Fusion Digital Power Studio GUI for Isolated Power Applications

User's Guide

Literature Number: SLUA676C
March 2013–Revised June 2019

Contents

1 Trademarks **7**

2 About This User's Guide **7**

2.1 Introduction 7

2.2 The Fusion Digital Power Studio (Studio GUI) 7

2.3 The Device GUI 8

2.4 Conventions 8

2.5 User Interface Terminology and Tips 8

2.6 Terminology 8

2.7 Additional Technical Support 9

3 Getting Started **9**

3.1 PC Requirements 9

3.2 USB Adapter 9

3.3 Download and Installation 9

3.4 Upgrading the GUI 9

3.5 Multiple Installations of the GUI 9

4 Fusion Digital Power Studio (Studio GUI) **9**

4.1 Starting the GUI 9

4.2 Connecting to a Remote USB Adapter 11

4.2.1 Client Configuration 11

4.2.2 Remote Server Configuration 13

4.2.3 Client and Server Running 14

4.2.4 Connection Troubleshooting Tips 16

4.2.5 SETUP_ID in Firmware is Not Recognized by the GUI 16

4.2.6 Enable GUI Protected Features 18

4.2.7 Monitor 18

4.2.8 Configure 19

4.2.9 Design – Model Stage and Compensator 21

4.3 Status 31

4.4 Capturing the State of the Device - Saving a Project File 31

4.5 Miscellaneous Tools 32

4.5.1 Multi-image 32

4.5.2 Isolated Bitmask Tool 34

4.6 Offline Mode 35

4.6.1 Starting in Offline Mode 35

4.6.2 Open Existing Project File 36

4.6.3 Open Sample Project 36

5 Device GUI **37**

5.1 Launching Device GUI 37

5.2 Moving Between ROM and Program Mode 39

5.2.1 ROM Mode to Program Mode for Multiple Flashes 40

5.3 Firmware Download Tool 41

5.3.1 Boot Support 43

5.3.2 Data Flash Download 46

Copyright © 2013–2019, Texas Instruments Incorporated

5.4 Checksum Functions 47
5.5 Multi-image Functions 48
5.5.1 Setting Image Index 50
5.5.2 Multi-image Download 50
5.5.3 Switch 50
5.5.4 Image Peek/Poke/Dump 50
5.5.5 Erase Image 52
5.5.6 Export Image 53
5.5.7 Image Checksums 53
5.6 Isolated Bitmask Tool 58
5.7 Firmware Memory Debugger 61
5.8 SMBus Debug 69
5.9 CCS conversion 69
5.9.1 How to Access 70
5.9.2 Usage 70
5.10 Function Command Summary 71
5.11 Override Commands 72
6 Command Line Tools 73
6.1 FusionFirmwareDownload.exe 74
7 API – Application Programming Interface 75
8 Production Tool 75
9 Documentation and References 75
9.1 References 75
10 Revision History 76

Copyright © 2013–2019, Texas Instruments Incorporated

List of Figures

1 Quick Launch Shortcut 9
2 No USB Adapter Found 10
3 Device Scanning 10
4 No Devices Found 11
5 USB Adapter Mode Selector from Start Menu 12
6 Adapter Mode Selection 12
7 Remote Adapter Connection Test Success 13
8 USB Adapter Server from Start Menu 13
9 Remote Adapter Server 14
10 E-mail Invite with Remote Server Settings 14
11 Running a Remote Server Connected to a Client 15
12 Client Machine Conducting a Scan on the Remote Adapter 15
13 Client Machine Interacting with a Remote USB Adapter Connected to a Device 16
14 Select Change Device Scanning Options 17
15 Click UCD3XXX Isolated at Top 17
16 Start>Texas Instruments...>Special>UCD3XXX Isolated Fallback Device Scan Mode 17
17 GUI Preferences 18
18 GUI Protected Features 18
19 Monitor Mode Displays Some of the Live Parameters Being Read From the Device 19
20 Configure Mode 20
21 Displays the List of Commands the Firmware Supports 21
22 Design Mode Selected 22
23 Stage Parameters for HSFB for Voltage Loop (CLA #0) 23
24 Bode Plots 24
25 HSFB Schematic Being Modeled 25
26 Scroll Down the Stage Parameters to See the Compensator 25
27 Three Ways to Program the Compensator 26
28 Sample Offline Topologies 26
29 Coefficient Set and Alpha Configuration 27
30 Favorites 28
31 Coefficient Set and Alpha Summary 28
32 Apply Bin 0 to All Bins (Linear) 29
33 Symmetric and Non-Symmetric 30
34 Writing Loop Coefficients and Global Reset of GUI Edits to Hardware Coefficients 30
35 Status Mode 31
36 Save Project File 32
37 Multi-image 32
38 Downloading to an Image While Monitoring at the Same Time 33
39 Tools> Isolated GUI Bit Mask Generator 34
40 Isolated Bitmask Tool in Fusion Studio (Online) 35
41 Starting in Offline Mode 36
42 Offline Options 36
43 Offline Sample Topologies 36
44 Studio GUI Tools Menu 37
45 Opening UCD3xxx and UCD9xxx Device GUI 37
46 UCD3XXX Device GUI 38
47 Program Scan and Rom Scan 39

Copyright © 2013–2019, Texas Instruments Incorporated

48 Moving Between ROM Mode to Program Mode 40
49 Executing Program for Block 0 (0xF0) 40
50 Executing Program for Block 1 (0xF7) 41
51 Firmware Download 41
52 Firmware Download Screen for the UCD3138 42
53 Firmware Download for the UCD3138064 42
54 Firmware Download the UCD3138A64 43
55 Boot Support 44
56 Bootflash Options 44
57 Firmware Writing Options 45
58 Two Checksums for Boot Flash Greater Than 2 kB 45
59 Writing pflash Checksum Options 45
60 Data Flash Download Options 46
61 Checksum Functions 48
62 2 kB Boot Checksum Functions 48
63 2 kB+ Boot Checksum Functions When Boot is Greater Than 2 kB 48
64 Scan for Device ID to Activate Multi-image 49
65 Click Read Multi-image Parameters to Activate Functions 49
66 Functions Enabled After Reading Multi-image Parameters 49
67 Image Download 50
68 Image to Switch to 50
69 Image Peek/Poke 51
70 Image Dump 52
71 Erase Image 53
72 Calculate Image Checkum 54
73 Dump Image Checksum 55
74 Recreate Image Checksum 56
75 Validating Image Checksum 57
76 Clearing Image Checksum 58
77 Click Iso Bitmask Tool 58
78 Bitmask Tool 59
79 Memory Debugger 62
80 GUI Debugger 62
81 Fusion Studio GUI Debugger Tool 63
82 GUI UCD3138 Debugger – Defaults 63
83 Device Debugger Bit Field Selector 64
84 Watch List Selection Star 64
85 Map File Selection 64
86 Debugger Customization Tool 65
87 ".pp" Generation Parameters 66
88 Map Filename 67
89 Watch List with Firmware Variables 68
90 SMBus Debug Link 69
91 SMBus Debug 69
92 CCS Conversion Tool Access from Start Menu 70
93 Conversion Tool Access from Device GUI 70
94 Access to Override Commands 72
95 Four Commands Have Been Overridden 73
96 Override in Top Right Shown 73

Copyright © 2013–2019, Texas Instruments Incorporated

List of Tables

1 ROM/Program Commands 71

Copyright © 2013–2019, Texas Instruments Incorporated

Fusion Digital Power Studio GUI for Isolated Power Applications

1 Trademarks

Fusion Digital Power is a trademark of Texas Instruments.
SMBus is a trademark of Intel.
Windows is a registered trademark of Microsoft Corporation.
PMBus is a trademark of SMIF, Inc.
All other trademarks are the property of their respective owners.

2 About This User's Guide

2.1 Introduction

The Fusion Digital Power™ Studio is a GUI tool that supports the evaluation and development of power supply solutions based on Texas Instruments' UCD3xxx family of digital power controllers. This user guide specifically addresses the following Texas Instrument controllers geared towards Isolated Power applications:

- UCD3138(A)
- UCD3138064(A)
- UCD3138A64(A)
- UCD3138128(A)

This tool is available for free download here: http://www.ti.com/fusion-gui.

There are a number of tools available upon installation. This user guide focuses on describing the functions of two important tools namely, the Fusion Digital Power Studio (Fusion GUI) and the Device GUI. These two essential GUIs, in addition to providing key functionality, serve as a launchpad to many of the other tools provided.

2.2 The Fusion Digital Power Studio (Studio GUI)

The Fusion Digital Power Studio or Studio GUI, essentially emulates a Host in a PMBus™ based power supply system (pmbus.org). If PMBus commands are implemented in the firmware of the device under test, then the Studio GUI aids in establishing communication and delivering the supported PMBus functions (such as telemetry). Additionally, when used in conjunction with TI-provided reference firmware, the Studio GUI provides certain additional capabilities related to optimizing the power supply such as adjusting loop compensation and more. Currently, TI provides reference firmware for the four isolated power topologies listed in table below, which are supported by the Studio GUI and can be used in conjunction with associated EVMs available for purchase from www.ti.com:

POWER SUPPLY TOPOLOGY	EVM PART #
Power Factor Correction	UCD3138PFCEVM-026
Phase Shifted Full Bridge	UCD3138PSFBEVM-027
Half-Bridge Resonant LLC	UCD3138LLCEVM-028
Hard Switching Full Bridge	UCD3138HSFBEVM-029

Copyright © 2013–2019, Texas Instruments Incorporated

2.3 The Device GUI

The Device GUI is a launchpad for several invaluable device-related tools that are necessary for working with the UCD3138 (064, A64, 128) devices and developing successful firmware. These tools allow the designer to execute critical tasks associated with the devices during the development phase such as switching between ROM mode and Program Flash mode, downloading firmware, debugging, investigating the contents of registers, and so forth.

A PMBus-based hardware interface, which allows communication between the GUI tool and the UCD3138 (064, A64, 128) devices, is available from Texas Instruments (part #: USB-to-GPIO, http://www.ti.com/tool/usb-to-gpio). One unit of this interface adaptor is provided with the previously mentioned EVMs, but the adaptor is available for stand-alone purchase for use with other UCD3138 (064, A64, 128) EVMs available from Texas Instruments that are not provided with one:

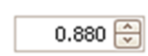
EVM PART #	DESCRIPTION
UCD3138CC64EVM-030	Control card featuring UCD3138RGC
UCD3138OL64EVM-031	Open Loop Evaluation board for UCD3138RGC
UCD3138OL40EVM-032	Open Loop Evaluation board for UCD3138RHA
UCD3138064EVM-166	Control card featuring UCD3138064RGC
UCD3138A64CEVM-660	Control card featuring UCD3138A64PFC

The reference firmware, EVMs, GUI, and interface adaptor constitute a complete and powerful development system that is available for designers to successfully develop power supplies based on the UCD3138 (064, A64, 128).

2.4 Conventions

Any hexadecimal number is prefixed by 0x. For example, 0xFF. Any other number should be assumed to be decimal.

2.5 User Interface Terminology and Tips

FUNCTION		IMAGE
Checkbox	You can select any number of boxes.	
Radio button	You can only select one of the circles at a time. For example, clicking "High" deselects "None."	
Spin edit	Used for numeric entry. You can type in a number directly or click the up and down arrows to increment or decrement the number. The up and down usually changes the last decimal place (adding or subtracting 0.001 in this example).	
Widget	A generic term used to describe a user interface component such as a button or checkbox.	
Disabled (Grayed out)	You cannot edit the widget. This is usually because the GUI has determined that a particular item is a "don't care" or does not make sense given the setting of some other widget or PMBus command.	

2.6 Terminology

'Studio GUI' or 'GUI'—Refers to Fusion Digital Power Studio GUI (main tool), described above

Device GUI'—Refers to the UCD3xxx Device GUI that delivers device-related functions indispensable for development purposes

Copyright © 2013–2019, Texas Instruments Incorporated

2.7 Additional Technical Support

For additional questions or clarifications, take advantage of TI's E2E community: http://e2e.ti.com/support/power_management/digital_power/default.aspx. Alternately, contact your Texas Instruments local representative.

3 Getting Started

3.1 PC Requirements

The GUI requires the following:

- A PC Windows® 7/Windows 10
- Microsoft.NET Framework, version 4.5

Microsoft.NET is the runtime application framework that the GUI uses. The GUI installer ensures version 4.5 of .NET is installed, and installs if necessary.

3.2 USB Adapter

As mentioned earlier, the EVM is attached to the PC through a Texas Instruments serial bus adapter, part number USB-to-GPIO. You should have received this adapter with certain EVMs, but you can also order it as a stand-alone product. The serial adapter must be running firmware v. 1.0.5 or higher. If the firmware of the adapter does not meet this requirement, a warning message appears when the GUI first starts. The GUI can be run in "Offline mode" without the serial bus adapter, which allows you to edit an existing device configuration or experiment with a default "virtual device."

3.3 Download and Installation

The latest public versions can be found at http://www.ti.com/tool/fusion-digital-power-studio.

3.4 Upgrading the GUI

When upgrading to a new release of the GUI, there is no need to un-install the current installed version first. In fact, doing so removes your program preferences, and is not recommended. The GUI installer takes care of updating all necessary files. The program preferences is not modified by the installer.

3.5 Multiple Installations of the GUI

You can install different versions of the GUI on same the PC. Because the preferences are stored within the program folder, each version of the GUI installed on your PC has its own set of preferences.

When you install a second copy of the GUI, ensure the name of the folder for the additional copy is named differently from the default folder name, "Texas Instruments Fusion Digital Power Studio." The easiest way to do this is to append something descriptive to the folder name like "-Beta".

4 Fusion Digital Power Studio (Studio GUI)

4.1 Starting the GUI

The previous form in the installer controls whether GUI "shortcuts" are added to the desktop and quick launch area. The quick launch area is the area next to the Start menu which contains shortcuts to commonly used applications.

Figure 1. Quick Launch Shortcut

When you launch the GUI, it attempts to find a supported device attached to the PMBus. The following sequence is followed:

1. The GUI looks for an attached USB serial bus adapter. If it is not found, you see Figure 2:

Copyright © 2013–2019, Texas Instruments Incorporated

Figure 2. No USB Adapter Found

2. The GUI sends SMBus™ commands to the “broadcast” address 11 telling any devices that are in ROM mode to execute their program (go to flash mode). While this is not necessary for production devices, it may be necessary for in-development products that are set to boot to ROM mode.
3. The GUI scans addresses 1 through 127 for an attached device. It does this by reading a special manufacturer command, DEVICE_ID, on each address. This parameter contains information about the device, including part number and firmware version. Address 12 is skipped because this is reserved for use in the SMBus Alert Response Protocol. After this command has been read then the SETUP_ID is analyzed. If the SETUP_ID is not recognized due to being part of new firmware, for example, then there are some steps that can be taken to still allow for communication with the GUI. See Section 4.2.5.
4. While the scanning process occurs, you see a dialog box like Figure 3:

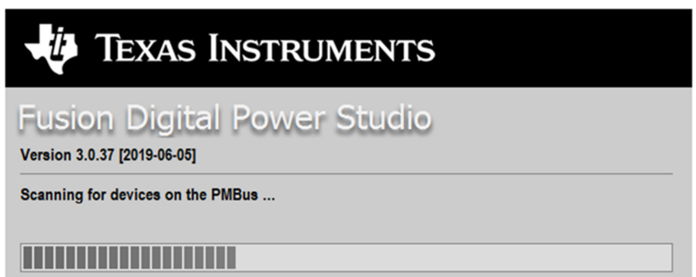


Figure 3. Device Scanning

5. If a supported device cannot be found, you see Figure 4:

Copyright © 2013–2019, Texas Instruments Incorporated

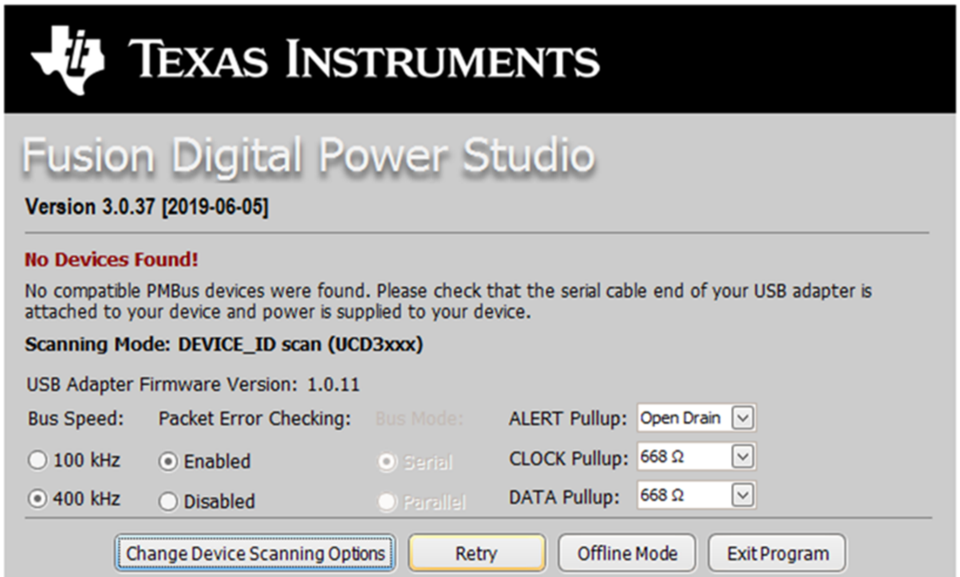


Figure 4. No Devices Found

Double check your USB adapter connection and power to your device and click “Retry” to re-scan.

If the GUI is still unable to detect the device, see the troubleshooting tips in Section 4.2.4.

If you expect the device to not be detected and are interested in working with the offline features for your device, simply click “Offline Mode”. This allows you to use most of the features while not electrically connected to a device. Offline Mode is described in more detail in Section 4.5.

4.2 Connecting to a Remote USB Adapter

The Fusion GUI supports connecting to a remote USB adapter on another PC running the Fusion GUI Adapter Server.

The client computer, (that is the one without the USB Adapter desired) needs to configure the scan process with the IP, port, and password for the other PC. The computer with the desired USB Adapter needs to run a server that is part of the Fusion GUI. The client is discussed first and then the server.

4.2.1 Client Configuration

The following are two ways to get to the client configuration. The first way is from the “Start” menu:

1. Click Start>Texas Instruments Fusion Digital Power>Tools>USB Adapter Mode Selector as shown in Figure 5.

Copyright © 2013–2019, Texas Instruments Incorporated

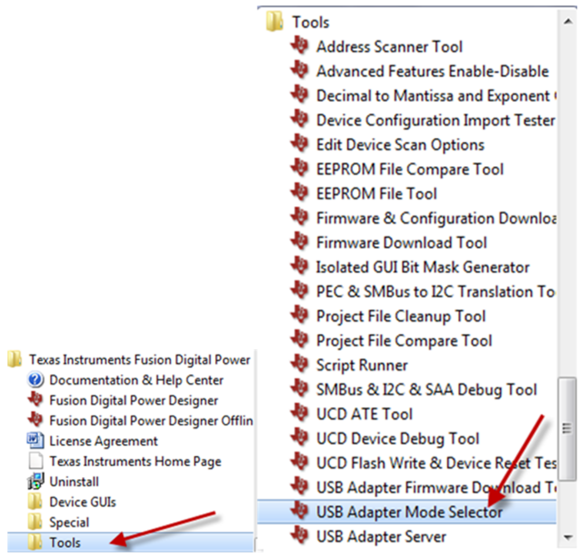


Figure 5. USB Adapter Mode Selector from Start Menu

The default setting is to have the GUI use the local USB adapter. However, to access the remote USB adapter, select "Remote" as shown in Figure 6. The IP and Port of the host needs to be specified (and password if one is set). This information is automatically available in the Fusion GUI Server running as shown in Figure 11. After entering the server information, click "Test Connection" as shown in Figure 8.

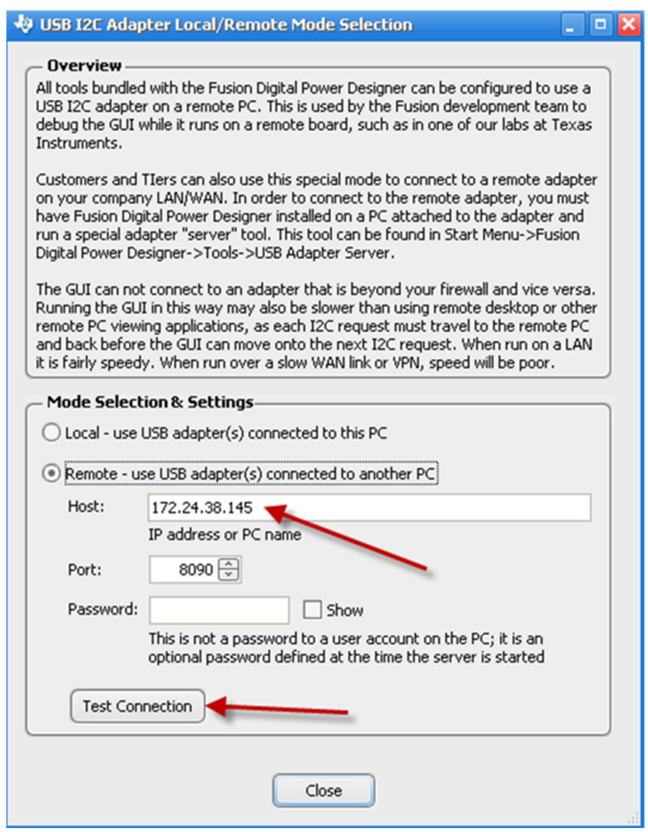


Figure 6. Adapter Mode Selection

Copyright © 2013–2019, Texas Instruments Incorporated

If you click “Test Connection”, you are able to test whether you are connected to the server and observe the following figure indicating success. If unsuccessful, ensure you are connected to the internet and that the information is entered correctly. Note: the GUI cannot connect to an adapter that is beyond your firewall and vice versa.

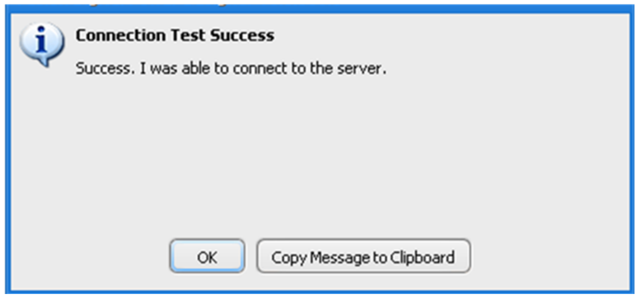


Figure 7. Remote Adapter Connection Test Success

4.2.2 Remote Server Configuration

The server machine needs to run the “USB Adapter Server” to allow clients to connect. The USB Adapter Server can be found in the Start>Texas Instruments Fusion Digital Power>Tools menu as shown in Figure 10.

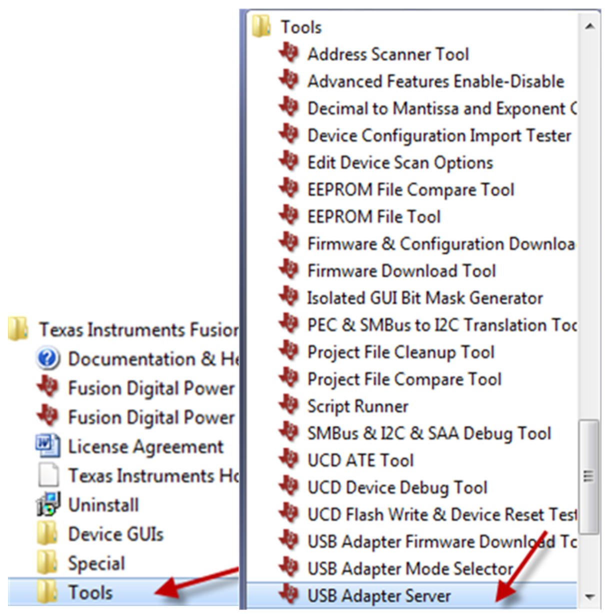


Figure 8. USB Adapter Server from Start Menu

The USB Adapter Server shows its IP Address that the client needs to use. Also, in order for clients to connect to the server, you must click the “Start Server” button.

Copyright © 2013–2019, Texas Instruments Incorporated

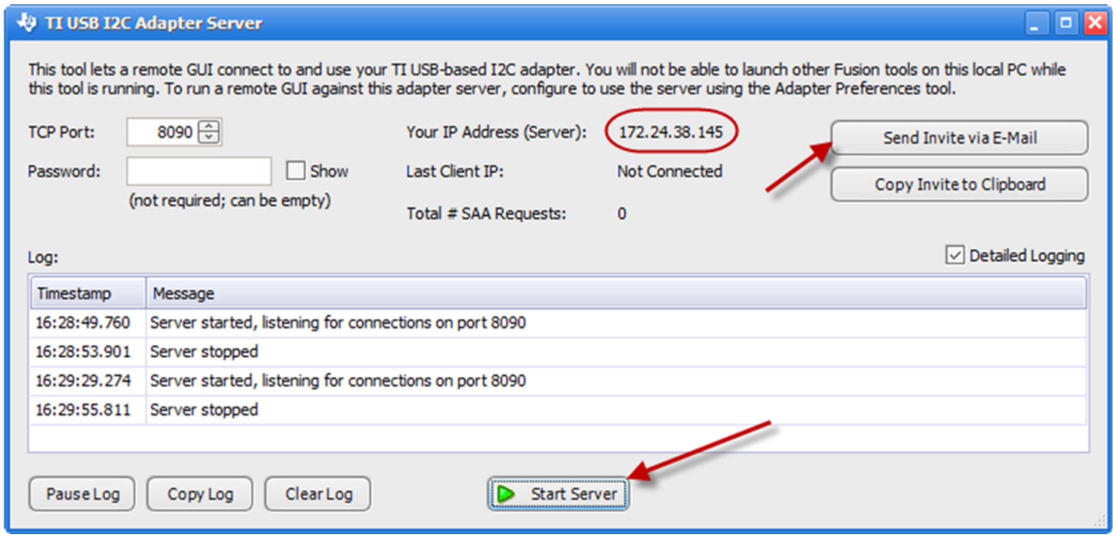


Figure 9. Remote Adapter Server

If you click “Send Invite via E-Mail”, a pre-filled email appears populated with the relevant information for the client to connect to the server. It includes the IP address, Port, and Password, if any. It also includes instructions on how to configure the client. Figure 10 shows the content of the pre-filled email. You need to replace “my-client-friend@firend.com” in the “To:” with the appropriate email address of the client.

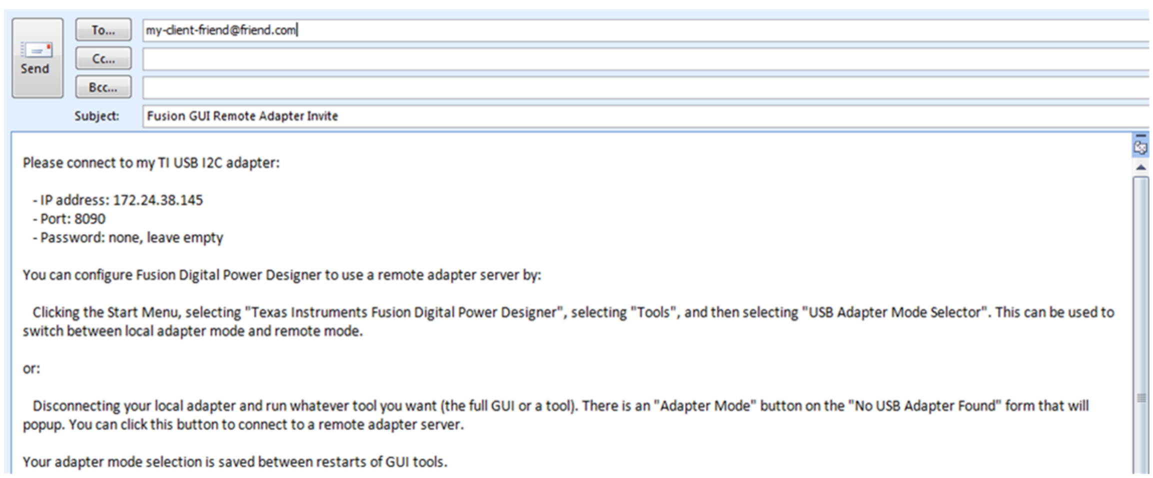


Figure 10. E-mail Invite with Remote Server Settings

4.2.3 Client and Server Running

The following are some figures of a live client and server interacting. For the client, the experience of running the GUI remotely would be the same as running it on a local USB adapter except for the speed being slower.

