Application Report
SLAA660B–March 2015–Revised November 2016

Software IP Protection on MSP432P4xx Microcontrollers

Rakesh Hariharan

ABSTRACT

Differentiations in embedded software applications enable differentiated products. Companies invest significant money in building differentiated software application. Hence, protecting this investment (application or portions of the application) is extremely important. This application note describes how to protect software intellectual property (IP) running on the Texas Instruments (TI) MSP432P4xx family of microcontrollers.

Contents

List of Figures

List of Tables

MSP432, Code Composer Studio are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

Copyright © 2015–2016, Texas Instruments Incorporated

1 Introduction

Software intellectual property (IP) plays a crucial role in embedded systems today. Software IPs can coexist with many other software components in the system. Hence, protecting these IPs becomes very important.

This application note describes:

- How to enable software IP protection feature on the MSP432P4xx device family to protect the software IPs from debug intrusions and malicious code
- How to perform a secure in-field update for software IPs

2 Typical Microcontroller Device Security

In embedded systems with microcontroller devices, enabling security typically means stopping debug access to the device. During a typical software development cycle, the application developer would develop and debug the application software completely. After this development cycle, the device debug path (JTAG or SWD) is blocked using a user-configurable setting. This is done to prevent unauthorized debug of the device.

Preventing debug access from JTAG/SWD only stops unauthorized debug of the device. This does not prevent reading out device memory locations where the critical software IP could be located. Ideally, software IP should be located in memories that are "execute only". This is to protect the software IP from being read out and being reverse engineered. Hence, any code (in the device) that uses the software IP should only be able to call into it, and not read it out.

TI's MSP432P4xx family of devices enable software IP protection by providing options to users to define "Software IP" memory zones.

3 Boot Overrides

Before discussing the IP protection in MSP432P4xx devices, this section introduces the concept of boot overrides.

Boot overrides are special functions executed by the device boot-code (TI code executing in a secure environment). Boot overrides can be requested by a user to execute predefined commands. Boot-Override-Mailbox (BOM) is the communication interface between the user and device boot-code to invoke boot-overrides. BOM consists of commands and parameters recognized by the device boot-code.

BOM is located in the flash information memory area of the device. For example, in the MSP432P401x device, it is located at the beginning of the flash information memory at 0x0020:0000. BOM follows a fixed structure. This structure includes commands, arguments for commands, and acknowledgments to commands. The details of BOM structure are available in the *MSP432P4xx Family Technical Reference Manual*.

BOM can be setup by the user using any regular flash write/program approaches followed during software development.

Copyright © 2015–2016, Texas Instruments Incorporated

Table 1 lists the different types of boot-override commands that can be requested by the user to the device boot-code.

Table 1. Boot Override Commands

Boot Override Command	Description
JTAG and SWD Lock	This command is used to secure the application code from debug through JTAG or Serial Wire Debug (SWD, 2-wire JTAG debug) mechanisms. This is similar to JTAG and SBW lock available on the TI MSP430 product families.
Secure Zone X Enable (X = 0, 1, 2, 3)	These commands are used to configure and secure the software IPs that the user wishes to create. Code executing in these secure zones has all the characteristics mentioned in Section 4.
Secure Zone X Update (X = 0, 1, 2, 3)	These commands are used to perform an in-field update to software IP already secured using the "Secure Zone X Enable" command.
JTAG and SWD Lock Encrypted Update	This command is used to perform an encrypted update to any application code in the memory of the device that is **not** protected by "Secure Zone X Enable".
Bootstrap Loader (BSL) Configuration	This command can be used to enable a "custom BSL" or to enable the hardware invocation configurations for the TI BSL.
Factory Reset	This command erases all application code from the flash main memory of the system and removes all device security protections **after** the flash main memory has been successfully erased.
Factory Reset Configuration	This command can be used to configure the Factory Reset command. The configurations supported are: • Protecting Factory Reset using a password • Disabling Factory Reset

This application note focuses on the configuration procedures for enabling IP protection and also lists out steps for performing secure in-field updates to these IP protected zones (that is, the options not greyed out in Table 1). The greyed out fields of the table are described in *Configuring Security Features and Bootstrap Loader (BSL) on MSP432™ Microcontrollers*.

Figure 1 shows the flowchart of how to invoke boot override in MSP432P4xx devices.

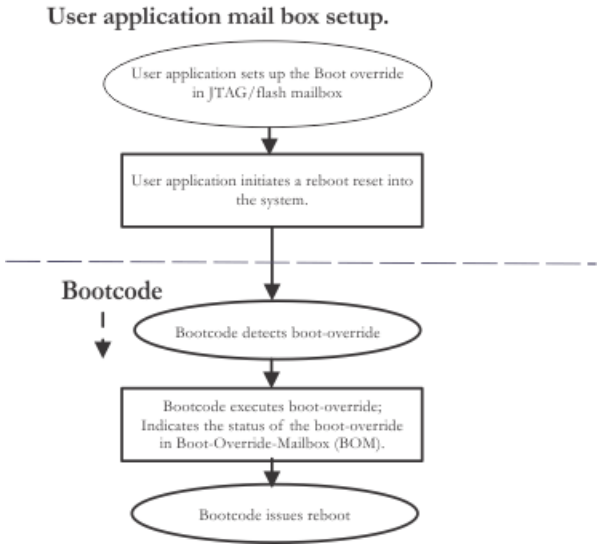

Figure 1. Invoking Boot Override

Copyright © 2015–2016, Texas Instruments Incorporated

4 IP Protection in MSP432P4xx Devices

MSP432P4xx devices provide for up to four different configurable IP protected zones. These IP protected zones:

- Prevent device debug when software IP is being executed.
- Allow JTAG/SWD based device debug to other regions in the device memory map when execution is outside the software IP memory zone.
- Prevent read out of software IP execution memory zone from regions of the memory map which are outside the zone defined for software IP.
- Can be configured to match the software IP size.
- Has many configurable options and parameters that are described in Section 7.

