

Moving From Evaluation to Production With SimpleLink™ MSP432P401x Microcontrollers

ABSTRACT

This application report is intended for SimpleLink™ MSP432™ microcontroller (MCU) developers currently evaluating the prerelease SimpleLink XMS432P401x Revision B and Revision C devices sampled from www.ti.com/product/msp432p401r. With the change to production silicon, TI recommends updating your device to the most recent revision of the MSP432 device.

NOTE: The SimpleLink MSP432 SDK can be used with only Revision C or later of the XMS432P401R or MSP432P401x devices.

There have been a number of functional enhancements to the MSP432 device since samples were first released as Rev B. This application report highlights the key differences between Rev B and newer revisions, and provides useful guidelines on necessary software changes for the production release (Rev C and beyond) devices including how to port to the SimpleLink MSP432 software development kit (SDK). For more information on the use of the MSP432P401x MCU features, see the device-specific data sheet and the [MSP432P4xx SimpleLink™ Microcontrollers Technical Reference Manual](#).

Contents

1	Introduction	2
2	Functional Improvements	2
3	Software Modifications	5
4	Development Ecosystem	8
5	References	8
6	Appendix	9

List of Figures

1	Version Comment in driverlib.h	9
2	About Code Composer Studio	9
3	Installation Details.....	10
4	Version Details.....	10

List of Tables

1	CPU Frequency and Flash Wait-State Requirements in Flash Normal Read Mode.....	3
2	Differences of MSP-EXP432P401R LaunchPad Onboard Debugger	8
3	MSP432P401x Development Tool Chain Minimum Versions.....	8

Trademarks

SimpleLink, MSP432, LaunchPad, BoosterPack are trademarks of Texas Instruments.
Bluetooth is a registered trademark of Bluetooth SIG.
 Wi-Fi is a registered trademark of Wi-Fi Alliance.
 All other trademarks are the property of their respective owners.

1 Introduction

The purpose of this application report is to highlight the key differences between the XMS432P401R Rev B and MSP432P401x Rev C and beyond and to ensure a smooth migration. It is especially intended for developers who are using XMS432P401x Rev B for evaluation and are now moving to the production released version MSP432P401x Rev C and future revisions.

You can find version information printed on the top of the MSP432P401x chip, or see the *Package Markings* section in the device-specific errata sheet. In addition to this application report, the device-specific errata sheet is another useful resource to understand hardware bugs and associated fixes when moving between revisions.

Developers who have used the XMS432P401x Rev B for evaluation also need to update software to reflect the new SimpleLink MSP432 SDK and perform an easy port from MSP432Ware software.

The MSP432 device is part of the SimpleLink microcontroller (MCU) platform, which consists of Wi-Fi®, Bluetooth® low energy, Sub-1 GHz, and host MCUs. All share a common, easy-to-use development environment with a single core software development kit (SDK) and rich tool set. A one-time integration of the SimpleLink platform lets you add any combination of devices from the portfolio into your design. The ultimate goal of the SimpleLink platform is to achieve 100 percent code reuse when your design requirements change. For more information, visit www.ti.com/simplelink.

This application report is divided into three sections:

1. Functional improvements (see [Section 2](#))
2. Required software changes (see [Section 3](#))
3. Updates to the development tool chain (see [Section 4](#))

2 Functional Improvements

2.1 Power Supply System (PSS)

On Rev B devices, two supply voltage supervisors were available – SVSMH and SVSL. Starting with Rev C, the monitoring function for V_{core} (SVSL) is now handled internally. This feature no longer requires external configuration and hence the corresponding functions have been removed from DriverLib. These functions include `PSS_setLowSidePerformanceMode`, `PSS_getLowSidePerformanceMode`, `PSS_disableLowSide`, and `PSS_enableLowSide`.

2.2 LPM3 and AM-LF Transition

User can now directly switch between LPM3 and LF-AM (Low-Frequency Active Mode); therefore, the required transition through AM-LDO mode (required on Rev B) is no longer needed. For more information, see the *Power Mode Transition* section of the *Power Control Manager (PCM)* chapter in the [MSP432P4xx SimpleLink™ Microcontrollers Technical Reference Manual](#).