Copyright © 2013–2019, Texas Instruments Incorporated

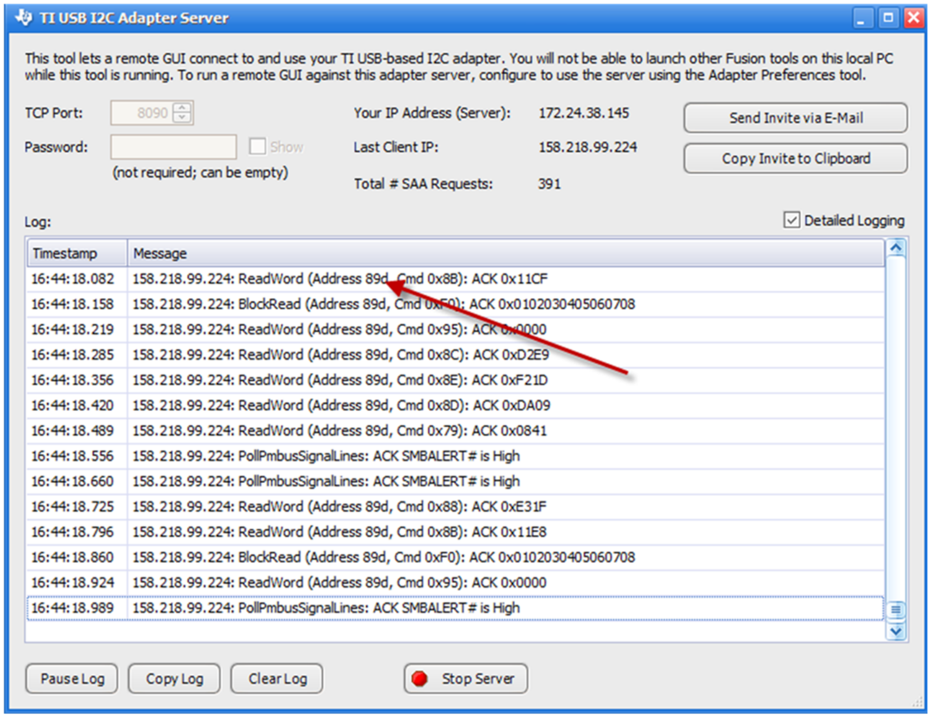


Figure 11. Running a Remote Server Connected to a Client

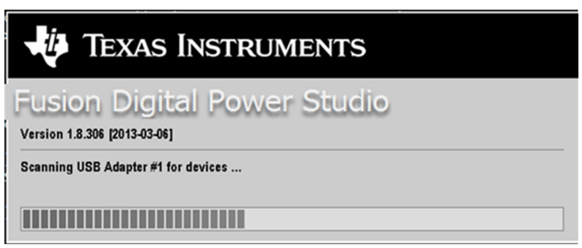


Figure 12. Client Machine Conducting a Scan on the Remote Adapter

Copyright © 2013–2019, Texas Instruments Incorporated

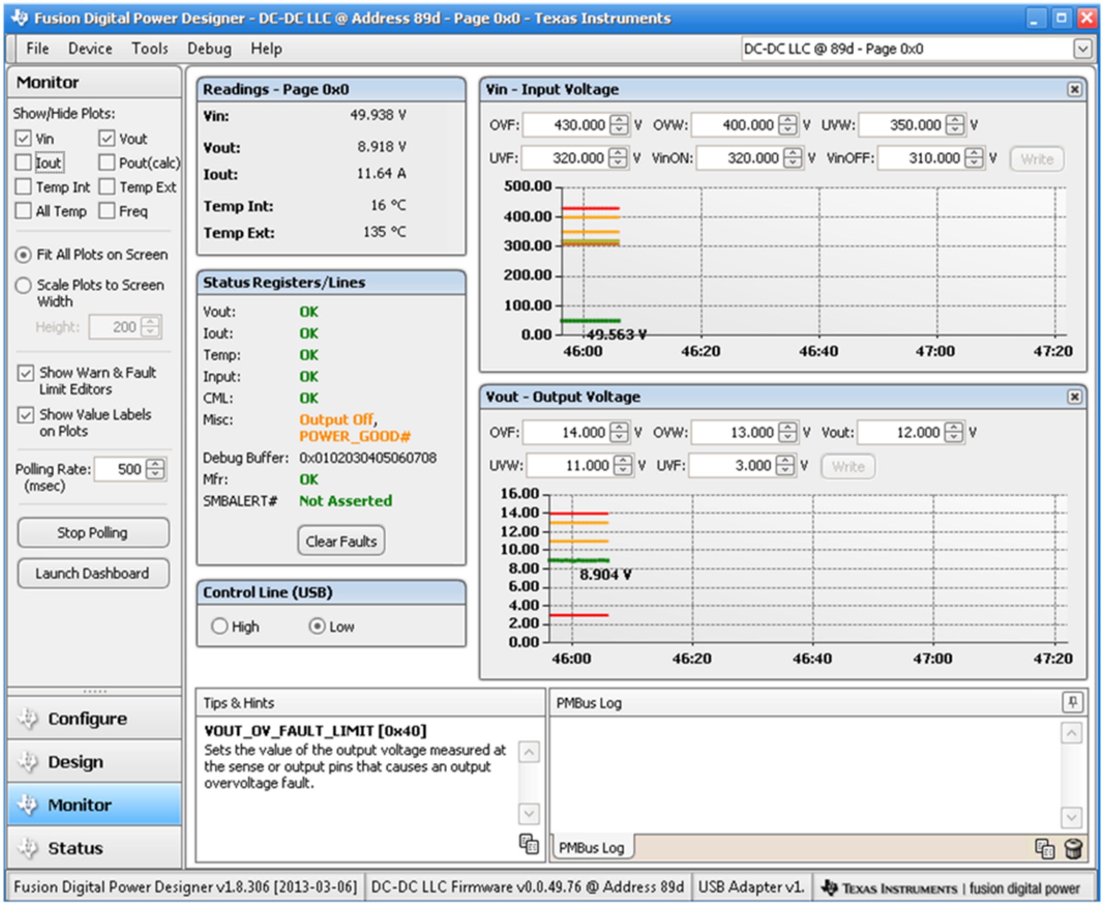


Figure 13. Client Machine Interacting with a Remote USB Adapter Connected to a Device

4.2.4 Connection Troubleshooting Tips

PROBLEM	RESOLUTION
The scan never occurs. The GUI immediately comes up with the error form. When retry is clicked, the error form reappears immediately.	This usually indicates the USB serial adapter is not attached to the PC or is malfunctioning. Verify that the green LED on the serial adapter is ON. If it is not, unplug the adapter, power off your device, reconnect the adapter, and then power on your device.
The GUI scans each address, but cannot find the device.	Verify that power is on to the device. Try re-applying power to the EVM. Also, try resetting the USB adapter as described above.

4.2.5 SETUP_ID in Firmware is Not Recognized by the GUI

Generally in order for the GUI to recognize your firmware, it needs to recognize the manufacturer commands Device_ID and SETUP_ID. However, in the case where you are developing a new firmware and the SETUP_ID is not supported by the GUI, you can change your scan preferences to ignore your SETUP _ID and continue to try to communicate with your device through the GUI. If communication can be established, then you have the ability to interact with the PMBus commands that you have implemented in your firmware. You are not be able to access the Design features of model compensation and the stage of your topology since this requires knowledge of your SETUP_ID which indicates to the GUI the topology of the device.

You can skip the SETUP_ID recognition scan by doing the following.

Copyright © 2013–2019, Texas Instruments Incorporated

4.2.5.1 Change the Device Scanning Options

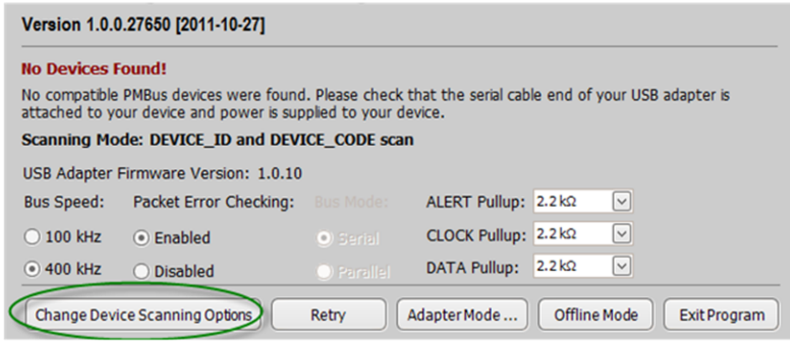


Figure 14. Select Change Device Scanning Options

The following dialog allows you tell the scanner what type of device is to be expected at each address. Click the button "UCD3XXX Isolated" at the top right. Click "OK" and then "Retry" the scan.

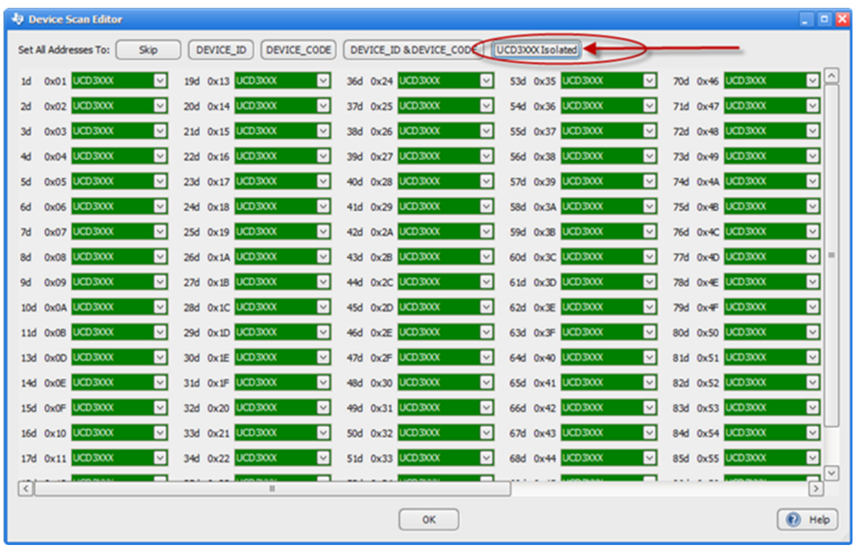


Figure 15. Click UCD3XXX Isolated at Top

4.2.5.2 Click Fallback Mode from Start Menu

An alternative way to change the scanning options is to select this scan mode from the Start Menu as shown below.

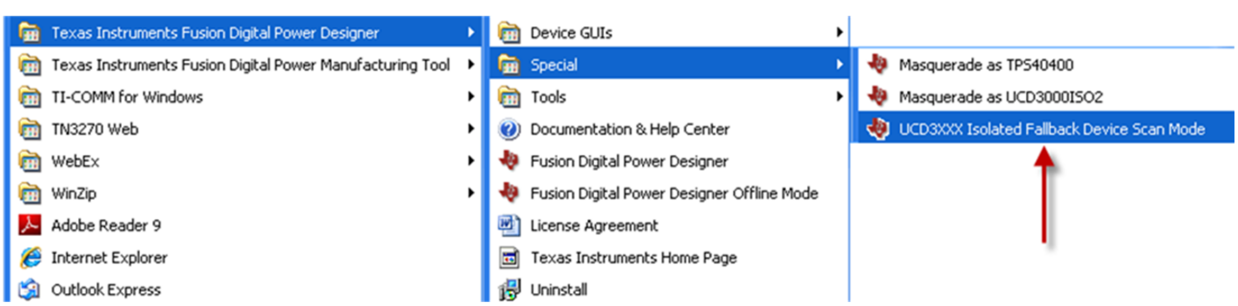


Figure 16. Start>Texas Instruments...>Special>UCD3XXX Isolated Fallback Device Scan Mode

Copyright © 2013–2019, Texas Instruments Incorporated

4.2.6 Enable GUI Protected Features

Figure 19 shows how to access the configuration screen to enable the GUI protected features. Figure 20 shows the screen. Make sure the selections are checked as shown and in the password box type the word “forestln.” Click OK and then many features are available.

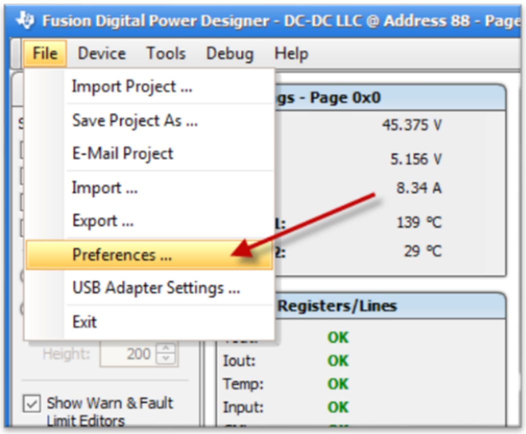


Figure 17. GUI Preferences

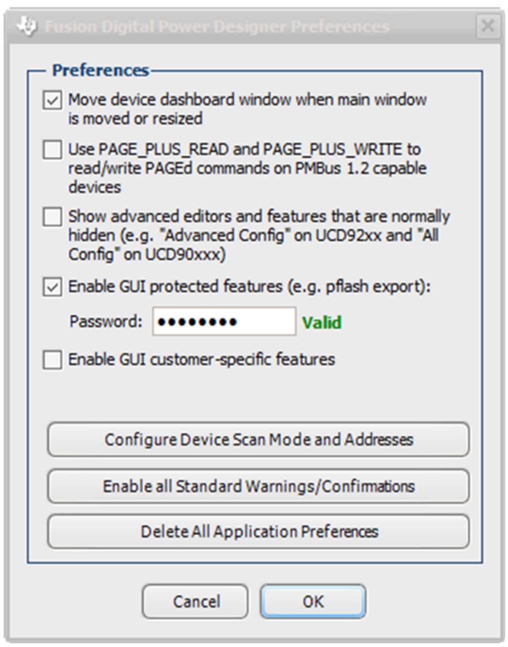


Figure 18. GUI Protected Features

4.2.7 Monitor

After a device is found, the first screen that appears is the Monitor Screen. Depending on which commands are implemented, the corresponding monitor graphs are available. In Figure 19, the commands for reading Vin, Vout, Iout, Pout, Temp 1, Temp 2, and Frequency were all implemented so the graphs are available. The Monitor tab gives you a live view of the active power supply. In addition to plotting the values, it also shows the latest values in the “Readings” group. It also shows a snapshot of the “Status Registers/Lines”. The word “Fault” appears in red when a register is at fault, otherwise a green “OK” is visible. The polling of the parameters being read can also be halted by clicking “Stop Polling” on the left side.

Copyright © 2013–2019, Texas Instruments Incorporated

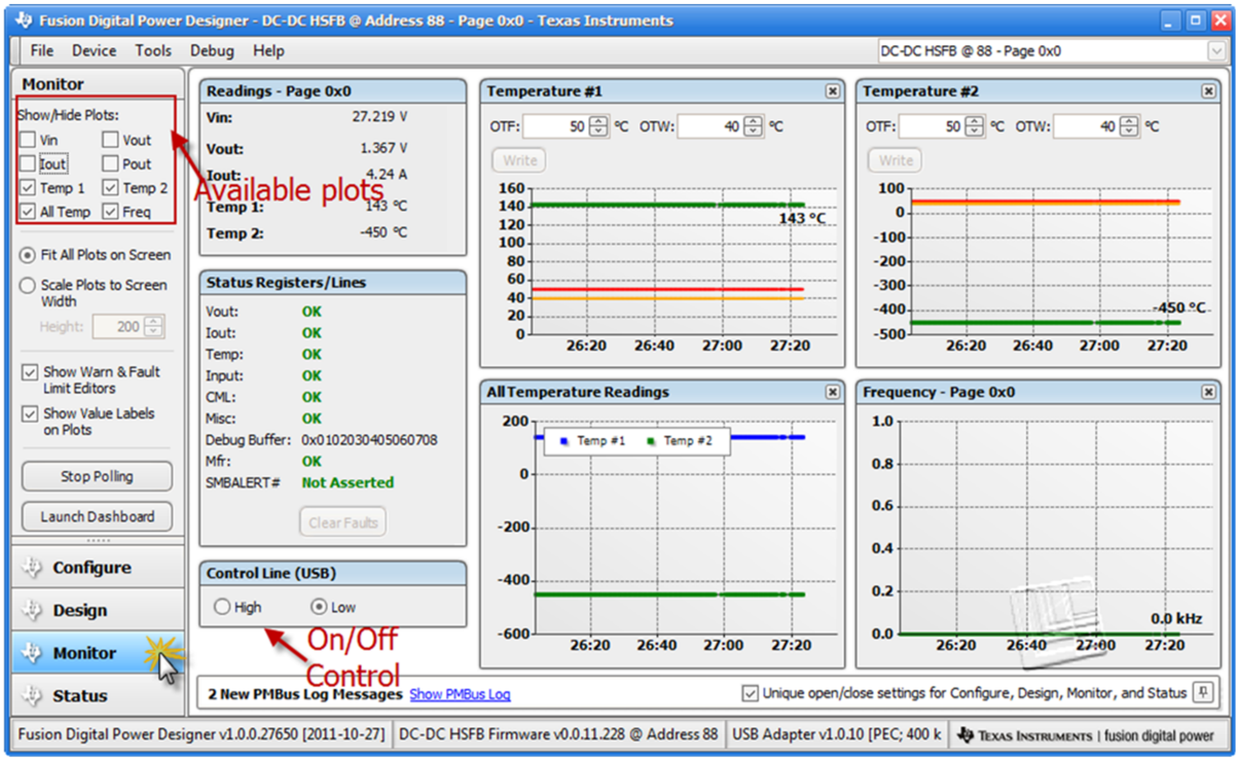


Figure 19. Monitor Mode Displays Some of the Live Parameters Being Read From the Device

4.2.8 Configure

As can be seen from Figure 19, there are a four clickable categories on the bottom left. To get to the Configure mode, select "Configure". The Figure 20 displays some of the features of the Configure mode.

Copyright © 2013–2019, Texas Instruments Incorporated

4.2.8.1 PMBus Commands, Edits, and Writing to Hardware

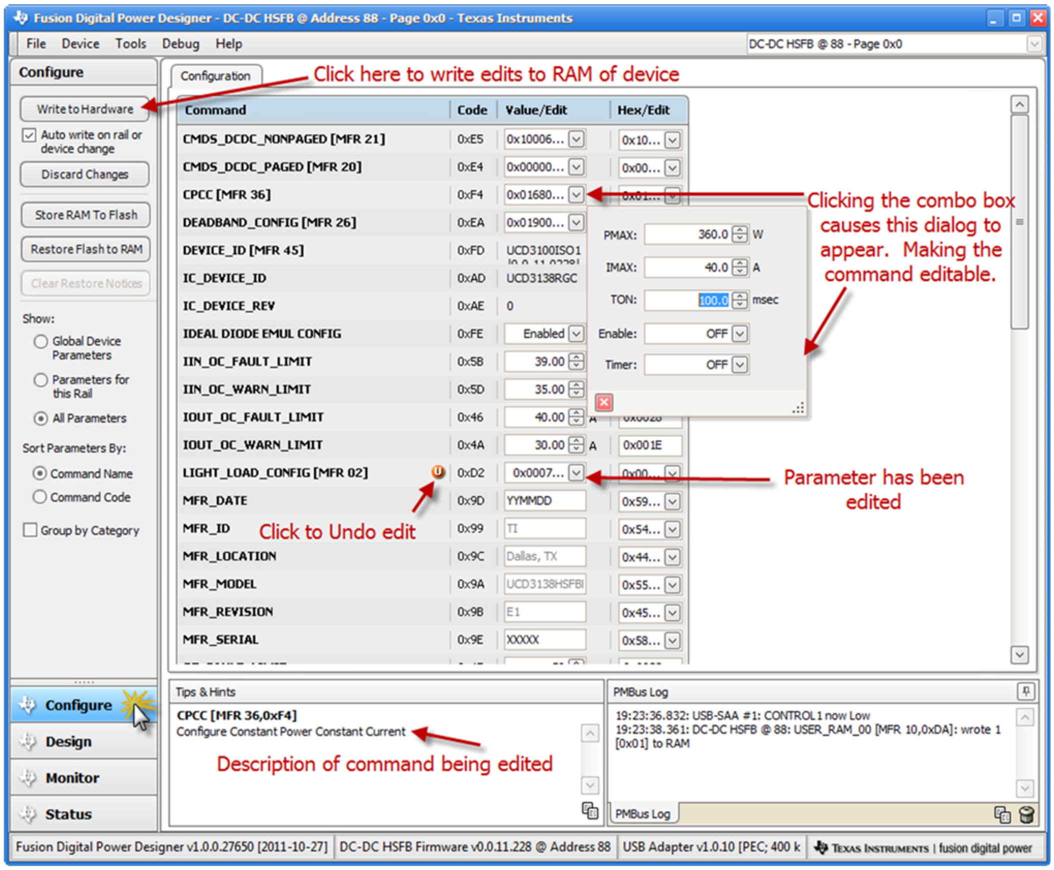


Figure 20. Configure Mode

When the Configure mode appears, all of the implemented PMBus Commands are visible. A discussion of the relationship between what is visible and what is implemented in the firmware is discussed in Section 4.2.8.2. A read was done on all the PMBus Commands and their values are immediately visible.

On the left, there are some controls to decide how they can be ordered to help view them. They may be listed by category, or sorted by name or by hex code.

Some values are read-only (uneditable) and some are writable. Figure 20 shows the parameter LIGHT_LOAD_CONFIG was edited by changing the value. When a command is edited, A appears beside it. This indicates that the value can be undone, or reverted back to the device value stored in RAM. As a command is edited, the value is not automatically written to the device. To write all edits to the device, you need to click “Write to Hardware.” Then if you would like to store those to flash, the button “Store RAM To Flash” would need to be clicked. Section 4.4 discusses storing the current state of all commands to a local file that can be used to write to another device.

Another feature that is highlighted in this figure is the dialog box that appears to edit the Constant Power Constant Current “CPCC” command. Not all commands are direct value edits like “IIN_OC_WARN_LIMIT” that is set for “35 A”. Rather some of them are more complex and require unique dialogs to edit them. CCPC is just one example from many.

Copyright © 2013–2019, Texas Instruments Incorporated

4.2.8.2 How Does Implemented Commands on the Firmware Appear in the GUI?

The Studio GUI is dynamic. It automatically lays out the PMBus commands that are implemented in the firmware. The firmware developer can make a change and then restart the GUI, noticing the change immediately without a new Studio GUI installation. How does the GUI know which commands are implemented? The answer is there are certain Manufacturer commands that indicate which commands are implemented. The command “CMDS_DCDC_NONPAGED [MFR 21] 0xE5” is one such important command that helps the GUI to configure itself. It contains a bitmask. That bitmask is determined in firmware. Each bit in the bitmask indicates whether a command is implemented or not. Each bit refers to a specific command according to the PMBus 1.2 spec. When the GUI reads this bitmask, it looks for all the “1”s and then displays those commands in the GUI.

The Isolated Bitmask Tool, discussed in Section 5.6 of this document, is a valuable tool to help firmware developers set this important bitmask. Figure 21 displays the read-only command “CMDS_DCDC_NONPAGED [MFR 21] 0xE5”.

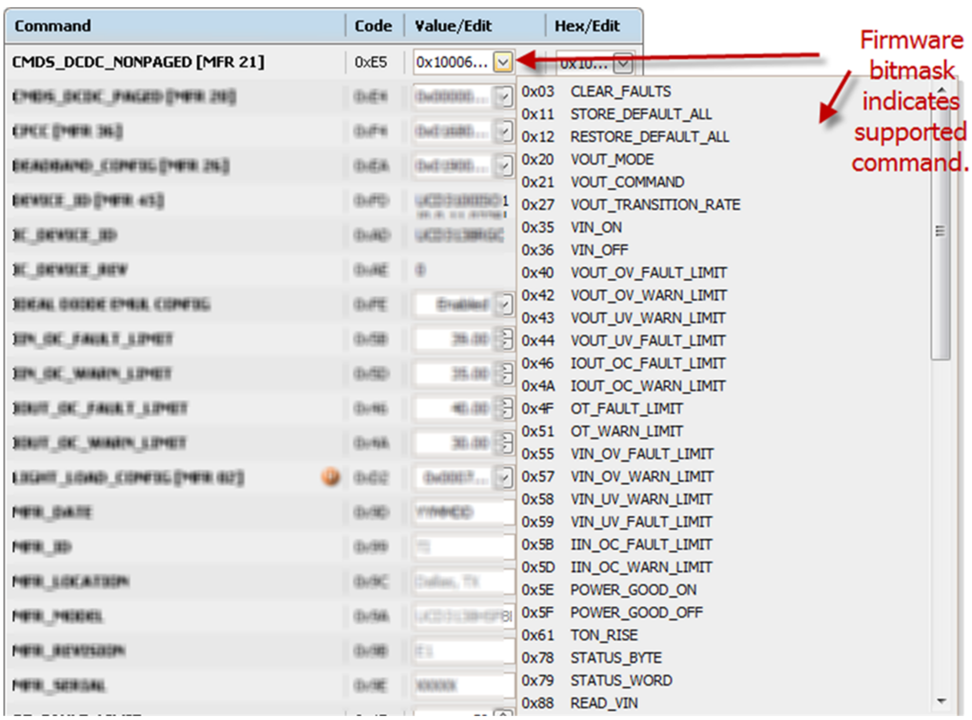


Figure 21. Displays the List of Commands the Firmware Supports

4.2.9 Design – Model Stage and Compensator

To get to the Design mode, click the “Design” button on the bottom left. Figure 22 should appear. The number of loops to configure and parameters in the power stage may differ depending on which of the four power supply topologies is represented by the firmware in the device. Figure 22 illustrates what is available with the “Hard Switching Full Bridge (HSFB)” firmware (featured in UCD3138HSFBEVM-029).

Copyright © 2013–2019, Texas Instruments Incorporated

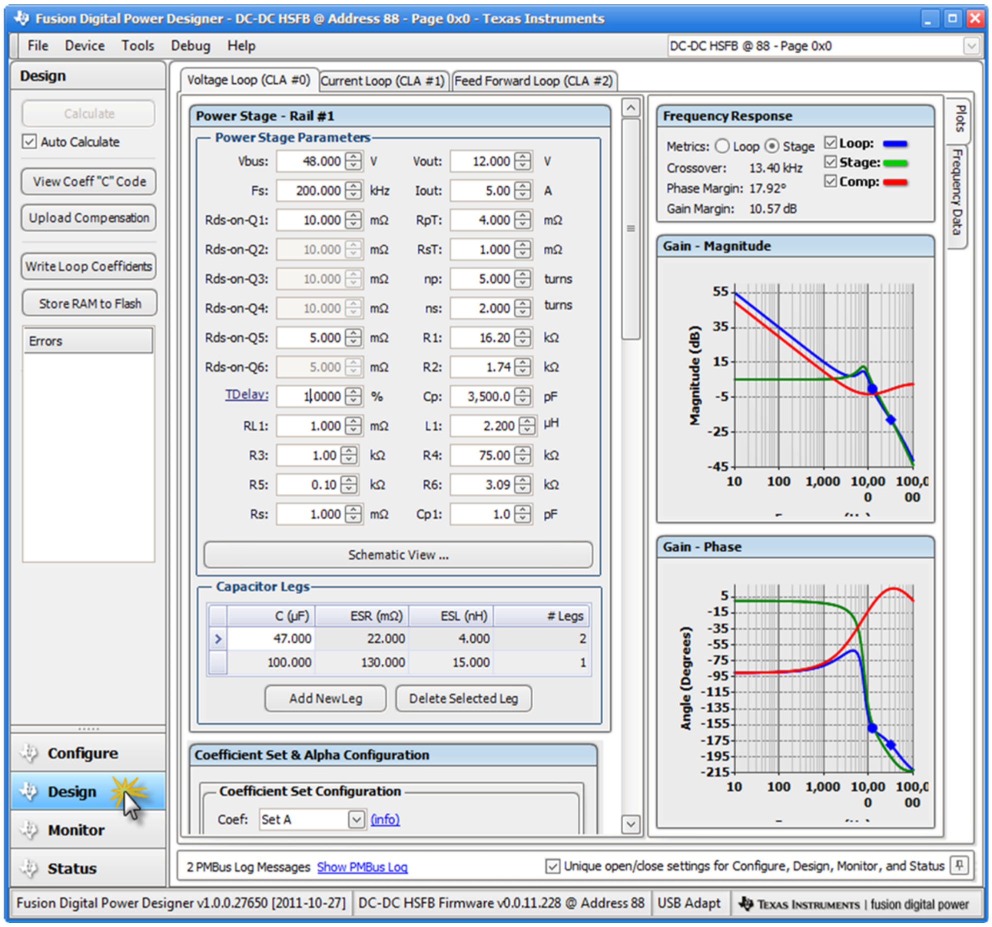


Figure 22. Design Mode Selected

4.2.9.1 Power Stage

Depending on which topology is being modeled, the relevant parameters for the stage are displayed. In the previous example for HSFB, the following parameters for the stage were shown:

Copyright © 2013–2019, Texas Instruments Incorporated

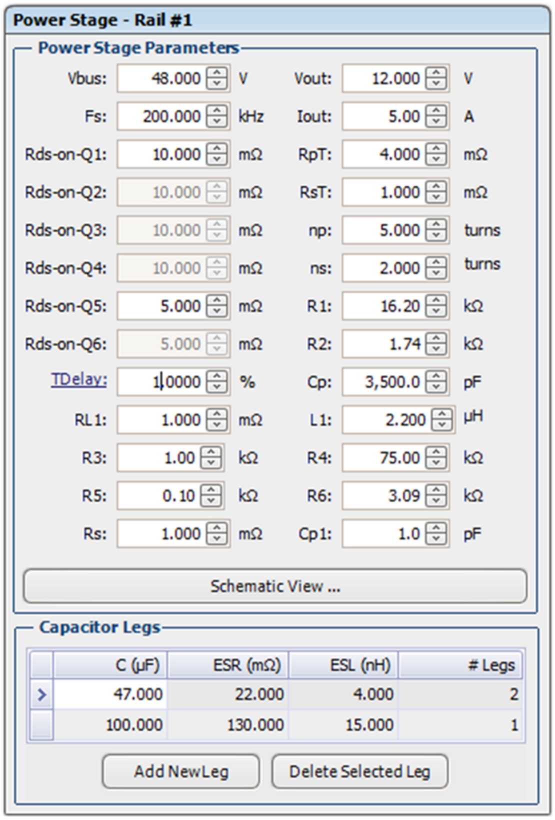


Figure 23. Stage Parameters for HSFB for Voltage Loop (CLA #0)

To model the power stage for the topology, certain parameters need to be specified. Based on the values set, the Bode plot for the power stage is calculated and displayed on the right. This powerful feature is provided to aid designers with fast loop compensation based on analytical models built inside the GUI, representing the power stage equations appropriate for the topology. The power stage equation differs from loop to loop. Figure 23 is part of the voltage loop as shown in Figure 24.