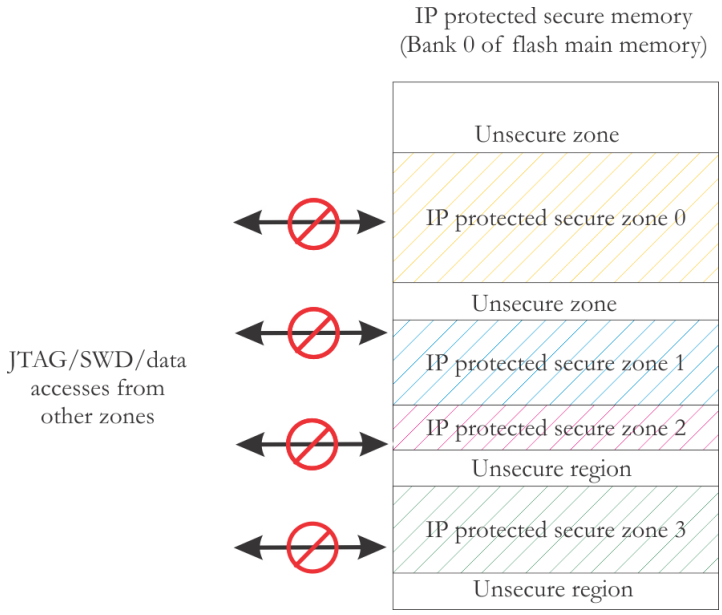

Figure 2. Software IP Protection Using MSP432P4xx Devices

Configuring the device for IP protection is performed by setting up a boot-override in the BOM.

The device boot-code copies over the contents of the BOM and then sets the device up for IP protection in the required memory areas.

5 In-Field Updates

MSP432P4xx devices allow for in-field updates to the code stored in any IP protected secure zone.

Data/code to be updated (payload) to the IP protected secure zone should be available in a free memory region (not IP protected) in bank 1 of the flash main memory of the device before the in-field update is invoked. This can be performed using the BSL present in the device (TI or custom). BSL cannot be used to directly update the contents of the IP protected zone since BSL (code outside IP protected secure zone) would not be able to perform read or write operations on the IP protected secure zone.

Enabling in-field updates to a secure zone is similar to configuring an IP protected secure zone in the device. User needs to configure the BOM with the appropriate commands and parameters and issue a boot-override to the system. The parameters for in-field update are describes in Section 7.

The device boot-code then updates the IP protected secure zone. The device boot-code uses either authentication, or, decryption and authentication to determine the authenticity of the payload being updated to the IP protected secure zone. Authentication or, decryption and authentication approach is chosen by the device boot-code based on the settings defined for the IP protected secure zone. This needs to be defined by the user at the time of securing the IP protected secure zone.

Copyright © 2015–2016, Texas Instruments Incorporated

A payload setup phase is needed before it is transported to a free flash memory region within the device. Figure 3 and Figure 4 show the payload setup phases for the different types of in-field updates possible to an IP protected secure zone.

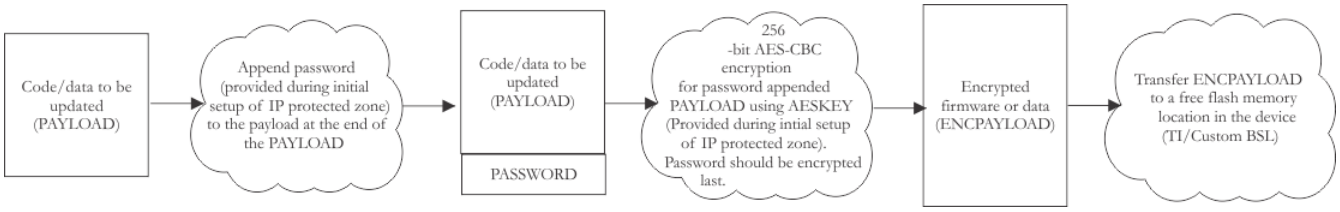

Figure 3. Payload Setup Phase for Device Boot-Code Decrypted and Authenticated In-Field Update to IP Protected Secure Zone

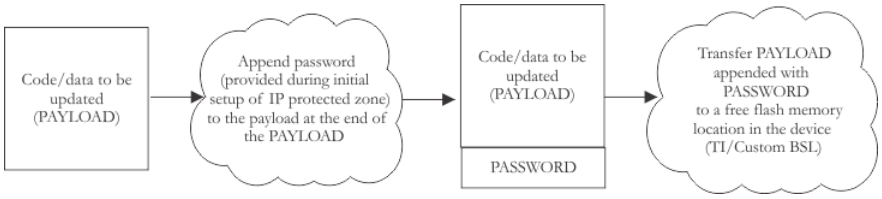

Figure 4. Payload Setup for Device Boot-Code Authenticated In-Field Update to IP Protected Secure Zone

6 IP Protection Zone Setup Configuration Parameters

MSP432P4xx family of microcontrollers provides users with different options when setting up the Software IP protection. Table 2 describes the different parameters available in the BOM.

Table 2. IP Protection BOM Parameters

BOM Parameter	Description
Secure Zone Start Address	This is the start address of the memory area where the software IP is located. This should be in bank 0 of flash main memory and aligned to the flash sector size. (Aligned to 4KB for the MSP432P401x MCU.)
Secure Zone Length	Length of the secure zone configurable by the user. This should be in multiples of the flash sector size. (Multiples of 4KB for the MSP432P401x MCU.)
Secure Zone Data Enable	This field is used to enable/disable data reads from the memory area starting with "Secure Zone Start Address" and having a length of "Secure Zone Length".If this field is marked as **enabled**, data reads in this area of memory is allowed **only for code residing within the same area**. Code residing in this area has to follow the procedure mentioned in the SYSCTL chapter of TRM to perform data reads within this secure zone. Data reads to this area are **not** allowed for **any** code if this field is marked as **disabled** when setting up the IP protected secure zone. Data reads to this area are **never** allowed for codes residing outside this secure zone.
Secure Zone Encrypted Update Enable	This field allows the user to choose the security level for in-field updates of an IP protected secure zone. If enabled, the user will need to provide an AES-CBC encrypted payload to the device boot-code whenever an in-field update to the secure zone is needed. If disabled, the user must provide a password appended nonencrypted payload to the device boot-code whenever an in-field update to the secure zone is needed. To provide additional security, this field has to be selected by the user at the time of setting up the secure zone and cannot be changed later.
Secure Zone Unencrypted Password	This field allows the user to choose a 128-bit password used to authenticate the validity of an in-field update. This password is defined by the user at the time of setting up the secure zone. This same password should be appended to the new code and presented to the device boot-code whenever an in-field update is needed to the secure zone. This password should be kept confidential by the owner of the content of the secure zone.
Secure Zone AES-CBC initialization vectors	This 128-bit quantity is used as the initialization vector for the AES-CBC decryption. This is used by the device boot-code to decrypt an encrypted payload received for in-field update to the secure zone. This initialization vector should be kept confidential by the owner of the content of the secure zone.
Secure Zone AES-CBC Security Key	This 256-bit quantity is used as the security key for the AES-CBC decryption. This is used by the device boot-code to decrypt an encrypted payload received for in-field update to the secure zone. This security key should be kept confidential by the owner of the content of the secure zone.
Secure Zone Enable	This field is used to enable a secure zone. This field has be set to ENABLE (0x0000:0000) in the BOM before a reboot reset is issued to the device to instruct the device boot-code to set up the secure zone.