Also see `msp432p401_pcm_12` in the [MSP432P401x code examples](#) or in the [SimpleLink MSP432 SDK Examples](#).

2.3 Flash

The default wait state value for flash was changed from 3 to 0, found in register `FLCTL_BANKn_RDCTL`; hence, users are now required to be aware of the dependency between CPU frequency (f_{MCLK}) and the flash wait-state setting.

Wait-state performance has been improved. Now you can use a flash wait-state value of 0 up to 16 MHz CPU frequency with V_{core} level 0, or up to 24 MHz with V_{core} level 1. This is an improvement over the previous 12 MHz with V_{core} level 0 or 16 MHz with V_{core} level 1 (see [Table 1](#)). Application software must first verify that the correct flash wait-state setting is configured before increasing the CPU frequency.

Table 1. CPU Frequency and Flash Wait-State Requirements in Flash Normal Read Mode

Parameter	Number of Flash Wait States	Maximum Supported MCLK Frequency				Unit
		AM_LDO_VCORE0, AM_DCDC_VCORE0		AM_LDO_VCORE1, AM_DCDC_VCORE1		
		Rev C Device	Rev B Device	Rev C Device	Rev B Device	
f _{MAX_NRM_FLWAIT0}	0	16	12	24	16	MHz
f _{MAX_NRM_FLWAIT1}	1	24	24	48	32	MHz

For more information, see the *Operating Mode Execution Frequency vs Flash Wait-State Requirements* section of the *Specifications* chapter in [MSP432P401x SimpleLink™ Mixed-Signal Microcontrollers](#).

2.4 Real-Time Clock (RTC)

In XMS432P401R Rev B, only PS1 channel interrupt can be used to wake up the device from LPM3. This channel has a minimum interrupt interval of approximately 15.6 ms (64 Hz). In MSP432P401R Rev C, the PS0 channel can also be used to wake up the device from LMP3 and provides an interrupt interval of 62.5 μs (16 kHz).

Users who require a shorter wakeup interval in LPM3 can take advantage of this functional modification.

Also see `msp432p401_rtc_04` in the [MSP432P401x code examples](#) or in the [SimpleLink MSP432 SDK Examples](#).

2.5 UART

A Rev C device can wake up and receive data from the UART in LPM3, by using UART RxD pin as a GPIO interrupt.

To wake up the UART in LPM3, set RxD as an input pin with high-to-low interrupt enabled and, in its interrupt service routine, set it as the RxD signal and start UART communication.

For a code example, see `msp432p401_euscia0_uart_09` in the [MSP432P401x code examples](#) or in the [SimpleLink MSP432 SDK Examples](#).

2.6 Device Security and Bootloader (BSL)

See [Configuring Security and Bootloader \(BSL\) on MSP432P4xx](#) for configuring MSP432P401x device security and BSL features. Users who do not plan to use any BSL or security features can skip this section.

2.6.1 Factory Reset

NOTE: A "factory reset" is different from a general "reset". A factory reset means to erase all the flash main memory and to remove all the security definitions in the system. While a reset that is generated from the reset pin, peripheral modules, software, or some other reset sources does not reset any nonvolatile memory.

2.6.1.1 BSL Factory Reset

A factory reset can now be performed using a BSL command; this is achieved using the flash mailbox boot override. This BSL command is not BSL-password protected. This method provides a factory reset when using BSL to update the device firmware.

2.6.1.2 Security Protection

In the preproduction Rev B devices, factory reset was an unprotected boot-override command. Anyone could initiate a factory reset and clear the device security infrastructure using boot override through either JTAG or flash mailbox.

In Rev C devices, factory reset behaves the same as in Rev B by default. However, it can be configured differently by using the following options:

- Option 1: Factory reset disable – This option disables the factory reset on the device; this means that the device is permanently locked from being factory reset.
- Option 2: Factory reset password enable – This option allows users to lock factory reset with a password. Only when the user provides the factory reset command, along with a correct password field, will the factory reset be successful.

NOTE: Do not confuse this password with the BSL-password protection mentioned in [Section 2.6.1.1](#). That section just means it is not password-protected by BSL command level, but for boot-override level through either JTAG or flash mailbox boot-override, the factory reset command is password protected. For details, see the *System Controller (SYCTL)* chapter in the *MSP432P4xx SimpleLink™ Microcontrollers Technical Reference Manual*.