Copyright © 2013–2019, Texas Instruments Incorporated

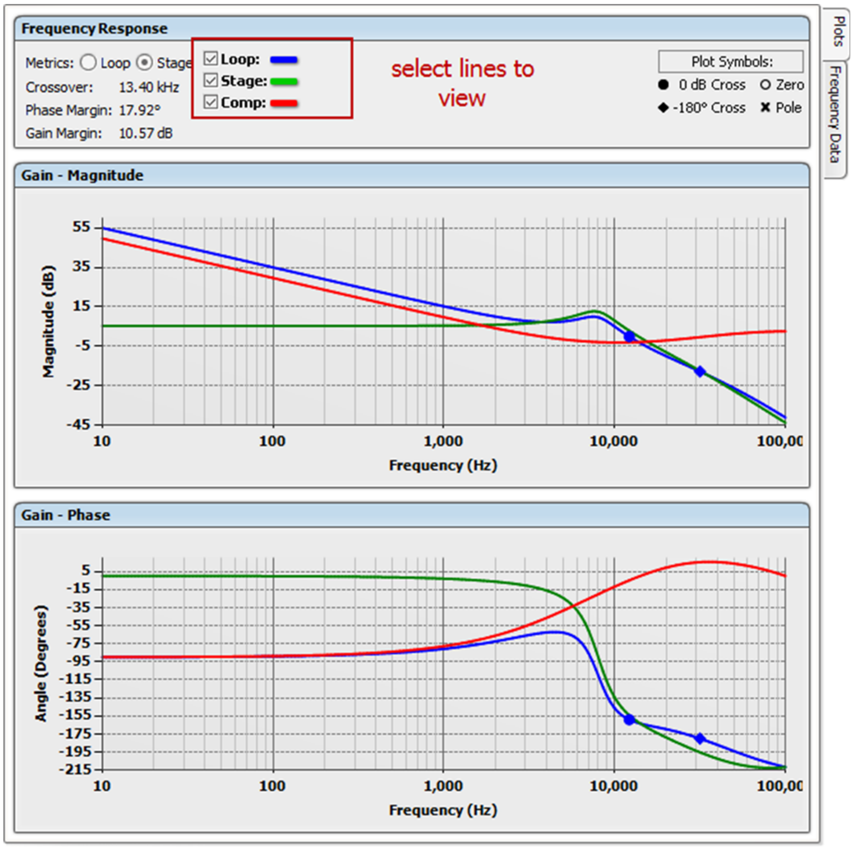


Figure 24. Bode Plots

There are three lines. The green line indicates the power stage. The other two lines are the Compensator and the Loop. Lines can be deselected as shown in Figure 24. The Compensator is discussed in Section 4.2.9.2

Clicking “Schematic View” in Figure 25 opens a dialog with a picture of the schematic.

Copyright © 2013–2019, Texas Instruments Incorporated

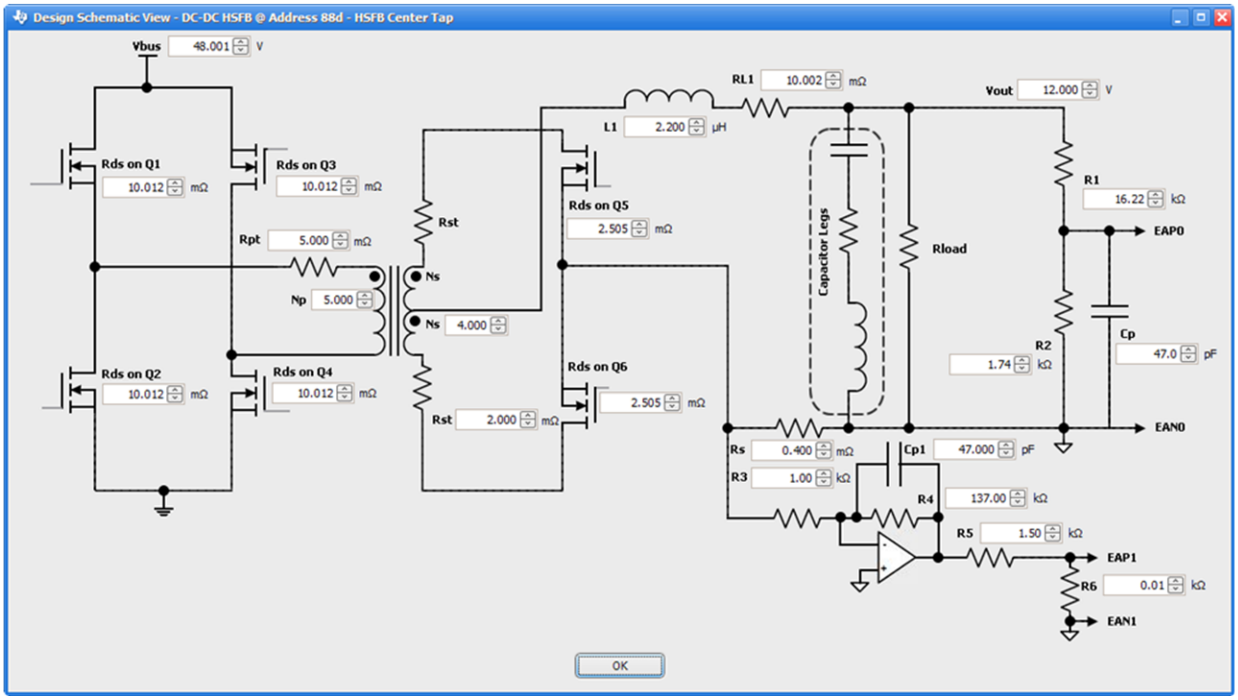


Figure 25. HSFB Schematic Being Modeled

The bode plots are updated automatically as the values are set.

4.2.9.2 Compensator

To model the compensator, there are a number of values to configure. The values to configure for the compensator are the Coefficient Sets (A to G), Alphas (0 and 1), Bins (0 to 6), and Threshold Limits (0 to 5). This needs to be done for each loop. The compensator area is just below the Power Stage Parameters. Simply scroll down to bring the controls into view.

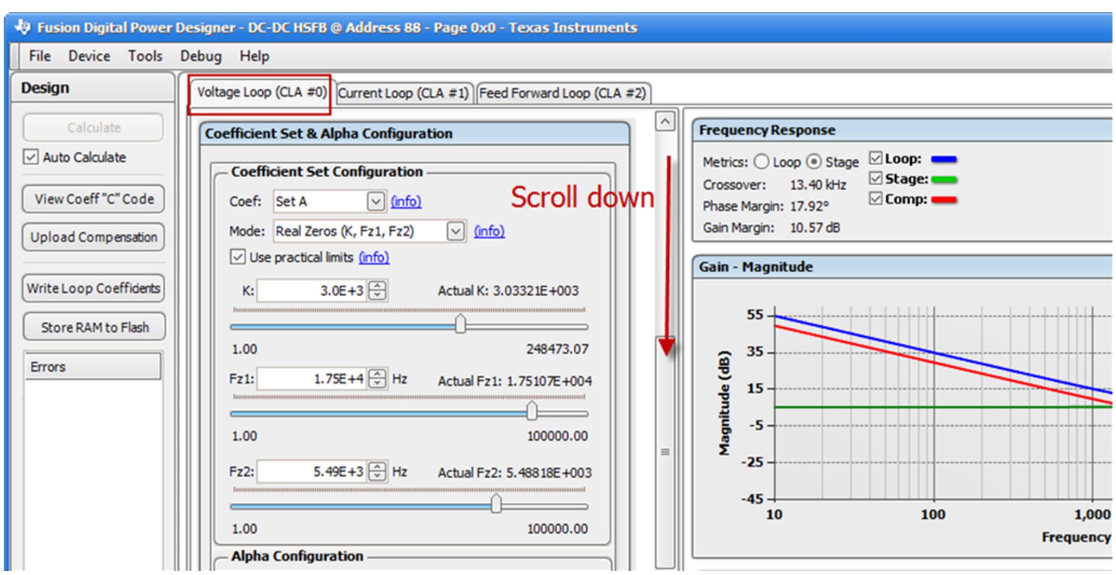


Figure 26. Scroll Down the Stage Parameters to See the Compensator

Copyright © 2013–2019, Texas Instruments Incorporated

The GUI comes equipped with three different ways to program the UCD3138 digital compensator. Figure 27 lists these options. The compensator hardware is described by the forth equation (Device PID). In this context; Kp, Ki, Kd, and α are the raw register values used to configure the positions of the poles and zeros of the compensator. SC is a gain scaling term. Although it is normally set to zero, it provides additional gain for situations where the power stage gain may be low. PRD is used to configure the minimum operating period and KCOMP is used to configure the maximum operating period. In the context of the compensator, they are simply gain terms that modify the overall transfer function by a fixed value. It is important to be aware that the proper way to configure PRD and KCOMP varies based on the control topology implemented. Consult the relevant EVM user guide and training materials for details. Figure 27 is the general equation for the compensator.

Real Zeros

$$\frac{2\pi K\left(\frac{s}{2\pi fz1}+1\right)\left(\frac{s}{2\pi fz1}+1\right)}{s\left(\frac{s}{2\pi fz1}+1\right)}$$

Complex Zeros

$$\frac{2\pi K\left(\frac{s^2}{4\pi^2 fz^2}+\frac{s}{2\pi fzQz}+1\right)}{\left(\frac{s}{2\pi fp}+1\right)}$$

PID

$$\frac{125\,PRD\left(-\frac{256\,Kd\,(z-1)}{a-256\,z}+\frac{Ki(z+1)}{z-1}+Kp\right)}{1\,048\,576\,(PRD+1)}$$

Figure 27. Three Ways to Program the Compensator

4.2.9.2.1 Sample Project Compensation Equations

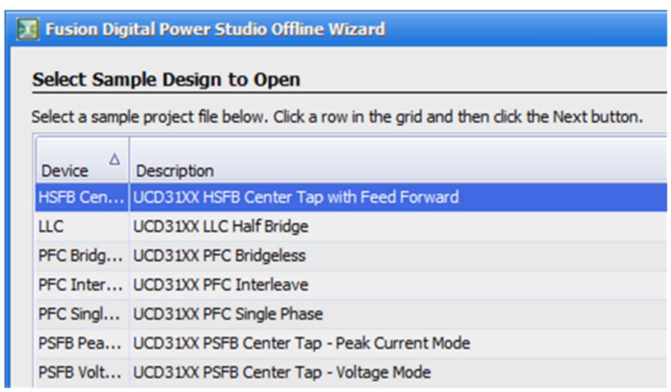


Figure 28. Sample Offline Topologies

Below are the compensator equations used in the GUI for the above sample projects accessible in offline mode.

$$z = e^{s \times Plant.Ts\ /\ CLA\_Design.Oversample} \quad (1)$$

4.2.9.2.2 HSFB Center Tap with Feed Forward

YN0 = 5898240;

$$Compensator(z) = (125.00 \times 2.00^{-42.00\ -\ 1.00 \times SC}) \times KCOMP \times YNO \times (256.00 \times Round(Kd) \times (-1.00 + z)^2 + (-1.00 \times Alpha + 256.00 \times z) \times (Round(Kp) \times (-1.00 + z) + Round(Ki) \times (1.00 + z)))) / ((1.00 + PRD) \times (-1.00 + z) \times (-1.00\ Alpha + 256.00 \times z)) \quad (2)$$

4.2.9.2.3 LLC Half Bridge

Compensator(z) = Filter_LLC(z) × Zopen_loop × Gdelay2

TDPWM = 250e-12

Copyright © 2013–2019, Texas Instruments Incorporated

$$\text{Filter\_LLC}(z) = ((125.00 \times 2.00^{-16.00\ -\ 1.00\ \times\ SC} \times KCOMP \times TDPWM \times (256.00 \times Round(Kd) \times (-1.00 + z)^2) + (-1.00 \times Alpha \times 256.00 \times z) \times (round(Kp) \times (-1.00 + z) + Round(Ki) \times (1.00 + z)))) / ((-1.00 + z) \times (-1.00 \times Alpha + 256.00 \times z))) \quad (3)$$

4.2.9.2.4 PFC Bridgeless, Interleave, Single Phase

$$\text{Compensator}(z) = (125.00 \times Math.Pow(2.00, -20.00 - 1.00 \times SC) \times KCOMP \times (256.00 \times Round(Kd) \times Complex.Pow(-1.00 + z, 2) + (-1.00 \times Alpha + 256.00 \times z) \times (Round(Kp) \times (-1.00 + z) + Round(Ki) \times (1.00 + z)))) / ((1.00 + PRD) \times (-1.00 + z) \times (-1.00 \times Alpha + 256.00 \times z)); \quad (4)$$

4.2.9.2.5 PSFB Center Tap - Peak Current Mode

double VDAC = 1.6;

$$\text{Compensator}(z) = (0.00047683897719946744 \times 2.00^{-16.00\ -\ 1.00\ \times\ SC} \times KCOMP \times VDAC \times (256.00 \times Round(Kd) \times (-1.00 + z)^2 + (-1.00 \times Alpha + 256.00 \times z) \times (Round(Kp) \times (-1.00 + z) + Round(Ki) \times (1.00 + z)))) / ((-1.00 + z) \times (-1.00 \times Alpha + 256.00 \times z)); \quad (5)$$

4.2.9.2.6 PSFB Center Tap – Voltage Mode

$$\text{Compensator}(z) = (125.00 \times 2.00^{-19.00\ -\ 1.00\ \times\ SC} \times KCOMP \times (256.00 \times Round(Kd) \times (-1.00 + z)^2 + (-1.00 \times Alpha + 256.00 \times z) \times (Round(Kp) \times (-1.00 + z) + Round(Ki) \times (1.00 + z)))) / ((1.00 + PRD) \times (-1.00 + z) \times (-1.00 \times Alpha + 256.00 \times z)); \quad (6)$$

4.2.9.2.7 Coefficient Sets and Alpha

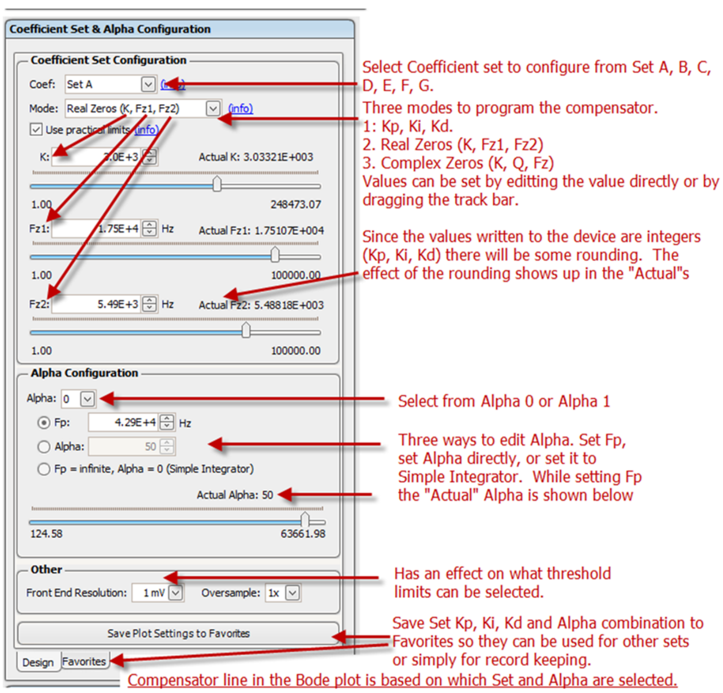


Figure 29. Coefficient Set and Alpha Configuration

4.2.9.2.8 Bode Plot

The Bode plot located on the right of Figure 28 is based on the selected Set and Alpha.

4.2.9.2.9 Saving Favorites

Sometimes you might like to keep copies of their Sets and Alphas so you may use them later or apply them to another Set and Alpha. This is possible by clicking the “Save Plot Settings to Favorites” button in Figure 30.

Copyright © 2013–2019, Texas Instruments Incorporated

You can also access the “Favorites” tab directly to view all their Alpha-Set combinations. They can also copy favorites and add descriptions. See Figure 31.

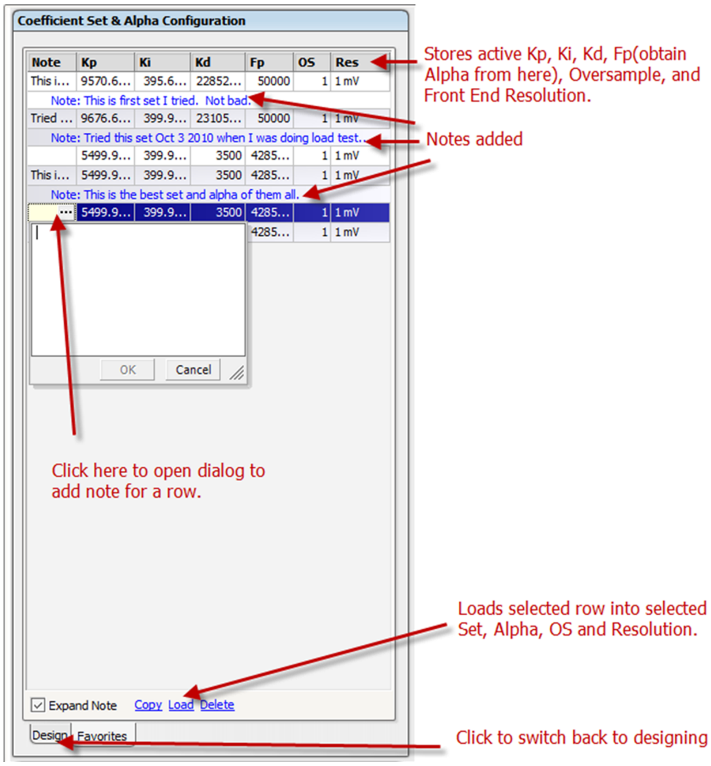


Figure 30. Favorites

4.2.9.2.10 Coefficient Set and Alpha Summary

Immediately below the Set configuration is the “Coefficient Set and Alpha Summary.” This section displays all the alphas and coefficient sets.

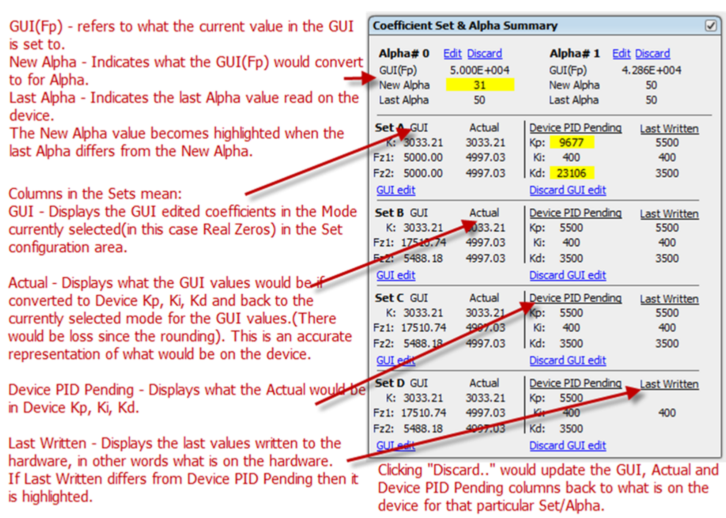


Figure 31. Coefficient Set and Alpha Summary

Another way to discard all GUI edits globally is to click “Upload Compensation” as described in Section 4.2.9.2.5.

Copyright © 2013–2019, Texas Instruments Incorporated

4.2.9.2.11 Bin Assignment and Non-Linear Table Configuration

To configure the non-linear table, specify which sets and alphas are to be used within the configurable limits. One of the rules of the limits is that Lim 0 should be less than Lim 1, Lim 1 should be less than Lim 2, and so forth...Lim (n) < Lim (n+1). If the limits are not configured validly, then the "Write Loop Coefficients" button is disabled.

4.2.9.2.11.1 Make Non-Linear Table Linear – Apply Bin 0 to All

If you wish to simply use the same Set and Alpha for all the limits, making it essentially Linear, then select the convenience option "Apply Bin 0 configuration to all bins". All the errors are removed in this case even though all the Limits are the same. See Figure 32 where all the bins are configured for Set C and Alpha 1.

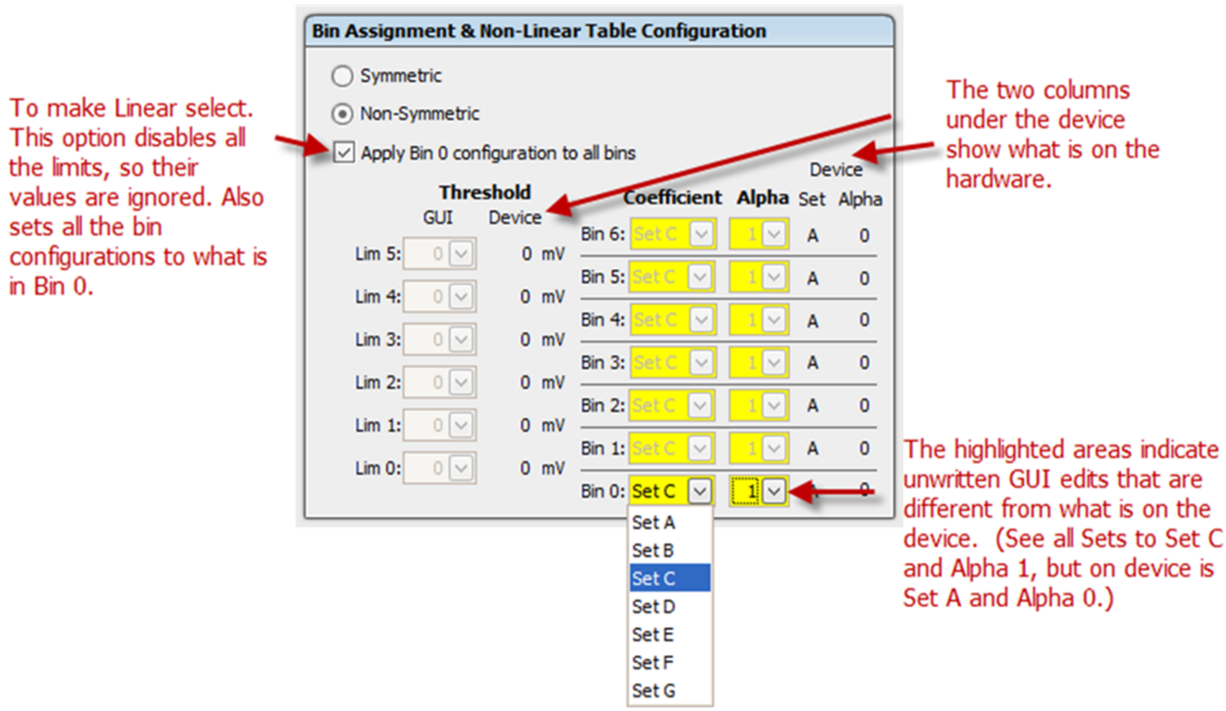


Figure 32. Apply Bin 0 to All Bins (Linear)

4.2.9.2.11.2 Non-Symmetric and Symmetric

There is an option to make the Limits Symmetric or Non-Symmetric. For Non-Symmetric, the limits can be positive or negative. For Symmetric, the limits specified must be positive since the symmetric part is automatic and negates all the positive limits. See Figure 33.

Copyright © 2013–2019, Texas Instruments Incorporated

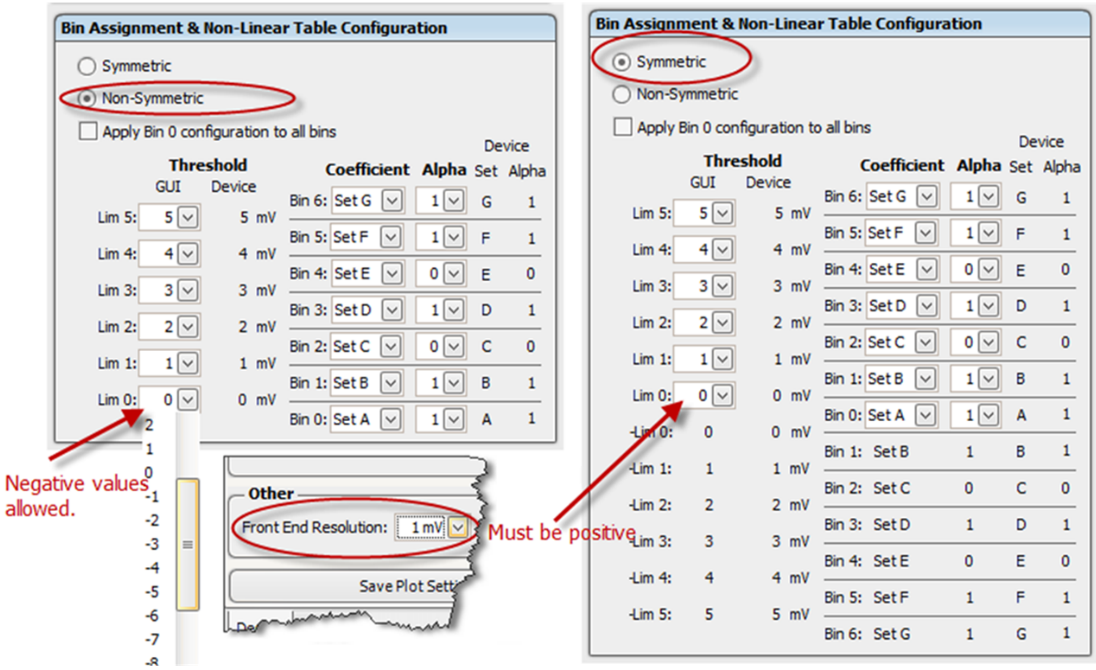


Figure 33. Symmetric and Non-Symmetric

4.2.9.2.12 *Writing Loop Coefficients, C Code, Upload Compensation*

After you are satisfied with their configuration, you can then proceed to writing it to the hardware. This does not happen automatically but requires you to "Write Loop Coefficients." If there are errors, they need to be corrected before the writing can proceed. What is written? All the highlighted values are an indication of what is different from what is on the device so those values are written. If you wish to discard all their GUI edits or the highlighted values, they can do a global discard by simply clicking "Upload Compensation." These buttons mentioned are located on the left side. You can also view the C code that represents the coefficients in firmware by clicking "View Coeff 'C' Code". See Figure 34.

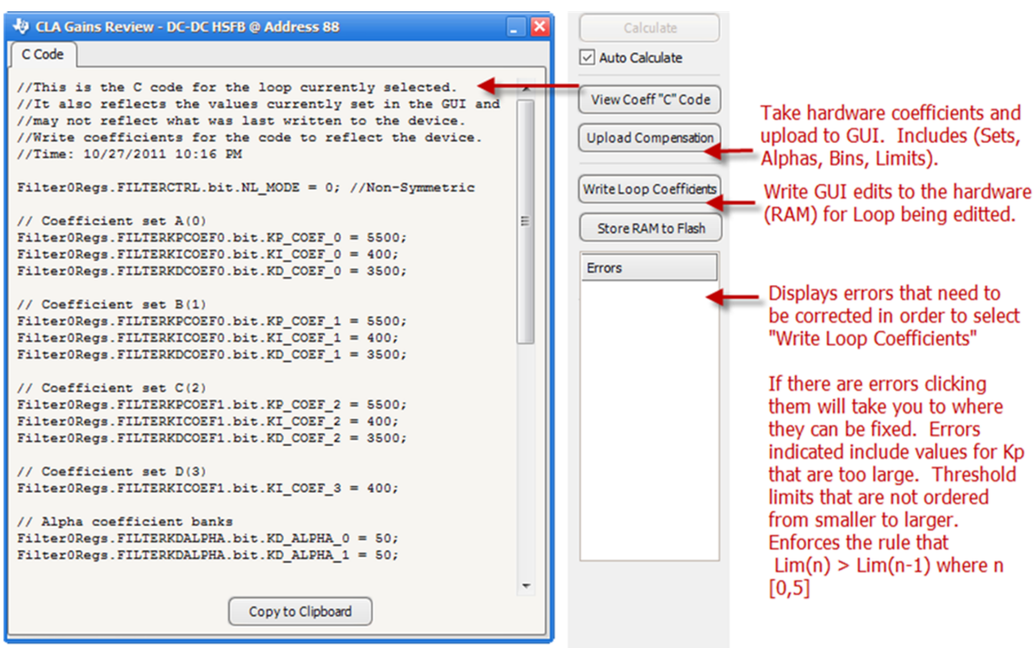


Figure 34. Writing Loop Coefficients and Global Reset of GUI Edits to Hardware Coefficients

Copyright © 2013–2019, Texas Instruments Incorporated

4.3 Status

The final mode is the status tab. It provides additional details on the type of fault or warning. Figure 35 shows a screen shot of this tab.