Copyright © 2015–2016, Texas Instruments Incorporated

Table 2. IP Protection BOM Parameters (continued)

BOM Parameter	Description
Acknowledgment	This is the field where the device boot-code indicates the status for the IP protection boot-override command. This field should be left in the erased flash state of 0xFFFF:FFFF by the user. After the boot-override is complete, this field is updated by the boot-code to reflect SUCCESS (0x0000:0ACE) or ERROR (0x0000:DEAD)

7 IP Protected Zone In-Field Update Parameters

The parameters for the in-field updates as specified in the BOM are described in Table 3.

Table 3. BOM Parameters for In-Field Updates to IP Protected Secure Zones

BOM Parameter	Description
Secure Zone X Payload Address	This is the start address of the memory area where payload to be updated to the IP protected secure zone is located. This should be in Bank 1 of the main flash memory.
Secure Zone X Payload Length	This is the length of the payload in bytes. This should be equal to "Secure Zone X Length" + 128-bit password.
Acknowledgment	This is the field where the device boot-code indicates the status for the IP protection boot-override command. This field should be left in the erased flash state of 0xFFFF:FFFF by the user. After the boot-override is complete, this field is updated by the boot-code to reflect SUCCESS (0x0000:0ACE) or ERROR (0x0000:DEAD)

8 Data Unlock When Executing in IP Protected Secure Zone

Section 6 describes the parameters to be used when securing an IP protected secure zone. Of all the parameters, "Secure Zone Data Enable" is of special interest in this section. As described, data access to an IP protected secure zone can be enabled for the code executing within the secure zone using this parameter.

Enabling this parameter is a necessary condition to enable data access. This however is not a sufficient condition. The code executing in the secure zone also needs to unlock data access intentionally. This can be done by writing an unlock key (0x0000695A) to SYS_SECDATA_UNLOCK register in the SYSCTL space. This is illustrated in Figure 5.

This is done to ensure that a non IP protected code cannot access data in an IP protected secure zone by calling a function in the parts of the secure zone where data reads are allowed.

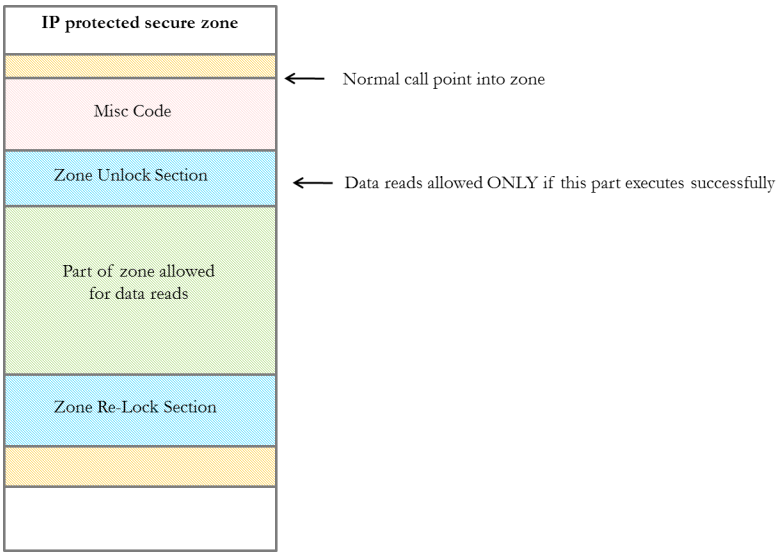

Figure 5. Data Access Unlock for IP Protected Secure Zones

Copyright © 2015–2016, Texas Instruments Incorporated

9 Examples

9.1 Example 1: IP Protection Setup

The code example in Figure 6 assumes blink LED software IP. This section discusses the step-by-step approach to enable IP protection for function softwareIP0() in Figure 6.

```
#include "msp432.h"

// Software IP to be protected
void softwareIP0(void);

// Main code
int main(void)
{
  WDTCTL = WDTPW | WDTHOLD;          // Stop watchdog timer

  softwareIP0();                            // Call software IP

  return 0;
}


#pragma CODE_SECTION(softwareIP0,".sec_zone0")
void softwareIP0(void)
{
        volatile uint32_t i, j ;             // volatile is used here otherwise code
optimizations remove
                                                                  // for loop delay in-
side the while loop.
        j= 10;
        // The following code toggles P1.0 port
        P1DIR |= BIT0;                         // Configure P1.0 as output
        while(j>0)
        {
          P1OUT ^= BIT0;                       // Toggle P1.0
          for(i=100000; i>0; i--);         // Delay
          j--;
        }
}
```

Figure 6. Code Snippet for Software IP Protection

The first thing to consider when securing software IP is to link the function(s) in a separate memory section. This memory section should have the following properties:

- Aligned to the start address of a flash sector
- Length of the memory section should be a multiple of the flash sector size.
- Memory section should be in flash bank 0. This is to prevent the DMA access to the software IP (1)
- Placing the memory section in the same flash sector along with interrupt vectors should be avoided if use of TI BSL is desired. (2)

In the example above, softwareIP0() is compiled at a flash sector started at address 0x0000:4000 (flash bank 0, sector 4) and has a length of 4KB. This satisfies all the four properties listed above.

To protect softwareIP0() using IP protection we now need to setup the BOM in the flash information memory. Here we will setup Secure Zone 0 using BOM. The BOM for this boot-override command will be as in Table 4.

(1) DMA access to bank 0 of the flash is not allowed when IP protection is enabled in the device.

(2) TI BSL uses the device interrupt vector table as the password. Hence BSL, password check will fail when IP protection is enabled on flash bank 0 sector 0, since BSL (code outside of IP protected zone) will not be able to read the interrupt vector locations (IP protected zone).