2.6.2 Boot Override Commands

All boot override commands now clear the mailbox (that is, they set the value of the mailbox flash locations to 0) except for the ACK fields. You do not need to explicitly clear the mailbox.

2.6.3 Encrypted In-Field Update

When performing an in-the-field firmware update using an encrypted image, the image is decrypted by the device before being written to its destination location.

Now, after in-field update has been performed, but before control is handed back to the user code, the bootloader now cleans the decrypted image payload, which is generated by the device security infrastructure during an encrypted in-field update. This decreases the risk of hacking the decrypted payload.

See [Secure In-Field Firmware Updates for MSP MCUs](#) for more information.

2.6.4 Code Execution From SRAM

Code execution from SRAM space (0x20000000 to 0x2000FFFF) is now allowed when device security ("JTAG and SWD lock" or "IP protection") is not enabled. When "JTAG and SWD lock" or "IP protection" is enabled, code execution is only possible from the SRAM mirrored addresses range (0x01000000 to 0x0100FFFF).

2.6.5 BSL

The BSL factory reset command is described in [Section 2.6.1.1](#). This section addresses a few more updated BSL features.

In addition to the topics found here, also see the [MSP432™ SimpleLink™ Microcontrollers Bootloader \(BSL\) User's Guide](#) and the [Bootloader \(BSL\) for MSP low-power microcontrollers page](#) for more information.

2.6.5.1 Mass Erase Command

The mass erase command is now password protected. When using the mass erase feature in your BSL script, you are now required to send the RX Password command along with the correct password before sending the mass erase command.

2.6.5.2 SPI Speed

Now a higher baud rate is available when using the SPI interface for BSL. This benefits BSL in-field updates in both time and power saving.

2.6.5.3 Dealing With Time-Out

BSL time-out occurs when BSL is invoked, but no communication happens within 10 seconds. When this occurs, the BSL now sets all I/O pins to the GPIO input state without their pullup or pulldown resistors enabled.

Take note of this new feature and handle this change in an appropriate way.

3 Software Modifications

For developers moving from the Rev B platform, two steps are necessary to get running on production MSP432 silicon.

First, between Rev B and new revisions, there are many changes regarding the CMSIS support for the MSP432 MCUs. These improvements bring the header files into full CMSIS compliance. Additionally, a few updates were made to the Driver Library (also known as DriverLib) found in the MSP432Ware package.

Second, as of December 2016, MSP432Ware has been superseded by its new and improved successor, the SimpleLink MSP432 SDK. The SimpleLink MSP432 SDK offers seamless transition from MSP432Ware and introduces additional software components and improvements including TI Drivers with preintegrated TI-RTOS and POSIX-compliant OS kernels as well as flexibility to extend functionality through SDK plugins.

3.1 Moving From Rev B Software to the SimpleLink MSP432 SDK

TI recommends using the [SimpleLink MSP432 SDK](#) for all future software development for the SimpleLink MSP432 MCU. To port existing applications using the MSP432Ware software package, developers should follow these steps:

1. Update the software to be fully CMSIS compliant. This requires developers to port the application to use the latest MSP432Ware software (v3.50.xx.xx). This version can be found on the [MSP432Ware for MSP432 Microcontrollers page](#).
2. Move the application to use MSP432 Revision C or later silicon.
3. Migrate the to the SimpleLink MSP432 SDK by following the [SimpleLink MSP432 SDK Migration Guide](#).

3.1.1 SimpleLink MSP432 CMSIS Updates

Several changes have been made in the header file named msp432p401r.h and MSP432 DriverLib. Make sure to use header file msp432p401r.h of version 2015-10-26 or later. You can find the header file version at the top of header file comments. For developers who use their own software, see [Section 3.2](#). For developers who use DriverLib for development, see [Section 3.3](#).

For detailed information, see the device header file msp432p401r.h, which is located at:

- CCS: ccsv6\ccs_base\arm\include
- IAR: IAR Systems\Embedded Workbench 7.x\arm\inc\TexasInstruments
- Keil: Keil_v5\ARM\Pack\TexasInstruments\MSP432P4xx_DFP\2.x.x\Device\Include
- MSP432Ware: MSPWare_x_x_x_x\driverlib\inc

For help finding your MSPWare/DriverLib version, see [Section 6](#).