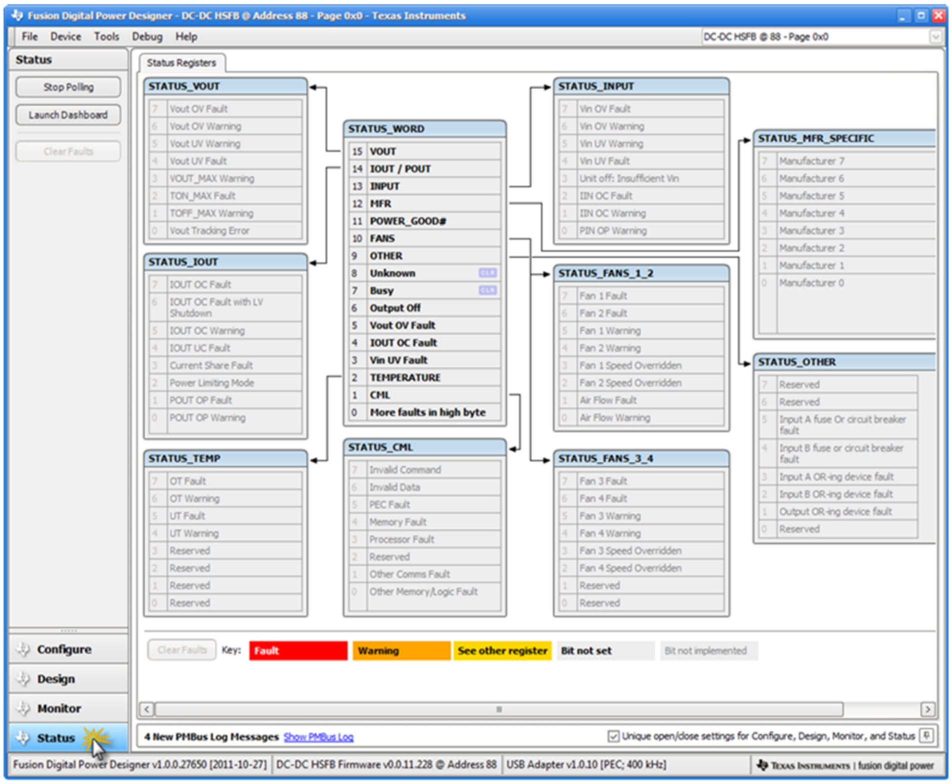


Figure 35. Status Mode

4.4 Capturing the State of the Device - Saving a Project File

After editing PMBus commands in Configuration Mode or editing the Compensation, you can simply click the “Write …” button on the left to commit those changes to the RAM of the hardware. They can then follow that with a “Store RAM to Flash” to save the hardware changes to Flash so that they would remain after the device undergoes a reset. If the changes on the hardware are not flashed, then a reset would simply restore what is in flash and overwrite what was previously written to RAM.

However, the above only covers writing device-related parameters. What about the parameters set in the Power Stage in Design mode? These are not stored on the device. The only way these can be stored is by saving a “Project File”. The Project File is an .XML file stored on the PC. Not only does it contain design parameters, but it also stores the current state of all PMBus commands. So it is a snapshot of the device and more.

To save a “Project File”, simply click File> Save Project As …

Copyright © 2013–2019, Texas Instruments Incorporated

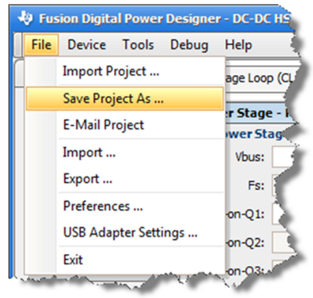


Figure 36. Save Project File

What can be done with a project file? If a new device was hooked up to the PC, you can simply import the project file and write that to the device. The project file can also be used in Offline mode and act as a virtual device.

4.5 Miscellaneous Tools

4.5.1 Multi-image

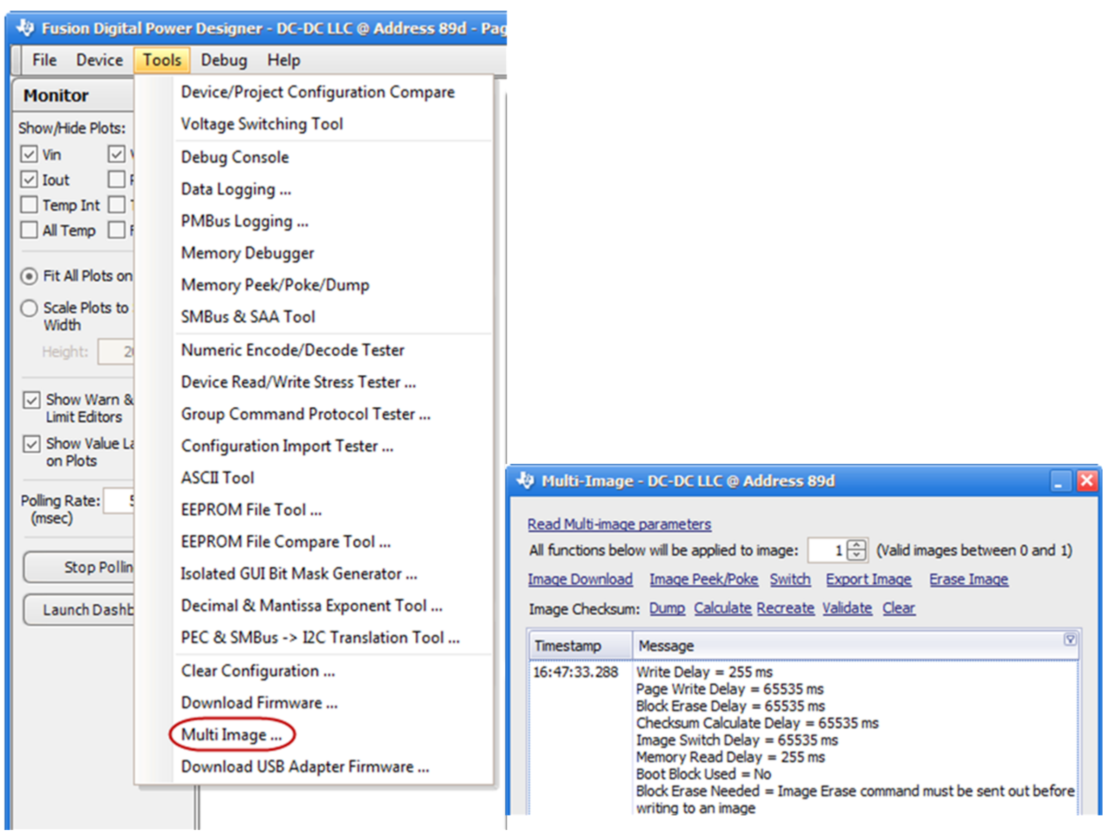


Figure 37. Multi-image

There are a number of other functions that can be performed from the “Tools” menu. Clicking “Multi Image …” shows a dialog with a number of multi-image functions as shown in Figure 38. These functions are also available from the Device GUI and are covered in detail in Section 5.5. One feature that can be observed in the Fusion Studio that is not seen in the Device GUI is the ability to download to a non-executing image and still observe the device monitoring various parameters. This can be seen in the background of Figure 39.

Copyright © 2013–2019, Texas Instruments Incorporated

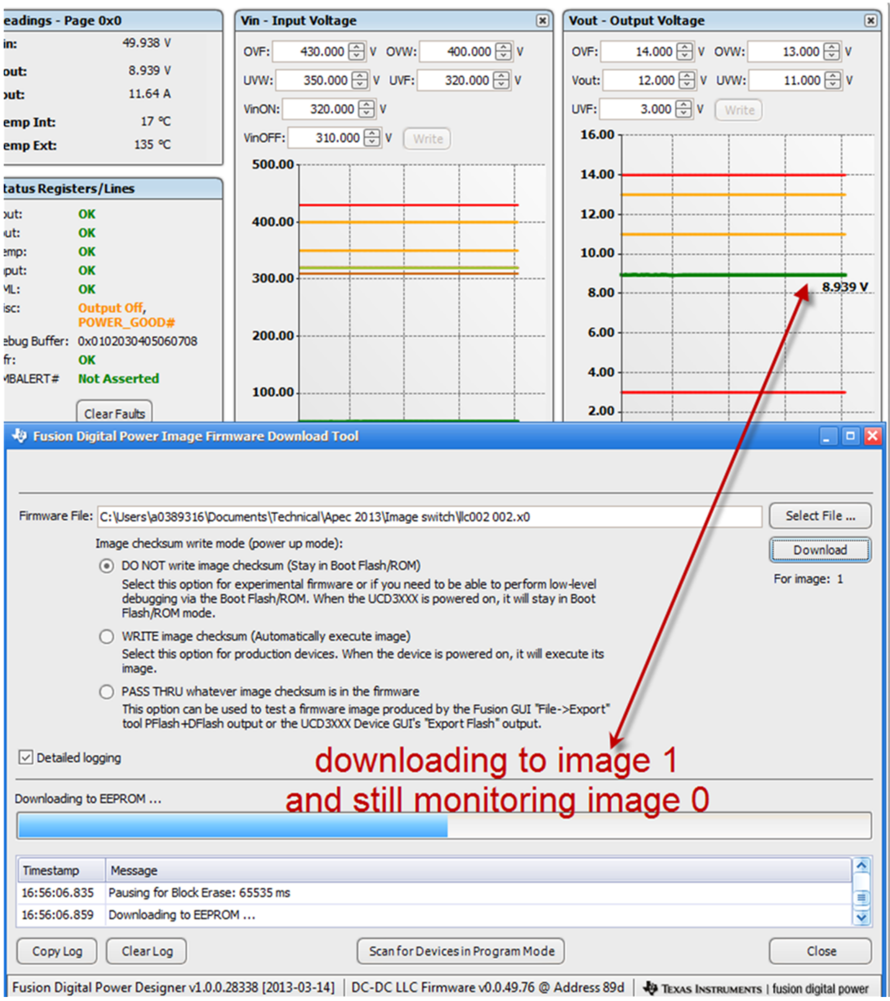


Figure 38. Downloading to an Image While Monitoring at the Same Time

Clicking “Switch” in the Multi-image window activates the new image. The GUI needs to restart to load the new image. NOTE: The power supply is not reset.

Copyright © 2013–2019, Texas Instruments Incorporated

4.5.2 Isolated Bitmask Tool

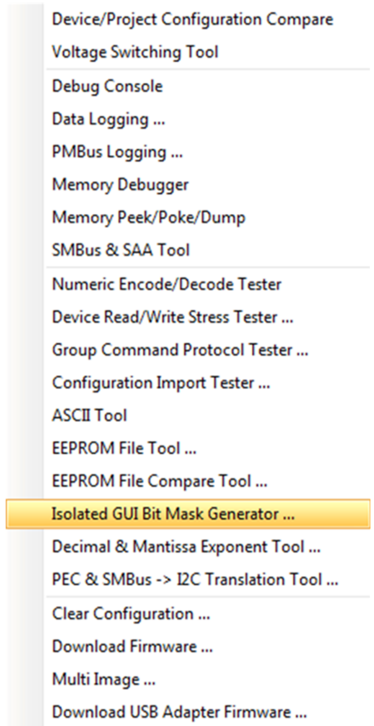


Figure 39. Tools> Isolated GUI Bit Mask Generator …

The “Isolated GUI Bit Mask Generator” is also detailed in the part of this document describing the functions of the Device GUI in Section 5.6. One feature that is available in the Online Fusion Studio that is not in the Device GUI is the ability to view the PMBus command bitmasks set in the firmware. Simply click “Upload bitmask from device” as shown in Figure 41. This is a quick way to debug why a command may not be visible in the configuration tab if the reason is the bit of the command was not set in the bitmask.

Copyright © 2013–2019, Texas Instruments Incorporated

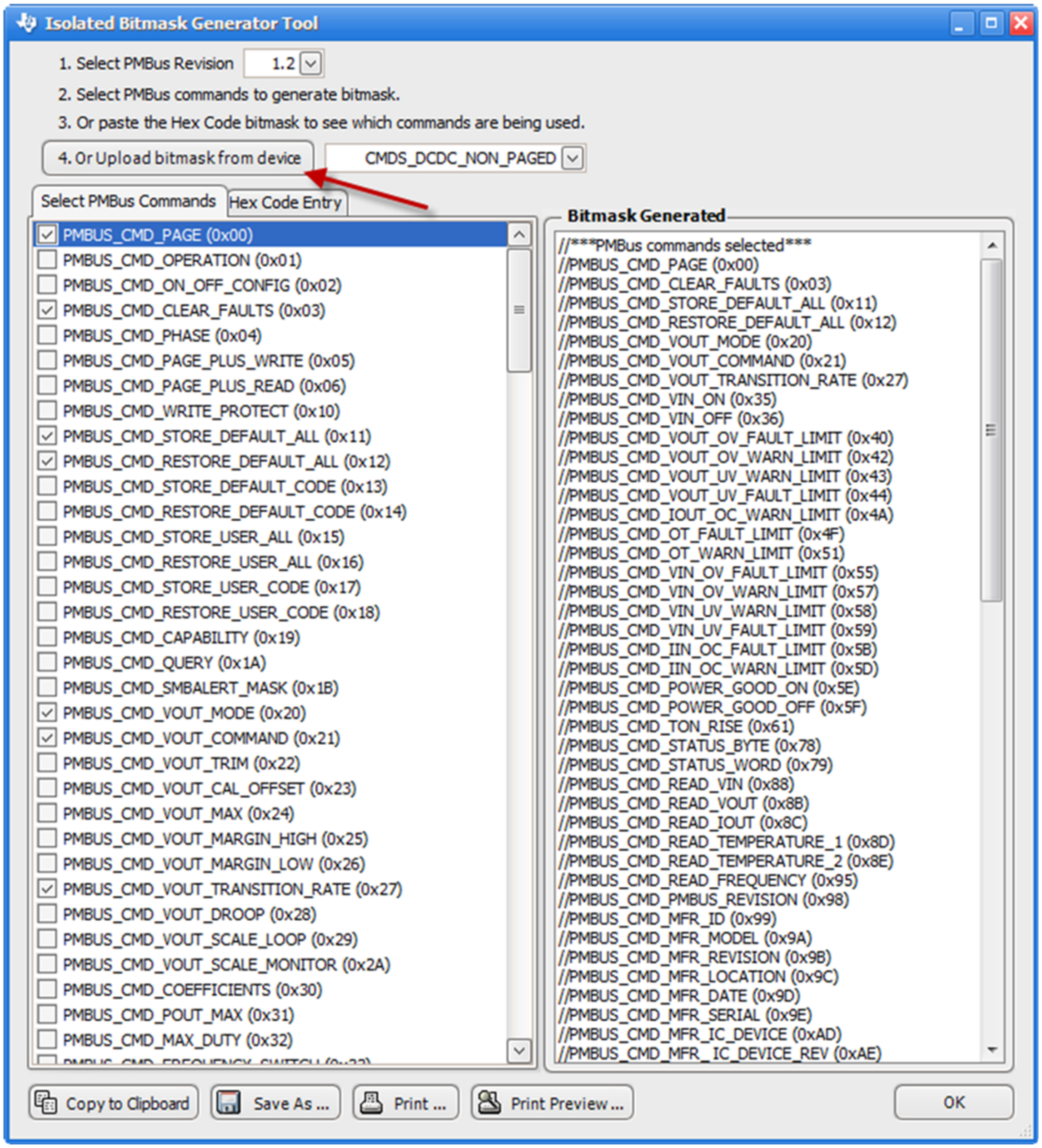


Figure 40. Isolated Bitmask Tool in Fusion Studio (Online)

4.6 Offline Mode

So far, all the discussion has been related to communicating with a device that is connected and online. There is also a concept of working with the device in offline mode. This is done by working with a previously saved Project File as discussed in the last section or by working with Sample Project Files that are already embedded in the GUI. In offline mode, you can write PMBUS commands to a “virtual device” and you can also do modeling in Design mode. When the you get a device, you can simply import this project file that they have worked offline with and sync the device to that.

4.6.1 Starting in Offline Mode

To start offline, you can click the other shortcut that came when the GUI was installed. See Figure 41.

Copyright © 2013–2019, Texas Instruments Incorporated

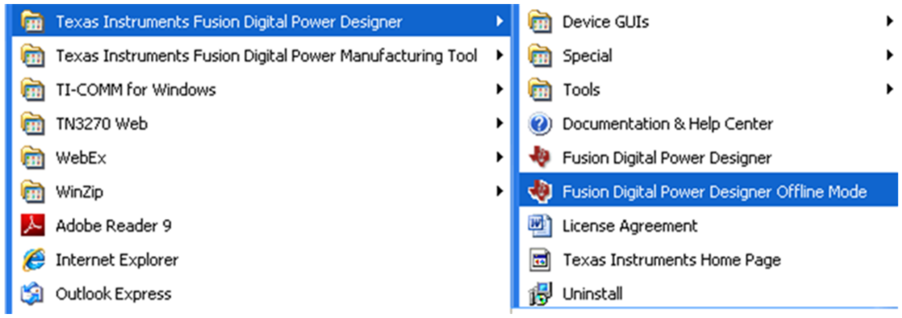


Figure 41. Starting in Offline Mode

Another way to start in offline mode is to unplug any connected devices and start the GUI normally with the other shortcut. This causes the GUI to scan for devices and then upon the fail prompts you to Retry, or work in offline mode.

4.6.2 Open Existing Project File

In offline mode, select from three options. The first option is to open an existing project file that has been previously saved.

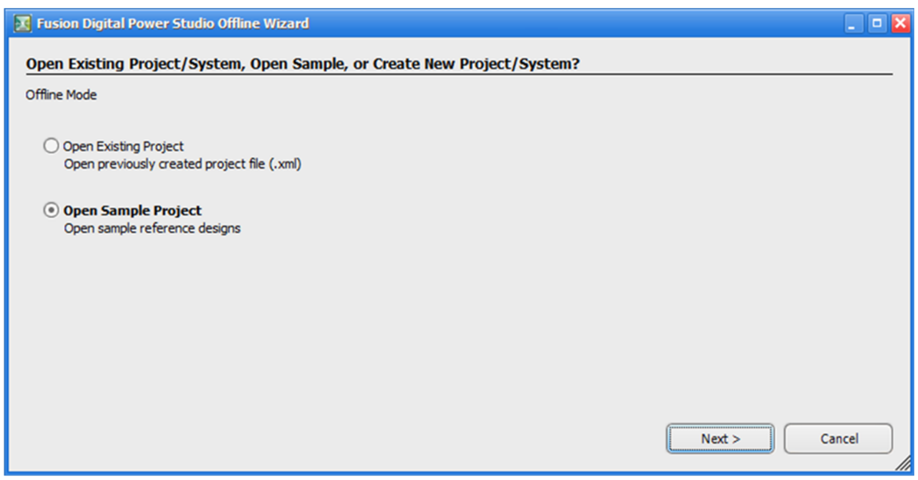
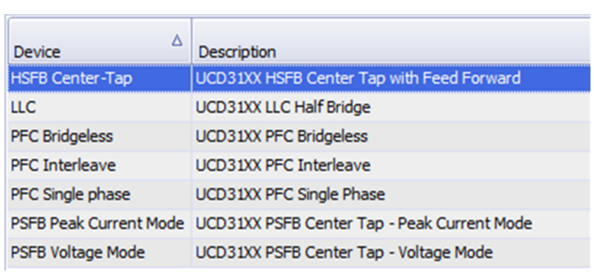


Figure 42. Offline Options

4.6.3 Open Sample Project

You can also open a sample project file and work with that. They can then save that afterwards as a project file to their PC and use it later to import to a device. The following sample projects are available at this time.

After clicking "Next" the sample projects appear. This list increases as new topologies are supported.



Device	Description
HSFB Center-Tap	UCD31XX HSFB Center Tap with Feed Forward
LLC	UCD31XX LLC Half Bridge
PFC Bridgeless	UCD31XX PFC Bridgeless
PFC Interleave	UCD31XX PFC Interleave
PFC Single phase	UCD31XX PFC Single Phase
PSFB Peak Current Mode	UCD31XX PSFB Center Tap - Peak Current Mode
PSFB Voltage Mode	UCD31XX PSFB Center Tap - Voltage Mode

Figure 43. Offline Sample Topologies

Copyright © 2013–2019, Texas Instruments Incorporated

5 Device GUI

In the previous section, the Fusion Studio GUI was described. In this section, the Device GUI is described. The device GUI provides an entry point to a number of important development tools indispensable for working with the UCD3138(064, A64, 128) devices. You also find out that a number of these tools are also available in the Studio GUI under the Tools menu. You may use whichever entry point they wish to launch these tools. Figure 44 shows the entry point to some of the tools that are described now from the Studio GUI previously discussed. Note that you need to enable the "Protected Features" with the password “forestln” in the Studio GUI to see this. See Figure 17. This password should also be used for the Device GUI if prompted for a password.

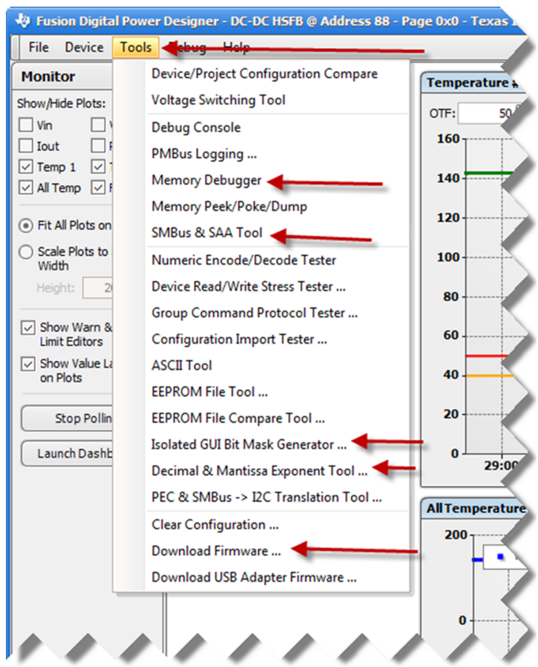


Figure 44. Studio GUI Tools Menu

5.1 Launching Device GUI

During the installation, you had the option to create a shortcut for the UCD3xxx Device GUI. If that option was not selected, the UCD3xxx Device GUI can be accessed from the Start Menu.

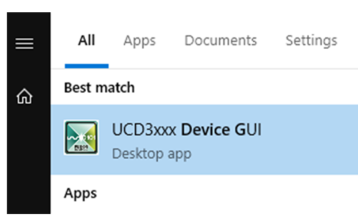


Figure 45. Opening UCD3xxx and UCD9xxx Device GUI

The Device GUI looks like Figure 46.

Copyright © 2013–2019, Texas Instruments Incorporated

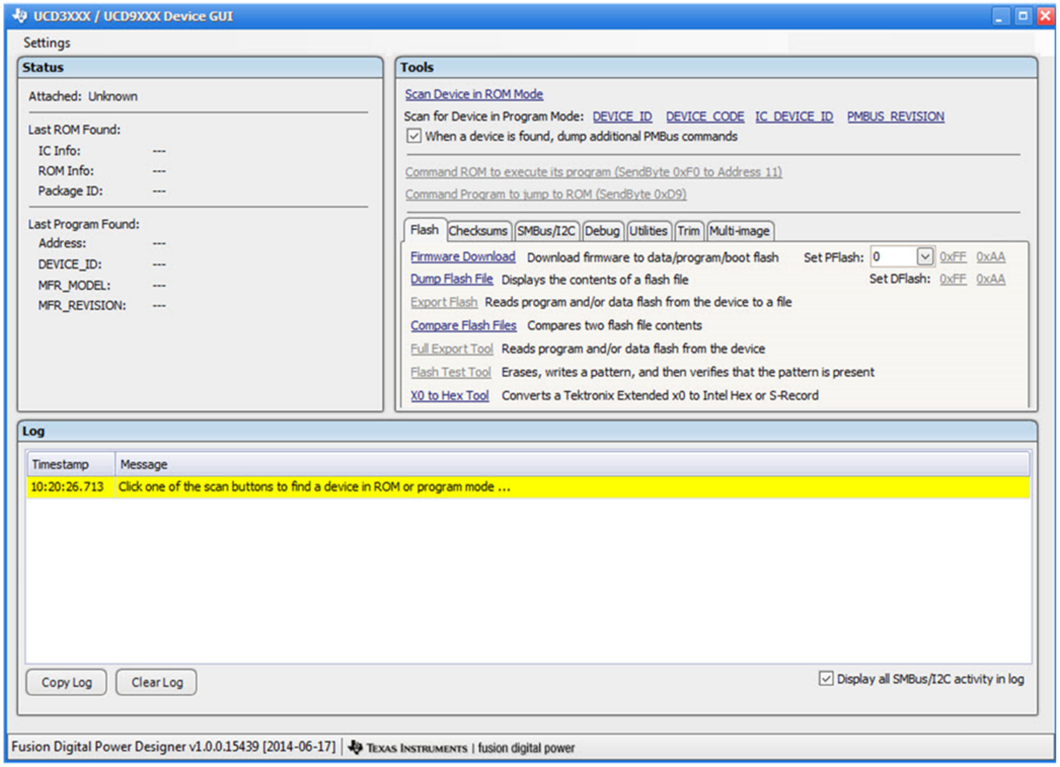


Figure 46. UCD3XXX Device GUI

After the Device GUI starts up, there are a number of links that are enabled and some disabled. Which links are clickable depends on whether the GUI is in ROM mode or Program mode. To start off, you should click “Scan Device in Rom Mode” if the device is in ROM mode. If you click this and the device is not in ROM mode, a message is logged that there is No ROM detected. If the device is in Program mode, then you should select “Device ID” or “PMBus REVISION”.

Copyright © 2013–2019, Texas Instruments Incorporated

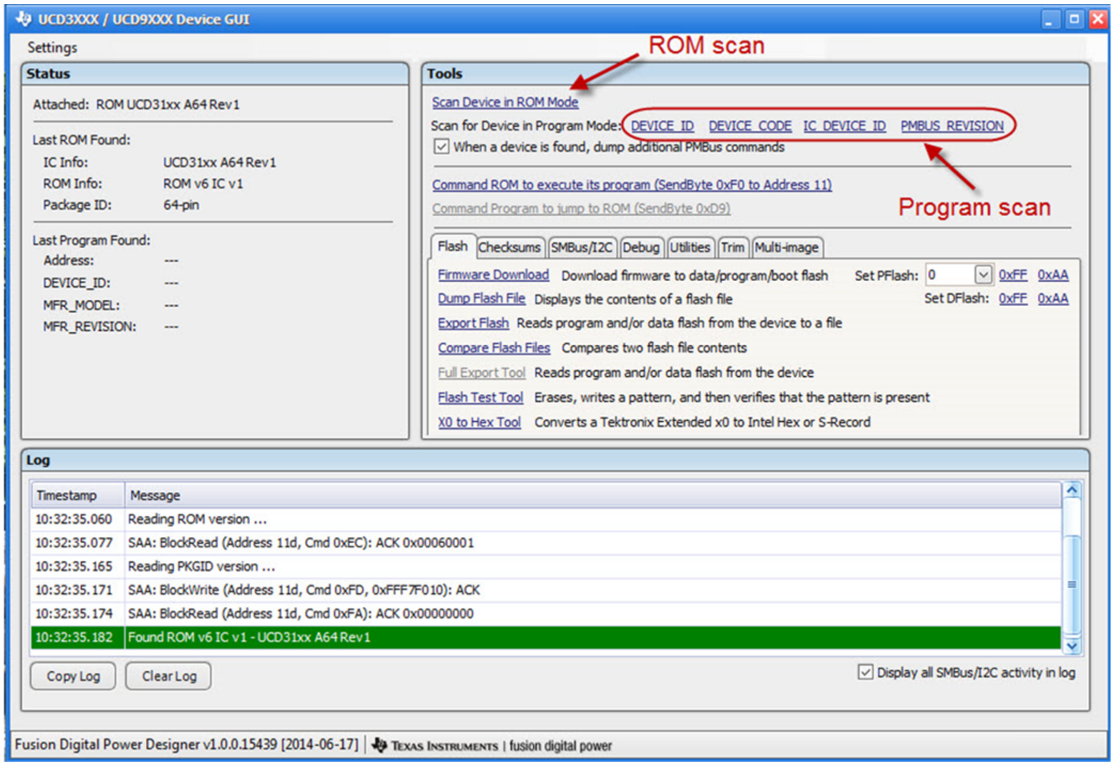


Figure 47. Program Scan and Rom Scan

5.2 Moving Between ROM and Program Mode

To move between ROM mode and Program mode, you can select the following links respectively:

- Command ROM to execute its program (SendByte 0xF0 to Address 11)
- Command Program to jump to ROM (SendByte 0xD9 to Address xx)

Figure 48 displays these links in the Device GUI.

