the `## Enable eHCILL` section.

The screenshot shows the HCI Tester application interface. The main window displays the script content for `initscripts-TIInit_6.12.26_ble_add-on.bts`. The script is as follows:

```

## Enable fast clock XTAL support
Send_HCI_VS_Fast_Clock_Configuration_btsp 0xF01C, 0x01, 0x00001388, 0x00000740, 0xFF, 0xFF, 0x04, 0xFF, 0xFF, 0xFF, 0x0A, 0x00, 0x00, 0x0000
Wait_HCI_Command_Complete_VS_Fast_Clock_Configuration_btsp_Event 5000, 0x00, 0xFFd0, 0x00

Send_HCI_VS_AVPR_Enable 0xF092, 0x01, 0x00, 0x01, 0x0000
Wait_HCI_Command_Complete_VS_AVPR_Enable_Event 5000, 0x00, 0xFFd2, 0x00

# Disable Cortex Sleep
Send_HCI_VS_LOAD_Cortex 0xFDA8, 0x01
Wait_HCI_Command_Complete_VS_LOAD_Cortex_Event 5000, 0x00, 0xFFd8, 0x00

# Cortex Reset (hold in reset while loading IMEM)
Send_HCI_VS_Read_Modify_Write_Hardware_Register 0xFD09, 0x001bc000, 0x0001, 0x0001
Wait_HCI_Command_Complete_VS_Read_Modify_Write_Hardware_Registers_Event 5000, 0x00, 0xFFd9, 0x00
*****
# Patch Start
*****

Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b2778, 0x04, "F4:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b3b48, 0x04, "F4:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b3644, 0x20, "D4:1F:00:20:DB:1F:00:20:DC:1F:00:20:E0:1F:00:20:CC:1F:00:20:CE:1F:00:20:D0:1F:00:20:D2:1F:00:20"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b3a1c, 0x04, "F4:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b60c4, 0x04, "F4:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b67bc, 0x04, "F7:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b6aa8, 0x04, "F8:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b6d14, 0x04, "F7:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
Send_HCI_VS_Write_Memory_Block 0xFF08, 0x001b7280, 0x04, "F7:75:00:00"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xFF08, 0x00
*****
# Patch End
*****

# Init the position offset (MC200195318)
Send_HCI_VS_Write_Hardware_Register 0xFF01, 0x001bbfca, 0x0000
Wait_HCI_Command_Complete_VS_Write_Hardware_Registers_Event 5000, 0x00, 0xFF01, 0x00

#release Cortex Reset
Send_HCI_VS_Read_Modify_Write_Hardware_Register 0xFD09, 0x001bc000, 0x0000, 0x0001
Wait_HCI_Command_Complete_VS_Read_Modify_Write_Hardware_Registers_Event 5000, 0x00, 0xFFd9, 0x00

# Enable Cortex Sleep
Send_HCI_VS_LOAD_Cortex 0xFDA8, 0x00
Wait_HCI_Command_Complete_VS_LOAD_Cortex_Event 5000, 0x00, 0xFFd8, 0x00

|
Content of initscripts-TIInit_6.12.26_ble_add-on.bts

## Enable eMCILL
Send_HCI_VS_hcill_parameters 0xF02B, 0x0010, 0x0080, 0x96
Wait_HCI_Command_Complete_VS_hcill_parameters_Event 5000, 0x00, 0xFF2B, 0x00

```

The script is titled "Content of initscripts-TIInit\_6.12.26\_ble\_add-on.bts". The interface also shows a Command Library on the left and a Trace window at the bottom.

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

```

# Description : Orca C ROM Initialization Script
#
# Compatibility: Orca, 12.0.26 ROM
#
# Last Updated: 29-Sep-2016 14:08:47.69
#
# Version   : 12_26.12
#
#
# Notes    : Use this script on Orca C, 12.0.26 ROM device only (FW v12.0.26)
#-----
#####
## START of CC256x Add-On
#####

## Change UART baudrate
Send_HCI_VS_Update_Uart_HCI_Baudrate 0xFF36, 3000000
Wait_HCI_Command_Complete_VS_Update_Uart_HCI_Baudrate_Event 5000, 0x01, 0xff36, 0x00

Com_SetParams 3000000, 1

#####
## END of CC256x Add-On
#####

Send_HCI_VS_Start_VS_Lock 0xFE37, 0x0c, 0x1a
Wait_HCI_Command_Complete_VS_Start_VS_Lock_Event 5000, 0x00, 0xfe37, 0x00

Send_HCI_VS_Write_Memory_Block 0xFF05, 0x000865d0, 0xfa, "0C:1A:09:0C:01:6A:C8:7B:00:02:89:7B:01:43:"
Wait_HCI_Command_Complete_VS_Write_Memory_Block_Event 5000, 0x00, 0xff05, 0x00

```

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/
```

**Note:** When using an SD Card, the target filesystem [target-root] may be /media/rootfs.

{{ Keystone= C2000=For DaVinci=For MSP430=For OMAP35x=For OMAPL1=For MAVRK=For For technical support



1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **CC256x Service Pack User's Guide** here.

- For technical support on MultiCore devices, please post your questions in the [C6000 MultiCore Forum](#)
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the [BIOS Forum](#)

Please post only comments related to the article **CC256x Service Pack User's Guide** here.

*technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article **CC256x Service Pack User's Guide** here.*

*technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article **CC256x Service Pack User's Guide** here.*

*technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article **CC256x Service Pack User's Guide** here.*

*technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **CC256x Service Pack User's Guide** here.*

*technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **CC256x Service Pack User's Guide** here.*

*technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article **CC256x Service Pack User's Guide** here.*

*please post your questions at <http://e2e.ti.com>. Please post only comments about the article **CC256x Service Pack User's Guide** here.*

}}

## Links



[Amplifiers & Linear](#)

[Audio](#)

[Broadband RF/IF & Digital Radio](#)

[Clocks & Timers](#)

[Data Converters](#)

[DLP & MEMS](#)

[High-Reliability](#)

[Interface](#)

[Logic](#)

[Power Management](#)

[Processors](#)

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

[Switches & Multiplexers](#)

[Temperature Sensors & Control ICs](#)

[Wireless Connectivity](#)

Retrieved from "[https://processors.wiki.ti.com/index.php?title=CC256x\\_Service\\_Pack\\_User%27s\\_Guide&oldid=229296](https://processors.wiki.ti.com/index.php?title=CC256x_Service_Pack_User%27s_Guide&oldid=229296)"

This page was last edited on 3 July 2017, at 22:23.

Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.