Visit the [MSP432 CMSIS Update wiki page](#) for a detailed description of whether or not this change affects your development.

The CMSIS improvements revolve around three system software changes:

- The fundamental change updates the MSP432 MCU header file structure to the CMSIS-Core version 4.2.0 or later. To this end, the CMSIS struct definitions of peripheral registers (and bit definitions) have been modified in the msp432p401r.h device header file (see [Section 3.2](#) for more details).
- The start-up file (which contains the interrupt vector table) has been updated.
 - Along with a slight name change `startup_msp432p401r_{ide}.c/s`, this file now adds the `SystemInit` function call.

- This file now includes a pragma to retain the interrupt vectors and prevent them from being optimized out of the program (because this code is not called directly by most programs). The retain feature was previously handled by the default linker command file.
- If you have not made any changes to the either the "startup" or "linker command" files, you should replace them with the updated versions. (It is easiest to find the new ones by creating a new project. If your IDE and MSP432Ware files are up-to-date, the latest ones are automatically added to the new project.)
- If you have edited either of these files, merge your changes into the new files.
- Addition of system_msp432p401r.c and .s files, which provides the definition of the *SystemInit* function call.
 - CMSIS defines the Systeminit() function to perform some rudimentary system initialization, such as setting up the main system clock and creating a few basic system #defines. The goal of including this file in the standard is to help users get started more quickly and easily.
 - If your code already configures every system and peripheral setting that is required, the addition of this file should not affect your code.
 - Be aware that this file defines the system clock rate, which may be different from the hardware default. This could affect your code if you do not explicitly set the clock rate in your own system code. Thankfully, CMSIS defines a system clock rate variable, which you can interrogate to verify the clock rate. (Of course, you can also review the Systeminit() source code, included in your project.)

3.2 Modifications to the Device Header Files

As described in [Section 3.1.1](#), the CMSIS definitions have been updated. Specifically, the register defines have been reformatted to `ModuleName[ModuleInstance]->RegisterName`. Additionally, to better follow industry standard practices, the bit-field structure elements (along with their Hungarian notation) was removed. Finally, most bit-field defines have been changed to follow the model `ModuleName_RegisterName_BitFieldName`. The following examples show how to write to CMSIS bit fields through register level or through bit-band area access, by using the new register and bit field defines:

- `ADC14->CTL0 |= ADC14_CTL0_ENC;`
- `BITBAND_PERI(ADC14->CTL0, ADC14_CTL0_ENC_OFS) = 1.`

You can use the code conversion tool, which is included in MSP432Ware, to update application code. For more information about these changes and the code conversion tool, see [Code Conversion Application for MSP432P401R](#) and the [MSP432 CMSIS Update wiki page](#).

3.3 DriverLib

Along with the following notes, DriverLib users should also be aware of the header file changes described in [Section 3.2](#). For more information, see the DriverLib v3.xx.xx.xx release notes.

NOTE: DriverLib in ROM has been enhanced in both efficiency and robustness. For users who need foundation functions with higher efficiency, lower power consumption and faster execution speed, it is recommended that you use DriverLib from ROM by adding "MAP_" before each DriverLib function call; for example, `MAP_PCM_gotoLPM3()`.

3.3.1 Macro of Module Base Address Changes

The base address and module instance names have been changed from `{PERIPHERAL_NAME}_MODULE` to `{PERIPHERAL_NAME}_BASE` per the new device header file definitions. For example, `TIMER_A0_MODULE` to `TIMER_A0_BASE`.

3.3.2 Power Supply System (PSS) API changes

Two new APIs, *PSS_enableForcedDCDCOperation* and *PSS_disableForcedDCDCOperation*, are available to switch the "Force DC-DC operation". In "Force DC-DC operation", the fail-safe mechanism forces to use DC-DC regulator as power supply instead of switching it to LDO, when the supply voltage falls below the minimum required for DCDC operation.

Additionally, per the functional improvements to PSS (see [Section 2.1](#)), the following functions have been removed from DriverLib: *PSS_setLowSidePerformanceMode*, *PSS_getLowSidePerformanceMode*, *PSS_disableLowSide* and *PSS_enableLowSide*.