Copyright © 2013–2019, Texas Instruments Incorporated

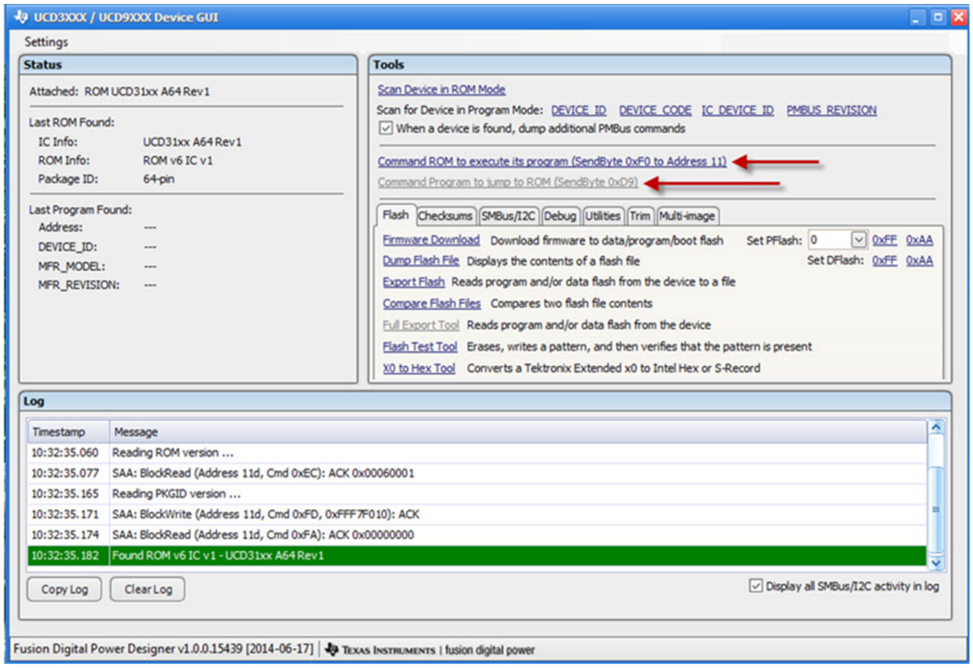


Figure 48. Moving Between ROM Mode to Program Mode

5.2.1 ROM Mode to Program Mode for Multiple Flashes

In devices that have multiple flash blocks, you have more than one option when commanding ROM to execute its program. This applies to devices that allow execution from more than one block. For example, in the UCD3138064, a device with two flash blocks, you would send a different byte depending on which block you wanted to execute. You would send byte 0xF0 to execute Block 0. This would be the same byte to send if the firmware you wanted to run was the size of both blocks. This is due to the address beginning at the same place as Block 0. To execute Block 1, you would send 0xF7. See Figure 49 and Figure 50 showing what to click to send the device from ROM to Program mode. The options for the two blocks appear after scanning for the device in ROM mode.

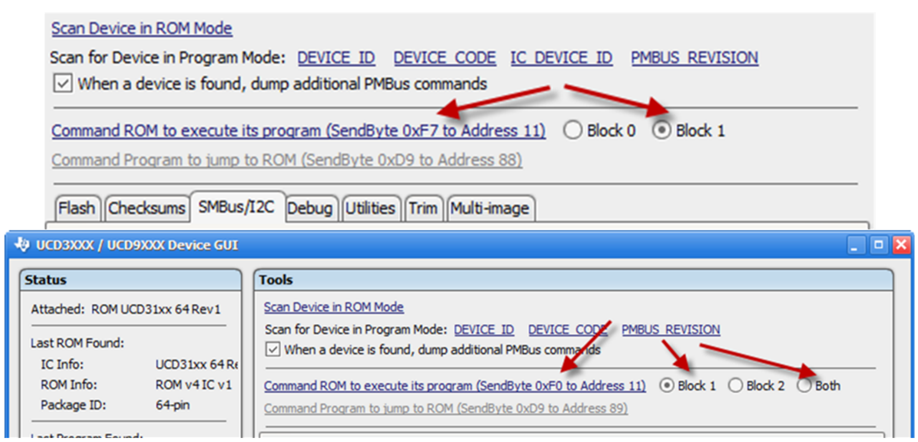


Figure 49. Executing Program for Block 0 (0xF0)

Copyright © 2013–2019, Texas Instruments Incorporated

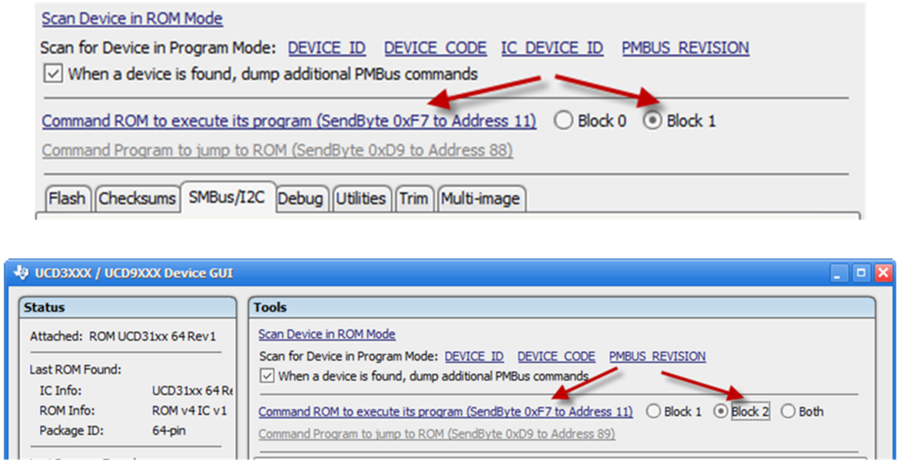


Figure 50. Executing Program for Block 1 (0xF7)

5.3 Firmware Download Tool

To open the Firmware Download tool, click “Firmware Download” as shown in Figure 52.

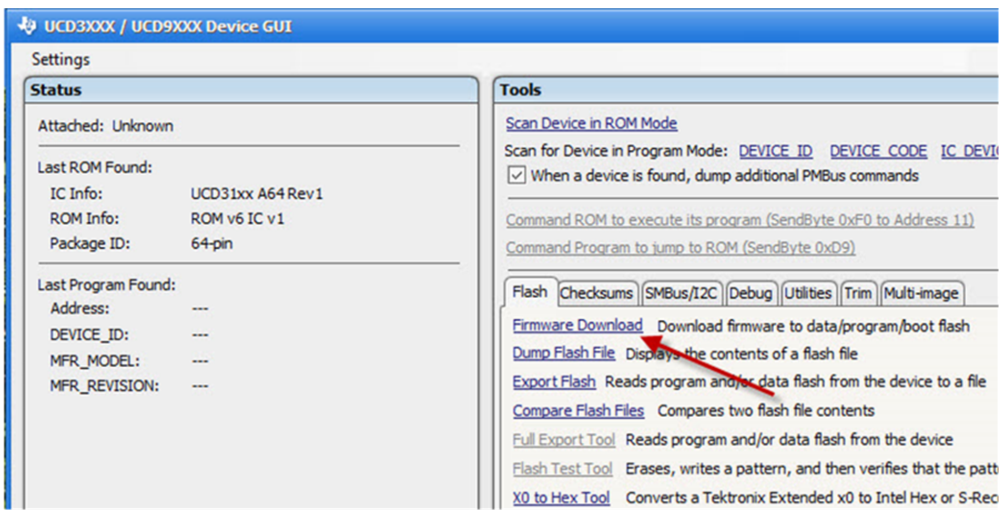


Figure 51. Firmware Download

The firmware download screen launched differs due to the available block configurations specific to each IC.

For UCD3138 the screen looks as follows:

Copyright © 2013–2019, Texas Instruments Incorporated

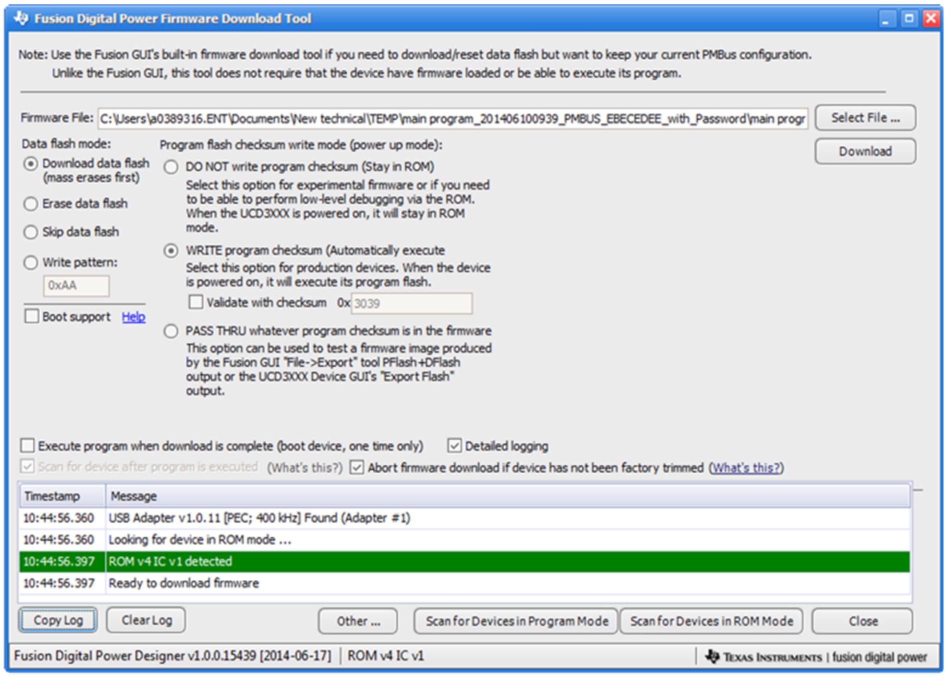


Figure 52. Firmware Download Screen for the UCD3138

For the UCD3138064, notice the flash block selection available:

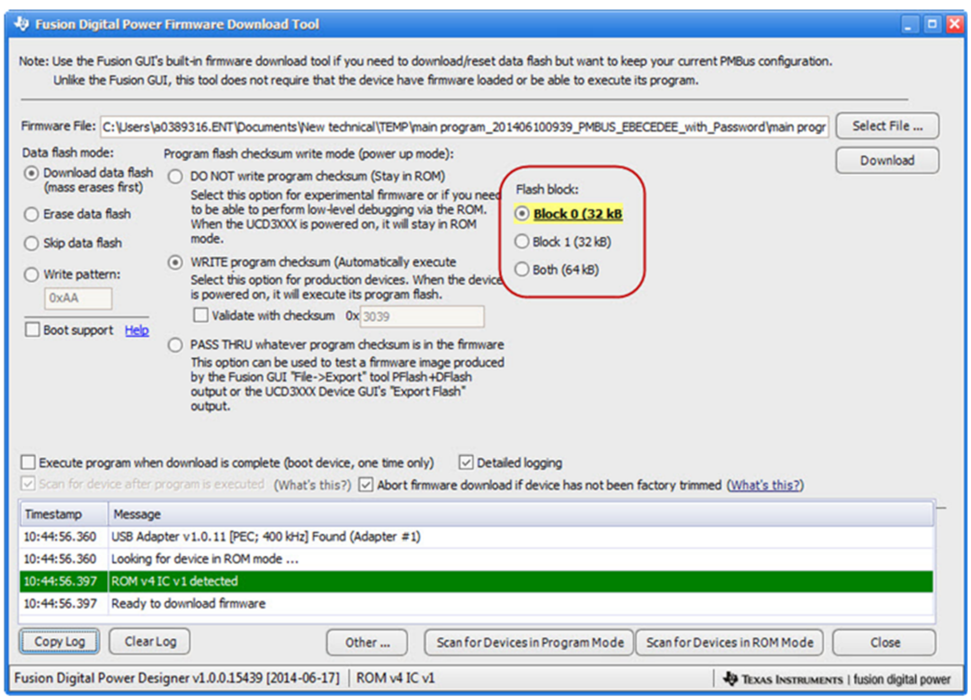


Figure 53. Firmware Download for the UCD3138064

For the UCD3138A64:

Copyright © 2013–2019, Texas Instruments Incorporated

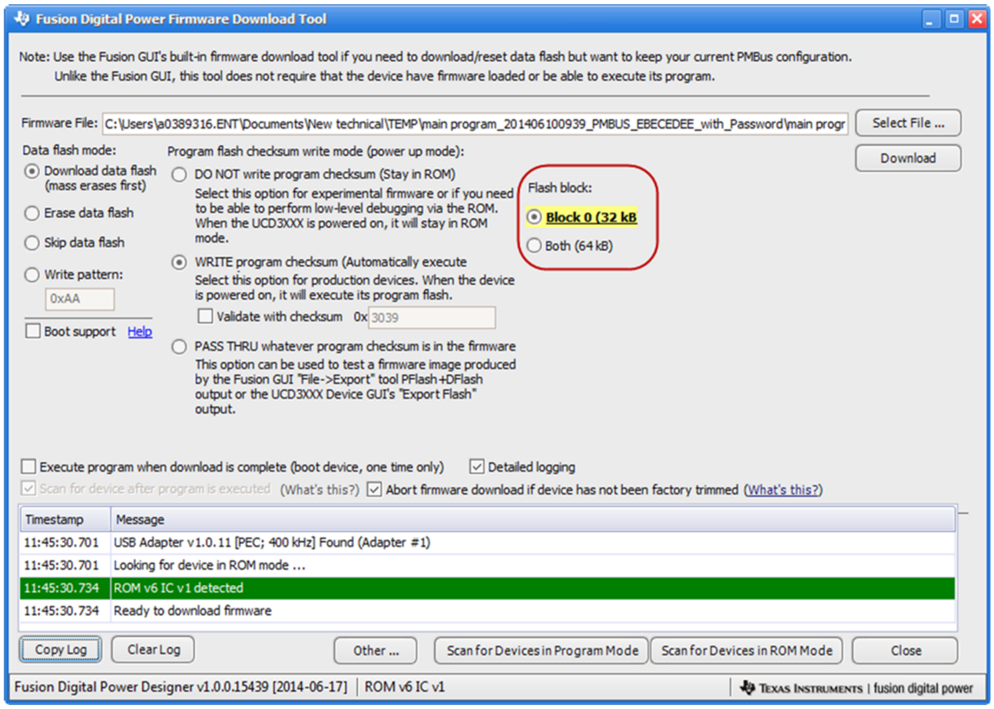


Figure 54. Firmware Download the UCD3138A64

You can choose what they would like to download with regards to the Program Flash, and Data Flash.

WARNING

It is important to note that if the program checksum is written, the device boots up in program mode upon a reset. This may be a source for a device lockup if the firmware has not implemented the commands to jump back to ROM. Hence, it is advised not to write the program checksum for firmware in initial stages or implement the commands to jump back to ROM first.

For devices that have multiple flashes, an extra set of radio buttons appears for you to decide which block to download to as shown in the previous figures.

You pick the firmware file and clicks download.

NOTE: Sometimes this tool may be launched when the device is running in program mode. In that case, you can use the button “Other …” at the bottom to put the device in ROM mode so that they can proceed with the download.

5.3.1 Boot Support

To write firmware to the boot flash click “Boot support” as shown in Figure 56.

Copyright © 2013–2019, Texas Instruments Incorporated

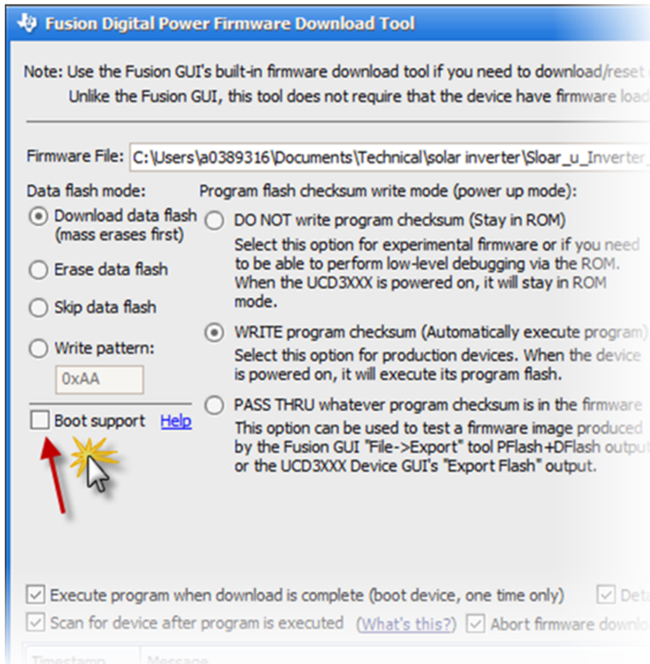


Figure 55. Boot Support

The following screen shows the new options circled below related to boot flash.

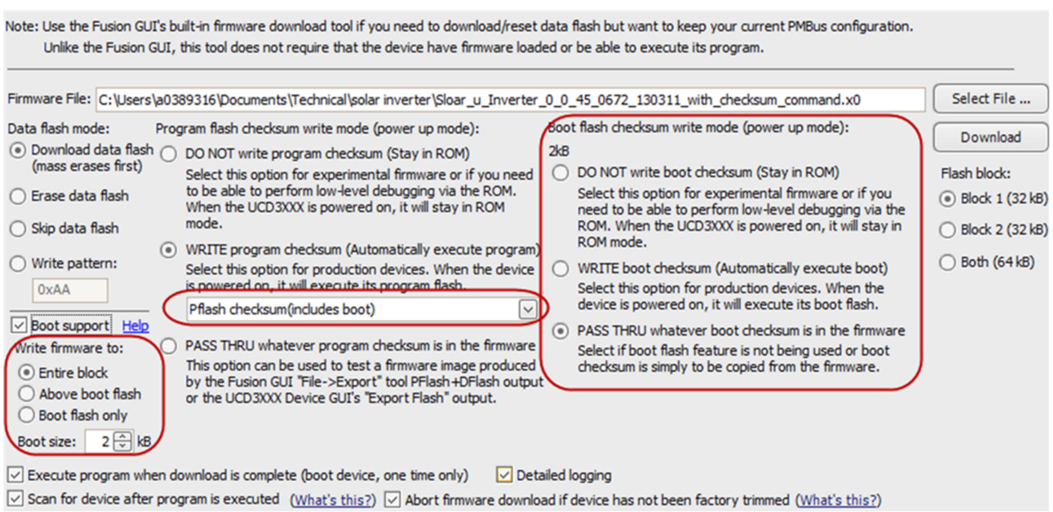


Figure 56. Bootflash Options

Each of the options is described. Figure 56 shows the "Help" screen describing the various options that the firmware can be written to and the checksums related to it.

The first option to configure for Boot Support is "Write firmware to:" as circled in Figure 57.

- Write firmware to "Entire block": The program and the boot is taken from the firmware file.
- Write firmware to "Above boot flash": Only the program is taken from the firmware file.
- Write firmware to "Boot flash only": Only the boot is taken from the firmware file.
- "Boot size": Can range from 2 kB to 31 kB. For a boot size of 2 kB, there is only one option for the boot flash checksum as shown in Figure 57. If the boot size is greater than 2 kB, there is another option to set a checksum for the remainder of the boot flash as shown in Figure 59.

Copyright © 2013–2019, Texas Instruments Incorporated

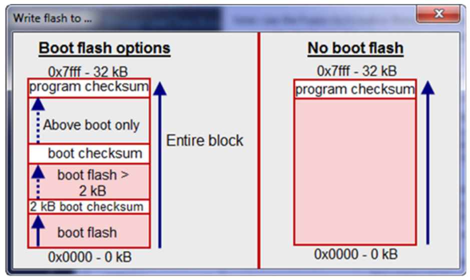


Figure 57. Firmware Writing Options

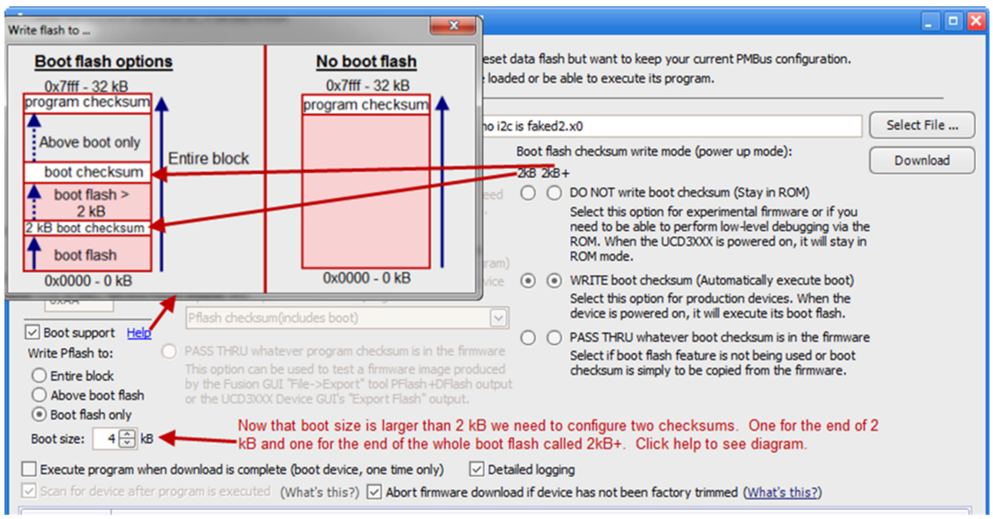


Figure 58. Two Checksums for Boot Flash Greater Than 2 kB

There are two options if you are writing the program checksum after downloading program flash. You need to specify if the checksum calculated should include in addition to the program, the boot or not. See Figure 60.

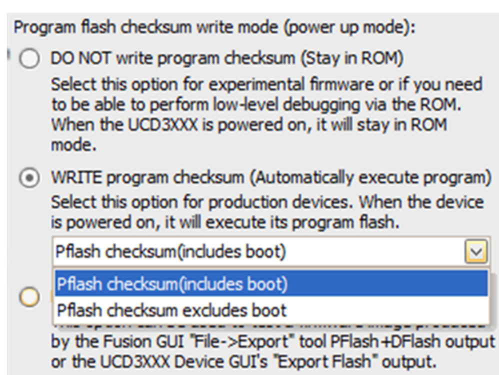


Figure 59. Writing pflash Checksum Options

Copyright © 2013–2019, Texas Instruments Incorporated

5.3.2 Data Flash Download

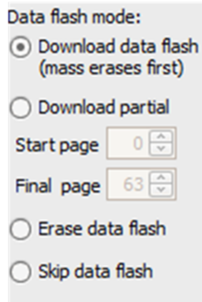


Figure 60. Data Flash Download Options

5.3.2.1 Data Flash Download Options

There are three options regarding downloading of data flash.

5.3.2.1.1 Download

The option “Download data flash” writes the data flash portion defined in the .x0 file to the data flash location on the device. Before the writing of data flash, a mass erase is issued where all the pages are cleared simultaneously.

5.3.2.1.2 Erase

The “Erase data flash” option simply issues the mass erase without downloading the .x0 file.

5.3.2.1.3 Partial Download

The second option is “Download partial.” For this case, you must specify an initial start page index and a final page index of the pages defined in your .x0 you wish to download. The data flash pages outside the range of these indices on the device are not edited.

5.3.2.2 Download Partial Flash Clarification

5.3.2.2.1 Erase Time

Before the continuous set of pages (defined by the start and final page indices) are written, the page erase command is issued sequentially beginning with the “Start page.” This erase is done sequentially, one page at a time, including the appropriate wait time after a page erase has been issued. Therefore, if there are 10 pages and “y” is the wait time per page erase, then the total wait time needed would be 10y. For the first option above, the wait time is only “y”, as the mass erase applies a simultaneous erase to all the pages as opposed to the sequential erase in this option.

5.3.2.2.2 Identifying the Pages

Once the data flash beginning address, and the address of the data variables with their respective data lengths are known then finding the start page index and final page index for a partial download can be found as follows:

Start_page_index = (data_variables_begin_address – data_flash_begin_address)/0x20

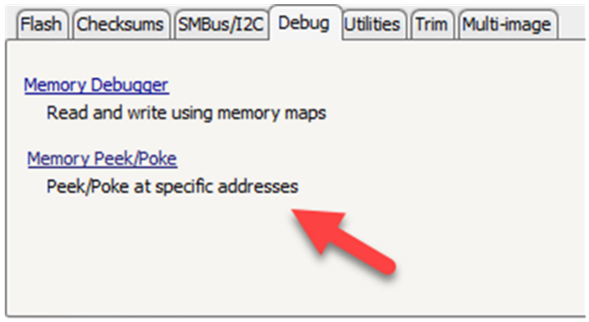
Final_page_index = Start_page_index + (sum_of_data_lengths/0x20) -1

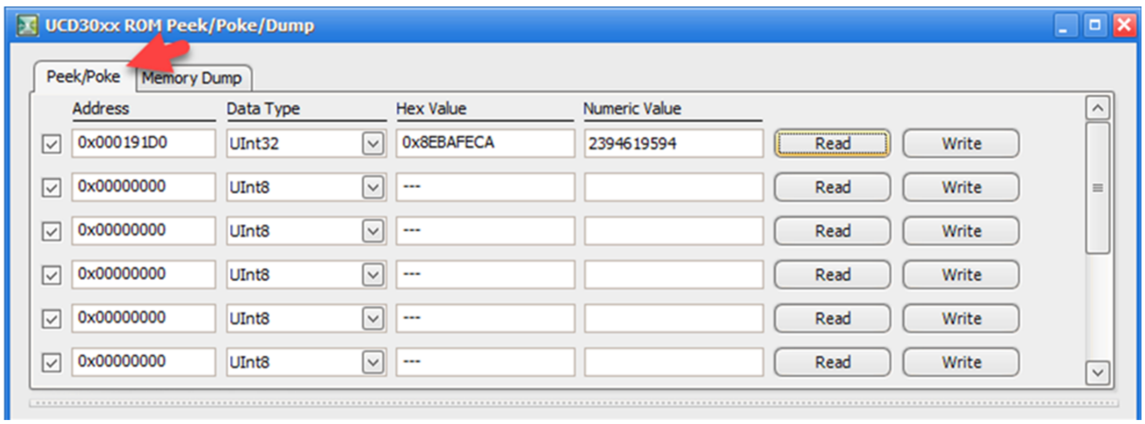
Note: Usually the data that is being partially downloaded to the device is defined in the firmware along page boundaries.

Copyright © 2013–2019, Texas Instruments Incorporated

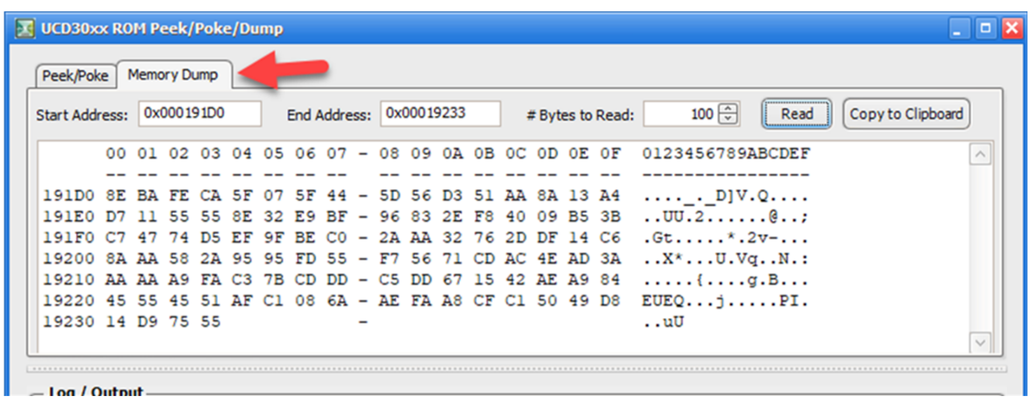
5.3.2.2.3 Helpful Tools

The “Memory Peek/Poke” tool is helpful for observing the flash.





After you specify the begin and end address, you can view the flash contents in the “Memory Dump” tab.



5.4 Checksum Functions

In the Checksums tab, there are a number of functions available to view, calculate, create, validate, and clear checksums on the device as shown below. The tab visually displays the checksums to more easily apply the appropriate function. Depending on the boot flash size or whether boot flash is even needed, the visualization of the checksums updates as shown in Figure 61 and Figure 62.

Copyright © 2013–2019, Texas Instruments Incorporated

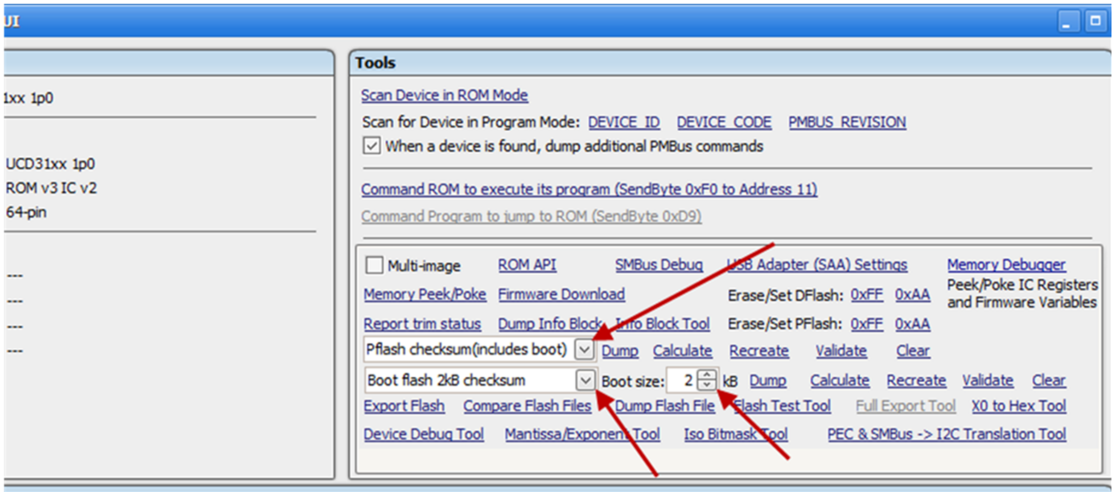


Figure 61. Checksum Functions

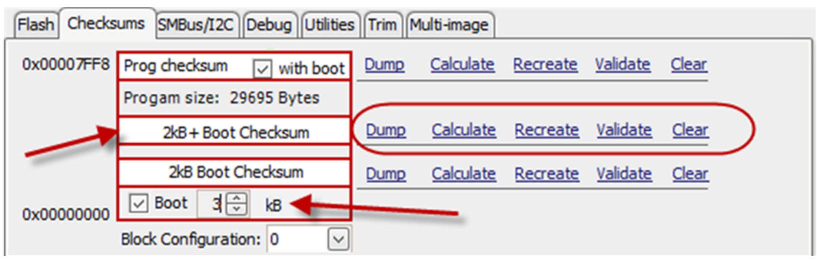


Figure 62. 2 kB Boot Checksum Functions

5.5 Multi-image Functions

Click the Multi-image tab to use functions for firmware that implement multiple images. See Figure 64.

