

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	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 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







O TP2 VDDA

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

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 (any supported Si5332, 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 so	ocket board detected. Attach socket now to burn NVM on loose parts.
if you want to burn NVM c you are programming usin	on a device attached to the field programmer via wired serial connection, select the device family of the selector above.
, <u>-</u> <u>-</u> <u>-</u>	······

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:

NVM Program Tool - Clock	Builder Pro v2.15	ΩΣ
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:	Select Project 	Clear
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	OPICEOKUP
Project Design ID:	5342BP3	
Project Design Check:	ОК	
Project File NVM Hash:	0x0F8E0A6F76FE61BD2DBC1EA751316866 (copy to clipboard	0
Device Part Number:	Present, Si5342I-B-GM	Scan for Device Clear
	(empty)	
Device Design ID:		
Device Design ID: Device NVM State:	1 bank used, 2 banks available for burn	
Device Design ID: Device NVM State: # Valid Burns:	1 bank used, 2 banks available for burn 0	2

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 firmwa Trying to send Success Flashing firmw	are file i device to bootloader mode mare	A v	Firmwar Detecte Firmwar DESIGN	re flashed d Si5385 in program mode re verified to be 1.1 (Pre-Release Build 31) _IDx is TEST1	
	Close			OK	



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.

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		Comm : News : 201492 : 15592 : E42 Log in Register
Products - Applications -	Support & Training - Buy or Sample -	Q Enter keyword
Silicon Labs > Products > Clocks :	and Oscillators > Clock and Oscillator Design Services	
Part Number Searc	th Results	
You searched for: Oscillator Clock or But	. 🛹 🕘	
Q SI5346B-A03260-	GM D C	
Existing Custom Parts	1 result	
\$15346B-A03260-QM		
Request Date	7/2/2014	
Part Number	SI63468-A03260-GM	
Part Number Revision	0	
Product	\$153448	
Data Sheet	E Data Sheet	
Data Sheet Addendum	a) 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: VI	DDA (3.3V)		
SPI Mode: 4-Wire			
I2C Address Range: 1	116d to 119d / 0x74	to 0x77 (selected	via AO/A1 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 device for I ² C communication. (Refer to specific part pin- out and the programming Field Programmer Socket Board to determine 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?	I 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 device for I2C communication. (Refer to specific part pin- out and the programming Field Programmer Socket Board to determine 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 $\mathsf{I}^2\mathsf{C}$ 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 - ClockB	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 of	on a device attached to the field programmer via wired serial connection, select the o	device family
you are programming usin	g the selector above.	

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 Sc	ocket Detected)		
Target Device:	Si5383/84 (firmware l	based)	🖬 🗰 1	
Host Interface:	I2C Address 0x6C / 10	08d; 400 kHz; 3.3V		
Firmware Source: 3 -	Configuration + Pr	ogram from Proje	ct File 😡	
	Project File:	C:\Users\tturner Project.slabtimes	Desktop\Si5383-RevD-5 proj	383EVB1- Select Clear
	Creator:	C8Pro v2.15		1
	Part:	Si5383 Rev D		4
	Firmware Release:	1.0 (Official Rele	ise)	
	Design ID:	5383EVB1		
	Design Check:	OK		
3	O Configuration + Pr	ogram from Firms	ware File 🛈	
Firmware Hash:	0xA54188451152105F	4E04D49868E027	3D (copy to dipboard)	0
Device Part Number:			5 •	Scan for Device Clear
Device Firmware:	***			· Commission and Commission
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 I²C, 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 (No S	ocket Detected)		
Target Device:	Si538x/4x (not firmw	vare based) 📱 📛	1	
Host Interface:	I2C Address 0x6C / 1	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	SPI 4-Wire SPI 3-Wire I2C 3 V	3 Select Project	Clear
Device Part Number: Device Design ID: Device NVM State:	12C Address 0x 7-1 12C Bus Speed 40	6C 108 D bit address, range 1-127 00 kHz	4 🗪 Scan for Device	Clear
# Valid Burns: # Burns with Error:	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:

CB	Flash Firmware	×	CB	Flash Firmware	- 🗆 🗙
Parsing firmwa Trying to send Success Flashing firmw	are file i device to bootloader mode ware	k. V	0	Firmware flashed Detected Si5383 in program mode Firmware verified to be 1.1 (Pre-Release Build 31) DESIGN_IDx is TEST1	
	Close		-	OK	



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 Desi	gn - ClockBuilder Pro 🛛 🗕 🗖 🗙
ClockBuilder Pro v2.14.4 🎭	SILICON LABS
Design Dashboard 🔻	Configuring Si5345 Rev D
Loaded sample design 5345EVB1. You can review or edit the config section below or using the pulldown step menu above.	uration by clicking a link in the "Edit Configuration with Wizard"
Edit Configuration with Wizard Design ID & Notes · Revision · Host Interface · XA/XB · Free Run · ZDM · Inputs · Input Select · Outputs · Planner · DCO · Output Skew · Output Drivers · DSPLL · LOS · OOF · LOL · INTR	Field Programmer Detected Target: Wired to PCB (serial) Interface: I2C Address 0x68 / 104d; 400 kHz; 3.3V Write Design to DUT Open EVB GUI
Save Design to Project File Your configuration is stored to a project file, which can be opened in ClockBuilder Pro at a later time.	You can export your configuration to a format suitable for in-system programming.
Design Report & Datasheet Addendum You can view a <u>design report (text)</u> or create a <u>draft datasheet addendum (PDF)</u> for your design.	Documentation Si5345/44/42 Rev D Family Reference Manual Si5345/44/42 Rev D Datasheet Si5345 Rev D EVB User's Guide
Filicon Labs Cloud Services You can <u>create a custom part number</u> for your design, which can be used to order factory pre-programmed devices. Or <u>request a phase noise report</u> for this design.	Ask for Help Have a question about your design? Click <u>here</u> to get assistance.
Frequency Plan Valid Obesign OK OPd: 1.215 W, Tj: 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.

	Field P	rogrammer Detected
	Target:	Wired to PCB (serial)
	Interface:	I2C Address 0x6C / 108d; 100 kHz; 3.3V
	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.

				Field Pr Target:	rogrammer I Wired to PCB (se	Detected rial)	
				Interface:	I2C Address 0x6	C / 108d; 100 kHz; 3.3V	1
				Write	Protocol	○ SPI 4-Wire ○ SPI 3-Wire	
	Field Programmer Detected Target: Wired to PCB (serial) Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V		Þ	Export You can e in-system	I/O Voltage	 I2C 3.3 V 	ble for
	Write I I2C Address 0x6C 108 7-bit address, range 1-127				I2C Address	0x6C 108	
•	You can e I2C Bus Speed 100 kHz IIC Bus Speed I00 kHz	ble for		Docum Si5341/4 Si5341/4	I2C Bus Speed	100 kHz	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:

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

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	🗒 Fi
Target: Socket, LGA56 Off	Ta Ta
Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V	In
Write Design to DUT Open EVB GUI	

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



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 Pr Target: Interface:	Socket, QFN44	Detected Off (Hz; 3.3V	1
Field Programmer Detected Target: Socket, LGA56 Off Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V		Write Export	Protocol	SPI 4-Wire SPI 3-Wire I2C		-
Write I2C Address 0x6C 108 7-bit address, range 1-127		You can e in-system	I/O Voltage I2C Address	3.3 V 0x6C	108	ble for
You can e in-system programming.	ble for	Docum Si5345/4 Si5345/4 Si5344 Re	I2C Bus Speed	7-bit address, r 100 kHz sheet Guide	ange 1-127	

Figure 4.29. Interface Settings

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

		granner			
	Target:	Socket, LGA56	On		
	Interface:	I2C Address 0x	6C/1	08d; 100 kHz; 3.3V	
_	Write	Design to DU	T	Open EVB GUI	

Figure 4.30. Write Design to DUT

Or on any configuration page in the wizard:

Write to FP < Back Next >	Finish Cancel

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:

	Config Sca	in
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 📓
	in the find the second	7-bit address, range 1-127
	12C 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

Field Programmer

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.

		Family: Si538x/4x (not firmware based) Target: Wired to PCB (serial) Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V
	Field Programmer	Part Number:
	Family: Si5383/84 (firmware based)	Design ID:
	Target: Wired to PCB (serial)	Config Scan
	Interface: I2C Address 0x6C / 108d: 100 kHz: 3.3V	Control Device Family Si538x/4x (not firmware based)
Field Programmer		Control Device Fulling Sission 4x (not infinitial busica)
Family: Si538x/4x (not firmware based)	Part Number: Design ID:	Protocol OSPL 4-Wire
Target: Wired to PCB (serial)	Config Scan	I2C
Interface: SPI 4-Wire; 1 MHz; 3.3V	Control Device Family Si5383/84 (firmware based)	I/O Voltage 3.3 V
Part Number Design ID:	I2C Address 0x6C 108	I2C Address 0x6C 108 7-bit address, range 1-127
Config Scan	I2C Bus Speed 100 kHz	I2C Bus Speed 100 kHz

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 Program	mmer	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	r: Si5383A-D06791-GM	OPN_REVISION	0					
Design ID:	5383EVB1	DESIGN_ID	5383EVB1					
Co	onfig 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 Program	mer	VCO_VARIANT	0					
Family:	Si5383/84 (firmware	TEMP_GRADE	Industrial					
Target:	based) Socket LGA56	PKG_ID	3					
Socket Power:	On the second se	BASELINE_ID	18					
Interface:	I2C Address 0x6C /	DEVICE_GRADE	A					
	108d; 100 kHz; 3.3V	OPN_ID	06791					
Part Number:	Si5383A-D06791-GM	OPN_REVISION	0					
Design ID:	(empty)	DESIGN_ID						
Cont	fig 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 5.5. 56-Pin Socket Board Schematic









Figure 5.7. Si5332 48-Pin Socket Board Schematic





Figure 5.8. Si5332 40-Pin Socket Board Schematic





Figure 5.9. Si5332 32-Pin Socket Board Schematic

6. Bill of Materials

6.1 CBPROG-DONGLE Bill of Materials

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB Footprint	ManufacturerPN	Manufacturer
	Quantity	C2 C3 C8 C11 C26	Funde	ite ting	Fortuge	Torcrunce	Type			mananacarci
	0	C2 C3 C8 C11 C20	1		101	+10%	V70	coco2	C0C02X7D1C0 10FK	Verkel
	9	C28 C30 C31 C32	TUP		101	10%	A/K	0603	C0603X7R160-105K	venkei
	3	C24 C27 C29	0.01uF		16V	±20%	X7R	C0603	C0603X7R160-103M	Venkel
	1	C39	33pF		25V	±10%	COG	C0402	C0402C0G250-330K	Venkel
		C4 C6 C12 C13								
		C14 C35 C36 C37								
	9	C38	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
	2	C5 C25	4.7uF		10V	±20%	X7R	C1206	C1206X7R100-475M	Venkel
	1	C9	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	2	D14 D15	ITL-14CHI	20m∆				LED-T1-KK	ITI-14CHI	LITE-ON TECHNOLOGY CORP.
	1	D14015	SDOEO2DAUT	200mW	2017		TV/S	SOT142 AVVVISOT142	SDOE02RAUTC	Littlefuce
	-	DC D7 D0 D0 D10	SF0503BAITT	3001114	200		103	301143-AKKK 301143	SF0303DAITIG	Littlefuse
	_	06 07 08 09 010								
	/	D11 D12	GL05T-E3-08	5A	11V		Dual Common Anode	SO123-123	GL05T-E3-08	Vishay
	1	FB1	22 Ohm	6000mA			SMT	L0805	BLM21PG220SN1	MuRata
	1	J2	USB Type B				USB	CONN-USB-B	61729-0010BLF	FCI
	1	13	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	DE0 D67	10%	1/1014		+19/	ThighEiles	P04031 P04031	CR0402 16W 10025	Veskel
	10	K39 K67	TOK	1/10//		1170	THICKFIIM	N0402 N0402L	CR0402-1000-1002F	venker
	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	R3	100	1/16W		+1%	ThickFilm	R0402	CR0402-16W-1000F	Venkel
	1	R33	4 42K	1/16W		+1%	ThickFilm	80402	CR0402-16W-4421F	Venkel
	1	R3/	0 524	1/161		+1%	ThickFilm	R0402	CP0402 16W 95215	Venkel
	1	D24	5.JSK	1/1611		+19/	ThickFilm	80402	CR0402-10W-5551F	Venkel
	1	ROD	5.9K	1/1677		11%	THICKFILM	R0402	CR0402-16W-5901F	venkei
	2	R39 R66	1.0	3/4W		±1%	ThickFilm	R1210	CRCW12101R00FKEAHP	Vishay Dale
	2	R44 R47	8.06K	1/16W		±0.1%	±25PPM	R0402	TFCR0402-16W-E-8061B	Venkel
	2	R45 R48	17.4K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1742F	Venkel
	2	R46 R49	21.5K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-2152F	Venkel
	2	R50 R51	20K	1/10W		±1%	ThickFilm	R0603	CR0603-10W-2002F	Venkel
	4	R55 R56 R57 R58	402	1/16W		±1%	ThickFilm	R0402	CR0402-16W-4020F	Venkel
	1	R62	499	1/16W		+1%	ThickFilm	R04021R04021	CR0402-16W-4990F	Venkel
	- 2	R63 R64	470	1/161		+5%	ThickFilm	R0402	CR0402-16W-4711	Venkel
	2	P0 P10	10.04	1/101		+0.1%	#2EDDM	R0402		Panaconic
	2	N3 N10	DUMPED	1/10//		10.170	TZJELIMI	RUDDED FOOT O 250	CIE 202	
	2	5F1 5F2	BUIVIPER					RUBBER_FOOT_0.250	515382	31VI
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
	2	2 SF3 SF4	SPACER					N/A	7363	Keystone Electronics
	1	L U1	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	1112	TP\$76201	100mA			LDO	SOT5N2 8P0 95	TPS76201DBV	TI
	-	01121114	TPS70501	500mA			LDO	DEN9N2 OP0 65E2 4V1 65	TDS70501DBBT	т
	-	1116	ADG710	JOOTTA			100	SOTEN2 900 05	ADG7198877	Analog Dovices
	-	010	ADG/19		0.4.514		D. ((3010N2.8P0.93		Analog Devices
		UI/	NLSV11244		.9-4.5V		Buπer	UDFN6NIPU.4	NLSV11244WUTBG	Un Semi
	1	L U2	C8051F380				мси	QFP48N9X9P0.5	CF380P1104AGQ	SiLabs
	2	2 U3 U8	ADG712					TSSOP16N6.4P0.65	ADG712BRU	Analog Devices
	t	L U5	LTC4311		5.5V			SC70-6N2.1P0.65	LTC4311CSC6#TRMPBF	Linear Technology
	1	L U6	PCA9517D				12C	SO8N6.0P1.27	PCA9517D	NXP
	1	L X1	2X5 Header				Shrouded	CONN2X5-RA-SBH11	SBH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Com	ponents									
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB Footprint	ManufacturerPN	Manufacturer
NI		C10 C40	1uE		16V	+10%	X7R	C0603	C0603X7R160-105K	Venkel
NI		C10C40	0.105		101	+10%	V7D	C04031 C04031	C0403X7R100-104K	Vonkol
		0/			100	110%	A/R		C0402A7R100-104K	venkel Gewiter
INI NU	1	JP1	HEADER 4X1				Header	CONN-1X4	15vv-104-07-1-S	samtec
NI	2	R60 R61	0	1A			ThickFilm	RU603	CR0603-16W-000	Venkel
		TP1 TP2 TP3 TP9								
NI	5	5 TP10	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	1	L TP7	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI	1	L TP8	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn
NI	1	114	AD8628		5V		OPAMP	SOT23-5N	AD8628AUI-R2	Analog Devices

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	Type	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	Туре	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 Boar	d Name	SI5332-48SKT	1							
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
	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 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

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

Eval Boa	ard Name	SI5332-40SKT]							
Rev	ision	1.0	_							
CreationDate	Proto Rev	Released								
5/18/20	17 1.00	0								
5/10/20	1.00	, ·								
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Туре	PCB_Footprint	ManufacturerPN	Manufacturer
	1	2 C3 C8	0.1uF		10V	±10%	X7R	C0402 C0402L	C0402X7R100-104K	Venkel
	1	2 C6 C7	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	4	SF1 SF2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	ЗM
	:	L U1	40QFN-SKT, 6x6x0.5mm				QFN		790-42040-101G	Sensata
	:	L U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
		L X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Co	mponents									
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	:	L TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI		L TP7	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

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

Eval Boa	rd Name		SI5332-32SKT								
Rev	ision		1.0								
CreationDate	Proto Rev		Released								
5/18/20	17	1.00	0								
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	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	mponents										
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

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	Builder 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: Broject Docian ID:		Select Project Clear OPN Lookup
Project Design ID. Project Design Check: Project File NVM Hash:	Ø	
Device Part Number: Device Design ID: Device NVM State:	Error communicating with device	Scan for Device Clear
# Valid Burns: # Burns with Error:	0 0 Drogram 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.

ield Programmer Mode:	Wired to Board (I	No Socket Detected)		
arget 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 I2C Bus Speed 	0x6C 108 7-bit address, range 1-122	4 4	Scan for Device Clear
# Valid Burns: # Burns with Error:	0			

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 - ClockBui	ider Pro	
File Help		
Info DUT Register Editor	Status Registers	Field Programmer
Field Programmer Identifica	tion:	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 🔄 Auto Scroll:	On 📱 Insert Marker Clear Copy to Clipboard Pause	Hard Poset Suns R
Timestamp Source I	Message	Power Down
	peration failed on MCU; error code 0xFA (general failure)	FW83_HARD_RST
09:43:40.988 EVB 5	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 S	Starting Read_DUT_Byte(address=0x026B)	PDN: 0
09:43:40.992 EVB	error Read_DUT_Byte(address=0x026B) => 12C_Write(i2c_bus=1, i2c_slave_address=0x6C, data=0x0102):	
		Frequency Adjust
EVB Firmware 0.70 Device Unk	nown (scan needed) Field Programmer: Wired to PCB (serial); Si538x/4x (not firmware based); I2C Address 0x6C / 108d; 400 kH	z; 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	elp		
info	DUT Register Editor	Status Registers	- Field Programmer
ield I	Programmer Identifi	ration:	Family: Si538x/4x (not
Se	rial Number:	00-00-04-06-2C-CE	Target: Wired to PCB (serial
DUT I	D Registers:	Refresh ID Registers	Interface: SPI 4-Wire; 12 MHz; 3.3V
D	EVICE_PN_BASE		Part Number: -ERR-
DI	E_REV		Design ID: ???????
D	EVICE_REV:		Config Scan
TE	MP_GRADE		
Pk	(G_ID		Control Registers
BA	ASELINE_ID		Soft Reset and
DI	EVICE_GRADE		
O	PN_ID		
O	PN_REVISION		
DI	ESIGN ID		DSPLLB_SOFTRESET
	-		DSPLLC_SOFTRESET
og			DSPLLD_SOFTRESET
Filtere	ed 🔽 Auto Scro	II: On Insert Marker Clear Copy to Clipboard Pause	Hard Reset, Sync, & Power Down
9:44:	32.826 EVB	hinished Kead_DU1_Byte(address=0x026F) => 0xFF	FW83 HARD RST
9:44:	32.826 EVB	Starting Read_DUT_Byte(address=0x0270)	
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
9:44:	32.833 EVB	Starting Read_DUT_Byte(address=0x0272)	
9:44:	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.

ile Help				
Info DUT Register Editor	Status Registers	+ Field Pr	ogrammer	
Field Programmer Identifi	cation:	Family:	Si538x/4x	(not
Serial Number:	00-00-04-06-2C-CE	Target:	Wired to P	CB (serial)
DUT ID Registers:	Refresh ID Registers	Interfac	e: I2C Addres	ss 0x68 / kH7: 1.8V
DEVICE_PN_BASE		-		
DIE_REV		Part Nu Design	mber: ID:	
DEVICE REV:				
TEMP GRADE			Config Sc	an
PKG_ID		Contro	Device Family	Si538x/4x (not firmware based)
BASELINE ID			Protocol	O SDI 4-Wire
DEVICE GRADE			PIOLOCOI	O SPI 3-Wire 1 and
OPN ID				I2C
OPN REVISION			I/O Voltage	18V 🗖 🛻 3
DESIGN ID			, o ronage	
TOOL VERSION			I2C Address	0x68 104 🔡
TOOL_VERSION				7-bit address, range 1-127
			I2C Bus Speed	400 kHz
			- romer pom	
			FW83_HARD_F	RST
			RST_REG	
			SYNC_REG	
			PDN: 0	
00			1.2000000000000000000000000000000000000	
Filtered 🔽 Auto Scro	II: On 🗧 Insert Marker Clear Copy to Clipboard Pause		Frequency Adj	ust
Timortamp Source			F_INC_KEG	
11:13:54.167 EVB	tinished Kead DUI_Byte(address=0x026F) => 0xF+			
11:13:54.167 EVB	Starting Read_DUT_Byte(address=0x0270)			
11:13:54.170 EVB	finished Read_DUT_Byte(address=0x0270) => 0xFF			
11:13:54.170 EVB	Starting Read_DUT_Byte(address=0x0271)			
11:13:54.175 EVB	finished Read_DUT_Byte(address=0x0271) => 0xFF			
11:13:54.175 EVB	Starting Read_DUT_Byte(address=0x0272)			
11.13.54 101 51/0	finished Read DUIT Bute/address=0v0272) => 0vEE			

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.

	- Fall Programmer
In Program (III) Non-Kappen International (IIII) Non-Kappen	- Sald Programmer Family Elizibatio (not Emission family
	family USAbate pet
	Topy Look (1964) Start War (1969) Start War (1969) Start War (1969) Start War (1969) Start War (1969) Control Start (1969) Start War (1969) Start (1
ent 📓 Adi Sout (n 🗋 (mer Meter) (per) (per 6 Option) Point	
MARTIN WAS MADE IN A REAL ON A REAL AND	
ADALTI SM Starting Read, DVT, Autopolities-MARCE	
dolla 1/7 (full finished Natel, D.C. Julyanikers, -boll(2), s = ball	
604.2% Evb Starting fixed (U.C. Astrophics - 0x000)	
IDEAL THE Read AND AND AND ADDRESS - DOIN	
minait (VE Starting fast), 27, Sylveyabless - Selful Aver, Sylver 15	
All and a state of the second state of the sec	14
and the first first hardware had the life of the hardware had been and	Double in the state of the later.

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

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 We Make Timing Simp	Wizard O le Q
Work With a Design	Quick Links
Create New Design	Clock Generators & Jitter Attenuators Knowledge Base
🖶 Open Design Project File	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
	ClockBuilder Pro Documentation
18 12 Mg	CBPro Overview CBPro Tools & Support for In-System Programming Includes walthcoughs of frequency-on-the-fly, full configuration, and partial configuration programming scenarios. CILUSers's Guide Release. Notes • Knowledge Base
o,	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 Programmer Family: Si538x/4x (not firmware based) Target: Socket OFN64		Field Prog Family:	rammer Si538x/4x (firmware b	(not lased)	
Socket Power: Off Interface: SPI 4-Wire; 12 MI 1.8V	■ 2 Hz;	Target: Interface:	Wired to P SPI 4-Wire 3.3V	CB (serial) ; 12 MHz;	
Part Number: Design ID: Config Scan	— 3	Design ID:	Config Sc	an 4 3 Si538x/4x (not firmw	vare based)
Control Protocol SPI SPI SPI I20	4-Wire 3-Wire		Protocol	 SPI 4-Wire SPI 3-Wire I2C 	
I/O Voltage 1.8 V			/O Voltage	3.3 V	
SPI Bus Speed 12 M	1Hz		SPI Bus Speed	12 MHz	

Figure 7.12. Configuring Communication Settings

Examples of a Communication failure for I2C and SPI:

	Charles Internet	- Field Brogrammer
Info DUT Register Editor	Status Registers	Field Programmer
ield Programmer Identific	ation:	firmware based)
Serial Number:	00-00-04-06-2C-CE	Target: Wired to PCB (seria
OUT ID Registers:	Refresh ID Registers	Interface: 12C Address 0x0C / 108d; 400 kHz; 3.3V
DEVICE_PN_BASE		Part Number -FRB-
DIE_REV		Design ID: -ERR-
DEVICE_REV:		Config
TEMP_GRADE		
PKG_ID		Control Registers
BASELINE_ID		Soft Reset and Calibration
DEVICE_GRADE		SOFTRESET G
OPN_ID		
OPN_REVISION		
DESIGN_ID		
		DSPLLC_SOFTRESET
iltorod 🔲 Auto Coort	Los 🗖 Inset Marker Class Convite Cliphoard Prive	DSPLLD_SOFTRESET
Auto scrol	. on ander marker clear copy to clipboard Pause	Hard Reset, Sync, &
imestamp Source	Message	Power Down
0-42-40.088 EVR	operation tailed on MCU; error code 0xFA (general failure) Starting Read DLIT Byte/address=0x0000)	FW83_HARD_RST
9:43:40.989 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
9:43:40.991 EVB	Starting Read_DUT_Byte(address=0x026B)	PDN: 0
9:43:40.992 EVB	error Read_DUT_Byte(address=0x026B) => 12C_write(12C_bus=1, 12C_slave_address=0x6C	, data=0x0102):
B Firmware 0.70 Device Uni Field Programmer - ClockBu	operation failed on MCU; error code 0xFA (general failure) cnown (scan needed) Field Programmer: Wired to PCB (serial); Si338:/4x (not firmware based); I iider Pro	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 3.3V
8 Firmware 0.70 Device Uni Field Programmer - ClockBu e Help	operation failed on MCU; error code 0xFA (general failure) cnown (scan needed) Field Programmer: Wired to PCB (serial); Si538x/4x (not firmware based); I iider Pro	Frequency Adjust
8 Firmware 0.70 Device Uni Field Programmer - ClockBu e Help nfo DUT Register Editor	operation failed on MCU; error code 0xFA (general failure) crown (scan needed) Field Programmer: Wired to PCB (serial); Si338:/4x (not firmware based); I iider Pro	Frequency Adjust Frequency Adjust CAddress 0x6C / 108d; 400 KHz; 3.3V Field Programmer Fi
8 Firmware 0.70 Device Uni Field Programmer - ClockBu e Help nfo DUT Register Editor ield Programmer Identific	operation failed on MCU; error code 0xFA (general failure) crown (scan needed) Field Programmer: Wired to PCB (serial); Si338/4k (not firmware based); I iilder Pro Status Registers tion:	Y Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 3.3V Image: Comparison of the
8 Firmware 0.70 Device Uni Field Programmer - Clock8u e Help nfo DUT Register Editor ield Programmer Identificu Serial Number:	operation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (serial); 5538s/4k (not firmware based); I iiider Pro Status Registers Ition: 00-00-04-06-2C-CE	Frequency Adjust Frequency Adjust CAddress 0x6C / 108d; 400 kHz; 33V Field Programmer Family: \$\$538X/dx (not firmware based) Target: Wired to PCB (seria
8 Firmware 0.70 Device Uni Field Programmer - ClockBu e Help nfo DUT Register Editor ield Programmer Identific: Serial Number: DUT ID Registers:	operation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (serial); 5538s/4x (not firmware based); iiider Pro Status Registers Ution: 00-00-04-06-2C-CE Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 33V Field Programmer Family: Si538x/4x (not family: Si538x/4x (not Target: Wired to PC8 (seria Interface: SPI 4-Wire; 12 MHz 33V
B Firmware 0.70 Device Uni Field Programmer - ClockBu e Help nfo DUT Register Editor Serial Number: DUT ID Registers: DEVICE,PN_BASE	operation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (serial); 5538x/4x (not firmware based); I iider Pro Status Registers Torn: 0 0-00-04-06-2C-CE Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 33V Field Programmer Family: Si538X/4x (not family: Si538X
B Firmware 0.70 Device Uni Field Programmer - ClockBu e Help nfo DUT Register Editor ield Programmer Identifici: Serial Number: UTI ID Registers: DEVICE_PN_BASE DIE_REV	operation failed on MCU; error code 06FA (general failure) crown (scan needed) Field Programmer: Wired to PCB (serial); 5538/4x (not firmware based); I ii/der Pro Status Registers tion: 00 00-04-06-2C-CE Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 3.3V Field Programmer Family: Si538x/4x (not family: Si538
Firmware 0.70 Device Uni Field Programmer - ClockBu Help Info DUT Register Editor ield Programmer Identifica Serial Number: UUT ID Registers: DEVICE_PN_BASE DEVICE_PN_BASE DEVICE_REV:	aperation failed on MCU; error code 06FA (general failure) crown (scan needed) Field Programmer: Wired to PCB (serial); 5/338/4k (not firmware based); I iilder Pro Status Registers tion: 00-00-04-06-2C-CE Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 3.3V Field Programmer Family: Si538x/4x (not firmware based) Target: Wired to PCB (seria lineface: SPI 4-Wire; 12 MHz 3.3V Part Number; EER- Design ID: 7777777 Confid
B Firmware 0.70 Device Uni Field Programmer - ClockBu Help for DUT Register Editor ield Programmer Identifici- Serial Number: UT ID Register Editor DUT ID Register Editor DEVICE, PN_BASE DIE, REV DEVICE, REV: TEMP_GRADE	operation failed on MCU; error code 06FA (general failure) nnown (scan needed) Field Programmer: Wired to PCB (serial); S538x/4k (not firmware based); nilder Pro Status Registers ttion: 00-00-40-60-2C-CE Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 kHz; 33V
Firmware 0.70 Device Unit Field Programmer - ClockBu e Help DUT Register Editor ield Programmer Identific Serial Number: DUT D Registers: DEVICE_PN_BASE DIE_REV DEVICE_REV: TEMP_GRADE PKG_JD	operation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (serial); 5538x/4x (not firmware based); ilder Pro Status Registers Status Registers Refresh ID Registers Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 14Hz; 3.3V Field Programmer Family: Si538Xdx (not Target: Wired to PCB (seria Interface: SPI 4-Wire; 12 MHz 3.3V Part Number: -ERP. Design ID: 7777777 Config Scan Control Registers
Firmware 0.70 Device Uni Field Programmer - ClockBu e Help nfo DUT Register Editor ield Programmer Identific Serial Number: VUT ID Registers: DEVICE_PN_BASE DIE_REV DEVICE_REV: DEVICE_REV: TEMP_GRADE PKG_JD BASELINE_ID	aperation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (scrial); 5538x/4x (not firmware based); iider Pro Status Registers Status Registers Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 33V
B Firmware 0.70 Device Uni Field Programmer - ClockBu Help Info DUT Register Editor Serial Number: VUT ID Registers: DEVICE_PN_BASE DIE_REV DEVICE_REV: TEMP_GRADE PKG_ID BASELINE_ID DEVICE_GRADE	aperation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (serial); 5538x/4x (not firmware based); iider Pro Status Registers Status Registers Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 33V Field Programmer Family: Si338X/4x (not firmware based) Target: Wired to PCB (seria Interface: S9T 4-Wire; 12 MHz 33V Patt Number: E8P- Design ID: TTTTTTT Config Scan Control Registers Soft Reset and Calibration Soft Reset and Calibration Control Register Control
Firmware 0.70 Device Unit Field Programmer - Clock8u Help Info DUT Register Editor ield Programmer Identific: Serial Number: UT ID Registers: DEVICE, PN_BASE DIE REV DEVICE, REV: TEMP_GRADE PKG_JD BASELINE, JD DEVICE, GRADE OPN_JD	aperation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (serial); 5538/4x (not firmware based); milder Pro Status Registers Status Registers Refresh ID Registers Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 33V Field Programmer Family: SIS38X/4x (not firmware based) Target: Wired to PCB (seria Interface: S0F 4-Wire; 12 MHz Beigin ID: #777777 Control Registers Soft Reset and Calibration SOFTRESF_C6 DEVIA SVETECET
B Firmware 0.70 Device Unit Field Programmer - ClockBu Help for DUT Register Editor ield Programmer Identific Serial Number: UTI DR Registers: DEVICE, PN_BASE DIE, REV DEVICE, REV. TEMP_GRADE PKG_ID BASELINE, ID DEVICE, GRADE OPN_REVISION	operation failed on MCU; error code 06FA (general failure) nnown (scan needed) Field Programmer: Wired to PCB (serial); S538s/Ak (not firmware based); nilder Pro Status Registers stion: 00-00-04-06-22-CE Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 kHz; 33V
B Firmware 0.70 Device Unit Field Programmer - ClockBu e Help for DUT Register Editor ield Programmer Identific Serial Number: DUT ID Registers: DEVICE,PN_BASE DEVICE,PN_BASE DEVICE,REV: TEMP_GRADE PKG_JD BASELINE,JD DPVICE,GRADE OPN_REVISION DESIGN_JD	aperation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (serial); 5538s/4k (not firmware based); I ilder Pro Status Registers Status Registers Refresh ID Registers Refresh ID Registers	Frequency Adjust 2C Address 0x6C / 108d; 400 kHz; 33V Field Programmer Family: SIS38x/dx (not Target: Wired to PCB (serial Interface: Syr 4-Wire; 12 MHz; 33V Part Number: -ERR- Design ID: ???????? Config Scan Control Registers Soft Reset and Calibration SOFTRESET DEPLA_SO
B Firmware 0.70 Device Unit Field Programmer - ClockBu e Help fo DUT Register Editor ield Programmer Identific Serial Number: DEVICE_PN_BASE DE_REV DEVICE_REV: TEMP_GRADE PKG_JD BASELINE_JD DEVICE_GRADE OPN_JD OPN_REVISION DESIGN_JD	aperation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (scrial); 5538/4k (not firmware based); I ilder Pro Status Registers Status Registers Refresh ID Registers Refresh ID Registers Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 HHz; 33V
B Firmware 0.70 Device Uni Field Programmer - ClockBu e Help nfo DUTR egister Editor Serial Number: DEVICE, PN_BASE DIE_REV DEVICE, REV: DEVICE, REV: TEIMP_GRADE PKG_JD BASELINE_ID DEVICE_GRADE OPN_REVISION DESIGN_ID 9 9 9	aperation failed on MCU; error code 06FA (general failure) mown (scan needed) Field Programmer: Wired to PCB (scrial); 5538/4x (not firmware based); iider Pro Status Registers Total Status Registers Refresh ID Registers Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 kHz; 33V Field Programmer Family: Si538x/4x (not firmware based) Target: Wirde to PC8 (seria Interface: S01 4-Wire; 12 MHz 33V Part Number: -ER8- Design ID: ???????? Control Registers Soft Reset and Calibration SofTRESET DSPLIA_SOFTRESET
B Firmware 0.70 Device Uni Field Programmer - ClockBu e Help for DUT Register Editor ield Programmer Identific Serial Number: UT ID Registers: DEVICE, PN_BASE DEVICE, PN_BASE DEVICE, RAV. TEMP_GRADE PKG_ID DEVICE, GRADE PKG_ID DEVICE, FKG_ID PKG_I	operation failed on MCU; error code 067A (general failure) inder Pro Status Registers ation: 00-00-04-06-2C-CE Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 kHz; 33V Field Programmer Family: \$338X/dx (nd fm/wave based) Target: Wired to PCS (seria Interface: \$97 + Wire; 12 MHz 33V Part Number: E8R- Design ID: 7777777 Config Scan Control Registers Soft Reset and Calibration SOFTRESET_G DSPLIG_SOFTRESET
B Firmware 0.70 Device Uni Field Programmer - ClockBu le Help Info DUT Register Editor Serial Number: DUT ID Register: DUT ID Register: DUT ID Register: DEVICE_PN_BASE DE_REV DEVICE_REV: TEMP_GRADE PKG_ID BASELINE_ID DEVICE_GRADE PKG_ID BASELINE_ID DEVICE_GRADE OPN_RUSION DESIGN_ID OPN_RUSION DESIGN_ID OPN_RUSION DESIGN_ID	appendion failed on MCU; error code 06FA (general failure) and on MCU;	Frequency Adjust 2C Address 0x6C / 108c; 400 kHz; 33V
B Firmware 0.70 Device Uni Field Programmer - ClockBu le Help hol DUT Register Editor Serial Number: DUT ID Registers: DUT		Field Programmer Family: Si538/dx (rod Target: Wirde to PCS (serial Interface: Sy14-Wire; 12 MHz 33V Part Number: -ERR- Design ID: 7777777 Confg Scan Control Registes Soft Reset and Calibration SOFTRESET OSPLIA_SOFTRESET OSPLIA
B Firmware 0.70 Device Uni Field Programmer - ClockBu le Help noto DUT Register Editor Serial Number: DUT ID Register Editor DEVICE, PN, BASE DE, REV DEVICE, REV: TEMP_GRADE PKG_JD BASELINE, JD DEVICE, GRADE OPN_JR DEVICE, GRADE DEVICE, GRA	apperation failed on MCU; error code 06/A (general failure) anown (scan needed) nider Pro status Registers status Registers status Registers ation: 00-00-04-06-2C-CE Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 HHz; 33V Field Programmer Family: Si538x/4x (not Target: Wirde to PC8 (serial Interface: S91 4-Wire; 12 MHz 33V Part Number: -ERR- Design ID: 7777777 Config Scan Control Registers Soft Reset and Calibration SofTRESET DSPLIA_SOFTRESET DSPLIA_SOFTRE
B Firmware 0.70 Device Uni Field Programmer - ClockBu the Help Info DUT Register Editor Serial Number: JUT ID Register: DEVICE, PN_BASE DEVICE, PN_BASE	operation failed on MCU; error code 067A (general failure) inder Pro iider Pro Status Registers ation: 00-00-04-06-2C-CE Refresh ID Registers	Frequency Adjust 22 Address 0x6C / 108d; 400 HHz; 33V Field Programmer Family: SIS38X/4x (not firmware based) Target: Vield DPCB (serial Interface: S91 4-Wire; 12 MHz 3-3V Part Number: -ERE- Design ID: PTTTTTTT Control Registers Soft Reset and Calibration SOFTRESET DSPLID.SOFTRESET STRLCRE SYNC.REG SYNC.REG
B Firmware 0.70 Device Uni Field Programmer - ClockBu the Help Info DUT Register Editor Serial Number: DUT ID Register: DEVICE_PN_BASE DEVICE_PN_BASE DEVICE_REV: DEVICE_REV: DEVICE_GRADE PKG_JD BASELINE_JD DEVICE_GRADE PKG_JD BASELINE_JD DEVICE_GRADE OPN_RUSION DESIGN_ID OPN_RUSION DESIGN_ID SP34432826 EVB SP34432826 EVB SP34432829 EVB SP34432829 EVB	apprendiction failed on MCU; error code 06FA (general failure) answer (scan needed) ilder Pro ilder Pro Status Registers attion: 00-00-04-06-2C-CE Refresh ID Registers	Frequency Adjust ZC Address 0x6C / 108c; 400 kHz; 33V Field Programmer Family: \$338x/4x (not firmware based) Target: Wired to PC8 (serial Interface: \$3' + Wire; 12 MHz 33' Part Number: #68: Design ID: TTTTTTT Config Scan Control Register Soft Reset and calibration SOFTRESET OSPLIA_SOFTRESET OSPLIA_SOFTRE

Figure 7.13. I2C and SPI Communication Failure Examples

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.

nfo DUT Register Editor		
	Status Registers -	Field Programmer
Register Peek/Poke Herr Address: 0:0943 # Bytes: 1 Unsigned bit	Decimal 2271 Read Write 2 1 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2	Panhy: 5323/26 (od) Target: Scotet, CPN4 Scotet Force: Grow South CPN4 Scotet Force: Grow South CPN4 Scotet Force: Grow South CPN4 Part Number: SS342.64 - COM Design ID: SS432.04 - COM Design ID: SS43
		FDEC
S.		FDEC
a Rered • Auto Scroll	On Theref Mader Clear Copy to Opboard Face	FDEC
tered 🚺 Auto Scrol	Cin 🖥 [Faert Mahar] Ciezy to Capoter of Pacce	FDEC
) tered a Auto Scrol nestamp Scurce 23342.216 EV8	On Disent Maker Cear Copy to Opboard Pane Array Copy to Opboard Pane Array Copy to Opboard Copy of Opboard Pane Array Copy of Opb	FDEC
0 Auto Scrol mestings Sourco R132215 EV8 R1312226 EV8	Cin	FDEC
Auto Scroll nectump: Source 21342.216 EV8 21342.216 EV8	On Exect Maker Cear Copy to Diploand Pacer Annop State DUD _ Syncholsees(=:0001) > State Distance Annop State DUD _ Syncholsees(=:0001) > annop State DUD _ Sy	FDEC
Itered Auto Scoll mestamp Sourco E11542216 EV8 E11542220 EV8 E11542220 EV8 E11542220 EV8	Con E Shert Marker Clear Copy to Clyboard Pauce	FOEC
9 Rered Auto Scroll RIS12216 UV8 RIS12216 UV8 RIS12226 EV8 RIS12228 EV8 RIS12222 EV8 RIS12222 EV8	Cn	FORC
9 Rered Auto Scrol mestamp. Science RE1542216 EV8 RE154220 EV8 RE154220 EV8 RE1542228 EV8 RE1542228 EV8 RE1542384 EV8	Con E Sheet Marker Coar Corpto Cipboard Pause Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menane Menan	FDEC.

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

3. Write your new plan to the device.

le Help		
Write Projec	t File to Device	
Write Setting Write Regist Create DUT	gs File to Device er File to Device Dump File for Silicon Labs Support	
Preferences		
Exit		
Unsigned In Hex: Binary:	t: 1 : 0x01 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0	
Binary:	(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		
1 1 0 1 0 A1 A0		
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.		
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.		
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.		

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.

licon Laboratories	
f Silicon Laboratories IDE	Travis Turner
CBPro Project File Inspector	
ClockBuilder Pro	Documents
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	D · 10·1
😽 Uninstall ClockBuilder Pro	Devices and Printers
🐌 Misc 🔤	Default Programs
CB CBPro Internal Tools	beldaler rögräfils
🛯 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	-
d Davis	
1 DdCK	
Search amorane and files	
search programs and files	Logon

CB Si538x/4x I2C Address Burn To	ol 🗆 🗖 📈
This tool can be used to change the base I2C address on a supported Si538x/4x device. A new I2C address can only be burned one time. At least once, use the test button to change the address and verify the address change without actually performing the burn.	
Field Programmer Mode:	Kit socket board Wired to your own board, I2C
I/O Voltage	3.3 V V
Address Pin State:	A0 A1
Base I2C Address to Program:	0x74 Clear Copy to Clipboard
Detect Test (Det	ect & Change Volatile) Burn (Detect, Change Volatile, & Burn NVM)

Figure 7.17. I2C Address Burn Tool



Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific attribution Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for althorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Trademark Information

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga®, Bluegiga®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLaido®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, ISOmodem@, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



Silicon Laboratories Inc. 400 West Cesar Chavez Austin, TX 78701 USA

http://www.silabs.com