USER GUIDE

UG286: ClockBuilder Pro Field Programmer Kit

This user guide describes how to use the ClockBuilder Field Programmer Kit (CBPROG-DONGLE) with ClockBuilder Pro (CBPro) to support several programming models.

The ClockBuilder Pro Field Programmer supports a variety of uses:

1. Real Time Status Monitoring and Device Debug
- Read status and telemetry registers and write control registers on Si5332/57, Si5338/56, Si5351, and Si534x/8x/9x devices.
- Call device API commands on Si536x, Si540x, and Si55xx devices. APIs are available for device status and control.

2. Non-Volatile Firmware/Configuration Programming
- Select devices support burning/flashing device configuration and firmware.
- Firmware can be flashed multiple times on Si5383/4/8/9.
- Firmware can be burned to Si536x, Si540x, and Si55xx devices based on available NVM space (non-PPS PLL configurations only).
- Configuration can be burned up to two times on non-firmware-based devices (Si534x/8x/9x).
- Configuration can be flashed multiple times on firmware-based Si5383/4/8/9.
- Configuration can be burned to Si536x, Si540x, and Si55xx based on available NVM space (non-PPS PLL configurations only).
- Configuration can be burned to Si5332/7 based on available NVM space.

3. Volatile Firmware/Configuration Programming
- All field programmer supported devices support writing device firmware or configuration volatily.
- CBPro projects are used to calculate configuration programming data and optionally firmware image to download. CBPro handles all pre- and post-programming tasks.
- Programming data can be written via EVB GUI and command line tools, allowing customers to prototype flow their host would use to reprogram a device in-system.
- Si5332/57, Si5338/56, Si5351, and Si534x/7x/8x/9x devices can be fully reconfigured via device register scripts that can be exported from CBPro.
- Si536x, Si540x, and Si55xx device firmware and configuration can be reprogrammed via firmware and configuration boot files that can be exported from CBPro.

Most operations can be performed by either:

1. Wiring the Field Programmer's serial interface directly to system board for "in-system" volatile or non-volatile programming. Users are encouraged to include a standard 10-pin header on their PCB to allow the Field Programmer board and ribbon cable to easily connect to the USB to SPI/I^2C adapter.

2. Placing a loose Skyworks Timing device into a dedicated field programmer socket for "in-socket" non-volatile programming (volatile is also supported but generally not useful in this scenario except for testing). Skyworks provides 32-pin, 40-pin, 44-pin, 48-pin, 64-pin, and 72-pin QFN socket adapter boards, as well as 44-pin and 64-pin LGA socket adapter boards.

1. Features

- Field Programmer kit contents
- CBPro download and installation instructions
- Hardware connections
- Usage examples for the Field Programmer
- Schematics of the Field Programmer and socket board
- Bill of materials
- Troubleshooting appendix for common issues
- Supported devices include Si5332/8, Si5350/1/6/7, Si534x/6x/8x/9x, Si540x, and Si55xx part families.

2. Kit Contents

Table 1 below shows the kit contents for the Field Programmer Kit. Several sockets are supported for monitoring or programming a loose device. The following table summarizes available sockets:

Table 1. Field Programmer Supported Devices

Supported Devices	Socket Part Number	Notes
Si5332	Si5332-32SKT-DK, Si5332-40SKT-DK, Si5332-48SKT-DK	The number of pins in the socket must match the device being programmed. The Si5332E/F/G/H embedded crystal products in 40-pin LGA and 48-pin LGA packages are currently not supported.
Si5338/56	N/A	Socket boards are available as a standalone programmer, P/N Si5338/56-PROG-EVB
Si5350/1/7	N/A	No sockets are available for this part.
Si534x/8x/9x	Si538x4x-44SKT-DK, Si538x4x-56SKT-DK, Si538x4x-64SKT-DK, Si538x4x-44LGA-SKT, and Si538x4x-64LGA-SKT	For QFN packages: Si5392/94 44-pin QFN devices require the Si538x4x-44SKT-DK. Si5395 64-pin QFN devices require the Si538x4x-64SKT-DK. For LGA packages: Si5392/94 44-pin LGA devices require the Si538x4x-44LGA-SKT. Si5395 64-pin LGA devices require the Si538x4x-64LGA-SKT.
Si536x, Si540x, Si55xx	Si55xx-72SKT-DK	The Si55xx-72SKT-DK requires an external 5V power supply.

The 32-pin, 40-pin, 44-pin, 48-pin, 56-pin, 64-pin, and 72-pin sockets pictured in the following pages are available separately as part numbers. The ClockBuilder Pro Field Programmer resources including schematics, layout files, and BOM can be found on the Skyworks ClockBuilder Pro Field Programmer page. Note that the sockets are sold as separate kits.

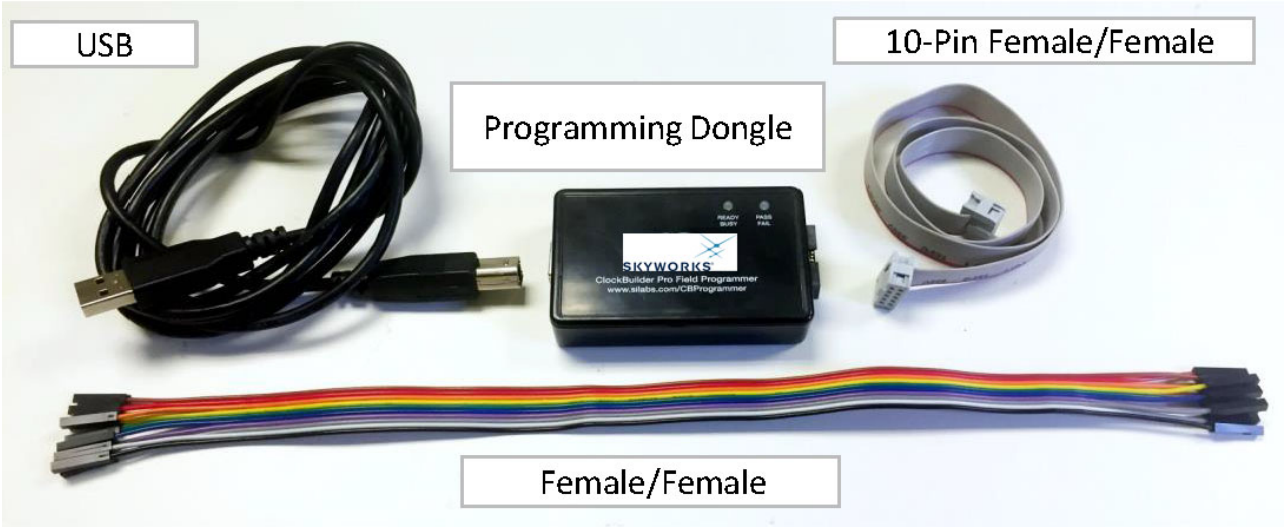


Figure 1. Field Programmer Kit Contents

The diagram below shows how the Field Programmer kit is intended to be the bridge between a computer running ClockBuilder Pro software (CBPro) and the QFN socket adapter boards/customer PCB for in-system firmware and volatile programming.

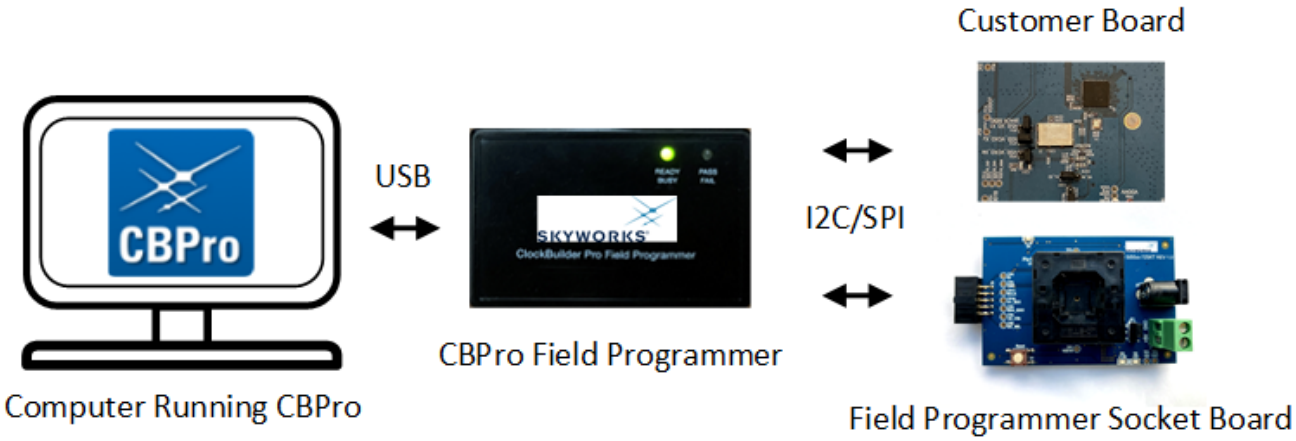


Figure 2. Example Hardware Configuration (Using QFN Socket Board or Customer PCB)

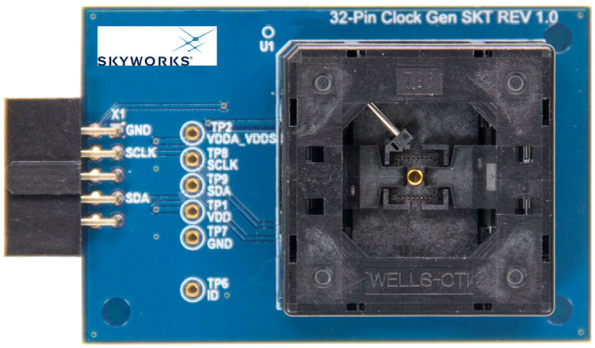


Figure 3. Si5332-32SKT-DK

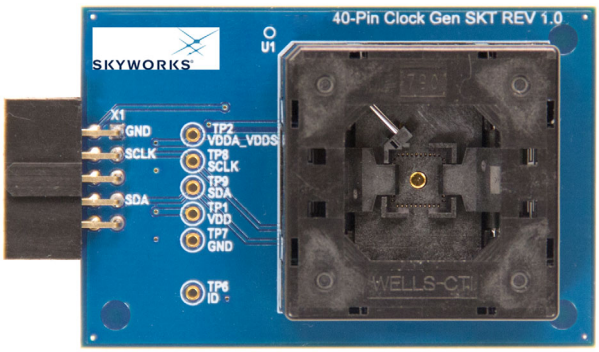


Figure 4. Si5332-40SKT-DK

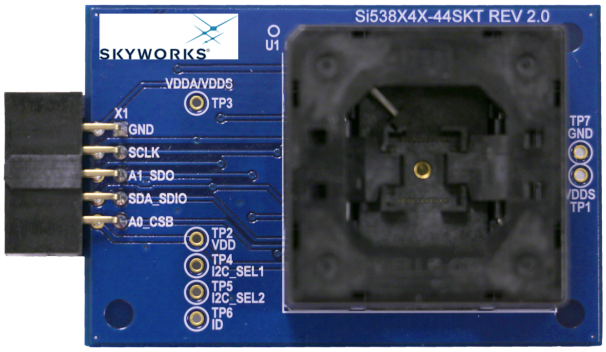


Figure 5. Si538x4x-44SKT-DK

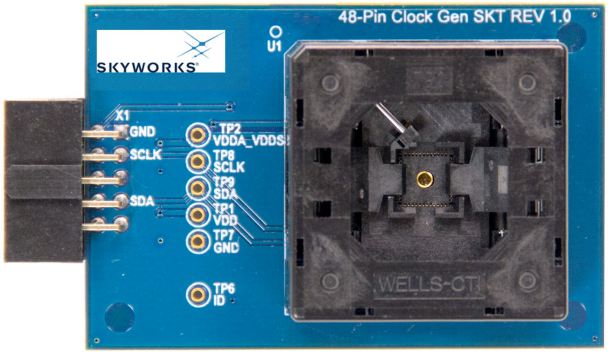


Figure 6. Si5332-48SKT-DK

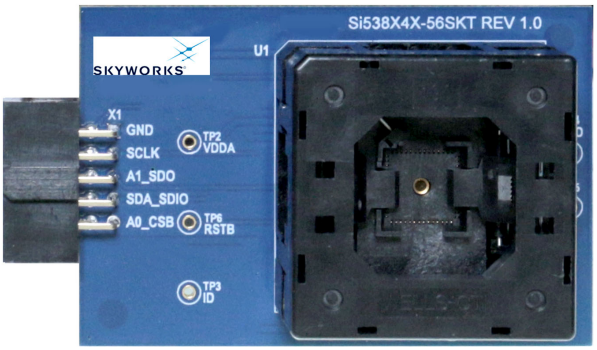


Figure 7. Si538x4x-56SKT-DK

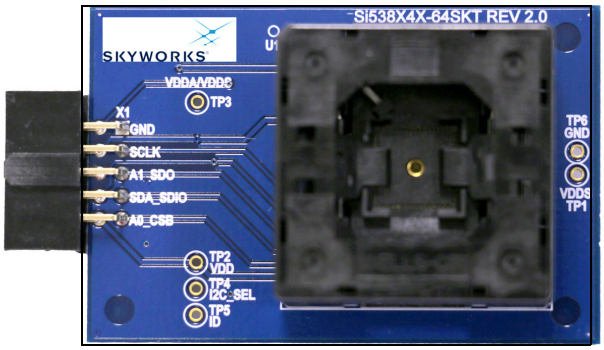


Figure 8. Si538x4x-64SKT-DK

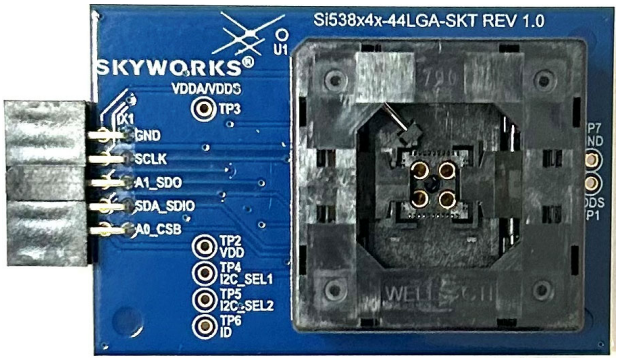


Figure 9. Si538x4x-44LGA-SKT

Figure 10. Si538x4x-64LGA-SKT

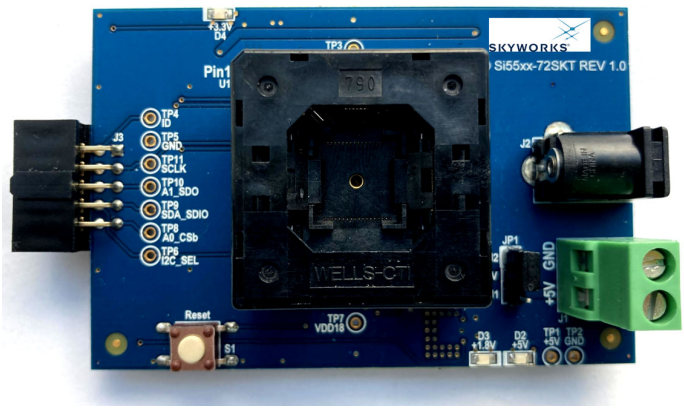


Figure 11. Si55xx-72SKT-DK

3. Software Download and Installation

To install the ClockBuilder Pro Software (CBPro) on any Windows 7 (or above) PC, visit the Skyworks ClockBuilder Pro Field Programmer page and download ClockBuilder Pro (CBPro) to install the software on your host PC. ClockBuilder Pro software allows the user to easily configure many of Skyworks' jitter attenuator and clock generator products, as well as directly write the configuration to the device over the Field Programmer with the click of a button.

4. Hardware Configuration

The Clockbuilder Pro Field Programmer acts as an interface between the CBPro GUI (software running on a computer) and the target device. Connect the provided USB cable to your computer and the Field Programmer. The Field Programmer is then connected to the target device using the provided jumper cables or a programming socket, depending upon the desired end configuration as detailed in the following section, “5. Ways You Can Use the Programmer” on page 9.

5. Ways You Can Use the Programmer

The following four sections describe four ways you can use the Field Programmer.

5.1. In-Socket Firmware/NVM Programming

This workflow describes the process of programming loose devices using the Si5332-32/40/48SKT, Si534x/8x-56SKT, Si534x/8x-64SKT, or Si55xx-72SKT programming socket board. For non-firmware-based solutions, this flow will “burn” a complete configuration from CBPro into available NVM in the device. Si534x/8x/9x devices shipped from Skyworks have two NVM banks available to program (“burn”). Si5332 devices have a flexible NVM space. Si536x, Si540x, and Si55xx devices may have NVM space in select configurations. CBPro manages available NVM and programs (“burns”) the available NVM when feasible. For Si5383/4/8/9 (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 Field Programmer (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 Field Programmer.

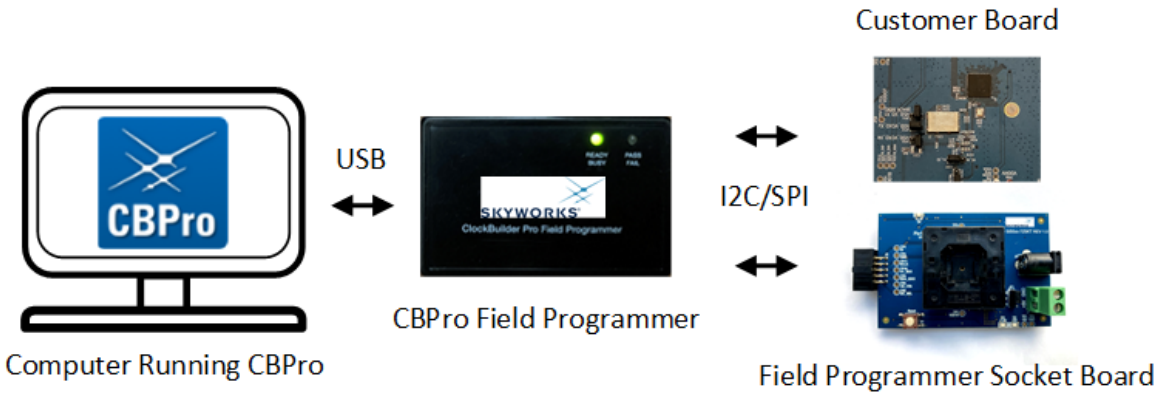


Figure 12. Computer to Field Programmer Connection

2. Insert a base or previously preprogrammed (e.g., OPN) device into the socket.

   Socket and device Orientation: It is important that the device is correctly oriented before powering up the board. However, if there is no device in the socket or if the device is not correctly orientated, the software will auto-detect the fault and will not connect to the part. 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 socket board silk screen. 64-QFN and 44-QFN orientations are shown below. The same idea applies to 32-QFN, 40-QFN, 48-QFN, 56-QFN and 72-QFN package ICs.

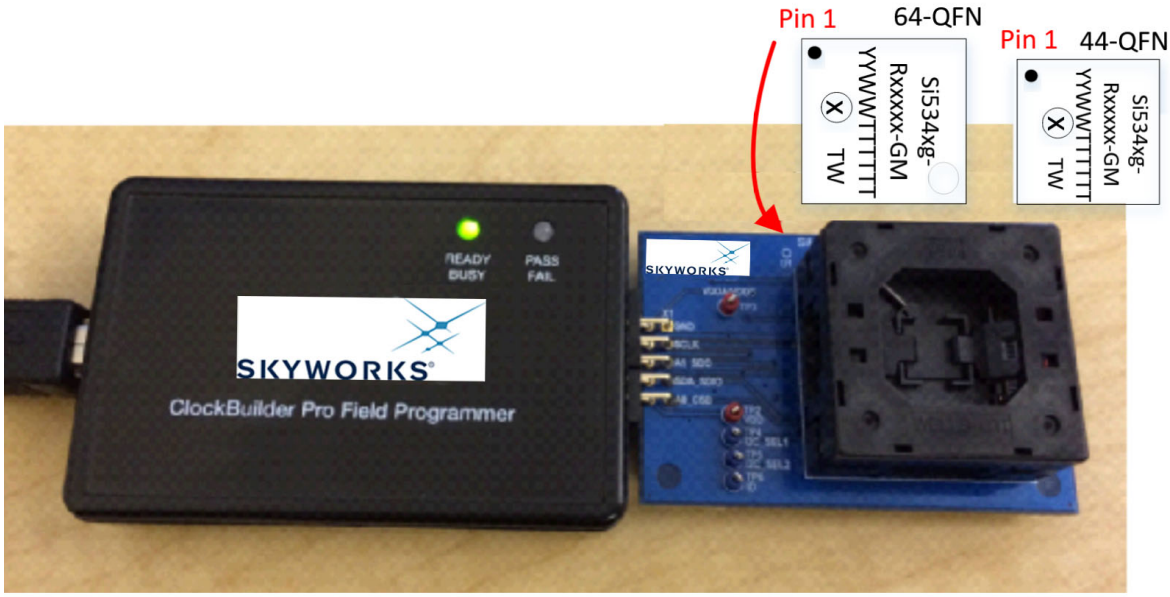


Figure 13. Correct Orientation of a Device in the Socket

3. Power is only applied to the device when you perform a scan or initiate a burn. Power is off at all other times. Power is not applied to the socketed device unless explicitly triggered by the user 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.

4. Connect the QFN Field Programmer Socket Board with the device into the Field Programmer.

Figure 14. Connections from PC to the Target Device

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

Figure 15. ClockBuilder Pro Icon

6. 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.***

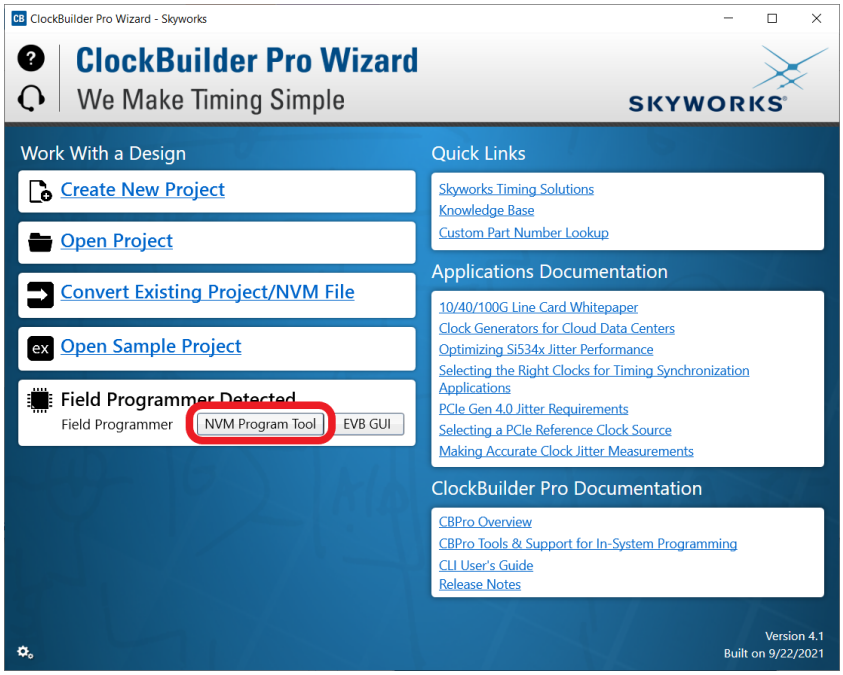


Figure 16. ClockBuilder Pro Wizard

7. 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 following figure:

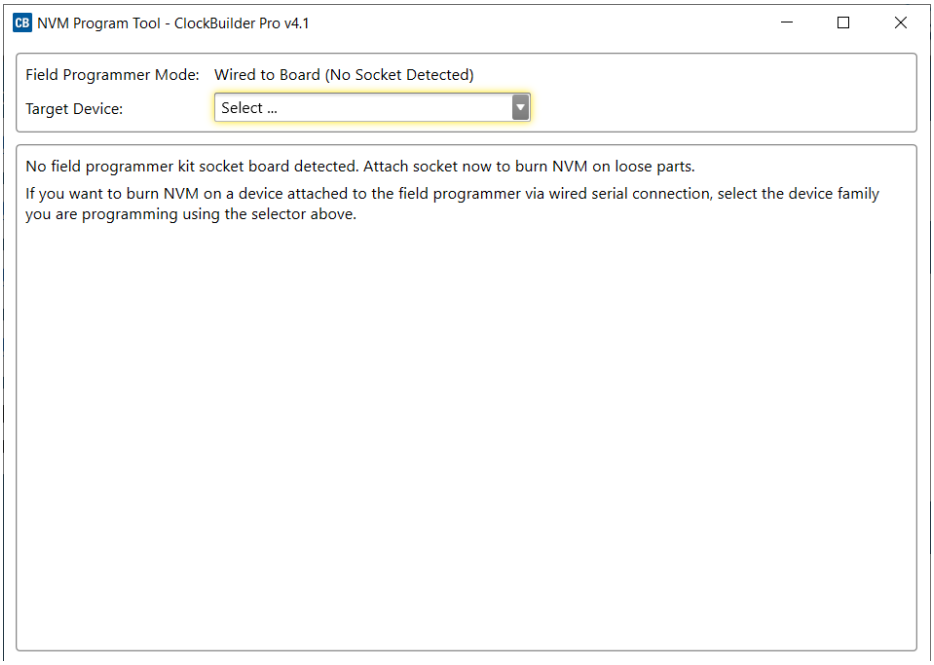


Figure 17. Select Device Family Prompt

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

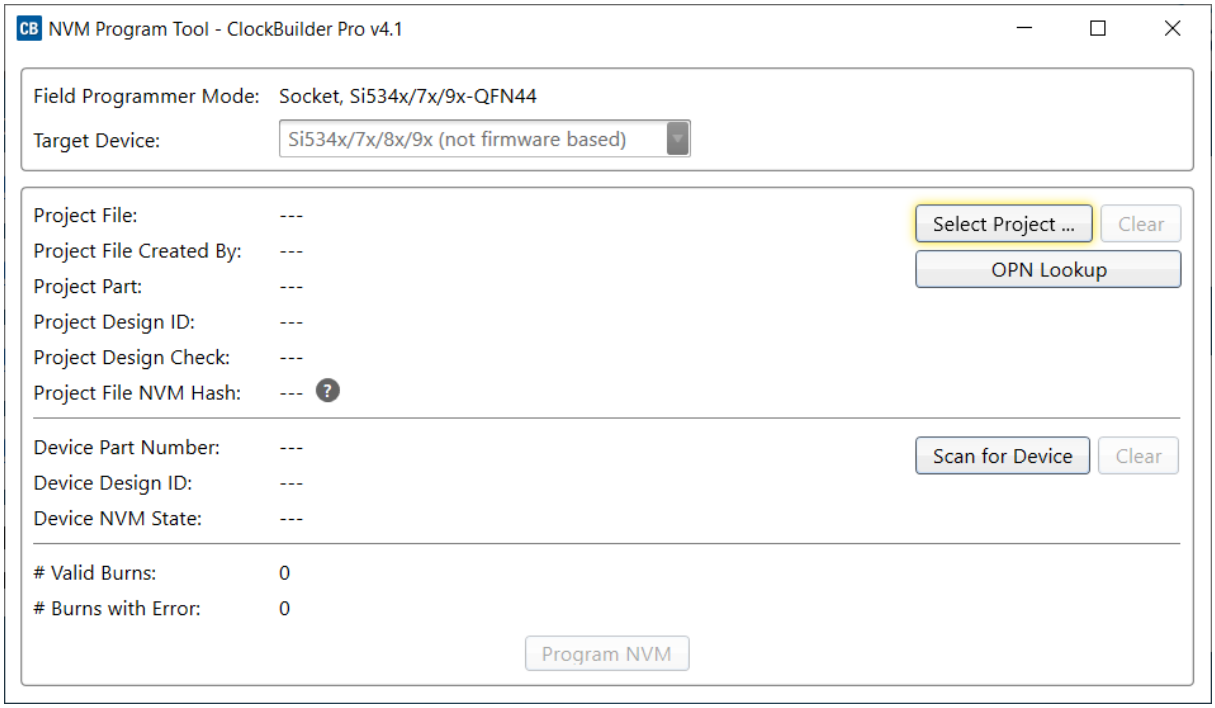


Figure 18. Programming Panel

5.1.1. Programming In-Socket, Si5383/4/8/9 Firmware Based Devices

Refer to Figure 20, “Programming In-Socket, Firmware Based Devices,” on page 15 below.