Copyright © 2015–2016, Texas Instruments Incorporated

Table 4. BOM for IP Protection Secure Zone 0 Setup

BOM Address Offset	BOM Field	Value
0x0	Mailbox Start	0x0115:ACF6 (Mailbox start marker value mandatory)
0x4	Mailbox Command	0x0010:0000 (Secure Zone 0 boot-override command)
0x60	Secure Zone 0 Enable	0x0000:0000 (Enable Secure Zone 0)
0x64	Secure Zone 0 Start Address	0x0000:4000 (Start address where softwareIP0() is placed in memory)
0x68	Secure Zone 0 Length	0x0000:1000 (1 sector (4KB) reserved for softwareIP0() code)
0x6C-0x78	Secure Zone 0 AES-CBC initialization vectors	Confidential AES-CBC initialization vector which is to be used for AES-CBC decryption on an ENCPAYLOAD if Secure Zone 0 Encrypted Update Enable parameter is 0x0000:0000. 0xFFFF:FFFF if the Secure Zone 0 Encrypted Update Enable parameter is 0xFFFF:FFFF.
0x7C-0x98	Secure Zone 0 AES-CBC Security Key	Confidential AES-CBC security key which is to be used for AES-CBC decryption on an ENCPAYLOAD if Secure Zone 0 Encrypted Update Enable parameter is 0x0000:0000. 0xFFFF:FFFF if the Secure Zone 0 Encrypted Update Enable parameter is 0xFFFF:FFFF.
0x9C-0xA8	Secure Zone 0 Unencrypted Password	Confidential PASSWORD field to be used for (ENC)PAYLOAD authentication during an in-field update to Secure Zone 0. In this example, 0x12345678123456781234567812345678 is used as the 128-bit password.
0xAC	Secure Zone 0 Encrypted Update Enable	0xFFFF:FFFF if encryption is not-desired for in-field updates. 0x0000:0000 if encryption is desired for in-field updates.
0xB0	Secure Zone 0 Data Enable	0xFFFF:FFFF since softwareIP0() does not have any data access to flash in the function.
0xB4	Acknowledgment	0xFFFF:FFFF
0x280	Mailbox End	0x0011:E11D (mail box end character)
All other memory locations in mailbox up to 0x280	Don't care for this boot-override command	0xFFFF:FFFF

After the BOM setup, the device needs to be issued with a reboot reset. This can be performed by writing a value of 0x0000:6901 (WKEY | REBOOT) into the SYS_REBOOT_CTL register in the device.

The device boot-code now sees a boot-override command and performs necessary action and updates the BOM with the status of the boot-override in the Acknowledgment field (Address offset 0xB4). After this, the device boot-code hands control back to the application. A value of 0x0000:0ACE in the acknowledgment field indicates that the software IP protection is now applicable on softwareIP0() function and the user would now not be able to halt the CPU and perform regular debug in the function softwareIP0(). Also, memory browser view through the debugger for the address 0x0000:4000 (softwareIP0() address) will return back errors.

Copyright © 2015–2016, Texas Instruments Incorporated

Figure 7 shows the picture of an active debug session before the IP protection is enabled on softwareIP0(). We can see the single stepping being done on this code and the memory window reads the contents of the memory where the function is loaded.

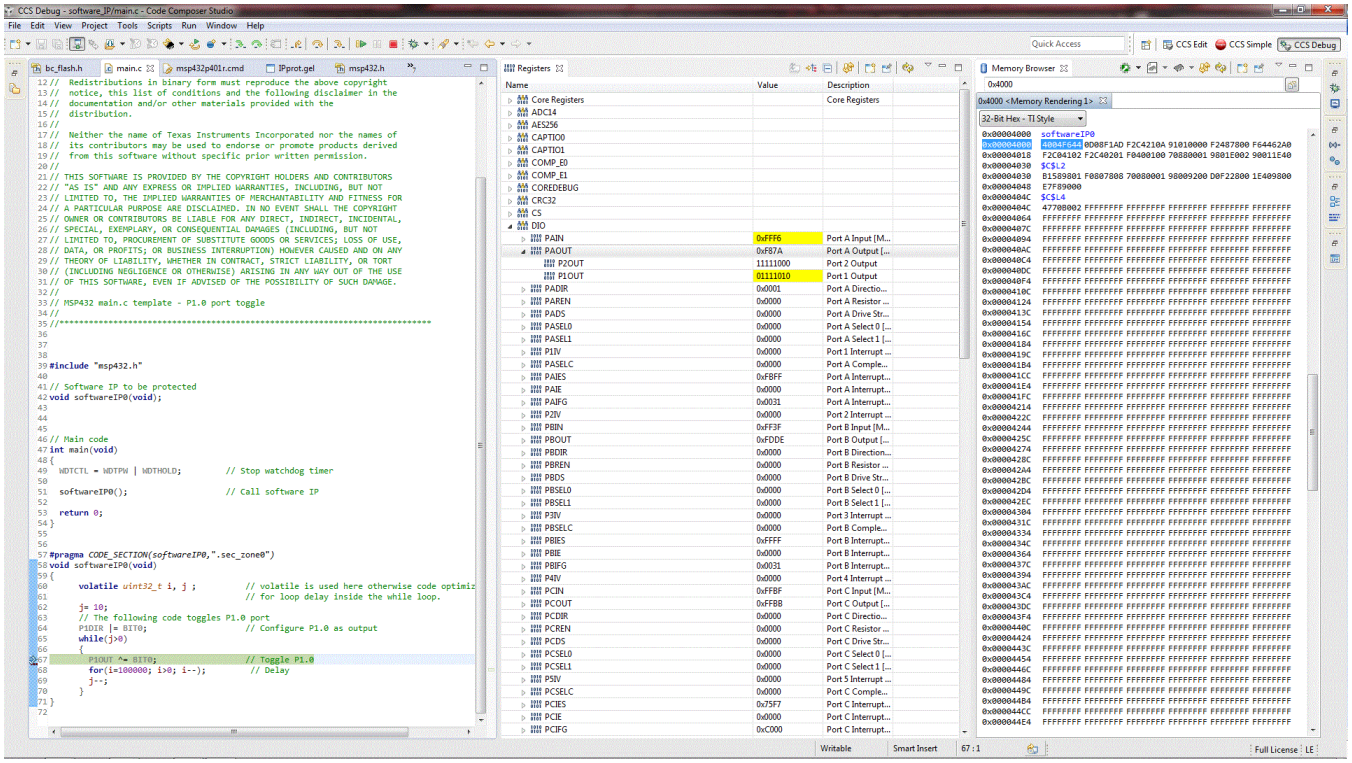

Figure 7. Single Stepping Inside softwareIP0() Before the IP Protection is Enabled

Copyright © 2015–2016, Texas Instruments Incorporated

Figure 8 shows the BOM memory window after completion of the boot-override for securing the softwareIP0(). The figure shows that the successful completion of the boot-override indicated by the boot-code back to the user at the Acknowledgment location 0x0020:00B4.