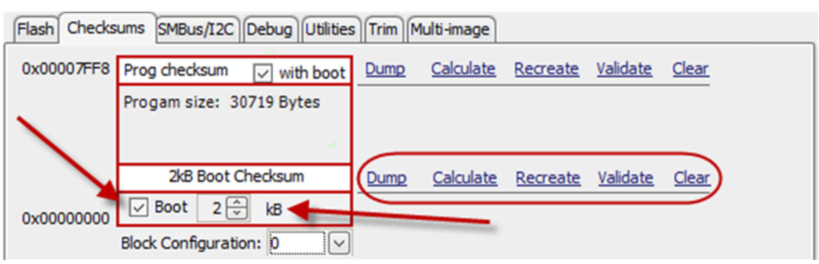


Figure 63. 2 kB+ Boot Checksum Functions When Boot is Greater Than 2 kB

Copyright © 2013–2019, Texas Instruments Incorporated

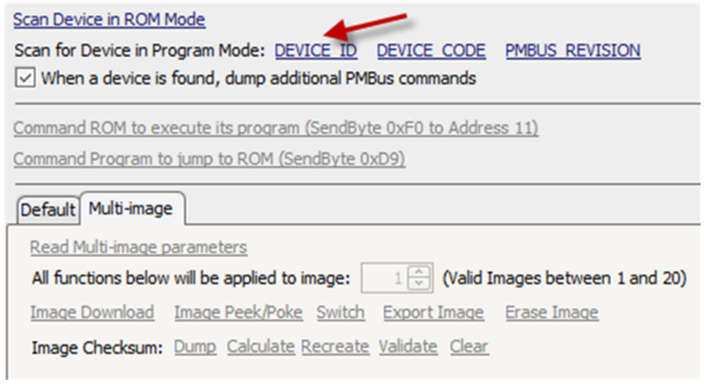


Figure 64. Scan for Device ID to Activate Multi-image

The Multi-image tab provides functions for working with other images while an image is executing. After scanning for "Device ID" as shown in Figure 64, you see the link "Read Multi-image parameters" become enabled. Click this to read important parameters that describe the images and how the GUI interacts with them as shown in Figure 65 and Figure 66.

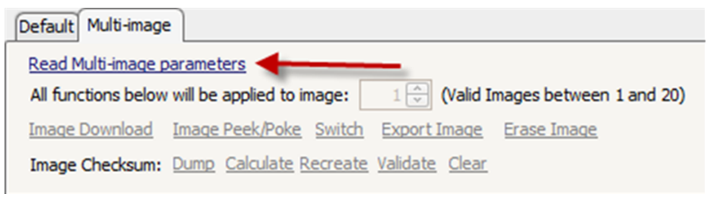


Figure 65. Click Read Multi-image Parameters to Activate Functions

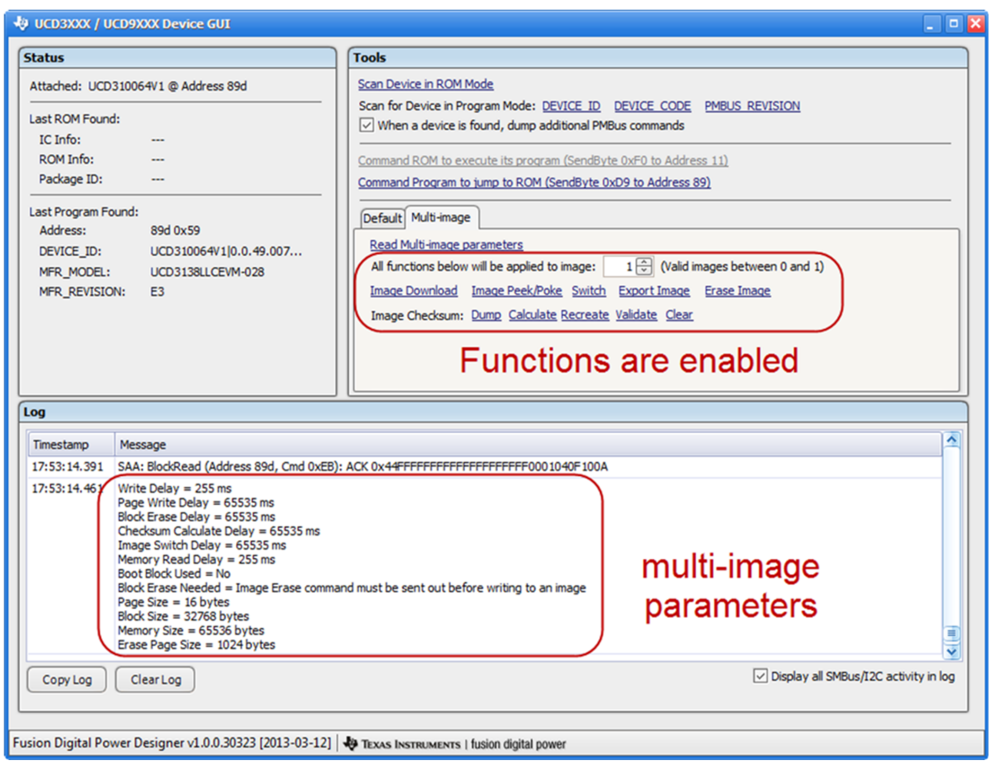


Figure 66. Functions Enabled After Reading Multi-image Parameters

Copyright © 2013–2019, Texas Instruments Incorporated

The following sections are descriptions of the functions for multi images.

5.5.1 Setting Image Index

Before using any of the functions shown in Figure 66, you must set which image index you are working with.

5.5.2 Multi-image Download

After setting the appropriate image index and clicking "Image Download", Figure 67 is displayed.

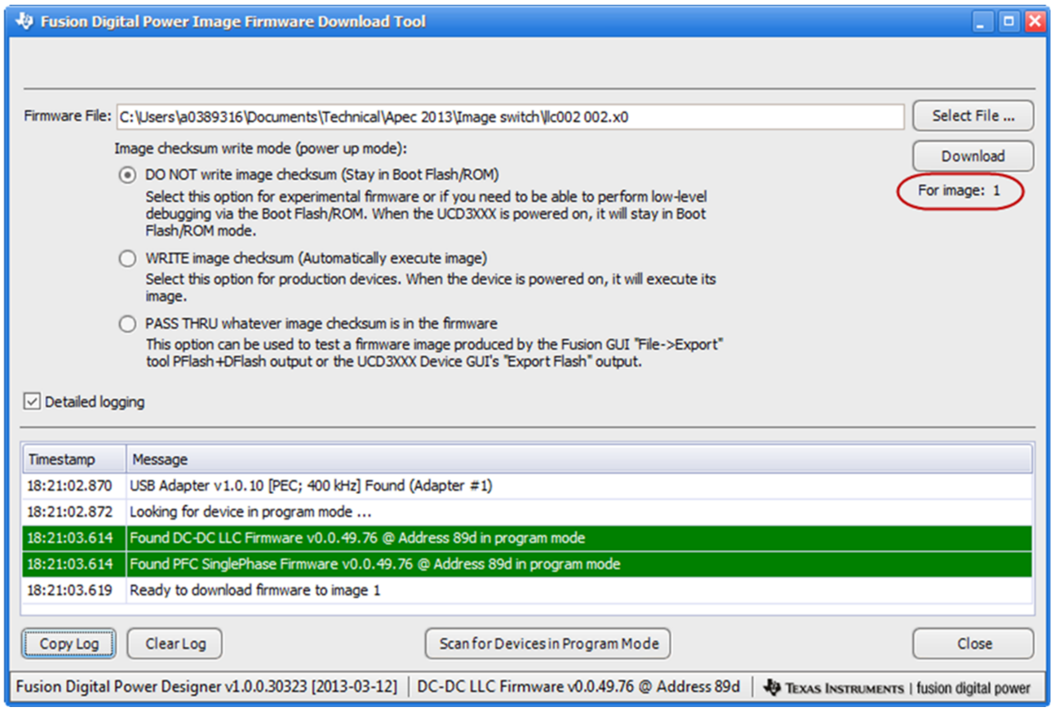


Figure 67. Image Download

5.5.3 Switch

In order to activate the downloaded image, you need to click “Switch”. See Figure 68.

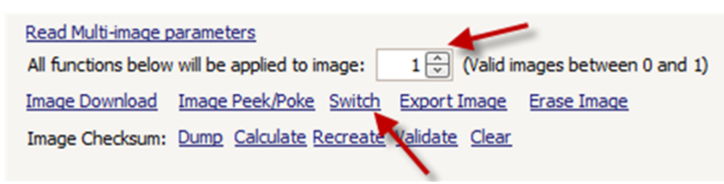


Figure 68. Image to Switch to

5.5.4 Image Peek/Poke/Dump

You can specify which address to read/write to as shown in Figure 69 and Figure 70.

Copyright © 2013–2019, Texas Instruments Incorporated

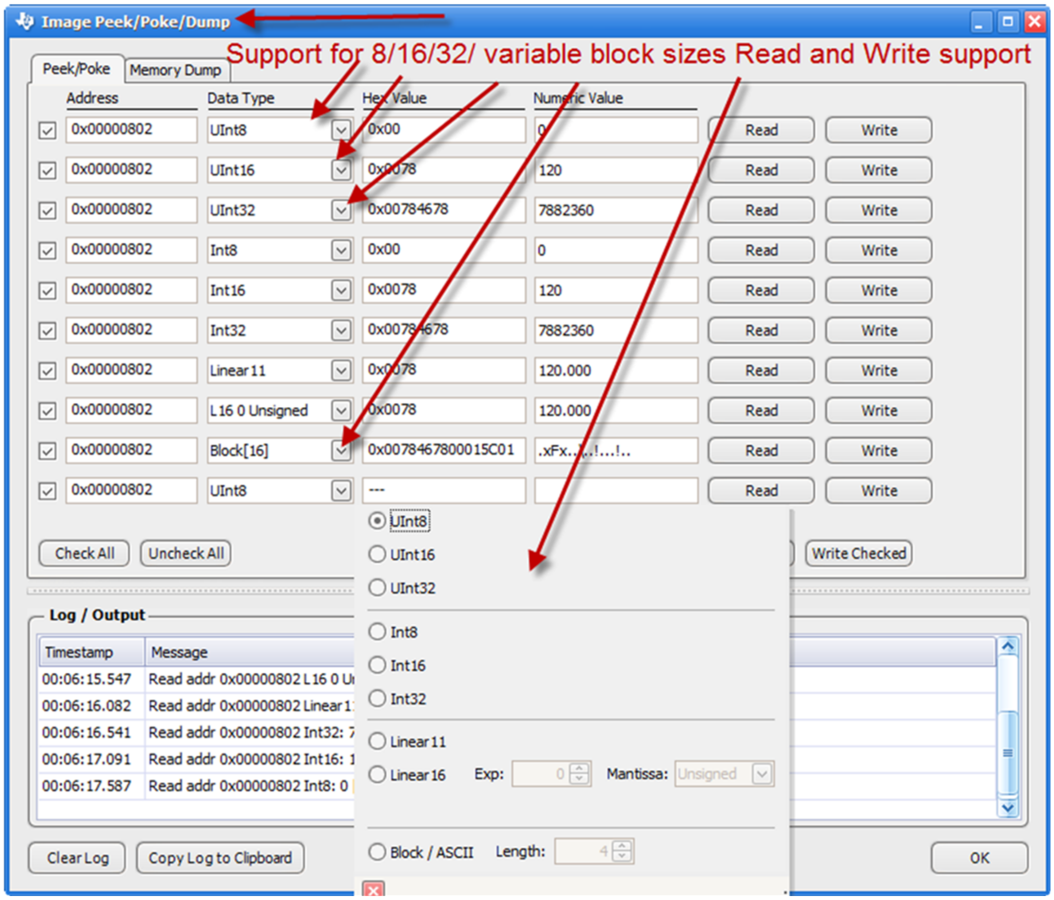


Figure 69. Image Peek/Poke

Copyright © 2013–2019, Texas Instruments Incorporated

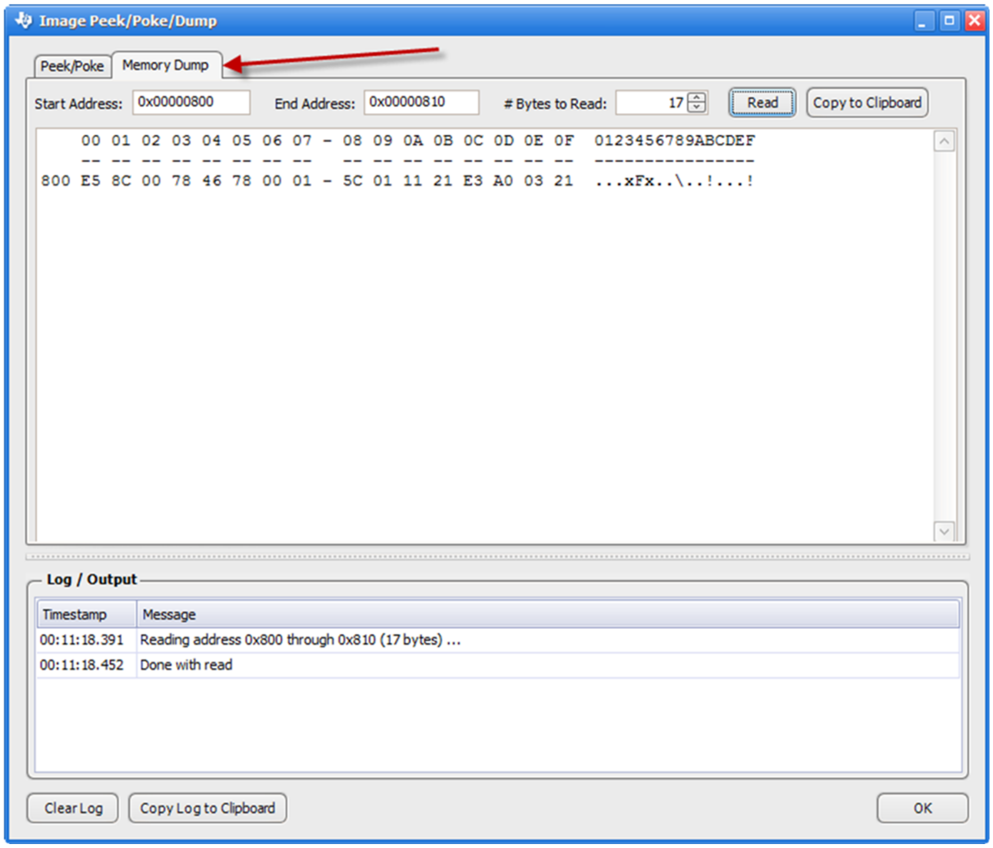


Figure 70. Image Dump

5.5.5 Erase Image

Click Erase image to send the firmware command to erase the image selected as shown in Figure 71.

Copyright © 2013–2019, Texas Instruments Incorporated

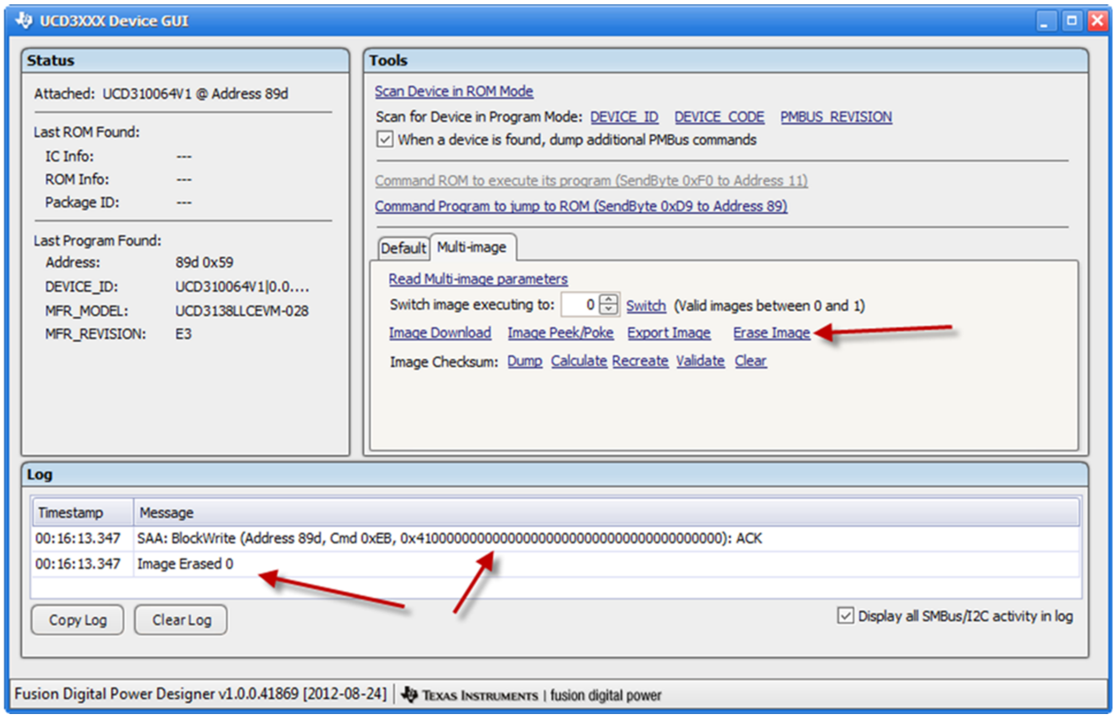


Figure 71. Erase Image

5.5.6 Export Image

You can also export the image currently on the device.

5.5.7 Image Checksums

The following pictures illustrate the image checksum commands.

5.5.7.1 Calculate Image Checksum

To calculate a checksum based on the image selected, click “Calculate.”

Copyright © 2013–2019, Texas Instruments Incorporated

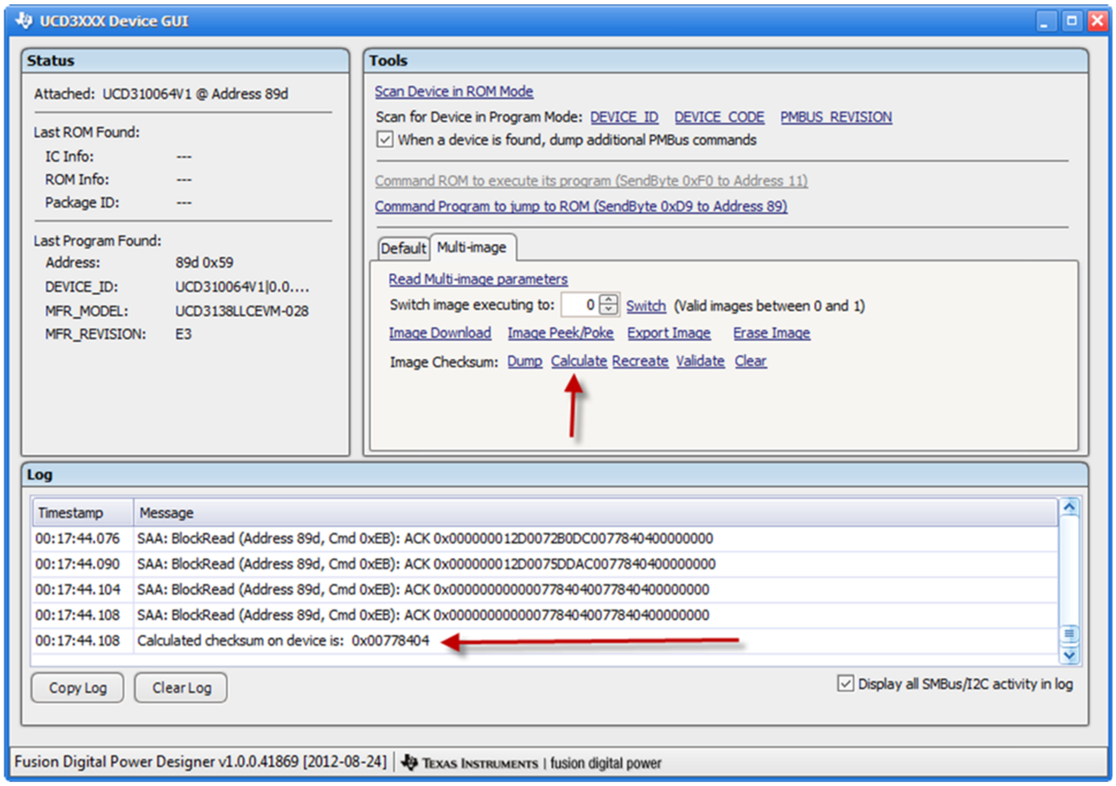


Figure 72. Calculate Image Checkum

5.5.7.2 *Dump Image Checksum*

To display the last written checksum or bytes currently in the location of where the image checksum would be, click “Dump.”

Copyright © 2013–2019, Texas Instruments Incorporated

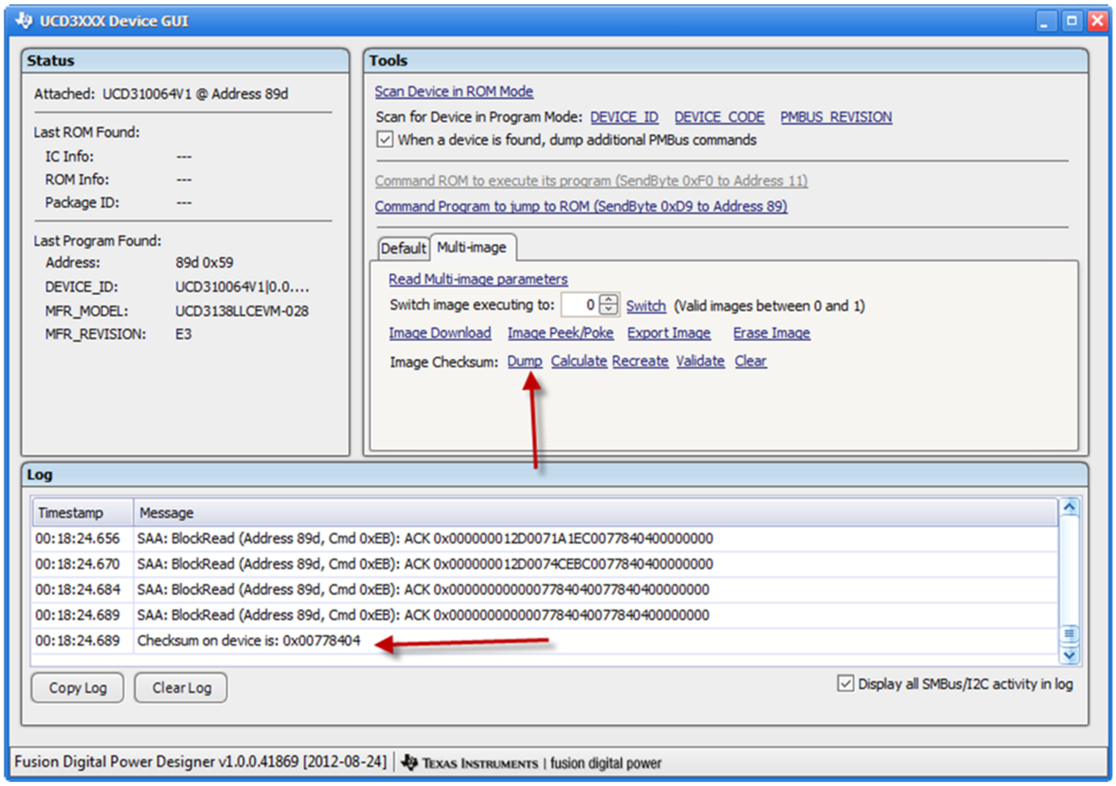


Figure 73. Dump Image Checksum

5.5.7.3 Create Image Checksum

To create a checksum in the checksum location for the image selected, click "Recreate."

Copyright © 2013–2019, Texas Instruments Incorporated

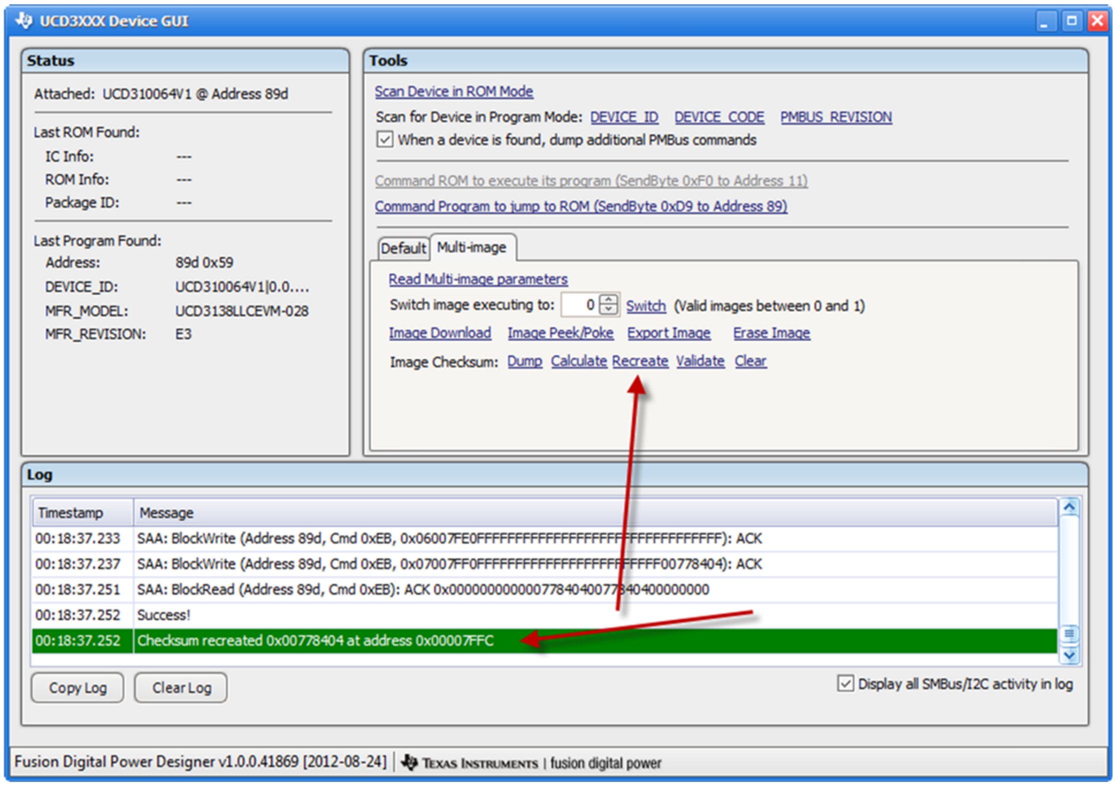


Figure 74. Recreate Image Checksum

5.5.7.4 *Validate Image Checksum*

To validate that the calculated checksum equals the dump checksum, click “Validate.”

Copyright © 2013–2019, Texas Instruments Incorporated

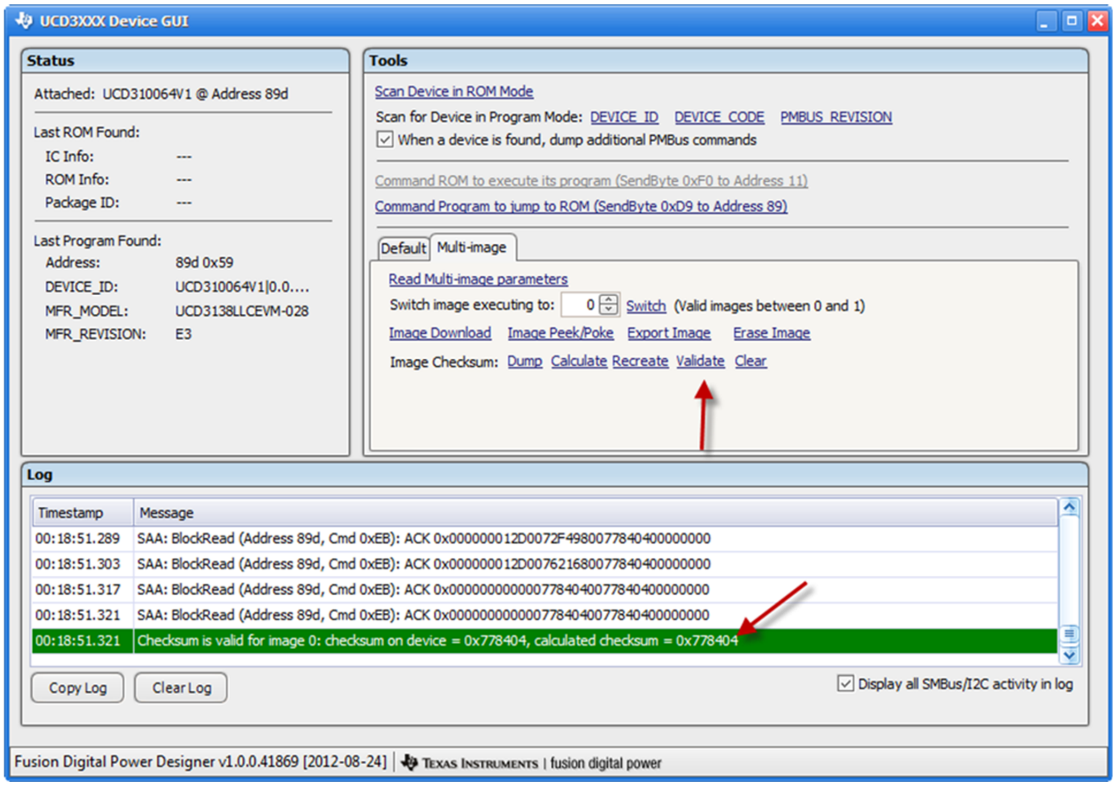


Figure 75. Validating Image Checksum

5.5.7.5 Clear Image Checksum

To clear the checksum for the image selected, click “Clear” and 0xFFFFFFFF is written to that location.