1. Configure the serial interface.
2. Common issues:
- If I^2C is used, check that the I^2C address matches what is expected (see datasheet).
- For both SPI and I^2C, start with a low bus speed. There can be connection issue due to signal integrity at high bus speeds.
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 Skyworks. 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.

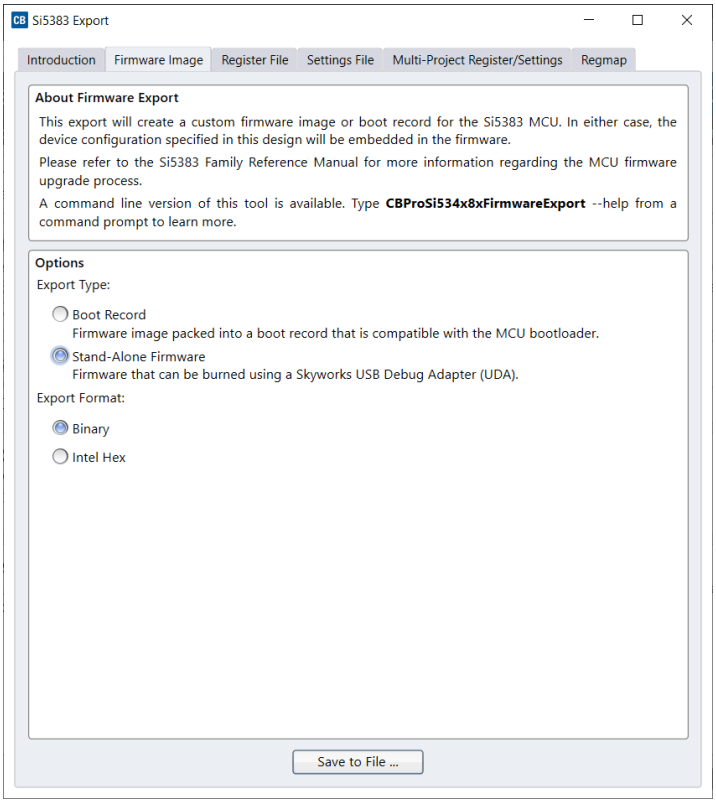


Figure 19. Stand-Alone Firmware Selection in the CBPro Export Menu for a Si5383 Device

4. Click the “Select …” button and select the file to flash to the device.
5. 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.
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.

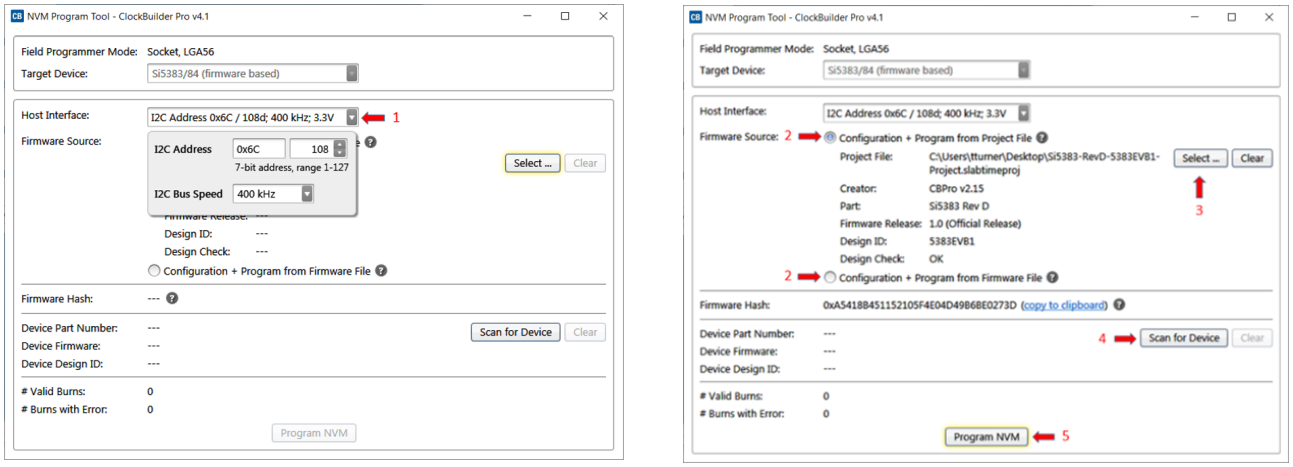


Figure 20. Programming In-Socket, Firmware Based Devices

5.1.2. Programming In-Socket, Si534x/8x/9x Non-Firmware Based Devices

Refer to Figure 21 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.

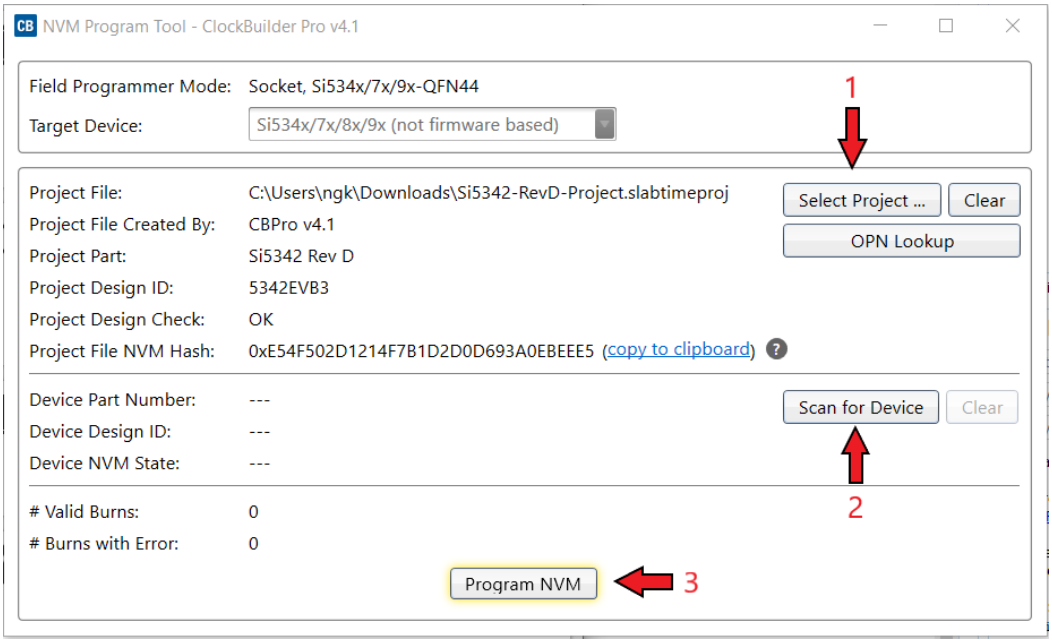


Figure 21. Programming In-Socket, Non-Firmware Based Devices

5.1.3. In-Socket Programming Status

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

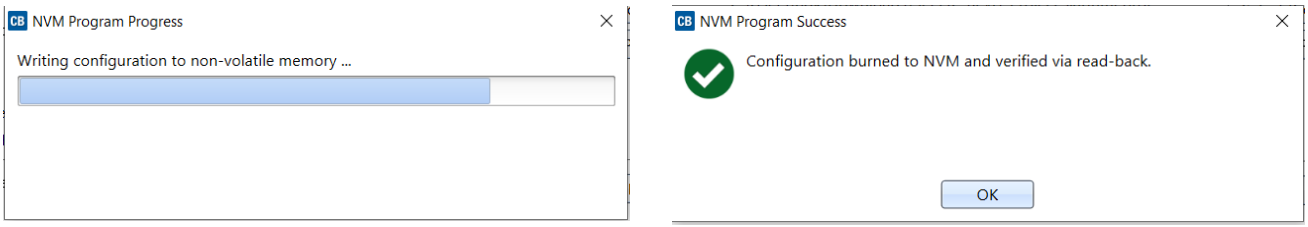


Figure 22. In-Socket Programming Status

5.2. In-System Firmware/NVM Programming

This workflow describes the process of programming a device mounted on a PCB. For Si534x/8x/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 Skyworks 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. NVM Programming is supported for select configurations for the Si536x/Si540x/Si55xx devices. Check the device reference manuals for details.

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

1. Install the CBPro software if you have not already done so.
2. Connect the adapter (Field Programmer) board with the USB cable to the PC on which CBPro was installed.

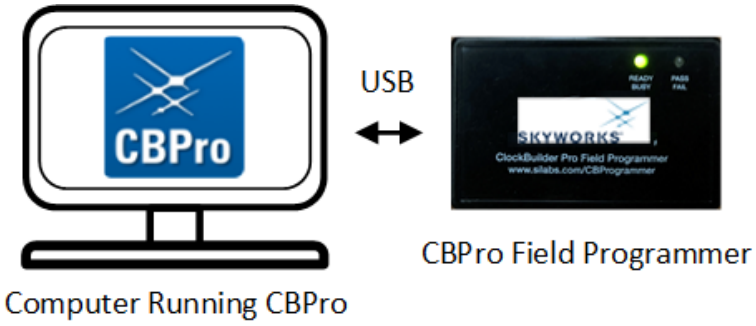


Figure 23. PC to Field Programmer Connection

3. Look up and verify the host I/O mode (I^2C or SPI), the I^2C address, and the interface I/O voltage level compatibility of your host's I/O voltage (for I^2C or SPI) and the device.

   On Si534x/8x/9x devices, 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 Field Programmer. You can look up your device host I/O voltage using the "OPN Lookup" option in the NVM Burn tool, as shown below.

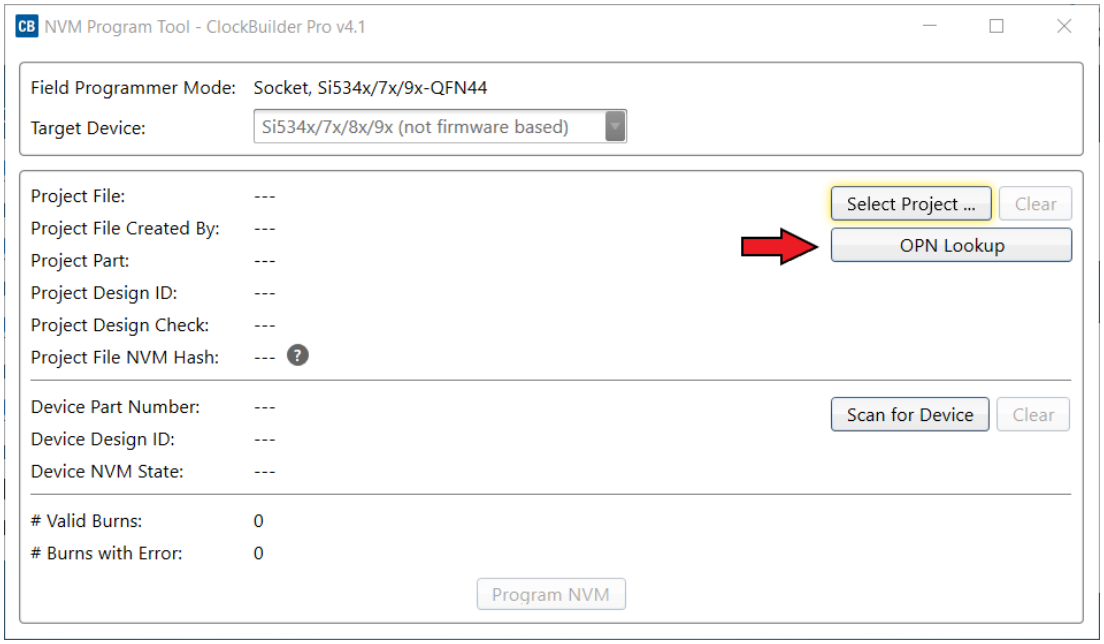


Figure 24. OPN Lookup Option

If you have a custom OPN mounted on your board (a part number with a five-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 your custom OPN, as shown below.

a. Enter in your full ordering part number (OPN). E.g., Si5342D-D14249-GM.
b. Click Search.
c. Click the addendum link to download the device addendum.

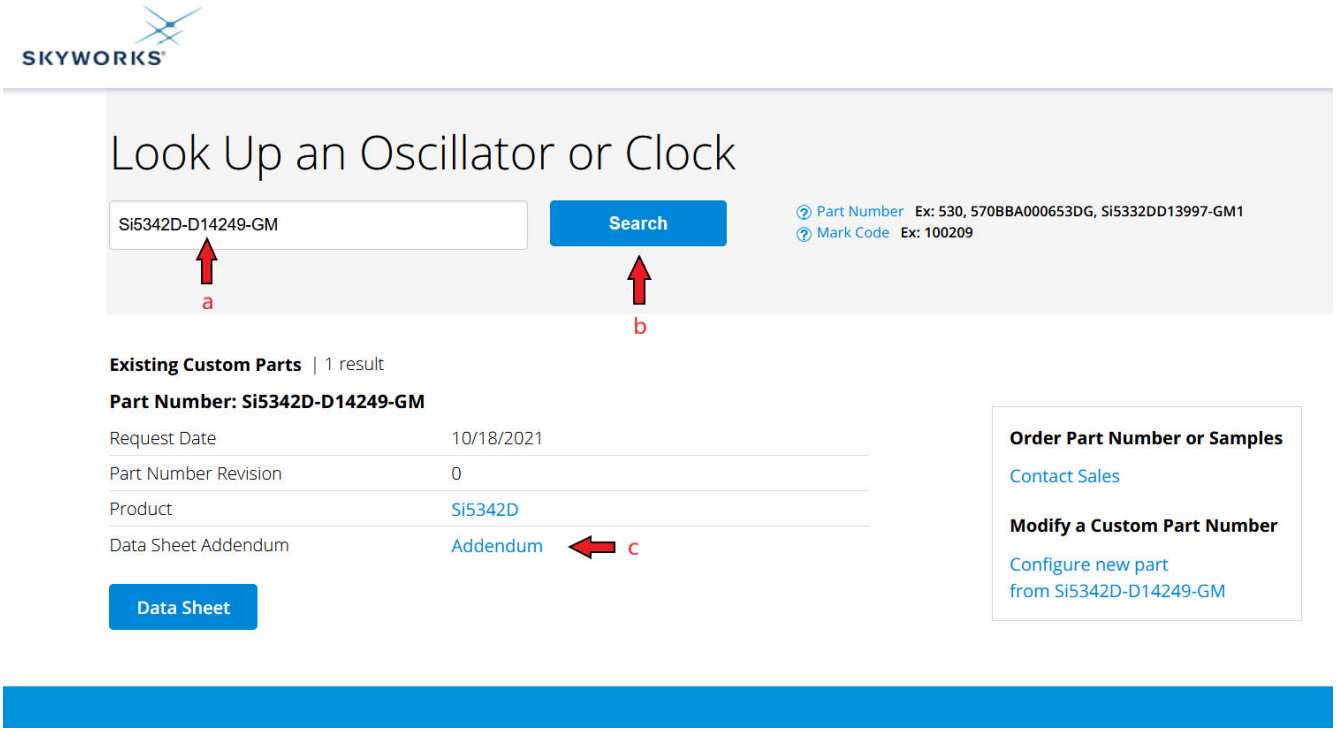


Figure 25. OPN Lookup

4. 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/9x I^2C/SPI interface will operate from a 1.8 V supply.
- "VDDA (3.3 V)" indicates the I/O supply for the Si534x/8x/9x I^2C/SPI interface will operate from a 3.3 V supply.
- "VDDD" indicates the I/O supply for the Si5332 I^2C interface. "VDDIO" indicates the I/O supply for the Si536x, Si540x, and Si55xx devices.

The image below shows an example data sheet addendum showing VDDA (3.3 V).

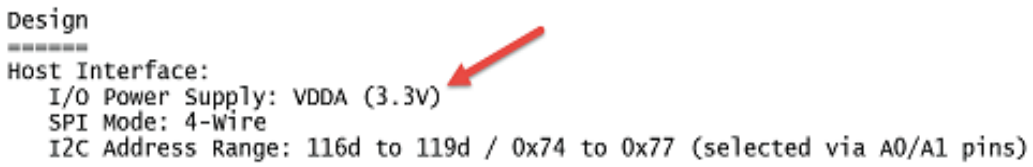


Figure 26. Finding VDDA Value

Table 2. Supported Serial Protocols for the Si534x/8x/9x Devices

	1.8 V	2.5 V	3.3 V
4-wire SPI	Supported		
3-wire SPI			
I^2C			

For Si536x, Si540x, and Si55xx devices, I/O supply voltage is listed under "VDDIO".

```
Host Interface
==============

VDDIO Supply Voltage: +1.8V
Mode: SPI 3-Wire
Secondary SPI 3-Wire Interface: Disabled
```

Figure 27. Finding the VDDIO Setting

The list of supported SPI modes and voltages for the Si536x, Si540x, and Si55xx are listed in Table 3.

Table 3. Supported Serial Protocols for the Si536x, Si540x, and Si55xx Devices

	1.8V	2.5V	3.3V
4-wire SPI	Supported		
3-wire SPI	Not Supported	Not Supported	Supported
I^2C	Supported		

5. Connect/wire the pins of the Field Programmer 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.

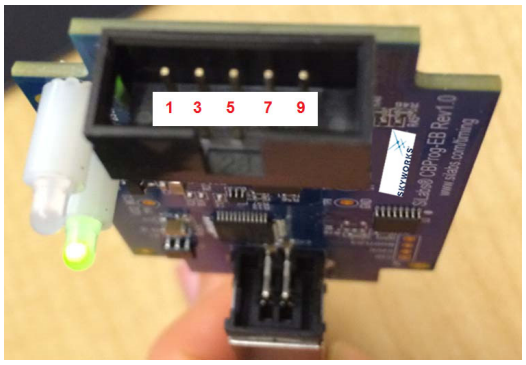


Figure 28. Interface Pins on Header (Front View of the Field Programmer)

Table 4. Interface Pin Connections from the Field Programmer

Pin #	Description	Wire to Your PCB?	I^2C	4-wire SPI	3-wire SPI
1	GND	Always	GND		
2	ID	Never	Select 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.		
3	SCLK	Always	Serial clock signal for I^2C transactions.	Serial clock signal for SPI transactions.	
4	VDDA_VDDS (Si536x, Si540x, and Si55xx devices do not use this pin)	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.		
5	A1_SDO (applies only for Si534x/8x/9x, Si55xx, Si540x, Si536x devices)	4-Wire SPI Only	For Si534x/8x/9x devices, this pin can be used to set I^2C address bit A1 high or low. Routed to A1 device pin on the programming Field Programmer Socket Boards. For Si536x, Si540x, Si55xx devices, this pin is used to received data from the device in 4-wire SPI mode.	Serial data from device for 4-wire SPI transactions (MISO).	Not used
6	I2C_SEL2 (applies only for Si534x/8x/9x 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)	
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/9x 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)	
9	A0_CSB (applies only for Si534x/6x/8x/9x, Si55xx, Si540x, 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	
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.		

5.2.1. I²C Hardware Configuration

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

Field Programmer

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

Si534x/8x/9x Devices:

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

Si5332/7, Si536x, Si540x, Si55xx Devices:

- The device must be explicitly configured for I²C from the Host Interface page of ClockBuilder Pro. The field programmer supports I²C at 1.8V, 2.5V, and 3.3V.
- A0/CSb: Drive this pin high or low to set the I²C Address.

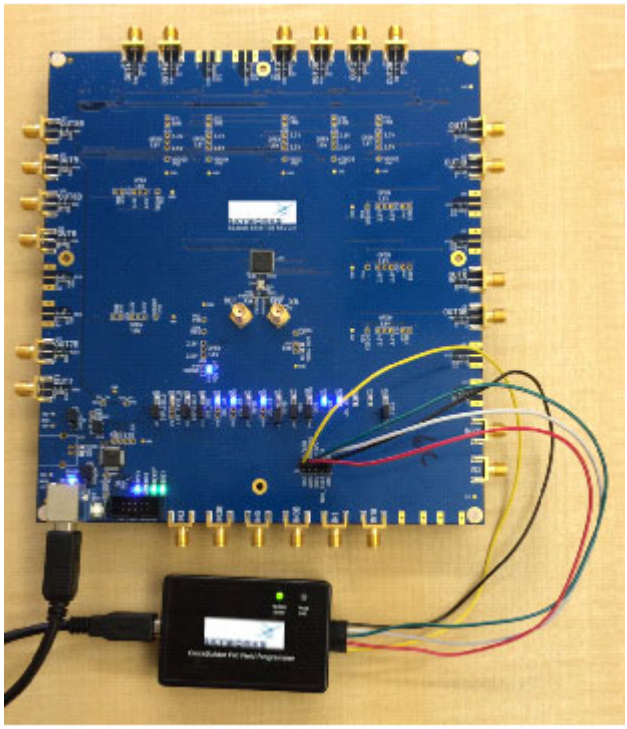


Figure 29. Example I²C Connection to External System Target Board Using Jumper Wires (Si5346-EVB)

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

5.2.2. SPI 3-Wire Hardware Configuration

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

Field Programmer

- 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

Si534x/8x/9x device

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

Si536x, Si540x, Si55xx device

- The device must be explicitly configured for 3-wire SPI from the Host Interface page of ClockBuilder Pro. The field programmer only supports 3.3V communications with these devices.

5.2.3. SPI 4-Wire Hardware Configuration

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

Field Programmer

- 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

Si534x/8x/9x Device

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

Si536x, Si540x, Si55xx Device

- The device must be explicitly configured for 4-wire SPI from the Host Interface page of ClockBuilder Pro. The field programmer supports 3.3V, 2.5V, and 1.8V 4-wire SPI communications with these devices.

5.2.4. Programming In-system, Si5383/4/8/9 Firmware Based Devices

Refer to Figure 30. Programming In-system, Firmware Based Devices below.

After verifying the CBPro Field Programmer to device connections, execute the following steps. This example assumes a device is configured with an I^2C address of 0x6C, and an I^2C bus speed of 400 kHz.

1. Select “Si5383/4/8/9 (firmware based)” in the Target Device drop down.
2. Click the Host Interface drop down:
   a. Enter the I^2C 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 Skyworks. 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.

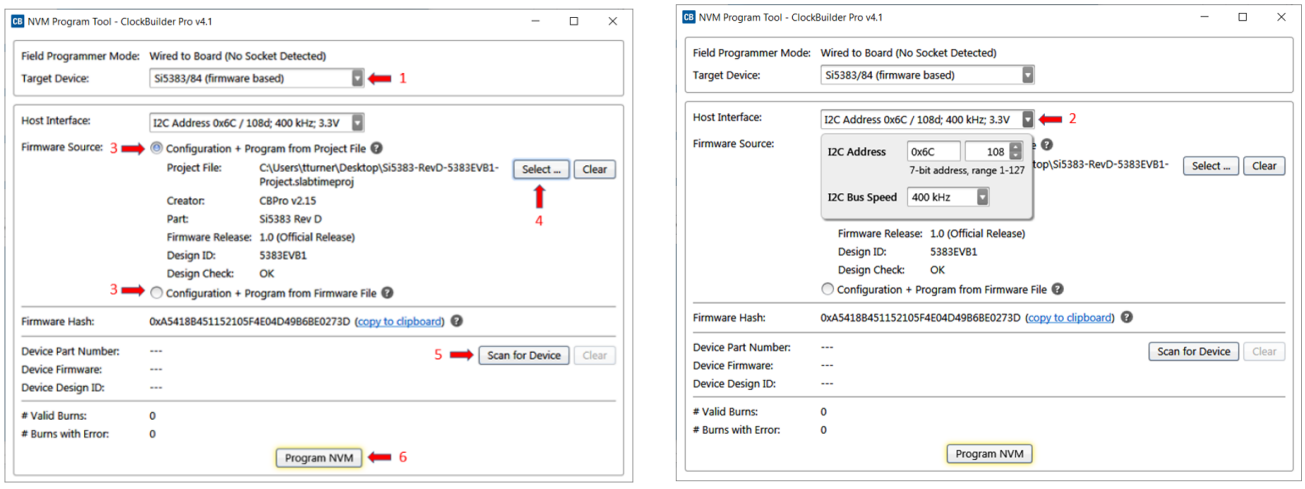


Figure 30. Programming In-system, Firmware Based Devices

5.2.5. Programming In-system, Si534x/8x/9x Non-firmware Based Devices

Refer to Figure 31. Programming In-system, Non-firmware Based Devices below.

After verifying the CBPro Field Programmer 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 0x68, and an I^2C bus speed of 400 kHz. For Si536x, Si540x, and si55xx devices, this section does not apply because the device does not have any available NVM banks. The example below shows the workflow for a Si534x/8x/9x device.

1. Select "Si534x/8x/9x (not firmware based) in the Target Device drop down.
2. Click the Host Interface drop down (Review: **host I/O mode (I^2C or SPI)**, **the I^2C 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^2C, 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.

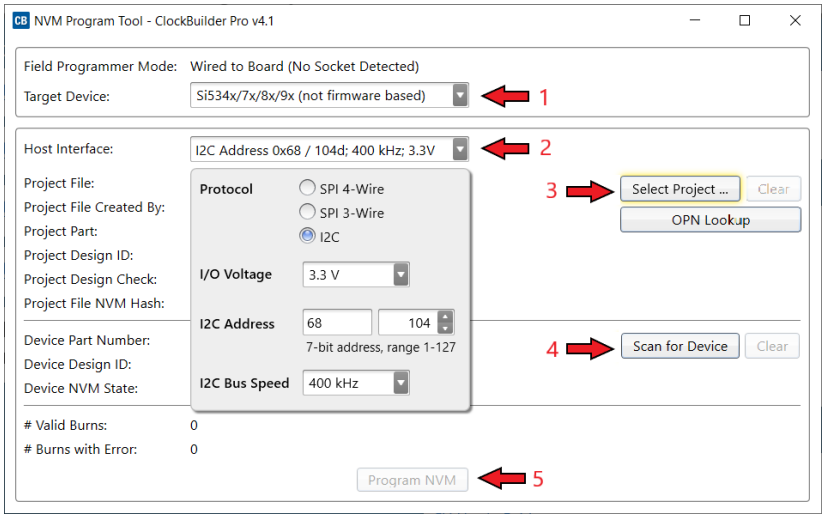


Figure 31. Programming In-system, Non-firmware Based Devices

5.2.6. Programming Status

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

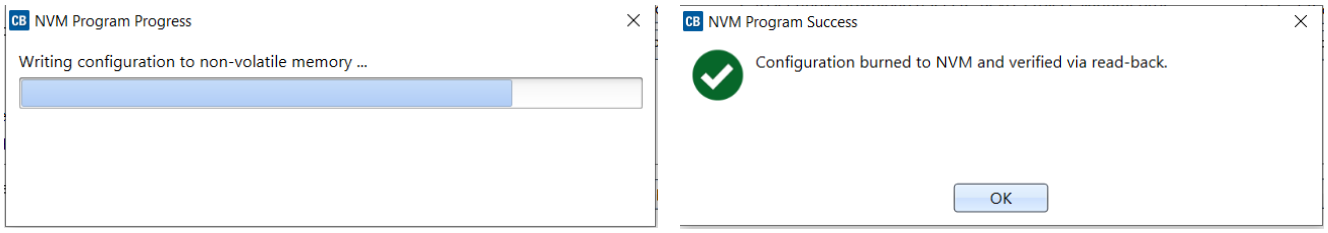


Figure 32. Programming Status

5.3. In-System Volatile Register Programming and Register Debug

This workflow guides users through the full CBPro configuration Wizard to make volatile changes to a device's configuration, and also walks the user through the CBPro EVB GUI application to inspect the state of various status registers and other information via the Device APIs (where supported) in real-time. 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 and execute Device API commands where supported.

All of the relevant CBPro features available when working with a Skyworks 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 Skyworks Si534x/8x/9x evaluation boards, but are written to the Si536x, Si540x, and Si55xx evaluation boards. If using a Si536x, Si540x, or Si55xx evaluation board, check the board schematic and make the required board modifications).

5.3.1. 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 Figure 33.

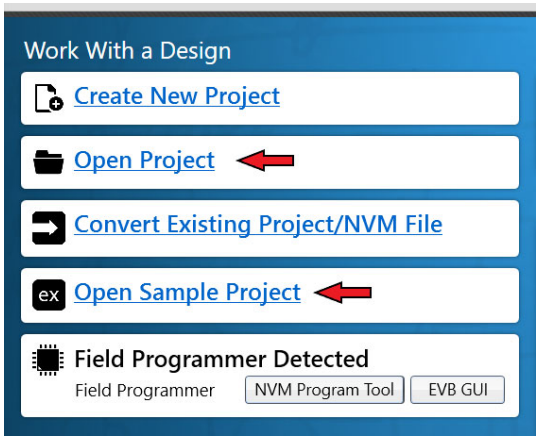


Figure 33. Open Design Project File, and see Field Programmer Detected

5.3.2. 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 Field Programmer. For example, in the figure below, a Si5397 project has been opened and the CBPro Field Programmer has been detected, and no socket is present:

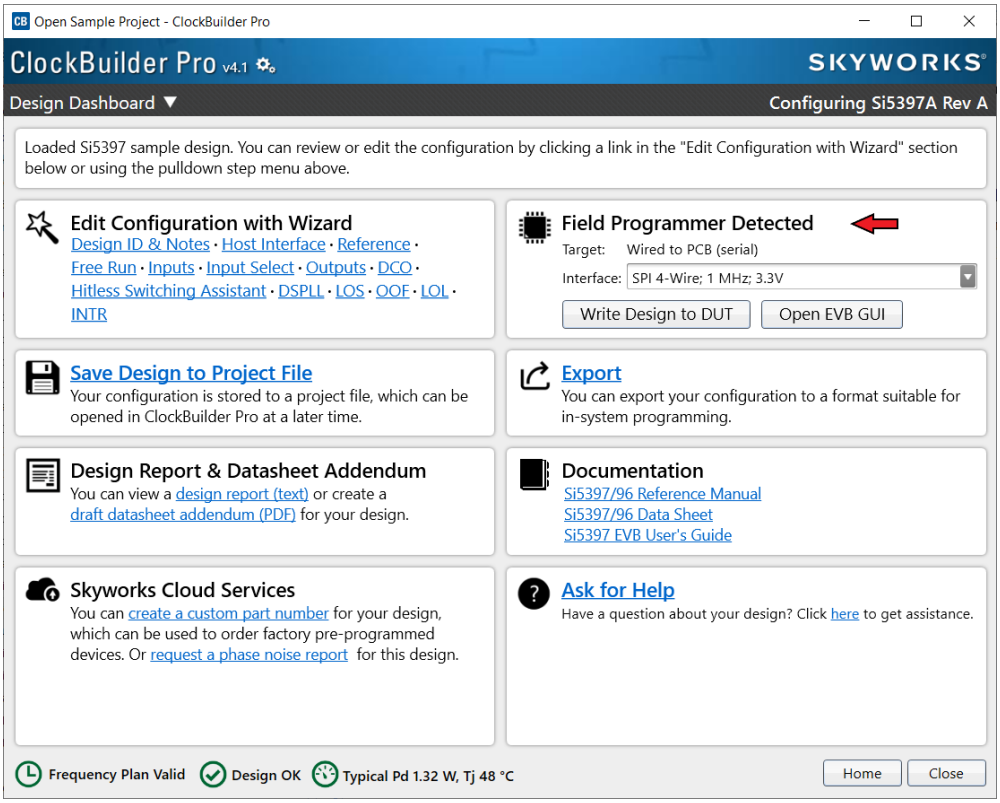


Figure 34. Overview of CBPro Configuration Wizard and the Field Programmer

With a click of the “Write Design to DUT” button, you can reconfigure the Si5397 in-system to test changes to your design. The “Open EVB GUI” button can be used to launch the EVB GUI. The EVB GUI presents a real-time view of the status of the clock device, and provides an interface to interact with the device during runtime. For Si534x/8x/9x devices, the EVB GUI allows you to peek/poke registers, and watch realtime lock flags and alarms on the in-system device. For Si536x, Si540x, and Si55xx devices, the EVB GUI allows you to view the realtime PLL lock status of the device, view alarms, and interact with the device through the device API. See “5.3.4. Using the EVB GUI with In-System Devices” on page 32 to learn more.

5.3.2.1. Using the Dashboard with In-system Devices

If the CBPro Field Programmer is connected via USB and detected by CBPro, you will see will see a pulldown to configure the host interface between the Field Programmer and your PCB, as shown in the figure below. Refer to “5.2. In-System Firmware/NVM Programming” on page 17 for information on connecting the CBPro Field Programmer to your hardware.

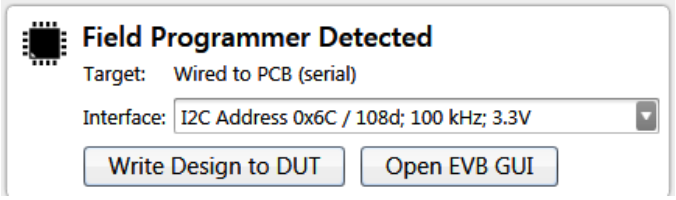


Figure 35. Field Programmer Detected

Once detected, select the appropriate settings via the dropdown menus.

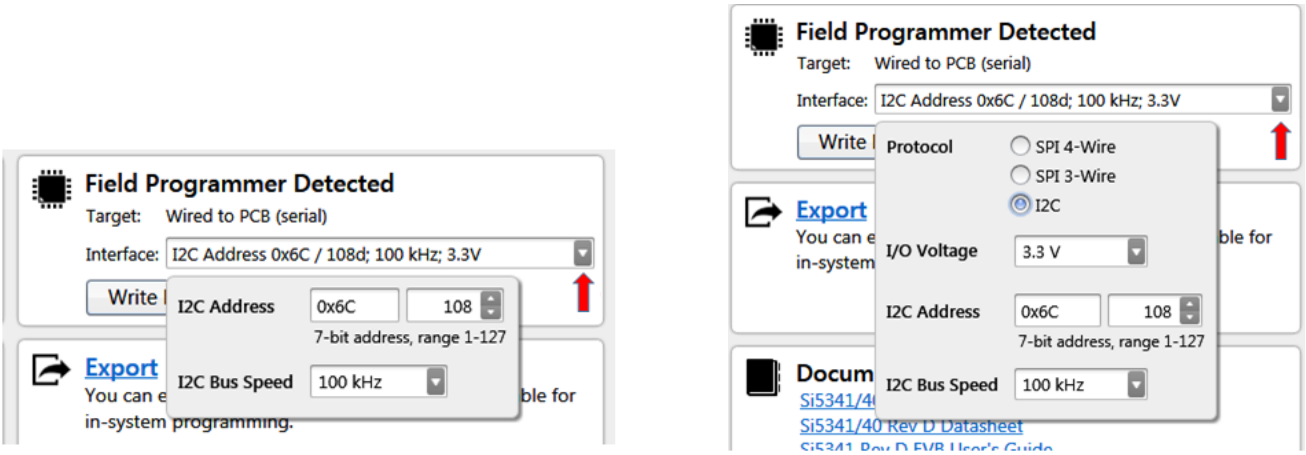


Figure 36. Communication Interface Selection

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

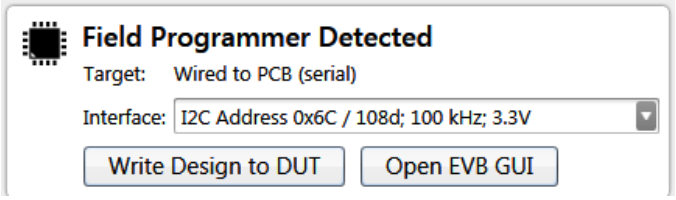


Figure 37. Write Design to DUT

Or on any configuration page in the wizard:

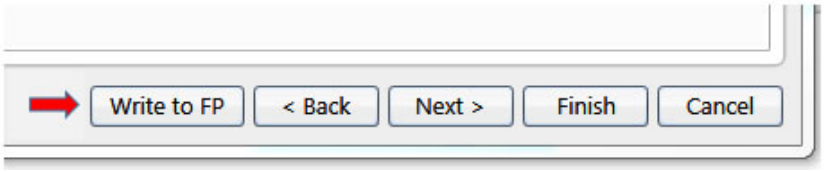


Figure 38. Write to FP

When you initiate a project write to the device, CBPro will first try to verify the device 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 Si534x/8x/9x devices, or by trying the SIO_TEST API command for Si536x, Si540x, and Si55xx devices.

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

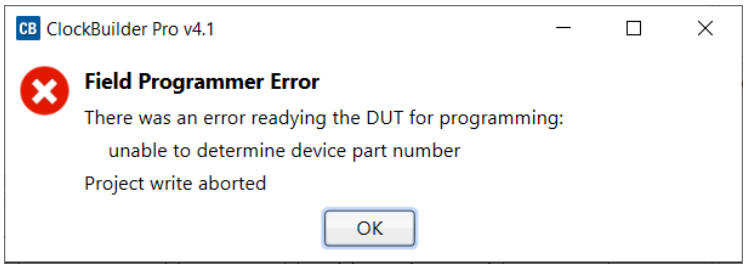


Figure 39. Error Message

Si536x, Si540x, and Si55xx devices will generate an error based on a failed API command read/write:

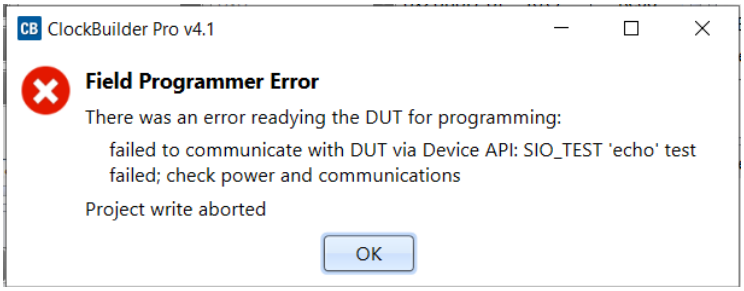


Figure 40. Si536x, Si540x, and Si55xx (API based) Write Error

5.3.2.2. Using the CBPro Dashboard with In-socket Devices

In the design dashboard, you will see a dropdown menu to configure the host interface between the CBPro Field Programmer 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 Figure 41.

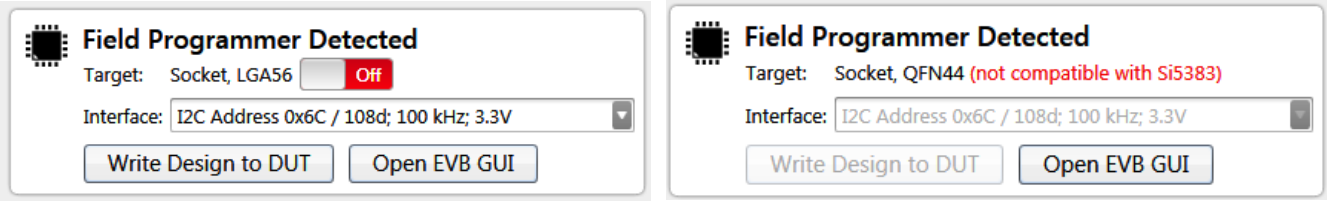


Figure 41. Socket Compatibility

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.

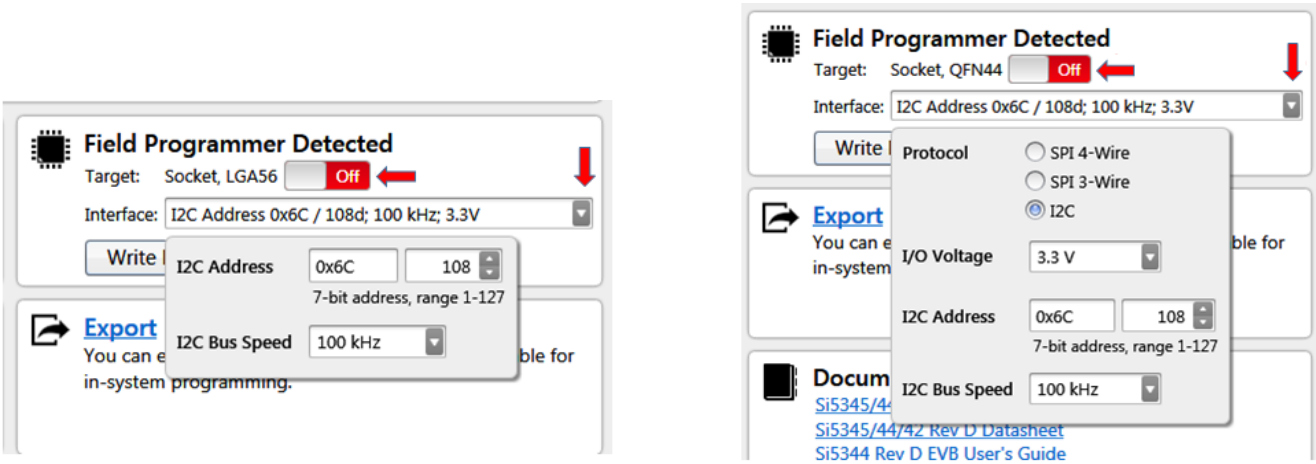


Figure 42. Interface Settings

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

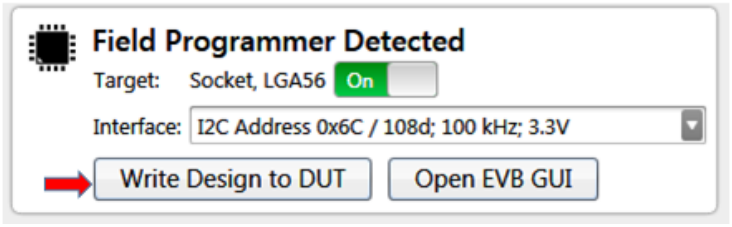


Figure 43. Write Design to DUT

Or on any configuration page in the wizard:

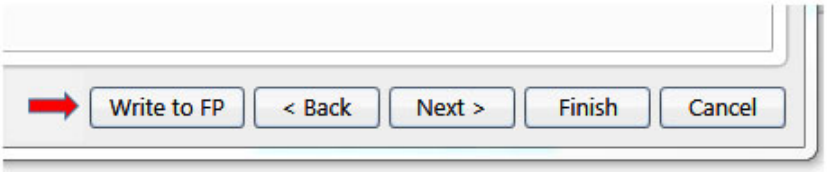


Figure 44. Write Design to FP

5.3.3. Launching the CBPro EVB GUI

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

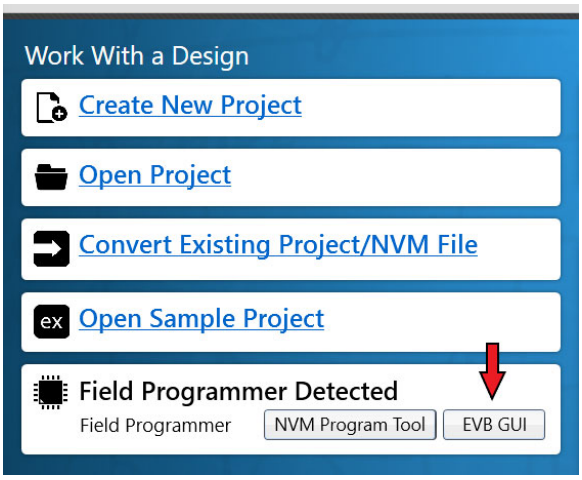


Figure 45. Open EVB GUI Screen

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:

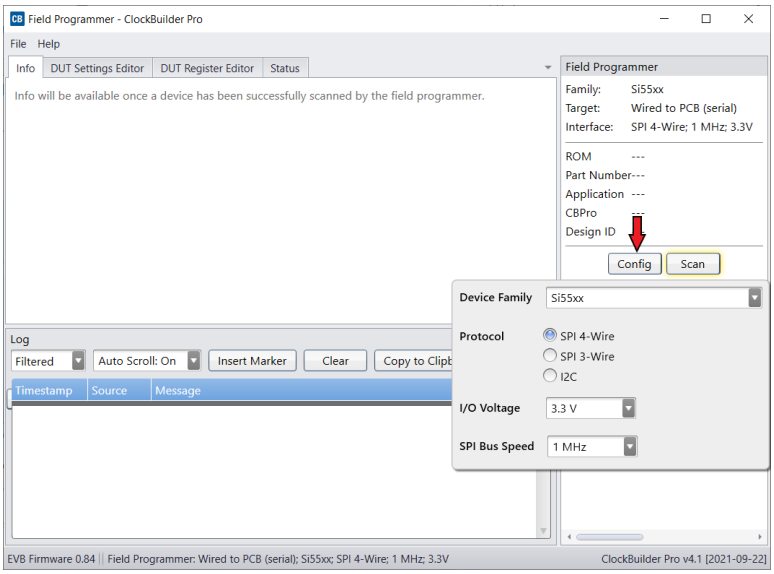


Figure 46. Select Device Family Prompt

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.

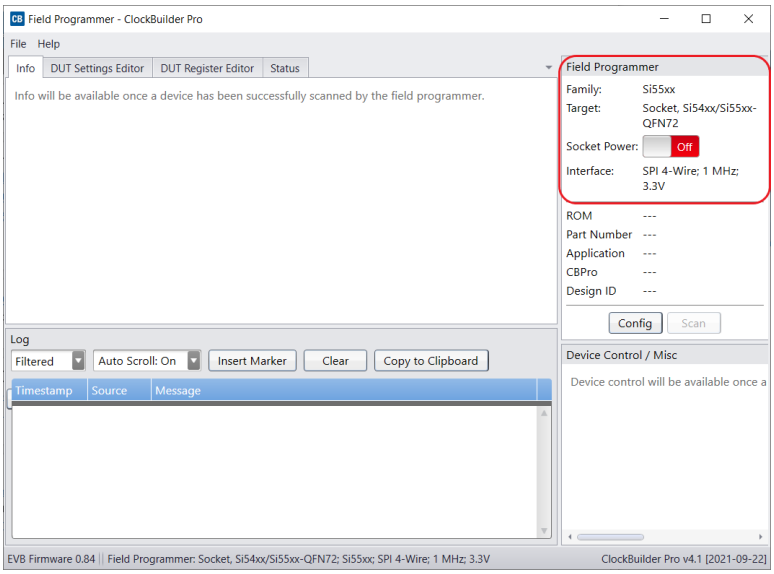
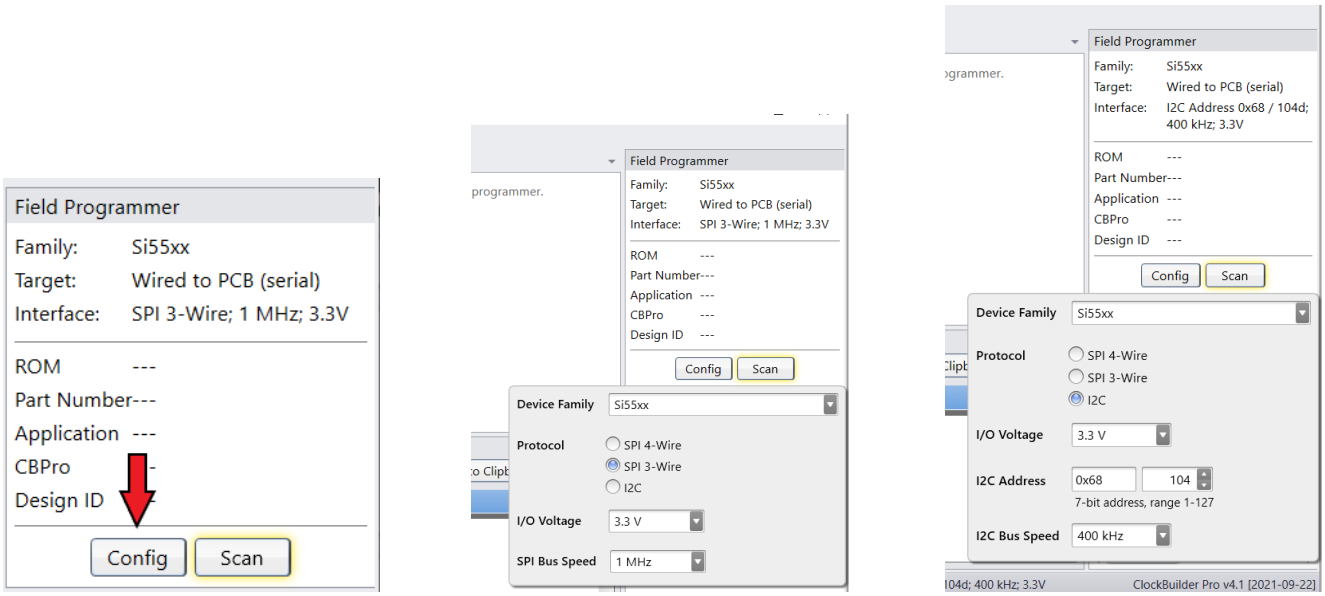


Figure 47. Socket Detected, Auto-Selected Family Prompt

5.3.4. Using the EVB GUI with In-System Devices

Connect the CBPro Field Programmer to the PCB mounted device. Refer to "5.2. In-System Firmware/NVM Programming" on page 17 for information on connecting the CBPro Field Programmer to your hardware.

1. Check that the appropriate voltage rails of the device are active. Attempting to communicate with a powered-off device will result in errors.
2. Click the Config button and click the Device Family pulldown to select the appropriate target device.
3. Configure the host interface to match what is being used on the device. In the example below, the Field Programmer is connected to a Si5518 device over 3-wire SPI.
4. I/O voltage should be set to match the voltage of the device. If you do not know the I/O voltage setting of the device, follow the steps in "5.2. In-System Firmware/NVM Programming" on page 17 to find out what I/O voltage is being used on your device.
5. After the configuration is complete, click the Scan button.



SPI Configuration I²C Configuration

Figure 48. Configuring an In-System Device

The Part Number and Design ID fields should update with the device information under the Info tab.
For Si534x/8x/9x devices, 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. For Si536x/540x/55xx devices, the Device API tab can be used to interact with the device. Screen shots that follow walk through some features of the EVB GUI on Si55xx devices. Some features are shared between part families, while others may be unique.

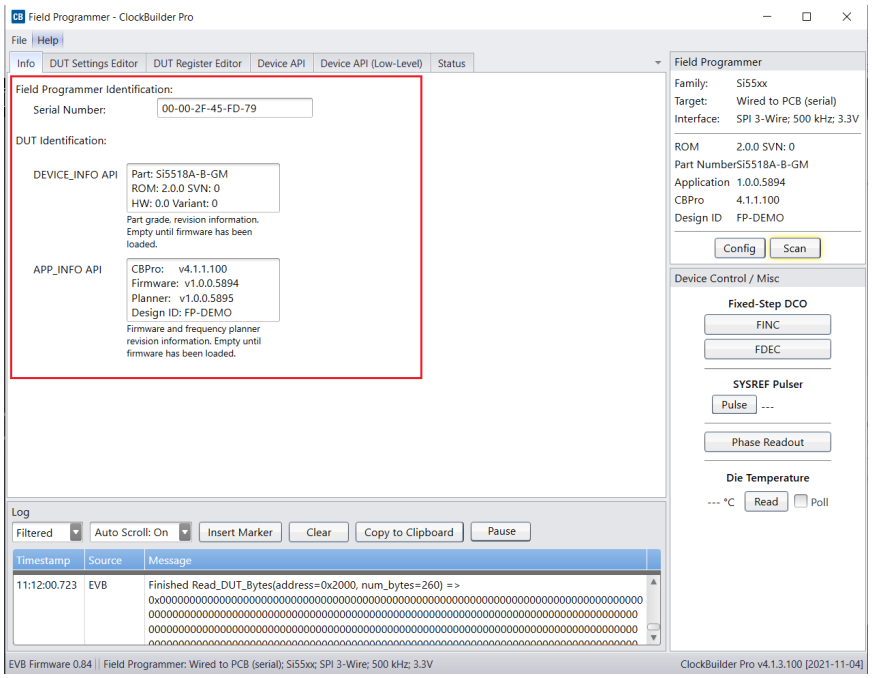


Figure 49. Device info is Displayed after a Successful Scan Operation (Si5518 Device)

To view the device status in real-time, go to the “Status API” tab (you may have to use the small arrows to scroll over). In the example below, the real-time display shows that there is an issue with the reference clock input, and all the PLLs are not locked due to this issue.

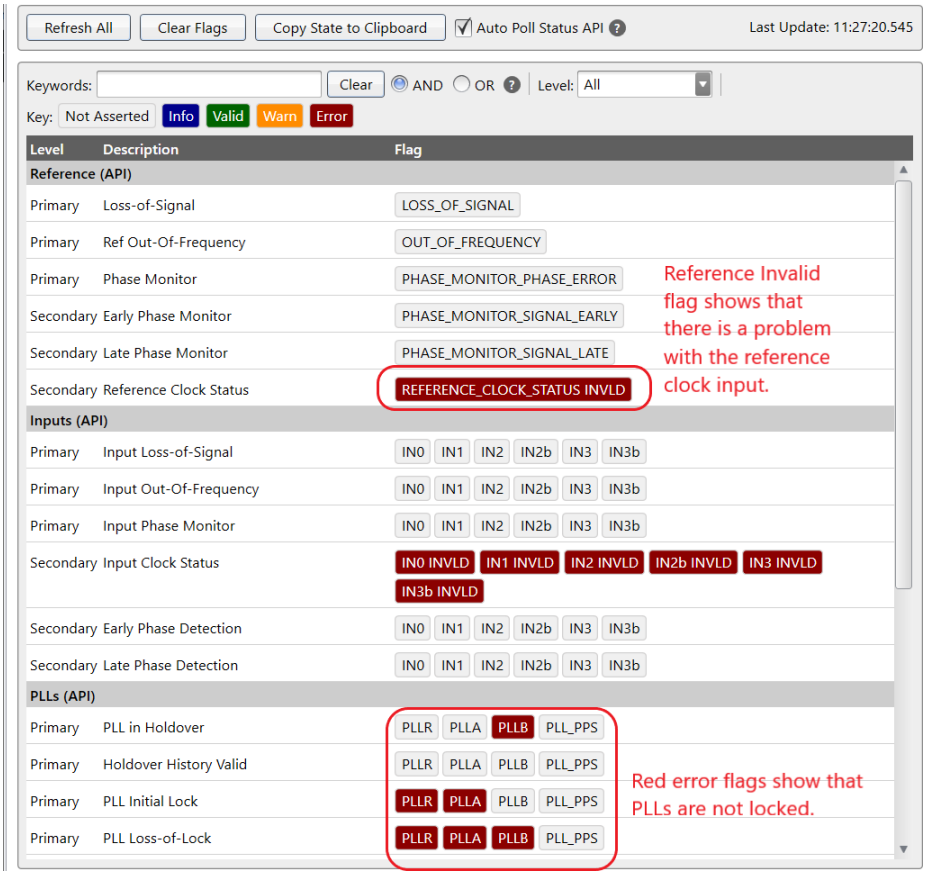


Figure 50. EVB GUI Reference Error Example, Si5518

To generate the scenario above, the CBPro frequency plan that was loaded onto this device had an incorrect reference frequency of 12.345MHz. On a customer board, a more likely error is that the external clock is powered on but at the incorrect frequency. Correcting the reference frequency to what is actually fed to the reference input (54MHz) and overwriting the plan onto the Si5518 device clears the reference clock status invalid error.

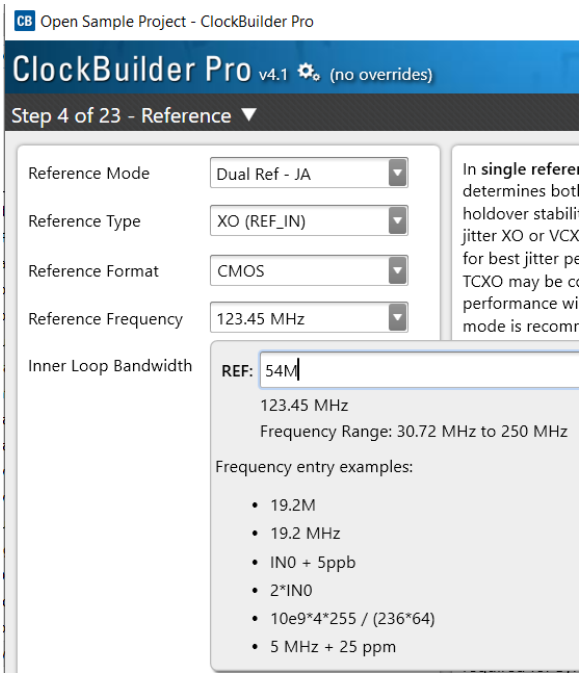


Figure 51. Correcting the Reference Frequency

After correcting the frequency, the reference invalid flag turns green. However, all the input flags are red, and the PLLs are still unlocked.

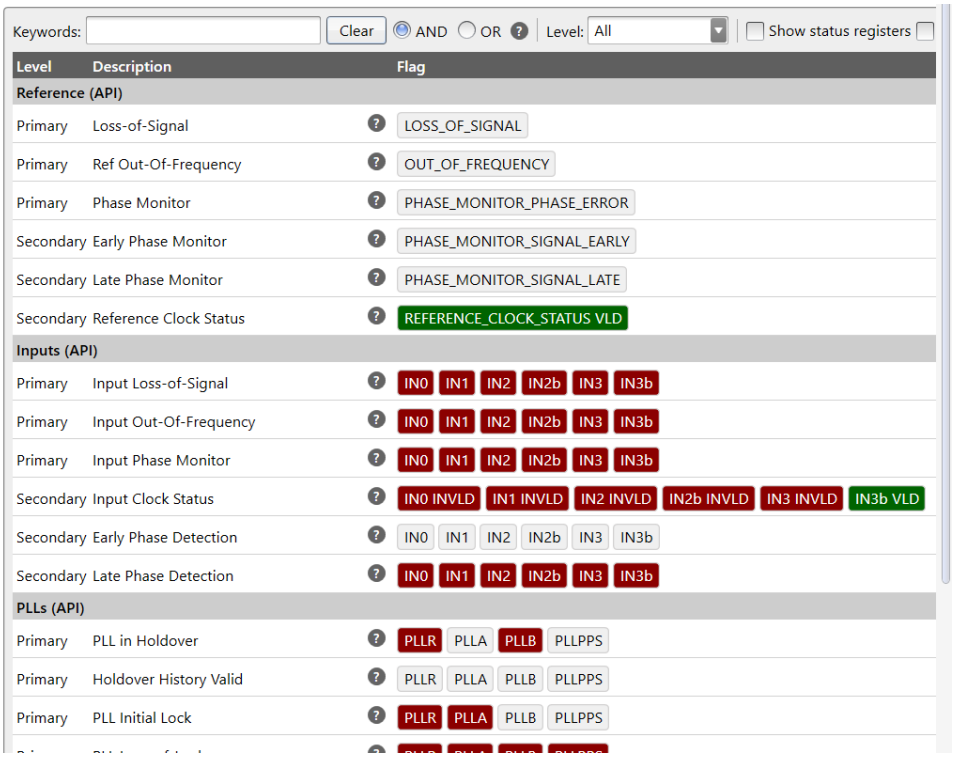


Figure 52. External Reference Valid but Inputs are Invalid and PLLs Not Locked

The input Loss-of-Signal errors, as well as all the other errors under the Inputs (API) section, are caused by the input clocks not being present. Connecting valid inputs will clear those errors, and PLLs should start locking.

On the Si536x, Si540x, and Si55xx devices, the Device API tab can be used to further interact with the clock device. In the image below, the Device API is used to read the die temperature of the Si5518.

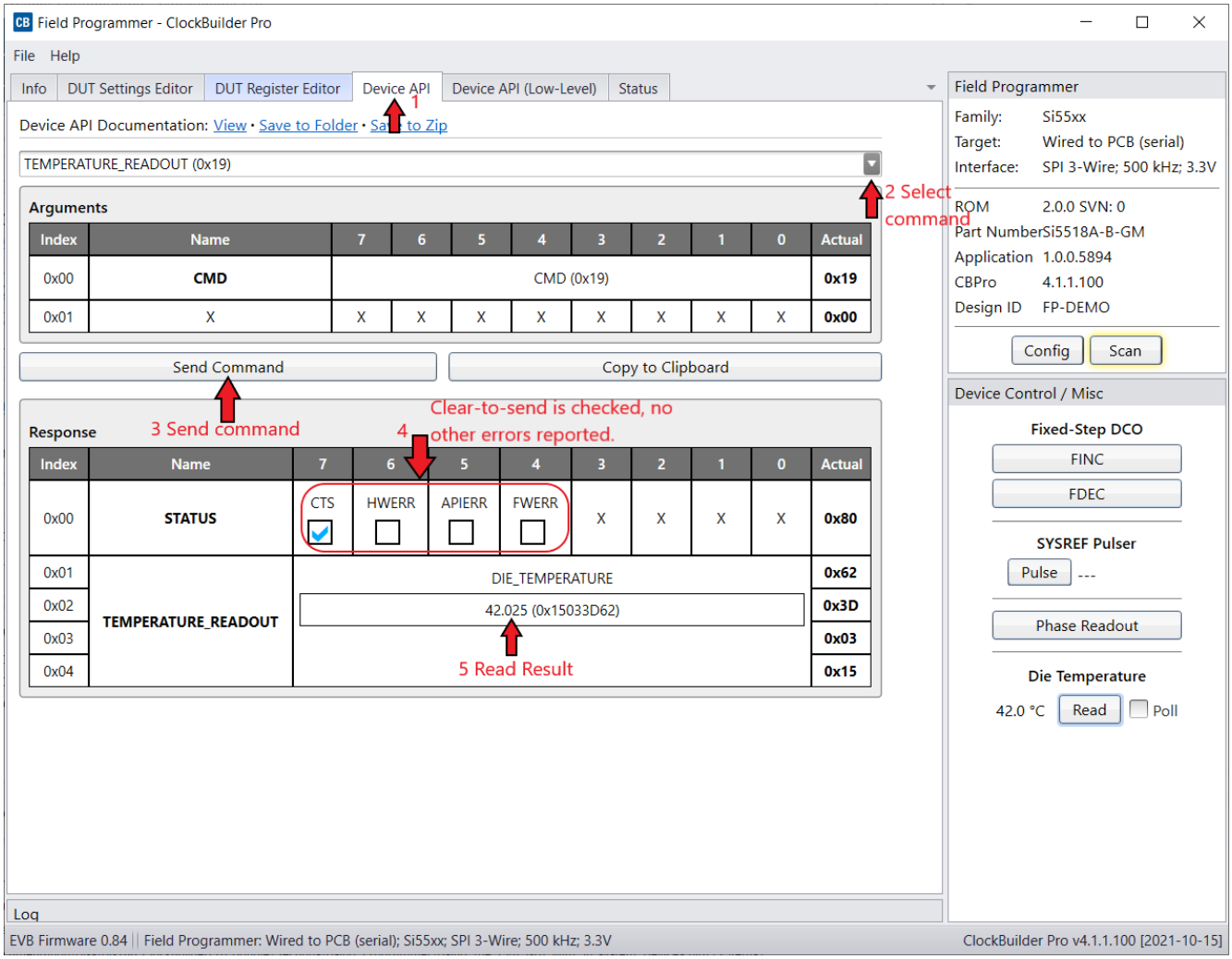


Figure 53. Si55xx API Command Example (Temperature Readout)

The field programmer supports reading and writing individual registers to in-system Si5338/51 family devices. The EVB GUI interface can be used to write exported setting or project files to these devices.

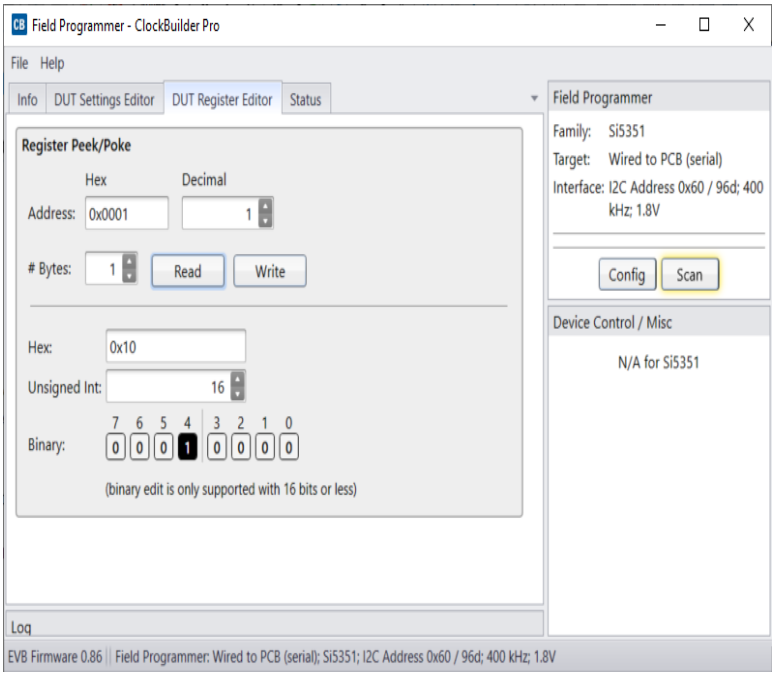


Figure 54. Registers Can be Read/Written on the Si5332/8 and Si5350/1/7 Devices

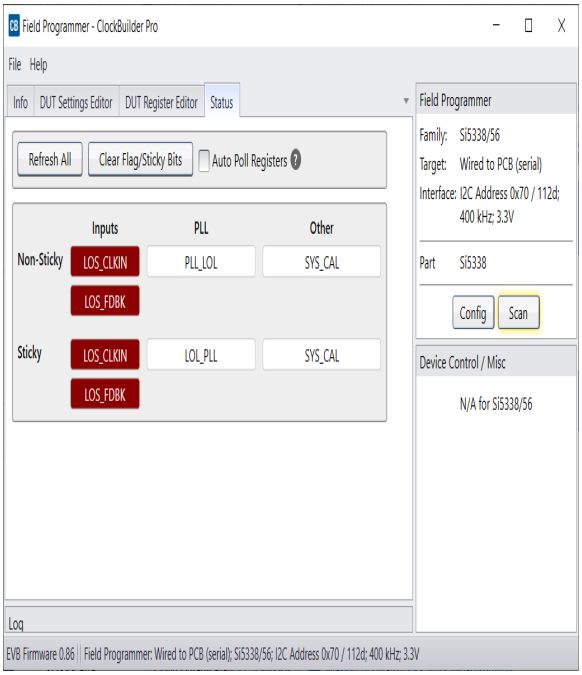
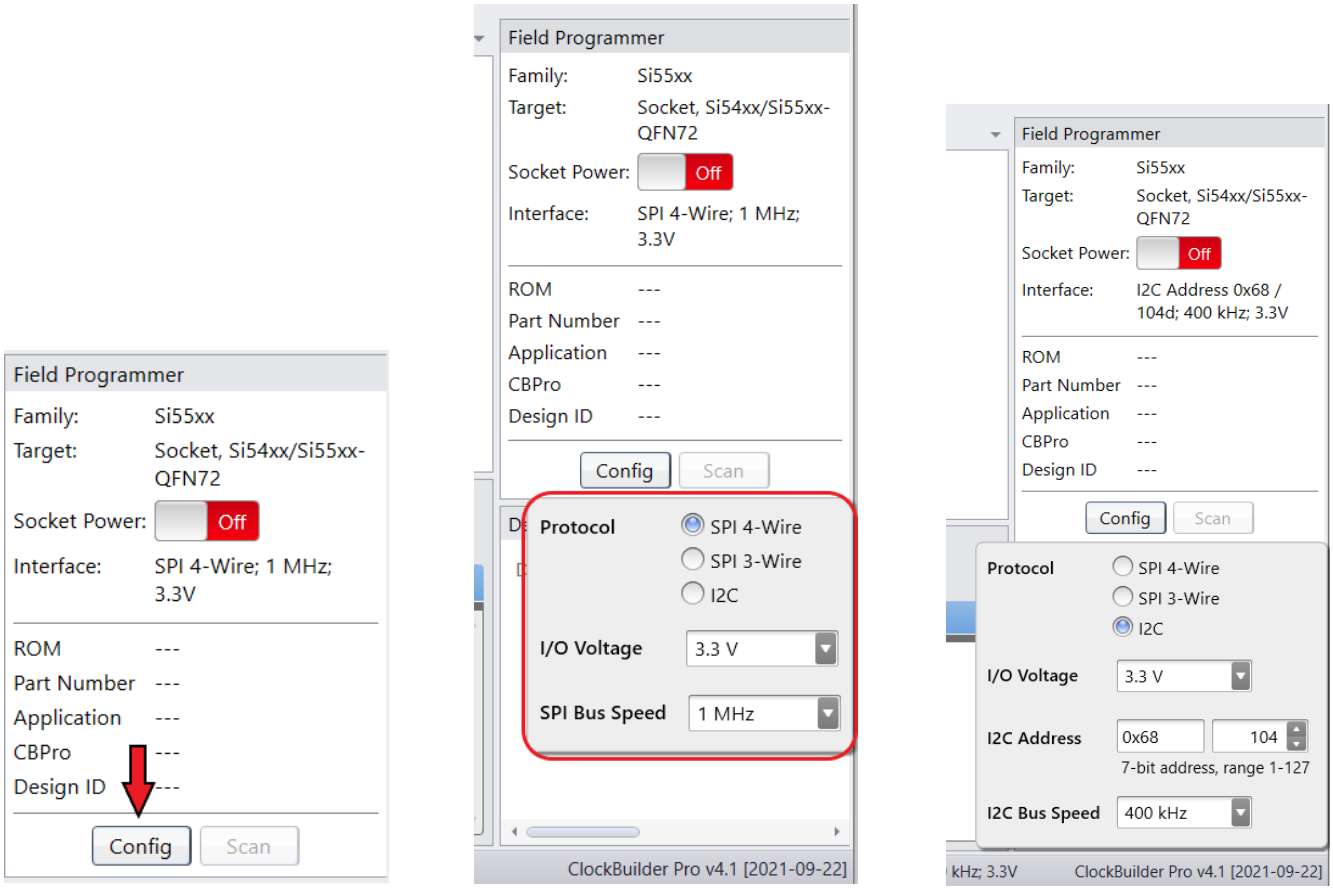


Figure 55. Reading Live Status Bits on the Si5332/8 and Si5350/1/7 Devices

5.3.5. 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 (I^2C), bus speed, and the I/O voltage (non-firmware based devices), as shown in the figure below. For Si536x, Si540x, and Si55xx devices, remember to plug in DC power adapter to the DC jack on the Field Programmer socket board.

Note: For firmware based devices the communication protocol available is I^2C 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 I^2C and the I/O voltage must be selected.



The "Config" button brings up the host interface configuration menu.

SPI Devices Configuration

I^2C Devices Configuration

Figure 56. 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 field. On Si534x/8x/9x parts, 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. On Si536x/540x/55xx parts, the Device API tab can be used to monitor the status of the device.

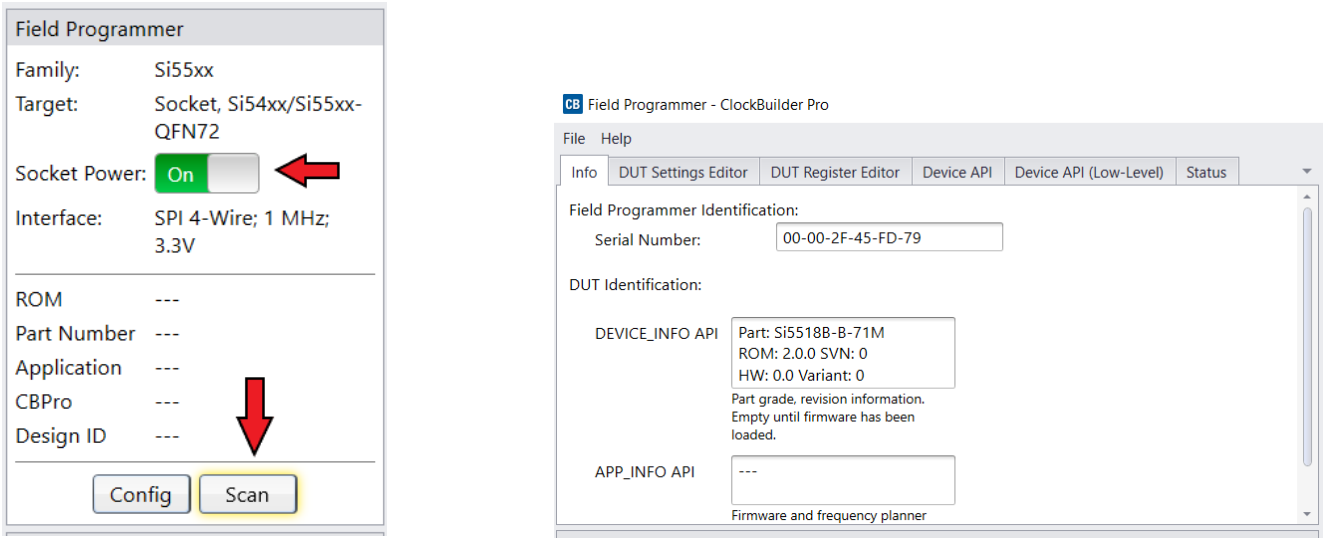


Figure 57. In-Socket Scan Prompt and DUT Info Tab

5.4. Firmware Update

Firmware update of the field programmer is now automatically done if CBPro detects an old firmware version on a connected Field Programmer.

To manually update the firmware at the direction of Skyworks Support, follow the instructions below. The firmware of the field programmer must be updated to be used with the Si536x, Si540x, and Si55xx devices.

5.4.1. Procedure for Updating the Field Programmer Firmware

1. Plug in the Field Programmer to a PC, with the latest version of ClockBuilder Pro installed.
2. Search for and open the "EVB, Field Programmer Flash Utility". This utility is bundled with the regular ClockBuilder Pro installation.

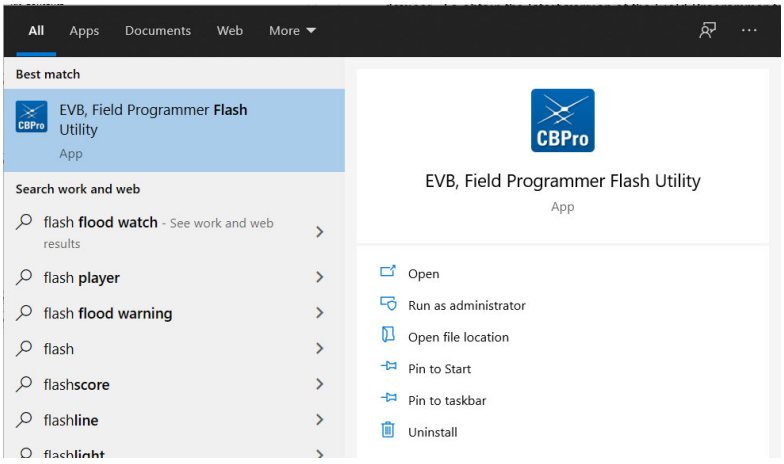


Figure 58. Searching for the Flash Utility

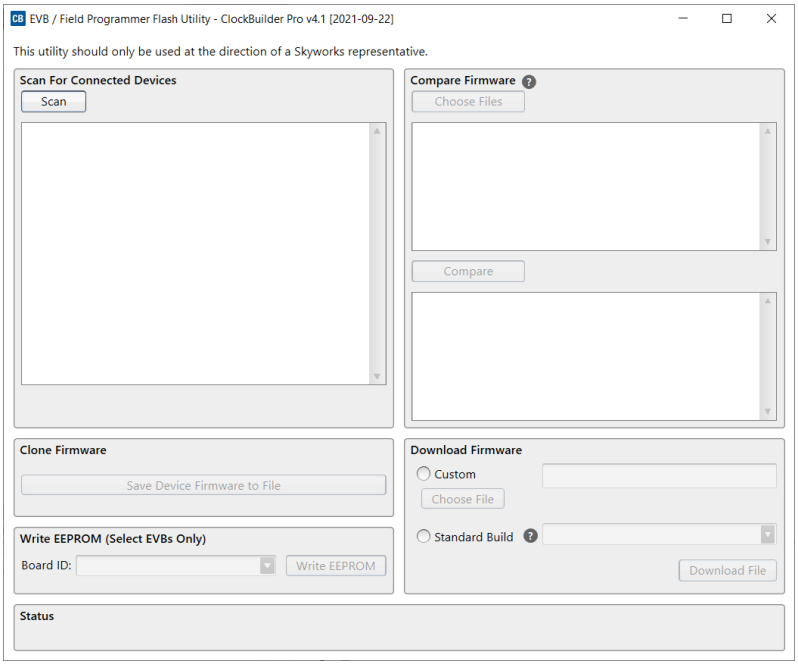


Figure 59. Flash Utility

3. With the Field Programmer plugged in, the Flash Utility open, click Scan. The connected Field Programmer should be displayed. If the scan did not detect a field programmer, make sure you have all other open CBPro windows closed, disconnect and reconnect the Field Programmer USB connection, then try scanning again.

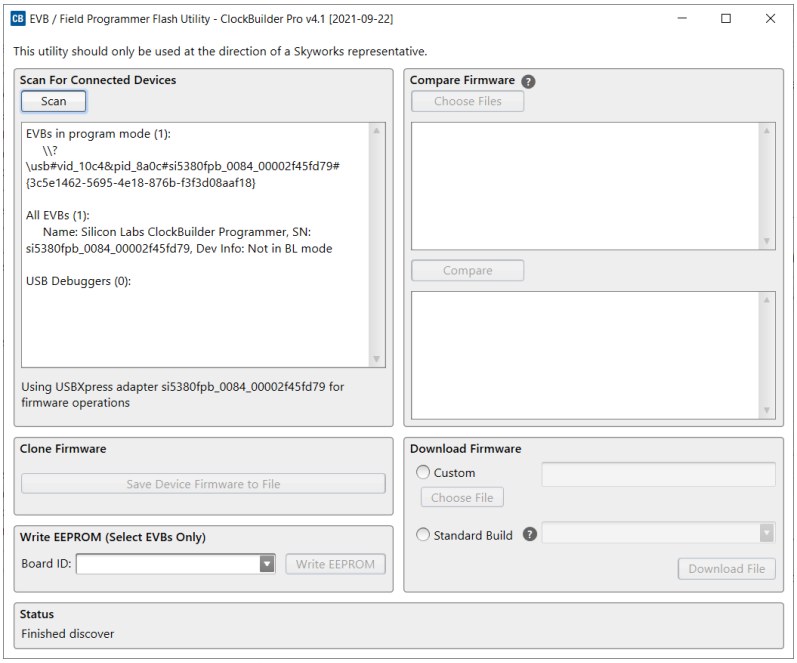


Figure 60. Field Programmer Successful Scan

4. Select the Standard Build option, then find the latest firmware image to flash to the Field Programmer. Select the latest “cbpro_fp_v#.##.hex”, where “v#.##” is the version number. In this example, the version number is v0.85.

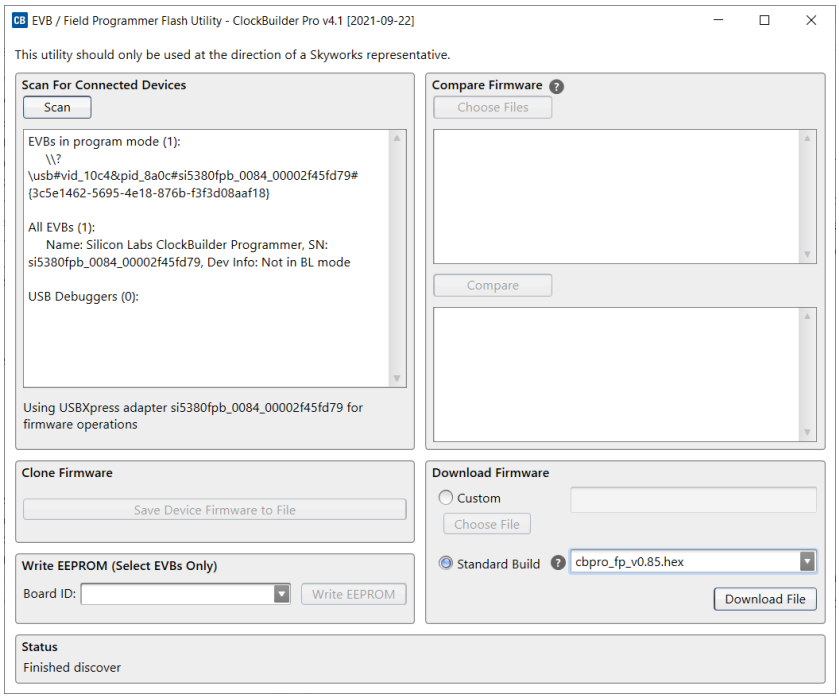


Figure 61. Select the Latest Field Programmer Firmware Image

5. Click “Download File” and wait for the operation to complete.

5.5. Obtaining a Debug Log for Skyworks Support

In case Skyworks support requests a debug log of a device, follow these steps to obtain the log. The log contains a snapshot of the device volatile and non-volatile states.

5.5.1. Procedure for Updating the Debug Log through the Field Programmer

1. Follow Steps 1–5 in "5.3.4. Using the EVB GUI with In-System Devices" on page 32 to connect to the device under test.
2. After a successful scan, click File, then "Create DUT Dump for Skyworks Support" and wait for the log collection to complete.
3. Save the resulting file on your system, then send it to Skyworks Support.

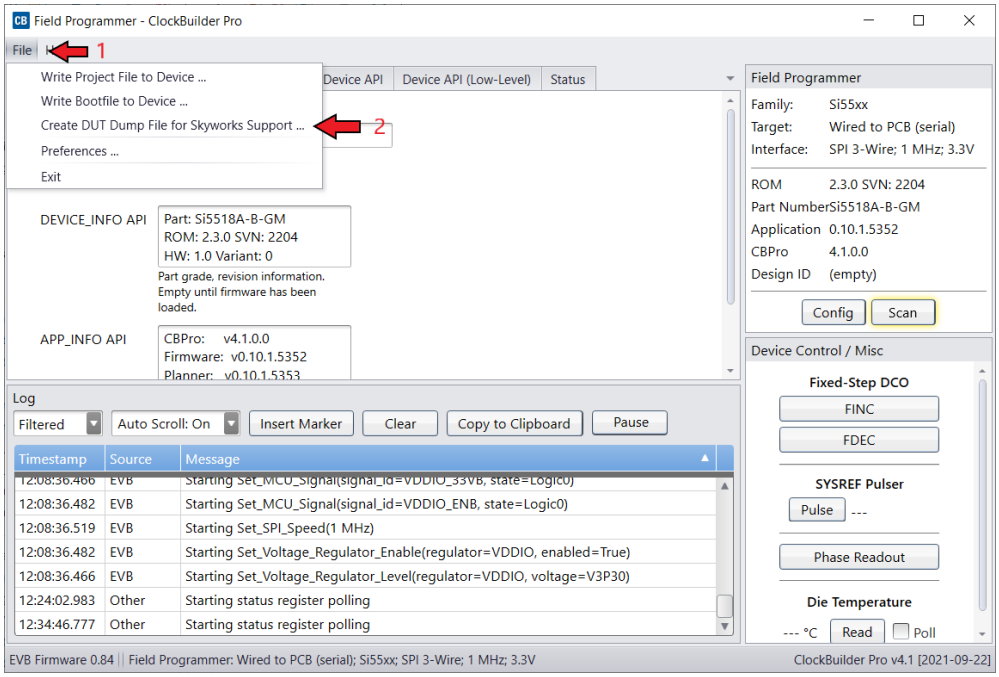


Figure 62. Accessing the DUT Dump Tool

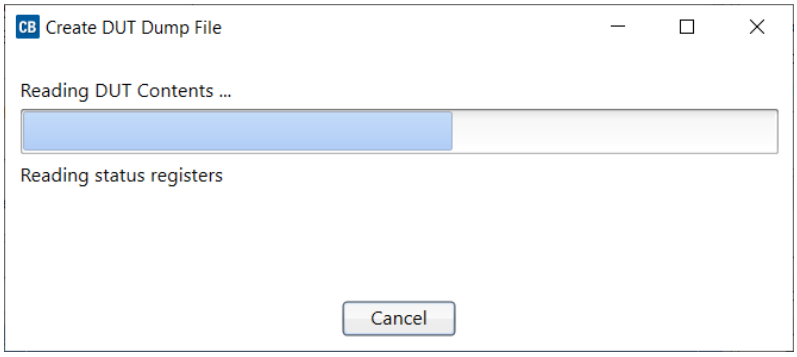


Figure 63. Waiting for the Log Collection to be Complete

6. Appendix A. Troubleshooting

6.1. Why Can't I Communicate with the Device on My Hardware Using the CBPro Field Programmer?

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

6.1.1. General Steps for Resolving Communication Issues

1. Verify which communication protocol your hardware is using – SPI or I^2C. If using SPI, check whether it is 3-wire or 4-wire SPI.
2. For Si534x/8x/9x devices - if using I^2C, use a multimeter and measure the voltage on the I2C_SEL control pin on the DUT. The voltage 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 I^2C. I2C_SEL is not a pin on the Si536x/Si540x/Si55xx devices, so this point can be ignored for those devices.
3. For Si534x/8x/9x devices, 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. For Si536x/Si540x/Si55xx devices, check that the communication protocol selected under the HOST INTERFACE page in CBPro matches the protocol the host (master) device is using.
5. For Si534x/8x/9x devices, if the communication protocol is I^2C, verify the I^2C 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 I^2C address. If these are connected to the CBPro Field Programmer, they are both driven low.
6. For Si536x/Si540x/Si55xx devices, make sure the regulators used to power the device can supply enough current. On the evaluation boards, make sure the external 5VDC adapter is plugged in.

6.1.2. General Steps for Resolving Communication Issues (Si5383/4/8/9)

1. Verify the I^2C 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 I^2C address. If these are connected to the CBPro Field Programmer, they are both driven low.

6.1.3. Communication Error Using the Design Dashboard Window

If the design dashboard experiences an error communicating the device, the following error window will appear. Errors communicating to the device can be due to the device not being powered on, or the selected communication protocol does not match what is expected by the device.

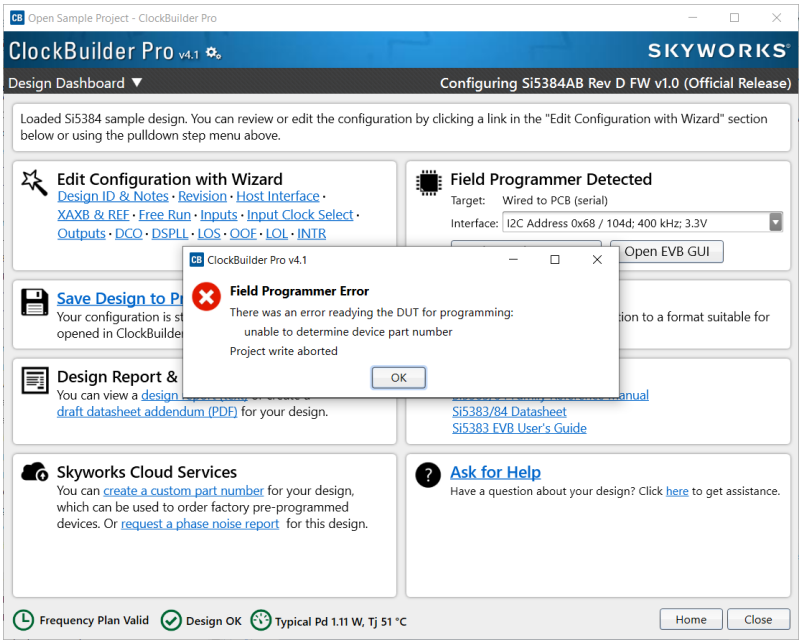


Figure 64. Communication Error Using Design Dashboard

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

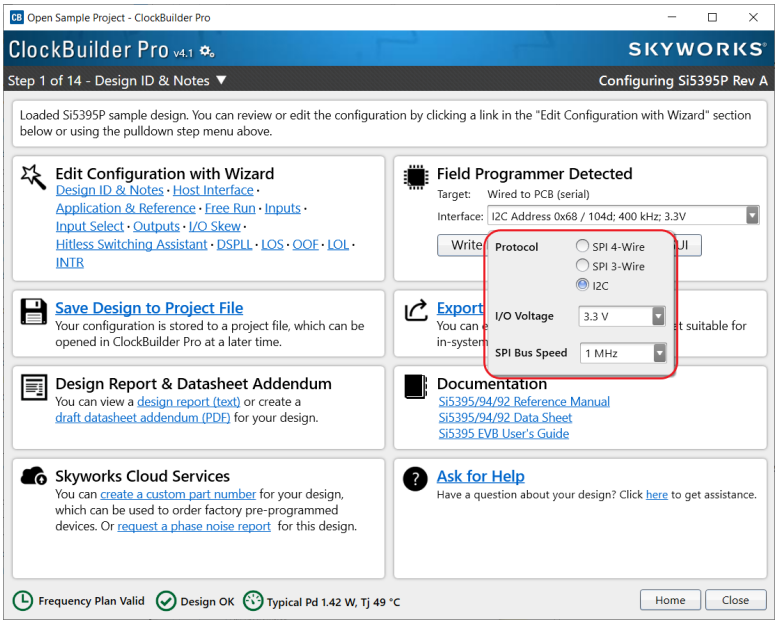


Figure 65. Design Dashboard Communication Error Solution

6.1.4. USB Communication Error

You may encounter a USB related error. To solve this, close out all cbpro instances, unplug the USB cable, and re-open CBPro and plug in the USB cable once CBPro is loaded.

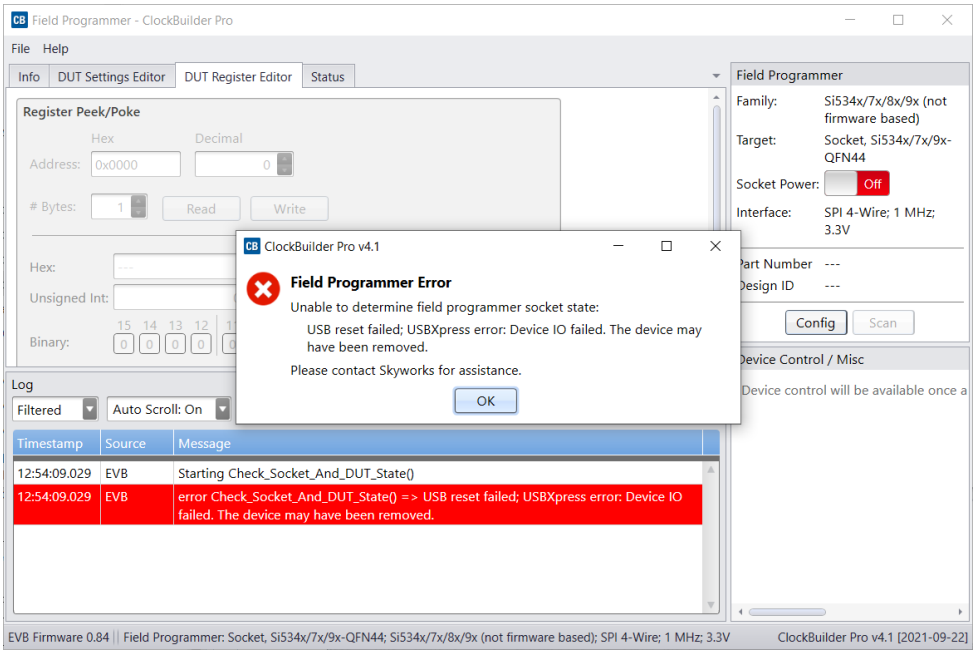


Figure 66. USB Error Message

6.1.5. Communication Error Using the Burn NVM Window

The following window shows a communication error in the NVM Burn window. This error can appear after the Scan for Device button is pressed if the incorrect communication protocol was selected.

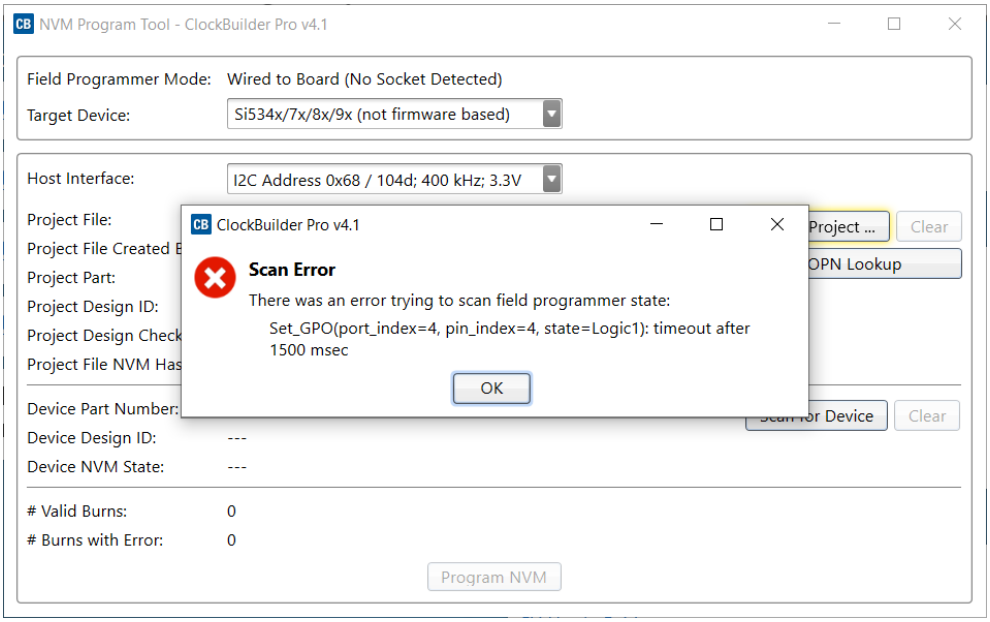


Figure 67. Burn NVM Error Message

To address the error, select the correct communication protocol the device is configured for. If I^2C is used, check all the addresses between 0x68 and 0x71 (the allowable range of addresses set via hardware pin). It is also possible that the I^2C address was programmed to a number outside the normal range using the I^2C Address burn tool. The following window shows the dropdown used to adjust the communication settings to resolve communication error.

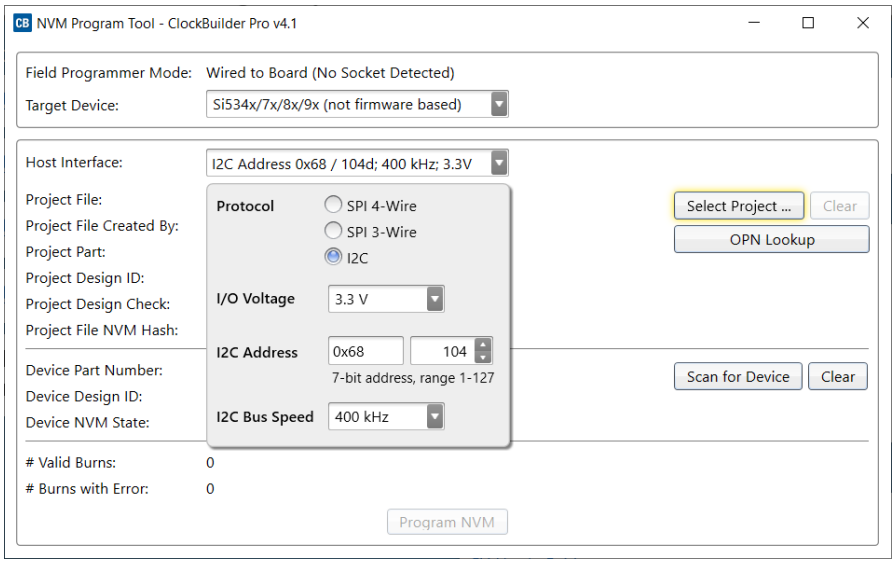


Figure 68. Burn NVM Error Message Solution

6.1.6. Communication Error Using the EVB GUI Window

The following window shows an example of the error produced when the EVB GUI experiences an I^2C error.

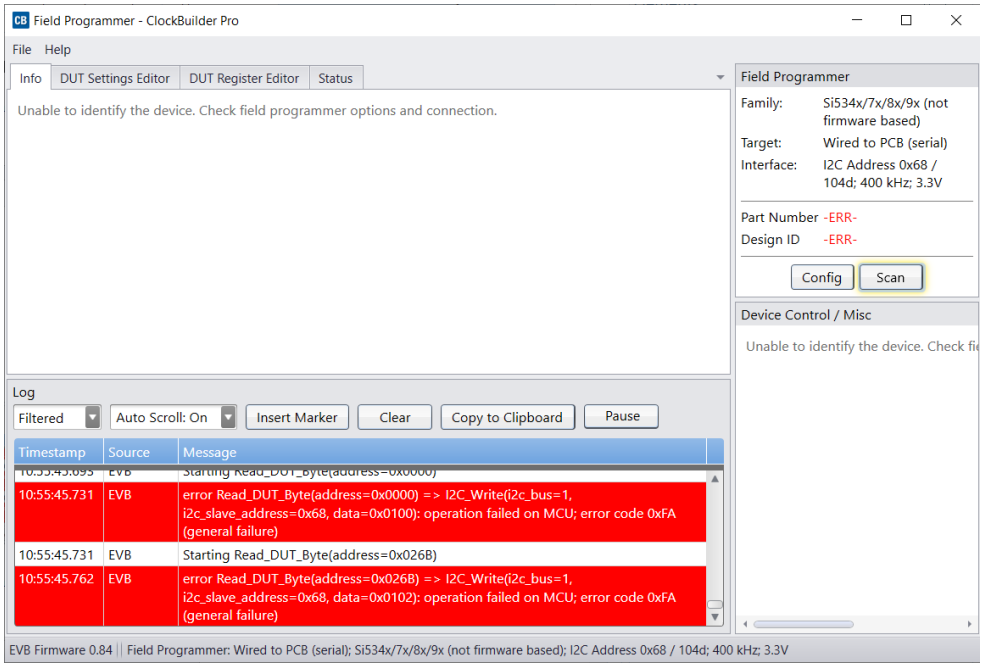


Figure 69. EVB GUI I^2C Error

The following window shows an example of the error produced when the EVB GUI experiences an SPI error. An error in communicating through SPI can occur due to many things, including:

- Incorrect connection of SPI wires.
- SPI host is communicating at a voltage level not expected by the clock device.
- I^2C pullup resistors being present on a custom board.
- Long traces or heavy capacitive loads on a custom board leading to errors in high-frequency SPI communications.
- If the Field Programmer is being used with an evaluation board, double check the schematic for any missing (or extra) 0 Ω resistors in the path. On some newer devices, it is possible for a SPI pin to double as a GPIO pin.

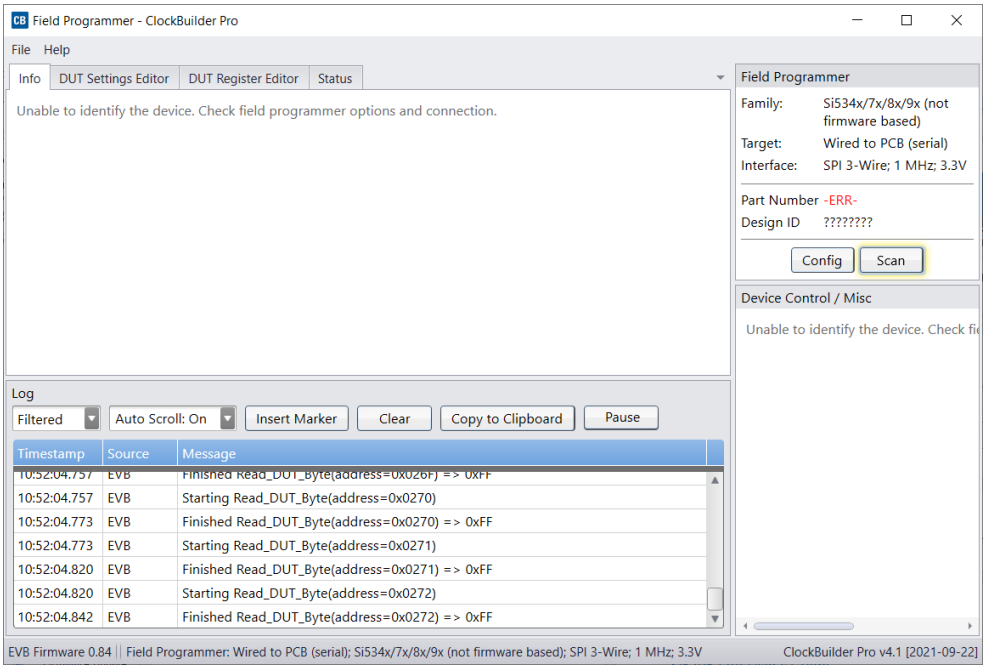


Figure 70. EVB GUI SPI Error

The following window shows how to change the communication settings using the EVB GUI window. Selecting the correct communication protocol, and double checking the signal at the SPI pins with an oscilloscope should resolve the -ERR- message.

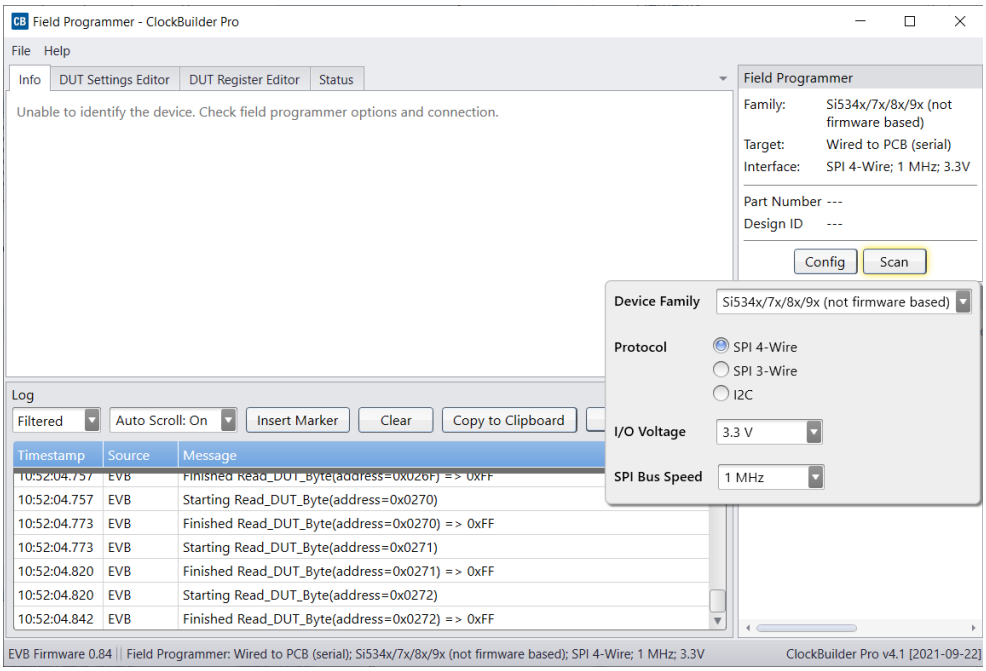


Figure 71. EVB GUI Solution

6.2. Why do I Have a Communication Error when I Write My New Project to the Si534x/8x/9x Device?

6.2.1. Description of What Happens when a New Plan Changes the IO_VDD_SEL Bit (Register 0x0943[0]) Value

In order for the CBPro 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. CBPro may force this bit when writing a new plan to a device. If the plan changes this bit during the writing process, communication can fail. To determine if the new plan is changing this bit, perform the following steps:

- Read the current IO_VDD_SEL value in the device by using the DUT Register Editor tab in the EVB GUI window.
- Check if IO_VDD_SEL read above is different from the settings in 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 "6.3. How do I write a project file to the device that changes the I/O Power Supply setting in Si534x/8x/9x devices (IO_VDD_SEL bit)?" on page 52.
  - 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 "6.3. How do I write a project file to the device that changes the I/O Power Supply setting in Si534x/8x/9x devices (IO_VDD_SEL bit)?" on page 52.

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

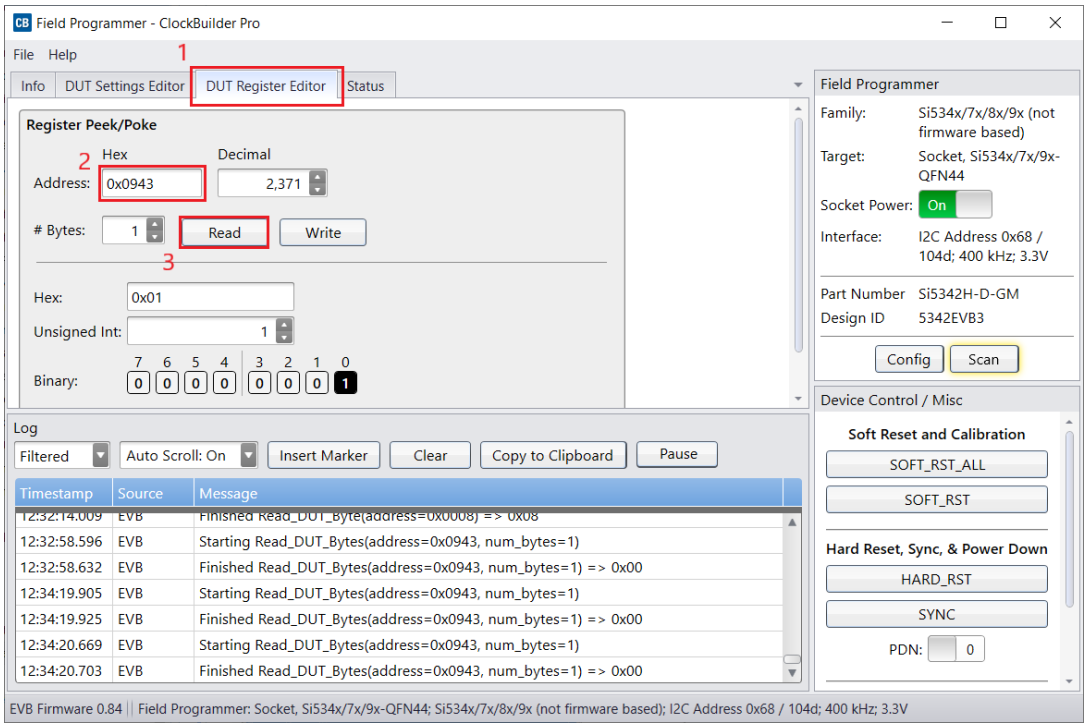


Figure 72. Read IO_VDD_SEL Bit from Device

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.

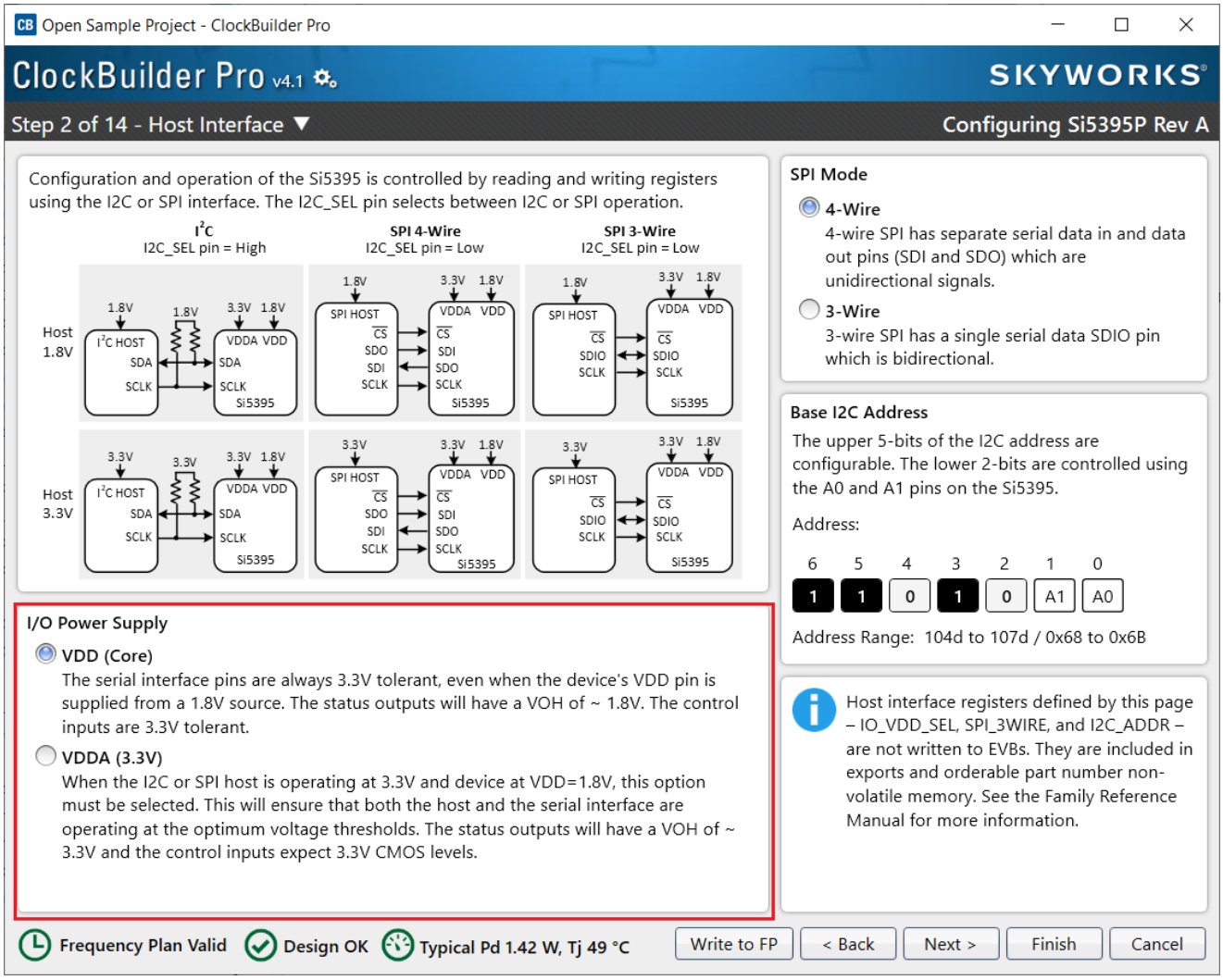


Figure 73. Determine the Value of IO_VDD_SEL Bit Written to Device

6.2.2. Fix: Changing the IO_VDD_SEL Bit

If it is suspected that IO_VDD_SEL has changed after writing a new plan to the device, you will not be able to communicate with the device. However, the device will still respond to a register write request to change the IO_VDD_SEL bit even when communicating with the incorrect IO_VDD. To do this, follow the steps below:

1. Power down and power on the clock device to ensure that the next write operation is the first bus transaction.
2. Write a 9 (0x9) to the page address register (address=0x1)
3. Write a value of 0x00 to register 67 (0x43). Steps 2 and 3 combined will write a 0 value to register address=0x0943.
4. Read back some registers to check if communications is now restored. A suggestion is to read registers 0x02 and 0x03. These registers should be non-zero and correspond to the device part number.

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

6.3.1. 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.

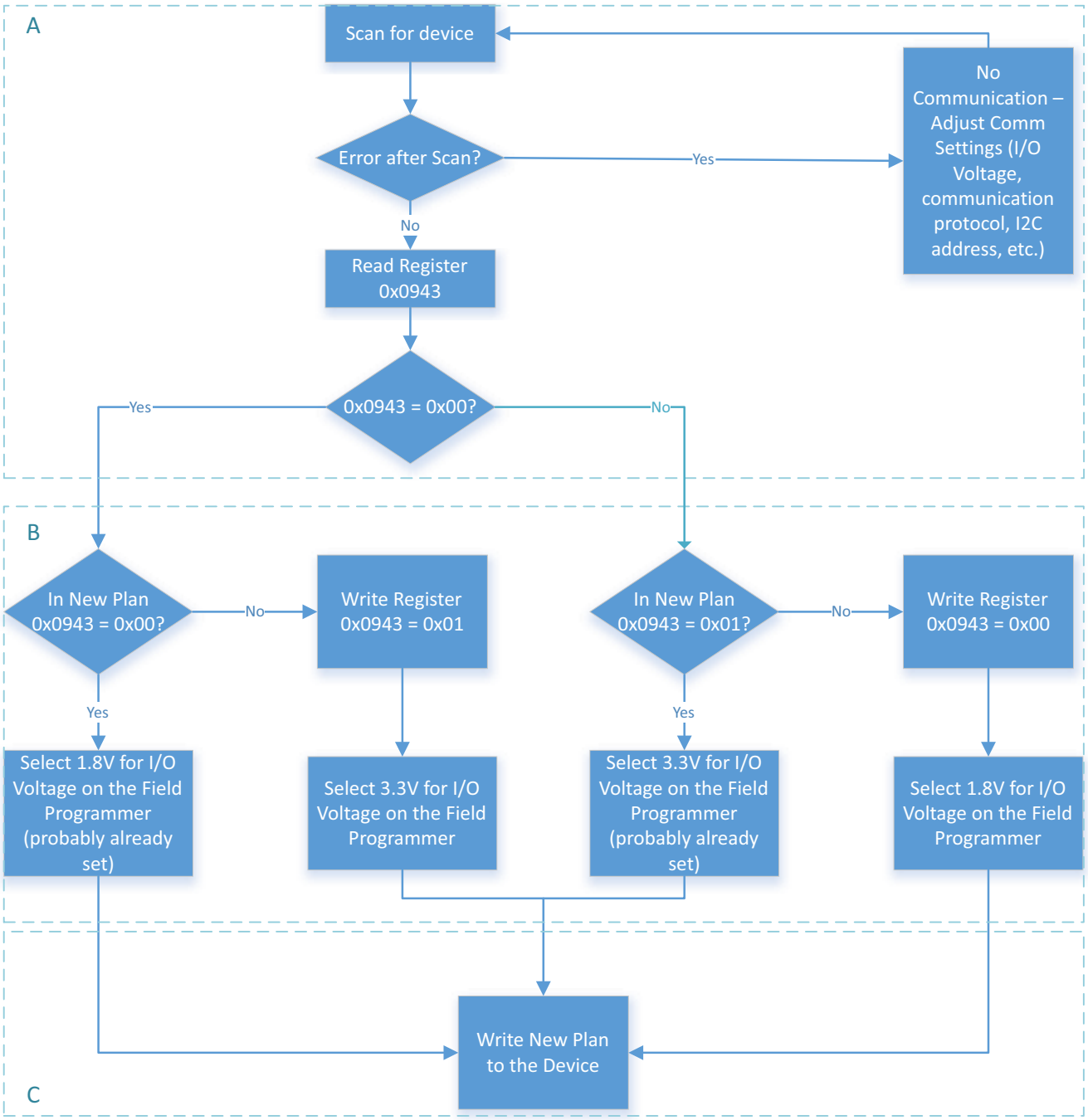


Figure 74. General Steps to Change I/O Power Supply Setting with a Project File

6.3.2. Steps Using CBPro Graphical User Interface

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

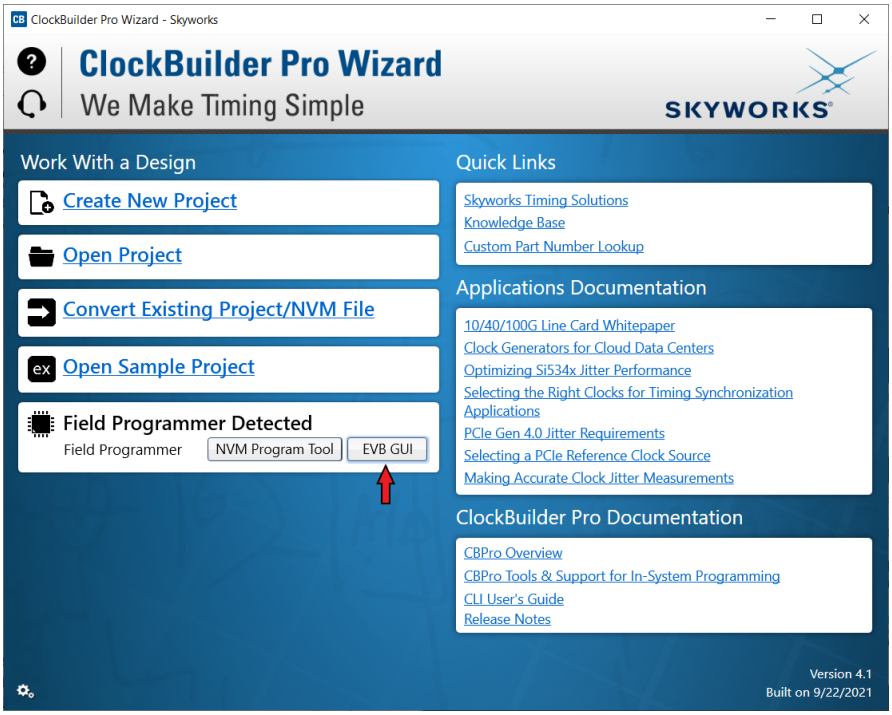


Figure 75. EVB GUI Button

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 are shown in “Figure 76. Configuring Communication Settings” on page 55.

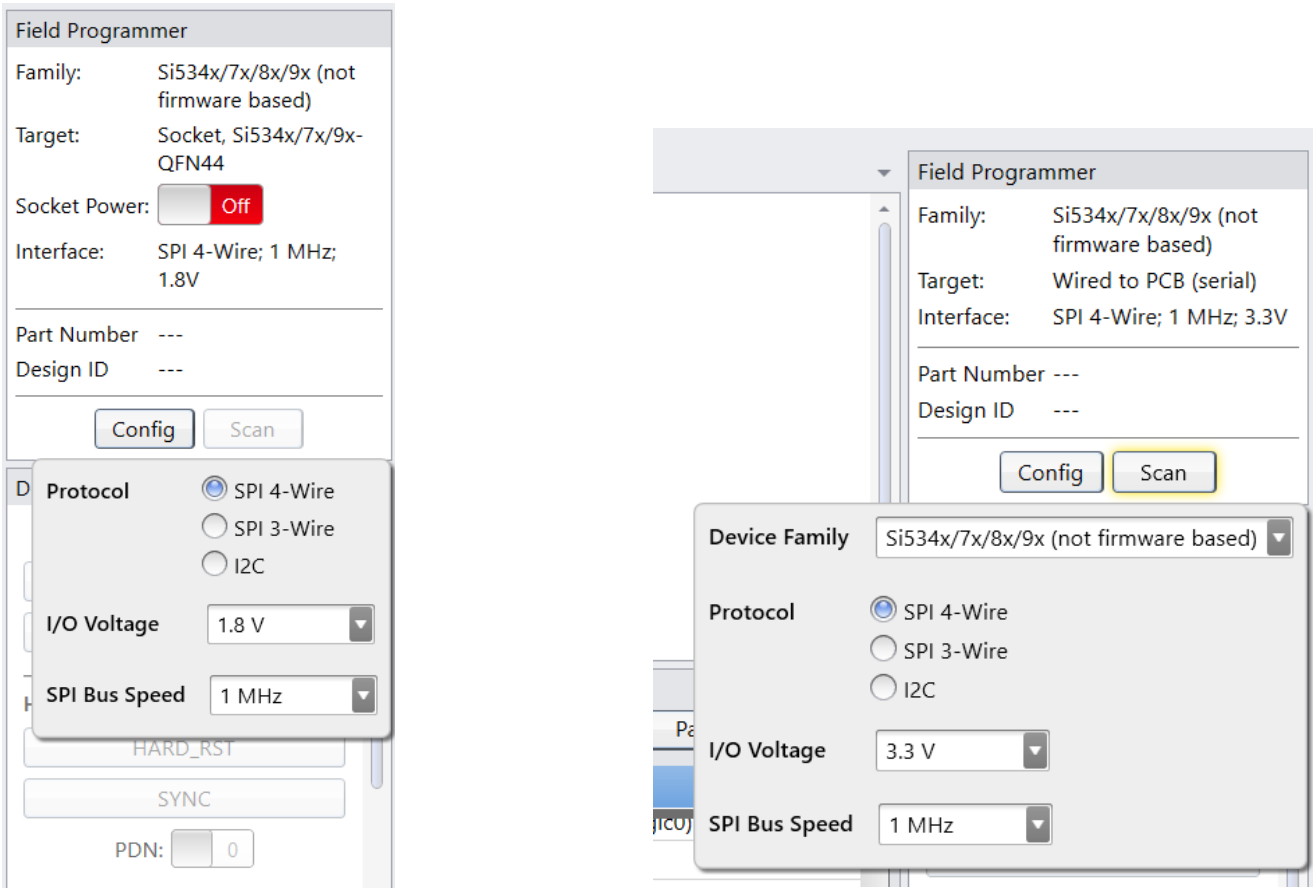


Figure 76. Configuring Communication Settings

Examples of a Communication failure for I²C and SPI:

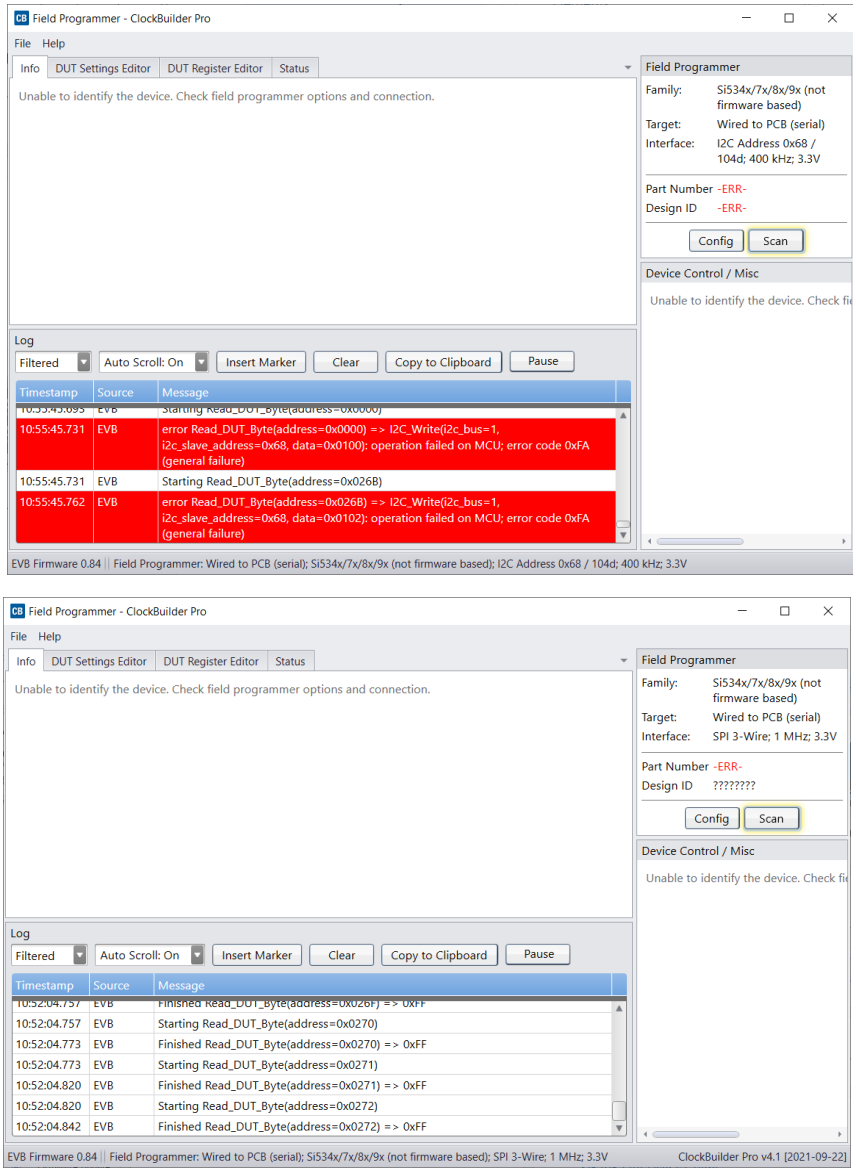


Figure 77. I²C and SPI Communication Failure Examples

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. Reconfigure the communication settings of the field programmer to reestablish communication to the device.

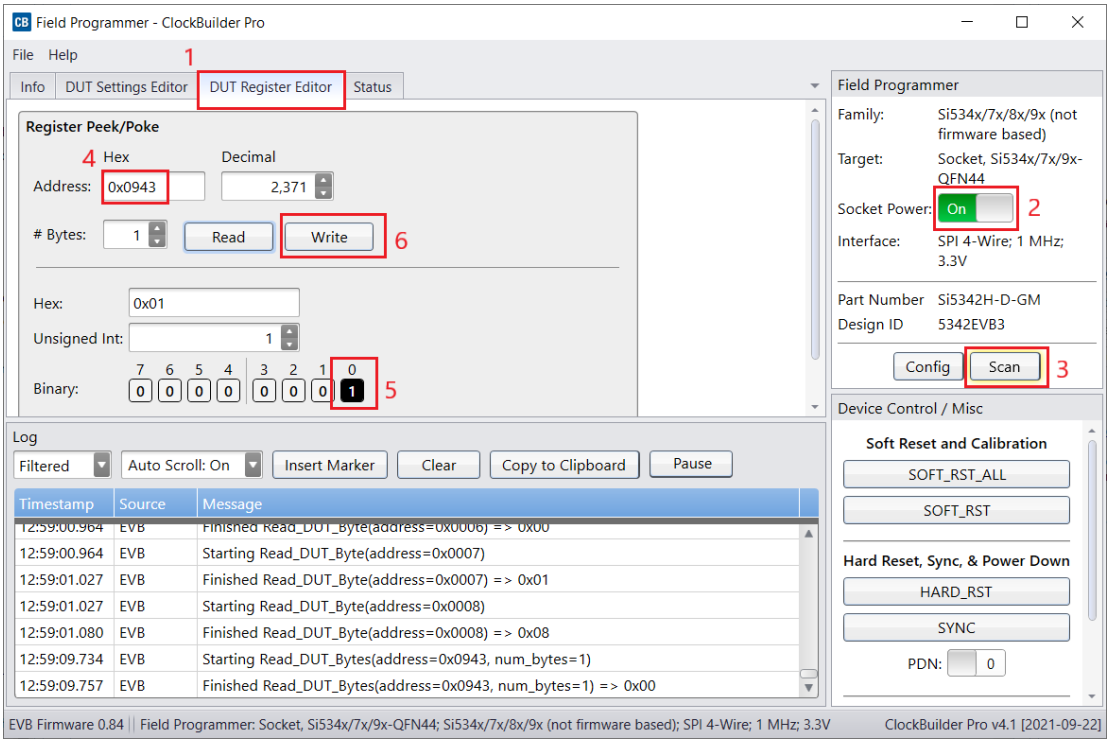


Figure 78. Reconfiguring Communication Settings of the Field Programmer

3. Write your new plan to the device.

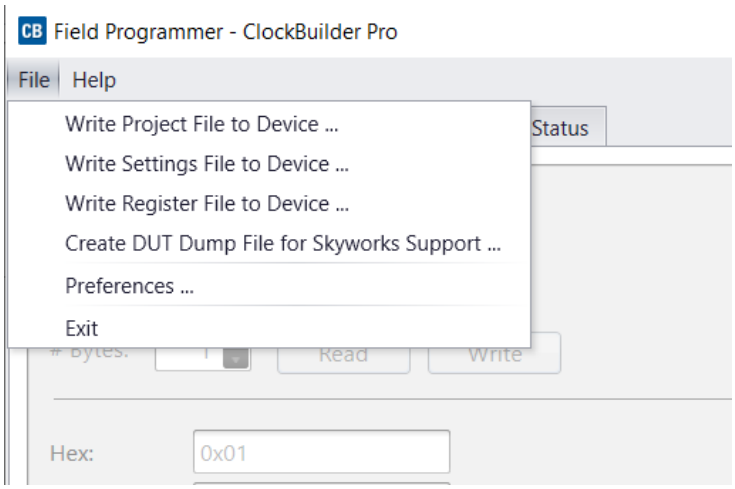


Figure 79. Write New Plan to Device

6.3.3. Steps Using CBPro Command Line Interface

1. Attempt to communicate with the Si534x/8x/9x 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.
   I^2C 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
```

      I^2C 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
```

   I^2C 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.

6.4. 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 Figure 80.

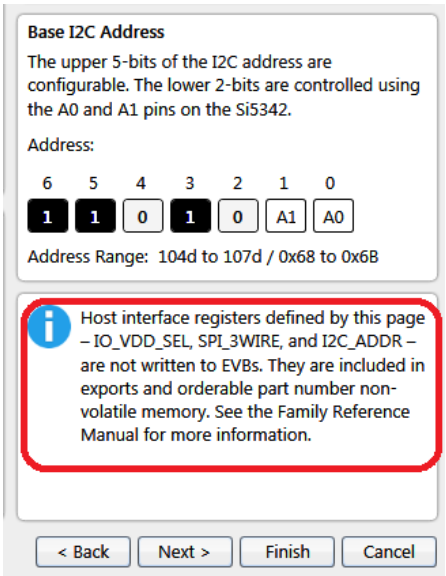


Figure 80. Base I²C Address

To permanently change the I²C base address on your device, use the I²C Address Burn Tool. See Figure 81 below.

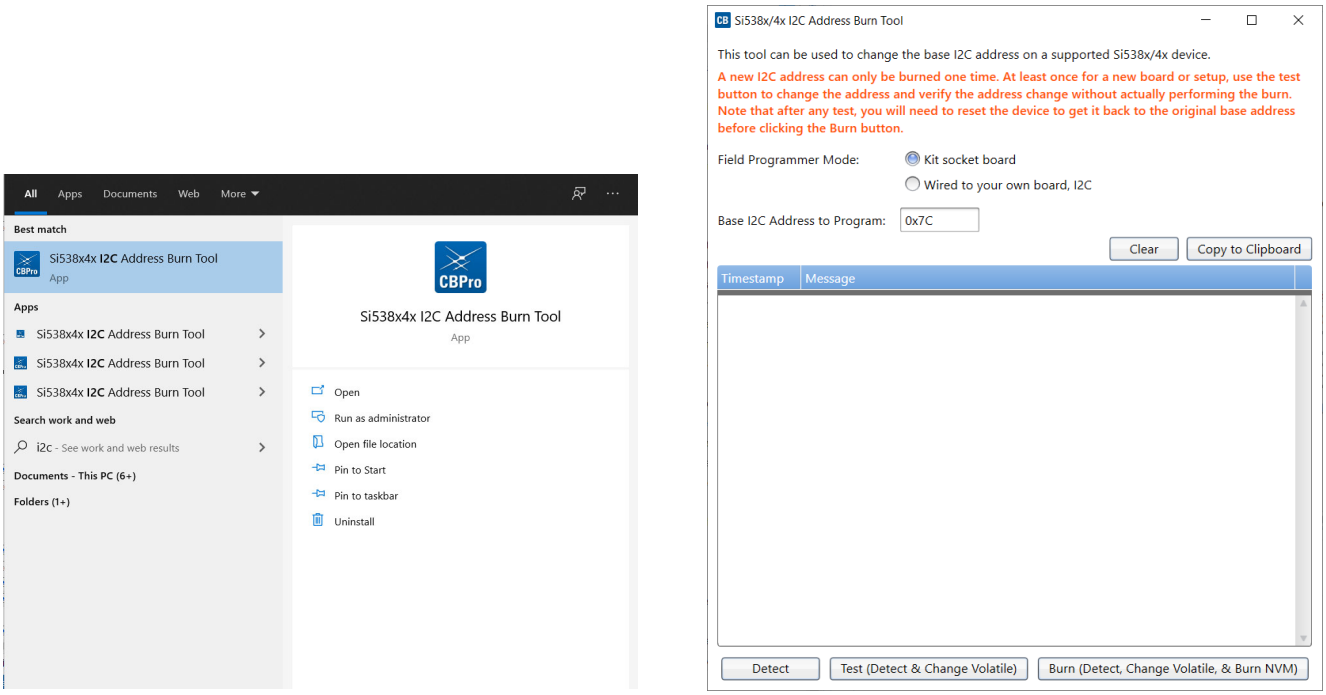


Figure 81. I²C Address Burn Tool

7. Appendix B. Field Programmer Schematic

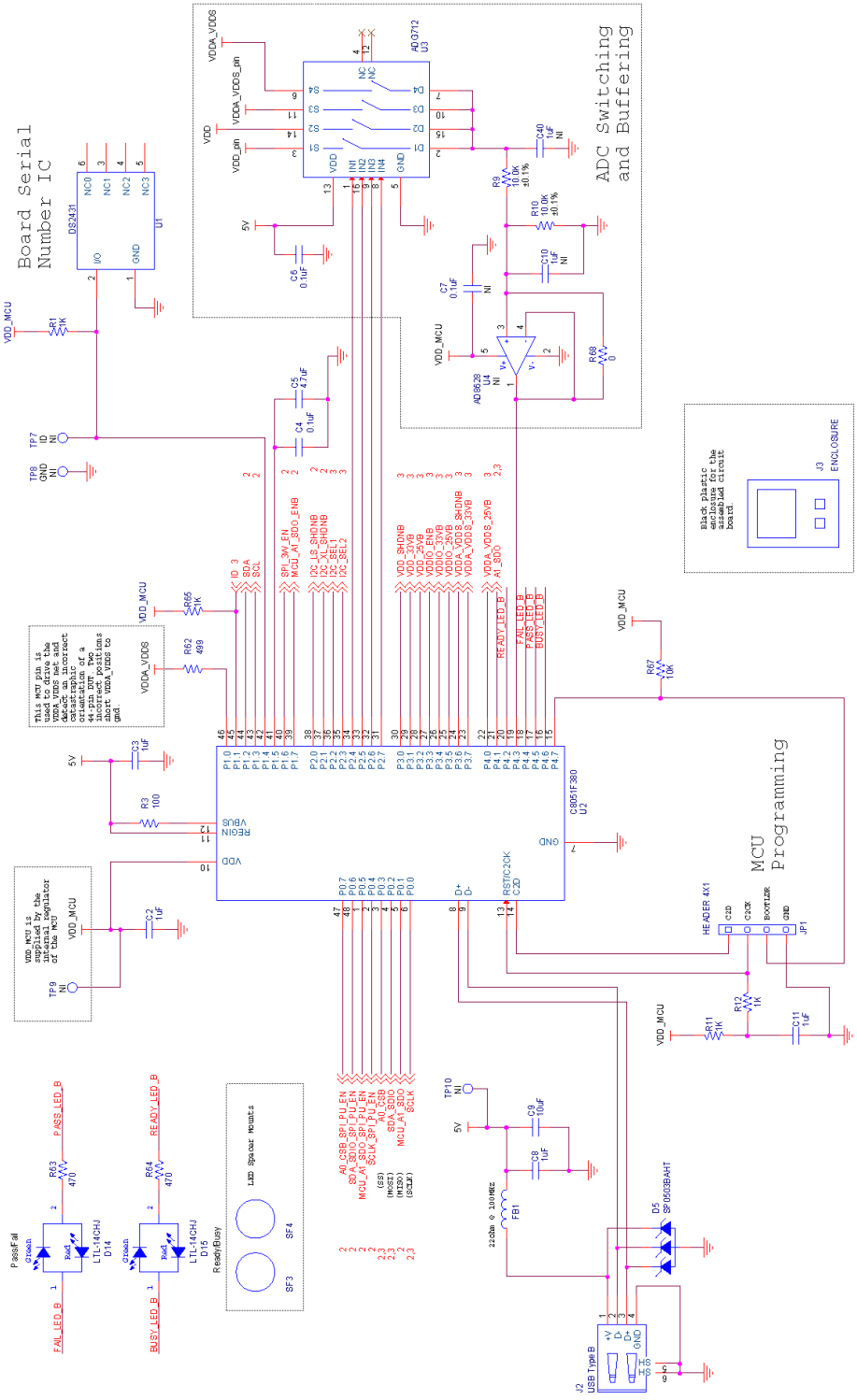


Figure 82. CBPROG-DONGLE Schematic (1 of 3)

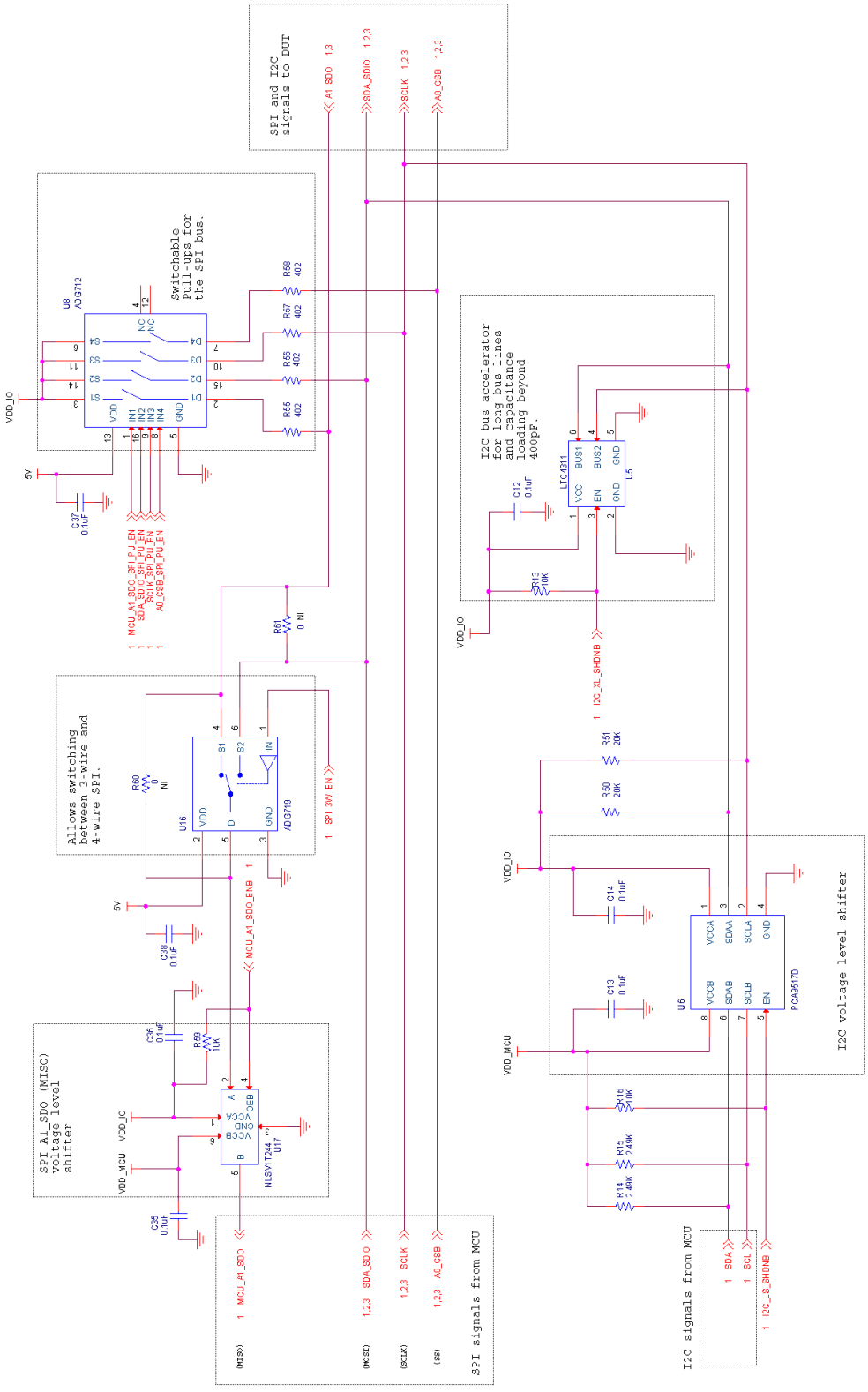


Figure 83. CBPROG-DONGLE Schematic (2 of 3)

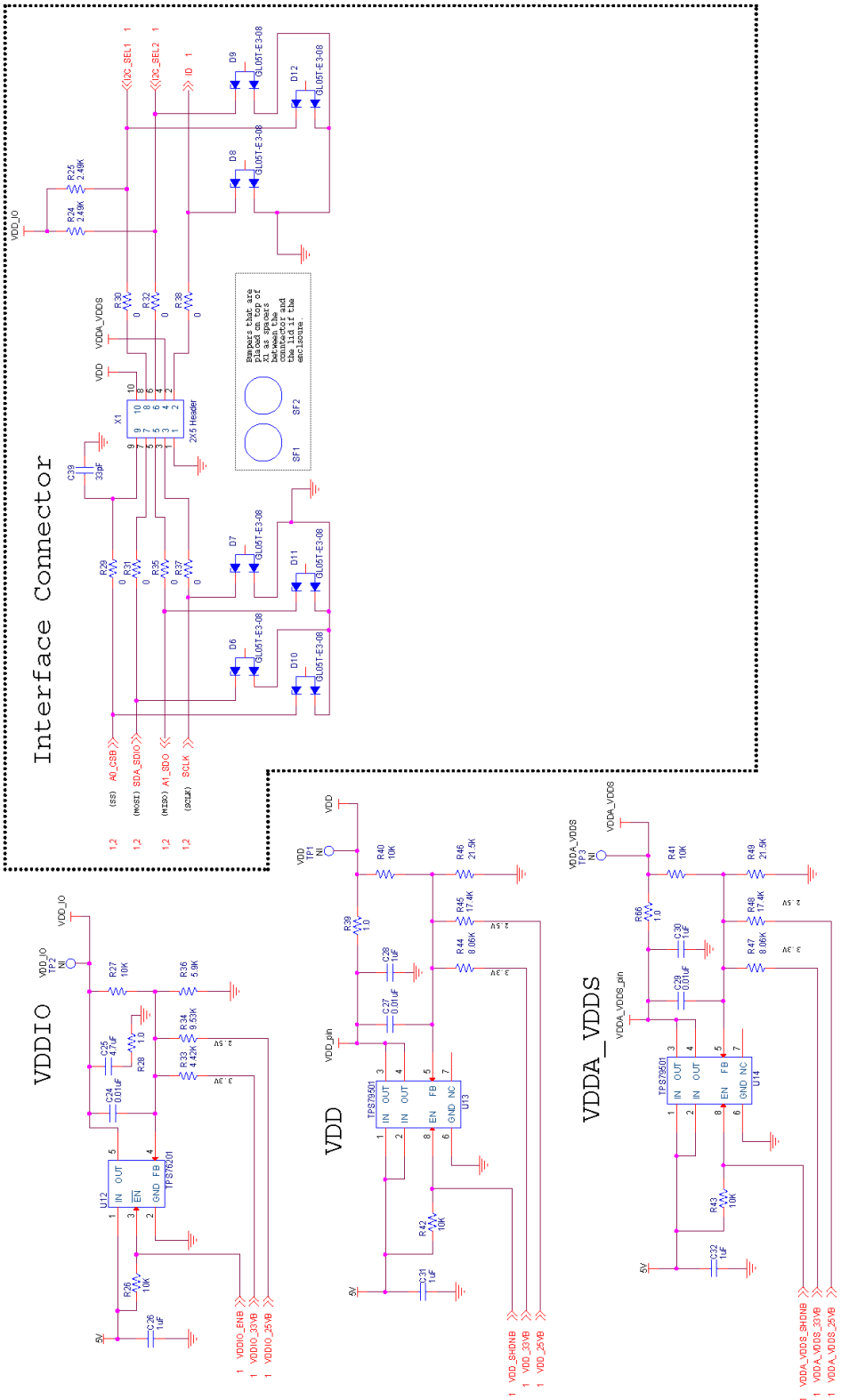


Figure 84. CBPROG-DONGLE Schematic (3 of 3)

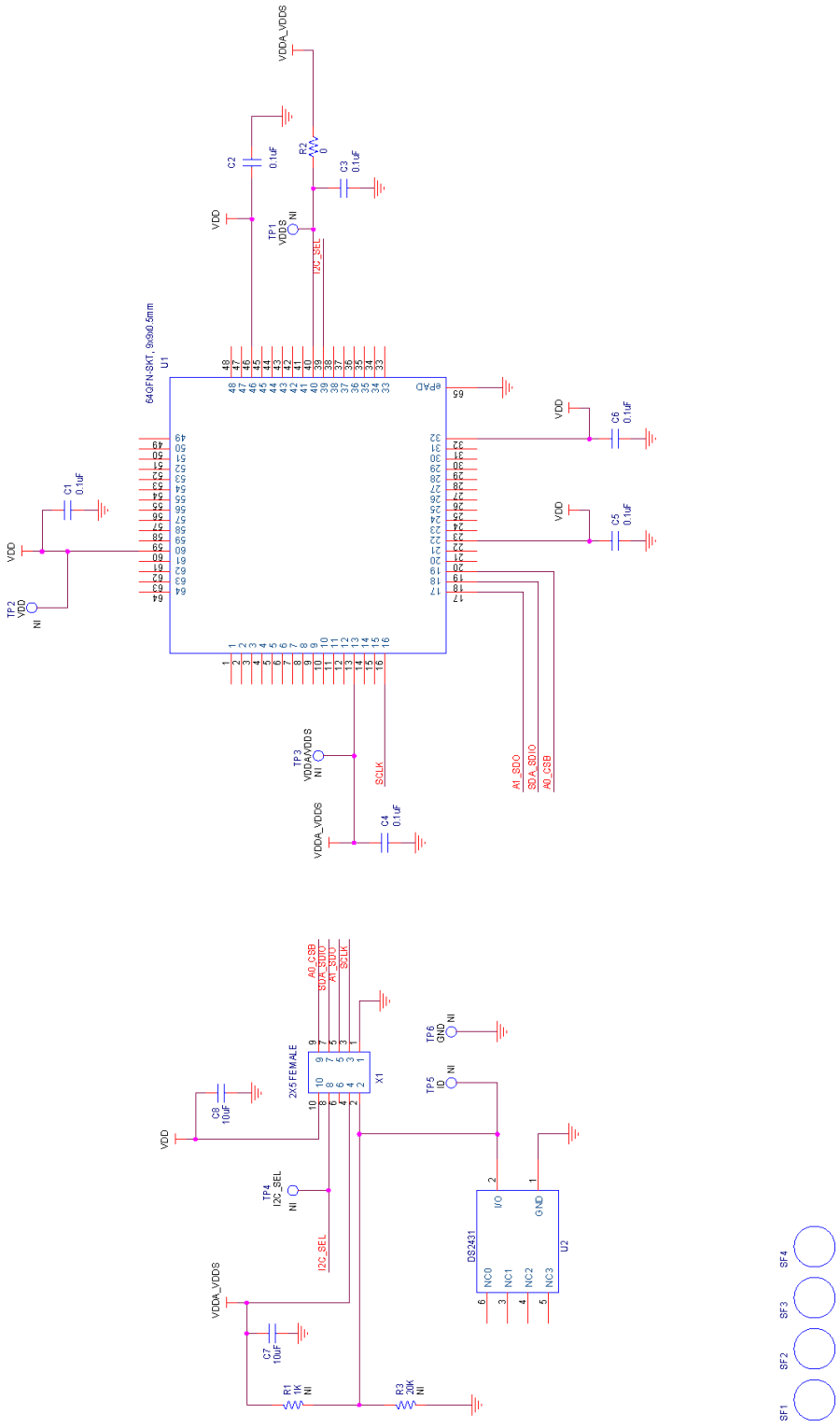


Figure 85. 64-Pin Socket Board Schematic

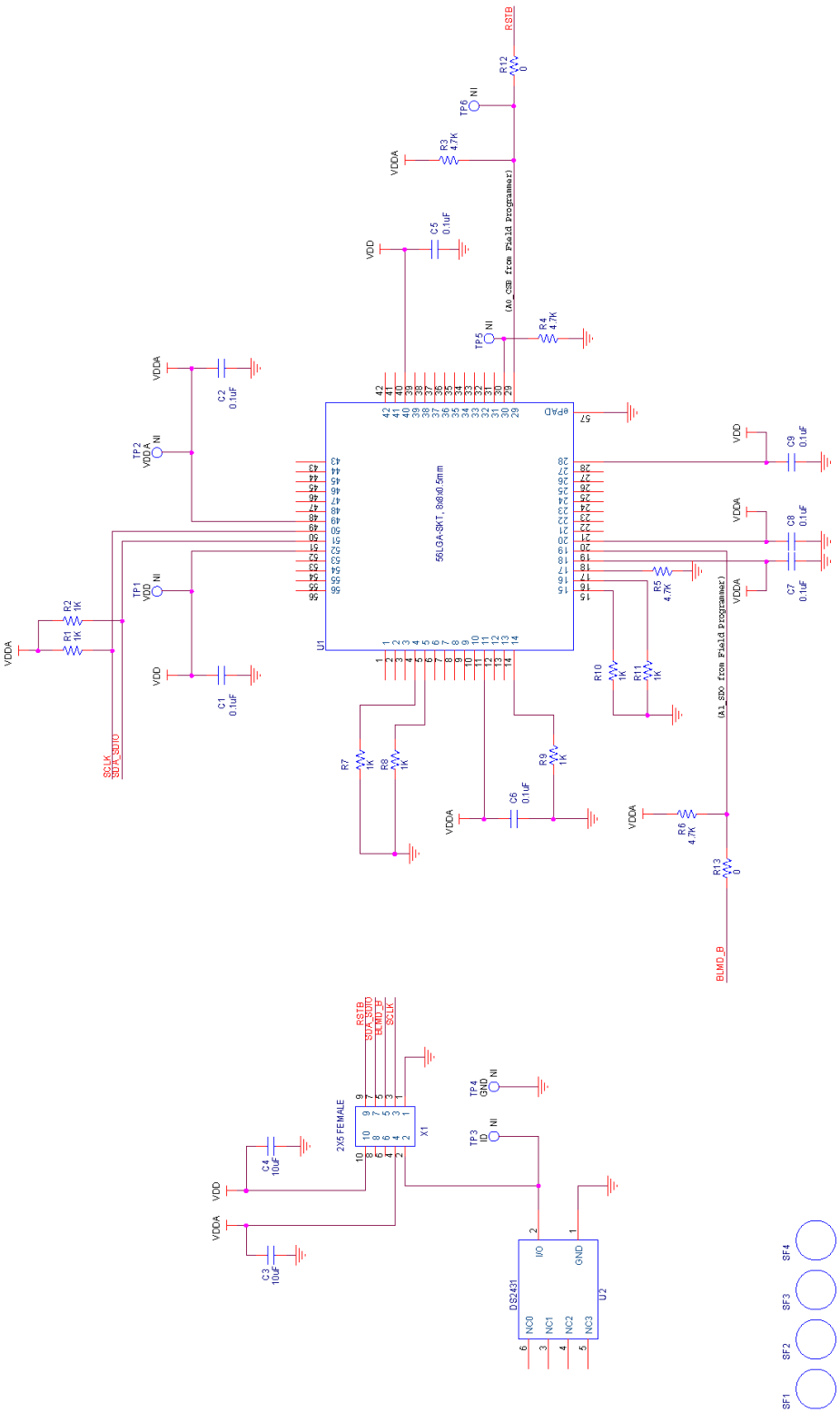


Figure 86. 56-Pin Socket Board Schematic

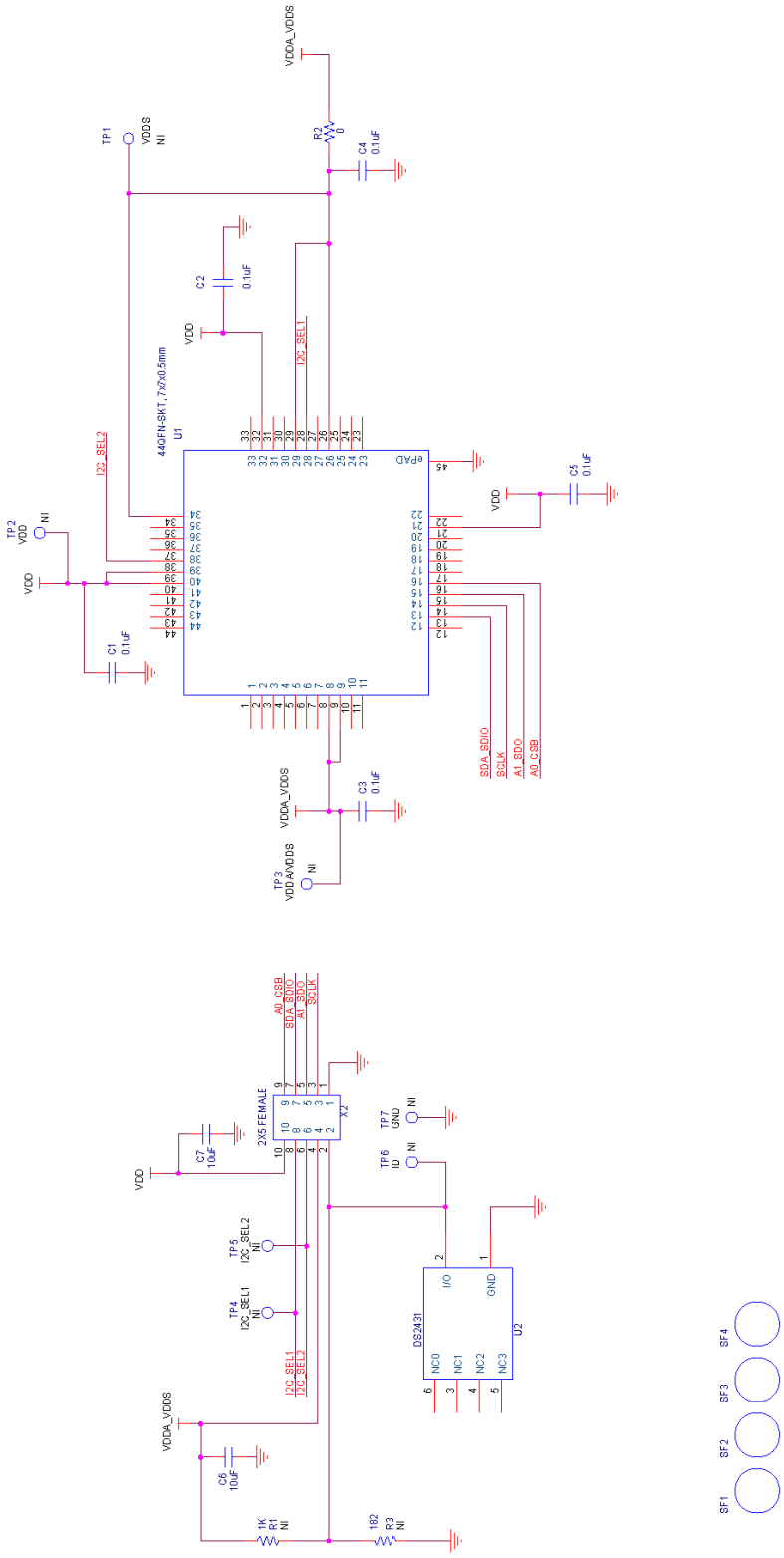


Figure 87. 44-Pin Socket Board Schematic

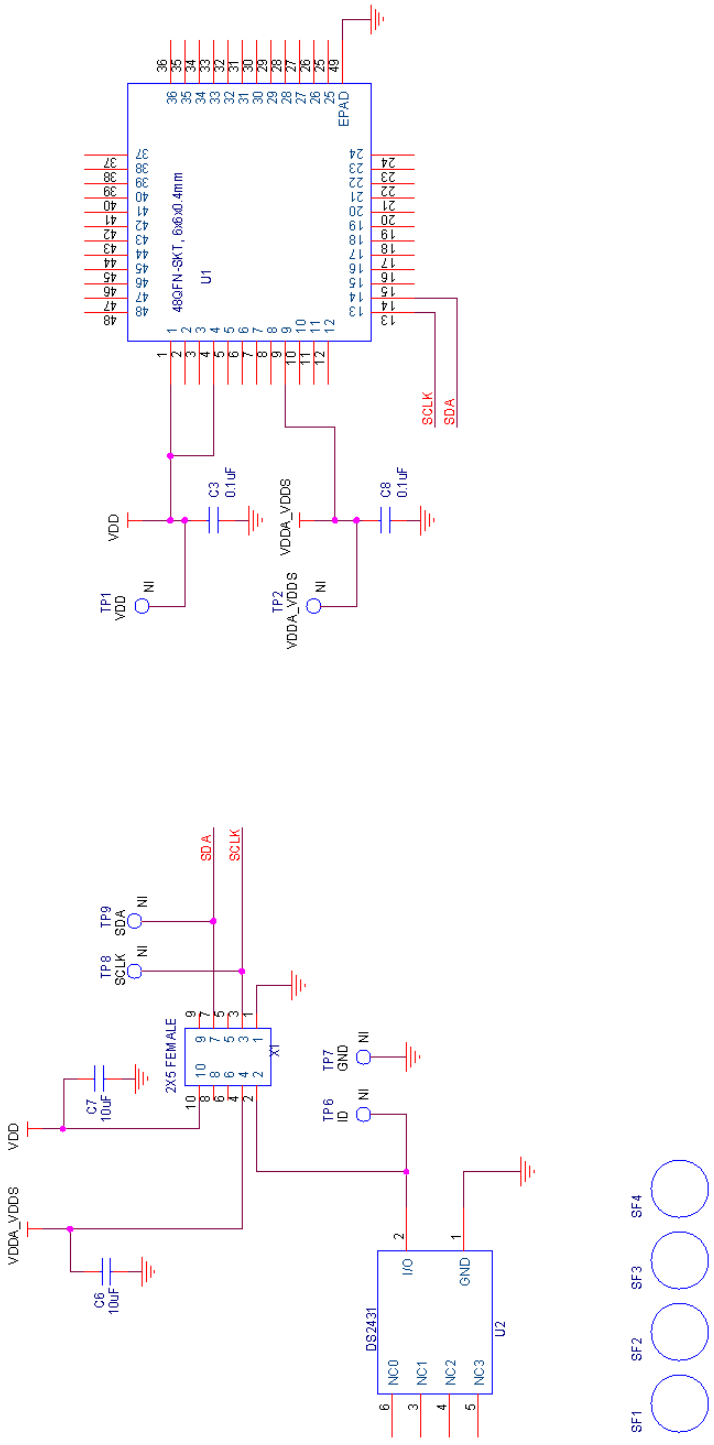


Figure 88. Si5332 48-Pin Socket Board Schematic

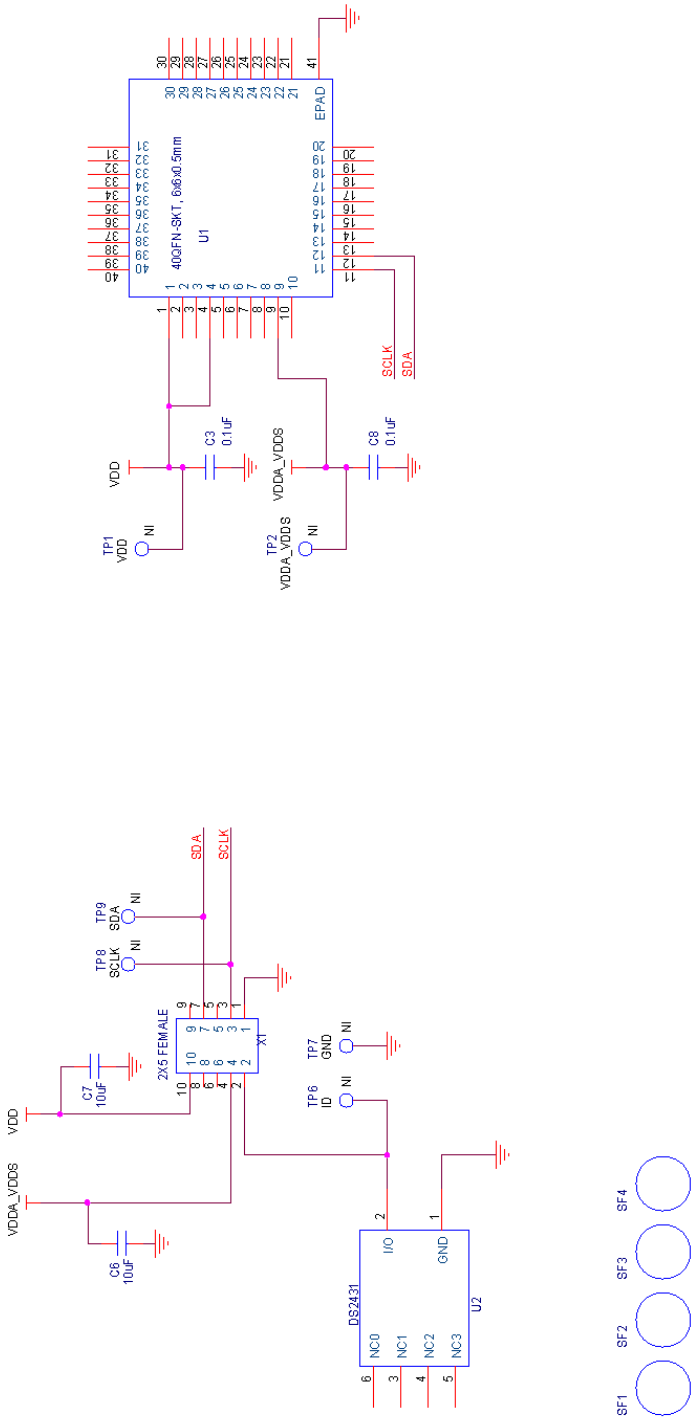


Figure 89. Si5332 40-Pin Socket Board Schematic

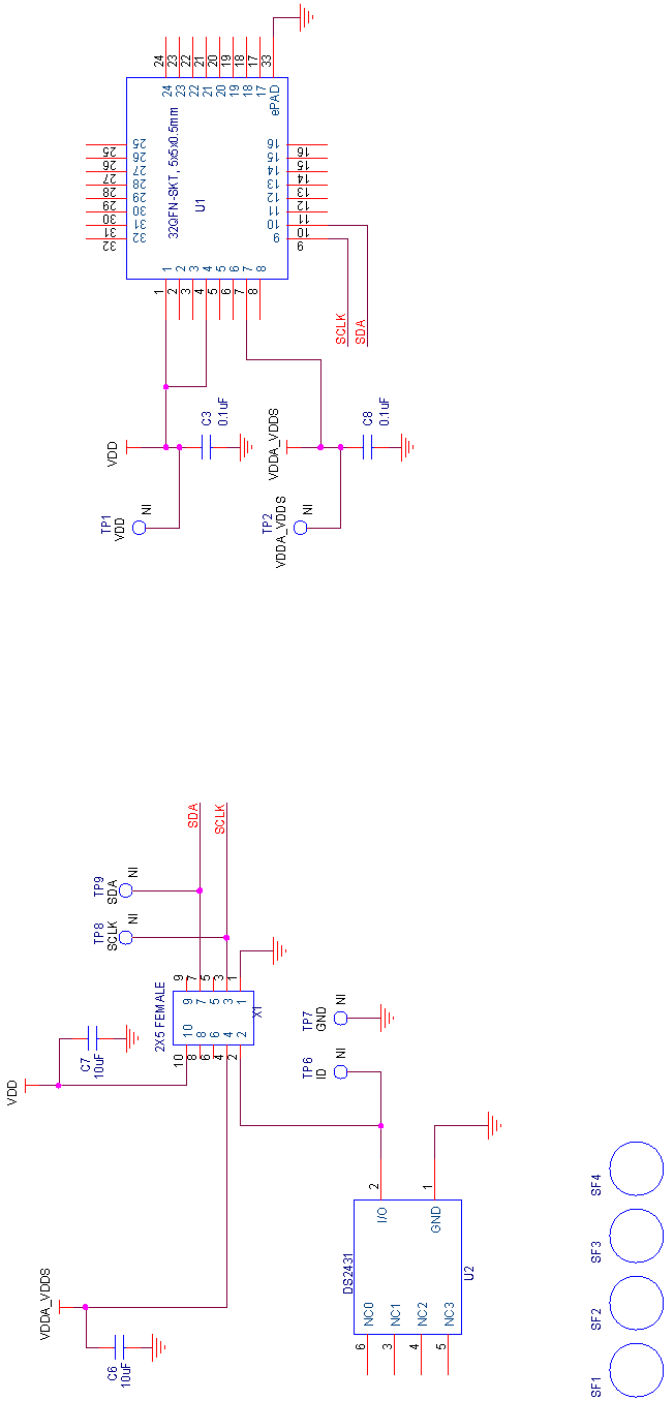


Figure 90. Si5332 32-Pin Socket Board Schematic

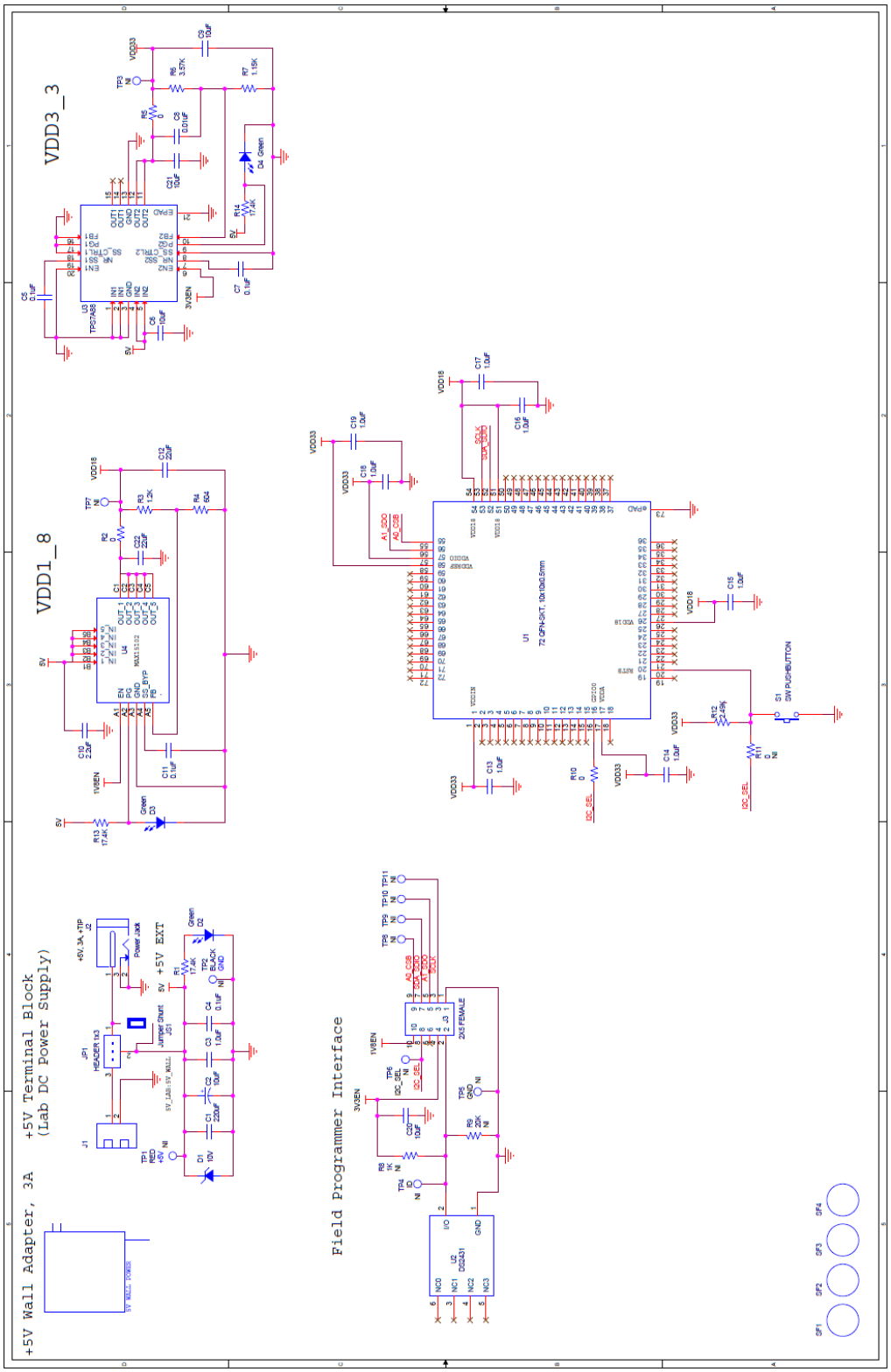


Figure 91. Si55xx 72 pin Socket Board Schematic

8. Appendix C. Bill of Materials

8.1. Field Programmer Bill of Materials

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
	9	C2 C3 C8 C11 C26 C28 C30 C31 C32	1uF		16V	±10%	X7R	C0603	C0603X7R160-105K	Venkel
	3	C24 C27 C29	0.01uF		16V	±20%	X7R	C0603	C0603X7R160-103M	Venkel
	1	C39	33pF		25V	±10%	C0G	C0402	C0402C0G250-330K	Venkel
	9	C4 C6 C12 C13 C14 C35 C36 C37 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	LTL-14CHJ	20mA				LED-T1-KK	LTL-14CHJ	LITE-ON TECHNOLOGY CORP
	1	D5	SP0503BAHT	300mW	20V		TVS	SOT143-AKKK|SOT143	SP0503BAHTG	Littlefuse
	7	D6 D7 D8 D9 D10 D11 D12	GL05T-E3-08	5A	11V		Dual Common Anode	SOT23-123	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	J3	ENCLOSURE					N/A	Emulator7045	Shanghai Zhongxingda Electronics
	4	R1 R11 R12 R65	1K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1001F	Venkel
	10	R13 R16 R26 R27 R40 R41 R42 R43 R59 R67	10K	1/16W		±1%	ThickFilm	R0402|R0402L	CR0402-16W-1002F	Venkel
	4	R14 R15 R24 R25	2.49K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-2491F	Venkel
	1	R28	1.0	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1R00F	Venkel
	8	R29 R30 R31 R32 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	R0402	CR0402-16W-4421F	Venkel
	1	R34	9.53K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-9531F	Venkel
	1	R36	5.9K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-5901F	Venkel
	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	R0402|R0402L	CR0402-16W-4990F	Venkel
	2	R63 R64	470	1/16W		±5%	ThickFilm	R0402	CR0402-16W-471J	Venkel
	2	R9 R10	10.0K	1/10W		±0.1%	±25PPM	R0603	ERA-3AEB103V	Panasonic
	2	SF1 SF2	BUMPER					RUBBER_FOOT_0.250"	SJ5382	3M
	2	SF3 SF4	SPACER					N/A	7363	Keystone Electronics
	1	U1	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	U12	TPS76201	100mA			LDO	SOT5N2.8P0.95	TPS76201DBV	TI
	2	U13 U14	TPS79501	500mA			LDO	DFN8N3.0P0.65E2.4X1.65	TPS79501DRBT	TI
	1	U16	ADG719					SOT6N2.8P0.95	ADG719BRTZ	Analog Devices
	1	U17	NLSV1T244		.9-4.5V		Buffer	UDFN6N1P0.4	NLSV1T244MUTBG	On Semi
	1	U2	C8051F380				MCU	QFP48N9X9P0.5	CF380P1104AGQ	SiLabs
	2	U3 U8	ADG712					TSSOP16N6.4P0.65	ADG712BRU	Analog Devices
	1	U5	LTC4311		5.5V			SC70-6N2.1P0.65	LTC4311CSC6#TRMPBF	Linear Technology
	1	U6	PCA9517D				I2C	SO8N6.0P1.27	PCA9517D	NXP
	1	X1	2X5 Header				Shrouded	CONN2X5-RA-SBH11	SBH11-PBPC-D05-RA-BK	Sullins Connector Solutions

Not Installed Components

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
NI	2	C10 C40	1uF		16V	±10%	X7R	C0603	C0603X7R160-105K	Venkel
NI	1	C7	0.1uF		10V	±10%	X7R	C0402|C0402L	C0402X7R100-104K	Venkel
NI	1	JP1	HEADER 4X1				Header	CONN-1X4	TSW-104-07-T-S	Samtec
NI	2	R60 R61	0	1A			ThickFilm	R0603	CR0603-16W-000	Venkel
NI	5	TP1 TP2 TP3 TP9 TP10	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	1	TP7	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI	1	TP8	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn
NI	1	U4	AD8628		5V		OPAMP	SOT23-5N	AD8628AUJ-R2	Analog Devices

8.2. Si538x4x-64SKT-DK Socket Board BOM

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	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 Components										
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
NI	1	R1	1K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1001F	Venkel
NI	1	R3	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

8.3. Si538x4x-56SKT-DK Socket Board Bill of Materials

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
	7	C1 C2 C5 C6 C7 C8 C9	0.1uF		10V	±10%	X7R	C0402|C0402L	C0402X7R100-104K	Venkel
	2	C3 C4	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	7	R1 R2 R7 R8 R9 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 Components										
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

8.4. Si538x4x-44SKT-DK Socket Board Bill of Materials

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	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 Components										
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
NI	1	R1	1K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1001F	Venkel
NI	1	R3	182	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1820F	Venkel
NI	3	TP1 TP2 TP3	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	3	TP4 TP5 TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn
NI	1	TP7	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn

8.5. Si5332-48SKT-DK Socket Board Bill of Materials

Eval Board Name	SI5332-48SKT
Revision	1.0

CreationDate	Proto Rev	Released
5/18/2017	1.00	0

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
	2	C3 C8	0.1uF		10V	±10%	X7R	C0402|C0402L	C0402X7R100-104K	Venkel
	2	C6 C7	10uF		10V	±20%	X7R	C1206	C1206X7R100-106M	Venkel
	4	SF1 SF2 SF3 SF4	BUMPER					RUBBER_FOOT_SMALL	SJ61A6	3M
	1	U1	48QFN-SKT, 6x6x0.4mm				QFN		790-62048-101G	Sensata
	1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Components										
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	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

8.6. Si5332-40SKT-DK Socket Board Bill of Materials

Eval Board Name	SI5332-40SKT
Revision	1.0

CreationDate	Proto Rev	Released
5/18/2017	1.00	0

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	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	40QFN-SKT, 6x6x0.5mm				QFN		790-42040-101G	Sensata
	1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	X1	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
Not Installed Components										
NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	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

8.7. Si5332-32SKT-DK Socket Board Bill of Materials

Eval Board Name	SI5332-32SKT
Revision	1.0

CreationDate	Proto Rev	Released
5/18/2017	1.00	0

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	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 Components

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	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

8.8. Si55xx-72SKT-DK Socket Board Bill of Materials

Eval Board Name	SI55XX-72SKT
Revision	1.0

CreationDate	Proto Rev	Released
8/20/2020	12.00	0

End product must be RoHS compliant

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
	1	C1	220uF		6.3V	±20%	X5R	C1210	C1210X5R6R3-227MNE	Venkel
	1	C10	2.2uF		10V	±10%	X7R	C0603	C0603X7R100-225K	Venkel
	2	C12 C22	22uF		6.3V	±20%	X5R	C0603	C1608X5R0J226M080AC	TDK
	1	C2	10uF		25V	±20%	TANT	C6032	T491C106M025ZT	Kemet
	8	C3 C13 C14 C15 C16 C17 C18 C19	1.0uF		6.3V	±10%	X5R	C0402|C0402L	C0402X5R6R3-105K	Venkel
	4	C4 C5 C7 C11	0.1uF		16V	±10%	X7R	C0402	C0402X7R160-104K	Venkel
	4	C6 C9 C20 C21	10uF		6.3V	±20%	X5R	C0603|C0603L	C0603X5R6R3-106M	Venkel
	1	C8	0.01uF		25V	±10%	X7R	C0402	C0402X7R250-103K	Venkel
	1	D1	10V	500mW	10V	5%	Zener	SOD-123	MMSZ4697T1G	On Semi
	3	D2 D3 D4	Green	20mA	3.4V		SMT, ChipLED	LED-HSMX-C170	HSMQ-C170	Avago Technologies
	1	J1	CONN TRBLK 2	24A			TERM BLK	CONN-TB-1711026	1711026	PHOENIX CONTACT
	1	J2	Power Jack	2.5A			BARREL	CONN-3-PWR|PJ-002A	PJ-002A	CUI
	1	J3	2X5 FEMALE				CONN	CONN2X5-FRA-SFH11	SFH11-PBPC-D05-RA-BK	Sullins Connector Solutions
	1	JP1	HEADER 1x3				Header	CONN-1X3	TSW-103-07-T-S	Samtec
	1	JS1	Jumper Shunt				Shunt	N/A|SHUNT	SNT-100-BK-T	Samtec
	3	R1 R13 R14	17.4K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1742F	Venkel
	1	R10	0	1A			ThickFilm	R0603|R0603L	CR0603-16W-000	Venkel
	1	R12	2.49K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-2491F	Venkel
	2	R2 R5	0	1W			ThickFilm	R2512	RMCF2512ZT0R00	Stackpole Electronics Inc.
	1	R3	1.2K	1/16W		±5%	ThickFilm	R0402	CR0402-16W-122J	Venkel
	1	R4	604	1/16W		±1%	ThickFilm	R0402	CR0402-16W-6040F	Venkel
	1	R6	3.57K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-3571FT	Venkel
	1	R7	1.15K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-1151FT	Venkel
	1	S1	SW PUSHBUTTON	50mA	12Vdc		Tactile	SW4N10P4.5	2-1437565-8	Tyco Electronics
	4	SF1 SF2 SF3 SF4	BUMPER				FOOT	RUBBER_FOOT_SMALL	SJ61A6	3M
	1	U1	72 QFN-SKT, 10x10x0.5mm				QFN	QFN72N10X10P0.5-SKT-WELLS-CTI	790-42072-101G	Sensata
	1	U2	DS2431					SOJ6N4.45P1.27	DS2431P+	Maxim
	1	U3	TPS7A88	1A			LDO	QFN20N4X4P0.5E2.7	TPS7A88	TI
	1	U30	WSU050-3000	15W	5V		MISC		WSU050-3000	TRIAD
	1	U4	MAX15102EWL+	2A			LDO	WLBGA15N1.6X2.7P0.5	MAX15102EWL+	Maxim

Not Installed Components

NI	Quantity	Reference	Value	Rating	Voltage	Tolerance	Type	PCB_Footprint	ManufacturerPN	Manufacturer
NI	1	R11	0	1A			ThickFilm	R0402|R0402L	CR0402-16W-000	Venkel
NI	1	R8	1K	1/16W		±1%	ThickFilm	R0402|R0402L	CR0402-16W-1001F	Venkel
NI	1	R9	20K	1/16W		±1%	ThickFilm	R0402	CR0402-16W-2002F	Venkel
NI	7	TP1 TP3 TP7 TP8 TP9 TP10 TP11	RED				Loop	TESTPOINT	151-207-RC	Kobiconn
NI	2	TP2 TP5	BLACK				Loop	TESTPOINT	151-203-RC	Kobiconn
NI	2	TP4 TP6	BLUE				Loop	TESTPOINT	151-205-RC	Kobiconn