Application Report
SCAA121A–September 2012–Revised December 2012

A Step-by-Step Guide on Using the MSP430 as a Bootloader for the CDCM6208VxEVM

Gabe Ayala, Matt Sunna, Julian Hagedorn *SVA/SDS/CTP*

ABSTRACT

This application note describes how to use the MSP430 bootloader feature from the CDCM6208VxEVM in an easy-to-follow, step-by-step guide.

Contents

List of Figures

1 Required Software and Hardware

In order to use the MSP430G2001 as a bootloader for the CDCM6208 the following items are required:

- MSP430 USB-Debug-Interface (MSP-FET430UIF)
- USB Cable (male A to male B)
- 14-pin JTAG cable
- Code Composer Studio v5
- Programming files (including .INI to .H file converter) (SLAC541)

2 Installing Code Composer Studio v5 (CCS)

If CCS is already installed, please proceed with the next step.

Code Composer Studio is available on TI's website for free – this is a code-size limited distribution designed to work ONLY with the MSP430. This download is available at the following URL: http://processors.wiki.ti.com/index.php/Download_CCS.

A login account is required to verify download availability on TI's servers. Once the .ZIP file is successfully downloaded and extracted, the installation can begin by opening the file named *setup_CCS_MC_Core_5.x.x.exe* where the *x* refers to a specific version number. The CCS install screen appears – proceed to the following screen which displays the EULA. Accepting the EULA then prompts for an installation path. By default this path is *C:\Program Files\Texas Instruments*.

In the Product Configuration Menu Screen, select the *MSP430-only Core Tools* option unless support for the C2000 is needed. The components page is next, where the choice of which components of the CCSv5 to install – make a *custom* installation and make sure the MSP430 USB FET option is selected in order to install the appropriate drivers. The following pictures help to select the right packages (Figure 1 through Figure 3). Otherwise, make a complete installation.

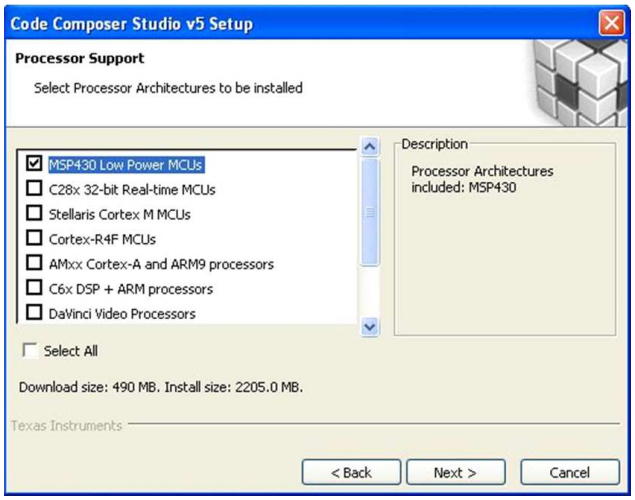

Figure 1. Processor Support Selection

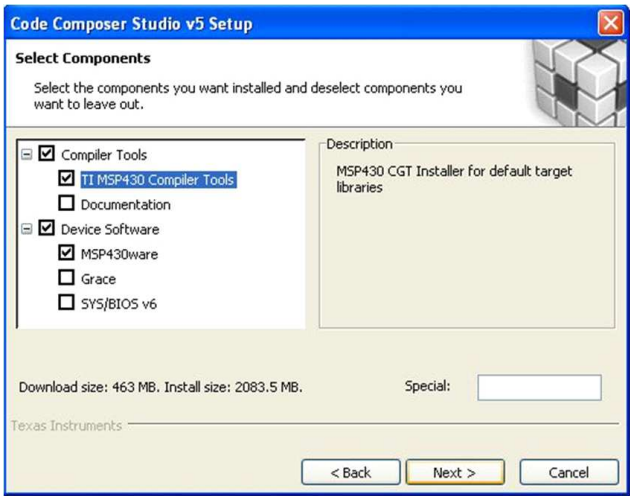

Figure 2. Component Selection

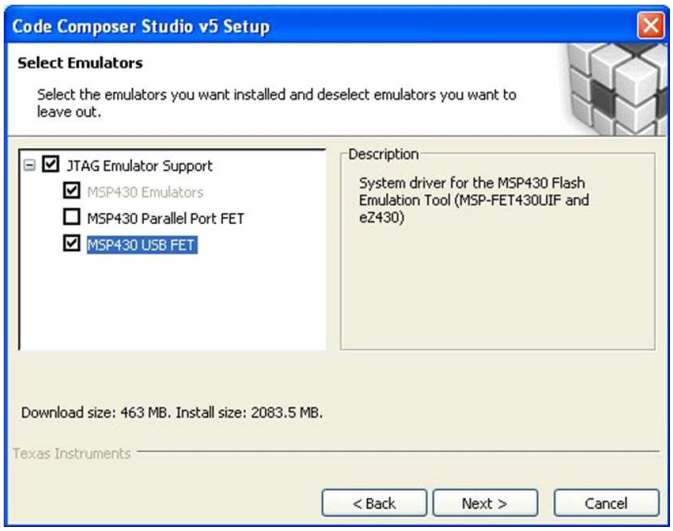

Figure 3. Installation: Driver Selection

3 Using Code Composer Studio v5

At the start of CCSv5, the default workspace has to be selected. By default this is *C:\Documents and Settings\<user name>\My Documents\workspace_v5_x*.