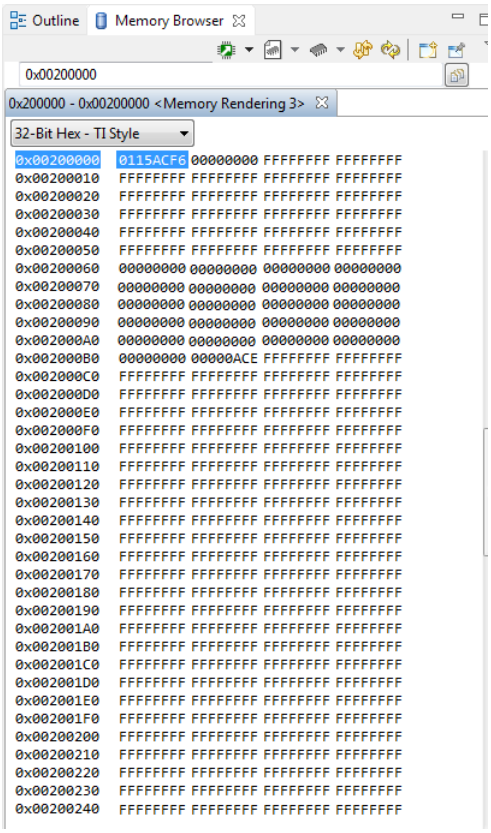

Figure 8. BOM Setup for Securing softwareIP0()

Copyright © 2015–2016, Texas Instruments Incorporated

Figure 9 shows the initial error from the debugger when a user tries to debug into softwareIP0(). Also, the memory window reads to the location corresponding to the function does not return any value. After this step, the debugger disconnects the user from the active debug session (see Figure 10).