3.3.3 Clock System (CS) API changes

The function *CS_setDCOExternalResistorCalibration*, now requires that the user pass DCO frequency range as a parameter, so as to indicate the different calibration values for the DCO external resistor.

For *CS_startHFXT*, *CS_startHFXTWithTimeout*, *CS_startLFXT* and *CS_startLFXTWithTimeout*, return value is now added to show if crystal start-up occurs correctly or not. These APIs all return true if crystal start-up occurs correctly, and return a false status on failure. While there are no changes to the API call or the parameters passed, users can now use this flag to monitor whether the crystal oscillator is functional.

3.3.4 Power Control Manager (PCM) API Changes

3.3.4.1 LPM4

Two APIs have been added: *PCM_gotoLPM4* and *PCM_gotoLPM4InterruptSafe*.

Both APIs can be used to enter LPM4, similar to using *PCM_gotoLPM3* and *PCM_gotoLPM3InterruptSafe* for LPM3 entry.

3.3.4.2 Nonblocking Mode

Nonblocking modes were added for setting core voltage, power mode and power state using the following APIs: *PCM_setCoreVoltageLevelNonBlocking*, *PCM_setPowerModeNonBlocking* and *PCM_setPowerStateNonBlocking*.

This nonblocking mode is similar to the blocking mode except that this is a one state transition; therefore, if the power mode request is busy when trying to change the power mode/core voltage level, the transition may not complete successfully. It is the user's responsibility to verify if the power mode, core voltage level, and power state transition was achieved successfully. See the *Power Control Manager (PCM)* chapter in the [MSP432P4xx SimpleLink™ Microcontrollers Technical Reference Manual](#).

3.3.5 I²C API Changes

The API *I2C_slaveSendNAK* was added for when the eUSCI module is used in I²C slave mode.

3.3.6 Port Mapping Controller (PMAP)

The PMAP DriverLib API was slightly modified by changing the enumerations for the port names. *PxMAP* definitions have been renamed to *PMAP_PxMAP*. For example, you would need to replace "P2MAP" with "PMAP_P2MAP" when configuring the Port 2 map register:

```
MAP_PMAP_configurePorts(
(  const uint8_t * ) myP2map,      // Pointer to port map setting (defined above)
  PMAP_P2MAP,                    // Which port to remap
  1,                              // Number of ports to configure
  PMAP_DISABLE_RECONFIGURATION  // Disallow reconfig of the port (until next reset)
);
```

4 Development Ecosystem

4.1 LaunchPad™ Development Kit

The MSP-EXP432P401R Rev 2.0 LaunchPad development kit uses production released Rev C devices and is available for purchase from <http://www.ti.com/tool/msp-exp432p401r>. The easiest way to distinguish a production-released LaunchPad development kit is by its color. The evaluation LaunchPad development kit (MSP-EXP432P401R Rev 1.0) was black, and the production release LaunchPad development kit (MSP-EXP432P401R Rev 2.0 and later) is red to indicate that Rev C devices are used.

In terms of the LaunchPad board design, the positions of the P1, P2, and reset buttons have been moved to better access them when a BoosterPack™ plug-in module is attached. In addition, the new LaunchPad development kit supports using the XDS110 debugger (emulator part of the LaunchPad development kit) for programming external devices. [Table 2](#) shows the differences of the onboard debugger.

Table 2. Differences of MSP-EXP432P401R LaunchPad Onboard Debugger

Revision	Onboard XDS110 to Program Onboard Device	External XDS110 to Program Onboard Device	Onboard XDS110 to Program External Device
1.0	Yes	Yes	No
2.0	Yes	Yes	Yes

4.2 Development Environment

Check your development tool chain to make sure that you have the following versions (or newer) as shown in [Table 3](#), when developing with MSP432 MCU Rev C and future devices:

Table 3. MSP432P401x Development Tool Chain Minimum Versions

Tool	Item	Version
Software development package	MSP432Ware	3.50.xx.xx
	SimpleLink MSP432 SDK	1.12.00.00
API	DriverLib	3.10.xx.xx
IDE and Compiler	CCS	7.1
	TI CGT	16.9.0.LTS
	Keil	5.22
	MSP432P4xx_DFP pack	2.2.1
	IAR EWARM	7.80.3

