Fusion Digital Power Studio GUI for Isolated Power Applications

User's Guide



Literature Number: SLUA676C March 2013–Revised June 2019



Contents

1	Tradem	narks		7
2	About 7	's Guide	7	
	2.1	Introduct	ion	7
		The Fusi	on Digital Power Studio (Studio GUI)	7
		The Devi	ce GUI	8
	2.4	Conventi	ons	8
		rface Terminology and Tips		
	2.6 Terminology			
	2.7		al Technical Support	
3				
-	3.1	•	irements	
	3.2	-	pter	
	3.3		d and Installation	
	3.4		g the GUI	
	3.4 3.5		nstallations of the GUI	
4		-		
4		-	ower Studio (Studio GUI)	
	4.1	•	he GUI	
	4.2		ng to a Remote USB Adapter	
		4.2.1	Client Configuration	
		4.2.2	Remote Server Configuration	
		4.2.3	Client and Server Running	
		4.2.4 4.2.5	Connection Troubleshooting Tips	
		4.2.5 4.2.6	SETUP_ID in Firmware is Not Recognized by the GUI Enable GUI Protected Features	
		4.2.7	Monitor	
		4.2.8	Configure	
		4.2.9	-	
	4.3	-		
	4.4	Capturing	g the State of the Device - Saving a Project File	31
	4.5		neous Tools	
		4.5.1	Multi-image	
		4.5.2	Isolated Bitmask Tool	
	4.6	Offline M	lode	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	Launchin	g Device GUI	37
	5.2	Moving E	Between ROM and Program Mode	39
		5.2.1	ROM Mode to Program Mode for Multiple Flashes	40
	5.3	Firmware	e Download Tool	41
		5.3.1	Boot Support	43
		5.3.2	Data Flash Download	46



www.	ti.com			
	5.4	Check	sum Functions	47
	5.5	Multi-image Functions		
		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	Isolate	d Bitmask Tool	58
	5.7 Firmware Memory Debugger			
	5.8			69
	5.9	•		69
		5.9.1	How to Access	70
		5.9.2	Usage	70
	5.10	Functio	on Command Summary	71
	5.11	Overrio	de Commands	72
6	Comma	and Line	e Tools	73
	6.1		FirmwareDownload.exe	
7	API – A	pplicat	ion Programming Interface	75
8	Production Tool			
9		Documentation and References		
-	9.1		nces	
10	Revisio	on Histo	ry	76



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



www.ti.com		
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	
67	Image Download	50
68	Image to Switch to	
69	Image Peek/Poke	
70	Image Dump	
71	Erase Image	
72	Calculate Image Checkum	
73	Dump Image Checksum	
74	Recreate Image Checksum	
75	Validating Image Checksum	
76	Clearing Image Checksum	
77	Click Iso Bitmask Tool	
78	Bitmask Tool	
79	Memory Debugger	
80		
81	Fusion Studio GUI Debugger Tool	
82	GUI UCD3138 Debugger – Defaults	
83		
84 85	Watch List Selection Star Map File Selection	
86	Debugger Customization Tool	
80 87	".pp" Generation Parameters	
88	Map Filename	
89	Watch List with Firmware Variables	
90	SMBus Debug Link.	
91	SMBus Debug	
92	CCS Conversion Tool Access from Start Menu	
93	Conversion Tool Access from Device GUI	
94	Access to Override Commands	
95	Four Commands Have Been Overridden	
96	Override in Top Right Shown	



List of	Tables
---------	--------

1	ROM/Program Commands	71
---	----------------------	----



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	

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

FUNG	IMAGE	
Checkbox	You can select any number of boxes.	🕑 Vin 📄 Vout 🗹 Iout 🔽 Temp
Radio button	You can only select one of the circles at a time. For example, clicking "High" deselects "None."	● None ◯ Low ◯ High
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).	0.880 🗸
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.	○ on ⊙ Off

2.6 Terminology

8

'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

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:

Texas Instruments				
Fusion Digital Power Studio Version 3.0.37 [2019-06-05]				
No USB Adapter Found! A Texas Instruments USB serial bus adapter does not appear to be connected to your PC. Please check your connection. You should see a green light on the adapter when it is attached to the PC.				
Retry Adapter Mode Offline Mode Exit Program				

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:

TEXAS INSTRUMENTS
Fusion Digital Power Studio Version 3.0.37 [2019-06-05]
Scanning for devices on the PMBus

Figure 3. Device Scanning

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



TEXAS INSTRUMENTS					
Fusion Digital Power Studio Version 3.0.37 [2019-06-05]					
No Devices Found! No compatible PMBus devices were found. Please check that the serial cable end of your USB adapter is attached to your device and power is supplied to your device. Scanning Mode: DEVICE_ID scan (UCD3xxx)					
USB Adapter	Firmware Version: 1.0.11				
Bus Speed:	Packet Error Checking:		ALERT Pullup:	Open Drain 🖂	
🔾 100 kHz	 Enabled 	 Serial 	CLOCK Pullup:	668 Ω 🗸	
• 400 kHz	O Disabled		DATA Pullup:	668 Ω	
Change Device Scanning Options Retry Offline Mode Exit Program					

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.



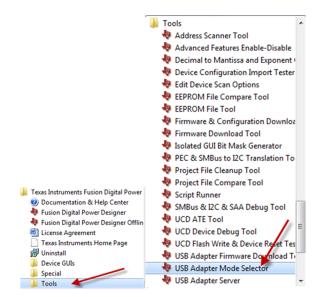


Figure 5. USB Adapter Mode Selector from Start Menu

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.

🜵 USB I2C Adapter Local/Remote Mode Selection	
Overview All tools bundled with the Fusion Digital Power Designer can be configured to use a USB I2C adapter on a remote PC. This is used by the Fusion development team to debug the GUI while it runs on a remote board, such as in one of our labs at Texas Instruments.	
Customers and Tiers can also use this special mode to connect to a remote adapter on your company LAN/WAN. In order to connect to the remote adapter, you must have Fusion Digital Power Designer installed on a PC attached to the adapter and run a special adapter "server" tool. This tool can be found in Start Menu->Fusion Digital Power Designer->Tools->USB Adapter Server.	
The GUI can not connect to an adapter that is beyond your firewall and vice versa Running the GUI in this way may also be slower than using remote desktop or othe remote PC viewing applications, as each I2C request must travel to the remote PC and back before the GUI can move onto the next I2C request. When run on a LAN it is fairly speedy. When run over a slow WAN link or VPN, speed will be poor.	er
— Mode Selection & Settings	
O Local - use USB adapter(s) connected to this PC	
Remote - use USB adapter(s) connected to another PC	
Host: 172.24.38.145	
IP address or PC name	
Port: 8090 🗸	
Password: Show	
This is not a password to a user account on the PC; it is an optional password defined at the time the server is started	
Test Connection	
Close	.1

Figure 6. Adapter Mode Selection



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.

Connection Test Success Success. I was able to connect to the server.
OK Copy Message to Clipboard

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.

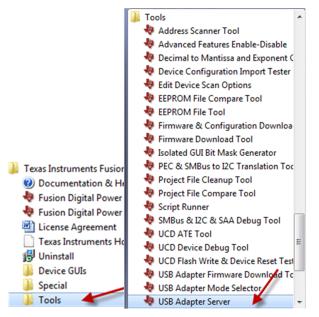


Figure 8. USB Adapter Server from Start Menu

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.



🧄 TI USB 12C 🗚	Adapter Server			_ 🗆 🔀
	emote GUI connect to and use you ng. To run a remote GUI against th			her Fusion tools on this local PC while ter Preferences tool.
TCP Port:	8090 🜩	Your IP Address (Server):	172.24.38.145	Send Invite via E-Mail
Password:	Show	Last Client IP:	Not Connected	Copy Invite to Clipboard
(n	ot required; can be empty)	Total # SAA Requests:	0	
Log:				Detailed Logging
Timestamp	Message			
16:28:49.760	Server started, listening for conn	ections on port 8090		
16:28:53.901	Server stopped			
16:29:29.274	Server started, listening for conn	ections on port 8090		
16:29:55.811	Server stopped			
PauseLog	Copy Log Clear Log	Start Serv	rer	

Figure 9. Remote Adapter Server

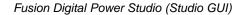
If you click "Send Invite via E-Mail", a pre-filled email appears populated with the relevant information for the client to connect to the server. It includes the IP address, Port, and Password, if any. It also includes instructions on how to configure the client. Figure 10 shows the content of the pre-filled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

	То	my-clent-friend@friend.com	
Send	Сс		
	Bcc		
	Subject:	Fusion GUI Remote Adapter Invite	_
Pleas	e connect to	my TI USB I2C adapter:	1 M 10
- Po	address: 172 rt: 8090 ssword: none	2.24.38.145 e, leave empty	
You	an configure	Fusion Digital Power Designer to use a remote adapter server by:	
		t Menu, selecting "Texas Instruments Fusion Digital Power Designer", selecting "Tools", and then selecting "USB Adapter Mode Selector". This can be used to cal adapter mode and remote mode.	
or:			
		our local adapter and run whatever tool you want (the full GUI or a tool). There is an "Adapter Mode" button on the "No USB Adapter Found" form that will ck this button to connect to a remote adapter server.	
Your	adapter mod	e selection is saved between restarts of GUI tools.	

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.



	emote GUI connect to and use your running. To run a remote GUI again			h other Fusion tools on this local PC the Adapter Preferences tool.
TCP Port:	8090 💭	Your IP Address (Server):	172.24.38.145	Send Invite via E-Mail
Password:	Show	Last Client IP:	158.218.99.224	Copy Invite to Clipboard
(r	not required; can be empty)	Total # SAA Requests:	391	
Log:				🖂 Detailed Loggir
Timestamp	Message			
16:44:18.082	158.218.99.224: ReadWord (Add	ress 89d Cmd 0x8B): ACK 0x1	11CF	
16:44:18.158	158.218.99.224: BlockRead (Addr	ess 89d, Cmd 0xF0); ACK 0x0	102030405060708	
16:44:18.219	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x95): ACK 0x	0000	
16:44:18.285	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x8C): ACK 0x0	D2E9	
16:44:18.356	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x8E): ACK 0xF	=21D	
16:44:18.420	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x8D): ACK 0x0	DA09	
16:44:18.489	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x79): ACK 0x0	0841	
16:44:18.556	158.218.99.224: PollPmbusSignal	ines: ACK SMBALERT# is High		
16:44:18.660	158.218.99.224: PollPmbusSignall	ines: ACK SMBALERT# is High		
16:44:18.725	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x88): ACK 0xE	531F	
16:44:18.796	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x8B): ACK 0x1	11E8	
16:44:18.860	158.218.99.224: BlockRead (Addr	ess 89d, Cmd 0xF0): ACK 0x0	102030405060708	
16:44:18.924	158.218.99.224: ReadWord (Add	ress 89d, Cmd 0x95): ACK 0x0	0000	
16:44:18.989	158.218.99.224: PollPmbusSignal	ines: ACK SMBALERT# is High		

Figure 11. Running a Remote Server Connected to a Client



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

Texas

www.ti.com

RUMENTS



File Device Tools	Debug Help	DC-DC LLC @ 89d - Page 0x0	•
Monitor	Readings - Page 0x0	Vin - Input Voltage	×
how/Hide Plots: Vin Vout Jout Pout(calc) Temp Int Temp Ext All Temp Freq Fit All Plots on Screen	Vin: 49.938 V Vout: 8.918 V Jout: 11.64 A Temp Int: 16 °C Temp Ext: 135 °C	OVF: 430.000 ⊕ V OVW: 400.000 ⊕ V UVW: 350.000 ⊕ V UVF: 320.000 ⊕ V VinON: 320.000 ⊕ V VinOFF: 310.000 ⊕ V 400.00	Write
Scale Plots to Screen Width Height: 200 💬	Status Registers/Lines Vout: OK Iout: OK Temp: OK	200.00 100.00 0.00 49.563 ¥ 46:20 46:40 47:00	47:20
Show Wall Craduc Limit Editors Show Value Labels on Plots Show Value Labels on Plots Stop Polling Stop Polling Launch Dashboard	Input: OK CML: OK Misc: Output Off, POWER_GOOD# Debug Buffer: 0x0102030405060708 Mfr: OK SMBALERT# Not Asserted Clear Faults	Vout - Output Voltage OVF: 14.000 ÷ V OVW: 13.000 ÷ V Vout: 12.000 ÷ V UVW: 11.000 ÷ V UVF: 16.00	
	Control Line (USB)	6.00 6.304 V 4.00 2.00 0.00 46:00 46:00 46:20 46:40 47:00	47:20
🖑 Configure	Tips & Hints VOUT_OY_FAULT_LIMIT [0x40]	PMBus Log	Ţ.
🖗 Design	Sets the value of the output voltage meas the sense or output pins that causes an or overvoltage fault.		
Monitor		V	~
③ Status		PMBus Log	Fa 6

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.



4.2.5.1 Change the Device Scanning Options

No Devices I	Found!		
	PMBus devices were found our device and power is sup		that the serial cable end of your USB adapter is
	de: DEVICE ID and DEVI		
-	_	_	"
USB Adapter	Firmware Version: 1.0.10		
Bus Speed:	Packet Error Checking:		ALERT Pullup: 2.2 kΩ 🗸
🔾 100 kHz	 Enabled 	 Serial 	CLOCK Pullup: 2.2 kΩ
• 400 kHz	Olisabled		DATA Pullup: 2.2 kΩ 🗸

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.