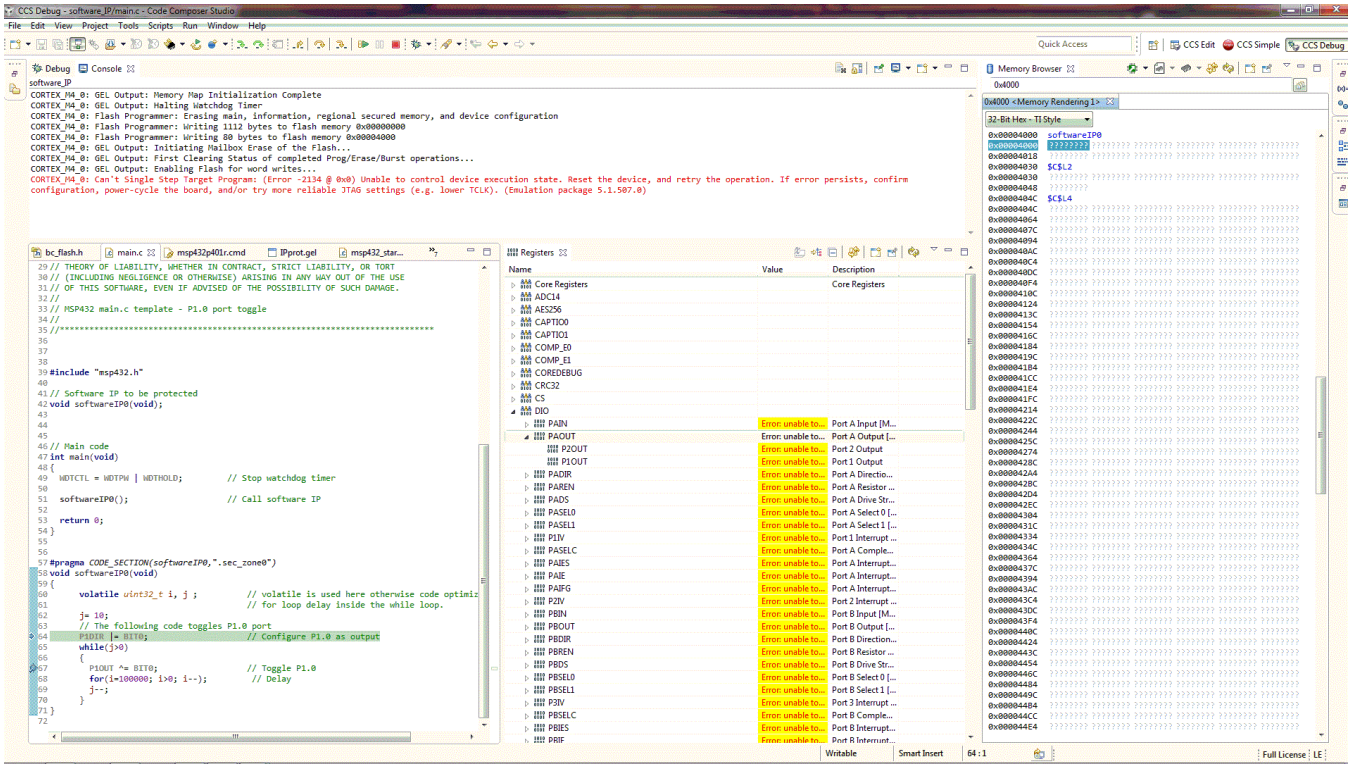

Figure 9. Error When Single Stepping Inside an IP Protected Code

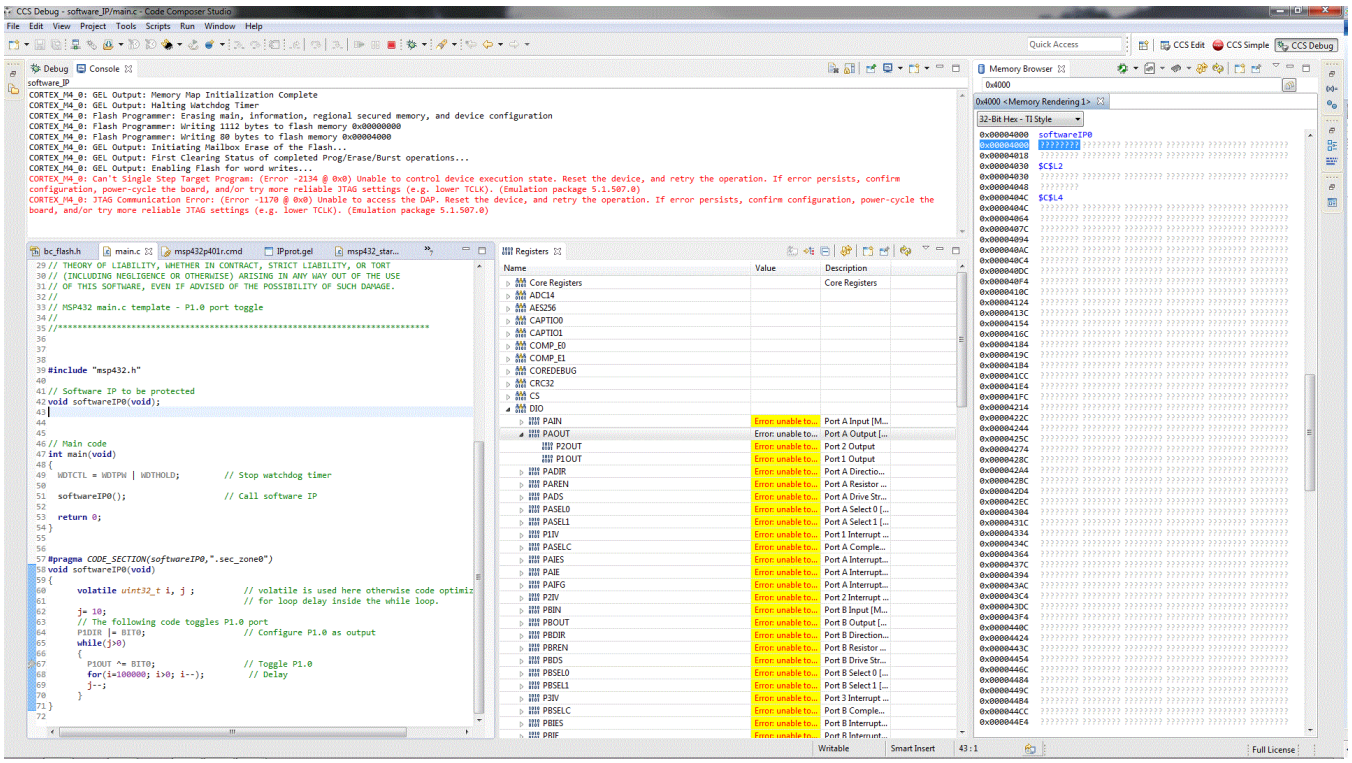

Figure 10. Disconnection From Debugger After Single Stepping in the Secure Zone

Copyright © 2015–2016, Texas Instruments Incorporated

9.2 Example 2: In-Field Update to an IP Protected Secure Zone

In this example, we assume that the device is already in the field with the code mentioned in "Example 1: IP Protection setup". Now, we need to update the softwareIP0() function with the new definition for the function given in Figure 11.

```
#include "msp432.h"

// Software IP to be protected
void softwareIP0(void);

// Main code
int main(void)
{
  WDTCTL = WDTPW | WDTHOLD;          // Stop watchdog timer

  softwareIP0();                            // Call software IP

  return 0;
}

// The following code toggles P2.0 port and is an update to the earlier softwareIP0
// function which toggled P1.0 port.
#pragma CODE_SECTION(softwareIP0,".sec_zone0")
void softwareIP0(void)
{
        volatile uint32_t i, j ;             // volatile is used here otherwise code
                                             // optimizations remove for loop delay
                                             // inside the while loop.

        j= 10;
        // The following code toggles P1.0 port
        P2DIR |= BIT0;                       // Configure P2.0 as output
        while(j>0)
        {
          P2OUT ^= BIT0;                     // Toggle P2.0
          for(i=100000; i>0; i--);           // Delay
          j--;
        }
}
```

Figure 11. Code Snippet for In-Field Update of Example 1 Code

softwareIP0() in example 1 is located at address 0x0000:4000. In this example, the above code is compiled in the same addresses as example 1. After the compile, 4KB (1 flash sector: IP protected secure zone 0 size in example 1) of data from address 0x0000:4000 (Start of IP protected secure zone 0 in example 1) is extracted from the hex image. This is the payload which needs to be used for update to the IP protected secure zone 0. This payload is now appended with the password specified at the time of securing the IP protected secure zone 0. This is the field at address offset 0x9C-0xA8 in Table 4. This password appended payload is now transported into the device using the TI BSL to location starting at address 0x0000:8000.

After the payload is available in address 0x0000:8000 in the device, the BOM is setup as given in Table 5.

Table 5. BOM for In-Field Update to IP Protected Secure Zone 0 of Example 1

BOM Address Offset	BOM Field	Value
0x0	Mailbox Start	0x0115:ACF6 (Mailbox start marker value mandatory)
0x4	Mailbox Command	0x0100:0000 (Secure Zone 0 update boot-override command)
0x20C	Secure Zone 0 Payload Address	0x0000:8000 (Address where the password appended payload is loaded in the device).
0x210	Secure Zone 0 Payload Length	0x0000:1010 (Length of the password appended payload)
0x214	Acknowledgment	0xFFFF:FFFF
0x280	Mailbox End	0x0011:E11D (mail box end character)

Copyright © 2015–2016, Texas Instruments Incorporated

After the BOM setup, the device needs to be issued with a reboot reset. This can be performed by writing a value of 0x0000:6901 (WKEY | REBOOT) into the SYS_REBOOT_CTL register in the device.

The device boot-code now sees a boot-override command and performs necessary action and updates the BOM with the status of the boot-override in the Acknowledgment field (Address offset 0x214). After this, the device boot-code hands control back to the application. A value of 0x0000:0ACE in the acknowledgment field indicates that the software IP was successfully updated with the new one. The program should now toggle port P2.0 instead of the port P1.0 in example 1.

Figure 12 indicates the BOM status on a in-field update performed with incorrect password.

Figure 13 indicates the BOM status on an in-field update performed with the correct password.

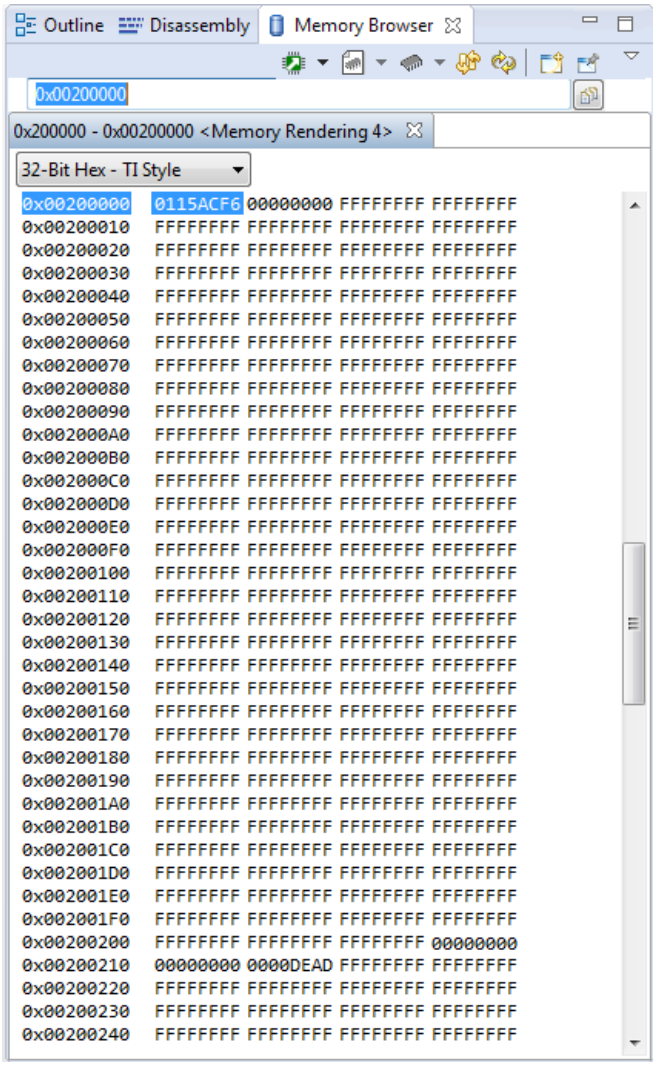

Figure 12. BOM Indicating Failed In-Field Update Due to Wrong Password

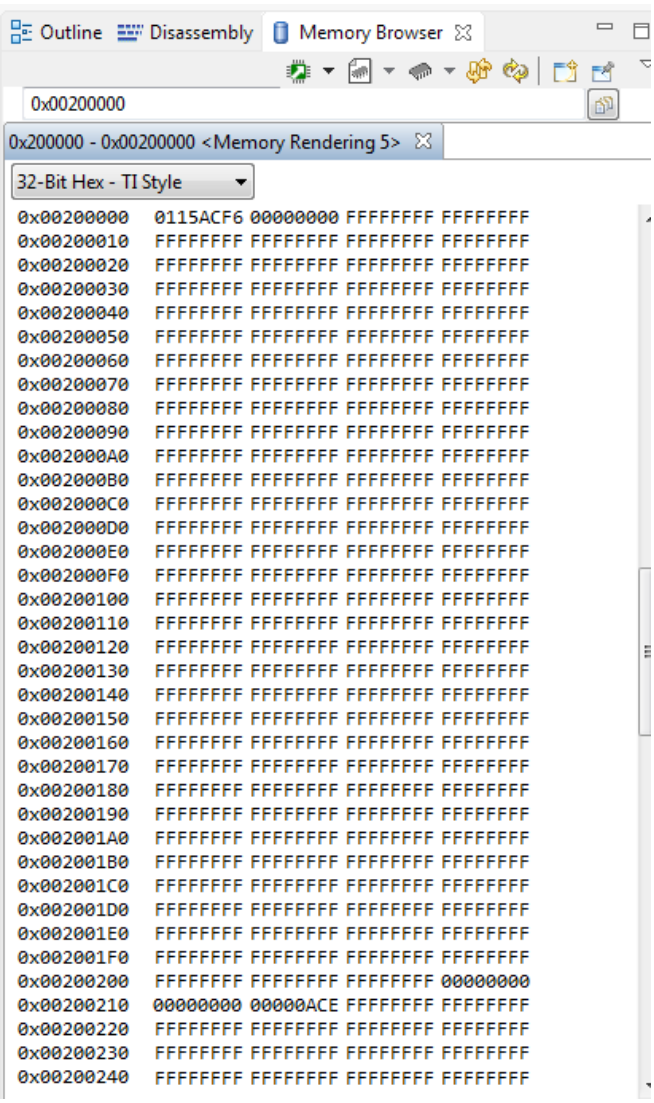

Figure 13. BOM Indicating Successful In-Field Update

9.3 Additional Examples

Additional example codes are provided as a part of MSPWare. These examples provide application developers with working code bases of setting up flash mailbox. Refer to the example project *FlashMailBox* for details.

Copyright © 2015–2016, Texas Instruments Incorporated

10 Tips for Developing Application Running in IP Protected Secure Zones

10.1 Special Consideration: Accessing Data Within IP Protected Zone

Applications can enable or disable the data access to locations within the IP protected secure zone from code running in the secure zone. As previously discussed, this is done using the "Secure Zone X Data Enable" field in the BOM. This section pertains to data access of application running in IP protected zone.

Compilers when compiling code can potentially insert some literal constants in the code space that may be referenced by the code during program execution. For example, the Code Composer Studio™ (CCS) compiler puts constants CCONSTxx/CLxx (xx = numbers) in the code after compile.

These can cause issues in two scenarios:

- In a situation like Section 9.1, *Example 1: IP Protection Setup*, where the secure zone data access is disabled.
- When secure zone data access is enabled when setting up the IP protection, but data access was made to the constants before the code unlocked data access to the secure zone.

These situations would cause the code to start executing the bus fault handlers. Hence, the generation of these literal constant should be suppressed at the time of code compile. This is done by using the compiler option to disable producing these literal constants. Figure 14 shows the option when using the CCS compiler to suppress the generation of literal constants.

