

UG286: ClockBuilderPro[™] Field Programmer Kit

This document describes how to use the ClockBuilder Field Programmer Kit ("CBPROG-DONGLE") with ClockBuilder Pro[™] ("CBPro") to support four programming models.

Refer to the text and table below for supported uses:

1. In-socket Firmware / NVM Programming

- Firmware programming of a Si5383/84 device. Silicon Labs provides a 56-pin socket adapter board for this purpose.
- NVM programming of "base" devices (e.g., Si5341A-A-GM), or any other factory "pre-programmed" device (e.g., Si5341A-A12345-GM) which has unused NVM banks. Silicon Labs provides 32-pin, 40-pin, 44-pin, 48-pin, and 64-pin QFN socket adapter boards for this purpose.

2. In-system Firmware / NVM Programming

- Firmware programming of a Si5383/84 devices already mounted on a system PCB. Users are encouraged to include a standard 10-pin header on their PCB to allow the Si538x/4x/7x/9x Field Programmer board and ribbon cable to easily connect to the USB to SPI/I2C adapter.
- NVM programming of Si538x/4x/7x/9x devices already mounted on a system PCB. Users are encouraged to include a standard 10-pin header on their PCB to allow the Si538x/4x/7x/9x Field Programmer board and ribbon cable to easily connect to the USB to SPI/I2C adapter.

3. In-system Volatile Register Programming

• Devices mounted on a PCB (e.g., use the Design Dashboard and EVB GUIs to inspect status registers, make volatile configuration updates, debug system firmware, etc.).

4. In-socket Volatile Register Programming

• Devices mounted in the socket (e.g., use the Design Dashboard and EVB GUIs to inspect status registers, make volatile configuration updates, debug system firmware, etc.).

Location of Target Device	Software Utility and Programming Model Supported		
Device	NVM Burn Tool	EVB GUI / Dashboard	
In-socket	Yes (1)	Yes (4)	
In-system	Yes (2)	Yes (3)	

Table .1. Supported Programming Models

KEY POINTS

- Shows and provides a brief explanation of the Field Programmer kit contents
- Points users to CBPro download and installation instructions
- Explains hardware configuration
- Describes the four programming models to use with the CBPROG-DONGLE
- Includes CBPROG-DONGLE and socket board schematics
- Offers bill of materials
- Includes troubleshooting appendix for common issues

1. Kit Contents

Shown below is a diagram of how the various components in the Field Programmer kit are connected to one of the QFN socket adapter boards, or to a PCB for in-system programming.



Figure 1.1. Example Hardware Configuration (Using QFN Socket Board or Customer PCB)

Figure 1.2 CBPROG-DONGLE Kit Contents on page 2 shows the kit contents for the CBPROG-DONGLE kit. Note in the figure on the following page that the 32-pin, 40-pin, 44-pin, 48-pin, and 64-pin sockets are available separately as part numbers Si5332-32SKT-DK, Si5332-40SKT-DK, Si538x4x-44SKT-DK, Si5332-48SKT-DK, Si538x4x-56SKT-DK, Si538x4x-64SKT-DK, respectively. Note the Si5372/71/92/94 44-pin devices work with Si538x4x-44SKT-DK and the Si5395 works with the Si538x4x-64SKT-DK sockets. The Si5332E/F/G/H embedded crystal products in 40-pin LGA and 48-pin LGA packages are currently not supported. The Clock Builder Pro Field Programmer resources including schematics, layout files, and BOM can be found at www.silabs.com/CBProgrammer. Note that the sockets are sold as separate kits.



Figure 1.2. CBPROG-DONGLE Kit Contents





Figure 1.3. Si5332-32SKT-DK, Si5332-40SKT-DK, Si538x4x-44SKT-DK, Si5332-48SKT-DK, Si538x4x-56SKT-DK, Si538x4x-64SKT-DK Sockets Sold Separately

UG286: ClockBuilderPro[™] Field Programmer Kit • Software Download and Installation

2. Software Download and Installation

To install the CBPro software on any Windows 7 (or above) PC, go to http://www.silabs.com/CBPro and download the ZIP file to install the software on your host PC.

3. Hardware Configuration

The Field Programmer Dongle acts as an interface between the CBPro GUI and the target device (Si5332A/B/C/D, Si5332E/F/G/H/L in 32-pin LGA, Si534x, Si537x, Si538x or Si539x IC). Connect the provided USB cable to your PC and the CBPROG-DONGLE. The CBPROG-DONGLE is then connected to the target device using the provided cables or a programming socket, depending upon the four ways you may use the programmer as detailed in Section 4. Ways You can Use the Programmer.

4. Ways You can Use the Programmer

The following four sections describe four ways you can use the CBPROG-DONGLE.

4.1 In-Socket Firmware / NVM Programming

This workflow describes the process of programming loose devices using the Si5332-32SKT, Si5332-40SKT, Si5332-48SKT, Si538X4X-56SKT, or Si538X4X-64SKT programming socket board. For nonfirmware-based solutions, this flow will "burn" a complete configuration from CBPro into available NVM in the device. Si534x-8x devices shipped from Silicon Labs have two NVM banks available to program ("burn"). Si5332 devices have a flexible NVM space. CBPro manages available NVM and programs ("burns") the available NVM when feasible. For Si5383/84 (firmware based) devices, this flow will flash a complete configuration from CBPro in to the device.

The steps needed to program a device's NVM are as follows:

1. Assuming the CBPro software is installed, connect the CBPROG-DONGLE adapter with the USB cable to the PC on which CBPro was installed. Use the USB extender cable (provided with the kit) if your host PC is located far from the CBPROG-DONGLE.





2. Insert a base or previously pre-programmed (e.g. OPN) device into the socket.

Socket and device Orientation: It is important to ensure the device is in the correct orientation before powering up the board. If not orientated correctly the software has a feature to auto-detect it is not able to read the part. Likely the reason is there is no part in the socket or it is oriented incorrectly. The part will not be damaged if oriented incorrectly. The device has two circles on the part. The smaller circle is the pin 1 indicator. Pin 1 on the socket is lined up with the U1 and dot symbol on the silk screen. 64-QFN and 44-QFN orientations are shown below. The same idea applies to 32-QFN, 40-QFN, and 48-QFN package ICs.

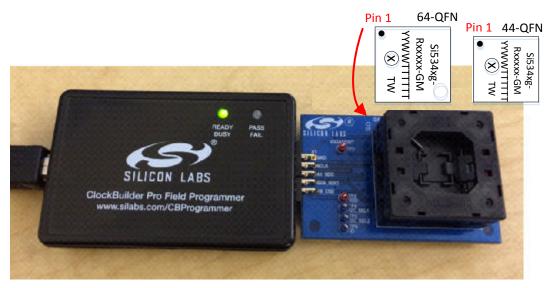


Figure 4.2. Correct Orientation of a Device in the Socket

Note: Power is not applied to the socket's VDD and VDDA pins unless explicit action by you within CBPro. It is safe to:

- · Insert or remove a device in the socket before or after the socket has been connected to the main board.
- Insert or remove a device in the socket before or after power has been applied to the main board by connecting the USB cable to your PC.

Power is only applied to the device when you perform a scan or initiate a burn. Power is off at all other times.

3. Connect the QFN Field Programmer Socket Board with the device into the CBPROG-DONGLE.



Figure 4.3. System from PC to Programming CBPROG-DONGLE Board to Field Programmer Socket Board

4. Start ClockBuilder Pro by locating the icon on your desktop or Windows Start Menu.



Figure 4.4. ClockBuilder Pro Icon

5. The ClockBuilder Pro Wizard main menu should now appear, as shown in the figure below. Select the "NVM Burn Tool" as shown. *Do not select EVB GUI.*



Figure 4.5. ClockBuilder Pro Wizard

6. If this is the first time you are launching the NVM Program Tool and no socket board has been detected, the tool will prompt you to select the device family you are targeting, as shown in the figure below:

Field Programmer Mode: Wired to Board (No Socket Detected) Target Device: No Selection No field programmer kit socket board detected. Attach socket now to burn NVM on loose parts. If you want to burn NVM on a device attached to the field programmer via wired serial connection, select the device family you are programming using the selector above.		B NVM Program Tool - ClockB
No field programmer kit socket board detected. Attach socket now to burn NVM on loose parts. If you want to burn NVM on a device attached to the field programmer via wired serial connection, select the device family	d (No Socket Detected)	Field Programmer Mode:
If you want to burn NVM on a device attached to the field programmer via wired serial connection, select the device family		Target Device:
	ected. Attach socket now to burn NVM on loose parts.	No field programmer kit so
		you are programming usin

Figure 4.6. Select Device Family Prompt

7. Once you insert the socket in the field programmer, the tool will detect it and automatically load the appropriate programming panel:

B NVM Program Tool - Clock	uilder Pro v2.15	23
Field Programmer Mode: Target Device:	Socket, QFN44 Si538x/4x (not firmware based)	
Project File: Project File Created By: Project Part: Project Design ID: Project Design Check: Project File NVM Hash:	Clear Clear OPN Lookup	
Device Part Number: Device Design ID: Device NVM State:	Scan for Device Clear	
# Valid Burns: # Burns with Error:	0 0 Program NVM	

Figure 4.7. Programming Panel

4.1.1 Programming In-socket, Firmware Based Devices

Refer to Figure 4.8 Programming In-socket, Firmware Based Devices on page 10 below.

1. Configure the I2C address and bus speed for the device.

2. Select the firmware source.

Configuration + Program from Project File

The configuration defined by the specified project + the firmware release selected in the project file will be used to generate the firmware image that will be flashed on the device. Note that different versions of CBPro may compute configuration registers differently for the same design goals as improvements are made to CBPro.

Configuration + Program from Firmware File

Flash a stand-alone hex or binary firmware file to the device. You must have previously exported the file in CBPro, or the file was sent to you by Silicon Labs. The firmware image contains both configuration and program data. This option is useful if you want to ensure the same configuration register data is flashed to the device regardless of the CBPro version this tool is running on. Firmware images can be created from the CBPro dashboard using the Export tool, selecting the stand-alone file option.

- 3. Click the "Select ..." button and select the file to flash to the device.
- 4. Click the "Scan for Device" button (optional): Click to detect device and report on part number, firmware version, and DESIGN_ID. This is optional. You can click 'Program NVM' without first scanning and all relevant pre-burn checks will be performed. Note a device scan is also performed after the NVM burn has been completed, regardless of whether the burn completed successfully or not.
- 5. Click the "Program NVM" button to flash device. In project file mode, CBPro will create a firmware image behind the scenes based on the project file configuration, and then flash this on the device. The firmware download is verified via read back.

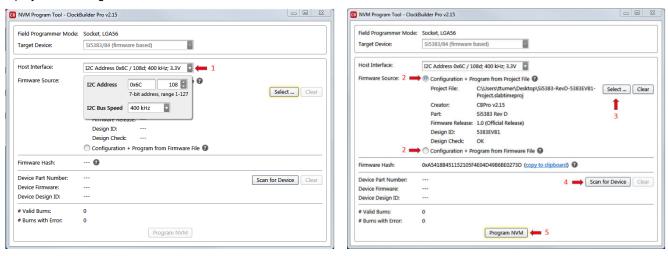


Figure 4.8. Programming In-socket, Firmware Based Devices

4.1.2 Programming In-socket, Non-Firmware Based Devices

Refer to Figure 4.9 Programming In-socket, Non-Firmware Based Devices on page 11 below.

- 1. Click the "Select Project" button and select the project file.
- 2. (Optional) Click the "Scan for Device" button to detect the device and report on part number, DESIGN_ID, and NVM bank state (number of banks already burned, number available for burn). This is optional. You can click 'Program NVM' without first scanning and all relevant pre-burn checks will be performed, such as verifying there is a bank available to burn. Note a device scan is also performed after the NVM burn has been completed, regardless of whether the burn completed successfully or not.
- 3. Click the "Program NVM" button to start the programming flow:
 - a. CBPro will compute the registers to program based on the design goals entered in the project file, using the latest algorithms embedded in CBPro.
 - b. CBPro will write volatile configuration registers corresponding to the project.
 - c. CBPro will initiate a bank burn.
 - d. CBPro will force an NVM reload on the device.
 - e. CBPro will verify the bank burn by inspecting the bank pointer and read back the programmed registers.
 - f. CBPro will rescan for the device and update burn count at the bottom of the window.

Field Programmer Mode:	Socket, QFN44	
Target Device:	Si538x/4x (not firmware based)	1
Project File:	C:\Users\tturner\Desktop\Si5342X-B-GM.slabtimeproj	Select Project Clear
Project File Created By:	CBPro v1.7.4	OPN Lookup
Project Part:	Si5342 Rev B	CFIV LOOKUP
Project Design ID:	5342BP3	
Project Design Check:	OK	
Project File NVM Hash:	0x0F8E0A6F76FE61BD2DBC1EA751316866 (copy to clipboard) 😧
Device Part Number:	Present, Si5342I-B-GM	Scan for Device Clear
Device Design ID:	(empty)	
Device NVM State:	1 bank used, 2 banks available for burn	I
# Valid Burns:	0	2
# Burns with Error:	0	

Figure 4.9. Programming In-socket, Non-Firmware Based Devices

4.1.3 In-Socket Programming Status

During the programming process and if the programming is successful, you should see the following windows.

B	Flash Firmware	×	0	Flash Firmware	- 🗆 🗙
Parsing firmwar Trying to send Success Flashing firmwa	device to bootloader mode	A. V.	Detecte Firmwar	re flashed ed Si5383 in program mode re verified to be 1.1 (Pre-Release Build 31) LJDx is TEST1	
	Close			ОК	



4.2 In-System Firmware / NVM Programming

This workflow describes the process of programming a device mounted on a PCB. For Si538x/4x/7x/9x (not firmware based) devices, this flow will "burn" a complete configuration from CBPro into one of the banks of NVM on the device, assuming an open NVM bank is available. Devices shipped from Silicon Labs always have two NVM banks available to program ("burn"). If you don't know how many banks are still open to burn on your target device, CBPro can detect and report the number of remaining NVM banks. For Si5383/84 (firmware based) devices, this flow will flash a complete configuration from CBPro into the device.

The steps needed to program an "in-system" device's NVM are as follows:

1. Assuming the CBPro software is installed, connect the adapter (CBPROG-DONGLE) board with the USB cable to the PC on which CBPro was installed.

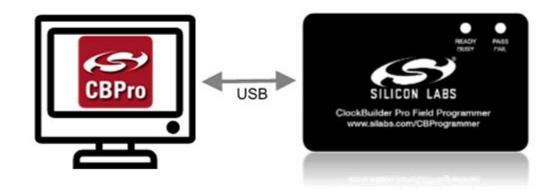


Figure 4.11. PC to CBPROG-DONGLE Connection

2. Lookup and verify the host I/O mode (I2C or SPI), the I2C address, and the interface I/O voltage level compatibility of your host's I/O voltage (for I2C or SPI) and the device.

The value set at the device register address of 0x0943 determines how the I/O supply voltages must be configured to communicate reliably with the CBPROG-DONGLE. You can look up your device host I/O voltage using the "OPN Lookup" option in the NVM Burn tool, as shown in Figure 4.12 OPN Lookup Option on page 13.

CB NVM Program Tool - ClockB	uilder Pro v2.15	ΣX
Field Programmer Mode: Target Device:	Socket, QFN64 Si538x/4x (not firmware based)	
Project File: Project File Created By: Project Part: Project Design ID: Project Design Check: Project File NVM Hash:	Select Project Clea OPN Lookup	I
Device Part Number: Device Design ID: Device NVM State: # Valid Burns: # Burns with Error:	Clea Scan for Device Clea O O Program NVM	r

Figure 4.12. OPN Lookup Option

If you have a custom OPN mounted on your board (a part number with a 5 digit code in the middle of the part number, such as Si5346B-A03260-GM), you should look up the host I/O setting (located at address of 0x0943) by selecting the OPN Lookup option. A browser will open and you will then enter in your custom OPN, as shown below.

- a. Select "Clock or Buffer".
- b. Enter in your full ordering part number (OPN). E.g., Si5346B-A03260-GM.
- c. Click the blue arrow to lookup your OPN to verify the host I/O voltage setting of your device.
- d. Click the addendum link.

SILICON LABS		Carem : News : 201492 : 120092 : 1342 Log in Register
Products - Applications	 Support & Training - Buy or Sample - 	Q Enter keyword
Silicon Late > Products > Clocks	and Oscillators > Clock and Oscillator Design Services	
Part Number Searc	ch Results	
You searched for: O Cacillator * Clock or Bu	. 🛹 🕄	
9. SI5346B-A03260-	GM D G	
Existing Custom Parts	1 result	
\$15346B-A03260-QM		
Request Date	7(2)/2014	
Part Number	S63468-A03260-GM	
Part Number Revision	0	
Product	\$(5)468	
Data Sheet	E Cata Sheet	
Data Sheet Addendum	Addendum	

3. When the utility displays the OPN's files, click on Addendum to verify the I/O Power Supply setting of your device in the Data Sheet Addendum.

"VDD (Core)" indicates the I/O supply for the Si534x-8x-7x-9x I2C/SPI interface will operate from a 1.8 V supply. "VDDA (3.3 V)" indicates the I/O supply for the Si534x-8x-7x-9x I2C/SPI interface will operate from a 3.3 V supply. "VDDD" indicates the I/O supply for the Si5332 I2C interface.

Figure 4.14 Finding VDDA Value on page 14 shows an example data sheet addendum showing VDDA (3.3 V).

Design	
Host Interface:	
I/O Power Supply: VDDA (3.3V)	
SPI Mode: 4-Wire	
I2C Address Range: 116d to 119d / 0x74 to 0x77 (selected via A0/	Al pins)

Figure 4.14. Finding VDDA Value

4. Connect/wire the pins of the CBPROG-DONGLE to your host system with the target device. Use the female-to-female ribbon cable to connect to your host board fitted with a standard 10-pin header. This assumes you included the 10-pin header on your PCB and followed the recommended pinout and connections to the target on your PCB. Note the pinout diagram and descriptions in the table below.



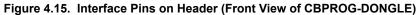


Table 4.1. Interface Pin Connections from CBPROG-DONGLE

Pin #	Description	Wire to Your PCB?	l ² C	4-wire SPI	3-wire SPI
9	A0_CSB (applies only for Si534x-8x-7x-9 x device)	3- or 4-Wire SPI	Can be used to set I ² C address bit A0 high or low. Routed to A0 device pin on the programming Field Programmer Socket Boards.	Drives the chip select sig- nal during SPI transac- tions	Drives the chip select sig- nal during SPI transac- tions
10	VDD	Never	Supplies the Core VDD voltage to the device when using a program- ming Field Programmer Socket Board. Do not use this pin for in-system pro- gramming.	Supplies the Core VDD voltage to the device when using a program- ming Field Programmer Socket Board. Do not use this pin for in-system pro- gramming.	Supplies the core VDD voltage to the device when using a program- ming Field Programmer Socket Board. Do not use this pin for in-system pro- gramming.
7	SDA_SDIO	Always	Serial data signal for I ² C transactions.	Serial data out to device for 4-wire SPI transac- tions (MOSI).	Bidirectional Serial data for 3-wire SPI transac- tions (SDIO).
8	I2C_SEL1 (applies only for Si534x-8x-7x-9 x device)	Never	Used to set I2C_SEL sig- nal high to set the de- vice for I ² C communica- tion. (Refer to specific part pinout and the pro- gramming Field Program- mer Socket Board to de- termine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL sig- nal low for SPI communi- cation. (Refer to specific part pinout and the pro- gramming Field Program- mer Socket Board to de- termine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL sig- nal low for SPI communi- cation. (Refer to specific part pinout and the pro- gramming Field Program- mer Socket Board to de- termine whether to use I2C_SEL1 or I2C_SEL2)

Pin #	Description	Wire to Your PCB?	l ² C	4-wire SPI	3-wire SPI
5	A1_CSB (applies only for Si534x-8x-7x-9 x device)	4-Wire SPI Only	Can be used to set I2C address bit A1 high or low. Routed to A1 device pin on the programming Field Programmer Socket Boards.	Serial data from device for 4-wire SPI transac- tions (MISO).	Not used
6	I2C_SEL2 (applies only for Si534x-8x-7x-9 x device)	Never	Used to set I2C_SEL sig- nal high to set the de- vice for I2C communica- tion. (Refer to specific part pinout and the pro- gramming Field Program- mer Socket Board to de- termine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL sig- nal low for SPI communi- cation. (Refer to specific part pinout and the pro- gramming Field Program- mer Socket Board to de- termine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL sig- nal low for SPI communi- cation. (Refer to specific part pinout and the pro- gramming Field Program- mer Socket Board to de- termine whether to use I2C_SEL1 or I2C_SEL2)
3	SCLK	Always	Serial clock signal for I2C transactions.	Serial clock signal for SPI transactions.	Serial clock signal for SPI transactions.
4	VDDA_VDDS	Never	Supplies the VDDA and VDDS voltages to the de- vice when using a pro- gramming Field Program- mer Socket Board. Do not use this pin for in-system programming.	Supplies the VDDA and VDDS voltages to the de- vice when using a pro- gramming Field Program- mer Socket Board. Do not use this pin for in-system programming.	Supplies the VDDA and VDDS voltages to the de- vice when using a pro- gramming Field Program- mer Socket Board. Do not use this pin for in-system programming.
1	GND	Always	GND	GND	GND
2	ID	Never	The programming Field Programmer Socket Boards provide a voltage on this pin to identify the board. For in-system pro- gramming, this pin should be grounded or not con- nected to any signal.	The programming Field Programmer Socket Boards provide a voltage on this pin to identify the board. For in-system pro- gramming, this pin should be grounded or not con- nected to any signal.	The programming Field Programmer Socket Boards provide a voltage on this pin to identify the board. For in-system pro- gramming, this pin should be grounded or not con- nected to any signal.

4.2.1 I²C Hardware Configuration

For I²C Communication connecting to an external device board, the following pins should be used from the:

CBPROG-DONGLE

- Pin 3: Serial Clock SCLK
- Pin 7: Serial Data SDA
- Pin 1: Ground

Si538x/4x/7x/9x DEVICE

- A0/CS: Drive this pin high or low to set the I 2 C Address.
- A1/SDO: Drive this pin high or low to set the I²C Address.
- I2C_SEL: Drive this pin high to select I²C communication.

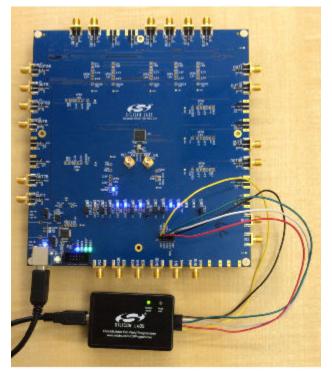


Figure 4.16. Example I2C Connection to External System Target Board Using Jumper Wires (Si5346-EVB)

When using SPI Communication with long wires as shown above it is advisable to use 6 Mb/s bus speed or less.

4.2.2 SPI 3-Wire Hardware Configuration

For 3-wire SPI communication, when connecting to an external device board, the following pins should be used from:

CBPROG-DONGLE

- Pin 3: Serial Clock SCLK
- Pin 7: Serial Data SDIO for Data In and Out
- Pin 9: A0_CSB for Chip Select
- Pin 1: Ground

Si538x/4x/7x/9x DEVICE

• I2C_SEL: Drive this pin low to select SPI communication.

4.2.3 SPI 4-Wire Hardware Configuration

For 4-wire SPI communication, when connecting to an external device board, the following pins should be used from:

CBPROG-DONGLE

- Pin 3: Serial Clock SCLK
- Pin 7: Serial Data SDIO for Data In to device (MOSI)
- Pin 5: A1_SDO for Data Out of device (MISO)
- Pin 9: A0_CSB for Chip Select
- Pin 1: Ground

Si538x/4x/7x/9x DEVICE

· I2C_SEL: Drive this pin low to select SPI communication.

If this is the first time launching the NVM Program Tool, the tool will prompt user to select the device family they are targeting:

CB NVM Program Tool - ClockBu	uilder Pro v2.15	
Field Programmer Mode:	Wired to Board (No Socket Detected)	
Target Device:	No Selection	
	Si538x/4x (not firmware based)	
No field programmer kit so	Si5383/84 (firmware based) Irn NVM on loose parts.	
If you want to burn NVM o you are programming using	n a device attached to the field programmer via wired serial connection, select the o g the selector above.	levice family

Figure 4.17. NVM Program Tool, Select Device Family

4.2.4 Programming In-system, Firmware Based Devices

Refer to Figure 4.18 Programming In-system, Firmware Based Devices on page 18 below.

After verifying the CBPro Dongle to device connections, execute the following steps. This example assumes a device is configured with an I2C address of 0x6F, and an I²C bus speed of 400 kHz.

- 1. Select "Si5383/43 (firmware based)" in the Target Device drop down.
- 2. Click the Host Interface drop down:
 - a. Enter the I²C address of the device.
 - b. Select the communication bus speed.
- 3. Select the firmware source.
 - Configuration + Program from Project File
 The configuration defined by the specified project + the firmware release selected in the project file will be used to generate the
 firmware image that will be flashed on the device. Note that different versions of CBPro may compute configuration registers
 differently for the same design goals as improvements are made to CBPro.
 - Configuration + Program from Firmware File
 Flash a stand-alone hex or binary firmware file to the device. You must have previously exported the file in CBPro, or the file
 was sent to you by Silicon Labs. The firmware image contains both configuration and program data. This option is useful if you
 want to ensure the same configuration register data is flashed to the device regardless of the CBPro version this tool is running
 on. Firmware images can be created from the CBPro dashboard using the Export tool, selecting the stand-alone file option.
- 4. Click the "Select Project ..." button and select the project file to be written to the device.
- 5. (Optional) Click the "Scan for Device" button to detect device and report on part number, firmware version, and DESIGN_ID. This is optional. You can click Program NVM' without first scanning and all relevant pre-program checks will be performed. Note a device scan is also performed after the NVM programming has been completed, regardless of whether the programming completed successfully or not.
- 6. Click the "Program NVM" button to flash device. In project file mode, CBPro will create a firmware image behind the scenes based on the project file configuration, and then flash this on the device. The firmware download is verified via read back.

Field Programmer Mode:	Wired to Board (No Se	ocket Detected)			
Target Device:	Si5383/84 (firmware	based)	1		
Host Interface:	I2C Address 0x6C / 1	08d; 400 kHz; 3.3V			
Firmware Source: 3 📩	Configuration + Pr	ogram from Proje	ct File 🕢		
	Project File:	C:\Users\tturner Project.slabtime		-RevD-5383EVB1	Select Clear
	Creator:	CBPro v2.15			1
	Part:	Si5383 Rev D			4
	Firmware Release:	1.0 (Official Rele	ase)		
	Design ID:	5383EVB1			
	Design Check:	OK			
3 🛁	Configuration + Pr	ogram from Firm	ware File 🕢		
Firmware Hash:	0xA5418B451152105F	4E04D49B6BE027	3D (copy to clip)	board) 🕜	
Device Part Number:				5 🔿 s	clear
Device Firmware:					
Device Design ID:					
# Valid Burns:	0				
# Burns with Error:	0				

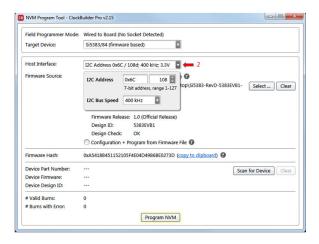


Figure 4.18. Programming In-system, Firmware Based Devices

4.2.5 Programming In-system, Non-firmware Based Devices

Refer to Figure 4.19 Programming In-system, Non-firmware Based Devices on page 19 below.

After verifying the CBPro Dongle to device connections, execute the following steps. This example assumes a device is configured with the host I^2C interface operating in 3.3 V I/O mode with an I^2C address of 0x6F, and an I^2C bus speed of 400 kHz.

- 1. Select "Si538x/4x/7x/9x (not firmware based) in the Target Device drop down.
- 2. Click the Host Interface drop down: (Review: host I/O mode (I2C or SPI), the I2C address, and I/O voltage level to determine these settings)
 - a. Select communication protocol for the device.
 - b. Select the I/O voltage for the device
 - c. For $\mathsf{I}^2\mathsf{C},$ enter the address of the device.
 - d. Select the communication bus speed.
- 3. Click the "Select Project ..." button and select the project file to be written to the device.
- 4. (Optional) Click the "Scan for Device" button to detect the device and report on part number, DESIGN_ID, and NVM bank state (number of banks already burned, number available for burn). This is optional. You can click Program NVM' without first scanning and all relevant pre-programming checks will be performed, such as verifying there is a bank available to burn. Note a device scan is also performed after the NVM burn has been completed, regardless of whether the burn completed successfully or not.
- 5. Click the "Program NVM" button to start the programming flow:
 - a. CBPro will compute the registers to program based on the design goals entered in the project file, using the latest algorithms embedded in CBPro.
 - b. CBPro will write volatile configuration registers corresponding to the project.
 - c. CBPro will initiate a bank burn.
 - d. CBPro will force an NVM reload on the device.
 - e. CBPro will verify the bank burn by inspecting the bank pointer and read back the programmed registers.
 - f. CBPro will rescan for the device and update burn count at the bottom of the window.

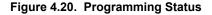
Field Programmer Mode:	Wired to Board (N	No Socket Detected)	
Target Device:	Si538x/4x (not fi	irmware based)	1
Host Interface:	I2C Address 0x6	C / 108d; 400 kHz; 3.3V 📳 📛	2
Project File: Project File Created By: Project Part: Project Design ID: Project Design Check: Project File NVM Hash:	Protocol I/O Voltage	SPI 4-Wire SPI 3-Wire I2C	3 Select Project Clear
Device Part Number: Device Design ID: Device NVM State:	I2C Address	0x6C 108 7-bit address, range 1-127 400 kHz	4 Scan for Device Clear
# Valid Burns: # Burns with Error:	0 0		

Figure 4.19. Programming In-system, Non-firmware Based Devices

4.2.6 Programming Status

During the programming process and if the programming is successful, you should see the following windows:

СВ	Flash Firmware	X	в	Flash Firmware	-	×
Parsing firmwar Trying to send Success Flashing firmwa	device to bootloader mode	×.	Detecte Firmwar	re flashed ed Si5383 in program mode re verified to be 1.1 (Pre-Release Build 31) J_IDx is TEST1		
	Close			ОК		



4.3 In-System Volatile Register Programming and Register Debug

This workflow allows users to use the full CBPro configuration Wizard and EVB GUI to make volatile changes to a device's configuration and inspect the state of various status registers. There are two ways you can interact with your PCB-based device using the field programmer:

- Use CBPro Design Dashboard to edit your device configuration, and write out changes directly to your device.
- Launch the EVB GUI, to inspect registers.

All of the relevant CBPro features available when working with a Silicon Labs EVB will be available to you, with these exceptions:

- There is no voltage regulator control or voltage/current readings of any kind.
- · You must configure the host interface settings so that CBPro can use the device correct communication scheme/wire out.
- If you write out your design/project file, all registers configured via the "Host Interface" section of the wizard **are** written to the device (these registers are skipped when writing a design to a Silicon Labs EVB).

4.3.1 Using the CBPro Design Dashboard

When you launch CBPro, instead of clicking the NVM Burn Tool, open your existing project file or a sample file to open the design dashboard window as shown in the figure below.

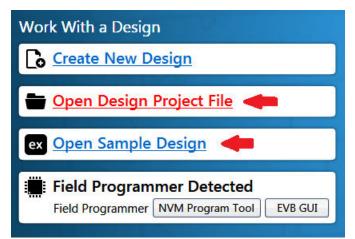


Figure 4.21. Open Design Project File, and see Field Programmer Detected

4.3.2 Overview of CBPro Configuration Wizard and the Field Programmer

When you open a ClockBuilder Pro project file, you are taken to the design dashboard. This is a gateway to perform activities against your design, including writing your project's configuration to a device using the CBPro Dongle. For example, in the figure below, a Si5345 project has been opened and the CBPro Dongle has been detected, and no socket is present:

CB Open Sample	e Design - ClockBuilder Pro 📃 🔍
ClockBuilder Pro v2.14.4 🌤 Design Dashboard ▼	SILICON LABS Configuring Si5345 Rev D
Loaded sample design 5345EVB1. You can review or edit the or section below or using the pulldown step menu above.	configuration by clicking a link in the "Edit Configuration with Wizard"
Edit Configuration with Wizard Design ID & Notes · Revision · Host Interface · XA/X Free Run · ZDM · Inputs · Input Select · Outputs · Plan DCO · Output Skew · Output Drivers · DSPLL · LOS · Output Skew · Output Drivers · DSPLL · LOS · Output · INTR Save Design to Project File Your configuration is stored to a project file, which can opened in ClockBuilder Pro at a later time.	Interface: I2C Address 0x68 / 104d; 400 kHz; 3.3V Write Design to DUT Open EVB GUI
Design Report & Datasheet Addendum You can view a design report (text) or create a draft datasheet addendum (PDF) for your design.	Documentation <u>Si5345/44/42 Rev D Family Reference Manual</u> <u>Si5345/44/42 Rev D Datasheet</u> <u>Si5345 Rev D EVB User's Guide</u>
Filicon Labs Cloud Services You can create a custom part number for your design, which can be used to order factory pre-programmed devices. Or request a phase noise report for this design	n.
Frequency Plan Valid 🕢 Design OK 😗 Pd: 1.215 W, Tj: 9	96 °C Home Close

Figure 4.22. Overview of CBPro Configuration Wizard and the Field Programmer

With a click of the "Write Design to DUT" button, you can reconfigure the Si5345 in-system to test changes to your design. The "Open EVB GUI" button can be used to launch the EVB GUI and peek/poke registers on the in-system device. See Section 4.3.4 Using the EVB GUI with In-system Devices to learn more.

4.3.2.1 Using the Dashboard with In-system Devices

If the CBPro Dongle is connected via USB and detected by CBPro, you will see will see a pulldown to configure the host interface between the dongle and your PCB, as shown in the figure below. Refer to Section 4.2 In-System Firmware / NVM Programming for information to connect the CBPro Dongle to your hardware.

ſ	Field P	rogrammer Detected									
l	 Target: Wired to PCB (serial)										
l	Interface:	I2C Address 0x6C / 108d; 100 kHz; 3.3V									
l	Write	Design to DUT Open EVB GUI									

Figure 4.23. Field Programmer Detected

Click the interface pulldown to configure the communication interface, as shown in the figure below. For firmware based devices (e.g. Si5383), the I2C address and bus speed need to be configured. For non-firmware based devices (e.g. Si5340, Si5341), The communication protocol and the I/O voltage need to be configured. If the communication protocol is I2C, the address and bus speed will need to be configured as well.

								rogrammer Detected Wired to PCB (serial) I2C Address 0x6C / 108d; 100 kHz; 3.3V			
							Interface:				
		_					Write	Protocol	O SPI 4-Wire SPI 3-Wire		
	Field Progra Target: Wired Interface: I2C Ac	to PCB (ser	ial)	kHz; 3.3V		Þ	Export You can e in-system	I/O Voltage	 I2C 3.3 V 	ble for	
	Write I I2C A	ddress	0x6C 7-bit address	108 📘				I2C Address	0x6C 108		
Þ	Export You can e in-system progr	us Speed amming.	100 kHz		ble for			IZC DUS Speed		J	

Figure 4.24. Communication Interface Selection

Once configured, you can write out your design to the device by clicking the Write Design to DUT button:



Figure 4.25. Write Design to DUT

Or on any configuration page in the wizard:

Write to FP < Back Next > Finish	Cancel

Figure 4.26. Write to FP

When you initiate a project write to the DUT, CBPro will first try to verify the DUT is present via the communication interface you have configured. This is normally accomplished by trying to read device identification register on the device, such as PN_BASE on Si538x/4x/7x/9x devices.

If it cannot read these registers, the DUT write will be aborted and you will see an error message like the example shown in the figure below:

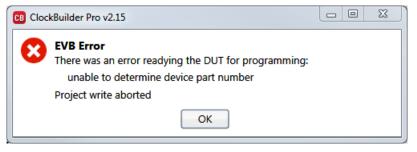


Figure 4.27. Error Message

4.3.2.2 Using the CBPro Dashboard with In-socket Devices

In the design dashboard, you will see a pulldown to configure the host interface between the CBPro Dongle and the socket. If the connected socket is not compatible with the selected CBPro project file, an error message will be displayed and the interface configuration pulldown will be disabled, as shown in the figure below.

Field Programmer Detected Target: Socket, LGA56	Field Target:
Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V	Interfac
Write Design to DUT Open EVB GUI	Writ

Field Programmer Detected									
 Target: Socket, QFN44 (not compatible with Si5383)									
Interface:	I2C Address 0x6C / 108d; 100 kHz; 3.3V								
Write	Design to DUT Open EVB GUI								

24

Figure 4.28. Socket Compatibility

Click the interface pulldown, configure the interface, and click the slider power to turn on the socket power. For firmware based devices (e.g. Si5383), the I2C address and bus speed need to be configured. For non-firmware based devices (e.g. Si5340, Si5341), the communication protocol and the I/O voltage need to be configured. If the communication protocol is I2C, the address and bus speed will need to be configured, as shown in the figure below.

Note: Manually powering up the socket is an optional step. If you click the "Write Design to DUT" button, CBPro will automatically power up the socket (and you will see it switch from Off to the On state). Socket power refers to VDD and VDDA power on the device.

	Field Programmer Detected Target: Socket, QFN44 Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V
Field Programmer Detected Target: Socket, LGA56 Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V Write I2C Address 0x6C 108	Write Protocol SPI 4-Wire SPI 3-Wire SPI 3-Wire I2C Vou can e in-system VO Voltage 3.3 V
Export You can e In-system programming.	I2C Address 0x6C 108 7-bit address, range 1-127 I2C Bus Speed 100 kHz Si5345/44 Si5345/44/42 Rev D Datasheet Si5344 Rev D EVB User's Guide

Figure 4.29. Interface Settings

Once configured, you can write out your design to the device by clicking the Write Design to DUT button:

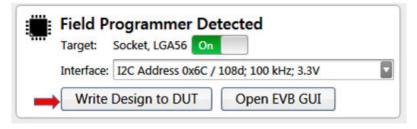


Figure 4.30. Write Design to DUT

Or on any configuration page in the wizard:

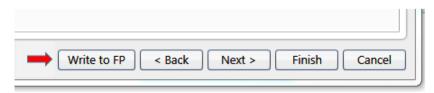


Figure 4.31. Write Design to FP

4.3.3 Launching the CBPro EVB GUI

From the CBPro Wizard screen, click the EVB GUI button to open the EVB GUI screen.

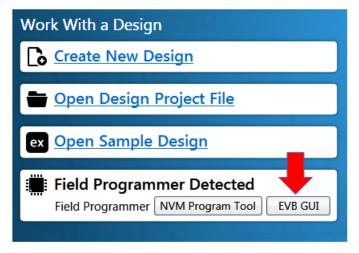


Figure 4.32. Open EVB GUI Screen

If this is the first time launching the EVB GUI and no socket board is detected, the tool will prompt user to select the device family they are targeting:

1	Config Sca	an
Control	Device Family	No Selection
	Protocol	O SPI 4-Wire
		O SPI 3-Wire
		I2C
	I/O Voltage	3.3 V
	I2C Address	0x68 104
	IZC Address	7-bit address, range 1-127
_	I2C Bus Speed	400 kHz

Figure 4.33. Select Device Family Prompt

If a socket is connected, the family is auto selected based on the socket. The tool polls for socket state every 500 milliseconds and will detect if a socket is present or has been changed.

Family:	Si538x/4x (not firmware based)
Target:	Socket, QFN44
Socket Power:	Off
Interface:	I2C Address 0x68 / 104d; 400 kHz; 3.3V
Part Number:	
Design ID:	

Figure 4.34. Socket Detected, Auto-selected Family Prompt

4.3.4 Using the EVB GUI with In-system Devices

Connect the CBPro Dongle to the PCB mounted device. Refer to Section 4.2 In-System Firmware / NVM Programming for information to connect the CBPro Dongle to your hardware. Click the Config button and click the Device Family pulldown to select either a firmware based device or a non-firmware based device. Then configure the communication protocol, bus speed and I/O voltage (non-firmware devices) for the device, as shown in the figure below.

Note: For firmware based devices the communication protocol available is I2C with a 3.3 V I/O voltage. For non-firmware based devices, there is a selection of SPI 4-wire, SPI 3-wire, or I2C and the I/O voltage must be selected.

28

							FIEIU PIO	grammer		
							Family:	Si538x/4x (i firmware ba		
							Target:	Wired to PO	CB (serial)	
							Interface:	I2C Address	s 0x6C /	
								108d; 100 k	(Hz; 3.3V	
		Field P	ogrammer				Part Num	ber:		
		Family:	Si5383/84	firmware			Design ID):		
			based)				o congri to			
		Target:	Wired to P	CB (serial)				Config Sca	an	
		Interfac	e: I2C Addres	s 0x6C /						
			108d; 100 l	Hz; 3.3V			Control	Device Family	Si538x/4x (not	firmware based) 🔽
Field Progra	immer									
Family:	Si538x/4x (not		imber:					Protocol	🔿 SPI 4-Wire	
	firmware based)	Design	ID:				c .		O SPI 3-Wire	
Target:	Wired to PCB (serial)		Config Sca				L		() I2C	
Interface:				<u> </u>		_	ſ		0 120	
Interface:	SPI 4-Wire; 1 MHz; 3.3V	Contro	Device Family	Si5383/84 (fir	mware based)	-		I/O Voltage	3.3 V	1
	2.24	Contro	Denceranny	515565764 (11	inware basedy		L	-		
Part Numbe	-		I2C Address	0.60	108		6	I2C Address	0x6C	108
Design ID:			I2C Address	0x6C			6	ize nutress		
Design ID.	•			7-bit address, r	range 1-127		U		7-bit address, rar	ige 1-12/
6	onfig Scan		I2C Bus Speed	100 kHz			-	I2C Bus Speed	100 kHz	1
			ince bus speed	100 1012						
			L.					ELIVER LAUREN		

Figure 4.35. Configuring an In-system Device

After the configuration is complete, click the Scan button. The Part Number and Design ID fields should update with the device information along with the Info tab fields, as shown in Figure 4.36 In-System Scan Prompt and DUT Register Editor Tab on page 29. Now the DUT Register Editor tab can be used to make volatile register value changes to the device and the Status Registers tab can be used to monitor the status of the device.

		Field Programmer Identification:	
		Serial Number:	00-00-04-06-2C-CE
		DUT ID Registers:	Refresh ID Registers
		FIRMWARE_TYPE	Production
		FIRMWARE_MAJOR_REV	1
		FIRMWARE_MINOR_REV	0
		FIRMWARE_BUILD	19
		DEVICE_PN_BASE	Si5383
		DIE_REV	B1
		DEVICE_REV:	D
		VCO_VARIANT	0
Field Programmer		TEMP_GRADE	Industrial
Family:	Si5383/84 (firmware	PKG_ID	3
Target:	based) Wired to PCB (serial)	BASELINE_ID	18
Interface:	I2C Address 0x6C /	DEVICE_GRADE	Α
	108d; 100 kHz; 3.3V	OPN_ID	06791
Part Number: Si5383A-D06791-GM		OPN_REVISION	0
Design ID:	5383EVB1	DESIGN_ID	5383EVB1
Config Scan		TOOL_VERSION	ClockBuilderPro v2.14.3.0

Figure 4.36. In-System Scan Prompt and DUT Register Editor Tab

4.3.5 Using the EVB GUI with In-socket Devices

CBPro will detect the connected socket when the EVB GUI is started. Click the Config button to configure the communication protocol, address (I2C), bus speed, and the I/O voltage (non-firmware based devices), as shown in the figure below.

Note: For firmware based devices the communication protocol available is I2C with a 3.3 volt I/O voltage. For non-firmware based devices, there is a selection of SPI 4-wire, SPI 3-wire, or I2C and the I/O voltage must be selected.

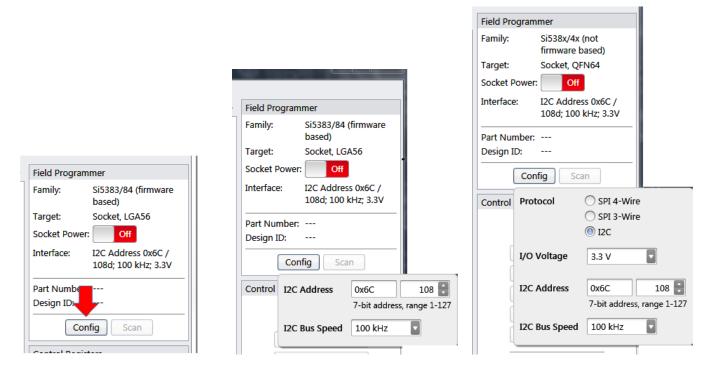


Figure 4.37. Configuring an In-socket Device

After the configuration is complete, click the Socket Power slider and the Scan button. The Part Number and Design ID fields should update with the device information along with the Info tab fields, as shown in Figure 4.38 In-Socket Scan Prompt and DUT Register Editor Tab on page 31. Now the DUT Register Editor tab can be used to make volatile register value changes to the device and the Status Registers tab can be used to monitor the status of the device.

		Field Programmer Identification:	
		Serial Number:	00-00-04-06-2C-CE
		DUT ID Registers:	Refresh ID Registers
		FIRMWARE_TYPE	Unknown
		FIRMWARE_MAJOR_REV	0
		FIRMWARE_MINOR_REV	17
		FIRMWARE_BUILD	0
		DEVICE_PN_BASE	Si5383
		DIE_REV	B1
		DEVICE_REV:	D
Field Programmer		VCO_VARIANT	0
Family:	Si5383/84 (firmware	TEMP_GRADE	Industrial
Target:	based) Socket, LGA56	PKG_ID	3
Socket Powe		BASELINE_ID	18
Interface:	I2C Address 0x6C /	DEVICE_GRADE	Α
	108d; 100 kHz; 3.3V	OPN_ID	06791
Part Number	: Si5383A-D06791-GM	OPN_REVISION	0
Design ID:	(empty)	DESIGN_ID	
Co	nfig Scan 🛑	TOOL_VERSION	ClockBuilderPro v2.11.4.0

Figure 4.38. In-Socket Scan Prompt and DUT Register Editor Tab

5. CBPROG-DONGLE Schematic

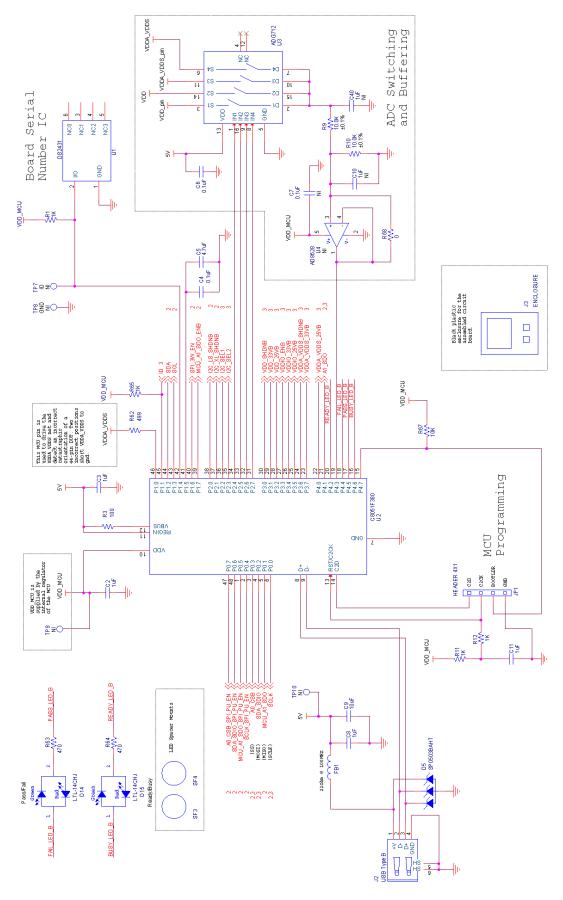


Figure 5.1. CBPROG-DONGLE Schematic (1 of 3)

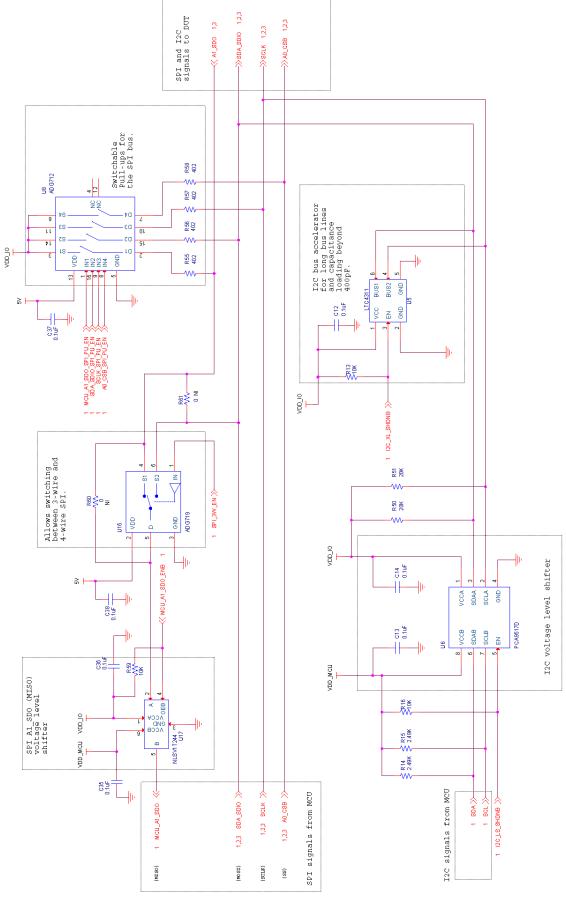


Figure 5.2. CBPROG-DONGLE Schematic (2 of 3)

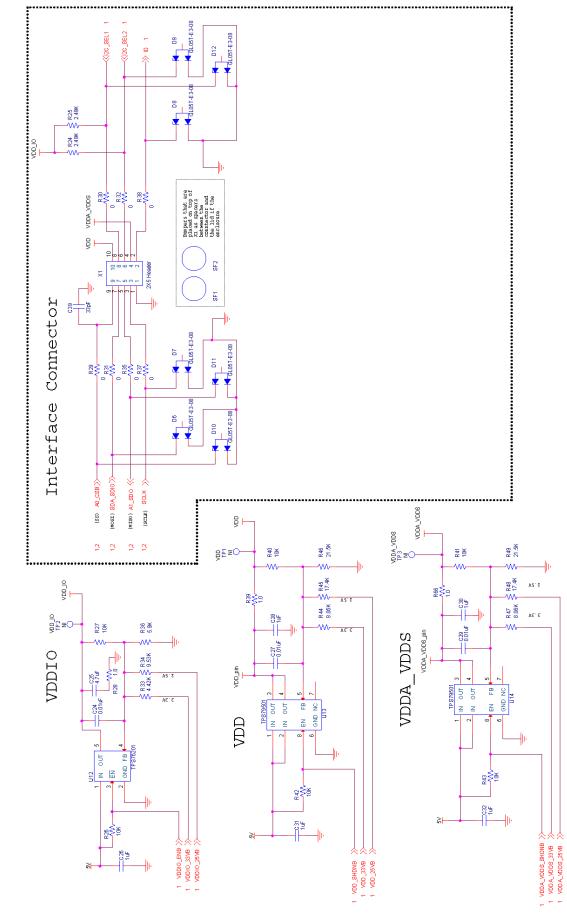


Figure 5.3. CBPROG-DONGLE Schematic (3 of 3)

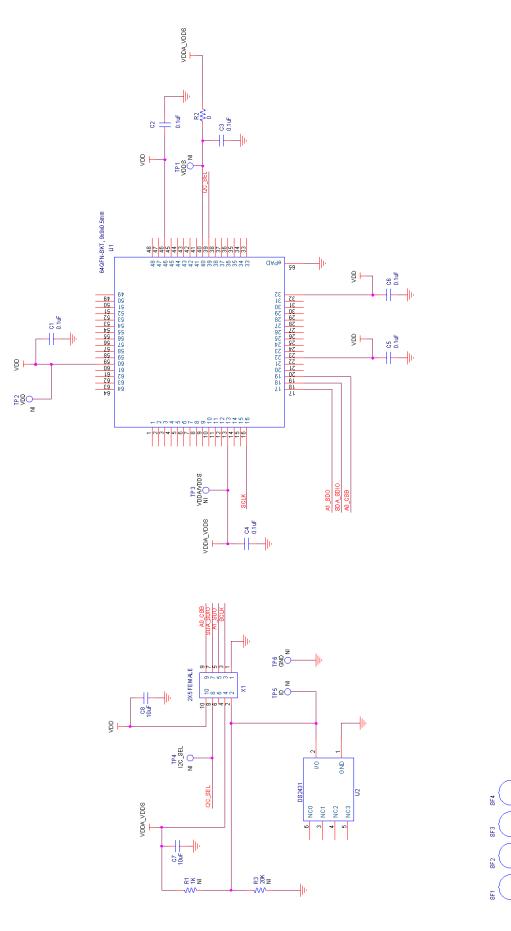


Figure 5.4. 64-Pin Socket Board Schematic

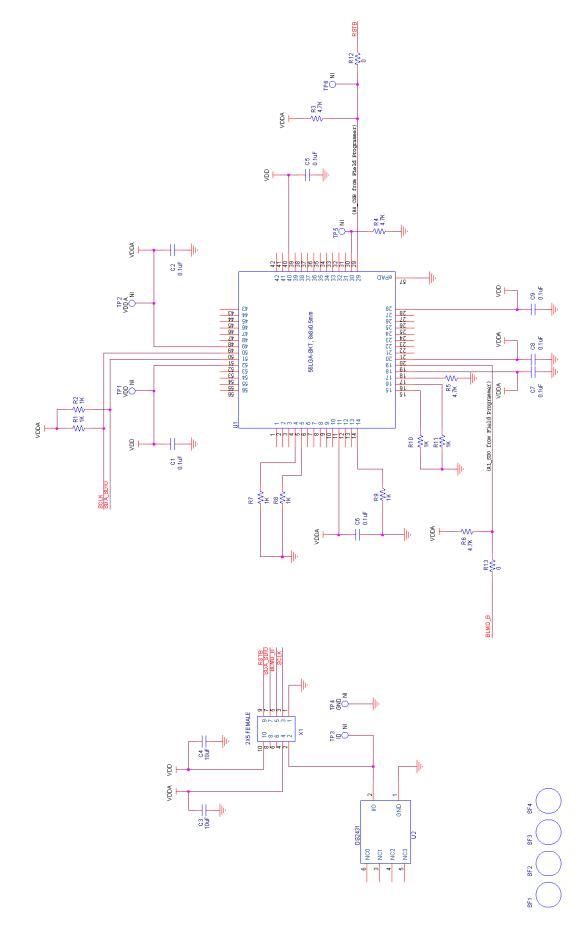


Figure 5.5. 56-Pin Socket Board Schematic

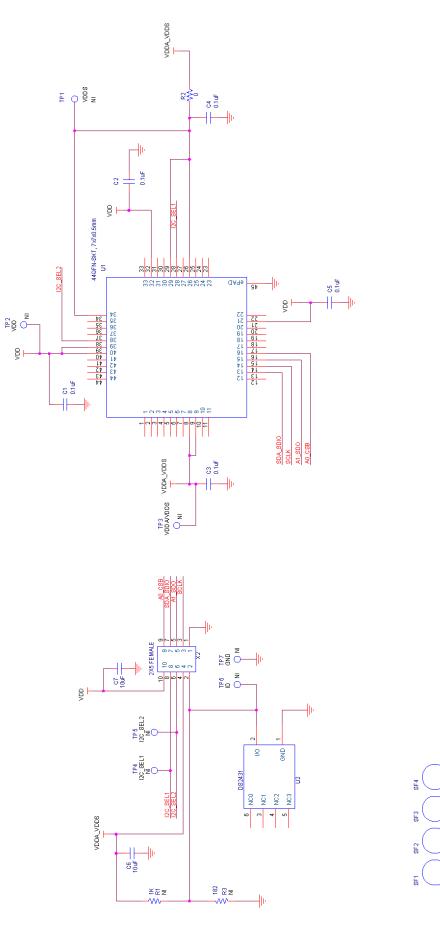
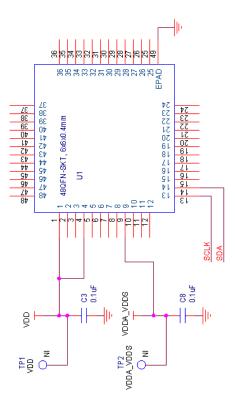
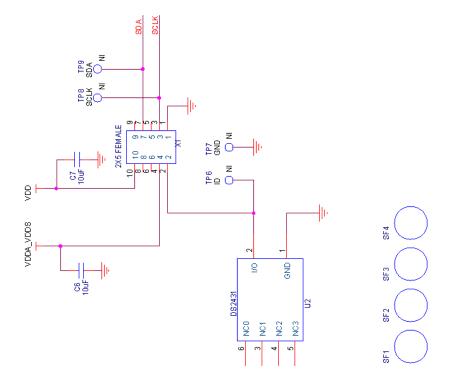
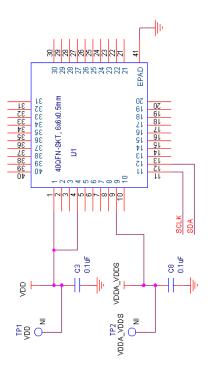


Figure 5.6. 44-Pin Socket Board Schematic









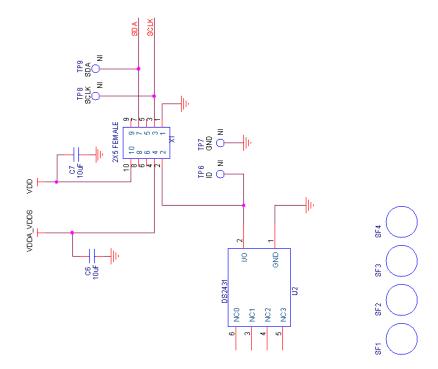
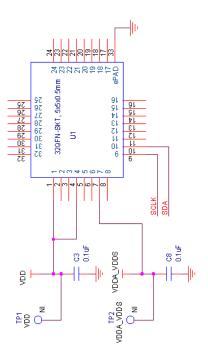
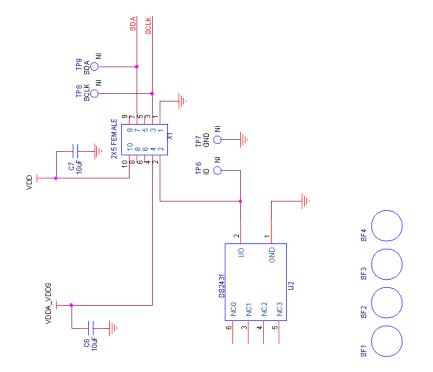


Figure 5.8. Si5332 40-Pin Socket Board Schematic







Skyworks Solutions, Inc. • Phone [781] 376-3000 • Fax [781] 376-3100 • sales@skyworksinc.com • www.skyworksinc.com Rev. 1.3 • Skyworks Proprietary Information • Products and Product Information are Subject to Change Without Notice • July 26, 2021

6. Bill of Materials

6.1 CBPROG-DONGLE Bill of Materials

		D (D 11	A. 1.		-			
NI	Quantity		Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
		C2 C3 C8 C11 C26	4. F		101	140%	¥70	0000	COCO20704CO 4051/	Marahal
	5	C28 C30 C31 C32	10+		16V	±10%	X7R	C0603	C0603X7R160-105K	Venkel
	3		0.01uF		16V	±20%	X7R	C0603	C0603X7R160-103M	Venkel
	1		33pF		25V	±10%	COG	C0402	C0402C0G250-330K	Venkel
		C4 C6 C12 C13								
		C14 C35 C36 C37								
	9		0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
	2		4.7uF		10V	±20%	X7R	C1206	C1206X7R100-475M	Venkel
	1		10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	2		LTL-14CHJ	20mA				LED-T1-KK	LTL-14CHJ	LITE-ON TECHNOLOGY CORP
	1		SP0503BAHT	300mW	20V		TVS	SOT143-AKKK SOT143	SP0503BAHTG	Littlefuse
		D6 D7 D8 D9 D10								
	7		GL05T-E3-08	5A	11V		Dual Common Anode		GL05T-E3-08	Vishay
	1		22 Ohm	6000mA			SMT	L0805	BLM21PG220SN1	MuRata
	1	J2	USB Type B				USB	CONN-USB-B	61729-0010BLF	FCI
	1	J3	ENCLOSURE					N/A	Emulator7045	Shanghai Zhongxingda Electronics
	4	R1 R11 R12 R65	1K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1001F	Venkel
		R13 R16 R26 R27								
		R40 R41 R42 R43								
	10	R59 R67	10K	1/16W		±1%	ThickFilm	R0402 R0402L	CR0402-16W-1002F	Venkel
	4	R14 R15 R24 R25	2.49K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-2491F	Venkel
	1	R28	1.0	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1R00F	Venkel
		R29 R30 R31 R32								
	8	R35 R37 R38 R68	0	1A			ThickFilm	R0402 R0402L	CR0402-16W-000	Venkel
	1		100	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1000F	Venkel
	1		4.42K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-4421F	Venkel
	1		9.53K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-9531F	Venkel
	1		5.9K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-5901F	Venkel
	2		1.0	3/4W		±1%	ThickFilm	R1210	CRCW12101R00FKEAHP	Vishay Dale
	2		8.06K	1/16W		±0.1%	±25PPM	R0402	TFCR0402-16W-E-8061B	Venkel
			17.4K					R0402	CR0402-16W-1742F	
	2			1/16W		±1%	ThickFilm			Venkel
	2		21.5K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-2152F	Venkel
	2	R50 R51	20K	1/10W		±1%	ThickFilm	R0603	CR0603-10W-2002F	Venkel
		R55 R56 R57 R58		1/16W		±1%	ThickFilm	R0402	CR0402-16W-4020F	Venkel
	1		499	1/16W		±1%	ThickFilm	R0402 R0402L	CR0402-16W-4990F	Venkel
	2		470	1/16W		±5%	ThickFilm	R0402	CR0402-16W-471J	Venkel
	2		10.0K	1/10W		±0.1%	±25PPM	R0603	ERA-3AEB103V	Panasonic
	2	SF1 SF2	BUMPER					RUBBER_FOOT_0.250"	SJ5382	3M
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
		2 SF3 SF4	SPACER					N/A	7363	Keystone Electronics
		L U1	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
			TPS76201	100mA			LDO	SOT5N2.8P0.95	TPS76201DBV	TI
		2 U13 U14	TPS79501	500mA			LDO	DFN8N3.0P0.65E2.4X1.65	TPS79501DRBT	TI
		U16	ADG719					SOT6N2.8P0.95	ADG719BRTZ	Analog Devices
		L U17	NLSV1T244		.9-4.5V		Buffer	UDFN6N1P0.4	NLSV1T244MUTBG	On Semi
			C8051F380		.5 4.50		MCU	QFP48N9X9P0.5	CF380P1104AGQ	SiLabs
		2 U3 U8	ADG712				NICO	TSSOP16N6.4P0.65	ADG712BRU	Analog Devices
			LTC4311		5.5V			SC70-6N2.1P0.65	LTC4311CSC6#TRMPBF	Linear Technology
					J.JV		12C	SO8N6.0P1.27		NXP
			PCA9517D				Shrouded	CONN2X5-RA-SBH11	PCA9517D SBH11-PBPC-D05-RA-BK	
Not Installed Co		L X1	2X5 Header				Shrouded	COMM2V3-RM-2RH11	SBHIIT-PDPC-DUS-KA-BK	Sullins Connector Solutions
Not Installed Co		Poteroneo	Value	Potin a	Valtera	Tolerance	Tumo	BCR Footprint	Manufacture-DN	Manufacturar
	Quantity	Reference	Value	Rating	Voltage		Type	PCB_Footprint	ManufacturerPN C0603X7R160-105K	Manufacturer
NI	-		1uF		16V	±10%	X7R	C0603		Venkel
			0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
N	-	L JP1	HEADER 4X1			-	Header	CONN-1X4	TSW-104-07-T-S	Samtec
		noonor	0	1A			ThickFilm	R0603	CR0603-16W-000	Venkel
NI		1 (D1 TD2 TD2 TD0								
NI		TP1 TP2 TP3 TP9								
NI		5 TP10	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI NI		5 TP10 L TP7	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI		5 TP10 L TP7			5V					

6.2 Si538x4x-64SKT-DK Socket Board BOM

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
	6	C1 C2 C3 C4 C5 C6	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
	2	C7 C8	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	1	R2	0	1A			ThickFilm	R0402 R0402L	CR0402-16W-000	Venkel
	4	SF1 SF2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	3M
	1	U1	64QFN-SKT, 9x9x0.5mm				QFN	QFN64N9X9P0.5-SKT-WELLS-CTI	790-42064-101G	Sensata
	1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Com	ponents									
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
NI	1	R1	1K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1001F	Venkel
NI	1	R3	20K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-2002F	Venkel
NI	3	TP1 TP2 TP3	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	2	TP4 TP5	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI	1	TP6	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

6.3 Si538x4x-56SKT-DK Socket Board Bill of Materials

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
		C1 C2 C5 C6 C7 C8								
	7	C9	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
	2	C3 C4	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
		R1 R2 R7 R8 R9								
	7	R10 R11	1K	1/16W		±1%	ThickFilm	R0603	CR0603-16W-1001F	Venkel
	2	R12 R13	0	1A			ThickFilm	R0603 R0603L	CR0603-16W-000	Venkel
	4	R3 R4 R5 R6	4.7K	1/10W		±1%	ThickFilm	R0603	CR0603-10W-4701F	Venkel
	4	SF1 SF2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	3M
	1	U1	56LGA-SKT, 8x8x0.5mm				LGA	QFN56N8X8P0.5-SKT-WELLS-CTI	790-42056-101G	Sensata
	1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Com	ponents									
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
NI	2	TP1 TP2	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	3	TP3 TP5 TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI	1	TP4	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

6.4 Si538x4x-44SKT-DK Socket Board Bill of Materials

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
	5	C1 C2 C3 C4 C5	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
	2	C6 C7	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	1	R2	0	1A			ThickFilm	R0402 R0402L	CR0402-16W-000	Venkel
	4	SF1 SF2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	3M
	1	U1	44QFN-SKT, 7x7x0.5mm				QFN	QFN44N7X7P0.5-SKT-WELLS-CTI	790-41044-101G	Sensata
	1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	X2	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Com	ponents									
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
NI	1	R1	1K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1001F	Venkel
NI	1	R3	182	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1820F	Venkel
NI	3	TP1 TP2 TP3	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	3	TP4 TP5 TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI	1	TP7	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

6.5 Si5332-48SKT-DK Socket Board Bill of Materials

Eval P	Board Name	SI5332-485KT	7							
	evision	1.0								
	(CV131011	1.0	1							
CreationDate	Proto Rev	Released								
5/18/2	2017 1.0	0 0								
NI	0	Defense	M-1	Detter	Nr-11	T -1	T		Adverse Construction DB1	
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
		2 C3 C8	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
		2 C6 C7	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
		4 SF1 SF2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	3M
		1 U1	48QFN-SKT, 6x6x0.4mm				QFN		790-62048-101G	Sensata
		1 U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
		1 X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed	Components									
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
NI		4 TP1 TP2 TP8 TP9	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI		1 TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI		1 TP7	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

6.6 Si5332-40SKT-DK Socket Board Bill of Materials

			-							
Eval Boar	d Name	SI5332-40SKT								
Revis	ion	1.0								
CreationDate	Proto Rev	Released								
5/18/2017	1.00	0								
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
141	Quantity	Reference	Value	Naung	Voltage	Torerance	Type	reb_rootprint	Mandiacturerri	Mandiacturei
	2	C3 C8	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
	2	C6 C7	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	4	SF1 SF2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	зм
	1	U1	40QFN-SKT, 6x6x0.5mm				QFN		790-42040-101G	Sensata
	1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Com	ponents									
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
NI	4	TP1 TP2 TP8 TP9	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	1	TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI	1	TP7	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

6.7 Si5332-32SKT-DK Socket Board Bill of Materials

Eval Bo	ard Name	SI5332-	32SKT								
Re	vision	1.0									
				-							
CreationDate	Proto Rev	Release	d								
5/18/20	17 1	.00	0								
NI	Quantity	Ref	erence	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
		2 C	3 C8	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
		2 C	6 C7	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
		4 SF1 SF	2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	ЗM
		1	U1	32QFN-SKT, 5x5x0.5mm				QFN		790-42032-101G	Sensata
		1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
		1	X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Co	omponents										
NI	Quantity	Ref	erence	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
NI		4 TP1 TP	2 TP8 TP9	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI		1	TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI		1	TP7	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

7. Appendix A. Troubleshooting

7.1 Why can't I communicate with the device on my hardware using the CBPro Dongle?

There are multiple windows in the CBPro software that use or provide communication to the device connected to the CBPro Dongle. The examples below show the windows and type of errors you may encounter. All of these situations can be resolved using the following steps.

General Steps to Resolve a Communication Issue (Non-Firmware based devices)

- 1. Verify which communication protocol your hardware is using SPI or I2C.
- 2. Verify the voltage level on the I2C_SEL control pin on the DUT. This level should be logic low (0 V) if your communication protocol is SPI. This level should be logic high (1.8 V or 3.3 V refer step 3 below) if your communication protocol is I2C.
- 3. Verify the value of the IO_VDD_SEL bit (Register 0x0943[0]) for the DUT. If IO_VDD_SEL is 0, the I/O Voltage setting should be 1.8V. If IO_VDD_SEL is 1, the I/O Voltage setting should 3.3V. If you do not know this value, you can try both voltages to determine which voltage level will work successfully.
- 4. If the communication protocol is I2C, verify the I2C address setting (Register 0x000B) for the device. You may also need to verify the voltage level on the A0/CSb and A1/SDO pins if they are not connected to the field programmer. The level on these pins set bit 1 and bit 0 in the I2C address. If these are connected to the CBPro Dongle, they are both driven low.

General Steps to Resolve a Communication Issue (Firmware based devices)

1. Verify the I2C address for the device.

2. Verify the voltage level on the A0/CSb and A1/SDO pins if they are not connected to the field programmer. The level on these pins set bit1 and bit 0 in the I2C address. If these are connected to the CBPro Dongle, they are both driven low.

Communication Error Using the Design Dashboard Window

If the design dashboard experiences an error communicating the device, the following error window will appear.

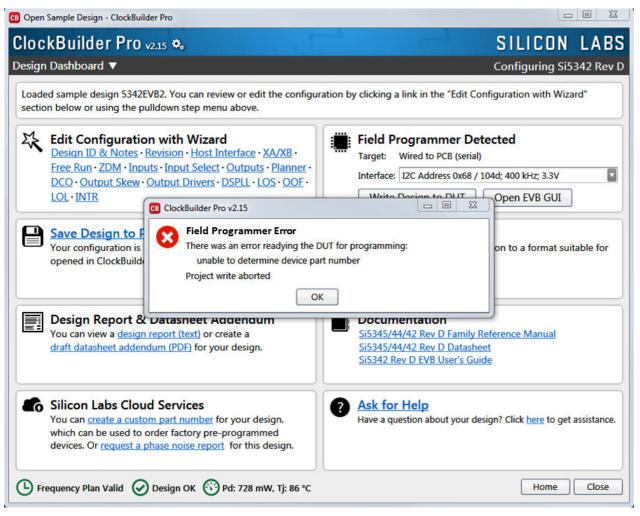


Figure 7.1. Communication Error Using Design Dashboard

This example window shows how to adjust the communication settings of the dashboard to resolve communication error.

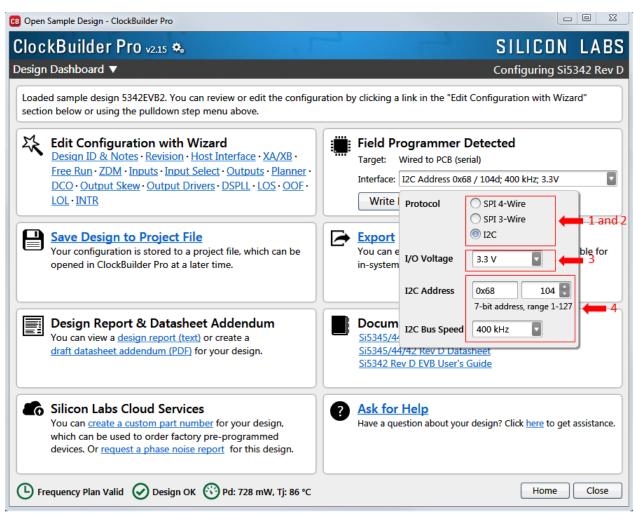


Figure 7.2. Design Dashboard Communication Error Solution

Communication Error Using the Burn NVM Window

The following window shows a communication error in the NVM Burn window. This error will appear after the Scan for Device button is pressed.

B NVM Program Tool - Clock	uilder Pro v2.15	
Field Programmer Mode: Target Device:	Wired to Board (No Socket Detected) Si538x/4x (not firmware based)	
Host Interface:	I2C Address 0x6C / 108d; 400 kHz; 3.3V	
Project File: Project File Created By: Project Part: Project Design ID:		Select Project Clear OPN Lookup
Project Design Check: Project File NVM Hash:	0	
Device Part Number: Device Design ID: Device NVM State:	Error communicating with device	Scan for Device Clear
# Valid Burns: # Burns with Error:	0 0 Program NVM	
	Plogram NVM	

Figure 7.3. Burn NVM Error Message

The following window shows how to adjust the communication settings of the dashboard to resolve communication error.

Field Programmer Mode:	Wired to Board (N	No Socket Detected)		
Target Device:	Si538x/4x (not fi	rmware based)	-	
Host Interface:	I2C Address 0x6	C / 108d; 400 kHz; 3.3V	-	
Project File: Project File Created By: Project Part:	Protocol	 SPI 4-Wire SPI 3-Wire I2C 	- 1 and 2	Select Project Clear OPN Lookup
Project Design ID: Project Design Check: Project File NVM Hash:	I/O Voltage	3.3 V	3	
Device Part Number: Device Design ID: Device NVM State:	I2C Address	7-bit address, range 1-127 400 kHz	4	Scan for Device Clear
# Valid Burns: # Burns with Error:	0		9	

Figure 7.4. Burn NVM Error Message Solution

Communication error using the EVB GUI window

The following window shows an example of the error produced when the EVB GUI experiences an I2C error.

B Field Programmer - Clock	Builder Pro	
File Help		
Info DUT Register Editor	Status Registers	Field Programmer
Field Programmer Identifi	cation:	Family: Si538x/4x (not firmware based)
Serial Number:	00-00-04-06-2C-CE	Target: Wired to PCB (serial)
DUT ID Registers:	Refresh ID Registers	Interface: I2C Address 0x6C / 108d; 400 kHz; 3.3V
DEVICE_PN_BASE		Part Number: -ERR-
DIE_REV		Design ID: -ERR-
DEVICE_REV:		Config Scan
TEMP_GRADE		
PKG_ID		Control Registers
BASELINE_ID		Soft Reset and Calibration
DEVICE_GRADE		SOFTRESET G
OPN_ID		DSPLLA_SOFTRESET
OPN_REVISION		DSPLLB SOFTRESET
DESIGN_ID		DSPLLC_SOFTRESET
Log		DSPLLD_SOFTRESET
Filtered Tauto Scro	oll: On 🧧 Insert Marker Clear Copy to Clipboard Pause	
Timestamp Source	Message	Hard Reset, Sync, & Power Down
	operation failed on MCU; error code 0xFA (general failure)	FW83_HARD_RST
09:43:40.988 EVB	Starting Read_DUT_Byte(address=0x0000)	RST_REG
09:43:40.989 EVB	error Read_DUT_Byte(address=0x0000) => I2C_Write(i2c_bus=1, i2c_slave_address=0x6C, data=0x0100): operation failed on MCU; error code 0xFA (general failure)	SYNC_REG
09:43:40.991 EVB	Starting Read_DUT_Byte(address=0x026B)	PDN: 0
09:43:40.992 EVB	error Read_DUT_Byte(address=0x026B) => I2C_Write(i2c_bus=1, i2c_slave_address=0x6C, data=0x0102):	
	operation failed on MCU; error code 0xFA (general failure)	Frequency Adjust +
VB Firmware 0.70 Device U	nknown (scan needed) Field Programmer: Wired to PCB (serial); Si538x/4x (not firmware based); I2C Address 0x6C / 108d; 400 kH	łz; 3.3V

Figure 7.5. EVB GUI I2C Error

The following window shows an example of the error produced when the EVB GUI experiences an SPI error.

le H	lelp		
nfo	DUT Register Editor	Status Registers	+ Field Programmer
ield	Programmer Identific	ation:	Family: Si538x/4x (not firmware based)
Se	erial Number:	00-00-04-06-2C-CE	Target: Wired to PCB (serial
DUT I	ID Registers:	Refresh ID Registers	Interface: SPI 4-Wire; 12 MHz 3.3V
D	EVICE_PN_BASE		Part Number: -ERR-
D	IE_REV		Design ID: ???????
D	EVICE_REV:		Config Scan
Т	EMP_GRADE		
P	KG_ID		Control Registers
B	ASELINE_ID		Soft Reset and Calibration
D	EVICE_GRADE		SOFTRESET G
0	PN_ID		DSPLLA_SOFTRESET
0	PN_REVISION		
D	ESIGN_ID		DSPLLC SOFTRESET
og Filter	ed 🔽 Auto Scro	I: On 📱 Insert Marker Clear Copy to Clipboard Pause	DSPLLD_SOFTRESET
			Hard Reset, Sync, &
	stamp Source	message finished Read DUT Byte(address=0x026F) => 0xFF	Power Down
9:44	:32.826 EVB	Starting Read_DUT_Byte(address=0x0270)	FW83_HARD_RST
)9:44	:32.829 EVB	finished Read_DUT_Byte(address=0x0270) => 0xFF	RST_REG
)9:44	:32.829 EVB	Starting Read_DUT_Byte(address=0x0271)	SYNC_REG
)9:44	:32.833 EVB	finished Read_DUT_Byte(address=0x0271) => 0xFF	PDN: 0
09:44	:32.833 EVB	Starting Read_DUT_Byte(address=0x0272)	
09.44	:32.837 EVB	finished Read_DUT_Byte(address=0x0272) => 0xFF	Frequency Adjust

Figure 7.6. EVB GUI SPI Error

The following window shows how to change the communication settings using the EVB GUI window.

Marcel Apogrammer Identification: Serial Number: 00:00:00:00:00:00:00:00:00:00:00:00:00:	e Help					
Nording intermate intermate intermate interval frequency Adjust Strill Number Ox0-04-05-2C-CE JT D Registers Referen ID Registers DEVCEL_INL_BASE ··· PRG_DD ··· PGG_DD ··· OPVLCE_GRUSE ··· ···	nfo DUT Register Edito	Status Registers	 Field Pr 	ogrammer		
Serial Number 00:00:00:00:2C.C Taget: Wred to DCB (price) JTD Register: Control Register: Used control Register:	ield Programmer Identi	cation:	Family:			
TI D legister: Refreh D Register: Interface: DivCC, PN, BASE Interface:: DivCC, PN, BASE Interface::: DivCC, PN, BASE Interface::: DivCC, PN, BASE Interface::: DivCC, PN, BASE Interface::: DivCC, PN, DASE PN, Eac::: DivC	Serial Number:	00-00-04-06-2C-CE	Target:			
DVXCL_PN_BAGE DER_RAV DVXCL_PN_EXC DVXC	NIT ID Registers:	Refeach ID Registers		e: I2C Addres	s 0x68 /	
DE_RAY DEVACE_RAY: DEVAC				104d; 400	kHz; 1.8V	
DVICE, RAY: TEMP_GRADE PRG_D BASELINE_D DVICE, RAYE OPN_D OPN_D OPN_D OPN_REVISION DESIGN_D TOOL_VERSION TOOL_VE						
TEMP_GRADE PKG_JD BASELINE_JD DEVICE_GRADE OFN_JD DEVICE_GRADE OFN_JD DEVICE_GRADE DEVICE_GRADE OFN_JD DEVICE_GRADE TOOL_VERSION TOOL_VERSION TEVERST T	DIE_REV		Design	ID:		
PKG_ID BASELINE_ID DEVICE_GRADE OPN_ID OPN_UR_SUSION TOOL_VERSION	DEVICE_REV:			Config Sc	an	
PAG_U	TEMP_GRADE		-			
DEVICE_GRADE OPN_JD OPN_JD OPN_REVISION DESIGN_JD TOOL_VERSION TOOL_VERS	PKG_ID		Contro	Device Family	51538x/4x (not	firmware based
DEVICE, GRADE ORV, JD ORV, JD ORV, RXISSION DESIGN JD TOOL, VERSION TOOL, VERSION PW83_HARD_RST RXIS Starting Read_DUT_Byfe(address=000270) 1354.107 VM Starting Read_DUT_Byfe(address=000270) 1354.107 VM Starting Read_DUT_Byfe(address=000271) 1354.107 <t< td=""><td>BASELINE_ID</td><td></td><td></td><td>Protocol</td><td>O SPI 4-Wire</td><td></td></t<>	BASELINE_ID			Protocol	O SPI 4-Wire	
OPN_D OPN_RX/VSION DESIGN_D TOOL_VERSION	DEVICE_GRADE					🛑 1 an
OPN_REVISION DESIGN_ID TOOL_VERSION TOOL_VERSION Result of the state of the stat	OPN ID				I2C	
DESIGN_ID TOOL_VERSION TOOL_VER				I/O Voltage	1.8V	3 🛻 3
TOOL_VERSION				, o ronage	1.0 1	<u> </u>
revel Auto Scrolt On Insett Marker Clear Copy to Clipboard Pause PW33_HARD_R5T RST_REG SYNC_REG PDN: 0 PDN: 0 0 1355A167 EV8 Innished Read_DUT_Byte(address=0x0270) => 0xFF Finshed Read_DUT_Byte(address=0x0270) => 0xFF 1355A107 EV8 Starting Read_DUT_Byte(address=0x0270) => 0xFF Finshed Read_DUT_Byte(address=0x0270) => 0xFF 1355A1207 EV8 Innished Read_DUT_Byte(address=0x0270) => 0xFF Finshed Read_DUT_Byte(address=0x0270) => 0xFF 1355A1207 EV8 finished Read_DUT_Byte(address=0x0270) => 0xFF Finished Read_DUT_Byte(address=0x0271) => 0xFF 1355A1207 EV8 finished Read_DUT_Byte(address=0x0271) => 0xFF Finished Read_DUT_Byte(address=0x0271) => 0xFF				I2C Address	0x68	104
Insert Marker Clear Copy to Clipboard Pause PDN: Insert Marker Clear Copy to Clipboard Pause Frequency Adjust F_INC_REG PDN: Insert Marker Clear Copy to Clipboard Pause Frequency Adjust F_INC_REG PDN: Insisted Read_DUT_Byte(address=0x0270) Insisted Read_DUT_Byte(address=0x0270) => 0xfF Insisted Read_DUT_Byte(address=0x0270) => 0xfF Insisted Read_DUT_Byte(address=0x0271) => 0xfF Insisted Read_DUT_Byte(address=0x0271) => 0xfF	TOOL_VERSION				7-bit address, rar	
Instend I and the set of the set				I2C Bus Speed	400 kHz	. +
ered Auto Scroll: On Insert Marker Clear Copy to Clipboard Pause PDN: 0 13:54.107 EV8 Innshed Read_DUT_Byte(address=0x0270) Frequency Adjust 13:54.107 EV8 Starting Read_DUT_Byte(address=0x0271) Frequency Adjust						
ered I Auto Scroll: On I Insert Marker Clear Copy to Clipboard Pause Plause Pla				EW83 HARD F	T	
ered Auto Scroll: On Insert Marker Clear Copy to Clipboard Pause PDN: 0 restamp Source Message FigNC_REG PDN: 0 13:54.107 EV8 Inished Read_DUT_Byte(address=0x0270) FigNC_REG 13:54.107 EV8 Starting Read_DUT_Byte(address=0x0271) FigNC_REG 13:54.107 EV8 Starting Read_DUT_Byte(address=0x0271) FigNC_REG 13:54.107 EV8 Starting Read_DUT_Byte(address=0x0271) FigNC_REG 13:54.107 EV8 finished Read_DUT_Byte(address=0x0271) FigNC_REG						
ered Auto Scroll: On Insert Marker Clear Copy to Clipboard Pause Frequency Adjust sextamp Source Message FUNC_REG 13:54.167 EV8 Innished Read_DUT_Byte(address=0x0270) Finished Read_DUT_Byte(address=0x0270) 13:54.170 EV8 Starting Read_DUT_Byte(address=0x0270) => 0xFF Finished Read_DUT_Byte(address=0x0271) => 0xFF 13:54.175 EV8 finished Read_DUT_Byte(address=0x0271) => 0xFF Finished Read_DUT_Byte(address=0x0271) => 0xFF						
ered in Instet Marker Clear Copy to Clipboard Pause Frequency Adjust FUNC_REG Frequency Adjust FUNC_REG Frequency Adjust FUNC_REG FilmC_REG FilmC_						
ered Insert Marker Clear Copy to Clipboard Pause F_INC_REG estamp Source Message Instrict Read_DUT_Byte(address=0x02/bF) => 0x7+ F 1354.167 EVB Inished Read_DUT_Byte(address=0x02/0) Inished Read_DUT_Byte(address=0x02/0) Inished Read_DUT_Byte(address=0x02/0) 1354.170 EVB Inished Read_DUT_Byte(address=0x02/0) => 0xFF Inished Read_DUT_Byte(address=0x02/1) => 0xFF 1354.175 EVB Inished Read_DUT_Byte(address=0x02/1) => 0xFF Inished Read_DUT_Byte(address=0x02/1) => 0xFF				PDN: 0		
extemp Source Message F_INC_REG 135×167 EV8 Inished Read_DUT_Byte(address=0x0270) => 0xFF Inished Read_DUT_Byte(address=0x0270) => 0xFF 135×175 EV8 Starting Read_DUT_Byte(address=0x0271) => 0xFF Inished Read_DUT_Byte(address=0x0271) => 0xFF				Frequency Adi	uet	
Source Message 13:54.167 EVB finished Read_DUT_Byte(address=0x02/0) 13:54.170 EVB finished Read_DUT_Byte(address=0x02/0) 13:54.170 EVB finished Read_DUT_Byte(address=0x02/0) => 0xFF 13:54.175 EVB finished Read_DUT_Byte(address=0x02/1) 13:54.175 EVB finished Read_DUT_Byte(address=0x02/1)	ered 📱 Auto Scr	II: On 📱 Insert Marker Clear Copy to Clipboard Pause				
13:54.16/ EVB finished Read_DUT_Byte(address=Ux026F) => 0xF+ 13:54.167 EVB Starting Read_DUT_Byte(address=0x0270) 13:54.170 EVB finished Read_DUT_Byte(address=0x0270) 13:54.170 EVB finished Read_DUT_Byte(address=0x0270) => 0xFF 13:54.175 EVB finished Read_DUT_Byte(address=0x0271) 13:54.175 EVB finished Read_DUT_Byte(address=0x0271) => 0xFF	nestamp Source	Message				
13:54:170 EVB finished Read_DUT_Byte(address=0x0270) => 0xFF 13:54:170 EVB Starting Read_DUT_Byte(address=0x0271) 13:54:175 EVB finished Read_DUT_Byte(address=0x0271) => 0xFF	13:54.167 EVB	Inished Kead_DUI_Byte(address=0x026F) => 0xFF	A			
13:54:170 EVB Starting Read_DUT_Byte(address=0x0271) 13:54:175 EVB finished Read_DUT_Byte(address=0x0271) => 0xFF	13:54.167 EVB	Starting Read_DUT_Byte(address=0x0270)				
13:54:175 EVB finished Read_DUT_Byte(address=0x0271) => 0xFF	L3:54.170 EVB	finished Read_DUT_Byte(address=0x0270) => 0xFF				
		Starting Read_DUT_Byte(address=0x0271)				
12:54.175 EV/P Starting Road DLIT Pute(address=0x0272)	L3:54.175 EVB	finished Read_DUT_Byte(address=0x0271) => 0xFF				
Conserts Examined (read "not "note formerss-not/515)	L3:54.175 EVB	Starting Read_DUT_Byte(address=0x0272)				
13:54.181 EVB finished Read_DUT_Byte(address=0x0272) => 0xFF						

Figure 7.7. EVB GUI Solution

7.2 Why do I have a communication error when I write my new project to the Si534x-8x-7x-9x device?

New Plan Changes the IO_VDD_SEL Bit (Register 0x0943[0]) Value

In order for the CBPro Dongle to communicate with the device correctly, the dongle's IO voltage needs to match the IO_VDD_SEL bit in the device. If the plan changes this bit during the writing process, communication will fail. To determine if the new plan is changing this bit, perform the following steps:

- Read the current value in the device by using the DUT Register Editor tab in the EVB GUI window.
- Determine if the new plan changes the value. This can be done by looking at the Host Interface tab in the Design Dashboard of the new project.
 - If VDD (Core) radio button selected and 0x943 = 0, no change from new plan,

Else VDD (Core) radio button selected and 0x943 = 1, new plan is changing IO_VDD_SEL refer to 7.3 How do I write a project file to the device that changes the I/O Power Supply setting in Si534x-8x-7x-9x devices (IO_VDD_SEL bit)?

• If VDDA (3.3 V) radio button selected and 0x943 = 1, no change from new plan,

Else VDDA (3.3 V) radio button selected and 0x943 = 0, new plan is changing IO_VDD_SEL refer to 7.3 How do I write a project file to the device that changes the I/O Power Supply setting in Si534x-8x-7x-9x devices (IO_VDD_SEL bit)?

The following window shows how to read the IO_VDD_SEL bit from the device.

r Hdp	
Is DUT Register Editor Status Registers	 Field Programmer
	Fanity SG336vik over Tanget Societ ("Weine Same) Tanget Societ ("Weine Societ Tween" (Weine Same) Bitterize: 974 4 Weiz 124 M Fanither SG344 A 4 40 Part Namber SG344 A 40 Deight D Satterize Config. Scan
Brwy: 7 6 5 4 3 2 1 0	Control Registers Soft Reset and Calibration SOFT_RST
	Hard Innet Types of The Sector
ener 🖉 Ans Sould In 🖉 (SeetMader) Char) Copy to Sylward (Faune	Power Down HARD,KST SYNC PORE 0 Frequency Adjust FPUC
tered Auto Scroit: Cn 📮 Ersert Marker Clear Copy to Clipboard. Pause	Power Down HABD_KST SYNC PORe 0 Frequency Adjust PDC
ered & Auto Scolit On & Penet Marker Clear Copy to Optional Pence entropy Scorer Manager Market National Restance (Market National Score) (1997) (1	Power Down H4800,KST SYNC POrts 0 Frequency Adjust PDC
Journal Manage 200427 TW Masket Matg.201_(1)(6)(5)(6)(6)(6)(1)(1)(6)(5)(6)(6)(7)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)	Power Down H4800,KST SYNC POrts 0 Frequency Adjust PDC
Source Ministry 2000LV7 Ministry March 1000 (117) 1000 2000LV7 Ministry March 1000 (117) 1000 2000LV7 Starting Anal, OUT, Syntaddews-tob0007 2000LV7 Starting Anal, OUT, Syntaddews-tob0007 2000LV7 Starting Anal, OUT, Syntaddews-tob0007	Poer Down HARD KST SYNC POIN: 0 Frequency Adjust FDK
Science Mexicoge 2002/17 With West Read (2017; My (W) (2009) With Trial (2000) 2002/17 With West Read (2017; My (W) (2009) 2002/17 With West Read (2017; My (W) (2019) 2002/17 With West Read (2019; My (W) (2019) 2002/17 With West Read (2019) 2002/17 </td <td>Poer Down HARD KST SYNC POIN: 0 Frequency Adjust FDK</td>	Poer Down HARD KST SYNC POIN: 0 Frequency Adjust FDK
Mail: Scott: Termet Maker Clavr Copy to Optional Period Mail: Marcal Monage Monage Monage Monage 2024/271 VM Monage Malage Copy to Optional > Monage	Power Down H4800,KST SYNC POrts 0 Frequency Adjust PDC

Figure 7.8. Read IO_VDD_SEL Bit from Device

The following window shows how to determine the value of the IO_VDD_SEL bit that will be written to the device from the project file.

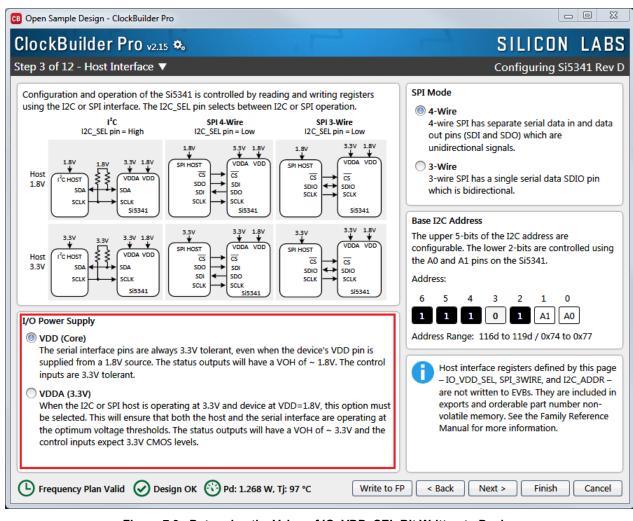


Figure 7.9. Determine the Value of IO_VDD_SEL Bit Written to Device

7.3 How do I write a project file to the device that changes the I/O Power Supply setting in Si534x-8x-7x-9x devices (IO_VDD_SEL bit)?

General Steps to Change I/O Power Supply Setting with a Project File

In order for the field programmer to communicate with the device correctly, the field programmer's IO voltage needs to match the IO_VDD_SEL bit in the device and use the correct serial communication protocol to match the I2C_SEL pin on the device. This is not automatically detected by the GUI or the CLI command.

If the new project changes the IO_VDD_SEL bit, the following summarized steps need to be performed. The flow chart and figures that follow provide the details for each of these steps. There are detailed steps using CBPro Graphical User Interface and detailed steps using the CBPro Command Line interface.

1. Establish communication with the device to be programmed and determine the current value of the IO_VDD_SEL (0x0943[0]) bit.

2. The current value of the IO_VDD_SEL bit matches the value of the new plan to be written to the device?

- · Yes Proceed to step 3.
- No Change the IO_VDD_SEL bit to match the value in the new plan. Re-establish communication with the device after changing the IO_VDD_SEL value (change the field programmer I/O Voltage to match new value for IO_VDD_SEL).

3. Write the new plan to the device.

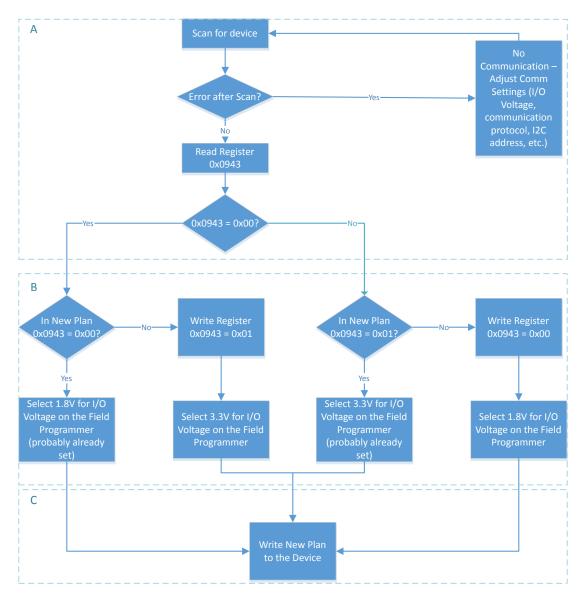


Figure 7.10. General Steps to Change I/O Power Supply Setting with a Project File

UG286: ClockBuilderPro[™] Field Programmer Kit • Appendix A. Troubleshooting

Steps using CBPro Graphical User Interface

1. Select the 'EVB GUI' button on the home screen as shown to attempt communication with the device.

ClockBuilder Pro Wizard - Silicon Labs	
SILICON LABS ClockBuilder Pro	
Work With a Design	Quick Links
Create New Design	Clock Generators & Jitter Attenuators Knowledge Base Custom Part Number Lookup ClockBuilder Go IOS App
ex Open Sample Design	Applications Documentation
Field Programmer Detected	10/40/100G Line Card White Paper Clock Generators for Cloud Data Centers White Paper Optimizing Si534x Jitter Performance App Note SyncE and IEEE 1588 App Note
and the second sec	ClockBuilder Pro Documentation
17 12 Mg	CBPro Overview CBPro Tools & Support for In-System Programming Includes walthroughs of frequency-on-the-fly, full configuration, and partial configuration programming scenarios. CLUSers's Guide Release Notes • Knowledge Base
a,	Version 2:15 Built on 5/10/2017

Figure 7.11. EVB GUI Button

- a. Select the 'DUT Register Editor' tab.
- b. Determine the correct device communication protocol and setup CBPro accordingly as shown. For an In-socket device, click the Socket Power slider to power up the device. For In-system devices, click the Device Family pulldown and select the appropriate device family.
- c. Click the Scan button to verify communication with the device.
- d. If communication is successful, the device part number and design ID will be updated. If communication is not successful, the part number field will display -ERR- and the DUT register tab will be disabled.

Configuring communication settings:

Field Progra Family:	Si538x/4x (not		ogrammer		
Target: Socket Pow Interface:	firmware based) Socket, QFN64 er: Off 2 SPI 4-Wire; 12 MHz; 1.8V	Family: Target: Interfac		ased) CB (serial)	
Part Numbe Design ID:	er: onfig Scan 4 3	Control	ID: Config Sca	an 4 3 Si538x/4x (not firmware based)	-
Control Pr	rotocol SPI 4-Wire SPI 3-Wire I2C		Protocol	 SPI 4-Wire SPI 3-Wire 12C 	
1/0	O Voltage 1.8 V	3	I/O Voltage	3.3 V	
SF	PI Bus Speed 12 MHz		SPI Bus Speed	12 MHz	

Figure 7.12. Configuring Communication Settings

Examples of a Communication failure for I2C and SPI:

	mmer - Clocki								
le Help									
Info DUT R	egister Editor	Status Registers					-	Field Progra	mmer
Field Program	nmer Identifi	ication:					â	Family:	Si538x/4x (not firmware based)
Serial Nu	mber:	00-00-04-06-2C	-CE					Target:	Wired to PCB (seri
OUT ID Regis	tors:	Refresh ID Re						Interface:	I2C Address 0x6C
DEVICE_P			gisters						108d; 400 kHz; 3.3
	IN_BASE							Part Numbe	
DIE_REV								Design ID:	-ERR-
DEVICE_R								Co	onfig Scan
TEMP_GR	ADE							Control Reg	isters
PKG_ID									oft Reset and
BASELINE									Calibration
DEVICE_G	GRADE							S	OFTRESET_G
OPN_ID								DSPI	LLA_SOFTRESET
OPN_REV							U	DSPI	LLB_SOFTRESET
DESIGN_I	D						-	DSPI	LLC_SOFTRESET
og								DSPI	LLD_SOFTRESET
iltered	Auto Scro	oll: On 📱 🛛 Insert I	Marker Clear	Copy to Clipb	oard Pause			11	Deset Sume Pr
imestamp	Source	Message							l Reset, Sync, & Power Down
		operation failed on	MCU; error code 0xF	A (general failure)				FW	83_HARD_RST
9:43:40.988		Starting Read_DUT	Byte(address=0x000	0)					RST_REG
9:43:40.989	EVB		te(address=0x0000) = MCU; error code 0xF		bus=1, i2c_slave_address	=0x6C, data=0x0100):			SYNC_REG
9:43:40.991	EVB		Byte(address=0x026					PI	DN: 0
9:43:40.992	D/D								
'B Firmware 0. Field Prograr	70 Device U	operation failed on	MCU; error code 0xF.	=> I2C_Write(i2c_t A (general failure)		= 0x6C, data=0x0102): ased); I2C Address 0x6C / 10	₹		quency Adjust
Field Program	70 Device U	operation failed on	MCU; error code 0xF.	=> I2C_Write(i2c_t A (general failure)			¥		
Field Prograr e Help	70 Device U mmer - Clocki	operation failed on	MCU; error code 0xF.	=> I2C_Write(i2c_t A (general failure)			•	3.3V Field Progra	mmer
Field Program e Help nfo DUT Re	70 Device U mmer - Clockl	operation failed on nknown (scan needed) Builder Pro Status Registers	MCU; error code 0xF.	=> I2C_Write(i2c_t A (general failure)			•	3.3V	Immer Si538x/4x (not
Field Program e Help nfo DUT Re	70 Device U mmer - Clockl egister Editor nmer Identifi	operation failed on nknown (scan needed) Builder Pro Status Registers	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			Ť	3.3V Field Progra	mmer
Field Program e Help nfo DUT R ield Program Serial Nut	70 Device U mmer - Clockl egister Editor nmer Identifi mber:	operation failed on nknown (scan needed) Builder Pro Status Registers ication:	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	3.3V Field Progra Family:	mmer Si538x/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH
Field Program e Help nfo DUT R ield Program Serial Nut	70 Device U mmer - Clockl egister Editor nmer Identifi mber: ters:	operation failed on nknown (scan needed) Builder Pro Status Registers ication: 00-00-04-06-2C	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	3.3V Field Progra Family: Target: Interface:	mmer SIS38X/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V
Field Program e Help nfo DUT R field Program Serial Nur DUT ID Regis	70 Device U mmer - Clockl egister Editor nmer Identifi mber: ters:	operation failed on nknown (scan needed) Builder Pro Status Registers ication: 00-00-04-06-2C	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	3.3V Field Progra Family: Target: Interface: Part Numbe	mmer SIS38X/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V
Field Program e Help nfo DUT Ru Field Program Serial Nur DUT ID Regis DEVICE_P	70 Device U mmer - Clockl egister Editor nmer Identifi mber: tters: N_BASE	operation failed on nknown (scan needed) Builder Pro Status Registers ication: 00-00-04-06-2C	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	5.3V Field Progra Family: Target: Interface: Part Numbe Design ID:	mmer SIS38X/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V rrERR- ???????
Field Program e Help nfo DUT Ru ield Program Serial Nur DUT ID Regis DEVICE_P DIE_REV	70 Device U mmer - Clockd egister Editor mmer Identifi mber: ters: N_BASE	operation failed on nknown (scan needed) Builder Pro Status Registers ication: 00-00-04-06-2C Refresh ID Re	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	5.3V Field Progra Family: Target: Interface: Part Numbe Design ID:	mmer SIS38x/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V rrERR-
Field Program e Help nfo DUT R iield Program Serial Nur DUT ID Regis DEVICE_P DIE_REV DEVICE_R	70 Device U mmer - Clockd egister Editor mmer Identifi mber: ters: N_BASE	operation failed on nknown (scan needed) Builder Pro Status Registers ication: 00-00-04-06-2C Refresh ID Re	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	5.3V Field Progra Family: Target: Interface: Part Numbe Design ID:	mmer SiS38x/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V r: ERR- TITTTT Scan
Field Program e Help nfo DUT R ield Program Serial Nuu DUT ID Regis DEVICE_P DIE_REV DEVICE_R TEMP_GR	70 Device U mmer - Clockl egister Editor mmer Identifi mber: ters: N_BASE EV: ADE	operation failed on nknown (scan needed) Builder Pro Status Registers iccation: C0-00-04-06-2C Refresh ID Re 	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	5.3V Field Progra Family: Target: Interface: Part Numbe Design ID: CC Control Reg	mmer Si538x/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V rs: -ERR- ???????? onfig Scan isters oft Reset and
Field Program e Help nfo DUT R/ Serial Nur DUT ID Regis DEVICE_P DIE_REV DEVICE_R TEMP_GR PKG_JD BASELINE	70 Device U mmer - Clockl egister Editor mmer Identifi mber: ters: N_BASE EV: ADE	operation failed on nknown (scan needed) Builder Pro Status Registers iccation: C0-00-04-06-2C Refresh ID Re 	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	5.3V Field Progra Family: Target: Interface: Part Numbe Design ID: Co Control Reg	sisters Sike and Sister and Sister and Sister and Calibration
Field Program e Help nfo DUT Ri Serial Nur DUT ID Regiss DEVICE_P DIE_REV DEVICE_R TEMP_GR PKG_ID	70 Device U mmer - Clockl egister Editor mmer Identifi mber: ters: N_BASE EV: ADE	operation failed on nknown (scan needed) Builder Pro Status Registers Cation: 00-00-04-06-2C Refresh ID Re 	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	3.3V Field Progra Family: Target: Interface: Part Numbe Design ID: Co Control Reg	mmer SiS38X/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V SPI 4-WIRE; 1
Field Program e Help finition DUT R Serial Nur Serial Nur DUT ID Regiss DEVICE, P DEVICE, R TEMP_GR PKG_ID BASELINE DEVICE, G OPN_ID	70 Device U mmer - Clockl egister Editor mber: tters: N_BASE EV: ADE ED iRADE	operation failed on nknown (scan needed) Builder Pro Status Registers Cation: 00-00-04-06-2C Refresh ID Re 	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	Field Progra Family: Target: Interface: Part Numbe Design ID: Control Reg	mmer SiSBX/4x (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V sr: ERR- 12171717 stors offig Scan isters off Reset and Calibration OFTRESET_G LLA_SOFTRESET
Field Program e Help nfo DUT R Serial Nur DUT ID Regis DEVICE, P DIE_REV DEVICE, R TEMP_GR PKG_ID BASELINE DEVICE, G	70 Device U mmer - Clocki egister Editor mmer Identifi mber: ters: N_BASE EV: ADE GID iRADE	operation failed on nknown (scan needed) Builder Pro Status Registers ication: (00-00-04-06-2C Refresh ID Re 	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	Field Program Family: Target: Interface: Part Numbe Design ID: Control Reg Control Reg DSPI	mmer SIS38x/dx (not firmware based) wired to PCB (seri SIS4 - Wire; 12 MH 3.3V rr - ERR- TZTTTTT TZTTTTTT onfig Scan isters off Reset and Calibration OFTRESET_G LIA_SOFTRESET LIB_SOFTRESET
Field Program fro DUT Ro Serial Dur Ro Serial Nur DEVICE_P DEVICE_R TEMP_GR PKG_ID BASELINE DEVICE_C OPN_ID OPN_REV DESIGN_I	70 Device U mmer - Clocki egister Editor mmer Identifi mber: ters: N_BASE EV: ADE GID iRADE	operation failed on nknown (scan needed) Builder Pro Status Registers ication: 00-00-04-06-2C Refresh ID Re 	MCU; error code 0xF	=> I2C_Write(i2c_t A (general failure)			•	Field Progra Family: Target: Interface: Part Numbe Design ID: Control Reg Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc	mmer SIS38x/dx (not firmware based) Wired to PCB (sen SPI 4-Wire; 12 MH 3.3V sr: -ERR- 77777777 77777777 77777777 T77777777
Field Program e Help field Program Serial Num DUT ID Regis DEVICE,P DIE,REV DEVICE,R PKG_ID BASELINE DEVICE,G OPN_ID OPN_REV DESIGN_I DESIGN_I	mmer - Clockl egister Editor mmer Identifi mber: ters: N_BASE EV: ADE EID ISION D	operation failed on nknown (scan needed) Builder Pro Status Registers Co-00-04-06-2C Refresh ID Re Co- Co- Co- Co- Co- Co- Co- Co- Co- Co-	MCU; error code 0x7 Field Programmer: W CE gisters	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Si538x/4x (not firmware bx		•	Field Progra Family: Target: Interface: Part Numbe Design ID: Control Reg Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc	mmer SIS38x/dx (not firmware based) wired to PCB (seri SIS4 - Wire; 12 MH 3.3V rr - ERR- TZTTTTT TZTTTTTT onfig Scan isters off Reset and Calibration OFTRESET_G LIA_SOFTRESET LIB_SOFTRESET
Field Program Field Program Serial Nur DEVICE,P DEVICE,P DEVICE,R PKG_ID BASELINE DEVICE,G OPN_ID OPN_ID OPN_REV DESIGN_I DESIGN_I	mmer - Clockl egister Editor mmer Identifi mber: ters: NL_BASE EV: ADE ELD SIGN D MAUTO Sere	operation failed on nknown (scan needed) Builder Pro Status Registers Co-00-04-06-2C Refresh ID Re Co- Co- Co- Co- Co- Co- Co- Co- Co- Co-	MCU; error code 0x7 Field Programmer: W CE gisters	=> I2C_Write(i2c_t A (general failure)	Si538x/4x (not firmware bx		•	Field Progra Family: Target: Interface: Part Numbe Design ID: Control Reg Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc Sc	mmer SIS38x/dx (not firmware based) Wired to PCB (sen SPI 4-Wire; 12 MH 3.3V sr: -ERR- 77777777 77777777 77777777 T77777777
Field Program Field Program Serial Nur DUT ID Regis DEVICE_P DEVICE_R DEVICE_R DEVICE_R DEVICE_G DEVICE_G DEVICE_G OPN_ID OPN_REV DESIGN_I OPN_REV DESIGN_I OPN_REV DESIGN_I	70 Device U mmer - Clockl egister Editor mmer Identifi mber: ters: N_BASE EV: ADE SID ISION D Auto Scroc Source	operation failed on nknown (scan needed) Builder Pro Status Registers Ication: 00-00-04-06-2C Refresh ID Re -	MCU; error code 0x7 Field Programmer: W CE gisters gisters Aarker Clear	DC WittingCo Line A (general failure) irred to PCB (serial);	Si538x/4x (not firmware bx		•	Field Program Field Program Field Program Farnity: Target: Interface: Part Numer Design ID: C Control Reg SSP DSPP DSPP DSPP DSPP DSPP DSPP DSPP	SIS38X/AX (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V ir: ERR- 2777777 isters onfig Scan isters fift Reset and calibration orTRESET_G CASOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET
Field Program Ie Help Info DUT R Serial DUT R Serial Nuu DUT ID Regis DEVICE,P DEVICE,P DEVICE,R TEMP_GR PKG_ID BASELINE DEVICE,C OPN_ID OPN_REV DESIGN_I OPN_REV DESIGN_I OPN_REV DESIGN_I	70 Device U mmer - Clockd egister Editor mmer Identifi mber: tters: N_BASE EV: ADE ID Auto Scro Source EV8	operation failed on nknown (scan needed) Builder Pro Status Registers ication: (00-00-04-06-2C Refresh ID Re -	MCU; error code 0x7 Field Programmer: W CE gisters gisters Aarker Clear Byte(address=0x026)	e> DC_Wink@C_UA A (general failuro) irred to PCB (serial); (Copy to Clipb b) => 0x)+	Si538x/4x (not firmware bx		•	Field Program Field Program Field Program Farnity: Target: Interface: Part Numer Design ID: C Control Reg SSP DSPP DSPP DSPP DSPP DSPP DSPP DSPP	sisal content of the second of
Field Program Field Program Serial Nur DUT ID Regis DEVICE_P DEVICE_R DEVICE_R TEMP_GR PKG_ID BASELINE DEVICE_G OPN_IE OPN_RE OPN_RE DESIGN_I OPN_RE OPS/44:32.826 093/44:32.826	70 Device U mmer - Clockl egister Editor mmer Identifimber: ters: N_BASE EV: ADE :EU: ADE :EID :RADE :ID :RADE :SOURCE EV8 EV8	loperation failed on nknown (scan needed) Builder Pro Status Registers ication: 00-00-04-06-2C Refresh ID Re -	MCU; error code 0x7 Field Programmer: W CE gisters gisters Aarker Clear	DC_White(6C_L) A (general failure) ired to PC8 (serial); copy to Clipht (Copy to Clipht +) => 08++00	Si538x/4x (not firmware bx		•	Field Program Field Program Field Program Farnity: Target: Interface: Part Numer Design ID: C Control Reg SSP DSPP DSPP DSPP DSPP DSPP DSPP DSPP	SIS38X/AX (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V ir: ERR- 2777777 isters onfig Scan isters fift Reset and calibration orTRESET_G CASOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET ILL_SOFTRESET
Field Program Field Program Serial Nut DUT ID Regis DEVICE,P DEVICE,P DEVICE,R PKG_ID BASELIME DEVICE,G OPN_ID OPN_REV DESIGN_I	70 Device U mmer - Clockd egister Editor mmer Identifi mber: ters: N_BASE EV: ADE EV: ISION D Auto Serce EVB EVB EVB EVB	operation failed on nknown (scan needed) Status Registers Status Registers 00-00-04-06-2C Refresh ID Re -	MCU; error code 0x7 Field Programmer: W CE gisters gisters Aarker CE Byteladdress=Ux020 B	DC_Wink(%C, L) A (general failure) ired to PCB (senial); ired to PCB (senial); Copy to Clipbe) => 0xh+ 0) 0)	Si538x/4x (not firmware bx		•	Field Program Freid Program Target: Interface: Part Numble Control Reg DSP DSP DSP DSP DSP DSP DSP DSP DSP DSP	SiSBX/Ak (not firmware based) Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V Wired to PCB (seri SPI 4-Wire; 12 MH 3.3V Wire; ERR- 27777777 Scan Star Star Scan Scan Scan Scan Scan Scan Scan Scan
Field Program Field Program Serial Nur DUT ID Regis DEVICE,P DEVICE,R	70 Device U nmmer - Clockd egister Editor mmer Identifi mber: tters: N_BASE EV: ADE ::D :RADE :StiON D :StiON D :EV8 :EV8	Intervention failed on Intervention failed on Intervention failed on Status Registers Status Registers Status Registers Status Registers Intervention	MCU; error code 0x7 Field Programmer: W CE gisters gisters darker Clear Verteaddress=0x027 Byte(address=0x027 Byte(address=0x027	DC_White(6C_L) A (general failure) ired to PC8 (serial); ired to PC8 (serial);<	Si538x/4x (not firmware bx		•	S3V Field Program Field Program Family: Target: Interface: Interface: Control Reg S S DSP DSP DSP DSP DSP P Harat	mmer SIS38x/dx (not firmware based) Wired to PCB (sein SIS4 - Wire 12 MH 3.3V rs - ERR- TTTTTTTT TTTTTTTTTTTTTTTTTTTTTTTTT

Figure 7.13. I2C and SPI Communication Failure Examples

UG286: ClockBuilderPro[™] Field Programmer Kit • Appendix A. Troubleshooting

2. Match the IO_VDD_SEL bit to the value in the plan that will be written to the device.

- a. If the IO_VDD_SEL bit already matches the value in the plan to be written, skip to step 3.
- b. If the IO_VDD_SEL bit is not correct, change the value and write the new value to the device (see the figure below).
- c. Re-configure the communication settings of the field programmer to re-establish communication to the device.

ile Help		
Info DUT Register Editor	Status Registers	- Field Programmer
Register Peek/Poke Her Address: 0x0048 # Bytes: 1 Umsigned Int:	Calcin Argins	Family: SIS38x/4k (not firmmare based) Target: Socket (PM64 Socket Power: On Interface: SIR14-Wire; 12 Mi42 3JV Part Number: SIS181A-A-GM Design ID: S3815VB2
		Frequency Adjust FINC FDEC
	On Different Maker Clear Copy to Opboard Pace	PDN: 0 Frequency Adjust FINC
itered 🚺 Auto Scroll: mestimp: Source: K	Animage	PDN: 0 Frequency Adjust FINC
itered I Auto Scroll: mestimp Source A 831342-216 EVB 10	esande Kead (DUT) (Stelladoress=0.0001) => 0x()	PDN: 0 Frequency Adjust FINC
itered Auto Scroll: mestamp Source A 4:31:42:216 EVB 10 4:31:42:216 EVB S	Hender Hender Hend, Dur, Dyfeldereis-Bollov) 1 => 1041+ Hender Hend, Dur, Dyfeldereis-Bollov) 1 => 1041+ Antor Paul (Dur, Dyfeldereis-Bollov) 1 => 1041+	PDN: 0 Frequency Adjust FINC
Auto Scroll: mestimp Source N 4231342216 EVB 10 431142216 EVB 50 431142216 EVB 50	Interaction	PDN: 0 Frequency Adjust FINC
Auto Scroll: mestamp Source N 431542216 EV8 1 431542216 EV8 1 431542216 EV8 5 431542220 EV8 5 431542220 EV8 5	Hence International State (Section 2007) - Section 2007 - Section	PDN: 0 Frequency Adjust FINC
Auto Scroll: Imestamp Source N 4/31342216 EV8 1 4/31342216 EV8 1 4/31342216 EV8 5 4/31342226 EV8 5 4/31342226 EV8 5	Interaction	PDN: 0 Frequency Adjust FINC
Source L (#231342216 EV8 S (#31342216 EV8 S (#31342226 EV8 S (#31342228 EV8 S (#31342228 EV8 S	Hence International State (Section 2007) - Section 2007 - Section	FOR 0

Figure 7.14. Re-configuring Communication Settings of the Field Programmer

3. Write your new plan to the device.

le Help		
Write Proje	ct File to Device	
Write Regis	ngs File to Device ter File to Device Dump File for Silicon Labs Support	
Preferences	Law Control of Control	
Exit		
Unsigned Ir Hex:	nt: 1 📄 0x01 7 6 5 4 3 2 1 0	
Binary:	0 0 0 0 0 0 0 1 (binary edit is only supported with 16 bits or less)	

Figure 7.15. Write New Plan to Device

Steps using CBPro Command Line Interface

1. Attempt to communicate with the Si534x8x7x9x device and determine the current value of the IO_VDD_SEL bit.

SPI communication Examples:

```
CBProDeviceRead.exe --io-voltage 1.8 --mode spi4wire --speed 1M --family si538x4x --registers 0x0943
CBProDeviceRead.exe --io-voltage 3.3 --mode spi4wire --speed 1M --family si538x4x --registers 0x0943
```

Note: The commands above are examples. Refer to the document and help for the CBPro CLI for your specific configuration.

I2C communication Examples:

```
CBProDeviceRead.exe --io-voltage 1.8 --mode i2c --speed 100k --i2c-address 0x68 --family si538x4x --
registers 0x0943
CBProDeviceRead.exe --io-voltage 3.3 --mode i2c --speed 100k --i2c-address 0x68 --family si538x4x --
registers 0x0943
```

Note: The commands above are examples. Refer to the document and help for the CBPro CLI for your specific configuration.

2. Match the IO VDD SEL bit to the value in the plan that will be written to the device.

a. A simple text file will need to be created that will write register 0x943 to 0x00 or 0x01.

To write 0x01 to 0x0943, the text file should contain the following single line of text:

0x0943,0x01

To write 0x00 to 0x0943, the text file should contain the following single line of text:

0x0943,0x00

b. Run the CLI command below to change the IO_VDD_SEL bit.

SPI Example:

```
CBProDeviceWrite.exe --mode spi4wire --speed 4M --io-voltage 3.3 --family si538x4x --registers simple_text_file.txt
```

I2C Example:

```
CBProDeviceWrite.exe --mode i2c -i2c-address 0x68 --speed 400K --io-voltage 3.3 --family si538x4x -- registers simple_text_file.txt
```

Note: The commands above are examples. Refer to the document and help for the CBPro CLI for your specific configuration.

3. Write the new plan to the part.

SPI Example:

CBProDeviceWrite.exe --mode spi4wire --speed 4M --io-voltage 3.3 --family si538x4x --project your_plan_name.slabtimeproj

I2C Example:

CBProDeviceWrite.exe --mode i2c -i2c-address 0x68 --speed 400K --io-voltage 3.3 --family si538x4x -project your_plan_name.slabtimeproj

Note: The commands above are examples. Refer to the document and help for the CBPro CLI for your specific configuration.

7.4 I burned a project file to my device with a new Base I2C address, but the base address in the device was not changed after the burn process was complete.

The I2C address will not be changed during the burn process. Changes to the base I2C address in the CBPro Configuration Wizard will be included in exports and the project file used to create orderable part numbers. However, this change is not burned to the device using the NVM Burn Tool. See the note highlighted in the figure below.

Base I2C Address
The upper 5-bits of the I2C address are configurable. The lower 2-bits are controlled using the A0 and A1 pins on the Si5342.
Address:
6 5 4 3 2 1 0
Address Range: 104d to 107d / 0x68 to 0x6B
Host interface registers defined by this page - IO_VDD_SEL_SPI_3WIRE, and I2C_ADDR - are not written to EVBs. They are included in exports and orderable part number non-
volatile memory. See the Family Reference Manual for more information.
volatile memory. See the Family Reference

Figure 7.16. Base I2C Address

To permanently change the I2C base address on your device, you need to use the I2C Address Burn Tool. See the figures below to use the tool.

icon Laboratories	
f Silicon Laboratories IDE	Travis Turner
CBPro Project File Inspector	
ClockBuilder Pro	Documents
CB CBPro EVB GUI	
🔁 CLI User's Guide	Pictures
CB ClockBuilder Pro	
🔁 In-System Programming Guide	Music
🔁 License	
🔁 Overview	Computer
🔁 Release Notes	
💿 Silicon Labs Timing	Control Panel
💿 Silicon Labs	Devices and Printers
😽 Uninstall ClockBuilder Pro	Devices and Printers
🌗 Misc 😑	Default Programs
CB CBPro Internal Tools	Deladit i Togranis
🕫 Rational Fraction Calculator	Help and Support
CB Si538x4x I2C Address Burn Tool	
CB Si538x4x Register Export Upgrade Tc	
PCIe Clock Jitter Tool	
Programmable Oscillator Software 4.0.1 👻	
4	
4 Back	
Count and the	
Search programs and files 👂	Log off D

B Si538x/4x I2C Address Burn To	ol 🗖 🗖 🗙
A new I2C address can only b	e the base I2C address on a supported Si538x/4x device. e burned one time. At least once, use the test button to change the s change without actually performing the burn.
Field Programmer Mode:	Kit socket board Wired to your own board, I2C
I/O Voltage	3.3 V
Current Base I2C:	0x6C
Address Pin State:	A0 A1
Base I2C Address to Program:	0x74
	Clear Copy to Clipboard
Timestamp Message	
	A
	v
Detect Test (Det	ect & Change Volatile) Burn (Detect, Change Volatile, & Burn NVM)

Figure 7.17. I2C Address Burn Tool

SKYWORKS

ClockBuilder Pro

Customize Skyworks clock generators, jitter attenuators and network synchronizers with a single tool. With CBPro you can control evaluation boards, access documentation, request a custom part number, export for in-system programming and more!

www.skyworksinc.com/CBPro



C

Portfolio www.skyworksinc.com/ia/timing

www.skyworksinc.com/CBPro



Quality www.skyworksinc.com/quality



Support & Resources www.skyworksinc.com/support

Copyright © 2021 Skyworks Solutions, Inc. All Rights Reserved.

Information in this document is provided in connection with Skyworks Solutions, Inc. ("Skyworks") products or services. These materials, including the information contained herein, are provided by Skyworks as a service to its customers and may be used for informational purposes only by the customer. Skyworks assumes no responsibility for errors or omissions in these materials or the information contained herein. Skyworks may change its documentation, products, services, specifications or product descriptions at any time, without notice. Skyworks makes no commitment to update the materials or information and shall have no responsibility whatsoever for conflicts, incompatibilities, or other difficulties arising from any future changes.

No license, whether express, implied, by estoppel or otherwise, is granted to any intellectual property rights by this document. Skyworks assumes no liability for any materials, products or information provided hereunder, including the sale, distribution, reproduction or use of Skyworks products, information or materials, except as may be provided in Skyworks' Terms and Conditions of Sale.

THE MATERIALS, PRODUCTS AND INFORMATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, INCLUDING FITNESS FOR A PARTICULAR PURPOSE OR USE, MERCHANTABILITY, PERFORMANCE, QUALITY OR NON-INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT; ALL SUCH WARRANTIES ARE HEREBY EXPRESSLY DISCLAIMED. SKYWORKS DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. SKYWORKS SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO ANY SPECIAL, INDIRECT, INCIDENTAL, STATUTORY, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUES OR LOST PROFITS THAT MAY RESULT FROM THE USE OF THE MATERIALS OR INFORMATION, WHETHER OR NOT THE RECIPIENT OF MATERIALS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Skyworks products are not intended for use in medical, lifesaving or life-sustaining applications, or other equipment in which the failure of the Skyworks products could lead to personal injury, death, physical or environmental damage. Skyworks customers using or selling Skyworks products for use in such applications do so at their own risk and agree to fully indemnify Skyworks for any damages resulting from such improper use or sale.

Customers are responsible for their products and applications using Skyworks products, which may deviate from published specifications as a result of design defects, errors, or operation of products outside of published parameters or design specifications. Customers should include design and operating safeguards to minimize these and other risks. Skyworks assumes no liability for applications assistance, customer product design, or damage to any equipment resulting from the use of Skyworks products outside of Skyworks' published specifications or parameters.

Skyworks, the Skyworks symbol, Sky5[®], SkyOne[®], SkyBlue[™], Skyworks Green[™], Clockbuilder[®], DSPLL[®], ISOmodem[®], ProSLIC[®], and SiPHY[®] are trademarks or registered trademarks of Skyworks Solutions, Inc. or its subsidiaries in the United States and other countries. Third-party brands and names are for identification purposes only and are the property of their respective owners. Additional information, including relevant terms and conditions, posted at www.skyworksinc.com, are incorporated by reference.



Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Skyworks:

Si5332-48SKT-DK Si5332-32SKT-DK Si5332-40SKT-DK