Copyright © 2013–2019, Texas Instruments Incorporated

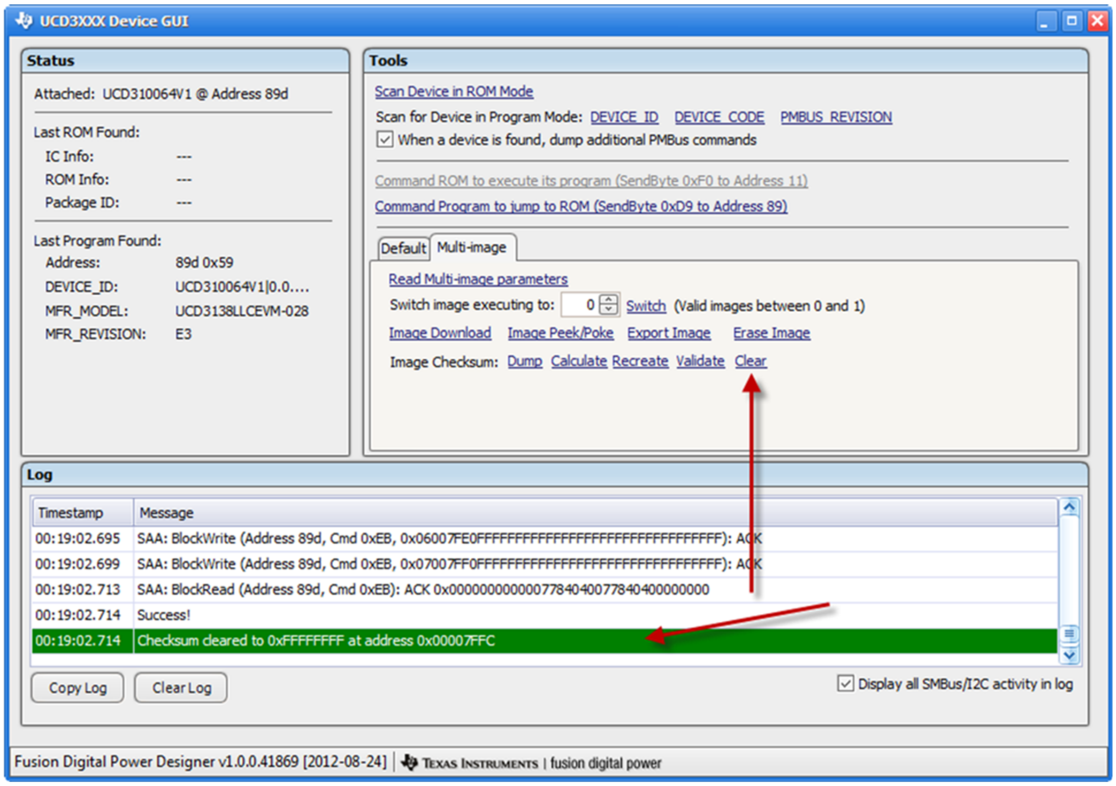


Figure 76. Clearing Image Checksum

5.6 Isolated Bitmask Tool

The Isolated Bitmask Tool provides firmware developers with a tool to help them set the bitmask for the commands that inform the GUI of what PMBus commands are supported. See Section 4.2.8.2.

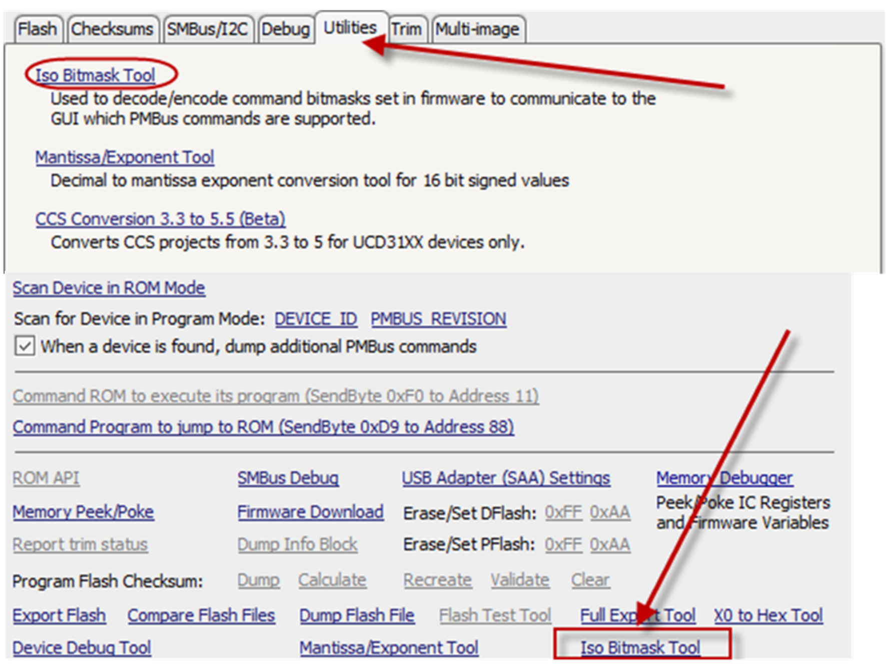


Figure 77. Click Iso Bitmask Tool

Copyright © 2013–2019, Texas Instruments Incorporated

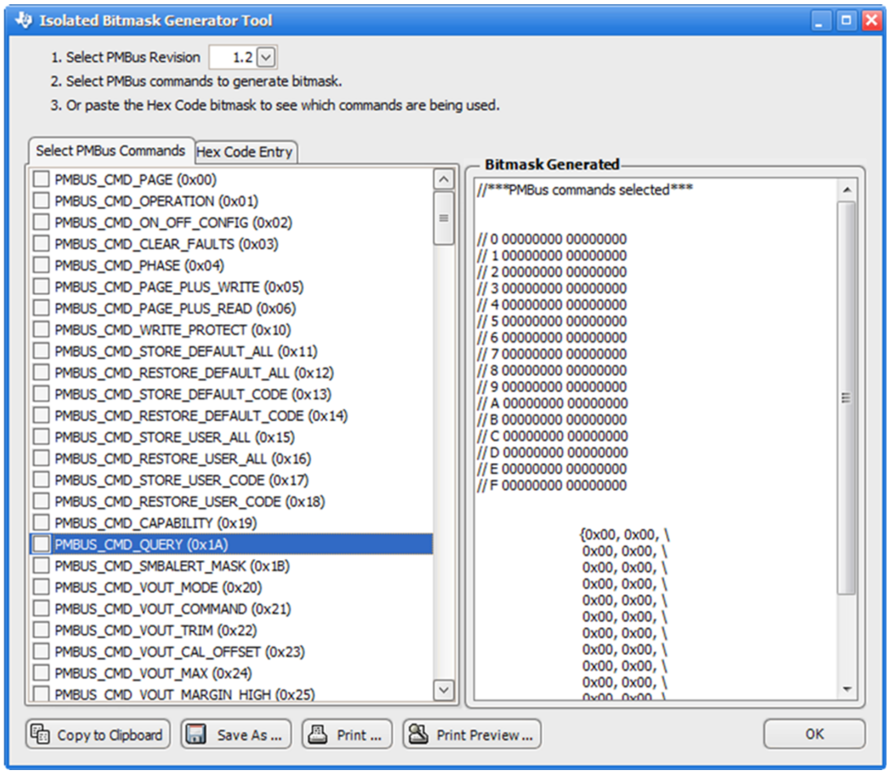
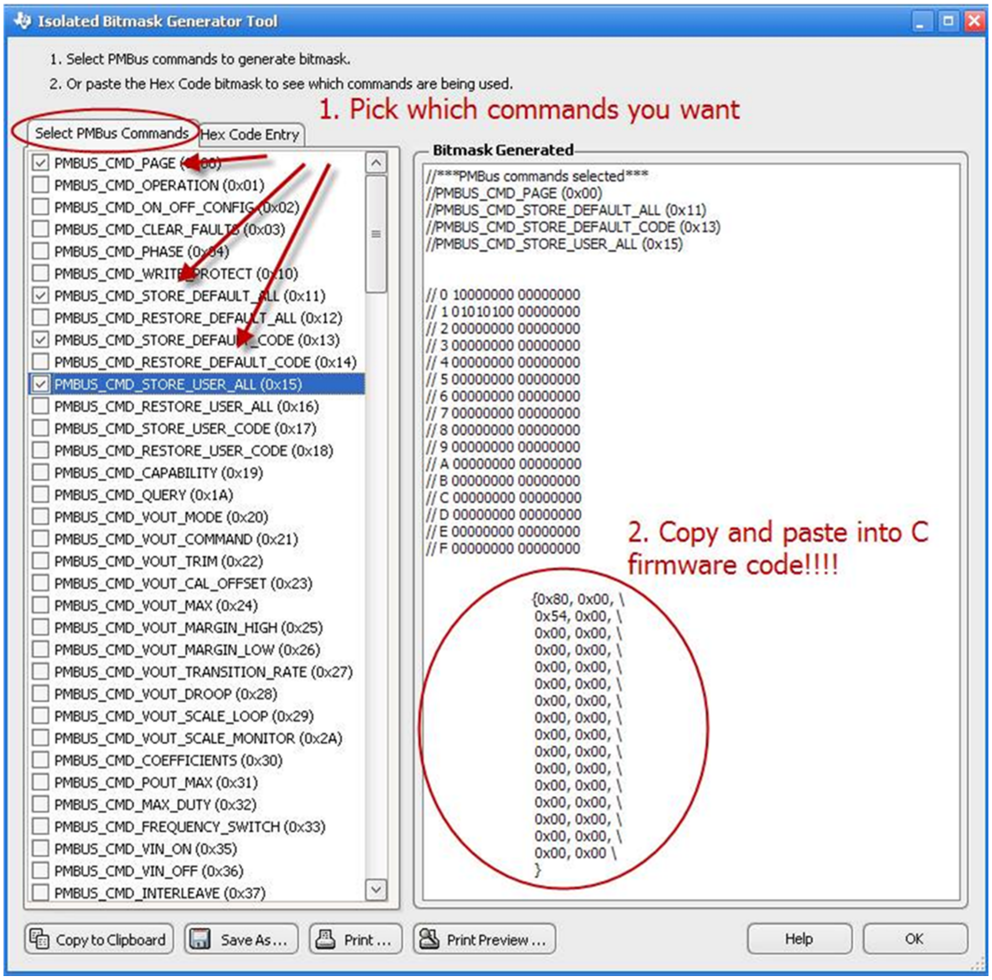


Figure 78. Bitmask Tool

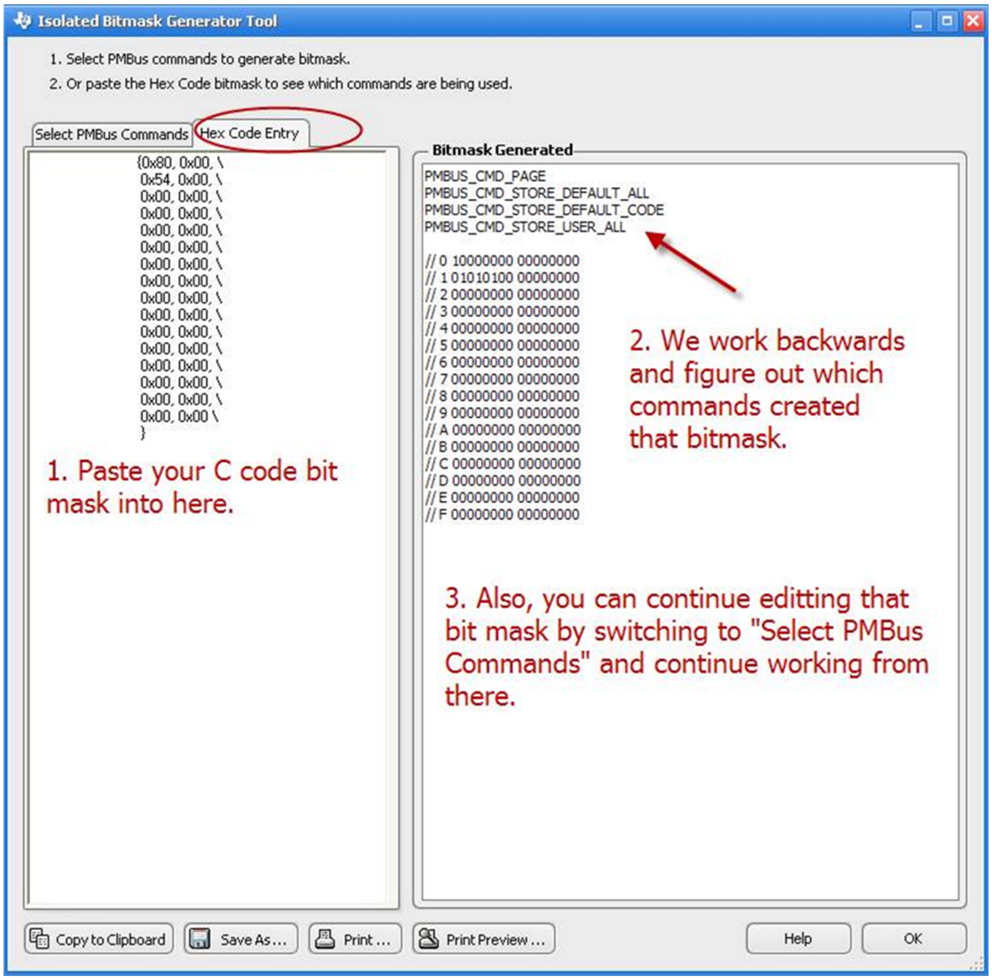
Select commands desired in the bitmask and the bitmask code on the right is automatically generated.

Copyright © 2013–2019, Texas Instruments Incorporated



You can also work in reverse by pasting a known bitmask in C code and then see what commands those bitmasks were indicating. You can also go back to the Select PMBus commands tab and all the indicated ones are checked.

Copyright © 2013–2019, Texas Instruments Incorporated



5.7 Firmware Memory Debugger

Included with the Fusion Digital Power Design software suite is a powerful low level GUI is available for debug using the PMBus. Click the Debug tab and click Memory Debugger.

Copyright © 2013–2019, Texas Instruments Incorporated

Figure 79. Memory Debugger

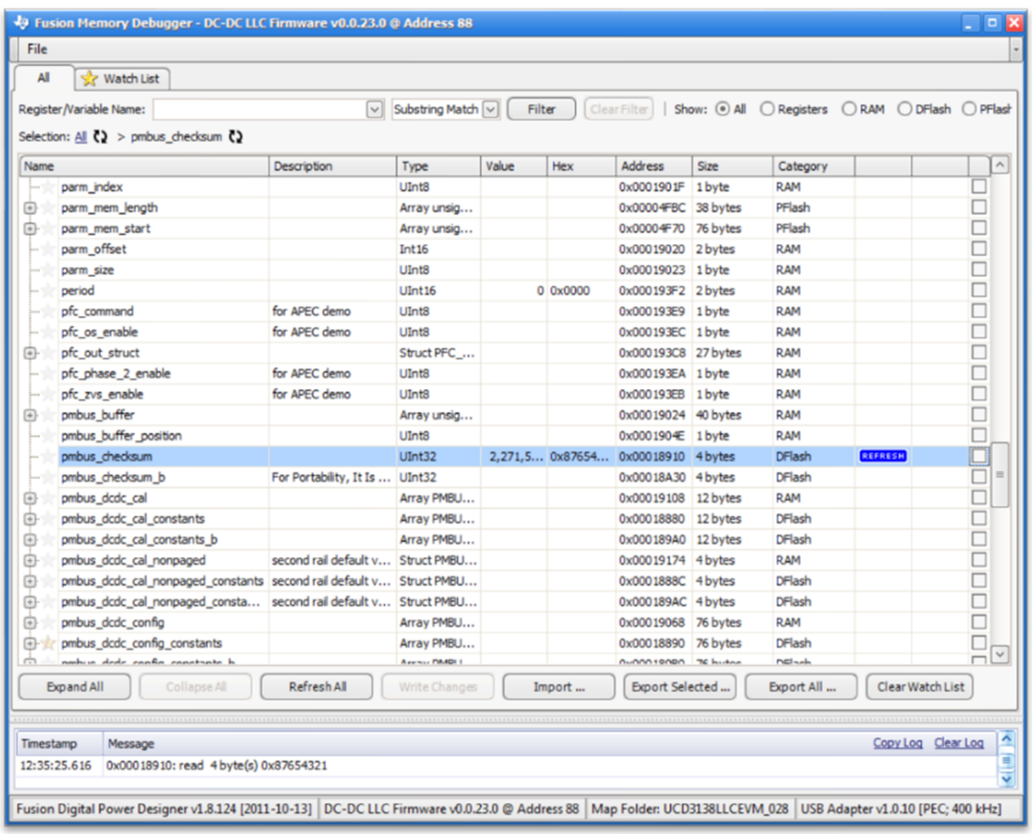


Figure 80. GUI Debugger

Copyright © 2013–2019, Texas Instruments Incorporated

To also access the GUI through the Design GUI, click the “Memory Debugger” item under tools, shown in Figure 81.

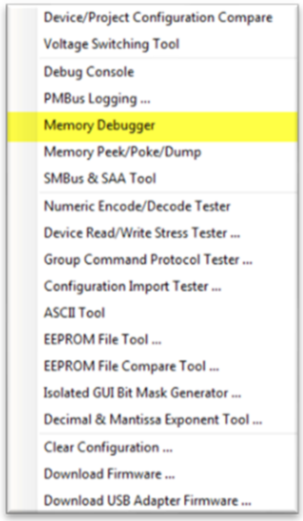
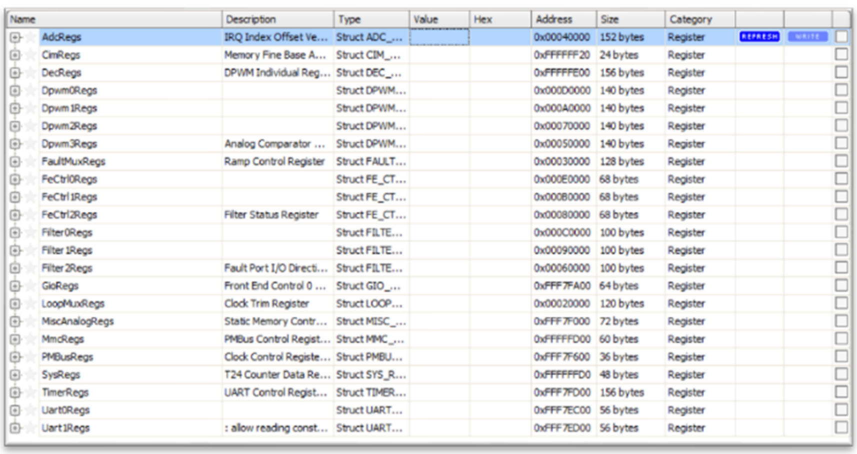


Figure 81. Fusion Studio GUI Debugger Tool

By default, the tool comes up displaying all of the hardware-based device registers.

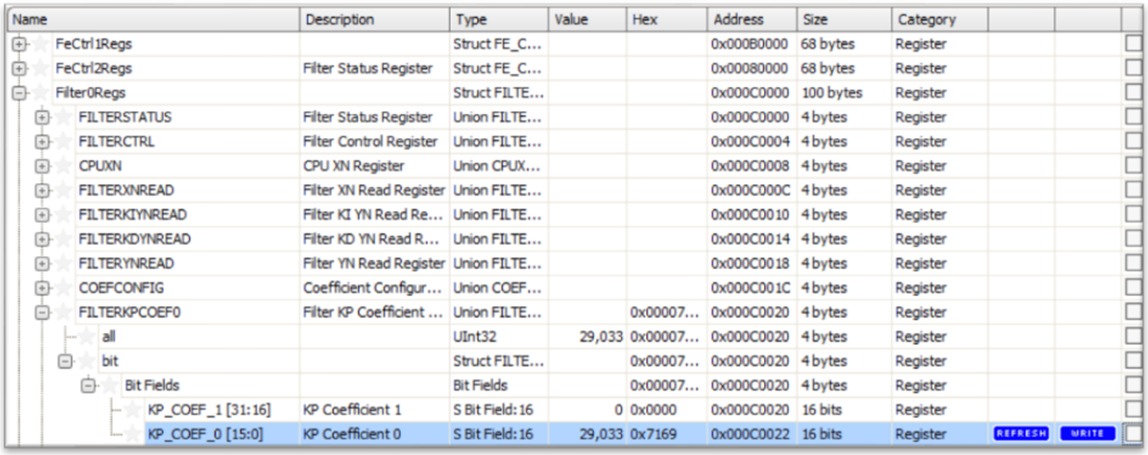


Name	Description	Type	Value	Hex	Address	Size	Category			
AdcRegs	IRQ Index Offset Ve...	Struct ADC_...			0x00040000	152 bytes	Register	REFRESH	WRITE	
CimRegs	Memory Fine Base A...	Struct CIM_...			0xFFFFFF20	24 bytes	Register			
DecRegs	DPWM Individual Reg...	Struct DEC_...			0xFFFFFE00	156 bytes	Register			
Dpwm0Regs		Struct DPWM...			0x000D0000	140 bytes	Register			
Dpwm1Regs		Struct DPWM...			0x000A0000	140 bytes	Register			
Dpwm2Regs		Struct DPWM...			0x00070000	140 bytes	Register			
Dpwm3Regs	Analog Comparator ...	Struct DPWM...			0x00050000	140 bytes	Register			
FaultMuxRegs	Ramp Control Register	Struct FAULT...			0x00030000	128 bytes	Register			
FeCtrl0Regs		Struct FE_CT...			0x000E0000	68 bytes	Register			
FeCtrl1Regs		Struct FE_CT...			0x000B0000	68 bytes	Register			
FeCtrl2Regs	Filter Status Register	Struct FE_CT...			0x00080000	68 bytes	Register			
Filter0Regs		Struct FILTE...			0x000C0000	100 bytes	Register			
Filter1Regs		Struct FILTE...			0x00090000	100 bytes	Register			
Filter2Regs	Fault Port I/O Directi...	Struct FILTE...			0x00060000	100 bytes	Register			
GioRegs	Front End Control 0 ...	Struct GIO_...			0xFFF7FA00	64 bytes	Register			
LoopMuxRegs	Clock Trim Register	Struct LOOP...			0x00020000	120 bytes	Register			
MiscAnalogRegs	Static Memory Contr...	Struct MISC_...			0xFFF7F000	72 bytes	Register			
MmcRegs	PMBus Control Regist...	Struct MMC_...			0xFFFFFD00	60 bytes	Register			
PMBusRegs	Clock Control Registe...	Struct PMBU...			0xFFF7F600	36 bytes	Register			
SysRegs	T24 Counter Data Re...	Struct SYS_R...			0xFFFFFFD0	48 bytes	Register			
TimerRegs	UART Control Regist...	Struct TIMER...			0xFFF7FD00	156 bytes	Register			
Uart0Regs		Struct UART...			0xFFF7EC00	56 bytes	Register			
Uart1Regs	: allow reading const...	Struct UART...			0xFFF7ED00	56 bytes	Register			

Figure 82. GUI UCD3138 Debugger – Defaults

If you expand any item on this list, you have access to every bit field inside the UCD3138 device. This access extends to both reading and writing to these registers.

Copyright © 2013–2019, Texas Instruments Incorporated



Name	Description	Type	Value	Hex	Address	Size	Category		
FeCtrl1Regs		Struct FE_C...			0x000B0000	68 bytes	Register		
FeCtrl2Regs	Filter Status Register	Struct FE_C...			0x00080000	68 bytes	Register		
Filter0Regs		Struct FILTE...			0x000C0000	100 bytes	Register		
FILTERSTATUS	Filter Status Register	Union FILTE...			0x000C0000	4 bytes	Register		
FILTERCTRL	Filter Control Register	Union FILTE...			0x000C0004	4 bytes	Register		
CPUXN	CPU XN Register	Union CPUX...			0x000C0008	4 bytes	Register		
FILTERXNREAD	Filter XN Read Register	Union FILTE...			0x000C000C	4 bytes	Register		
FILTERKIYNREAD	Filter KI YN Read Re...	Union FILTE...			0x000C0010	4 bytes	Register		
FILTERKDYNREAD	Filter KD YN Read R...	Union FILTE...			0x000C0014	4 bytes	Register		
FILTERYNREAD	Filter YN Read Register	Union FILTE...			0x000C0018	4 bytes	Register		
COEFCONFIG	Coefficient Configur...	Union COEF...			0x000C001C	4 bytes	Register		
FILTERKPCOEF0	Filter KP Coefficient ...	Union FILTE...		0x00007...	0x000C0020	4 bytes	Register		
all		UInt32	29,033	0x00007...	0x000C0020	4 bytes	Register		
bit		Struct FILTE...		0x00007...	0x000C0020	4 bytes	Register		
Bit Fields		Bit Fields		0x00007...	0x000C0020	4 bytes	Register		
KP_COEF_1 [31:16]	KP Coefficient 1	S Bit Field:16	0	0x0000	0x000C0020	16 bits	Register		
KP_COEF_0 [15:0]	KP Coefficient 0	S Bit Field:16	29,033	0x7169	0x000C0022	16 bits	Register	REFRESH	WRITE

Figure 83. Device Debugger Bit Field Selector

Figure 83 displays one register set fully expanded in the debugger. Clicking the “REFRESH” button on the right forces the debugger to read the corresponding register from the device. Entering a new value in the “Value” or “Hex” fields and then clicking “WRITE” writes the new values to the device. Keep in mind that reading and writing to any register in the device is very powerful and also dangerous. Some registers should not be changed and others are cleared on read so care should be used when selecting which registers you want to access. See the appropriate programmer manual for further details.

Since there are so many different fields inside of the UCD3xxx devices, a “Watch List” is available to create a convenient place to both read and write to the addresses of interest. Clicking one of the stars next to a variable name turns it gold indicating that it has been added to the watch list. To remove an item from the watch list, simply click the star again. Clicking the “Watch List” tab at the top of the window now displays the selected.

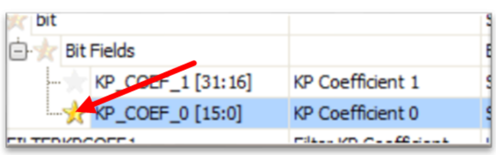


Figure 84. Watch List Selection Star

The debugger also has the ability to read and write to any global firmware variable. This can be done by providing the GUI with the path to find the “.map” and “.pp” files from the firmware build. Click the item shown in Figure 85.

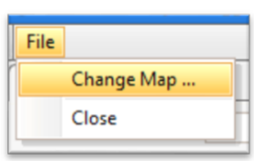


Figure 85. Map File Selection

After clicking this item, a window pops up providing detailed instruction on what to do. For an example, see Figure 86.

Copyright © 2013–2019, Texas Instruments Incorporated

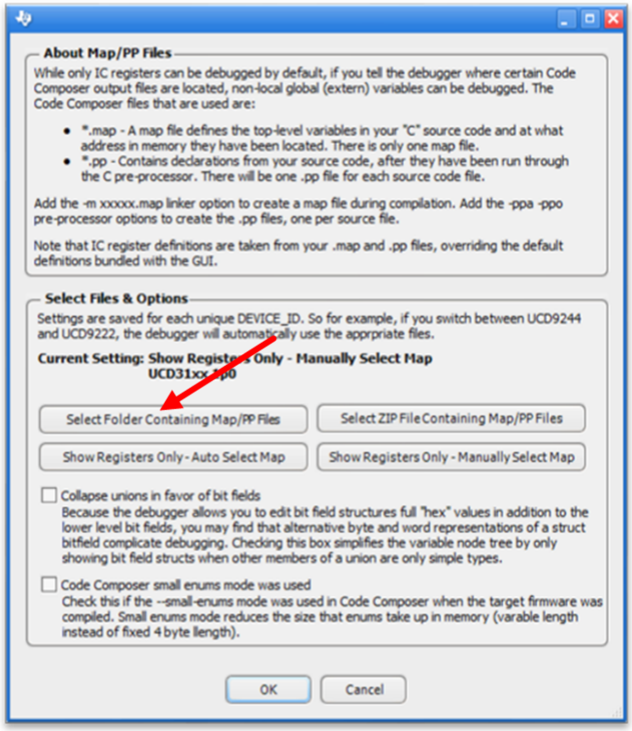


Figure 86. Debugger Customization Tool

The creation of the ".pp" files can be configured by modifying the Code Composer build options as shown in Figure 87.