Selecting the check box makes this path the default workspace, this can be changed later. To create a new project navigate to File > New > CCS Project and type in a project name (for example, Bootloader Project CDC6208) and select the *MSP430Gxxx Family* and *Empty Project* as shown in Figure 4.

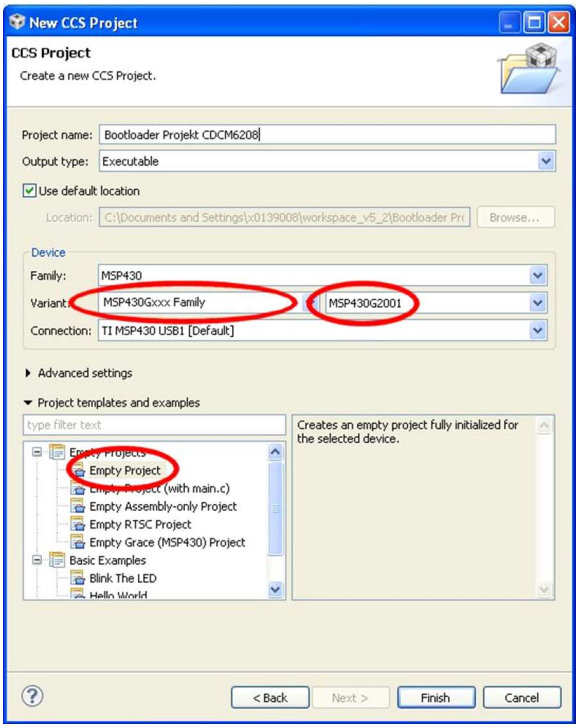

Figure 4. New Project Setup

4 Bootloader Environment Setup

Unzip the programming file (SLAC541).

Add the necessary files to the project using the menu *Project > Add Files to Active Project*. Navigate to the containing folder and add the following files: **spi_interface.c**, **spi_functions_G2001.h**, and **reg_val_header.h**. Note that the top-level file is **spi_interface.c** which depends on **spi_functions_G2001.h**, and **reg_val_header.h**.

Double-clicking the **spi_interface.c** in the C/C++ project window on the left reveals the source code structure.

Right click on the **spi_interface.c** and click *Properties*. Various compilation properties have to be set in order to successfully load the program onto the MSP430. Navigate to the Optimization menu (Figure 4) and set the *Optimization Level* to **4** and *Control Speed vs. size trade-offs* to **0**.