49 De	vice Scan Editor														• 🗙
Set A	Addresses To:	Skip	DEVICE		CODE	DEVICE	ID & DEVICE_0	COD	CD3X00X Iso	lated	>		_		
1d	0x01 UCD3000	\geq	19d 0x13	UCD 300X	\sim	36d 0x2	UCD3XXX	\checkmark	53d 0x35	UCD300X	\sim	70d 0x46	UCD3XXX	\sim	
2d	0x02 UCD3000	\sim	20d 0x14	UCD300X	\sim	37d 0x2	UCD300X	\sim	54d 0x36	UCD3XXX	\sim	71d 0x47	UCD3000	\sim	
3d	0x03 UCD3XXX	\sim	21d 0x15	UCD300X	\sim	38d 0x2	UCD300X	\sim	55d 0x37	UCD3XXX	\sim	72d 0x48	UCD300X	\sim	
4d	0x04 UCD3000	\sim	22d 0x16	UCD300X	\sim	39d 0x2	7 UCD300X	\sim	56d 0x38	UCD3XXX	\sim	73d 0x46	UCD300X	\sim	
Sd	0x05 UCD3000	\sim	23d 0x17	UCD 300X	\sim	40d 0x2	UCD300X	\sim	57d 0x39	UCD 3000	\sim	74d 0x44	UCD3XXX	\sim	
6d	0x06 UCD3000	\sim	24d 0x18	UCD300X	\sim	41d 0x2	UCD300X	\sim	58d 0x3A	UCD3000	\sim	75d 0x48	UCD3000	\sim	
7d	0x07 UCD3000	\sim	25d 0x19	UCD300X	\sim	42d 0x2	UCD300X	\sim	59d 0x38	UCD3XXX	\sim	76d 0x40	UCD3XXX	\sim	
8d	0x08 UCD3XXX	\sim	26d 0x1A	UCD 300X	\sim	43d 0x2	UCD300X	\sim	60d 0x30	UCD3XXX	\sim	77d 0x40	UCD300X	\times	
9d	0x09 UCD3000	\sim	27d 0x1B	UCD3000	\sim	44d 0x2	UCD300X	\sim	61d 0x30	UCD3000	\sim	78d 0x46	UCD 300X	\sim	
10d	0x0A UCD3XXX	\sim	28d 0x1C	UCD300X	\sim	45d 0x2	UCD300X	\sim	62d 0x3E	UCD3000	\sim	79d 0x4F	UCD300X	\sim	
11d	0x08 UCD3XXX	\sim	29d 0x1D	UCD300X	\sim	46d 0x2	UCD300X	\sim	63d 0x3F	UCD3000	\sim	80d 0x50	UCD3XXX	\sim	
	0x0D UCD3XXX	\sim	30d 0x1E	UCD300X	\sim	47d 0x2	UCD300X	\sim	64d 0x40	UCD3000	\times		UCD 300X	\times	
	0x0E UCD3XXX	\sim		UCD3XXX	\sim		UCD300X	~	65d 0x41		~		UCD3XXX	~	
	0x0F UCD3XXX	\sim	32d 0x20		\sim		UCD300X	~		UCD3XXX	\sim		UCD3XXX	~	
	0x10 UCD3000	~	33d 0x21		\sim		2 UCD300X	~	67d 0x43		~		UCD300X	~	
	0x11 UCD3000	~	34d 0x22		~		3 UCD300X	~	68d 0x44	UCD3XXX	~	85d 0x55	UCD300X	~	
	LICE MARKEN			1			110000000]		_			>	Γ
							ОК)						🕐 He	Þ

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.

💼 Texas Instruments Fusion Digital Power Designer	•	🛅 Device GUIs	•	
💼 Texas Instruments Fusion Digital Power Manufacturing Tool	•	💼 Special	•	🏘 Masquerade as TP540400
TI-COMM for Windows	•	🛅 Tools	•	🏘 Masquerade as UCD3000ISO2
m TN3270 Web	•	Ocumentation & Help Center		🍓 UCD3XXX Isolated Fallback Device Scan Mode
m WebEx	•	🌵 Fusion Digital Power Designer		▲
🖮 WinZip	•	🌵 Fusion Digital Power Designer Offline Mode		
Adobe Reader 9		License Agreement		
🔗 Internet Explorer		Texas Instruments Home Page		-
🗯 Outlook Express		🔂 Uninstall		

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



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.

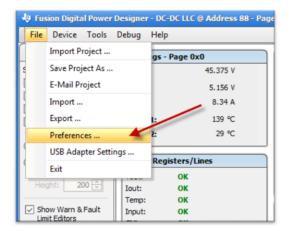


Figure 17. GUI Preferences

9 Fusion Digital Power Designer Preferences	2
- Preferences	-
Move device dashboard window when main window is moved or resized	
Use PAGE_PLUS_READ and PAGE_PLUS_WRITE to read/write PAGEd commands on PMBus 1.2 capable devices	
Show advanced editors and features that are normally hidden (e.g. "Advanced Config" on UCD92xx and "All Config" on UCD90xxx)	
Enable GUI protected features (e.g. pflash export):	
Password: ••••••• Valid	
Enable GUI customer-specific features	
Configure Device Scan Mode and Addresses	
Enable all Standard Warnings/Confirmations	
Delete All Application Preferences	
Cancel OK	

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.

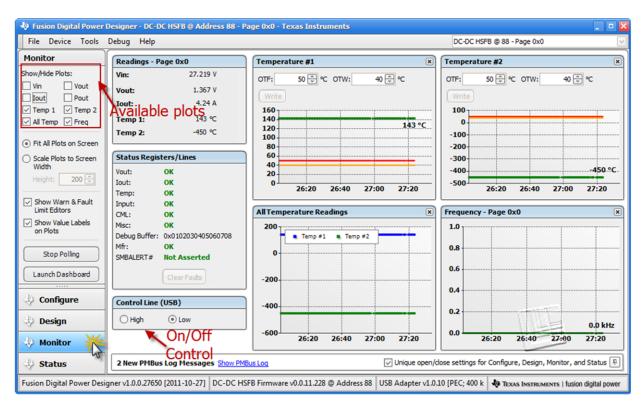


Figure 19. Monitor Mode Displays Some of the Live Parameters Being Read From the Device

4.2.8 Configure

TEXAS

www.ti.com

ISTRUMENTS

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.



File Device Tools D	lebug Help				DC-DC HSFB @ 88 - Page 0x0
Configure	Configuration Click here to w	rite e	dits to RAM o	of device	
Write to Hardware	Command	Code	Value/Edit	Hex/Edit	<u> </u>
Auto write on rail or device change	CMD5_DCDC_NONPAGED [MFR 21]	0×E5	0x10006 🗸	0x10 ∨	
Discard Changes	CMD5_DCDC_PAGED [MFR 20]	0xE4	0x00000 ✓	0x00 🗸	
	CPCC [MFR 36]	0xF4	0x01680 🗹 🗲	0x01 🗸	Clicking the combo box
Store RAM To Flash	DEADBAND_CONFIG [MFR 26]	0xEA	0x01900 🗸		causes this dialog to
Restore Flash to RAM	DEVICE_ID [MFR 45]	0xFD	UCD3100ISO1		appear. Making the
Clear Restore Notices	IC_DEVICE_ID	0xAD	UCD3138RGC	IMAX:	40.0 A command editable.
	IC_DEVICE_REV	0×AE	0	TON:	00.0 🕀 msec
Show: Global Device	IDEAL DIODE EMUL CONFIG	0xFE	Enabled V E	nable:	
Parameters	IIN_OC_FAULT_LIMIT	0×58	39.00 💭	Timer:	OFF V
 Parameters for this Rail 	IIN_OC_WARN_LIMIT	0x5D	35.00 🗘		
All Parameters	IOUT_OC_FAULT_LIMIT	0x46	40.00 🗘 🙀	0X0020	
Sort Parameters By:	IOUT_OC_WARN_LIMIT	0×4A	30.00 🗘 A	0x001E	
Command Name	LIGHT_LOAD_CONFIG [MFR 02]	0xD2	0x0007 🗸	0x00 🗸	Parameter has been
O Command Code	MFR_DATE	0x9D	YYMMDD	0x59 ∨	edited
Group by Category	MFR_ID Click to Undo edit	0×99	П	0x54 ✓	
	MFR_LOCATION	0x9C	Dallas, TX	0x44 ✓	
	MFR_MODEL	0x9A	UCD3138HSFBI	0x55 🗸	
	MFR_REVISION	0x9B	E1	0x45 ∨	
	MFR_SERIAL	0x9E	XXXXXX	0x58 ∨	
			(2)		
10 co co V	Tips & Hints			PMBus Log	[
Configure	CPCC [MFR 36,0xF4] Configure Constant Power Constant Current				38-SAA #1: CONTROL 1 now Low C-DC HSFB @ 88: USER_RAM_00 [MFR 10,0xDA]: wrote 1
Design			<u>^</u>	[0x01] to RAM	
Monitor	Description of command b	being	edited		
③ Status			E.	PMBus Log	ር ርስ ነ

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.

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".

Command	Code	Value/Edit	Hex/Edit	Firmwa
CMD5_DCDC_NONPAGED [MFR 21]	0×E5	0x10006 🖂		bitmas
CMDIS_DICOC_PANGED [PHER 20]	0.61	and the second s	03 CLEAR_FAULTS	indicate
CPCC [PHFR 36]	0.64	and the second s	<pre><11 STORE_DEFAULT_ALL <12 RESTORE_DEFAULT_ALL</pre>	support s
DEADBAND_CONTRE[INFR 26]	0.68	Tarrent Contraction of the	20 VOUT_MODE	comman
DEVICE_ID [PIPR 45]	0.60	UKCD/04000001 (0)	21 VOUT_COMMAND 27 VOUT_TRANSITION_RATE	
EC_DEVECE_RD	Du#D	TARLEY AND ADDRESS OF	35 VIN_ON	E
EC_DEWSCE_REW	DuRE		<pre><36 VIN_OFF <40 VOUT_OV_FAULT_LIMIT</pre>	
IDEAL DODDE EMIL CONFIG	OUFE	COTTON OF THE OWNER OF THE OWNER	(42 VOUT_OV_WARN_LIMIT	
EN.OC.FAIRT_LIPIET	0.58		<pre>(43 VOUT_UV_WARN_LIMIT (44 VOUT_UV_FAULT_LIMIT</pre>	
EPI OC MININ LIMET	0.60	372, (38D 115-	<pre>(46 IOUT_OC_FAULT_LIMIT</pre>	
BORT OK FAULT LIMET	0.46	0)	<pre>(4A IOUT_OC_WARN_LIMIT </pre> (4F OT_FAULT_LIMIT	
EDUT OC WARN LINET	0.46	787, 787, 117	51 OT_WARN_LIMIT	
LICHIT LICHO COMPIG (MER 02)	0.02	0	<pre>x55 VIN_OV_FAULT_LIMIT x57 VIN_OV_WARN_LIMIT</pre>	
NER DATE	0.90	0)	58 VIN_UV_WARN_LIMIT	
NFR_ID	0.99	0)	<pre><59 VIN_UV_FAULT_LIMIT <58 IIN_OC_FAULT_LIMIT</pre>	
NER LOCATION	0.90	0)	5D IIN_OC_WARN_LIMIT	
		0	(SE POWER_GOOD_ON (SF POWER_GOOD_OFF	
MER_MODEL	0,98	0)	(61 TON_RISE	
PHERE BLEWISSION	0/98		78 STATUS_BYTE	
NETR_SERGER	(Dr/RE	10/0/0/0/0	<pre><79 STATUS_WORD <88 READ_VIN</pre>	

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).



🚸 Fusion Digital Power Designer - DC-DC HSFB @ Address 88 - Page 0x0 - Texas Instru File Device Tools Debug Help DC-DC HSFB @ 88 - Page 0x0 Design Voltage Loop (CLA #0) Current Loop (CLA #1) Feed Forward Loop (CLA #2) Plots Power Stage - Rail #1 **Frequency Response** Power Stage Parameters Auto Calculate Metrics: 🔿 Loop 💿 Stage 🗹 Loop: 🛑 Frequency Data Vbus: 48.000 🖓 V Vout: 12.000 👻 V Crossover: 13.40 kHz Stage: View Coeff "C" Code Comp: Fs: 200.000 🕆 kHz Iout: 5.00 🕆 A Phase Margin: 17.92° Gain Margin: 10.57 dB Upload Compensation Rds-on-Q1: 10.000 🕆 mΩ RpT: 4.000 🗘 mΩ Gain - Magnitude Rds-on-Q2: 10.000 ⊕ mΩ RsT: 1.000 ⊕ mΩ Write Loop Coefficients Rds-on-Q3: 10.000 ⊕ mΩ np: 5.000 🕀 turns 55 Store RAM to Flash 10.000 🕀 mΩ 2.000 🗘 turns Rds-on-Q4: ns: 35 5.000 ⊕ mΩ R1: 16.20 🕀 kΩ Ð Rds-on-Q5: Errors 5.000 ⊕ mΩ 15 R2: 1.74 🌩 kΩ **Aagnitude** Rds-on-O6: TDelay: 1,0000 🕀 % Cp: 3,500.0 💭 pF -5 RL1: 1.000 ⊕ mΩ L1: 2.200 🌐 µH -25 R3: 1.00 🗘 kΩ R4: 75.00 🗘 kΩ 45 100 1,000 10,00 0 0.10 🗘 kΩ R6: 10 100 R5: 3.09 🕀 kΩ 00 1.000 ↔ mΩ Cp1: 1.0 pF Rs: Gain - Phase Schematic View ... - Capacitor Legs--15 C (µF) ESR (mΩ) ESL (nH) # Legs -35 -55 -75 -95 -115 > 47.000 22.000 4.000 2 (Degrees) 130.000 100.000 15.000 1 Add NewLeg) Delete Selected Leg Angle -135 -155 -175 Configure **Coefficient Set & Alpha Configuration** -195 -215 100 1,000 10,00 100, 🌵 Design Coefficient Set Configuration 10 00 Coef: Set A 🗸 (info) Monitor Unique open/close settings for Configure, Design, Monitor, and Status Status 2 PMBus Log Messages Show PMBus Log Fusion Digital Power Designer v1.0.0.27650 [2011-10-27] DC-DC HSFB Firmware v0.0.11.228 @ Address 88 USB Adapt 🚸 TEXAS INSTRUMENTS | fusion digital pow

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:



Power Stage - Rail #1							
Power Sta	ige Parar	neters—					
Vbus:	48.00	V 🗘 00	Vout:	12.000 💭	V		
Fs:	200.00	00 🕀 kHz	Iout:	5.00 💭	Α		
Rds-on-Q1:	10.00	Ωm 🗘 00	RpT:	4.000 🗘	mΩ		
Rds-on-Q2:	10.00	Ωm 🗘 00	RsT:	1.000 🐳	mΩ		
Rds-on-Q3:	10.00	Ωm 😳 00	np:	5.000 🗘	turns		
Rds-on-Q4:	10.00	Ωm 😳 00	ns:	2.000 🗘	turns		
Rds-on-Q5:	5.00	Ωm 🗘 00	R1:	16.20 🐳	kΩ		
Rds-on-Q6:	5.00	Ωm 😳 00	R2:	1.74 荣	kΩ		
<u>TDelay:</u>	1,000	0 🗘 %	Cp:	3,500.0 ≑	pF		
RL1:	1.00	Ωm 🗘 00	L1:	2.200 ≑	μH		
R3:	1.0	0 🌐 kΩ	R4:	75.00 🗘	kΩ		
R5:	0.1	0 🌩 kΩ	R6:	3.09 💭	kΩ		
Rs:	1.00	Ωm 🗘 00	Cp1:	1.0 💭	pF		
Schematic View							
Capacitor Legs							
	С (µF)	ESR (ms	Ω) E	SL (nH)	# Legs		
> 4	17.000	22.0	00	4.000	2		
100.000 130.000 15.000 1							
	Add NewLeg Delete Selected Leg						

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.



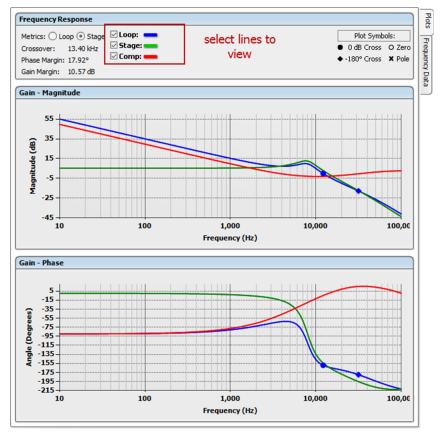


Figure 24. Bode Plots

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.



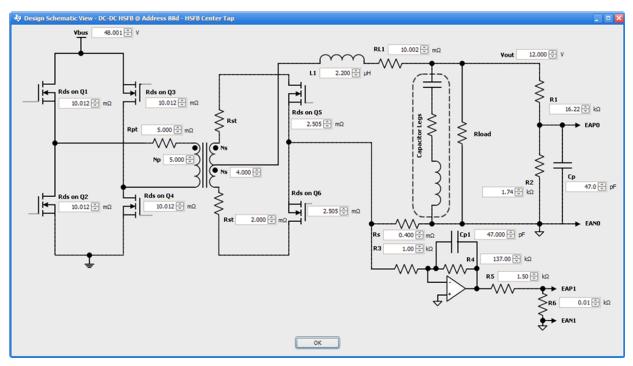
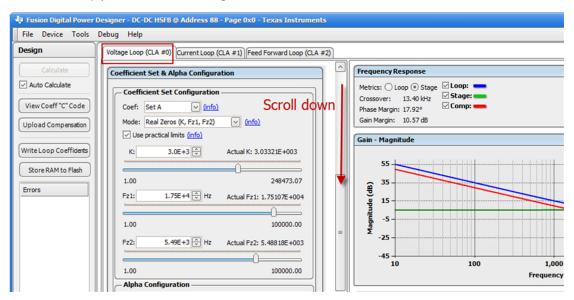


Figure 25. HSFB Schematic Being Modeled

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.



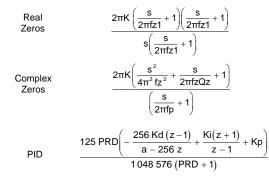




Fusion Digital Power Studio (Studio GUI)

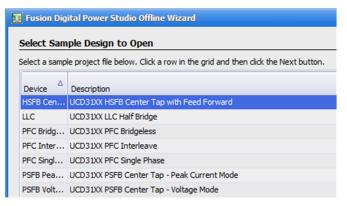
www.ti.com

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.





4.2.9.2.1 Sample Project Compensation Equations



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

z = e^{s × Plant.Ts / CLA_Design.Oversample}

(1)

4.2.9.2.2 HSFB Center Tap with Feed Forward

YN0 = 5898240;

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

4.2.9.2.3 LLC Half Bridge

 $Compensator(z) = Filter_LLC(z) \times Zopen_loop \times Gdelay2$

TDPWM = 250e-12



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

4.2.9.2.4 PFC Bridgeless, Interleave, Single Phase

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

4.2.9.2.5 PSFB Center Tap - Peak Current Mode

double VDAC = 1.6;

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

4.2.9.2.6 PSFB Center Tap – Voltage Mode

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

4.2.9.2.7 Coefficient Sets and Alpha

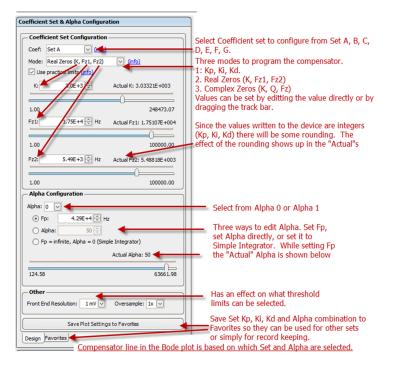


Figure 29. Coefficient Set and Alpha Configuration

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.



Fusion Digital Power Studio (Studio GUI)

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.

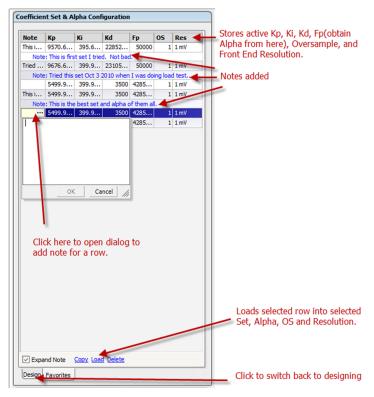
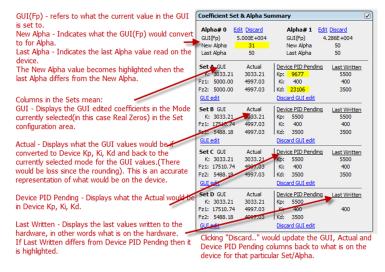


Figure 30. Favorites

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.





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

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.

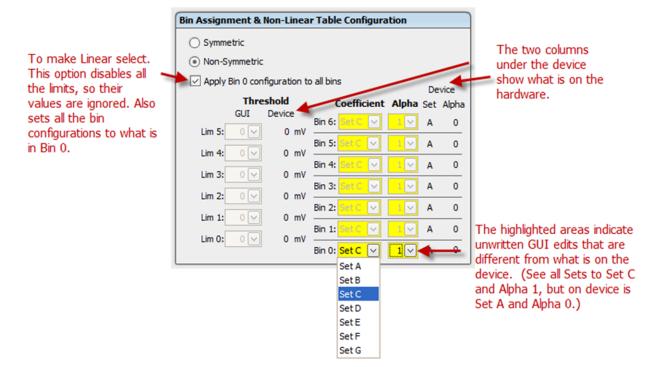


Figure 32. Apply Bin 0 to All Bins (Linear)

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.

Bin Assignment & Non-Linea	r Table Configura	tion			Bin	Assign	nent & I	Non-Linea	r Table Configura	ation		
O Symmetric						Symme	etric					
Non-Symmetric	>				0	Non-S	mmetric					
Apply Bin 0 configuration to	o all bins		Device			Apply 6	Bin 0 conf	figuration to	all bins		De	vice
Threshold	Coefficient	Alpha					Three		Coefficient	Alpha	Set	Alpha
GUI Device	Bin 6: Set G 🖂	1~	G	1		Lim 5:	GUI	Device 5 mV	Bin 6: Set G 🖂	1~	G	1
Lim 5: 5 🗸 5 mV	Bin 5: Set F 🖂	1 🗸	F	1		Lim 4:		4 mV	Bin 5: Set F 🖂	1~	F	1
Lim 4: 4 🗸 4 mV	Bin 4: Set E 🖂	0 🗸	E	0			4 🗸		Bin 4: Set E 🖂	0 🗸	Е	0
Lim 3: 3 🗸 3 mV	Bin 3: Set D 🗸	1~	D	1		Lim 3:	3 🗸	3 mV	Bin 3: Set D 🖂	1~	D	1
Lim 2: 2 🗸 2 mV	Bin 2: Set C 🗸	0 🗸	с	0		Lim 2:	2 🗸	2 mV	Bin 2: Set C 🖂	0 🗸	С	0
Lim 1: 1 🗸 1 mV	Bin 1: Set B 🗸	1	в	1		Lim 1:	1 🗸	1 mV	Bin 1: Set B 🖂	1~	в	1
Lim 0: 0 🗸 0 mV	Bin 0: Set A 🗸	1	A	1		Lim 0:	0 🗸	0 mV	Bin 0: Set A 🖂	1~	Α	1
1						-Lint 0:	0	0 mV	Bin 1: Set B	1	в	1
tive values of the other	>r		2ª			-Lim 1:	1	1 mV	Bin 2: Set C	0	с	0
	End Resolution:		3	1	1	-Lim 2:	2	2 mV	Bin 3: Set D	1	D	1
			Mus	t be j	DOSILIN	/eLim 3:	3	3 mV	Bin 4: Set E	0	E	0
-4 -5	Save Plo	t Sett				-Lim 4:	4	4 mV	Bin 5: Set F	1	F	1
-6	mannen m	m				-Lim 5:	5	5 mV	Bin 6: Set G	1	G	1
-7									birto. Set G	*	3	

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.

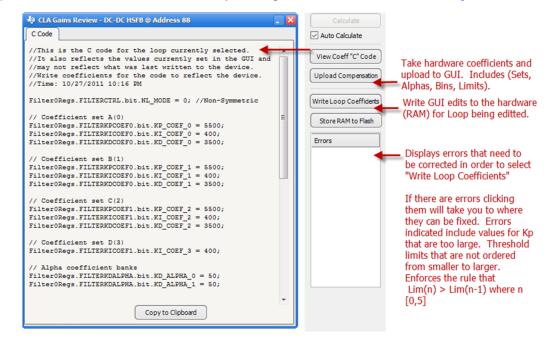


Figure 34. Writing Loop Coefficients and Global Reset of GUI Edits to Hardware Coefficients



Fusion Digital Power Studio (Studio GUI)

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.

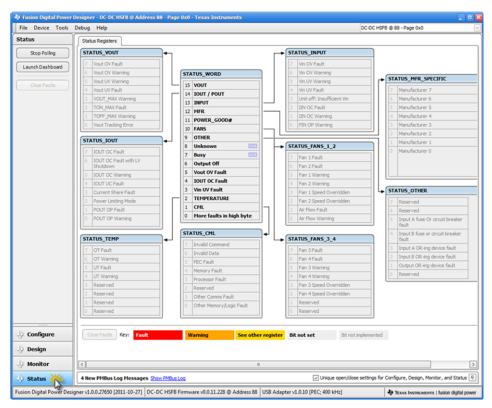


Figure 35. Status Mode

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 ...



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

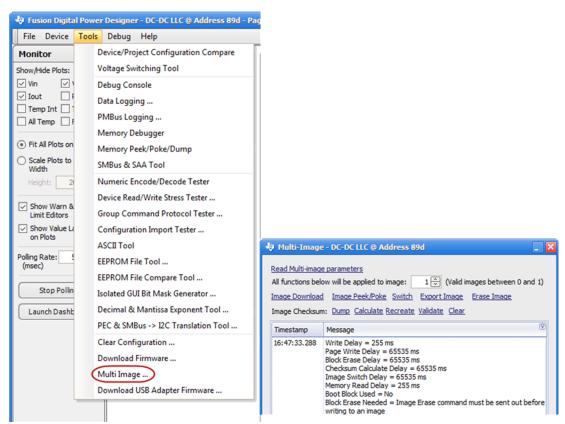


Figure 37. Multi-image

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.



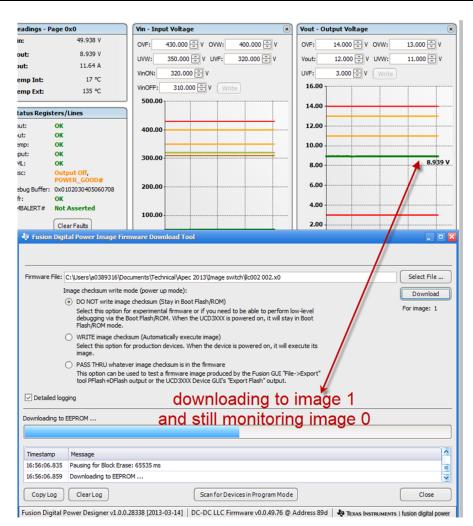


Figure 38. Downloading to an Image While Monitoring at the Same Time

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.

4.5.2 Isolated Bitmask Tool

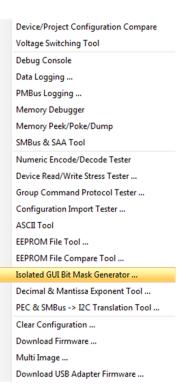


Figure 39. Tools> Isolated GUI Bit Mask Generator ...

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.





Isolated Bitmask Generator Tool		
1. Select PMBus Revision 1.2 🗸		
2. Select PMBus commands to generate bitmask.		
3. Or paste the Hex Code bitmask to see which commands are t	being used	
4. Or Upload bitmask from device CMDS_DCDC_NON_		
Select PMBus Commands Hex Code Entry	— Bitmask Generated	
PMBUS_CMD_PAGE (0x00)	//***PMBus commands selected***	
PMBUS_CMD_OPERATION (0x01)	//PMBUS_CMD_PAGE (0x00)	
PMBUS_CMD_ON_OFF_CONFIG (0x02)	//PMBUS_CMD_CLEAR_FAULTS (0x03)	
PMBUS_CMD_CLEAR_FAULTS (0x03)	■ //PMBUS_CMD_STORE_DEFAULT_ALL (0x11) //PMBUS_CMD_RESTORE_DEFAULT_ALL (0x12)	
PMBUS_CMD_PHASE (0x04)	//PMBUS_CMD_VOUT_MODE (0x20)	
PMBUS_CMD_PAGE_PLUS_WRITE (0x05)	//PMBUS_CMD_VOUT_COMMAND (0x21)	
PMBUS_CMD_PAGE_PLUS_READ (0x06)	//PMBUS_CMD_VOUT_TRANSITION_RATE (0x27)	
PMBUS_CMD_WRITE_PROTECT (0x10)	//PMBUS_CMD_VIN_ON (0x35) //PMBUS_CMD_VIN_OFF (0x36)	
PMBUS_CMD_STORE_DEFAULT_ALL (0x11)	//PMBUS_CMD_VOUT_OV_FAULT_LIMIT (0x40)	=
PMBUS_CMD_RESTORE_DEFAULT_ALL (0x12)	//PMBUS_CMD_VOUT_OV_WARN_LIMIT (0x42)	
PMBUS_CMD_STORE_DEFAULT_CODE (0x13)	<pre>//PMBUS_CMD_VOUT_UV_WARN_LIMIT (0x43) //PMBUS_CMD_VOUT_UV_FAULT_LIMIT (0x44)</pre>	
PMBUS_CMD_RESTORE_DEFAULT_CODE (0x14)	//PMBUS_CMD_IOUT_OC_WARN_LIMIT (0x4A)	
PMBUS_CMD_STORE_USER_ALL (0x15)	//PMBUS_CMD_OT_FAULT_LIMIT (0x4F)	
PMBUS_CMD_RESTORE_USER_ALL (0x16)	//PMBUS_CMD_OT_WARN_LIMIT (0x51)	
PMBUS_CMD_STORE_USER_CODE (0x17)	//PMBUS_CMD_VIN_OV_FAULT_LIMIT (0x55) //PMBUS_CMD_VIN_OV_WARN_LIMIT (0x57)	
PMBUS_CMD_RESTORE_USER_CODE (0x18)	//PMBUS_CMD_VIN_UV_WARN_LIMIT (0x58)	
PMBUS_CMD_CAPABILITY (0x19)	//PMBUS_CMD_VIN_UV_FAULT_LIMIT (0x59)	
PMBUS_CMD_QUERY (0x1A)	//PMBUS_CMD_IIN_OC_FAULT_LIMIT (0x5B) //PMBUS_CMD_IIN_OC_WARN_LIMIT (0x5D)	
PMBUS_CMD_SMBALERT_MASK (0x1B)	//PMBUS_CMD_POWER_GOOD_ON (0x5E)	
PMBUS_CMD_VOUT_MODE (0x20)	//PMBUS_CMD_POWER_GOOD_OFF (0x5F)	
PMBUS_CMD_VOUT_COMMAND (0x21)	<pre>//PMBUS_CMD_TON_RISE (0x61) //PMBUS_CMD_STATUS_BYTE (0x78)</pre>	
PMBUS_CMD_VOUT_TRIM (0x22)	//PMBUS_CMD_STATUS_WORD (0x79)	
PMBUS_CMD_VOUT_CAL_OFFSET (0x23)	//PMBUS_CMD_READ_VIN (0x88)	
PMBUS_CMD_VOUT_MAX (0x24)	//PMBUS_CMD_READ_VOUT (0x8B) //PMBUS_CMD_READ_IOUT (0x8C)	
PMBUS_CMD_VOUT_MARGIN_HIGH (0x25)	//PMBUS_CMD_READ_TEMPERATURE_1 (0x8D)	
PMBUS_CMD_VOUT_MARGIN_LOW (0x26)	//PMBUS_CMD_READ_TEMPERATURE_2 (0x8E)	
PMBUS_CMD_VOUT_TRANSITION_RATE (0x27)	//PMBUS_CMD_READ_FREQUENCY (0x95) //PMBUS_CMD_PMBUS_REVISION (0x98)	
PMBUS_CMD_VOUT_DROOP (0x28)	//PMBUS_CMD_PMBUS_REVISION (0X96)	
PMBUS_CMD_VOUT_SCALE_LOOP (0x29)	//PMBUS_CMD_MFR_MODEL (0x9A)	
PMBUS_CMD_VOUT_SCALE_MONITOR (0x2A)	<pre>//PMBUS_CMD_MFR_REVISION (0x9B) //PMBUS_CMD_MFR_LOCATION (0x9C)</pre>	
PMBUS_CMD_COEFFICIENTS (0x30)	//PMBUS_CMD_MFR_LOCATION (0X9C)	
PMBUS_CMD_POUT_MAX (0x31)	//PMBUS_CMD_MFR_SERIAL (0x9E)	
PMBUS_CMD_MAX_DUTY (0x32)	//PMBUS_CMD_MFR_IC_DEVICE (0xAD)	-
	//PMBUS_CMD_MFR_IC_DEVICE_REV (0xAE)	
🔄 Copy to Clipboard) 🕞 Save As) 📇 Print 😫	Print Preview	ок

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.



💼 Texas Instruments Fusion Digital Power Designer	►	Device GUIs
🛅 Texas Instruments Fusion Digital Power Manufacturing Tool	•	💼 Special 🔹 🕨
TI-COMM for Windows	•	🛅 Tools 🕨 🕨
m TN3270 Web	•	Occumentation & Help Center
🛅 WebEx	•	🌵 Fusion Digital Power Designer
🛅 WinZip	•	🚸 Fusion Digital Power Designer Offline Mode
Adobe Reader 9		License Agreement
🏉 Internet Explorer		🗂 Texas Instruments Home Page
😫 Outlook Express		🔂 Uninstall

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.

Open Existing Project/System, Open Sample, or Create New Project/System?	
- Ferr - meren	
Offline Mode	
Open Existing Project Open previously created project file (.xml)	
Open Sample Project Open sample reference designs	
Next > Cancel	

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



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.

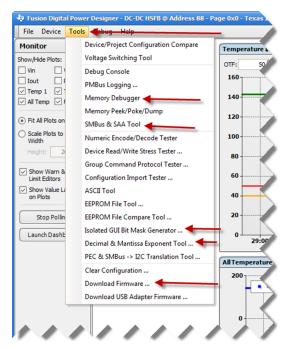
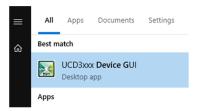


Figure 44. Studio GUI Tools Menu

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.





The Device GUI looks like Figure 46.

ettings tatus	Tools
Attached: Unknown	Scan Device in ROM Mode
ast ROM Found: IC Info: ROM Info: Package ID: ast Program Found: Address: DEVICE_ID: MFR_MODEL: MFR_REVISION:	Scan for Device in Program Mode: DEVICE_ID_DEVICE_CODE_IC_DEVICE_ID_PMBUS_REVISION Image: Scan for Device in Program Mode: DEVICE_ID_PMBUS_CODE_IC_DEVICE_ID_PMBUS_REVISION Image: Scan for Device in Program Mode: DEVICE_ID_PMBUS_CODE_IC_DEVICE_ID_PMBUS_REVISION Image: Scan for Device in Program (SendByte 0xF0 to Address 11) Command Program to iump to ROM (SendByte 0xF0) Image: SMBus/I2C\Debug Utilities) Trim [Multi-image] Image: SMBus/I2C\Debug Dump_Flash_File_Displays the contents of a flash file Set PFlash: 0 0xFF_0xAA Dump_Flash_File_Displays the contents of a flash file Set DFlash: 0xFF_0xAA 0xAA Compare_Flash_File_S Compares two flash file contents File Sport Flash File_S Compares two flash file contents File_Export_Flash Image: Full_Sport_Flash_File_S Compares two flash from the device File_Sport_Flash File_Scan program and/or data flash from the device Set DFlash: 0xFF_0xAA
Kg Timestamp Message	East-Test Tool Eases, writes a pattern, and then verifies that the pattern is present X0 to Hex Tool Converts a Tektronix Extended x0 to Intel Hex or S-Record
	d a device in ROM or program mode
Copy Log Clear Log	✓ Display all SMBus/f2C activity in k

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".

Settings	ROM scan
Status	Tools
Attached: ROM UCD31xx A64 Rev1	Scan Device in ROM Mode
Last ROM Found: IC Info: UCD31xx A64Rev1 ROM Info: ROM v6 IC v1 Package ID: 64-pin Last Program Found: Address: DEVICE_ID: MFR_MODEL: MFR_REVISION:	Scan for Device in Program Mode: DEVICE_ID_DEVICE_CODE_IC_DEVICE_ID_PMBUS_REVISION When a device is found, dump additional PMBus commands Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xF0 to Address 11) Compare Flash File Download firmware to data/program/boot flash Set DFlash: 0 VFF 0xAA Export Flash Files Compares two flash file contents Full Export Tool Reads program and/or data flash from the device Flash Test Tool Erases, writes a pattern, and then verifies that the pattern is present X0
Log	
Timestamp Message	
10:32:35.060 Reading ROM version	
10:32:35.077 SAA: BlockRead (Address 11d, Cmd 0xEC): ACK	0x00060001
10:32:35.165 Reading PKGID version	
10:32:35.171 SAA: BlockWrite (Address 11d, Cmd 0xFD, 0xFF	Adathan • Adathan
10:32:35.174 SAA: BlockRead (Address 11d, Cmd 0xFA): ACK	0x0000000
10:32:35.182 Found ROM v6 IC v1 - UCD31xx A64 Rev1	
Copy Log Clear Log	☑ Display all SMBus/I2C activity in lo

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.

Status		Tools
Attached: ROI Last ROM Four IC Info: Package ID: Last Program F Address: DEVICE_ID: MFR_MODEL MFR_REVISI	UCD31xx A64 Rev 1 ROM v6 IC v1 64-pin ound: 	Scan Device in ROM Mode: DEVICE ID DEVICE CODE IC DEVICE ID EMBUS REVISION Image: Scan for Device in Program Mode: DEVICE ID DEVICE CODE IC DEVICE ID EMBUS REVISION Image: Scan for Device in Source and the participation of the Device in Source and the part of the part in Source in S
og		
Timestamp	Message	
10:32:35.060	Reading ROM version	
10:32:35.077	SAA: BlockRead (Address 11d, Cmd 0xE): ACK 0x00060001
10:32:35.165	Reading PKGID version	
10:32:35.171	SAA: BlockWrite (Address 11d, Cmd 0xFl	0, 0xFFF7F010): ACK
10:32:35.174		
10:32:35.182	Found ROM v6 IC v1 - UCD31xx A64 Ret	1
Copy Log	ClearLog	✓ Display all SMBus/I2C activity in lo

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.

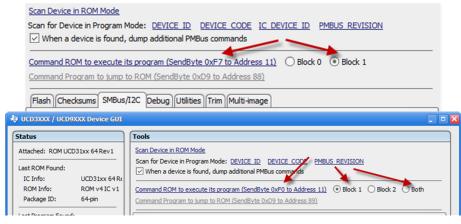


Figure 49. Executing Program for Block 0 (0xF0)



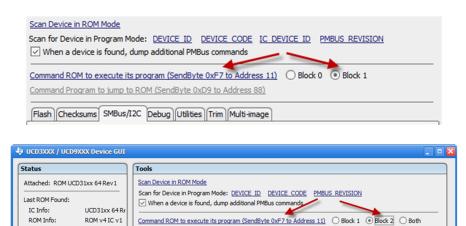


Figure 50. Executing Program for Block 1 (0xF7)

Command Program to jump to ROM (SendByte 0xD9 to Address 89)

5.3 Firmware Download Tool

Package ID:

64-pin

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

Settings	
Status	Tools
Attached: Unknown Last ROM Found: IC Info: UCD31xx A64 Rev1 ROM Info: ROM v6 IC v1 Package ID: 64-pin Last Program Found: Address: DEVICE_ID: MFR_MODEL: MFR_REVISION:	Scan Device in ROM Mode Scan for Device in Program Mode: Device in Program to imp to ROM (SendByte 0xF0 to Address 11) Command Program to imp to ROM (SendByte 0xF0) Flash Checksums SMBus/I2C Debug Utilities Trim Multi-image Firmware Download Firmware Download Download firmware to data/program/boot flash Dump Flash File Displays the contents of a flash file Export Flash Reads program and/or data flash from the device to a file Compare Flash Files Compares two flash file contents Full Export Tool Reads program and/or data flash from the device Flash Test Tool Erases, writes a pattern, and then verifies that the pattern

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:

	sion GUI's built-in firmware download tool if you need to download/reset data flash but want to keep your current PMBus configuration. 'usion GUI, this tool does not require that the device have firmware loaded or be able to execute its program.	
Data flash mode	Program flash checksum write mode (power up mode):	Select File Download
	first) Select this option for experimental firmware or if you need to be able to perform low-level debugging via the ROM. When the UCD300X is powered on, it will stay in ROM mode. WIRITE program checksum (Automatically execute Select this option for production devices. When the device is powered on, it will execute its program flash. Uldate with checksum 0X 3039	
Timestamp	Message	
	USB Adapter v1.0.11 [PEC; 400 kHz] Found (Adapter #1)	
	Looking for device in ROM mode	
	ROM V4 IG V1 detected Ready to download firmware Clear Log Other Scan for Devices in Program Mode Scan for Devices in ROM Mode	Close
copy Log	Uten Log Uter Scanfor Devices in Program Mode Scanfor Devices in ROM Mode	ciose

Figure 52. Firmware Download Screen for the UCD3138

For the UCD3138064, notice the flash block selection available:

Firmware File: C:\Users\s0389316.ENT\Documents\Wew technical\TEMP\main program_201406100939_PMBUS_EBECEDEE_with_Password\main progr Select File Data flash mode: Program flash checksum write mode (power up mode): Download data flash (mass erases first) D O NOT write program checksum (Stay in ROM) Select this option for experimental firmware or if you need to be able to perform low-level debugging via the ROM. (mode. Flash block: Download Skip data flash (WRITE program checksum (Automatically execute is powered on, it will sexu in ROM mode. Block 0 (32 kB) (Block 0 (32 kB) (Block 0 (44 kB)) Block 1 (32 kB) (Block 1 (32 kB) (Block 1 (32 kB)) OxAA (WRITE program checksum (Automatically execute is powered on, it will execute its program flash. (Validate with checksum 0x (3039) Both (64 kB) Both (64 kB) OxAA (Validate with checksum 0x (3039) PASS THRU whatever program checksum is in the firmware this option can be used to be test a firmware image produced by the Fusion GUI THE->Export* flash* output or the UCD 3000 Device GUI's "Export Flash* output. Detailed logging Execute program when download is complete (boot device, one time only) (Detailed logging Scan for device a flash program is executed (What's this?) Intersection of device a flash program is executed 10:44:55.360 USB Adapter v1.0.11 [PEC; 400 kHz] Found (Adapter #1) 10:44:55.397 Ready to download firmware 10:44:55.397 Ready to download firmware Scan for Devices in ROM		It's built-in firmware download tool if you need to download/reset data flash but want to keep your current PMBus configuration. GUI, this tool does not require that the device have firmware loaded or be able to execute its program.	
• Download data flash (mass erases first) DO NOT write program checksum (Stay in ROM) Select this option for experimental firmware or if you need to be able to perform low-level debugging via the ROM. Skip data flash Flash block: Flash block: Skip data flash Write pattern: WRITE program checksum (Automatically execute Select this option for production devices. When the device: s powered on, it will stay in ROM More and the device is program checksum (automatically execute Select this option for production devices. When the device: s powered on, it will execute its program flash. Select this option for production devices. When the device: s powered on, it will execute its program flash. Select this option for production devices. When the device: s powered on, it will execute its program flash. Select this option for production devices. When the device: s powered on, it will execute its program flash. Select this option for production devices. D vAA A Valadate with checksum 0x 3039 P ASS THU whatever program checksum is in the firmware This option can be used to test a firmware image produced by the Flusion GUI File->Export* tool PFlash+OFlash output. Detailed logging Execute program when download is complete (boot device, one time only) Secon for device after program is executed: Uk4:55.360	Firmware File: C:\User:	a0389316.ENT\Documents\New technical\TEMP\main program_201406100939_PMBUS_EBECEDEE_with_Password\main progr	Select File
Scan for device after program is executed (What's this?) Abort firmware download if device has not been factory trimmed (<u>What's this?</u>) Timestamp Message US8 Adapter v1.0.11 [PEC; 400 kHz] Found (Adapter #1) US44:56.300 Looking for device in ROM mode ROM v41C v1 detected US44:56.397 Ready to download firmware	Download data flas (mass erases first) Erase data flash Skip data flash Write pattern: DxAA Boot support Hele	 DO NOT write program checksum (Stay in ROM) Select this option for experimental firmware or if you need to be able to perform low-level debugging via the ROM. When the UCD300X is powered on, it will stay in ROM mode. WRITE program checksum (Automatically execute Select this option for production devices. When the device is powered on, it will execute its program flash. Validate with checksum 0x[3039 PASS THRU whatever program checksum is in the firmware This option can be used to test a firmware image produced by the Fusion GUT 'Files-Export' tool PFlash +0Flash output. 	Download
10:44:56.360 USB Adapter v1.0.11 [PEC; 400 kHz] Found (Adapter #1) 0:44:56.360 Looking for device in ROM mode 08:44:56.397 RoM v4 IC v1 detected 10:44:56.397 Ready to download firmware	Scan for device afte	r program is executed (What's this?) 🖂 Abort firmware download if device has not been factory trimmed (<u>What's this?</u>)	
ID:44:56.360 Looking for device in ROM mode ID:44:56.397 ROM v4 IC v1 detected ID:44:56.397 Ready to download firmware			
10:44:56.397 ROM v4 IC v1 detected 10:44:56.397 Ready to download firmware			
Copy Log Clear Log Other Scan for Devices in Program Mode Scan for Devices in ROM Mode Close	10:44:56.397 Ready	to download firmware	

Figure 53. Firmware Download for the UCD3138064

For the UCD3138A64:



🜵 Fusion Digital Power Firmware Download Tool	• 🗙
Note: Use the Fusion GUI's built-in firmware download tool if you need to download/reset data flash but want to keep your current PMBus configuration. Unlike the Fusion GUI, this tool does not require that the device have firmware loaded or be able to execute its program.	
Firmware File: C:\Users\a0389316.ENT\Documents\New technical\TEMP\main program_201406100939_PMBUS_EBECEDEE_with_Password\main progr	
Data fash mode: Program flash checksum write mode (power up mode): Download data flash (mass erases first) Do NOT write program checksum (Stay in ROM)	
Timestamp Message	-
11:45:30.701 USB Adapter v1.0.11 [PEC; 400 kHz] Found (Adapter #1)	
11:45:30.701 Looking for device in ROM mode	
11:45:30.734 ROM v6 IC v1 detected	
11:45:30.734 Ready to download firmware	
Copy Log Clear Log Other Scan for Devices in Program Mode Scan for Devices in ROM Mode Close	
Fusion Digital Power Designer v1.0.0.15439 [2014-06-17] ROM v6 IC v1 🛛 🕹 Texas Instruments fusion digital p	ower

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.

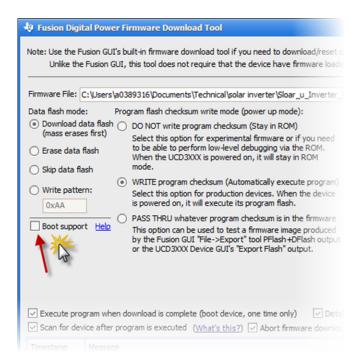


Figure 55. Boot Support

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

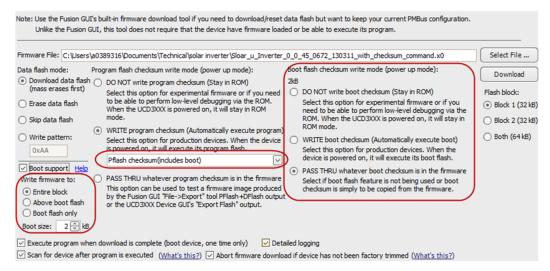


Figure 56. Bootflash Options

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.



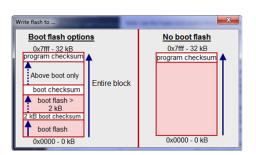


Figure 57. Firmware Writing Options

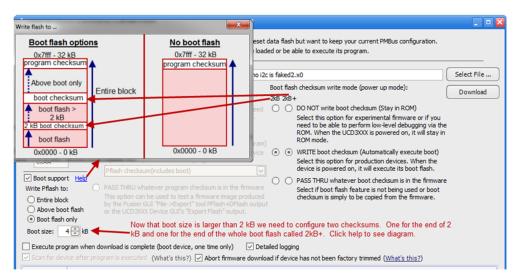
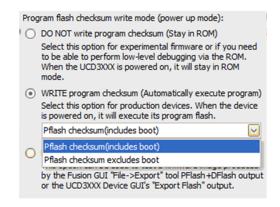


Figure 58. Two Checksums for Boot Flash Greater Than 2 kB

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.





5.3.2 Data Flash Download

Data flash m Download (mass era	
O Download	d partial
Start page	0 🌩
Final page	63 👽
🔿 Erase da	ta flash
🔿 Skip data	flash

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.



5.3.2.2.3 Helpful Tools

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



Pee	ek/Poke Memory	Dump					
	Address	Data Type		Hex Value	Numeric Value		[
~	0x000191D0	UInt32	~	0x8EBAFECA	2394619594	Read Write) (
~	0x0000000	UInt8	~			Read Write	
~	0x0000000	UInt8	\checkmark			Read Write	
~	0x0000000	UInt8	~			Read Write	
~	0x0000000	UInt8	\checkmark			Read Write	
~	0x0000000	UInt8				Read Write	

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

Peek/Pok	e I	Memo	ry D	ump														
Start Addr	ess:	0 x0	0001	9 1D0			End /	Addre	ss:	0 x	0001	9233] ;	≠ Byt	es to	Read:	: 100 🕆 Read Copy to Clipboard
	00	01	02	03	04	05	06	07	-	08	09	0A	0B	00	0D	0E	OF	0123456789ABCDEF
191D0	8E	BA	FE	CA	5F	07	5F	44	-	5D	56	D3	51	AA	8A	13	A4	D]V.Q
191E0	D7	11	55	55	8E	32	E9	BF	-	96	83	2E	F8	40	09	B 5	3B	UU.2@;
191F0	C7	47	74	D5	EF	9F	BE	C0	-	2A	AA	32	76	2D	DF	14	C6	.Gt*.2v
19200	8A	AA	58	2A	95	95	FD	55	-	F7	56	71	CD	AC	4E	AD	ЗA	X*U.VqN.:
19210	AA	AA	Α9	FA	C3	7B	CD	DD	-	C5	DD	67	15	42	AE	Α9	84	{g.B
19220	45	55	45	51	AF	C1	08	6A	-	AE	FA	A8	CF	Cl	50	49	D8	EUEQjPI.
19230	14	D9	75	55					-									uU

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.



Л	
	Tools
1xx 1p0	Scan Device in ROM Mode Scan for Device in Program Mode: <u>DEVICE ID</u> <u>DEVICE CODE</u> <u>PMBUS REVISION</u>
UCD31xx 1p0 ROM v3 IC v2 64-pin	When a device is found, dump additional PMBus commands Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xO9)
	Multi-image ROM API SMBus Debug Mass Adapter (SAA) Settings Memory Debugger Memory Peek/Poke Eirmware Download Erase/Set DFlash: 0xFE 0xAA Peek/Poke IC Registers and Firmware Variables
Ξ	Report trim status Dump Info Block Tool Erase/Set PFlash: 0xFF 0xAA Pflash checksum(includes boot) V Dump Calculate Recreate Validate Clear Boot flash 2kB checksum V Boot size: 2 kB Dump Calculate Recreate Validate Clear
	Export Flash Compare Flash Files Dump Flash File Flash Test Tool Full Export Tool X0 to Hex Tool Device Debug Tool Mantissa/Exponent Tool Iso Bitmask Tool PEC & SMBus -> I2C Translation Tool

Figure 61. Checksum Functions

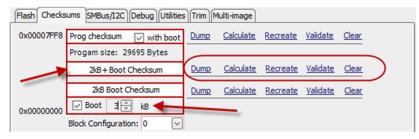


Figure 62. 2 kB Boot Checksum Functions

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



Scan Device in ROM Mode Scan for Device in Program Mode: <u>DEVICE ID</u> <u>DEVICE CODE</u> <u>PMBUS REVISION</u> When a device is found, dump additional PMBus commands
Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9)
Default Multi-image
Read Multi-image parameters
All functions below will be applied to image: 1 🖓 (Valid Images between 1 and 20)
Image Download Image Peek/Poke Switch Export Image Erase Image
Image Checksum: Dump Calculate Recreate Validate Clear

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.

Default Multi-image			
Read Multi-image param	eters		
All functions below will b	e applied to image:	1 💭 (Valid I	mages between 1 and 20)
Image Download Imag	e Peek/Poke Switch	Export Image	Erase Image
Image Checksum: Dump	p <u>Calculate</u> <u>Recreat</u>	e <u>Validate</u> <u>Clear</u>	
			Erase Image

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

👆 UCD3XXX / UCD9XXX Device GUI		_ 0 🗙
Status	Tools	
Attached: UCD310064V1 @ Address 89d Last ROM Found: IC Info: IC Info: ROM Info: Package ID: Last Program Found: Address: Address: 89d 0x59 DEVICE_ID: UCD310064V1[0.0.49.007 MFR_MODEL: UCD3138LLCEVM-028 MFR_REVISION: E3	Scan Device in ROM Mode Scan for Device in Program Mode: <u>DEVICE_ID</u> When a device is found, dump additional PM Command ROM to execute its program (SendBy Command Program to jump to ROM (SendByte (Default] Multi-image Read Multi-image parameters All functions below will be applied to image: Image Download Image Peek/Poke_Switch Image Checksum: <u>Dump_Calculate Recreations</u> Functions	#Bus commands /te 0xF0 to Address 11) 0xD9 to Address 89) 1 💬 (Valid images between 0 and 1) 1 Export Image
Log	C	
Timestamp Message		
	ACK 0x44FFFFFFFFFFFFFFFFFFFF0001040F100A	A
17:53:14.46 Witte Delay = 255 ms Page Witte Delay = 65535 ms Biock Fizee Delay = 65535 ms Checksum Calculate Delay = 65535 ms Image Switch Delay = 65535 ms Biock Trase Needed = Image Erase comma Page Size = 16 bytes Biock Size = 32766 bytes Erase Page Size = 1024 bytes	nd must be sent out before writing to an image	multi-image parameters
Copy Log Clear Log		✓ Display all SMBus/I2C activity in log
Fusion Digital Power Designer v1.0.0.30323 [2013-03-12]	TEXAS INSTRUMENTS fusion digital power	

Figure 66. Functions Enabled After Reading Multi-image Parameters

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.

👆 Fusion Digi	jital Power Image Firmware Download Tool	_ 🗆 🔀
Firmware File:	C:\Users\a0389316\Documents\Technical\Apec 2013\Image switch\llc002 002.x0	ect File
I	Image checksum write mode (power up mode):	ownload
	DO NOT write image checksum (Stay in Boot Flash/ROM)	
	Select this option for experimental firmware or if you need to be able to perform low-level debugging via the Boot Flash/ROM. When the UCD3XXX is powered on, it will stay in Boot Flash/ROM mode.	hage: 1
	 WRITE image checksum (Automatically execute image) Select this option for production devices. When the device is powered on, it will execute its image. 	
✓ Detailed log	 PASS THRU whatever image checksum is in the firmware This option can be used to test a firmware image produced by the Fusion GUI "File->Export" tool PFlash+DFlash output or the UCD3XXX Device GUI's "Export Flash" output. 	
	צ' ייצע	
Timestamp	Message	
18:21:02.870	USB Adapter v1.0.10 [PEC; 400 kHz] Found (Adapter #1)	
18:21:02.872	Looking for device in program mode	
18:21:03.614	Found DC-DC LLC Firmware v0.0.49.76 @ Address 89d in program mode	
18:21:03.614	Found PFC SinglePhase Firmware v0.0.49.76 @ Address 89d in program mode	
18:21:03.619	Ready to download firmware to image 1	
Copy Log	Clear Log Scan for Devices in Program Mode	Close
Fusion Digital F	Power Designer v1.0.0.30323 [2013-03-12] DC-DC LLC Firmware v0.0.49.76 @ Address 89d 🖑 TEXAS INSTRUMENTS fusion	digital power

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.



	Address		Data Type	U	Her Value	Numeric Value				
~	0x0000080)2	UInt8		0x00		Read		Write	
~	0x0000080)2	UInt16		0x9078	120	Read		Write	
~	0x0000080)2	UInt32	\checkmark	0x00784678	7882360	Read		Write	
~	0x0000080)2	Int8	~	0x00	0	Read		Write	
~	0x0000080)2	Int16	\checkmark	0x0078	120	Read		Write	
~	0x0000080)2	Int32	\checkmark	0x0078-678	7882360	Read		Write	
~	0x0000080)2	Linear 11	\checkmark	0x0078	120.000	Read		Write	
~	0x0000080)2	L 16 0 Unsigned		0x0078	120.000	Read		Write	
~	0x0000080)2	Block[16]		0x0078467800015	C01 .xFx	Read		Write	
~	0x0000080)2	UInt8	~			Read		Write	
					 UInt8 					
C	heck All	Unchec	k All		O UInt 16	1) (Write	Checked	
					O UInt32					
Lo	og / Outpu	t ——		_	Int8			-		
Tin	estamp	Messag	je		0					
00:	06:15.547	Read a	ddr 0x00000802L1	6 O Uı	O Int16					
00:	06:16.082	Read a	ddr 0x00000802 Lin	ear1	O Int32					
00:	06:16.541	Read a	ddr 0x00000802 Int	32: 7	O Linear 11					
00:	06:17.091	Read a	ddr 0x00000802 Int	16: 1						
00:	06:17.587	Read a	ddr 0x00000802 Int	8:0	C Linear 16 Ex	¢p: 0 💭 Mant	issa: Unsigned 🗸			

Figure 69. Image Peek/Poke



- 08 09 0A 0B 0 	0D 0E 0F 0123456789ABCDE A0 03 21xFx\!	-
- 5C 01 11 21 E	A0 03 21xFx\!	-
h 0x810 (17 bytes)		
h Ox	810 (17 bytes)	810 (17 bytes)

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.

🐶 UCD3XXX Device GUI	
Status	Tools
Attached: UCD310064V1 @ Address 89d	Scan Device in ROM Mode
Last ROM Found: IC Info:	Scan for Device in Program Mode: <u>DEVICE ID</u> <u>DEVICE_CODE</u> <u>PMBUS_REVISION</u> When a device is found, dump additional PMBus commands
ROM Info: Package ID:	Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 89)
Last Program Found: Address: 89d 0x59 DEVICE_ID: UCD310064V1 0.0 MFR_MODEL: UCD3138LLCEVM-028 MFR_REVISION: E3	Default Multi-image Read Multi-image parameters Switch image executing to: 0 🔄 Switch (Valid images between 0 and 1) Image Download Image Peek/Poke Export Image Erase Image Image Checksum: Dump Calculate Recreate Validate Clear
Log	
Timestamp Message	
00:16:13.347 SAA: BlockWrite (Address 89d, Cmc	d 0xEB, 0x4100000000000000000000000000000000000
00:16:13.347 Image Erased 0	
Copy Log Clear Log	✓ Display all SMBus/I2C activity in log
Fusion Digital Power Designer v1.0.0.41869 [2012-0	8-24] 😽 Texas Instruments fusion digital power

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."

UCD3XXX Device GUI	
Status	Tools
Attached: UCD310064V1 @ Address 89d Last ROM Found: IC Info: ROM Info: Package ID: Last Program Found: Address: 89d 0x59 DEVICE_ID: UCD310064V1 0.0 MFR_MODEL: UCD3138LLCEVM-028 MFR_REVISION: E3	Scan Device in ROM Mode Scan Device in Program Mode: DEVICE ID Device in Program Mode: Device in Program Mode: Device in Sound, dump additional PMBus commands Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 89) Default Multi-image Read Multi-image parameters Switch image executing to: 0 ⊕ Switch (Valid images between 0 and 1) Image Download Image Peek/Poke Image Checksum: Dump Calculate Recreate Validate Clear
Log	
00:17:44.090 SAA: BlockRead (Address 89d, Cmd (00:17:44.104 SAA: BlockRead (Address 89d, Cmd (00:17:44.108 SAA: BlockRead (Address 89d, Cmd (00:17:44.108 SAA: BlockRead (Address 89d, Cmd (00:17:44.108 Calculated checksum on device is: 0)	
Copy Log Clear Log usion Digital Power Designer v1.0.0.41869 (2012-08	-24] AP TEXAS INSTRUMENTS fusion digital power

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."

🐶 UCD3XXX Device GUI	
Status	Tools
Attached: UCD310064V1 @ Address 89d Last ROM Found: IC Info:	Scan Device in ROM Mode Scan for Device in Program Mode: DEVICE ID DEVICE CODE PMBUS REVISION Image: Scan for Device is found, dump additional PMBus commands DEVICE CODE PMBUS REVISION
ROM Info: Package ID:	Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 89)
Last Program Found: Address: 89d 0x59 DEVICE_ID: UCD310064V1 0.0 MFR_MODEL: UCD3138LLCEVM-028 MFR_REVISION: E3 Log	Default Multi-image Read Multi-image parameters Switch image executing to: 0 Switch image executing to: 0 Image Download Image Peek/Poke Export Image Erase Image Image Checksum: Dump Calculate Recreate Validate Clear
Timestamp Message	4
00:18:24.670 SAA: BlockRead (Address 89d, Cm	d 0xEB): ACK 0x00000012D0071A1EC0077840400000000 d 0xEB): ACK 0x000000012D0074CEBC0077840400000000 d 0xEB): ACK 0x00000000007784040077840400000000
00:18:24.689 SAA: BlockRead (Address 89d, Cmc 00:18:24.689 Checksum on device is: 0x0077840	d 0xEB): ACK 0x000000000778404007784040000000
Copy Log Clear Log	✓ Display all SMBus/I2C activity in log
Fusion Digital Power Designer v1.0.0.41869 [2012-0	8-24] 🖗 Texas Instruments fusion digital power

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."

tatus		Tools
Attached: UCD3	310064V1 @ Address 89d	Scan Device in ROM Mode
Last ROM Found IC Info:	:	Scan for Device in Program Mode: <u>DEVICE_ID_DEVICE_CODE</u> <u>PMBUS_REVISION</u> When a device is found, dump additional PMBus commands
ROM Info: Package ID:		Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 89)
Last Program Fo Address: DEVICE_ID: MFR_MODEL: MFR_REVISIO	89d 0x59 UCD310064V1 0.0 UCD3138LLCEVM-028	Default Multi-image Read Multi-image parameters Switch image executing to: 0 ÷ Switch (Valid images between 0 and 1) Image Download Image Peek/Poke Export Image Erase Image Image Checksum: Dump Calculate Recreate Validate Clear
og		
Timestamp	Message	A
00:18:37.233	SAA: BlockWrite (Address 89d, Ci	nd 0xEB, 0x06007FE0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00:18:37.237	SAA: BlockWrite (Address 89d, Ci	nd 0xEB, 0x07007FF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00:18:37.251	SAA: BlockRead (Address 89d, Cr	nd 0xEB): ACK 0x000000000007784040077 <mark>8</mark> 4040000000
00:18:37.252	Success!	
00:18:37.252	Checksum recreated 0x00778404	at address 0x00007FFC
Copy Log	ClearLog	☑ Display all SMBus/I2C activity in log

Figure 74. Recreate Image Checksum

5.5.7.4 Validate Image Checksum

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

UCD3XXX Device GUI	
Status	Tools
Attached: UCD310064V1 @ Address 89d	Scan Device in ROM Mode Scan for Device in Program Mode: DEVICE ID DEVICE CODE PMBUS REVISION
Last ROM Found: IC Info:	When a device is found, dump additional PMBus commands
ROM Info: Package ID:	Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 89)
Last Program Found: Address: 89d 0x59	Default Multi-Image
DEVICE_ID: UCD310064V1 0.0 MFR_MODEL: UCD3138LLCEVM-028	Read Multi-image parameters Switch image executing to: 0 ^ \$\stackstarter{V}\$ \$\stackstarter{V}\$
MFR_REVISION: E3	Image Download Image Peek/Poke Export Image Erase Image Image Checksum: Dump Calculate Recreate Validate Clear
Log	
Timestamp Message	<u>A</u>
00:18:51.289 SAA: BlockRead (Address 89d, Cmd	d 0xEB): ACK 0x000000012D0072F4980077840400000000
00:18:51.303 SAA: BlockRead (Address 89d, Cmd	d 0xEB): ACK 0x00000012D007621680077840400000000
00:18:51.317 SAA: BlockRead (Address 89d, Cmd	d 0xEB): ACK 0x00000000007784040077840400000000
00:18:51.321 SAA: BlockRead (Address 89d, Cmd	d 0xEB): ACK 0x00000000007784040077840400000000
00:18:51.321 Checksum is valid for image 0: checksum is valid	dsum on device = 0x778404, calculated checksum = 0x778404
Copy Log Clear Log	☑ Display all SMBus/I2C activity in log
Fusion Digital Power Designer v1.0.0.41869 [2012-0	8-24] 🛛 🎝 Texas Instruments fusion digital power

Figure 75. Validating Image Checksum

5.5.7.5 Clear Image Checksum

To clear the checksum for the image selected, click "Clear" and 0xFFFFFFF is written to that location.

🜵 UCD3XXX Device GUI	
Status	Tools
Attached: UCD310064V1 @ Address 89d	Scan Device in ROM Mode Scan for Device in Program Mode: DEVICE ID DEVICE CODE PMBUS REVISION
Last ROM Found: IC Info:	V When a device is found, dump additional PMBus commands
ROM Info: Package ID:	Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 89)
Last Program Found: Address: 89d 0x59 DEVICE_ID: UCD310064V1 0.0 MFR_MODEL: UCD3138LLCEVM-028 MFR_REVISION: E3	Default Multi-image Read Multi-image parameters Switch image executing to: 0 ÷ Switch (Valid images between 0 and 1) Image Download Image Peek/Poke Export Image Erase Image Image Checksum: Dump Calculate Recreate Validate Clear
Log	
Timestamp Message	4
00:19:02.695 SAA: BlockWrite (Address 89d, Cmd	0xEB, 0x06007FE0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00:19:02.699 SAA: BlockWrite (Address 89d, Cmd	0xEB, 0x07007FF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00:19:02.713 SAA: BlockRead (Address 89d, Cmd	0xEB): ACK 0x00000000007784040077840400000000
00:19:02.714 Success!	
00:19:02.714 Checksum deared to 0xFFFFFFFF at Copy Log Clear Log	t address 0x00007FFC
Fusion Digital Power Designer v1.0.0.41869 [2012-08	-24] & TEXAS INSTRUMENTS fusion digital power

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.

Flash Checksums SMBus/I2C Debug Utilities Trim Multi-image
Iso Bitmask Tool Used to decode/encode command bitmasks set in firmware to communicate to the GUI which PMBus commands are supported. <u>Mantissa/Exponent Tool</u> Decimal to mantissa exponent conversion tool for 16 bit signed values
CCS Conversion 3.3 to 5.5 (Beta)
Converts CCS projects from 3.3 to 5 for UCD31XX devices only.
Scan Device in ROM Mode
Scan for Device in Program Mode: DEVICE ID PMBUS REVISION
When a device is found, dump additional PMBus commands
Command ROM to execute its program (SendByte 0xF0 to Address 11)
Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 88) ROM API SMBus Debug USB Adapter (SAA) Settings Memory Debugger
Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 88)
Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 88) ROM API SMBus Debug USB Adapter (SAA) Settings Memory Debugger Peek Poke IC Registers
Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 88) ROM API SMBus Debug USB Adapter (SAA) Settings Memory Debugger Remory Peek/Poke Firmware Download Erase/Set DFlash: 0xFF_0xAA Memory Debugger
Command ROM to execute its program (SendByte 0xF0 to Address 11) Command Program to jump to ROM (SendByte 0xD9 to Address 88) ROM API SMBus Debug USB Adapter (SAA) Settings Memory Debugger Remory Peek/Poke Firmware Download Erase/Set DFlash: 0xFF_0xAA Memory Debugger Report trim status Dump Info Block Erase/Set PFlash: 0xFF_0xAA Memory Debugger

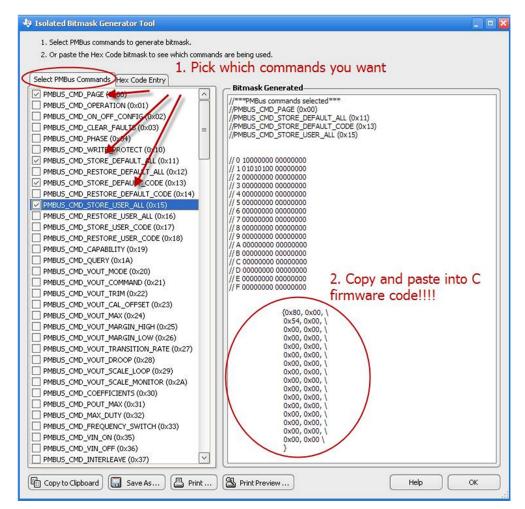
Figure 77. Click Iso Bitmask Tool



Isolated Bitmask Generator Tool Select PMBus Revision 1.2 			_ 0 🗙
2. Select PMBus commands to generate bitmask.			
 Or paste the Hex Code bitmask to see which commands 	are being	used	
5. Or paste the nex code brandsk to see which commands	one being	useu.	
Select PMBus Commands Hex Code Entry		- Bitmask Generated	
PMBUS_CMD_PAGE (0x00)		//***PMBus commands selected***	
PMBUS_CMD_OPERATION (0x01)		// Phods commands selected	<u> </u>
PMBUS_CMD_ON_OFF_CONFIG (0x02)	=		
PMBUS_CMD_CLEAR_FAULTS (0x03)		// 0 0000000 0000000 // 1 0000000 00000000	
PMBUS_CMD_PHASE (0x04)		// 2 0000000 00000000	
PMBUS_CMD_PAGE_PLUS_WRITE (0x05)		// 3 0000000 0000000	
PMBUS_CMD_PAGE_PLUS_READ (0x06)		// 4 0000000 0000000 // 5 0000000 00000000	
PMBUS_CMD_WRITE_PROTECT (0x10)		// 6 0000000 00000000	
PMBUS_CMD_STORE_DEFAULT_ALL (0x11)		// 7 0000000 0000000	
PMBUS_CMD_RESTORE_DEFAULT_ALL (0x12)		// 8 0000000 0000000	
PMBUS_CMD_STORE_DEFAULT_CODE (0x13)		// 9 0000000 0000000 // A 0000000 00000000	=
PMBUS_CMD_RESTORE_DEFAULT_CODE (0x14)		// B 0000000 0000000	
PMBUS_CMD_STORE_USER_ALL (0x15)		// C 0000000 0000000	
PMBUS_CMD_RESTORE_USER_ALL (0x16)		// D 0000000 0000000 // E 0000000 00000000	
PMBUS_CMD_STORE_USER_CODE (0x17)		// F 0000000 0000000	
PMBUS_CMD_RESTORE_USER_CODE (0x18)			
PMBUS_CMD_CAPABILITY (0x19)	_	{0x00, 0x00, \	
PMBUS_CMD_QUERY (0x1A)		0x00, 0x00, \	
PMBUS_CMD_SMBALERT_MASK (0x1B)		0x00, 0x00, \	
PMBUS_CMD_VOUT_MODE (0x20)		0x00, 0x00, \ 0x00, 0x00, \	
PMBUS_CMD_VOUT_COMMAND (0x21)		0x00, 0x00, \	
PMBUS_CMD_VOUT_TRIM (0x22)		0x00, 0x00, \	
PMBUS_CMD_VOUT_CAL_OFFSET (0x23)		0x00, 0x00, \ 0x00, 0x00, \	
PMBUS_CMD_VOUT_MAX (0x24)		0x00, 0x00, \	-
PMBUS CMD VOUT MARGIN HIGH (0x25)			
🕼 Copy to Clipboard 🔚 Save As 🖾 Print	Print	t Preview	ОК

Figure 78. Bitmask Tool

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



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.



Select PMBus Commands Hex Code Entry	ds are being used.
(0x80, 0x00, \ 0x54, 0x00, \ 0x00, 0x00, \ 0	Bitmask Generated PMBUS_CMD_PAGE PMBUS_CMD_STORE_DEFAULT_ALL PMBUS_CMD_STORE_USER_ALL // 0 1000000 00000000 // 3 0000000 00000000 // 5 0000000 00000000 // 6 0000000 00000000 // 8 0000000 00000000 // 9 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 8 0000000 00000000 // 9 0000000 000000000 // 9 0000000 000000000 // 8 0000000 00000000000000 // 9 0000000 00000000000000000000000000

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.

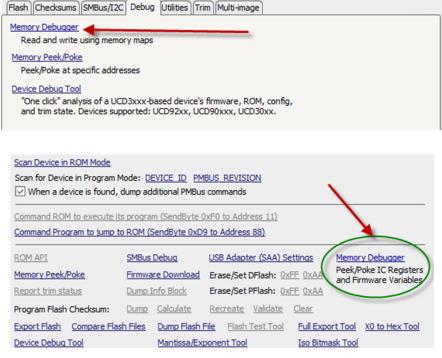


Figure 79. Memory Debugger

All	📌 Watch List									
	r/Variable Name:	~	Substring Match	✓ Filt	ter Clea	r Filter Sho	ow: 🖲 All	Registers		Flash 🔿 PFla
Name	n: All 🕻 > pmbus_checksum 🕻	Description	Type	Value	Hex	Address	Size	Category		- Tr
L.	parm index		UInt8			0x0001901F	1 byte	RAM		
	parm_mem_length		Array unsig			0x00004FBC	38 bytes	PFlash		
.	parm_mem_start		Array unsig			0x00004F70	76 bytes	PFlash		
-*	parm_offset		Int16			0x00019020	2 bytes	RAM		
	parm_size		UInt8			0x00019023	1 byte	RAM		
-*	period		UInt16	0	0x0000	0x000193F2	2 bytes	RAM		
-*	pfc_command	for APEC demo	UInt8			0x000193E9	1 byte	RAM		
$-\star$	pfc_os_enable	for APEC demo	UInt8			0x000193EC	1 byte	RAM		
	pfc_out_struct		Struct PFC			0x000193C8	27 bytes	RAM		
$-\star$	pfc_phase_2_enable	for APEC demo	UInt8			0x000193EA	1 byte	RAM		
-*	pfc_zvs_enable	for APEC demo	UInt8			0x000193EB	1 byte	RAM		
•	pmbus_buffer		Array unsig			0x00019024	40 bytes	RAM		
-*	pmbus_buffer_position		UInt8			0x0001904E	1 byte	RAM		
$-\pm$	pmbus_checksum		UInt32	2,271,5	0x87654	0x00018910	4 bytes	DFlash	REFRESH	
$-\pi$	pmbus_checksum_b	For Portability, It Is	UInt32			0x00018A30	4 bytes	DFlash		
•	pmbus_dcdc_cal		Array PMBU			0x00019108	12 bytes	RAM		
÷.	pmbus_dcdc_cal_constants		Array PMBU			0x00018880	12 bytes	DFlash		
÷.	pmbus_dcdc_cal_constants_b		Array PMBU			0x000189A0	12 bytes	DFlash		
÷	pmbus_dcdc_cal_nonpaged	second rail default v	Struct PMBU			0x00019174	4 bytes	RAM		
¢÷	pmbus_dcdc_cal_nonpaged_constants	second rail default v	Struct PMBU			0x0001888C	4 bytes	DFlash		
⊕ -)::	pmbus_dcdc_cal_nonpaged_consta	second rail default v	Struct PMBU			0x000189AC	4 bytes	DFlash		
• *	pmbus_dcdc_config		Array PMBU			0x00019068	76 bytes	RAM		
¢٠ჯ	pmbus_dcdc_config_constants		Array PMBU			0x00018890	76 bytes	DFlash		
÷.	nativa dede confia constante h		Arrest DMDL1			0-00010000	76 huton	nclash		
Ex	pand All Collapse All	Refresh All	Write Changes	In	nport	Export Sele	ected	Export All	Clear Wa	tch List
îmesta	imp Message								Copy Log	Clear Log

Figure 80. GUI Debugger



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	e	Description	Туре	Value	Hex	Address	Size	Category	_	_
.	AdcRegs	IRQ Index Offset Ve	Struct ADC			0x00040000	152 bytes	Register	REFRESH	WAITE
	CimRegs	Memory Fine Base A	Struct CIM			0xFFFFFF20	24 bytes	Register		
6	DecRegs	DPWM Individual Reg	Struct DEC			0xFFFFFE00	156 bytes	Register		
	Dpwm0Regs		Struct DPWM			0x000D0000	140 bytes	Register		
<u>.</u>	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		
3	FeCtriORegs		Struct FE_CT			0x000E0000	68 bytes	Register		
	FeCtrl 1Regs		Struct FE_CT			0x00080000	68 bytes	Register		
5	FeCtrl2Regs	Filter Status Register	Struct FE_CT			0x00080000	68 bytes	Register		
5	Filter0Regs		Struct FILTE			0x000C0000	100 bytes	Register		
-	Filter 1Regs		Struct FILTE			0x00090000	100 bytes	Register		
5	Filter 2Regs	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		
3	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		
3	TimerRegs	UART Control Regist	Struct TIMER			0xFFF7FD00	156 bytes	Register		
	UartoRegs		Struct UART			0xFFF7EC00	56 bytes	Register		
5	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.

me		Description	Туре	Value	Hex	Address	Size	Category		
- 10	FeCtrl 1Regs		Struct FE_C			0x000B0000	68 bytes	Register		[
rik:	FeCtrl2Regs	Filter Status Register	Struct FE_C			0x00080000	68 bytes	Register		[
нk	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		[
٥	COEFCONFIG	Coefficient Configur	Union COEF			0x000C001C	4 bytes	Register		
¢	FILTERKPCOEF0	Filter KP Coefficient	Union FILTE		0x00007	0x000C0020	4 bytes	Register		
			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 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.

Dit Dit		5
🖻 🤺 Bit Fields		E
🔶 KP_COEF_1 [31:16]	KP Coefficient 1	5
KP_COEF_0 [15:0]	KP Coefficient 0	5
	Film VD Conferent	

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.

	_
Change Map	
Close	F

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.



49							
- About Map/PP Files							
While only IC registers can be debugged by default, if you tell the debugger where certain Code Composer output files are located, non-local global (extern) variables can be debugged. The Code Composer files that are used are:							
 ".map - A map file defines the top-level variables in your "C" source code and at what address in memory they have been located. There is only one map file. ".pp - Contains dedarations from your source code, after they have been run through the C pre-processor. There will be one .pp file for each source code file. 							
Add the -m xxxxx.map linker option to create a map file during compilation. Add the -ppa -ppo pre-processor options to create the .pp files, one per source file.							
Note that IC register definitions are taken from your .map and .pp files, overriding the default definitions bundled with the GUI.							
Settings are saved for each unique DEVICE_ID. So for example, if you switch between UCD9244 and UCD9222, the debugger will automatically use the apprpriate files. Current Setting: Show Register Only - Manually Select Map UCD31xxx100							
Select Folder Containing Map/PP Files Select ZIP File Containing Map/PP Files							
Show Registers Only - Auto Select Map Show Registers Only - Manually Select Map							
Collapse unions in favor of bit fields Because the debugger allows you to edit bit field structures full "hex" values in addition to the lower level bit fields, you may find that alternative byte and word representations of a struct bitfield complicate debugging. Checking this box simplifies the variable node tree by only showing bit field structs when other members of a union are only simple types.							
Code Composer small enums mode was used Check this if thesmall-enums mode was used in Code Composer when the target firmware was compiled. Small enums mode reduces the size that enums take up in memory (varable length instead of fixed 4 byte liength).							
OK Cancel							

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.

<u> </u>	IEAAS
Y	INSTRUMENTS

Build Options for U	ICD3138LLCEVM_028.pjt (Debug)
General Compiler	Linker Link Order
g k -ppa ppc -al fs "\$(Proj_dir)\Deb endian=big Category: Basic Advanced Advanced Opt. Feedback Files Assembly Parser Preprocessor Diagnostics	-as o2 -ea asm fr"\$(Proj_dir)\Debug" - oug" +"/header files" -d"_DEBUG" -mt -mv4abi=tiabi Preprocessor Include Search Path (i):/header files Pre-Define Symbol (-d):DEBUG Undefine Symbol (-u): Preprocessing: With Comments (-ppc) ▼ Continue with Compilation (-ppa)
	OK Cancel Help

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.





	-
Category:	Basic
Basic Libraries	ABI (abi=): None, default to ARM9
Advanced	Suppress Banner (-q)
	Output Module:
	Output Filename (-o): .\Debug\cyclone.out
	Map Filename (-m): .\cyclone.map
	Autoinit Model: Run-Time Autoinitialization (-c) -
	Heap Size (her a)
	Stack Size (stack): 200
	Fill Value (1):
	Code Exry Point (-e):

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.

www	.ti.c	om

File									
All 😾 Watch List									
Register/Variable Name:	v s	ubstring Match 🛽	Filter Clea	ar Fi	Iter Show:		Registers 🔿 R		ash () PF
Selection: Watch List (2) > supply_state (2)	(Supply state enum for s	tate machine)							
Name	Description	Туре	Value		Address	Size	Category		
	DPWM Module Regist	Struct DPWM			0x000D0000	140 bytes	Register		
DPWMEV1		Union DPWM			0x000D0010	4 bytes	Register		
🖻 🤺 bit	DPWM Event 1 Regis	Struct DPWM			0x000D0010	4 bytes	Register		
🖃 🤺 Bit Fields		Bit Fields			0x000D0010	4 bytes	Register		
	Event 1 configuration	U Bit Field: 14	44		0x000D0011	14 bits	Register		
FeCtrloRegs	Front End Control Mo	Struct FE_CT			0x000E0000	68 bytes	Register		
- TRAMPCTRL	Ramp Control Register	Union RAMP			0x000E0000	4 bytes	Register		
🖻 🤺 bit	Ramp Control Register	Struct RAMP			0x000E0000	4 bytes	Register		
🕒 🤺 Bit Fields		Bit Fields			0x000E0000	4 bytes	Register		
- * ANALOG_PCM_INT_E	Analog Peak Current	U Bit Field: 1	0		0x000E0002	1 bit	Register		
MASTER_SEL [6:5]	Master Ramp I/F Select	U Bit Field:2	0		0x000E0003	2 bits	Register		
Filter0Regs	Filter Module Registe	Struct FILTE			0x000C0000	100 bytes	Register		
- transformed filterkpcoef0	Filter KP Coefficient	Union FILTE			0x000C0020	4 bytes	Register		
🖻 🤺 bit	Filter KP Coefficient	Struct FILTE			0x000C0020	4 bytes	Register		
🗇 🤺 Bit Fields		Bit Fields			0x000C0020	4 bytes	Register		
_ KP_COEF_0 [15:0]	KP Coefficient 0	S Bit Field: 16	29,033		0x000C0022	16 bits	Register		
+ mbus_dcdc_config_constants		Array PMBUS			0x00018890	76 bytes	DFlash		
mbus_dcdc_config_constants[0]	must be even numbe	Struct PMBU			0x00018890	76 bytes	DFlash		
		UInt16	6,144		0x00018890	2 bytes	DFlash		
	Supply state enum fo	Enum SUPPL	0 - STATE_IDLE 💌		0x0001936C	4 bytes	RAM	REFRESH	WRITE
			0 - STATE_IDLE 1 - STATE_RAMP_UP 2 - STATE_RAMP_DOV 3 - STATE_REGULATE						
) () (Refresh WatchList	Write Changes	4 - STATE_LIGHT_LOA 5 - STATE_CPCC 6 - STATE_FAULT 7 - STATE_VOUT_TRA		t Selecte	:d) D	cport All	Clear Wate	
Timestamp Message 09:51:48.488 0x000E0000: read 4 byte(s) 0	x00000001							Copy Log	Clear Log

Figure 89. Watch List with Firmware Variables

For the editable values there are up and down arrows.

0. 😂

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.



5.8 SMBus Debug

Flash Checksums SMBus/I2C Debug Utilities Trim Multi-image
SAA Adapter Settings
Configure SAA Adapter settings
SMBus Debug
Read/Write data and send commands
ROM API
Make calls to ROM functions for UCD30xx
PEC & SMBus -> I2C Translation Tool
PEC byte calculator + Converts SMBus requests into I2C transactions

Figure 90. SMBus Debug Link

The tool looks like Figure 91 when owned.

SMBus & SAA Tool Target / Miscellaneous	s	0 10							_ 🛛
Device Address: 88	58 h (All Values Below Are Hex)	 Specify 	address			(Group Pro	tocol)	SAA Settings
Read Data			- Write Data-						
Cmd	Data	Status		Cmd	Data				Status
Receive Byte		n/a	 Send Byte 	00					n/a
O Read Byte		n/a	O Write Byte	00	00				n/a
Read Word 00]	n/a	O Write Word	00	0000				n/a
Read Block E1	2233445566778899AABBCCDDEEFF01	ACK	O Write Block	00	00				n/a
Read Block 00 (96 byte)]	n/a							
Send			Send			ot include c	ount/length l also in block p		
Process Calls			_ Signals						
	Cmd Data	Status	SMBALERT#: H	ligh	Refresh				
 Process Call (Word write, word read) 	00 0000	n/a	Control Lines:	#1	#2	#3	#4	#5	
				High	High) High) High) High	Refresh All
Block Process Call	00 00	n/a		Low	Low	• Low	• Low	• Low	
(Block write, block read)	,)
Write Length: 1			GPIO Peek/Pe		b5 b4	b3	b2 b1	b0	
Read Length:	Read:		Read:						
Send			Write:			_			Read/Write
_ Log									
10:26:05.773: SMBALERT# 10:26:59.905: SAA #1: Bloc	IPmbusSignalLines: ACK signal 5 is High now High ckRead (Address 11d, Cmd 0xE1): NACK ckRead (Address 88d, Cmd 0xE1): ACK 0x22334455	66778899AABE	BCCDDEEFF01						<
Copy Log Clear Log		Adapt	er Overview						Close

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.



5.9.1 How to Access

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

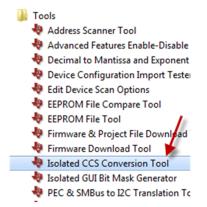


Figure 92. CCS Conversion Tool Access from Start Menu

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

Flash Checksums SMBus/I2C Debug Utilities Trim Multi-image
Iso Bitmask Tool Used to decode/encode command bitmasks set in firmware to communicate to the GUI which PMBus commands are supported.
Mantissa/Exponent Tool Decimal to mantissa exponent conversion tool for 16 bit signed values
CCS Conversion 3.3 to 5.5 (Beta) Converts CCS projects from 3.3 to 5 for UCD31XX devices only.

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).
- 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.



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.

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

Table 1. ROM/Program Commands

Device GUI	Mode:		Co			
Function	ROM/Progra m Code (hex) Command Trans. Typ		Trans. Type	Data Format	Description	
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	Program	0xE2	PARM_INFO	W	Block	PARM_INFO and PARM_VALUE
Debugger/Peek Poke	Program	0xE3	PARM_VALUE	R	Block	are both used. The first sets the address to be read and the second returns the value at that location.

Table 1. ROM/Program Commands (continued)

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.



Override PMBus Comma	nd hex co	des		×
Override command code(s):			
Commands De	fault Code	Override Code	e	
ENABLE_ROM_MODE	D9	0x d9	Override?	
PARM_INFO	E2	0x e5	Verride?	
PARM_VALUE	E3	0x e6	Verride?	
CMDS_DCDC_PAGED	E4	0x e1	Verride?	
CMDS_DCDC_NONPAGE	D E5	0x e2	Verride?	
CMDS_PFC	E6	0x e6	Override?	
ENABLE_ROM_MODE2	F9	0x f9	Override?	
DEVICE_ID	FD	0x fd	Override?	
			OK Cancel)

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.

ogram Files (x86) > Texas Instruments > Fusion Digital Power Studio3.0.37	> bin			
Name	Date modified	Туре	Size	
See.Hid.dll	6/5/2019 2:21 PM	Application extens	94 KB	
🖈 💿 Free.Misc.dll	6/5/2019 2:21 PM	Application extens	43 KB	
💉 🗟 Free.WinForms.Misc.dll	6/5/2019 2:21 PM	Application extens	23 KB	
💉 🥑 Fusion Digital Power Studio Help Center	6/5/2019 2:28 PM	Application	37 KB	
🖉 🔤 Fusion Digital Power Studio.exe	6/5/2019 2:28 PM	Application	206 KB	
🖬 Fusion Digital Power Studio.exe.config	6/5/2019 2:21 PM	XML Configuratio	2 KB	
* Eusion Tool Launcher.exe	6/5/2019 2:28 PM	Application	170 KB	
FusionConfigWriter.exe	6/5/2019 2:28 PM	Application	44 KB	
🖈 🛛 🗀 FusionCreateTesterMatlabMap.exe	6/5/2019 2:28 PM	Application	36 KB	
📧 FusionDebuggerExportDump.exe	6/5/2019 2:28 PM	Application	6 KB	
FusionDeviceExporter.exe	6/5/2019 2:28 PM	Application	43 KB	
FusionEepromDump.exe	6/5/2019 2:28 PM	Application	37 KB	
FusionFirmwareDownload.exe	6/5/2019 2:28 PM	Application	55 KB	
FusionHexToSVF.exe	6/5/2019 2:21 PM	Application	132 KB	
FusionParamReader.exe	6/5/2019 2:28 PM	Application	41 KB	
FusionParamWriter.exe	6/5/2019 2:28 PM	Application	41 KB	
FusionPMBusScan.exe	6/5/2019 2:28 PM	Application	33 KB	
FusionProjectFileTool.exe	6/5/2019 2:28 PM	Application	50 KB	
FusionScriptRunner.exe	6/5/2019 2:28 PM	Application	36 KB	
FusionX0Converter.exe	6/5/2019 2:28 PM	Application	35 KB	
EusionX0ToHex.exe	6/5/2019 2:28 PM	Application	37 KB	
ICSharpCode.SharpZipLib.dll	6/5/2019 2:21 PM	Application extens	140 KB	
MathNet.Numerics.dll	6/5/2019 2:21 PM	Application extens	865 KB	
MathNet.Numerics.IO.dll	6/5/2019 2:21 PM	Application extens	61 KB	
Microsoft.Expression.Drawing.dll	6/5/2019 2:21 PM	Application extens	120 KB	
Microsoft.Expression.Interactions.dll	6/5/2019 2:21 PM	Application extens	90 KB	
Microsoft.Office.Interop.Excel.dll	6/5/2019 2:21 PM	Application extens	1,247 KB	

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.

Command Prompt - FusionFirmwareDownload.exehelp	-		×
Microsoft Windows [Version 10.0.17763.503] (c) 2018 Microsoft Corporation. All rights reserved.			î
C:\WINDOWS\system32>cd C:\Program Files (x86)\Texas Instruments\Fusion Digital Power Studio3.0.37\bin			
C:\Program Files (x86)\Texas Instruments\Fusion Digital Power Studio3.0.37\bin>FusionFirmwareDownload.exe	he]	lp	
NAME FusionFirmwareDownload			
<pre>SYNOPSIS FusionFirmwareDownload [state rom program auto] [address 1-11,13-127rom-password password]pflash erase[skip]downloaddflash erase[skip]download [pflash-checksum calc none source] [infile firmware-image] [execute-program] [rescanrescan-delay milliseconds] [flash-block 0 1 2 3] [boot-size 1 (flashsize-1)] [boot-support-write antianblecklobeuchest [] </pre>			
entireblock aboveboot bootonly] [bflash-2k-checksum calc none source] [bflash-2kplus-checksum calc none source] [saa-identify X H L 5 times]			
FusionFirmwareDownloadhelp			~



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



Revision History

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

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