5 References

1. [MSP432P401x SimpleLink™ Mixed-Signal Microcontrollers](#)
2. [MSP432P4xx SimpleLink™ Microcontrollers Technical Reference Manual](#)
3. [MSP432P401R Device Erratasheet](#)
4. [Code Conversion Application for MSP432P401R](#)
5. [MSP432 CMSIS Update wiki page.](#)
6. [MSP432™ SimpleLink™ Microcontrollers Bootloader \(BSL\) User's Guide](#)
7. [Configuring Security and Bootloader \(BSL\) on MSP432P4xx](#)

6 Appendix

There are two ways to determine which version of MSPWare DriverLib is being used.

1. Look at the version number in the comment section of the driverlib.h file (see [Figure 1](#)).

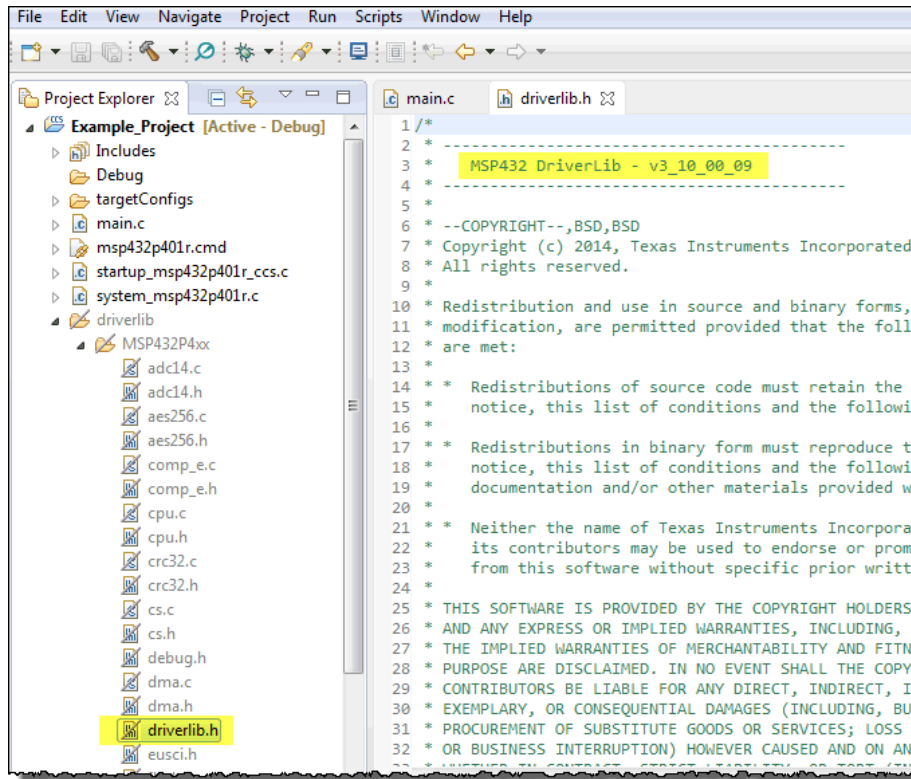


Figure 1. Version Comment in driverlib.h

2. CCS users discover the currently selected version of MSPWare/DriverLib by checking in the CCS **About** screen:

- (a) Select **About Code Composer Studio** from the **Help** menu (see [Figure 2](#)).