Copyright © 2013–2019, Texas Instruments Incorporated

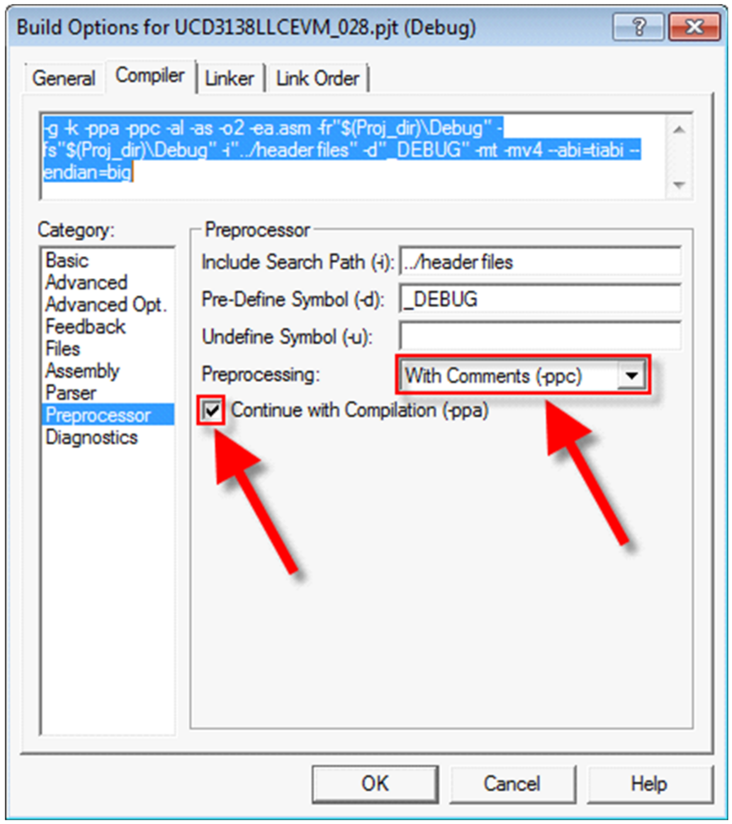


Figure 1 - “.pp” Generation Parameters

Figure 87. “.pp” Generation Parameters

The “*.map” file name and location can be specified in the code compose build options as shown in Figure 88.

Copyright © 2013–2019, Texas Instruments Incorporated

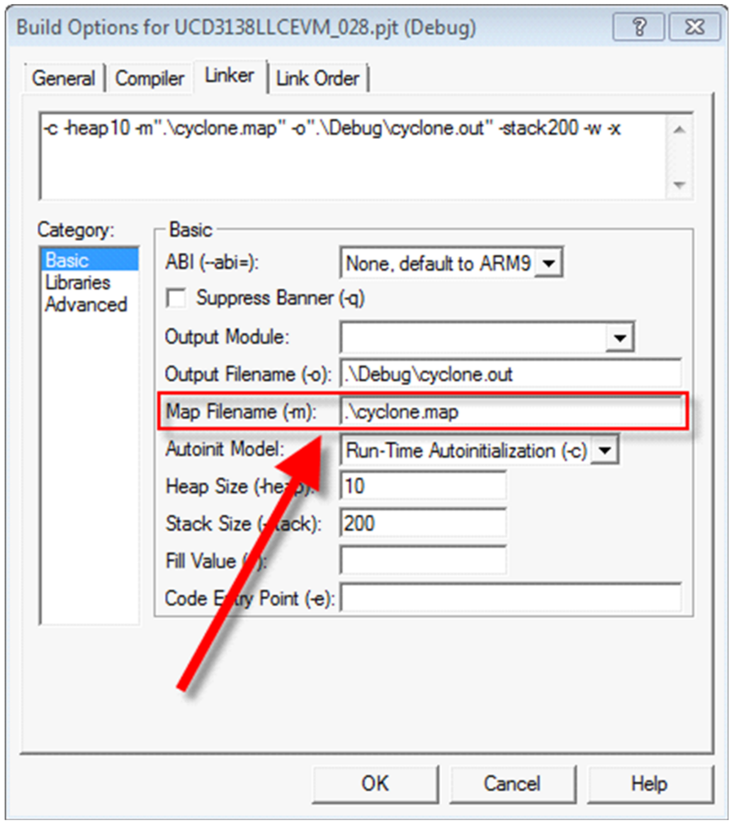


Figure 1 - Map Filename

Figure 88. Map Filename

After selecting the location of the ".map" and ".pp" files, the debugger extracts the information it needs to allow read/write access to all global firmware variables. Depending on the speed of the system, this can take a few moments. The GUI creates a local cache of the data it extracts. So as long as the files do not change subsequent launches of the debugger is much faster.

You now can interact with RAM, DFLASH, or PFLASH variables in the same way described above for device registers. Figure 89 shows an example where variables from RAM and DFLASH have been added to the watch list. "vout_cmd" is the mantissa of a linear16 variable and "supply_state" is a variable indicating the state of the IRQ state machine. Notice that the debugger picks up comments as well as the details of enumerated data types. These variables can be read or written to just like any other variable in the system.

Copyright © 2013–2019, Texas Instruments Incorporated

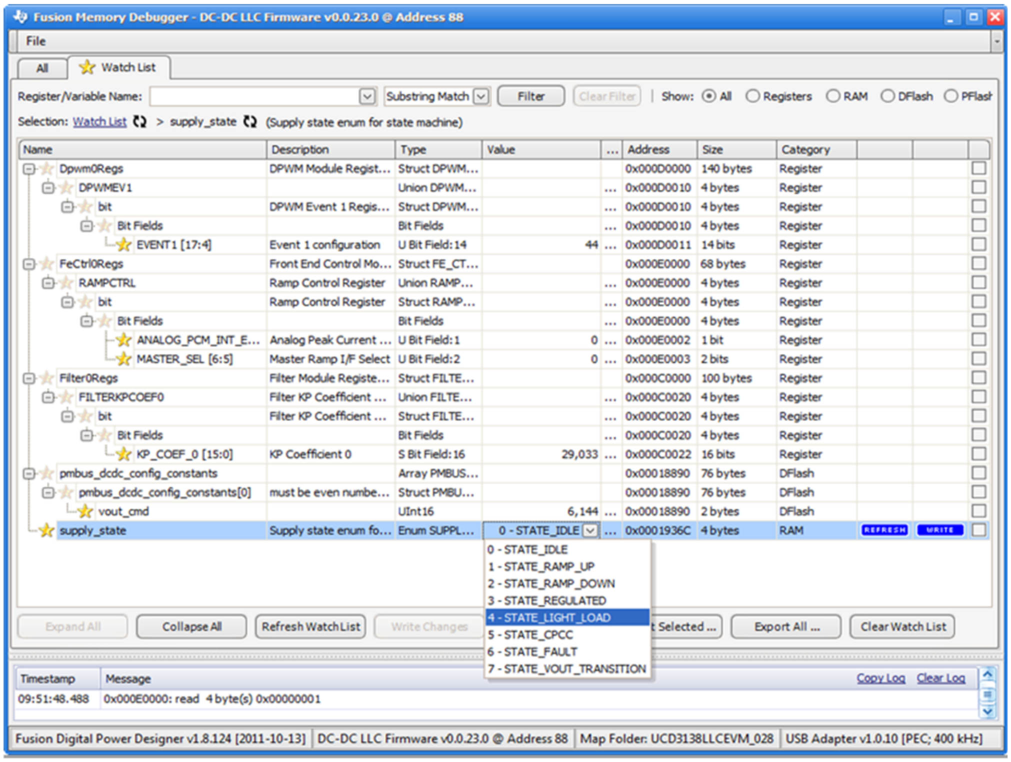


Figure 89. Watch List with Firmware Variables

For the editable values there are up and down arrows.

The increment is normally one. However, the firmware developer has the ability to specify how large the increments are and what the max and min of the variable is. They do this by specifying it in the comments. See highlights in comments below.

```
    extern Uint16 my_uint16; // test root node [min=5, max=200, step=5]

typedef struct
{
    Uint8  a;       // [step=10]
    Uint8 b;       // [min=0, max=100, res=5]
    Uint8 c;       // [min=100, res=5]
    float   d;       // [min=-1e-3, max=1e3] step/res do not make sense with floats
    Int8    e;       // [min=-100, max=100]
} struct1;
```

The order within the brackets does not matter. White space also does not matter.

Note there are two different ways to change how the up/down arrows work in the decimal editor:

- Step: simple increment/decrement. If the current value is 2 and the step is 5, clicking up, changes the value to 7.
- Res: modulo oriented resolution. If the current value is 2 and the res is 5, clicking up, changes the value to 5.

Copyright © 2013–2019, Texas Instruments Incorporated

5.8 SMBus Debug

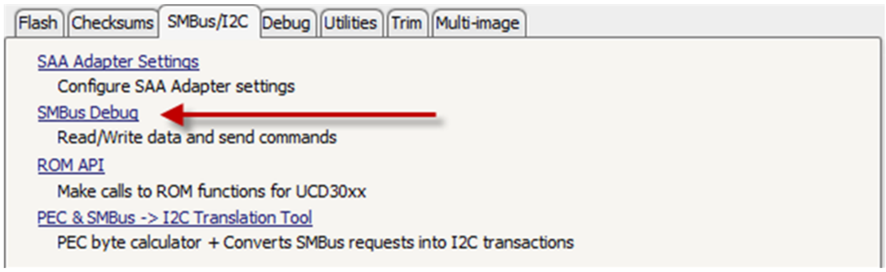


Figure 90. SMBus Debug Link

The tool looks like Figure 91 when owned.

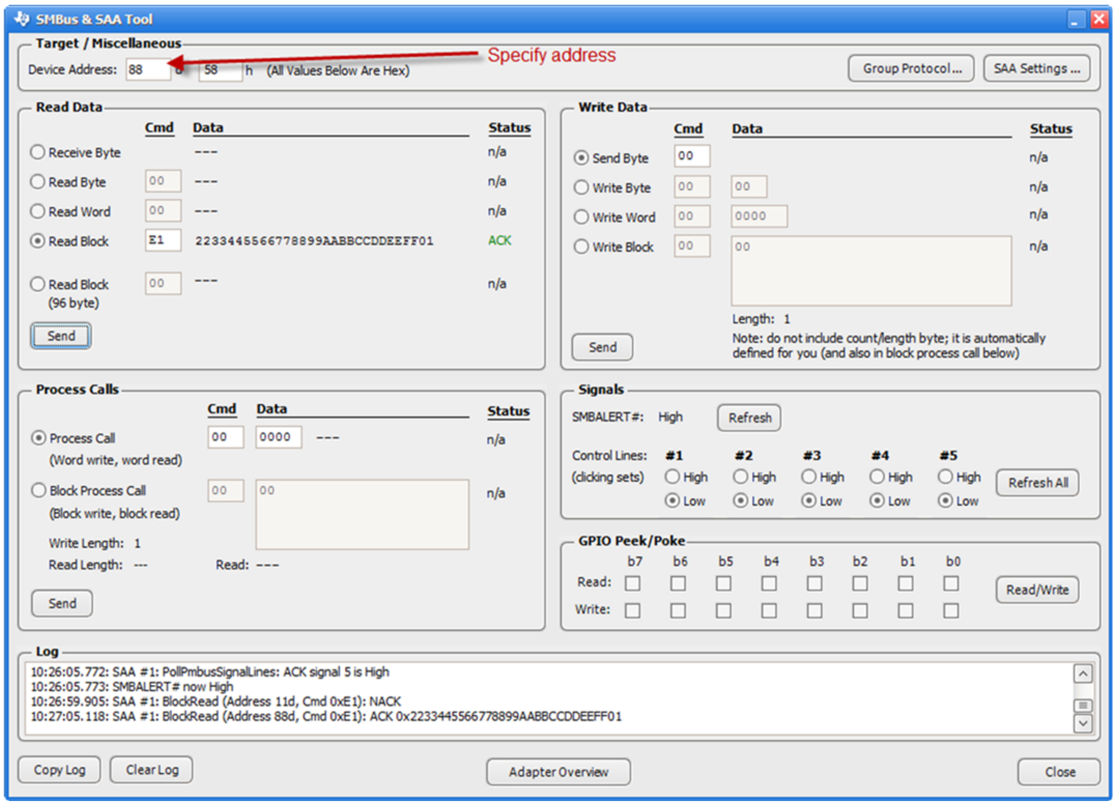


Figure 91. SMBus Debug

In order to use this tool, you need to specify the device address. This tool can be used to interact with PMBus commands. It can be used to Read commands by specifying the hex command and it can be used to write to commands specifying the command and the data.

5.9 CCS conversion

This tool converts UCD31XX device projects from CCS 3.3 to CCS 5.5. NOTE: Although the project should compile after conversion, in rear cases, some manual steps may be required. All files in the original folder ends up in the new folder. Only relevant files are updated or used. All files used, updated, or simply copied are reported in the log.

Copyright © 2013–2019, Texas Instruments Incorporated

5.9.1 How to Access

To access from the “Start” menu, click Texas Instruments Fusion Digital Power Studio->Tools->Isolated CCS Conversion Tool.

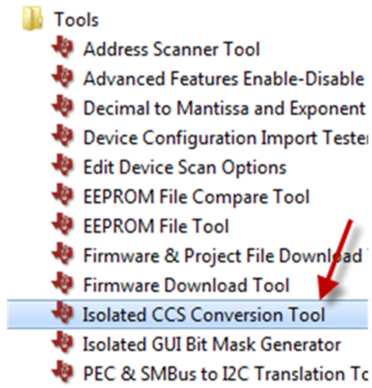


Figure 92. CCS Conversion Tool Access from Start Menu

From the “UCD3XXX/UCD9XXX Device GUI”, it can be found in the “Utilities” tab.

Figure 93. Conversion Tool Access from Device GUI

5.9.2 Usage

1. Browse to the location of the CCS 3.3 UCD31XX project file (*.pjt).
2. Browse to the location of where the new CCS 5.5 project is stored. By clicking the browse button, it suggests the new “Project name” based on the project name from CCS 3.3. The newly created project creates a folder with the “Project name” and a timestamp appended (for example “UCD3138LLCEVM_028_25-12-2014-12-05-22”).
3. Click “Convert.”
4. After the project has completed (usually after a couple seconds), the log is updated with the results. Sometimes it indicates warnings in yellow (for example, “Stale file zoiw.asm” may appear if the file was in the old folder but was not referenced in the project file, that is the original project was not even using this file). All updates made by the tool is displayed in the log. Code changes are displayed in a light green. The old and new versions of the code are both shown. Below are some snapshots of the log. A copy of the log is automatically stored in the converted project folder with a timestamp (for example, “Conversion-Log-2014-04-17-16-13-54.html”).
   a. After the conversion is completed, code changes made including filename, line number, and old and new code are displayed.
   b. The log can be copied or opened in a web browser. The log opened from the button is the same one stored in the converted project folder.
5. To quickly access the newly converted project, click “Open Folder” in the “New CCS 5 Project Location” area.

The project can now be imported to CCS 6.

Copyright © 2013–2019, Texas Instruments Incorporated

5.10 Function Command Summary

The following table lists the ROM/Program commands called for some of the common functions used in the Device GUI.

Table 1. ROM/Program Commands

<table>
<tr><th rowspan="2">Device GUI Function</th><th rowspan="2">Mode: ROM/Program</th><th colspan="4">Commands</th><th rowspan="2" colspan="2">Description</th></tr>
<tr><th>Code (hex)</th><th>Command</th><th>Trans. Type</th><th>Data Format</th></tr>
<tr><td rowspan="3">Scan Device in ROM Mode</td><td rowspan="3">ROM</td><td>0xEC</td><td>Read Version</td><td rowspan="3">Read Block (up to 32 bytes)</td><td rowspan="3"></td><td rowspan="3" colspan="2">Scans for the device in ROM mode and reads PKGID at address 0xFFF7F010 using 0xFD and 0xFA.</td></tr>
<tr><td>0xFD</td><td>Configure Read Address</td></tr>
<tr><td>0xFA</td><td>Read 4 Bytes</td></tr>
<tr><td rowspan="5">Scan for Device in Program Mode: DEVICE ID</td><td>Program</td><td>0xFD</td><td>DEVICE_ID</td><td>Read Block (up to 32 bytes)</td><td>String</td><td colspan="2">MFR command supported by UCD devices. Ex. “UCD310128V1|0.1.0.0010|131009”</td></tr>
<tr><td>Program</td><td>0xE4</td><td>CMDS_DCDC_PAGED</td><td>Read only</td><td>Bitmask</td><td colspan="2">Contains bitmask of paged DCDC supported commands in the firmware</td></tr>
<tr><td>Program</td><td>0xE5</td><td>CMDS_DCDC_NONPAGED</td><td>Read only</td><td>Bitmask</td><td colspan="2">Contains bitmask of non paged DCDC supported commands in the firmware</td></tr>
<tr><td>Program</td><td>0xE6</td><td>CMDS_PFC</td><td>Read only</td><td>Bitmask</td><td colspan="2">Contains bitmask of PFC supported commands in the firmware.</td></tr>
<tr><td>Program</td><td>0xE7</td><td>SETUP_ID</td><td>Read only</td><td>String</td><td colspan="2">Special value that maps to a topology and how it is compensated within the Fusion Studio GUI. Ex. “VERSION1|LLC001”</td></tr>
<tr><td>Scan for Device in Program Mode: DEVICE CODE</td><td>Program</td><td>0xFC</td><td>-</td><td>-</td><td>-</td><td colspan="2">Not applicable for UCD devices</td></tr>
<tr><td>Scan for Device in Program Mode: IC DEVICE ID</td><td>Program</td><td>0xAD</td><td>-</td><td>-</td><td>-</td><td colspan="2">Not applicable for UCD devices</td></tr>
<tr><td>Scan for Device in Program Mode: PMBUS REVISION</td><td>Program</td><td>0x98</td><td>PMBUS_REVISION</td><td>Read Byte</td><td>Byte</td><td colspan="2">Defined by PMBus spec. All PMBus devices support it.</td></tr>
<tr><td rowspan="6">[Check] When a device is found, dump additional PMBus commands</td><td rowspan="6">Program</td><td>0x9A</td><td>MFR_MODEL</td><td rowspan="6">R/W Block</td><td rowspan="6">String</td><td>Ex. “UCD3138LLC EVM-028”</td><td rowspan="6">If this box is checked then after any of the scans above are completed, these commands are read.</td></tr>
<tr><td>0X9B</td><td>MFR_REVISION</td><td>Ex. “E3”</td></tr>
<tr><td>0X9E</td><td>MFR_SERIAL</td><td>Ex. “SV001” – unique identifier</td></tr>
<tr><td>0X99</td><td>MFR_ID</td><td>Ex. “TI”</td></tr>
<tr><td>0X9D</td><td>MFR_DATE</td><td>Ex. “14033” YYMMDD</td></tr>
<tr><td>0X9C</td><td>MFR_LOCATION</td><td>Ex. “Dallas, TX”</td></tr>
<tr><td>Command ROM to execute its program (SendByte 0xF0 to Address 11) First Block</td><td>ROM</td><td>0xF0</td><td></td><td>Send Byte</td><td></td><td colspan="2">Executes the First Block of program flash</td></tr>
<tr><td>Command ROM to execute its program (SendByte 0xF7 to Address 11) Second executable block</td><td>ROM</td><td>0xF7</td><td></td><td>Send Byte</td><td></td><td colspan="2">For devices that support multiple program flashes, this command executes the flash from the second block (or the third block for UCD3138 128).</td></tr>
</table>

Copyright © 2013–2019, Texas Instruments Incorporated

Table 1. ROM/Program Commands (continued)

Device GUI Function	Mode: ROM/Program	Commands				Description
		Code (hex)	Command	Trans. Type	Data Format	
Command Program to jump to ROM (SendByte 0xD9)	Program	0xD9	ENABLE_ROM	-	-	ENABLE_ROM
Command Program to jump to ROM (SendByte 0xD9) with 0xF9 implemented	Program	0xF9	ENABLE_ROM2	W	String	When pressing the command to jump to ROM if 0xF9 is implemented, you must enter a password and it is sent with 0xF9
Memory Debugger/Peek Poke	Program	0xE2	PARM_INFO	W	Block	PARM_INFO and PARM_VALUE are both used. The first sets the address to be read and the second returns the value at that location.
	Program	0xE3	PARM_VALUE	R	Block	

5.11 Override Commands

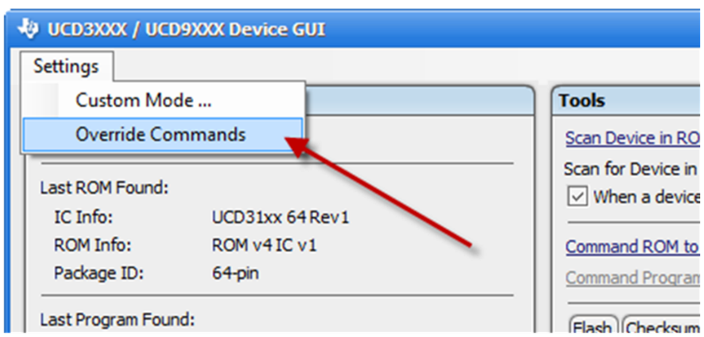


Figure 94. Access to Override Commands

In Section 5.10, a description was provided for the various commands used by the Device GUI. By default, the Device GUI assumes that certain MFR commands use a default hex code and are implemented a certain way. Sometimes this assumption is not valid and you need to override which command codes are used due to a conflict with another command having the same hex code. Assuming the implementation of the command is the same, the Device GUI provides a way to override or change the command code that the Device GUI assumes so that you can still benefit from the Device GUI. Figure 95 displays the available MFR commands that the you can override, assuming implementation has remained the same.

Copyright © 2013–2019, Texas Instruments Incorporated

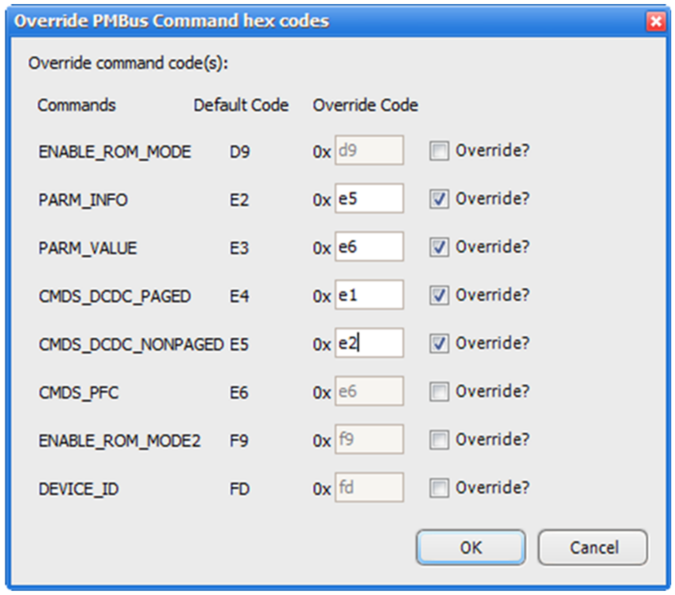


Figure 95. Four Commands Have Been Overridden

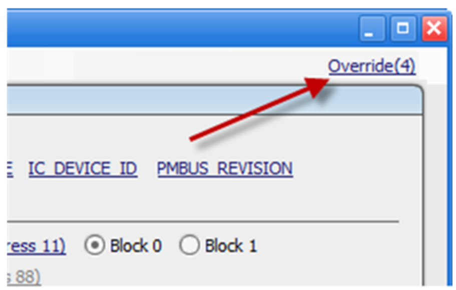
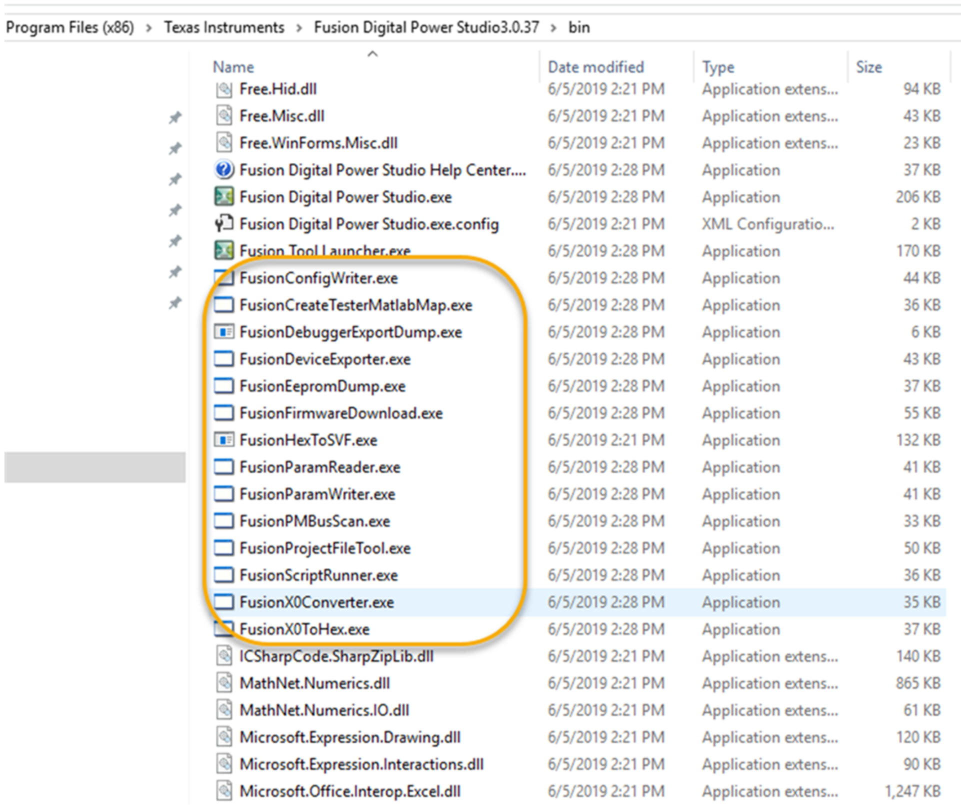


Figure 96. Override in Top Right Shown

6 Command Line Tools

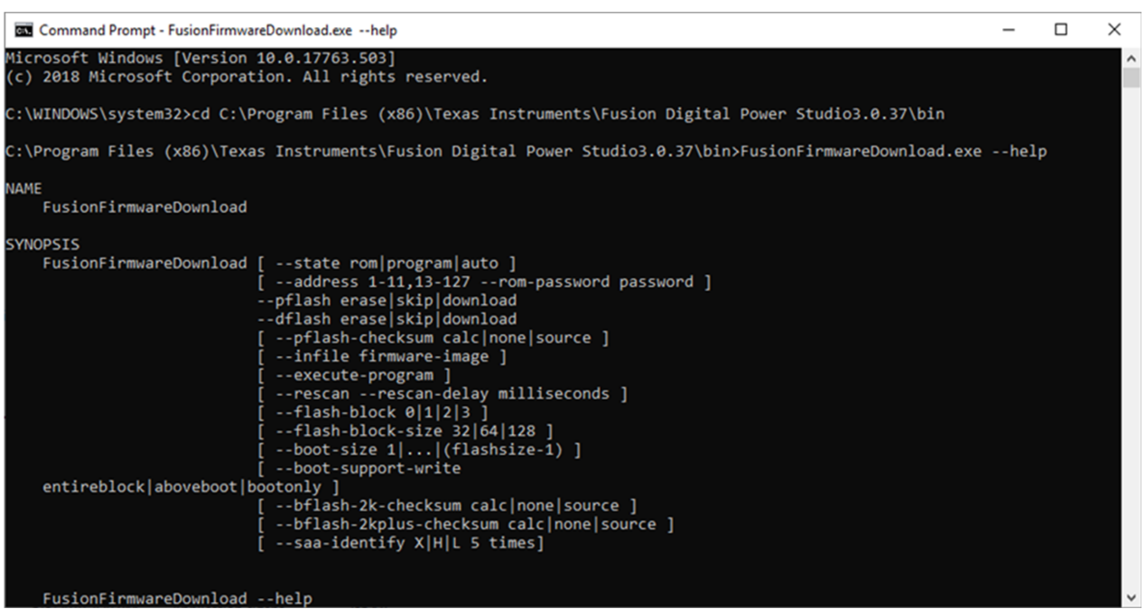
In the install directory, there are a number of command line tools that are included. These command line tools replicate the functionality of a number of tools described in the Device GUI. To get more information on each of the command line tools, simply go to the install directory in a command prompt session and append –help to the tool.

Copyright © 2013–2019, Texas Instruments Incorporated



6.1 FusionFirmwareDownload.exe

This command line allows you to download firmware to the device with the same configuration options as described in Section 5.3. Below is an image of what appears when looking up the help for this command line tool.



Copyright © 2013–2019, Texas Instruments Incorporated

7 API – Application Programming Interface

www.ti.com/tool/fusion_digital_power_api

There is a reusable API behind most of the functionality covered. It can be used via .NET: VB or C#. This can be used to automate tests or even create new custom GUIs. TI provides binary libraries, source code for examples, and documentation.

8 Production Tool

http://www.ti.com/tool/fusion-production-gui

When it is time for production, there is another tool that has been used to speed up the process of configuring devices. It is called the Manufacturing GUI. This graphical tool can be used to run scripts on the devices and provide a pass/fail result. All functions done through the device GUI can be automated through the MFR GUI. Some of the functions included are downloading or updating firmware, importing a project file on to a device, writing serial numbers and MFR date, calibrating devices using instrumentation (GPIB, SCPI, USB) or manual measurements, testing the output of the device, and various other functions. You can also develop their own functions to include in the manufacturing scripts.

9 Documentation and References

9.1 References

- PMBus specification

Copyright © 2013–2019, Texas Instruments Incorporated

10 Revision History

Version	Date	Comment
SLUA676	November 26, 2013	Initial document
SLUA676A	July 2, 2014	• Updated Device GUI screenshots • Updated Firmware download section • Added CCS conversion section • Added Override command section • Added Function command summary • Added device GUI checksum section • Added link for Fusion-API
SLUA676A	January 2015	• Include devices UCD3138128 and UCD3138A • Added section on device respins and program mode detection • Added section on partial dataflash download
SLUA676B	August 2017	• Rename Designer to Studio • Update pictures • Rename Manufacturing Tool to Production Tool
SLUA676C	June 2018	• Update images for memory dump and memory peek poke • Update image for data flash mode • Update versioning • Update introduction to include A version • Update PC requirments to Windows 7+/.net version 4.5 • Update download & installation • update startup images that mention designer with studio • Added equations for each sample topologies • Update picture for existing project • Update picture for sample project • Added section for Command Line Tools • Edited user's guide for clarity

Copyright © 2013–2019, Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated