**MSP432 OVER THE AIR FIRMWARE UPDATE USING BSL (SERIAL BOOTLOADER)**

**FOLLOW THE BELOW GIVEN STEPS FOR INTEGRATING AND TESTING FIRMWARE UPDAT FUNCTIONALITY AS PART OF YOUR APPLICATION CODE:**

**(Images above and context/caption below)**

(Step 1 is optional. Do it only when you think you are not sure whether BSL is present in the information memory space.)

1. Erase the entire main flash memory and information memory and flash fresh copy of BSL.
2. Now build (enable hex utility: output TI txt format) the code which has SW invocation procedure.
   1. If trying in your own code put the BSL SW invocation code inside main file at the beginning.
   2. Should be the first code to execute in main() after power cycle/reset.(Working method)
   3. Doing other tasks and then doing SW invocation changes the BSL ram space. (0x2000:0000 and 0x2000:07FF).This causes **[ERROR\_MESSAGE] Initialization of P4xx BSL failed!** while running BSL script.
   4. After building, keep the TI txt file in the same folder where the script is.
   5. You will find a wrong password command in script.
   6. When entered a wrong password. It does a memory erase. Only main flash memory. Does not erase BSL.(See reference section for more about wrong and right passwords)
3. Now install BSL-Scripter and write the script
   1. Download and install BSL-Scripter from reference section link.
   2. Enter the path in environment variable of BSL-Scripter.exe after installing.
   3. See samples. Select the correct script example for the device MSP432P401R (P4xx) UART method.
   4. Only change COM#, baud rate to **9600** binary image TI txt filename and remove PARITY option.
   5. Use only TI txt output format for **RX\_DATA\_BLOCK\_32** command. Intel hex does not work in some cases.
4. For the first time flash the application code which has BSL SW invocation code using CCS.

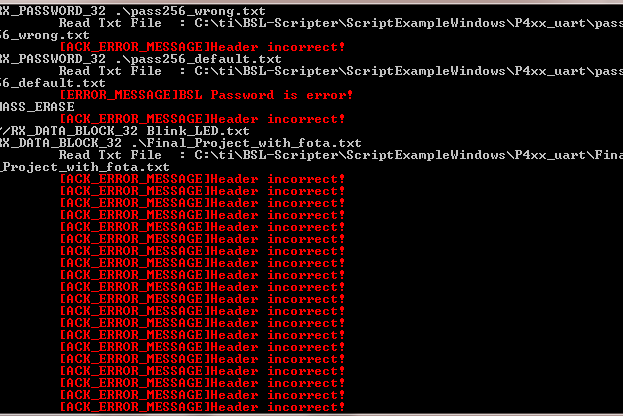


1. Run the application. In the rover console enter "fota\_begin" and press ENTER. **And close the serial terminal.**



(Now keep the BSL-Scripter ready to operate. Open cmd.Have the folder where script is, as your current working directory.)

1. Now reset the board.
   1. Power cycling is the best method.(Chosen)
   2. Doing a reset with help of CCS is a good alternative.



* 1. Using RESET button gives **HEADER INCORRECT ERROR**.
  2. Doing a RESET inside code never works.

1. **Now BSL invocation happens.**

(Calling reset inside code: The reason can be because the previous application has tampered the BSL RAM space.

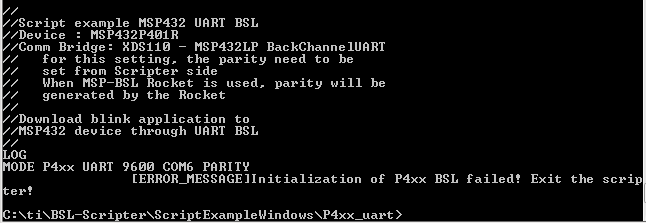
That is why a reset from inside code does not work. So performing a hardware reset is valid method.

This way proper boot sequence happens: Boot-code->Application code (user code) or BSL code.

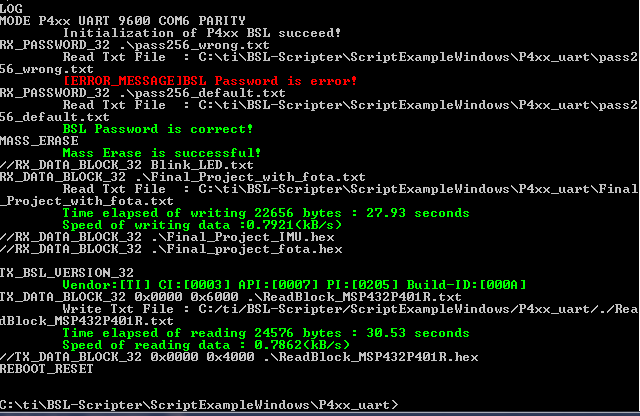
Now since the BSL SW invocation happens in main() as first set of instructions successful initialization BSL happens when script is run.)



1. Within **10s** , in the previously opened cmd, type BSL-Scripter.exe
   1. It prompts for file name. Since the script is in the pwd enter .\script.txt or simply script.txt
   2. Alternatively type : BSL-Scripter .\script.txt in a single step.



* 1. When not done within 10s you will get **[ERROR\_MESSAGE] Initialization of P4xx BSL failed!**



1. After firmware update complete. Compare the flash memory(using uniflash) with

the generated ReadBlock\_MSP432P401R.txt file (folder where script is) .Both must match.