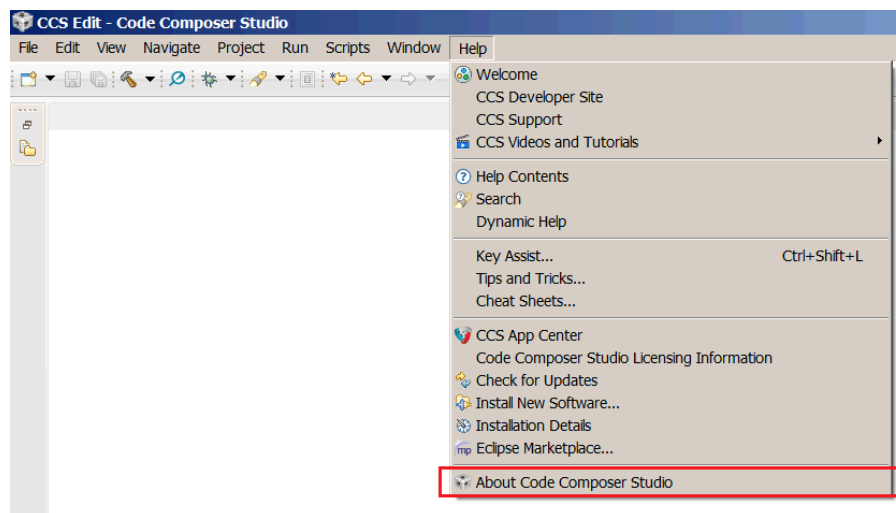


Figure 2. About Code Composer Studio

(b) Click **Installation Details** (see [Figure 3](#)).

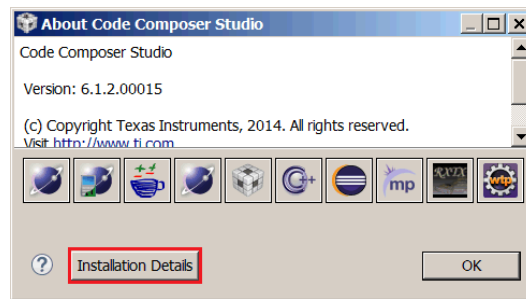


Figure 3. Installation Details

(c) Find the MSPWare and DriverLib versions (see [Figure 4](#)).

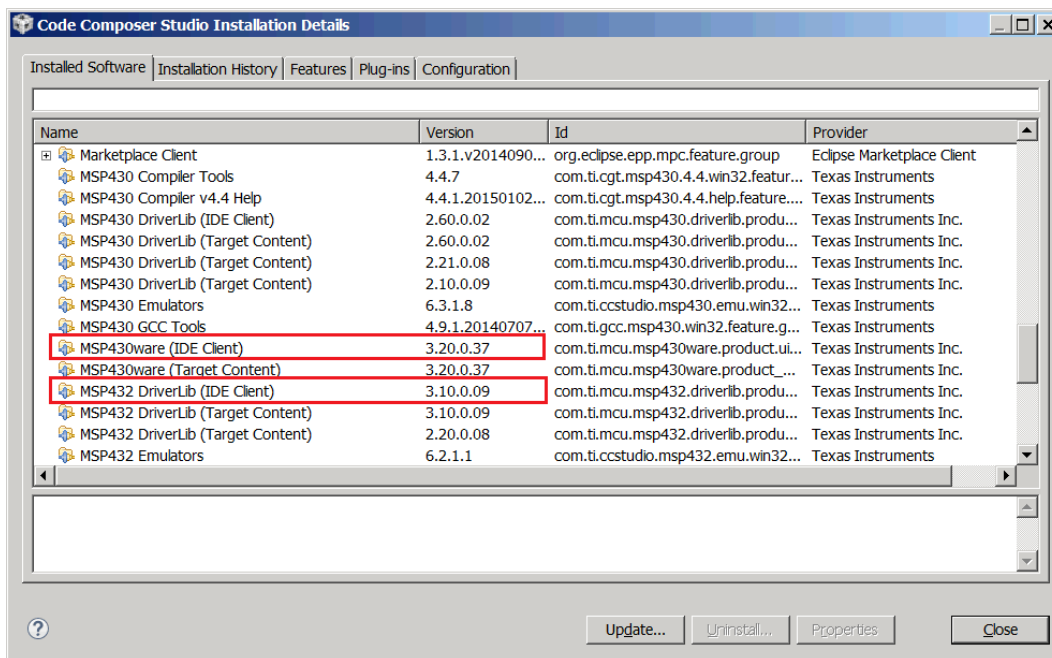


Figure 4. Version Details

Revision History

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

Changes from April 2, 2016 to March 6, 2017	Page
• Updated the abstract.....	1
• Added "SimpleLink" as necessary throughout document	1
• Updated Section 1 , <i>Introduction</i>	2
• Updated Section 3 , <i>Software Modifications</i> ; added information about moving from MSP432Ware to SimpleLink MSP432 SDK	5
• Updated all versions in Table 3 , <i>MSP432P401x Development Tool Chain Minimum Versions</i>	8

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2017, Texas Instruments Incorporated