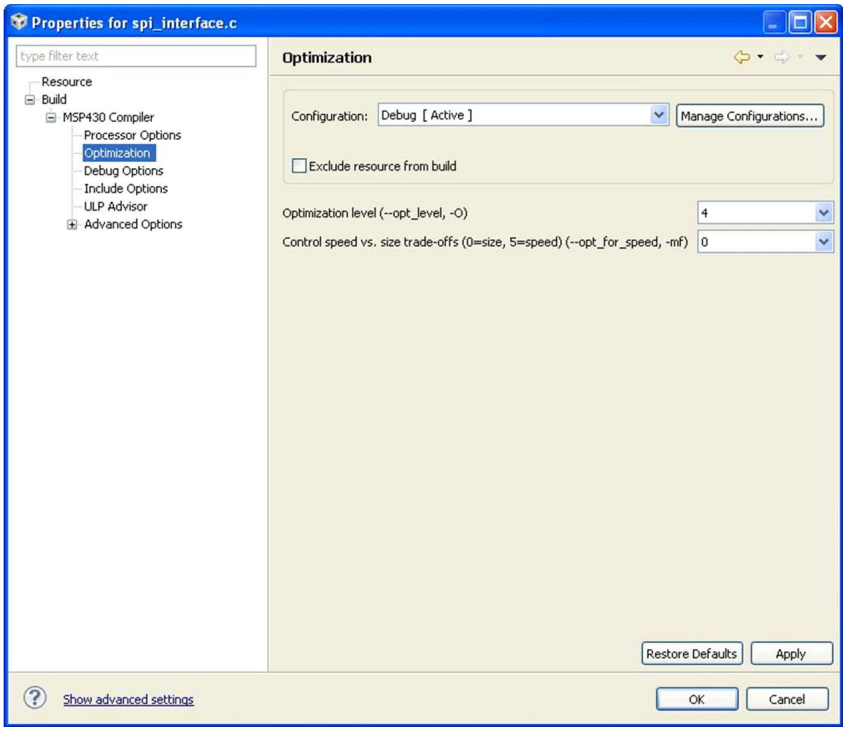

Figure 5. MSP430 Compiler Settings

At this point, the program is ready to be loaded to the board; however, the header file **reg_val_header.h** needs to be modified to reflect customized settings for the CDCM6208. If the register settings of the CDCM6208 are already known, it is possible to simply copy and paste those settings into the **reg_val_header.h**. That allows skipping step 5.

Otherwise, the .INI file can be converted to an .h-file by using the converter *CDCM6208_InitoH.exe*.

5 Creating a Readable Header

The included converter file is used to take an .INI file (which can be created by the CDCM6208_GUI) and convert it into a properly formatted .h file, readable by the MSP430G2001. To start the converter, execute the file named *CDCM6208_IniToH.exe*. The program opens the window shown in Figure 6.

Figure 6. CDCM6208_IniToH.exe Converter program

Click the **Set File** button in the *Input* section to select the desired .INI file to convert. Then press the **Set File** button in the Output section to select a desired output file. To create a seamless transition from the converter to CCSv5, simply overwrite the file **reg_val_header.h** in the project folder created in Section 4.

The default location is *C:\Documents and Settings\<user name>\My Documents\workspace_v5_x*.

To execute the conversion, press the **Convert File** button – a successful conversion is indicated with a pop-up prompt.

If a different configuration file name was chosen, ensure that the converted file is included the project. Also, only one formatted header file should be included in the top-level entity, **spi_interface.c**. Including more than one formatted header file may lead to compilation errors. To do this, simply replace the default line <2> #include "*reg_val_header.h*" with an updated name such as #include **"<desired_name>.h"**. Saving an .h file into the project folder automatically updates the project scope.

6 Program the Board

The final step in using the MSP430G2001 as a bootloader for the CDCM6208 is to enable SPI communication to the CDCM6208 as well as JTAG communication to the MSP430.

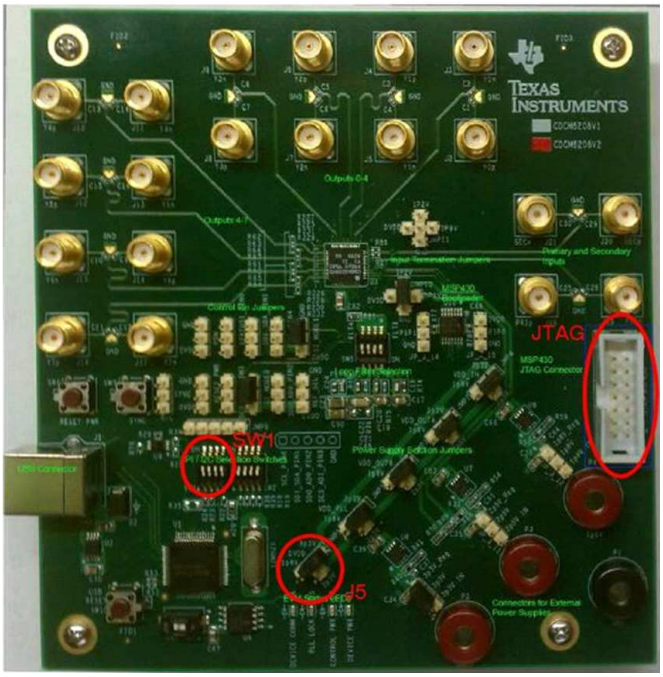

Figure 7. EVM Overview

Enable JTAG communication by switching the DVDD jumper (**J5**) to 3.3 V (Figure 7). JTAG programming requires a matching device voltage in order to communicate. However, the MSP430 can communicate to and program the CDCM6208 on a lower voltage. The MSP-FET430UIF should be connected to the PC as well as the EVM. Use the USB cable to connect the MSP-FET to the PC and the 14-pin JTAG cable to the EVM. The EVM must also be externally powered via USB or 3.3-V input.

To enable SPI communication from the MSP430G2001 to the CDCM6208, **SW1** and **SW2** must be in the OFF position while **SW3** must be in the ON position. **SW1** and **SW2** are located next to each other on the top layer (Figure 7).

SW3 is the only switch which is located on the bottom side of the board.

In CCSv5, programming the board is achieved by pressing the DEBUG button, shown in a red circle in Figure 8.

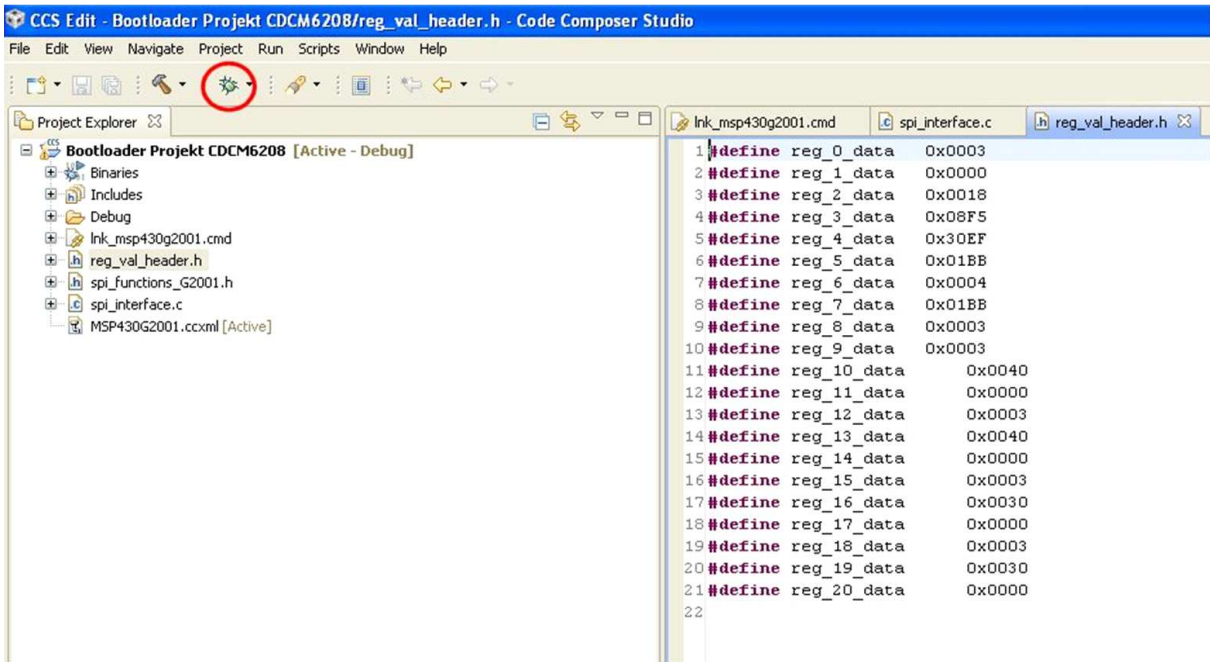

Figure 8. Debug Mode

To begin the program, press the RUN button  . The program is now on the MSP430 – to exit the debug mode, press the TERMINATE ALL  button.

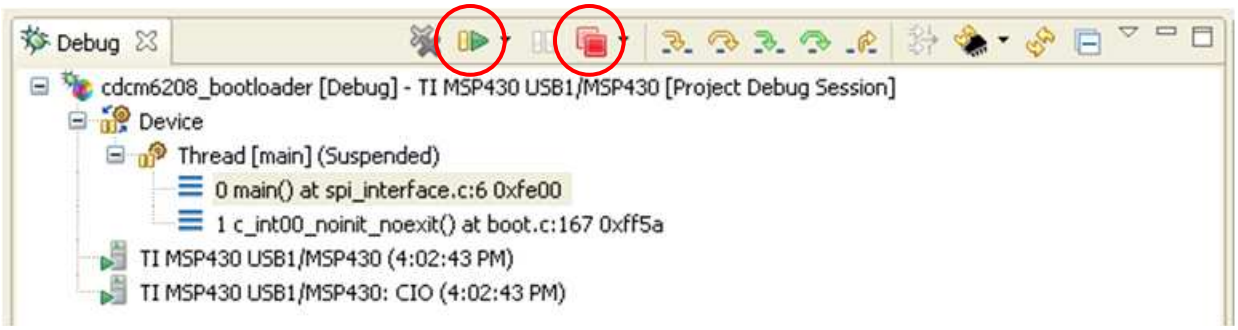

Figure 9. Run the Program

A progress information window appears showing the programming status – the window also changes from a program view to a debug view.

If the MSP430 USB-Debug-Interface does not have the right firmware, CCSv5 updates it at this point.

Now, the MSP430G2001 automatically programs the CDCM6208 on device power up. Additionally, SPI communications are limited between the MSP430 and the CDCM6208.

The JTAG board can be disconnected from the EVM. Additionally, DVDD can be set back from 3.3 V to 1.8 V, if needed.

NOTE: In order to communicate between the GUI and CDCM6208EVM again, **SW1** must be switched back to the ON position and **SW3** must be switched back to the OFF position.

References

Additional information about the CDCM6208 EVM is available in the EVM User Guide (SCAU049).

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