1. Need not reset. The script does a reset at the end. The new code starts automatically.

**NOTE**: If you decide to **cancel the firmware update**, but you have already given "fota\_begin" command (which means you are in the firmware update mode), simply press RESET button on the board or power cycle the board "TWICE" to enter application mode.

First RESET ->Mode enters to firmware update mode.

Second RESET ->The current application in the flash starts running

(Check by entering help in serial terminal. Rover console should appear.)

**REFERENCE SECTION:**

**To erase main flash memory and information memory and isntall fresh BSL:::**

(Fatmawati answer)

- download the package under: <http://www.ti.com/tool/MSPBSL>

- import the project in CCS (should work with version 6.1.3 above). you can copy the project into workspace or simply import it.

- In BSL432\_Peripheral\_Interface\_eUSCI\_UART\_SPI\_I2C\_IRQ.c ->remove all and paste the contents of **custom\_BSL432\_Peripheral\_Interface\_eUSCI\_UART\_SPI\_I2C\_IRQ.c**

- in CCS, build the project in Release mode

- after it is successfully built, but before programming the device, go to Project Properties

- on the list on the left side choose Debug

- Then on the next column on the right, there is MSP432 settings

- on the next column on the right, choose erase method to be Erase main and information memory, also check the checkbox of Allow BSL information memory

- then you can start the program the BSL.

- close / terminate the debug session, then try to communicate with BSL again

**Read** : https://e2e.ti.com/support/microcontrollers/msp430/f/166/p/594448/2192689#2192689

**How to generate output TI txt file(Select ti txt file from output format options)**

1. http://e2e.ti.com/support/tools/ccs/f/81/t/723312?CCS-How-to-create-hex-file-after-build-

**Sample codes and BSL-Scripter scripts:::**

1. https://e2e.ti.com/support/microcontrollers/msp430/f/166/p/684761/2551311

**To perform HW or SW update VISHNU K answer:::**

1. https://e2e.ti.com/support/microcontrollers/msp430/f/166/t/763633?Linux-MSP432P401R-programming-MSP432-using-BSL-UART

**NOTE**: Within 10s of the BSL SW invocation we should run the script.Else BSL will go to sleep mode.

And will get Initialization error.

**BSL\_Scripter commands:::**

1. http://www.ti.com/lit/ug/slau655g/slau655g.pdf

NOTE: Use sample codes and modify. They come with BSL-Scripter package.

Should add path of the .exe file in environment variables to use for cmd.

**BSL scripter.exe and sample scripts.**

1. <http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSPBSL_Scripter/latest/index_FDS.html>

**Install fresh BSL:::**

1. http://www.ti.com/lit/ug/slau622i/slau622i.pdf

**Only works for fresh installation of BSL problem:::**

1. https://e2e.ti.com/support/microcontrollers/msp430/f/166/t/684761

**BSL code(for fresh BSL installation):::**

1. http://software-dl.ti.com/msp430/msp430\_public\_sw/mcu/msp430/MSPBSL\_CustomBSL432/latest/index\_FDS.html

**Programming firmwares using uniflash:::**

1. http://www.ti.com/lit/ug/slau799b/slau799b.pdf

**BSL memory space:::**

1. https://books.google.com/books?id=TDENDgAAQBAJ&pg=PA311&lpg=PA311&dq=flash+memory+region+ti+bsl+sector&source=bl&ots=vcT0SoAnlM&sig=ACfU3U1G6ODS8Vqla9DFO-LJHWGWahDXOw&hl=en&sa=X&ved=2ahUKEwjD9qbuj\_\_oAhVFbc0KHfD3CFIQ6AEwAHoECAoQAQ#v=onepage&q=flash%20memory%20region%20ti%20bsl%20sector&f=false

**BSL ram memory overwrite when called from main:::**

1. http://www.ti.com/lit/ug/slau622i/slau622i.pdf pg5

**NOTE**: There is something called bootcode.This calls BSL when 0x0-0x100 is all FF (EMPTY/NO APPLICATION CODE) .Other ways to invoke BSL is SW or HW invocation.

On power ON I think boot-code execs first -> Checks if main flash empty (when 0x0-0x100 is all FF)

If YES calls BSL, else run application.

I think at the same point it also puts some data in its RAM space.

If application code is run, this space should be getting tampered.

That is why SW invocation should be first code in main after disabling watchdog, to do a successful firmware update.

Read: http://e2e.ti.com/support/microcontrollers/msp430/f/166/t/551970?MSP432P401R-Rev-C-How-to-invoke-the-entry-sequence-of-MSP-FET

Boot-code starts in reset.

**Boot order:** http://www.ti.com/lit/ug/slau622i/slau622i.pdf?ts=1587775445759 - pg8

For FLASH memory related operations. See sample codes from TI resources. Check under Driverlib

Cannot directly write to flash. Banks-> Sectors are protected after reset.

**Wrong and right password:::**

1. Compares the supplied pass with first 256 bytes. If same then proceed and do memory erase command.

After memory erase first 256 bytes will be all FF. (This pattern you can see in the default password.)

Now default pass becomes correct pass and so proceeds.

This is a technique when we don't know the start 256 bytes.

**CODE REUSE:**

**SW INVOCATION CODE:**

1. <http://www.ti.com/lit/ug/slau622i/slau622i.pdf?ts=1587836446378> Page 6

**BSL CODE UPDATE SCRIPT:**

1. <http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSPBSL_Scripter/latest/index_FDS.html>
   1. Used file: script\_P4xx\_uart.txt