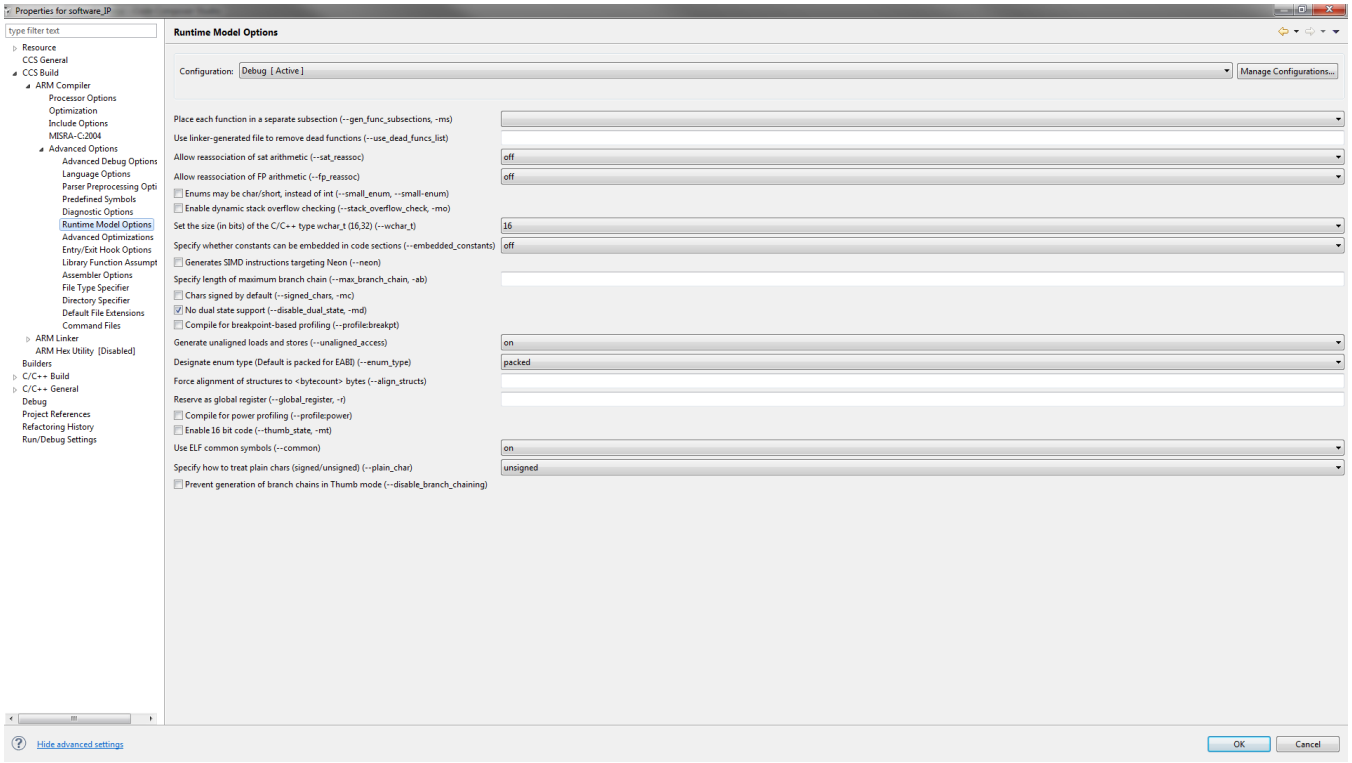

Figure 14. CCS Option to Suppress Literal Constant Generation

Copyright © 2015–2016, Texas Instruments Incorporated

10.2 Multiparty Software IP Development: Code and Data Security

MSP432P4xx devices support multiple software IP zones. These software IP zones can be configured by multiple parties who provide different parts of the system software. In this scenario, with the JTAG/SWD Lock not enabled, the IP protection framework in MSP432P4xx devices helps the software IP owners with support for code confidentiality and integrity, and data integrity for their software IPs. For securing software IPs that do not depend on literals or constant data, TI recommends disabling data access inside an IP protected secure zone using the "Secure Zone X Data Enable" field in the BOM (see Table 2 and Table 4).

For software IPs that need literals or constants in an IP protected secure zones for data integrity reasons, it is preferable to create separate IP protected zones for code and data, while keeping data access disabled in the code IP protected secure zone. The data IP protected zone should be enabled for data access. Additional functions should be written in the data IP protected secure zone region to unlock data access within this region, read the constant data, and return it to the code requiring this data.

10.3 Protecting the Contents in the BOM

So far, this document has described how to use the BOM if a boot-override is required. MSP432P4xx MCUs use plain text while communicating the BOM contents to the device. So for any system based on MSP432P4xx, if encryption or authentication of the BOM content is needed, TI recommends that the host handle the authenticated-and-encrypted communication with the network to which the system is connected. Figure 15 shows a high-level block diagram of this configuration.

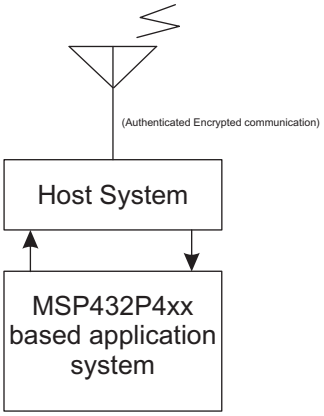

Figure 15. MSP432P4xx-Based Application System Together With a Host for External World Communication

Copyright © 2015–2016, Texas Instruments Incorporated

11 Summary

This application note describes and demonstrates:

- How to enable software IP protection on the MSP432P4xx family of microcontrollers

  and

- How to perform secure in-field updates to the software IP already configured in an IP protected secure zone.

The application note further provides some useful recommendations to application developers who intend to use the IP protection feature on microcontrollers in this device family.

12 References

1. *MSP432P4xx Family Technical Reference Manual*
2. *MSP432P401x Mixed-Signal Microcontrollers*
3. *Configuring Security Features and Bootstrap Loader (BSL) on MSP432™ Microcontrollers*

Copyright © 2015–2016, Texas Instruments Incorporated

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from June 8, 2016 to November 23, 2016 **Page**

- Added "in bank 1 of the flash main memory of the device" to the second paragraph in Section 5, *In-Field Updates* 4
- Changed the description of the Secure Zone Start Address in Table 2, *IP Protection BOM Parameters* 5
- Changed the description of Secure Zone X Payload Address in Table 3, *BOM Parameters for In-Field Updates to IP Protected Secure Zones* 6
- Changed the address of Mailbox End from 0x24C to 0x280 in Table 4, *BOM for IP Protection Secure Zone 0 Setup* 8
- Updated Figure 8, *BOM Setup for Securing softwareIP0()* for change of mailbox end to 0x280 10
- Changed the address of Mailbox End from 0x24C to 0x280 in Table 5, *BOM for In-Field Update to IP Protected Secure Zone 0 of Example 1* 12
- Removed the paragraph that started "After the successful in-field update..." in Section 9.2, *Example 2: In-Field Update to an IP Protected Secure Zone* 13
- Updated Figure 12, *BOM Indicating Failed In-Field Update Due to Wrong Password* for change of mailbox end to 0x280 13
- Updated Figure 13, *BOM Indicating Successful In-Field Update* for change of mailbox end to 0x280 13
- Removed former Section 10.1, *Interrupt Handling in IP Protected Secure Zone* 14
- Added Section 10.2, *Multiparty Software IP Development: Code and Data Security* 15
- Added Section 10.3, *Protecting the Contents in BOM* 15

Copyright © 2015–2016, Texas Instruments Incorporated

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated