Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

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 Si538x4x Field Programmer board and ribbon cable to easily connect to the USB to SPI/I2C adapter.
   - NVM programming of Si538x/4x devices already mounted on a system PCB. Users are encouraged to include a standard 10-pin header on their PCB to allow the Si538x4x 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.).

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

Table .1. Supported Programming Models

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

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.

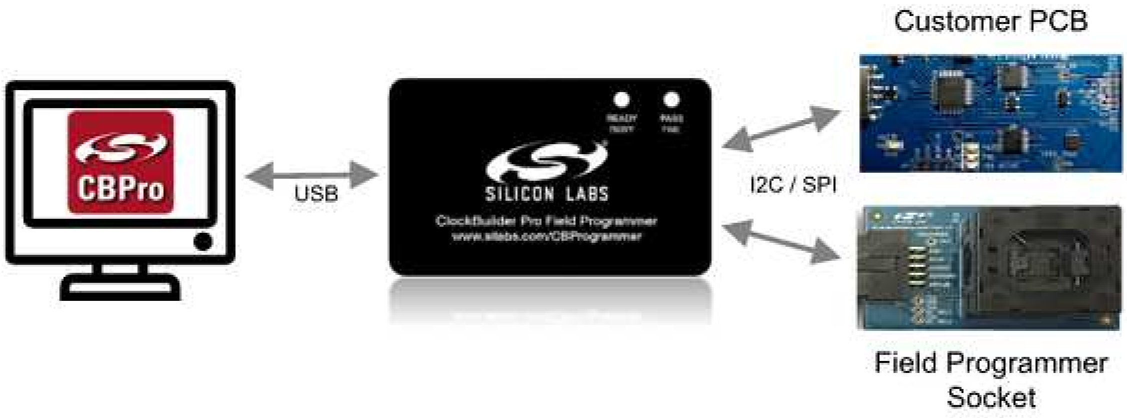


Figure 1.1. Example Hardware Configuration (Using QFN Socket Board or Customer PCB)

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. 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.

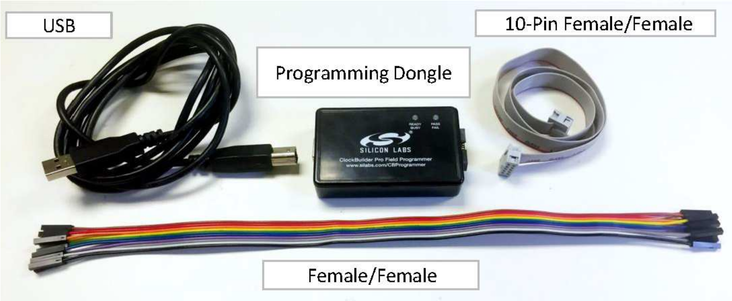


Figure 1.2. CBPROG-DONGLE Kit Contents

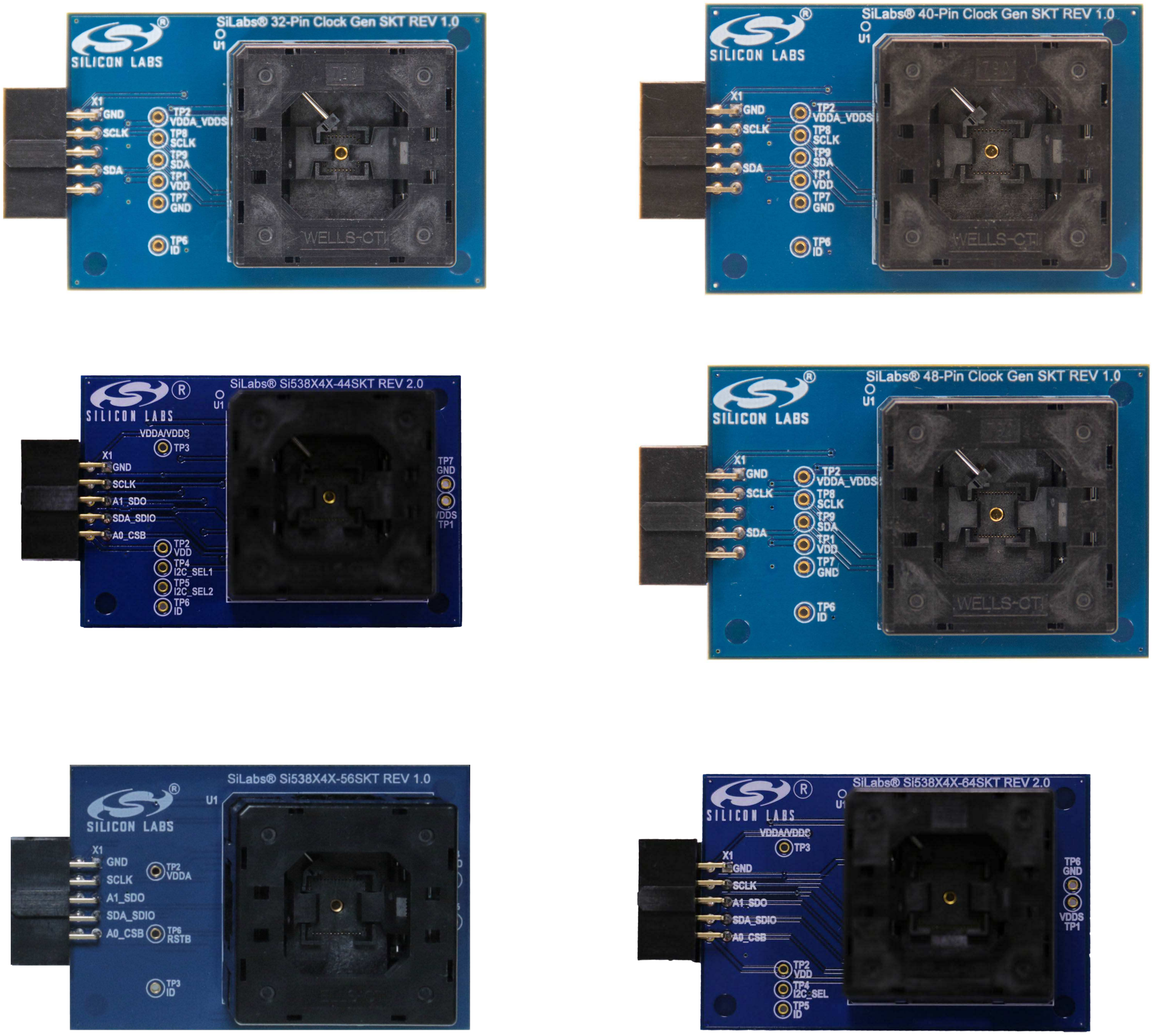

SiLabs® 32-Pin Clock Gen SKT REV 1.0
SiLabs® 40-Pin Clock Gen SKT REV 1.0
SiLabs® Si538X4X-44SKT REV 2.0
SiLabs® 48-Pin Clock Gen SKT REV 1.0
SiLabs® Si538X4X-56SKT REV 1.0
SiLabs® Si538X4X-64SKT REV 2.0
SILICON LABS
X1
GND
SCLK
SDA
A1_SDO
SDA_SDIO
A0_CSB
TP2 VDDA_VDDS
TP8 SCLK
TP9 SDA
TP1 VDD
TP7 GND
TP6 ID
U1
WELLS-CTI

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, or Si538x 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-44SKT, 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.

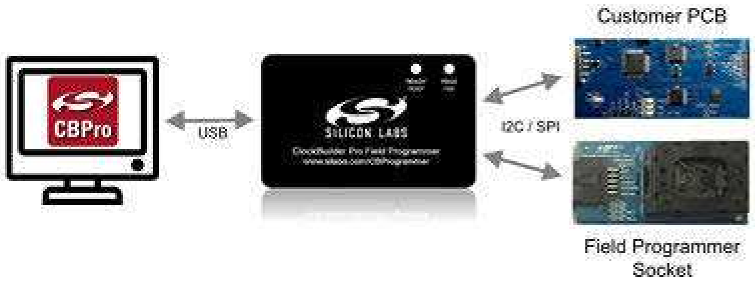


Figure 4.1. PC to CBPROG-DONGLE Connection

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.

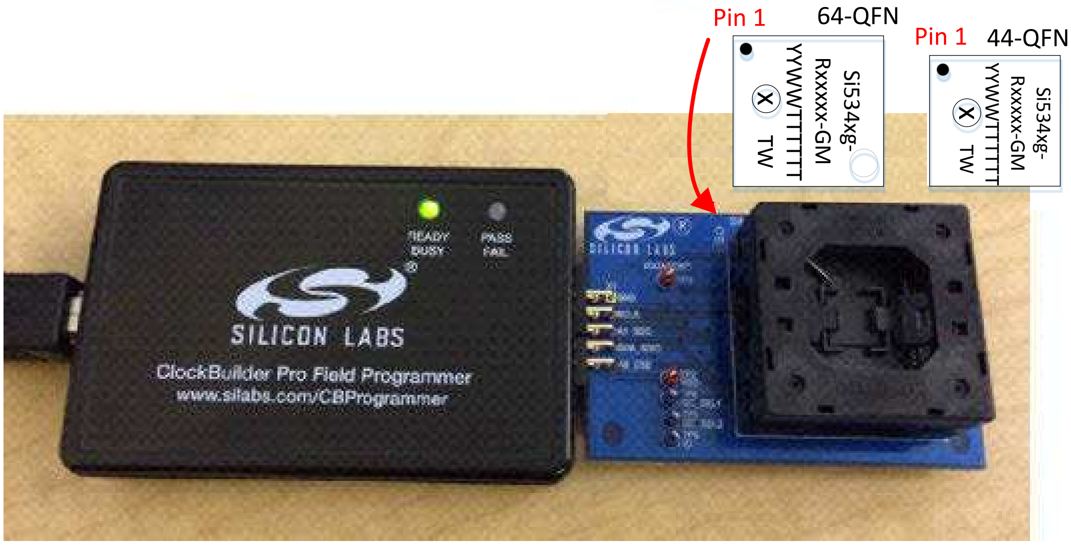


Figure 4.2. Correct Orientation of a Device in the Socket

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.

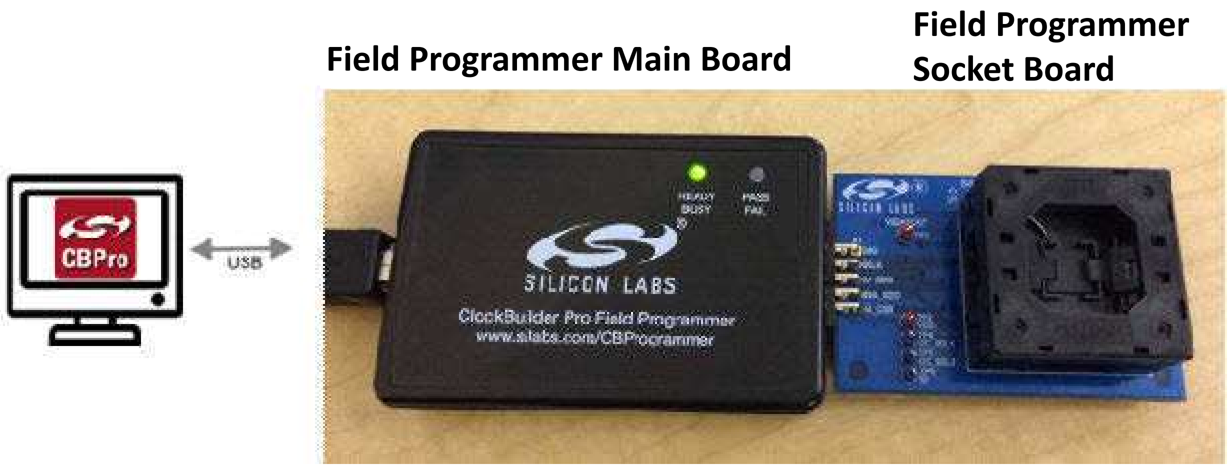
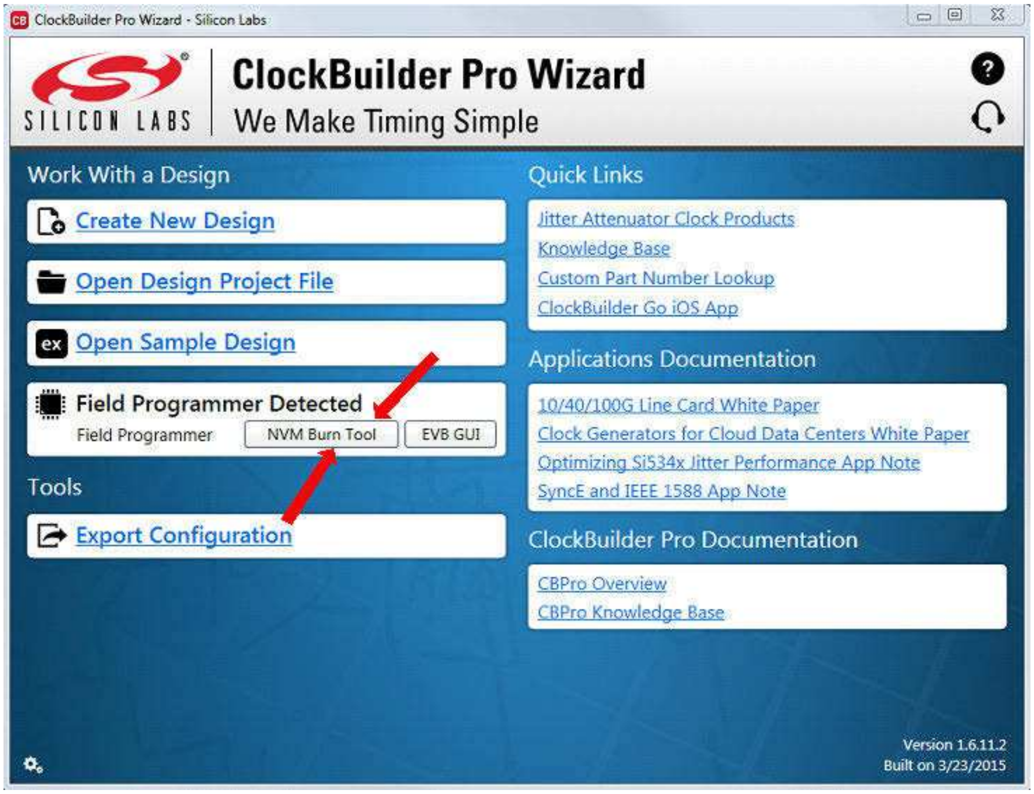

Field Programmer Main Board
Field Programmer Socket Board
CBPro
USB
READY BUSY
PASS FAIL
SILICON LABS
ClockBuilder Pro Field Programmer
www.silabs.com/CBProgrammer

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.***



ClockBuilder Pro Wizard - Silicon Labs

SILICON LABS

ClockBuilder Pro Wizard

We Make Timing Simple

Work With a Design

Create New Design

Open Design Project File

Open Sample Design

Field Programmer Detected

Field Programmer [NVM Burn Tool] [EVB GUI]

Tools

Export Configuration

Quick Links

Jitter Attenuator Clock Products

Knowledge Base

Custom Part Number Lookup

ClockBuilder Go iOS App

Applications Documentation

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

CBPro Overview

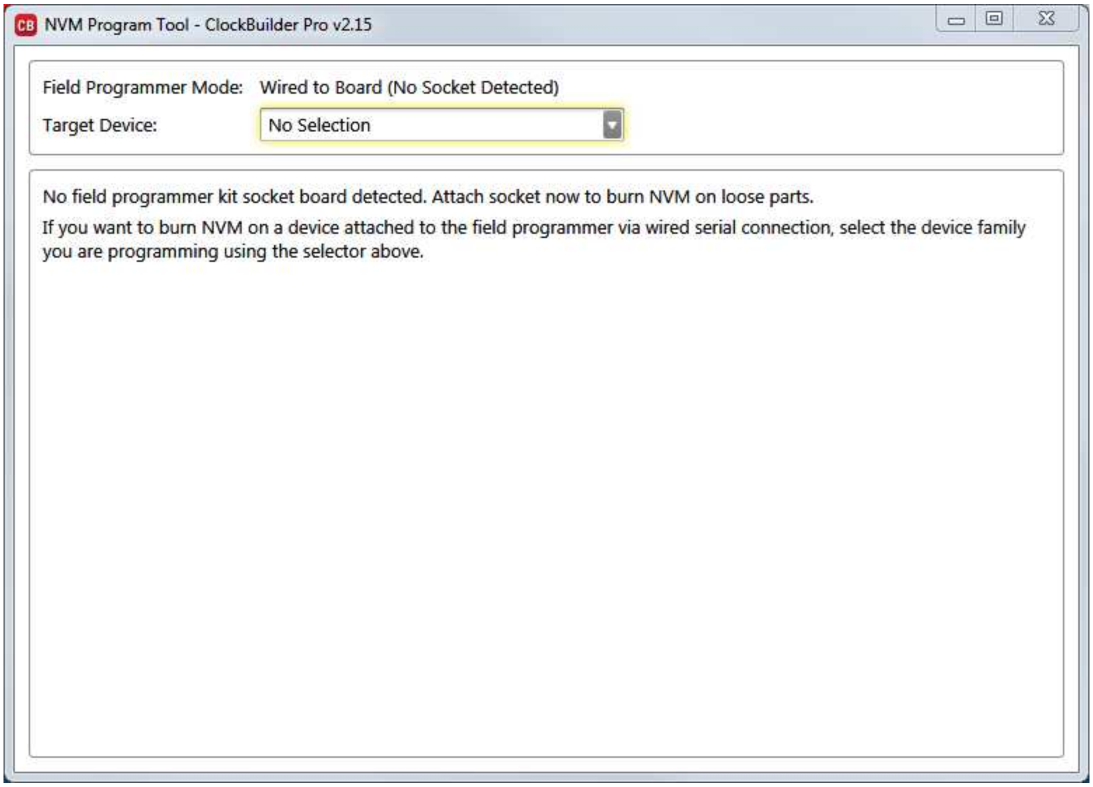
CBPro Knowledge Base

Version 1.6.11.2

Built on 3/23/2015

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:



NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Wired to Board (No Socket Detected)

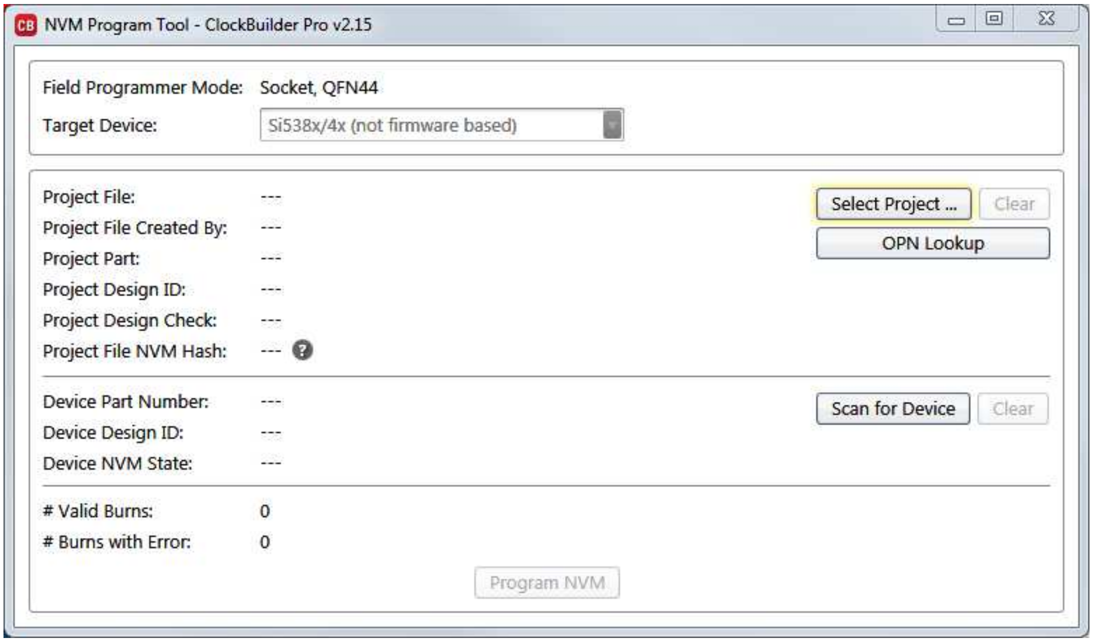
Target Device: No Selection

No field programmer kit socket board detected. Attach socket now to burn NVM on loose parts.

If you want to burn NVM on a device attached to the field programmer via wired serial connection, select the device family you are programming using the selector above.

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 - ClockBuilder Pro v2.15

Field Programmer Mode: Socket, QFN44

Target Device: Si538x/4x (not firmware based)

Project File: ---

Project File Created By: ---

Project Part: ---

Project Design ID: ---

Project Design Check: ---

Project File NVM Hash: ---

Device Part Number: ---

Device Design ID: ---

Device NVM State: ---

Valid Burns: 0

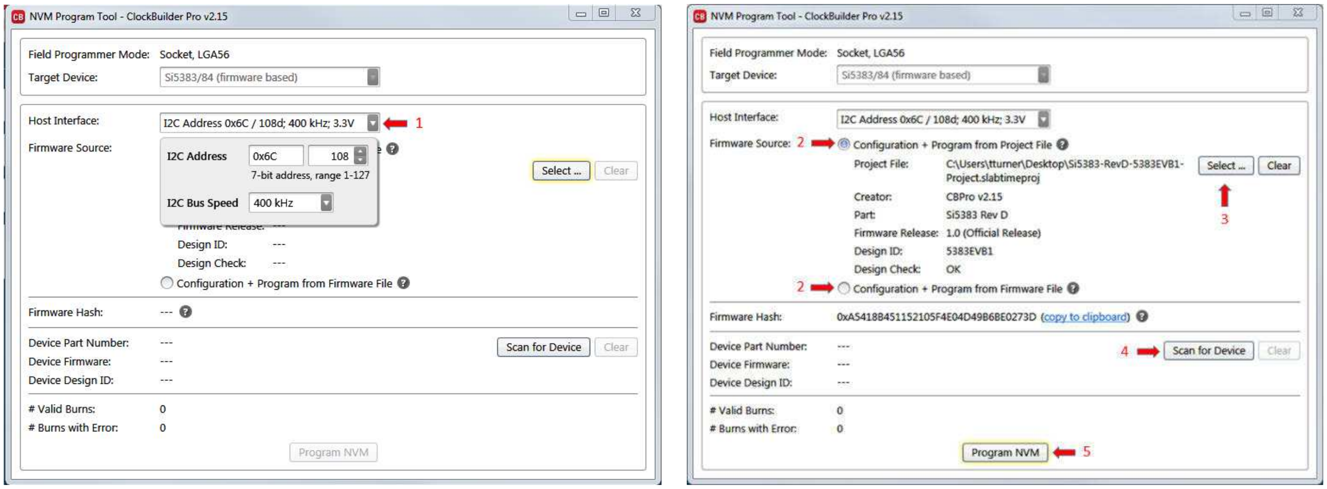
Burns with Error: 0

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.



NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Socket, LGA56
Target Device: Si5383/84 (firmware based)

Host Interface: I2C Address 0x6C / 108d; 400 kHz; 3.3V
Firmware Source:
I2C Address: 0x6C 108
7-bit address, range 1-127
I2C Bus Speed: 400 kHz
Design ID: ---
Design Check: ---
○ Configuration + Program from Firmware File

Firmware Hash: ---

Device Part Number: ---
Device Firmware: ---
Device Design ID: ---

Valid Burns: 0
Burns with Error: 0

NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Socket, LGA56
Target Device: Si5383/84 (firmware based)

Host Interface: I2C Address 0x6C / 108d; 400 kHz; 3.3V
Firmware Source: ● Configuration + Program from Project File
Project File: C:\Users\tturner\Desktop\Si5383-RevD-5383EVB1-Project.slabtimeproj
Creator: CBPro v2.15
Part: Si5383 Rev D
Firmware Release: 1.0 (Official Release)
Design ID: 5383EVB1
Design Check: OK
○ Configuration + Program from Firmware File

Firmware Hash: 0xA5418B451152105F4E04D49B6BE0273D (copy to clipboard)

Device Part Number: ---
Device Firmware: ---
Device Design ID: ---

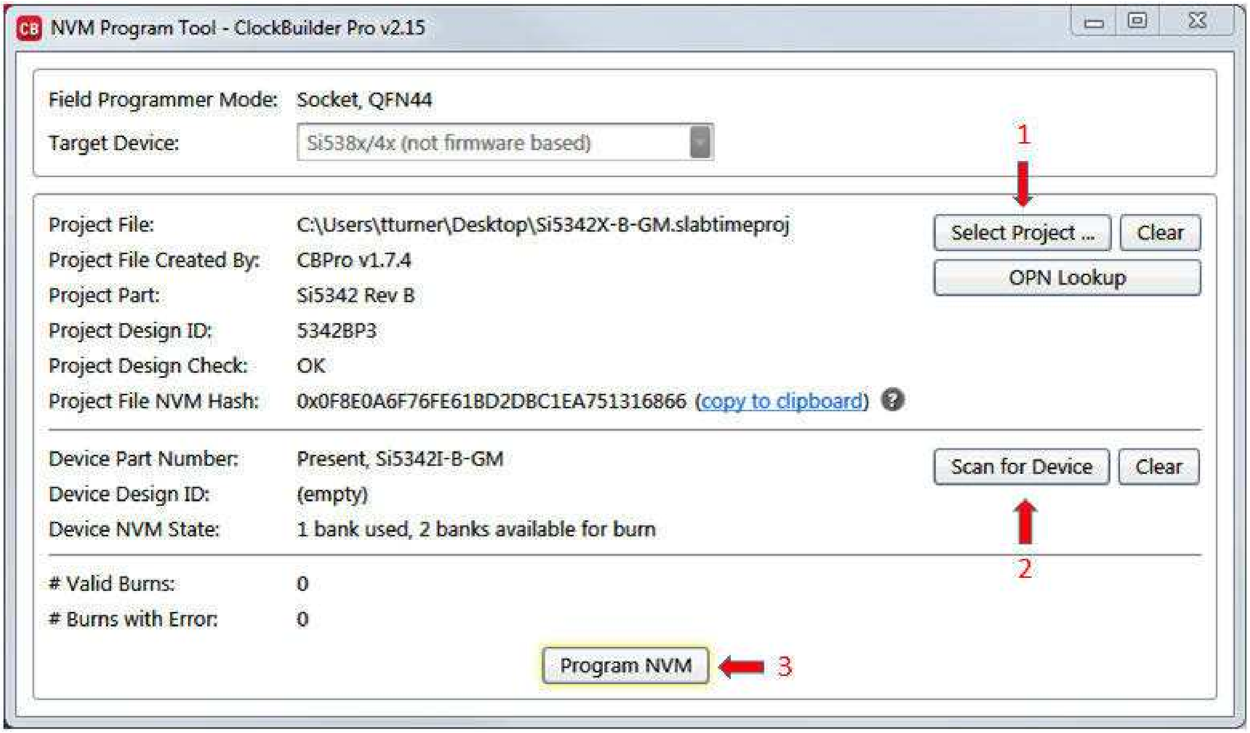
Valid Burns: 0
Burns with Error: 0

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.



NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Socket, QFN44
Target Device: Si538x/4x (not firmware based)

Project File: C:\Users\tturner\Desktop\Si5342X-B-GM.slabtimeproj
Project File Created By: CBPro v1.7.4
Project Part: Si5342 Rev B
Project Design ID: 5342BP3
Project Design Check: OK
Project File NVM Hash: 0x0F8E0A6F76FE61BD2DBC1EA751316866 (copy to clipboard)

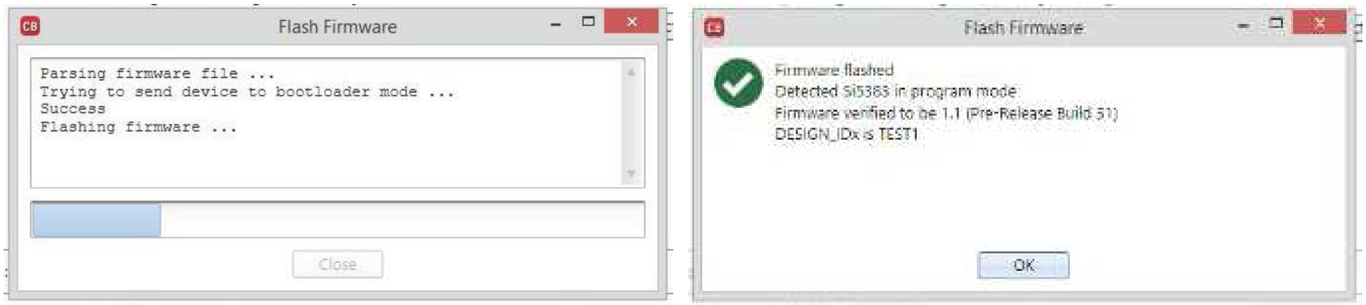
Device Part Number: Present, Si5342I-B-GM
Device Design ID: (empty)
Device NVM State: 1 bank used, 2 banks available for burn

Valid Burns: 0
Burns with Error: 0

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

4.1.3 In-Socket Programming Status

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



Flash Firmware

```
Parsing firmware file ...
Trying to send device to bootloader mode ...
Success
Flashing firmware ...
```

Flash Firmware

Firmware flashed
Detected Si5383 in program mode
Firmware verified to be 1.1 (Pre-Release Build 51)
DESIGN_IDx is TEST1

Figure 4.10. In-Socket Programming Status

4.2 In-System Firmware / NVM Programming

This workflow describes the process of programming a device mounted on a PCB. For Si538x/4x (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.

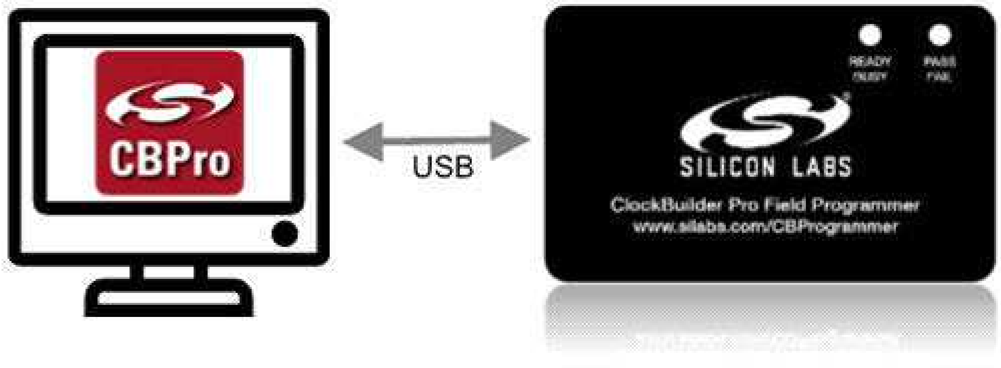
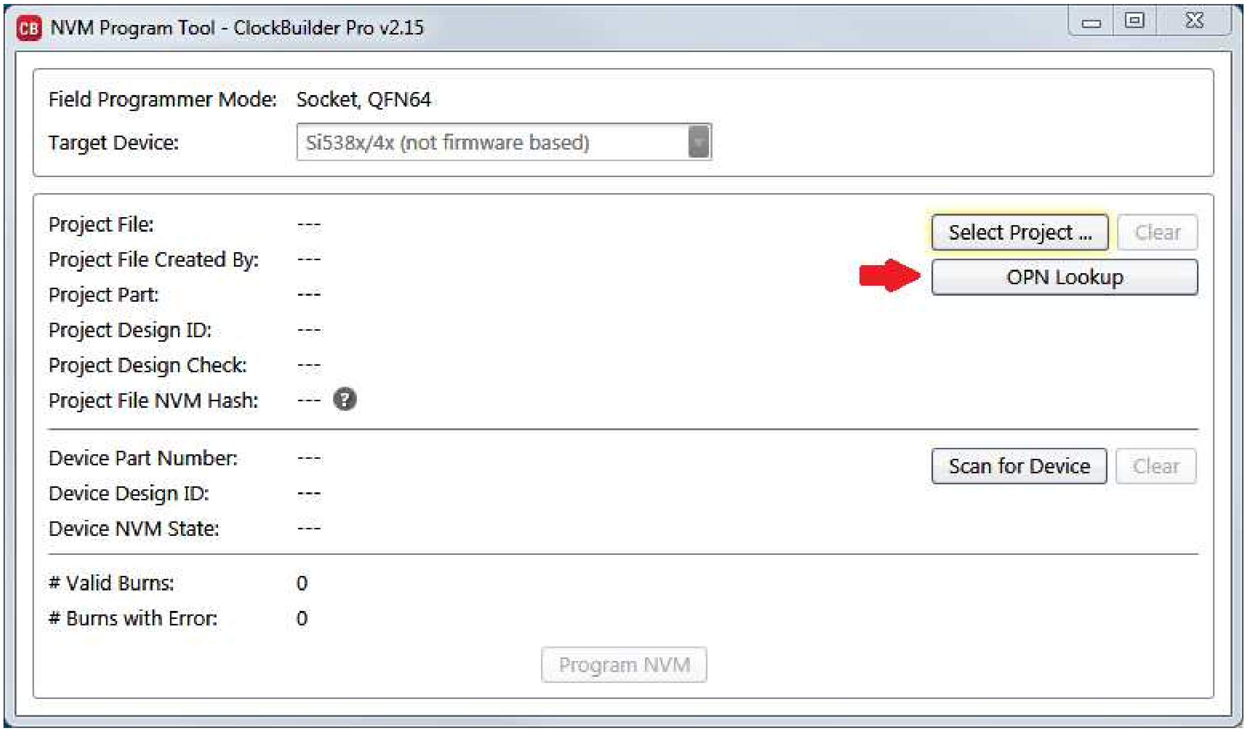

CBPro
USB
READY
BUSY
PASS
FAIL
SILICON LABS
ClockBuilder Pro Field Programmer
www.silabs.com/CBProgrammer

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 12.



NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Socket, QFN64
Target Device: Si538x/4x (not firmware based)

Project File: ---
Project File Created By: ---
Project Part: ---
Project Design ID: ---
Project Design Check: ---
Project File NVM Hash: ---

Device Part Number: ---
Device Design ID: ---
Device NVM State: ---

Valid Burns: 0
Burns with Error: 0

Figure 4.12. OPN Lookup Option

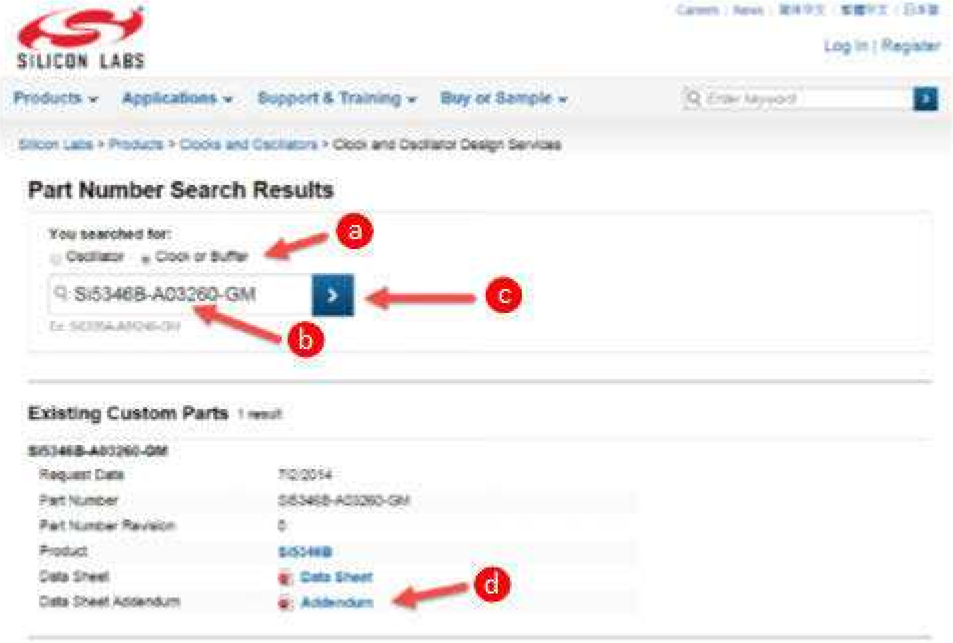
If you have a custom OPN mounted on your board (a part number with a 5 digit code in the middle of the part number, such as Si5346B-A03260-GM), you should look up the host I/O setting (located at address of 0x0943) by selecting the OPN Lookup option. A browser will open and you will then enter in your 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
Careers | News | 简体中文 | 繁體中文 | 日本語
Log In | Register
Products
Applications
Support & Training
Buy or Sample
Enter keyword
Silicon Labs > Products > Clocks and Oscillators > Clock and Oscillator Design Services
a
b
c
d

Part Number Search Results

You searched for:
○ Oscillator ● Clock or Buffer

Si5346B-A03260-GM

Ex: Si5335A-A00148-GM

Existing Custom Parts 1 result

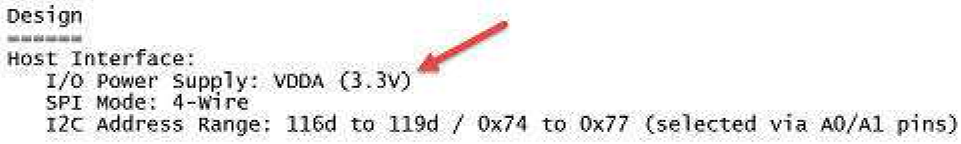
Si5346B-A03260-GM

Request Date	7/2/2014
Part Number	Si5346B-A03260-GM
Part Number Revision	0
Product	Si5346B
Data Sheet	Data Sheet
Data Sheet Addendum	Addendum

Figure 4.13. OPN Lookup

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 I2C/SPI interface will operate from a 1.8 V supply.
   “VDDA (3.3 V)” indicates the I/O supply for the Si534x-8x 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 13 shows an example data sheet addendum showing VDDA (3.3 V).



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

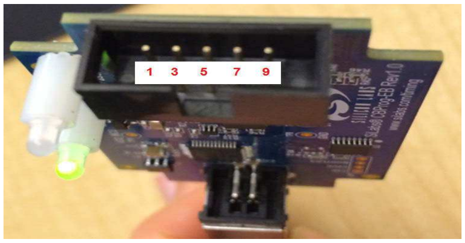


Figure 4.15. Interface Pins on Header (Front View of CBPROG-DONGLE)

Table 4.1. Interface Pin Connections from CBPROG-DONGLE

Pin #	Description	Wire to Your PCB?	I^2C	4-wire SPI	3-wire SPI
9	A0_CSB (applies only for Si534x-8x device)	3- or 4-Wire SPI	Can be used to set I^2C address bit A0 high or low. Routed to A0 device pin on the programming Field Programmer Socket Boards.	Drives the chip select signal during SPI transactions	Drives the chip select signal during SPI transactions
10	VDD	Never	Supplies the Core VDD voltage to the device when using a programming Field Programmer Socket Board. Do not use this pin for in-system programming.	Supplies the Core VDD voltage to the device when using a programming Field Programmer Socket Board. Do not use this pin for in-system programming.	Supplies the core VDD voltage to the device when using a programming Field Programmer Socket Board. Do not use this pin for in-system programming.
7	SDA_SDIO	Always	Serial data signal for I^2C transactions.	Serial data out to device for 4-wire SPI transactions (MOSI).	Bidirectional Serial data for 3-wire SPI transactions (SDIO).
8	I2C_SEL1 (applies only for Si534x-8x device)	Never	Used to set I2C_SEL signal high to set the device for I^2C communication. (Refer to specific part pinout and the programming Field Programmer Socket Board to determine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL signal low for SPI communication. (Refer to specific part pinout and the programming Field Programmer Socket Board to determine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL signal low for SPI communication. (Refer to specific part pinout and the programming Field Programmer Socket Board to determine whether to use I2C_SEL1 or I2C_SEL2)
5	A1_CSB (applies only for Si534x-8x 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 transactions (MISO).	Not used
6	I2C_SEL2 (applies only for Si534x-8x device)	Never	Used to set I2C_SEL signal high to set the device for I2C communication. (Refer to specific part pinout and the programming Field Programmer Socket Board to determine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL signal low for SPI communication. (Refer to specific part pinout and the programming Field Programmer Socket Board to determine whether to use I2C_SEL1 or I2C_SEL2)	Used to put I2C_SEL signal low for SPI communication. (Refer to specific part pinout and the programming Field Programmer Socket Board to determine 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 device when using a programming Field Programmer Socket Board. Do not use this pin for in-system programming.	Supplies the VDDA and VDDS voltages to the device when using a programming Field Programmer Socket Board. Do not use this pin for in-system programming.	Supplies the VDDA and VDDS voltages to the device when using a programming Field Programmer Socket Board. Do not use this pin for in-system programming.
1	GND	Always	GND	GND	GND

Pin #	Description	Wire to Your PCB?	I^2C	4-wire SPI	3-wire SPI
2	ID	Never	The programming Field Programmer Socket Boards provide a voltage on this pin to identify the board. For in-system programming, this pin should be grounded or not connected to any signal.	The programming Field Programmer Socket Boards provide a voltage on this pin to identify the board. For in-system programming, this pin should be grounded or not connected to any signal.	The programming Field Programmer Socket Boards provide a voltage on this pin to identify the board. For in-system programming, this pin should be grounded or not connected to any signal.

4.2.1 I^2C Hardware Configuration

For I^2C 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 DEVICE

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

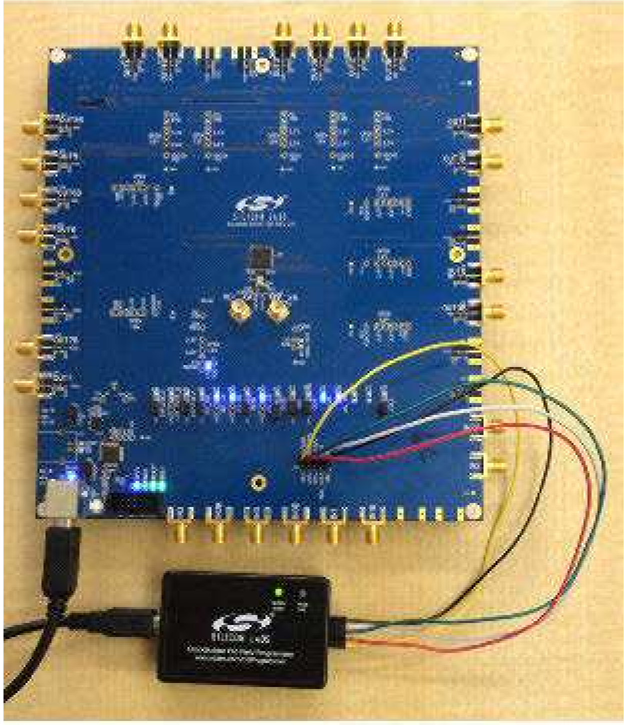


Figure 4.16. Example I2C Connection to External System Target Board Using Jumper Wires (Si5346-EVB)

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 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 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:

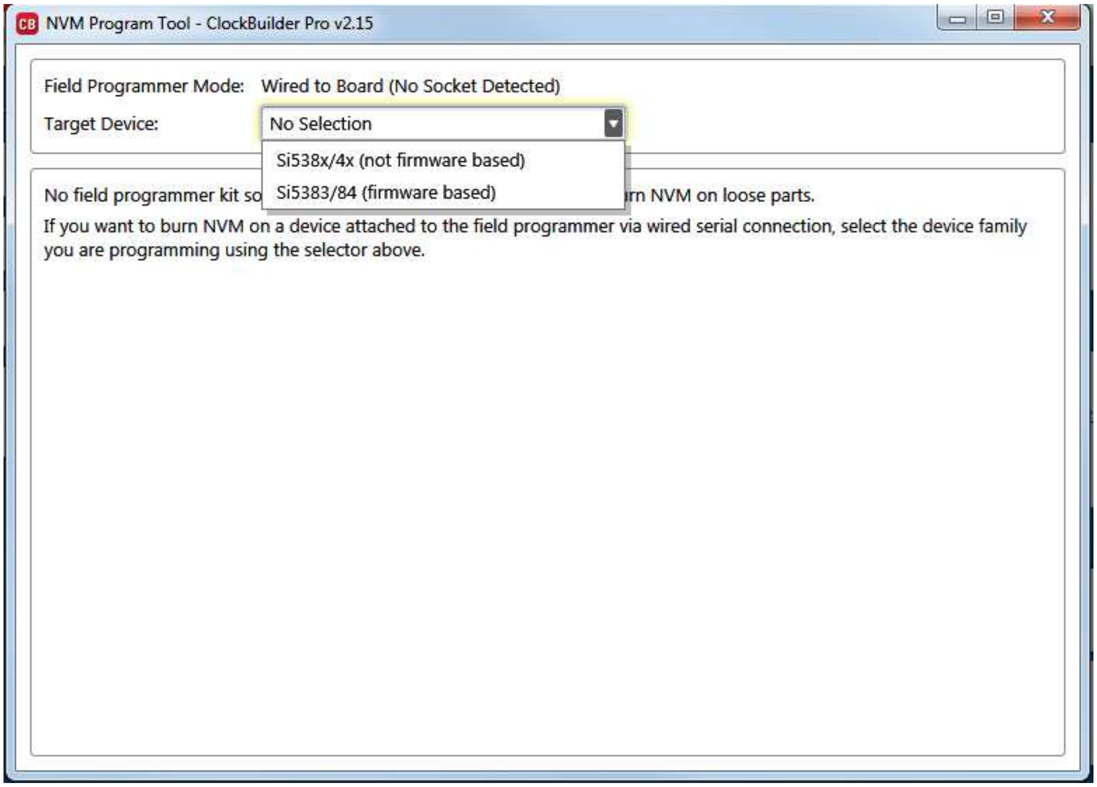

CB NVM Program Tool - ClockBuilder Pro v2.15
Field Programmer Mode: Wired to Board (No Socket Detected)
Target Device: No Selection
Si538x/4x (not firmware based)
Si5383/84 (firmware based)
No field programmer kit so rn NVM on loose parts.
If you want to burn NVM on a device attached to the field programmer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

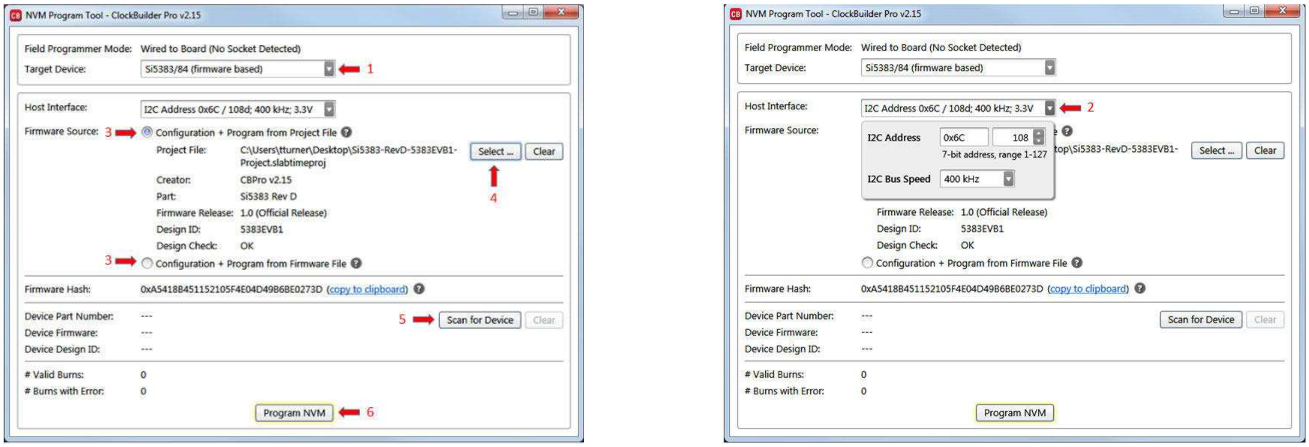
Figure 4.17. NVM Program Tool, Select Device Family

4.2.4 Programming In-system, Firmware Based Devices

Refer to Figure 4.18 Programming In-system, Firmware Based Devices on page 17 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 I^{2}C 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 I^{2}C 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.



NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Wired to Board (No Socket Detected)
Target Device: Si5383/84 (firmware based)
Host Interface: I2C Address 0x6C / 108d; 400 kHz; 3.3V
Firmware Source: ● Configuration + Program from Project File
Project File: C:\Users\tturner\Desktop\Si5383-RevD-5383EVB1-Project.slabtimeproj
Creator: CBPro v2.15
Part: Si5383 Rev D
Firmware Release: 1.0 (Official Release)
Design ID: 5383EVB1
Design Check: OK
○ Configuration + Program from Firmware File
Firmware Hash: 0xA5418B451152105F4E04D49B6BE0273D (copy to clipboard)
Device Part Number: ---
Device Firmware: ---
Device Design ID: ---
Valid Burns: 0
Burns with Error: 0

NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Wired to Board (No Socket Detected)
Target Device: Si5383/84 (firmware based)
Host Interface: I2C Address 0x6C / 108d; 400 kHz; 3.3V
Firmware Source:
I2C Address: 0x6C 108
7-bit address, range 1-127
I2C Bus Speed: 400 kHz
top\Si5383-RevD-5383EVB1-
Firmware Release: 1.0 (Official Release)
Design ID: 5383EVB1
Design Check: OK
○ Configuration + Program from Firmware File
Firmware Hash: 0xA5418B451152105F4E04D49B6BE0273D (copy to clipboard)
Device Part Number: ---
Device Firmware: ---
Device Design ID: ---
Valid Burns: 0
Burns with Error: 0

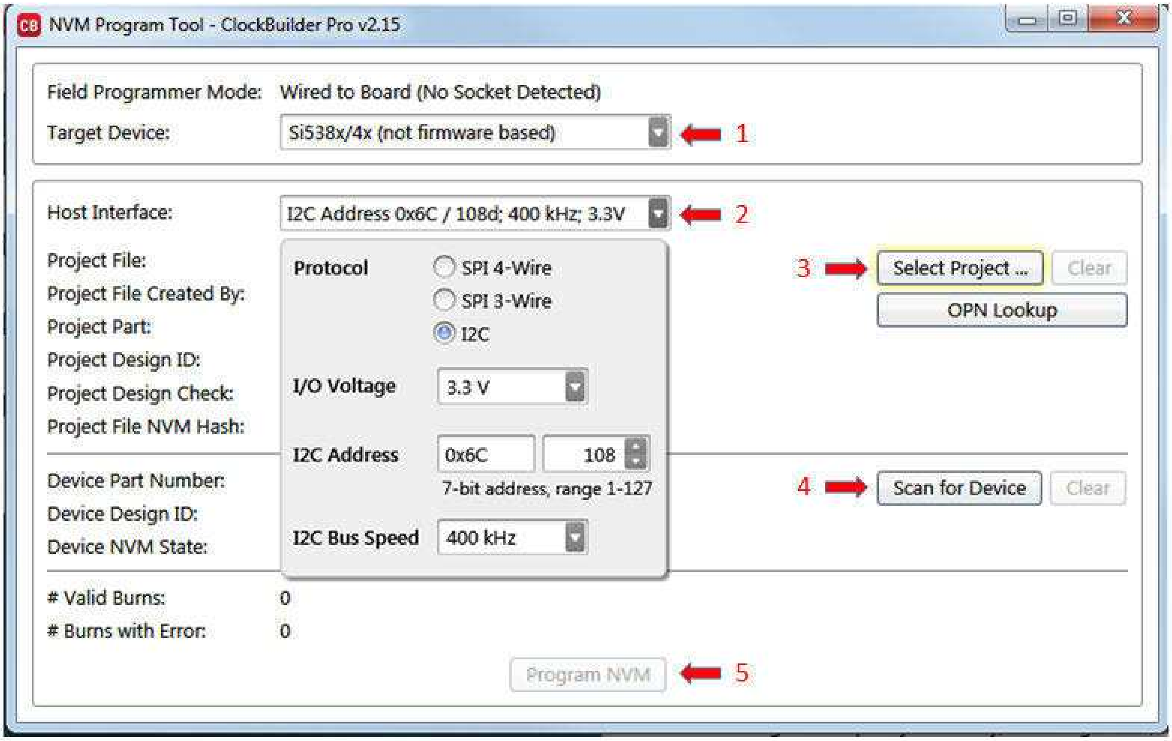
Figure 4.18. Programming In-system, Firmware Based Devices

4.2.5 Programming In-system, Non-firmware Based Devices

Refer to Figure 4.19 Programming In-system, Non-firmware Based Devices on page 18 below.

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

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



NVM Program Tool - ClockBuilder Pro v2.15

Field Programmer Mode: Wired to Board (No Socket Detected)
Target Device: Si538x/4x (not firmware based)

Host Interface: I2C Address 0x6C / 108d; 400 kHz; 3.3V

Project File:
Project File Created By:
Project Part:
Project Design ID:
Project Design Check:
Project File NVM Hash:

Protocol: ○ SPI 4-Wire ○ SPI 3-Wire ● I2C
I/O Voltage: 3.3 V
I2C Address: 0x6C 108
7-bit address, range 1-127
I2C Bus Speed: 400 kHz

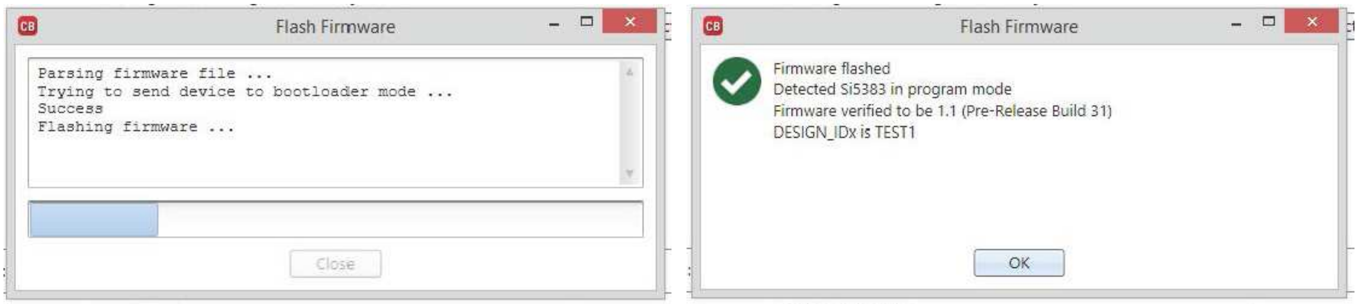
Device Part Number:
Device Design ID:
Device NVM State:

Valid Burns: 0
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:



```
Parsing firmware file ...
Trying to send device to bootloader mode ...
Success
Flashing firmware ...
```

Firmware flashed
Detected Si5383 in program mode
Firmware verified to be 1.1 (Pre-Release Build 31)
DESIGN_IDx is TEST1

Figure 4.20. Programming Status

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.

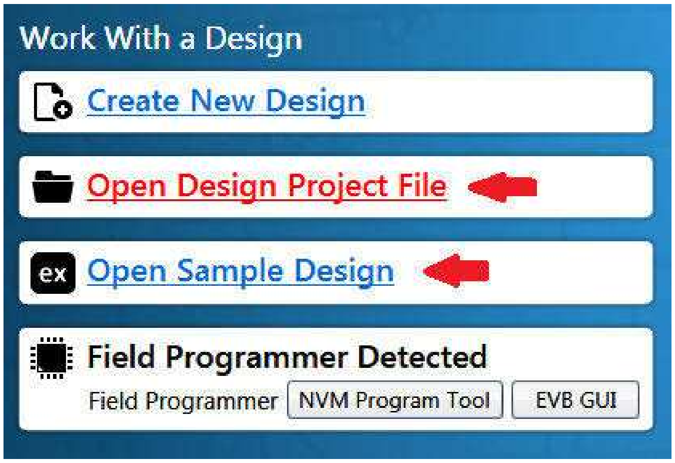
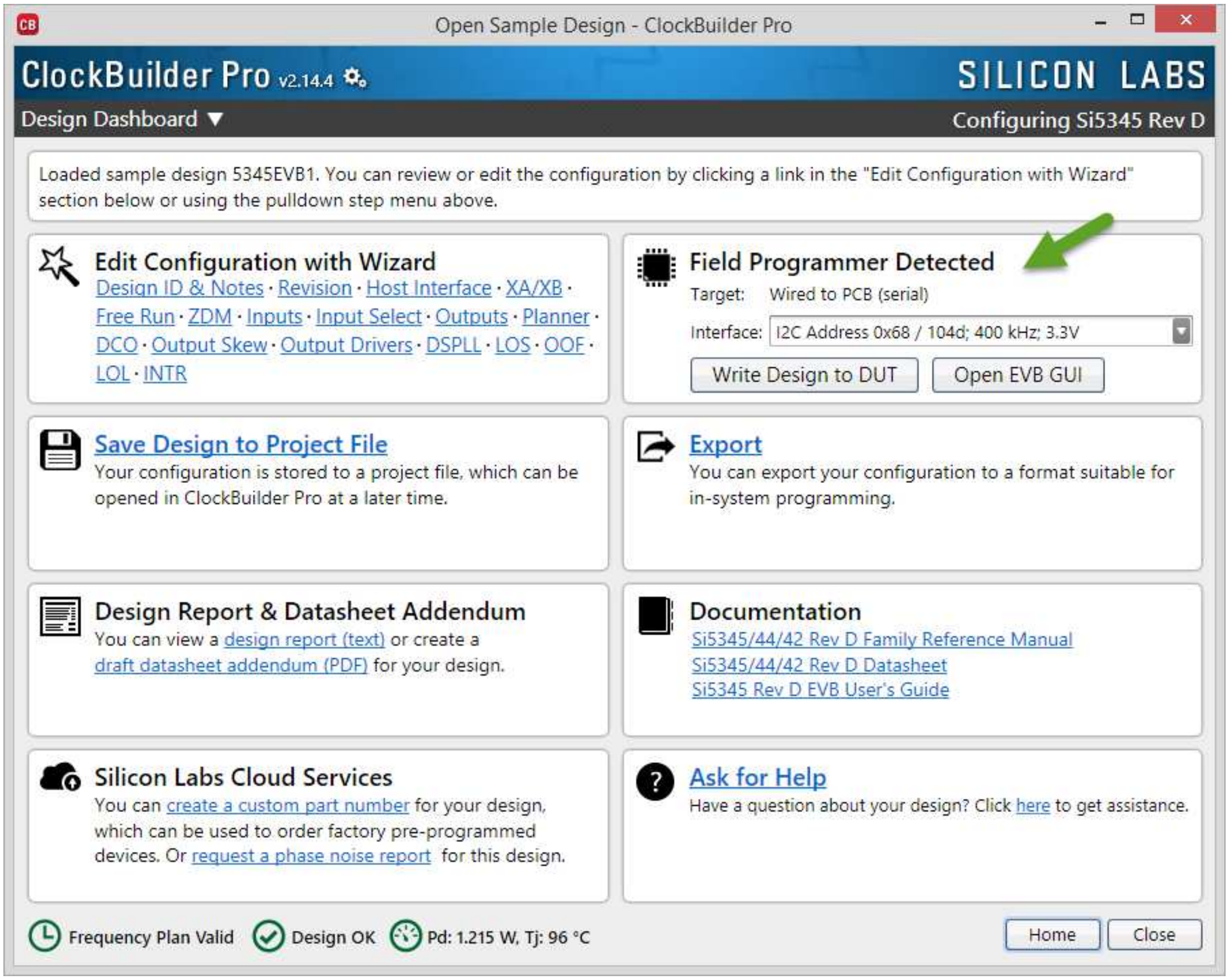


Figure 4.21. Open Design Project File, and see Field Programmer Detected

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:



Open Sample Design - 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 configuration by clicking a link in the "Edit Configuration with Wizard" section below or using the pulldown step menu above.

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

Save Design to Project File

Your configuration is stored to a project file, which can be opened in ClockBuilder Pro at a later time.

Export

You can export your configuration to a format suitable for in-system programming.

Design Report & Datasheet Addendum

You can view a design report (text) or create a draft datasheet addendum (PDF) 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

Silicon Labs Cloud Services

You can create a custom part number for your design, which can be used to order factory pre-programmed devices. Or request a phase noise report for this design.

Ask for Help

Have a question about your design? Click here to get assistance.

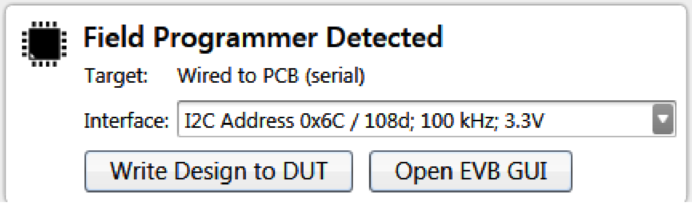
Frequency Plan Valid · Design OK · Pd: 1.215 W, Tj: 96 °C

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.



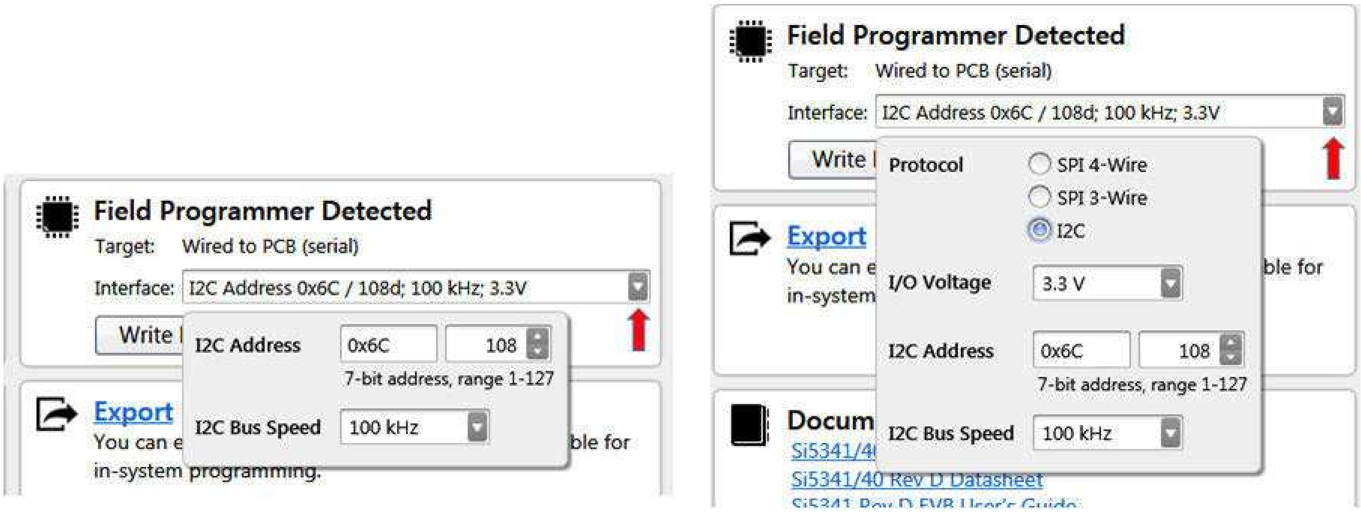
Field Programmer Detected

Target: Wired to PCB (serial)

Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V

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 Programmer Detected

Target: Wired to PCB (serial)

Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V

I2C Address: 0x6C 108

7-bit address, range 1-127

I2C Bus Speed: 100 kHz

Field Programmer Detected

Target: Wired to PCB (serial)

Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V

Protocol: ○ SPI 4-Wire ○ SPI 3-Wire ● I2C

I/O Voltage: 3.3 V

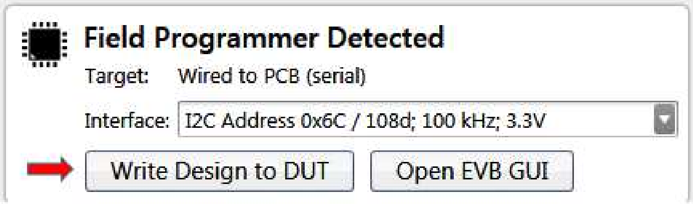
I2C Address: 0x6C 108

7-bit address, range 1-127

I2C Bus Speed: 100 kHz

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

Figure 4.25. Write Design to DUT

Or on any configuration page in the wizard:

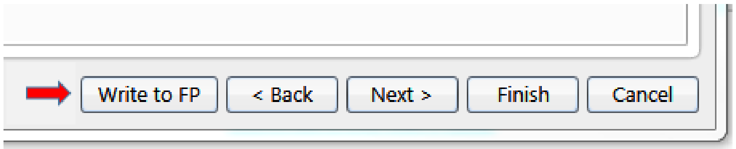
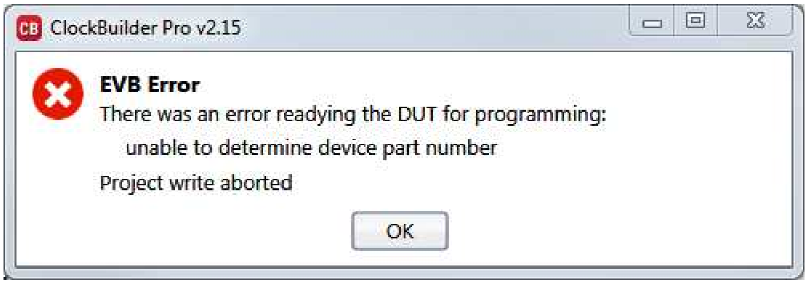


Figure 4.26. Write to FP

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 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:



EVB Error

There was an error readying the DUT for programming:

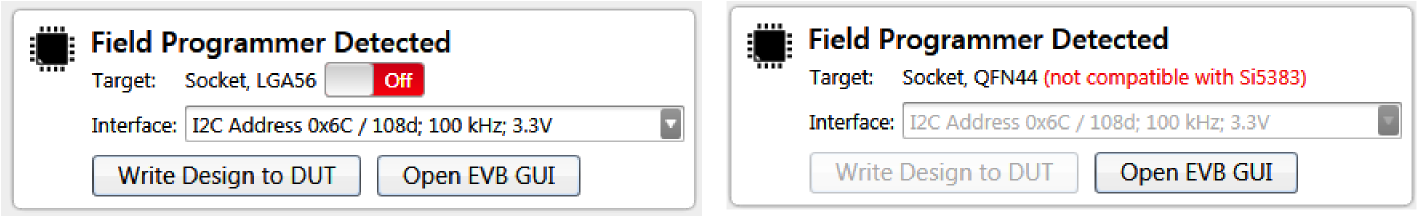
unable to determine device part number

Project write aborted

Figure 4.27. Error Message

4.3.2.2 Using the CBPro Dashboard with In-socket Devices

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



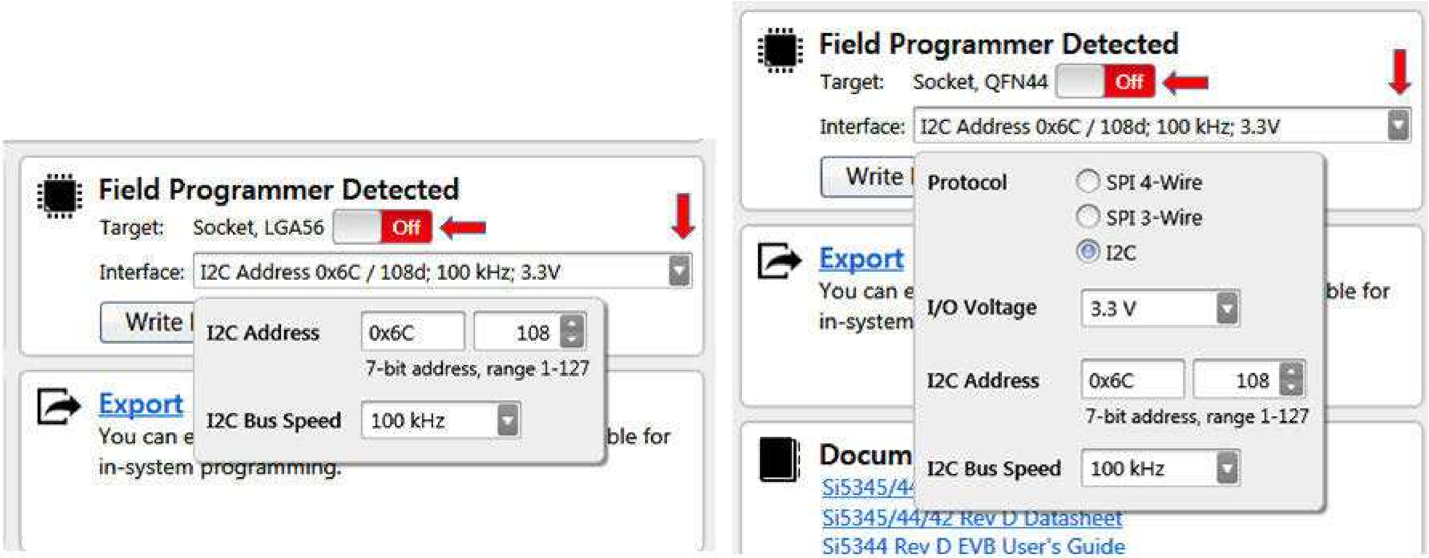
Field Programmer Detected
Target: Socket, LGA56 Off
Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V

Field Programmer Detected
Target: Socket, QFN44 (not compatible with Si5383)
Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V

Figure 4.28. Socket Compatibility

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

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

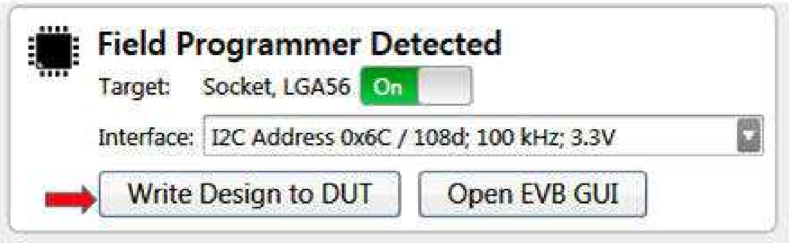


Field Programmer Detected
Target: Socket, LGA56 Off
Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V
I2C Address 0x6C 108
7-bit address, range 1-127
I2C Bus Speed 100 kHz

Field Programmer Detected
Target: Socket, QFN44 Off
Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V
Protocol ○ SPI 4-Wire ○ SPI 3-Wire ● I2C
I/O Voltage 3.3 V
I2C Address 0x6C 108
7-bit address, range 1-127
I2C Bus Speed 100 kHz

Figure 4.29. Interface Settings

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



Field Programmer Detected
Target: Socket, LGA56 On
Interface: I2C Address 0x6C / 108d; 100 kHz; 3.3V

Figure 4.30. Write Design to DUT

Or on any configuration page in the wizard:

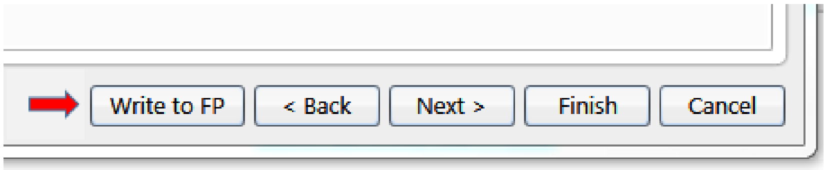


Figure 4.31. Write Design to FP

4.3.3 Launching the CBPro EVB GUI

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

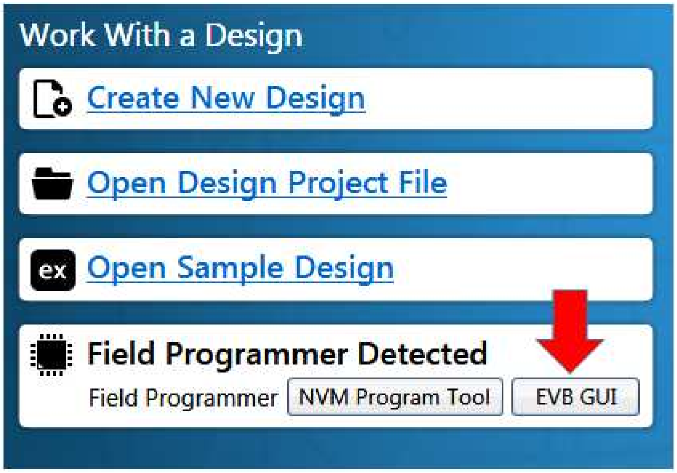
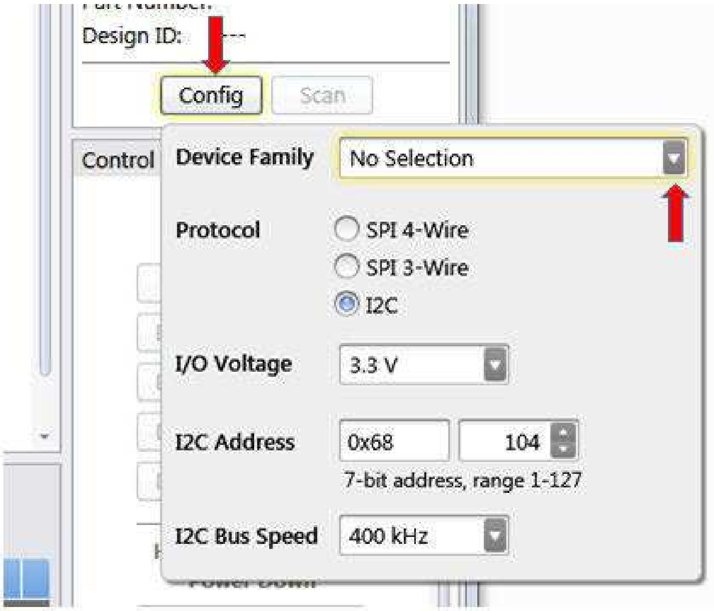


Figure 4.32. Open EVB GUI Screen

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



Device Family: No Selection

Protocol: ○ SPI 4-Wire ○ SPI 3-Wire ● I2C

I/O Voltage: 3.3 V

I2C Address: 0x68 104
7-bit address, range 1-127

I2C Bus Speed: 400 kHz

Figure 4.33. Select Device Family Prompt

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