Texas Instruments

Application Report
SLUA888–May 2018

NVM Programming for VR13 Power Controllers

ABSTRACT

TPS53679, TPS53659, TPS53622 are Intel™ VR13 Serial VID (SVID)-compliant power supply controllers that have programmable parameters. This guide also applies to proprietary footprint devices TPS53678, TPS53658 TPS53655, as well as TPS53681, even though it is not an Intel power controller. The PMBus interface configures the parameter values which are stored into non-volatile memory (NVM) as new boot-up default values. This guide gives a tutorial on NVM programming, and the tools TI provides for the programming process. This guide applies to all the devices in the TI VR13 controller family.

Contents

List of Figures

List of Tables

Trademarks

Intel is a trademark of Intel Corporation.

1 Introduction

This guide gives an overview of NVM programming for the following TI VR13 controller devices: TPS53679, TPS53659, TPS53622, TPS53678, TPS53658, TPS53655 and TPS53681. Local TI sales and field applications representatives can provide relevant documentation to these devices. This guide gives a technical overview and step-by-step instructions.

1.1 Software Tools

1.1.1 Fusion Digital Power Designer

Fusion Digital Power Designer is a graphical user interface (GUI) that configures and monitors Texas Instruments digital power controllers, sequencer monitors, and health monitors. The GUI uses the PMBus protocol to communicate with the device over a serial bus using a proprietary USB adapter.

Use the *Fusion Digital Power Designer* GUI for engineering development. The GUI gives access to all available user-configurable settings, fault information, and telemetry readings. After you configure the device, you can export the settings to a configuration file. The export options are:

- *System* file (.tifsp) contains configuration data for all of the devices in a system
- *Project* file (.xml) contains configuration data for a single device
- PMBus programmer script (.csv) is a text file containing step-by-step programming instructions for non-TI software tools

Download the latest version of *Fusion Digital Power Designer* from this URL: http://www.ti.com/tool/fusion_digital_power_designer

1.1.2 Fusion Manufacturing Tool

The *Fusion Manufacturing Tool* is a graphical user interface (GUI) that programs TI digital power controllers in a production environment. Download the latest copy of the *Fusion Manufacturing Tool* from this URL: http://www.ti.com/tool/fusion_mfr_gui

1.1.3 Online and Offline Modes

The *Fusion Digital Power Designer* software operates in online mode when you have connected a device to the system that hosts the software. The software operates in offline mode when you have not connected a device to the system that hosts the software. The software installer provides a different shortcut for each mode.

1.1.4 Relevant File Formats

Table 1. Configuration File Formats

File	Description	Extension	Type	Compatilble Software
Fusion project file	Contains configuration data for a **single** device. Intended for engineering development.	.xml	plain-text XML	*Fusion Digital Power Designer*
Fusion system file	Contains configuration data for multiple devices. Can be used in engineering development, and production programming.	.tifsp	plain-text XML	*Fusion Digital Power Designer*, *Fusion Manufacturing Tool*
PMBus programmer script	Simplified programming script (comma separated format) for production environment.	.csv	plain-text comma separated value	*Fusion Digital Power Designer* and third-party tools

1.2 Hardware Tools

1.2.1 USB-to-GPIO Dongle

Both *Fusion Digital Power Designer* and *Fusion Manufacturing Tool* GUIs use the TI USB-GPIO adapter to interface between a host computer and controller devices. Use this link to get a TI USB-GPIO adapter: http://www.ti.com/tool/usb-to-gpio.

2 Technical Overview

This section shows how to program NVM with the TI software tools, or with non-TI tools.

2.1 Hardware Connections

You can program NVM on a fully-populated application board. However, successful programming requires only a few connections be made to these devices. In some cases offline gang-programming in socketed boards works better than a full In Circuit Test (ICT) solution.

Table 2. Hardware Connections for Offline NVM Programming

Pin Name	Connection
V3P3	Connect to +3.3V supply, and bypass with a minimum of 1.0 µF to ground.
VREF	Bypass to ground with a minimum of 1.0 µF to ground.
ADDR	Connect a resistor divider from VREF to ADDR to Ground to set the PMBus address (required for programming), as described in the product datasheet.
SMB_DIO, SMB_CLK	Connect to programming host, pull-up to 3.3V with a 2.2 kΩ resistor. Note, the TI USB-to-GPIO interface adapter has internal pull-up resistors.
ATSEN, BTSEN, TSEN	Not used for NVM programming. Tie-off with a 1:1 resistor divider from VREF to TSEN to Ground, suggested resistor value is 49.9 kΩ. This sets the pin voltage to appx. 0.85 V, which the controller device interprets as 32°C.
AVR_EN, BVR_EN, VIN_CSNIN, CSPIN, AVSP, BVSP, AVSN, BVSN	Not used for NVM programming. Tie-off to ground.
ACSPx, BCSPx	Not used for NVM programming. Tie off to VREF.
APWMx, BPWMx, ASKIP, BSKIP, VRFAULT, SMBALERT, SALERT, PIN_ALT, SDIO, SCLK	Not used for NVM programming. Float these pins if not used.

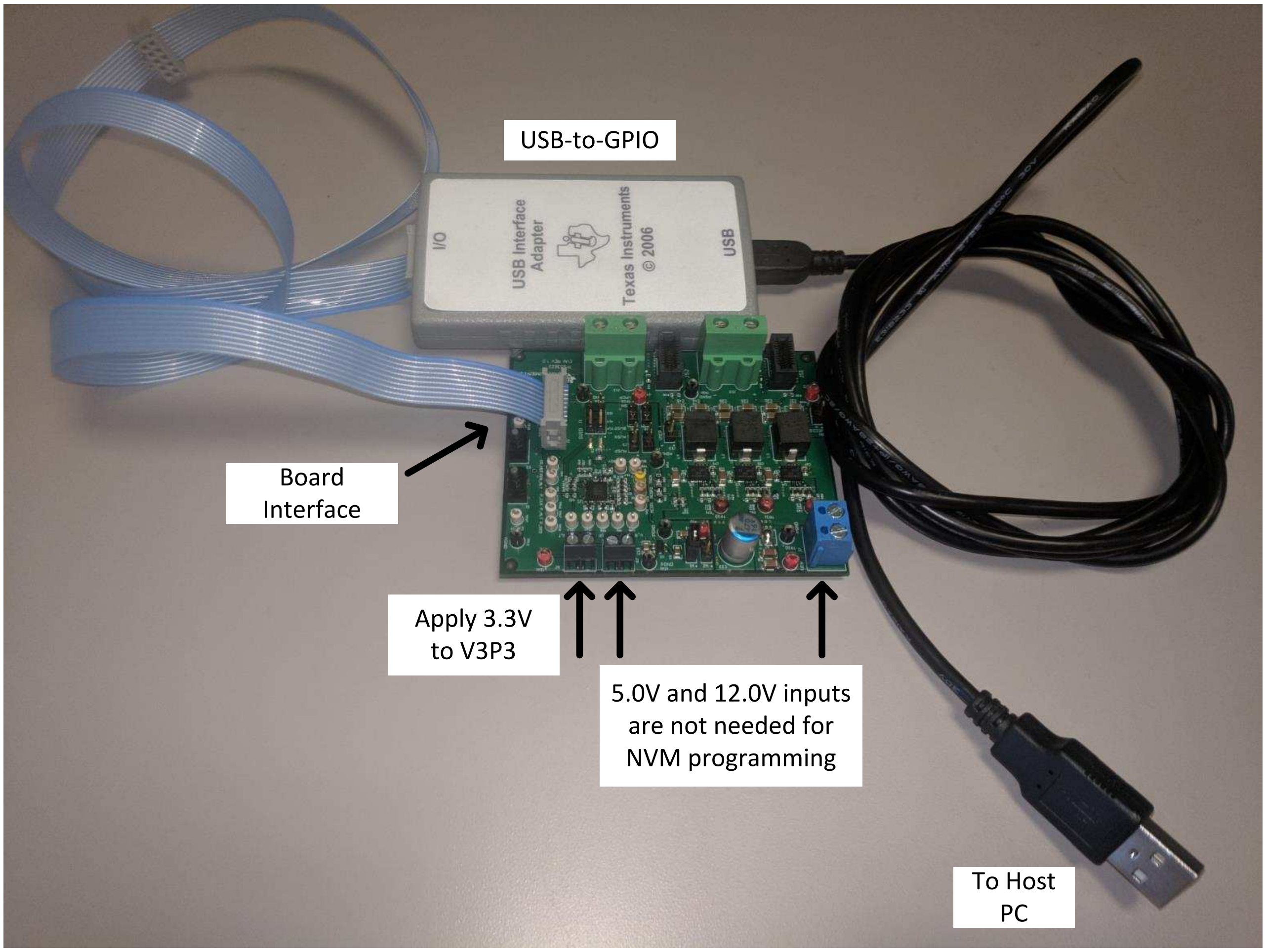

Figure 1. Hardware Connections

2.2 Communication Protocol and Interfacing

These devices comply with the PMBus version 1.3 specification. Use this link to find timing and electrical characteristics of the PMBus specification in the PMB Power Management Protocol Specification, Part 1, revision 1.3 available at http://pmbus.org.. The PMBus specification inherits its transport and network layer behavior from the SMBus specification. These devices comply with the SMBus 3.0 specification, which is available at this URL: http://smbus.org/specs/. Use the transaction prototypes to program the device NVM correctly. Refer to the Section 2.3 section for more information about prototype use in NVM programming.

The use of Packet Error Correction (PEC) is optional. If clock pulses are supplied for a PEC byte, these devices use PEC, otherwise they do not. For simplicity, Table 3 lists required transaction types without PEC bytes.

Table 3. SMBus Transaction Types

Transaction Type	Purpose
Send Byte	Used to issue NVM Store operations (STORE_DEFAULT_ALL).
Write Byte	Used to change the current PAGE
Write Word	Used to Write the value of VOUT_MAX.
Read Word	Used to Read the current value of VOUT_MAX.
Write Block	Used to Write the USER_DATA commands, which contain most of the NVM storable parameters these devcies support.
Read Block	Used to read the USER_DATA commands and checksum (MFR_SERIAL).

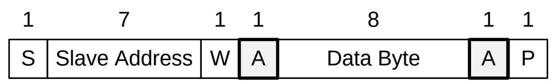

Figure 2. Send Byte Protocol

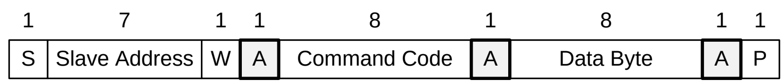

Figure 3. Write Byte Protocol

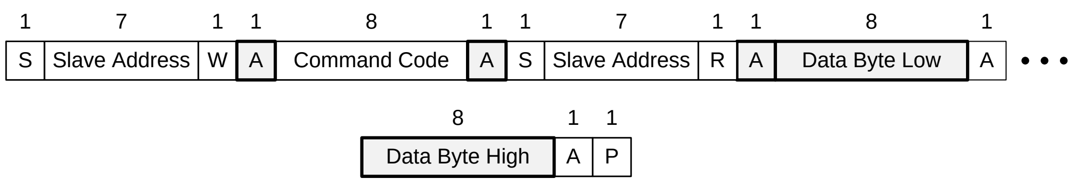

Figure 4. Write Word Protocol

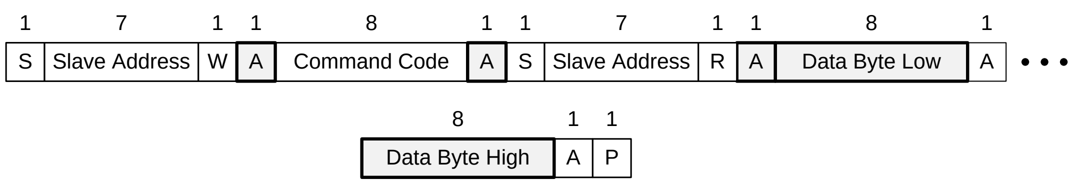

Figure 5. Read Word Protocol

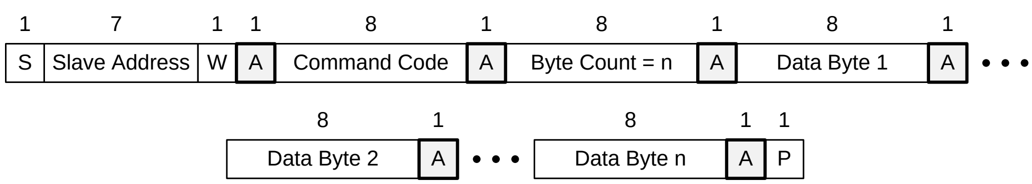

Figure 6. Block Write Protocol

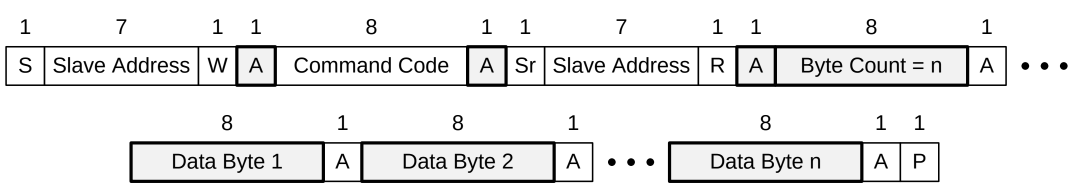

Figure 7. Block Read Protocol

2.3 Programming Process

To simplify the programming procedure and reduce programming time, TI VR13 controller devices combine NVM settings into a small number of registers. All settings in MFR_SPECIFIC commands map into registers USER_DATA_00 through USER_DATA_12. Only a few other registers are required. Program the USER_DATA command with the Block Read/Write command protocol described in the SMBus Specification. Table 4 lists the complete NVM configuration for a single device. All other settings map to USER_DATA commands.

The *Fusion Digital Power Designer* GUI lets power supply designers configure devices in a graphical environment. The GUI saves these settings in a configuration file or script that you can load on to other devices.

2.3.1 Programming Procedure

This section describes NVM programming through third-party tools. This procedure is completed automatically during import and export of *Project* files from *Fusion Digital Power Designer*.

Configure User-Programmable Parameters (one-time only)

1. Set all of the user-accessible parameters via the standard PMBus, and Manufacturer Specific commands, using *Fusion Digital Power Designer* or the *Technical Reference Manual* for the device.
2. Issue the STORE_DEFAULT_ALL command. This command commits these values to NVM, and updates the checksum value.
3. Wait approximately 100 ms.
4. Write PAGE to 00h.
5. Read-back and Record the value of IC_DEVICE_ID and IC_DEVICE_REV commands
6. Read-back and Record the value of the USER_DATA_00 through USER_DATA_12 commands
7. Read-back and Record the value of the MFR_SERIAL command
8. Read-back and Record the value of VOUT_MAX
9. Write PAGE to 01h
10. Read-back and Record the value of VOUT_MAX

Program and Verify NVM (repeat for each device)

1. Apply 3.3 V to the V3P3 pin to start the device. Make sure to disable power conversion for NVM programming.
2. Read-back and verify that IC_DEVICE_ID and IC_DEVICE_REV values match those recorded previously. This verification ensures that user-parameters being programmed correspond to the same device/revision as previously configured.
3. Write PAGE to 00h.
4. Write the USER_DATA_00 through USER_DATA_12 commands, with the values recorded previously.
5. Write VOUT_MAX (Page 0) with the value recorded previously.
6. Write PAGE to 01h
7. Write VOUT_MAX (Page 1) with the value recorded previously.
8. Issue STORE_DEFAULT_ALL.
9. Wait approximately 100 ms.
10. Read-back the MFR_SERIAL command, and compare the value to that recorded previously. If the new MFR_SERIAL matches the value recorded previously, the software has successfully programmed the NVM.

2.3.2 Example NVM Data

Table 4 gives an example configuration that contains **all programmable parameters** in a VR13 controller device. Every application has different data. All other parameters stored in configuration files are either calculated, measured, or derived from these values. The *Fusion Digital Power Designer* GUI displays block commands in ascending order of significance (for example byte 0, byte 1, ...). The GUI displays word commands in descending significance (for example byte 1, byte 0).

Table 4. Example NVM Data

PMBus Command	Transaction Type	CMD Code (hex)	Example Value (hex)
USER_DATA_00	Block Write, Block Read	B0h	1410020030F5
USER_DATA_01	Block Write, Block Read	B1h	000000000040
USER_DATA_02	Block Write, Block Read	B2h	8904000000D0
USER_DATA_03	Block Write, Block Read	B3h	030010050080
USER_DATA_04	Block Write, Block Read	B4h	0906C325C777

Table 4. Example NVM Data (continued)

PMBus Command	Transaction Type	CMD Code (hex)	Example Value (hex)
USER_DATA_05	Block Write, Block Read	B5h	E408C57983E1
USER_DATA_06	Block Write, Block Read	B6h	85DA1112E87F
USER_DATA_07	Block Write, Block Read	B7h	804B051053FA
USER_DATA_08	Block Write, Block Read	B8h	0000000000A6
USER_DATA_09	Block Write, Block Read	B9h	000170848080
USER_DATA_10	Block Write, Block Read	BAh	00262EC0C185
USER_DATA_11	Block Write, Block Read	BBh	080220C08FE1
USER_DATA_12	Block Write, Block Read	BCh	40F08D20FF01
VOUT_MAX[PAGE 0]	Write Word, Read Word	24h	00FF
VOUT_MAX[PAGE 1]	Write Word, Read Word	24h	00FF
MFR_SERIAL	Block Write, Block Read	9Eh	05C60AD2

3 Use of TI Programming Tools

This section gives short tutorials for common tasks using the *Fusion Digital Power Designer* software.

3.1 *Fusion Digital Power Designer*

WARNING

Create *Fusion Digital Power Designer System* **files and** *Project* **files in online mode only. Do not update settings in offline mode. Files that you edit offline will not have the correct USER_DATA and checksum values stored in the** *System* **file, and can lead to unpredictable results.**

3.1.1 Build and Export a System (.tifsp) File

The *Fusion Digital Power Designer* software has a convenient graphical interface to configure and test TI VR13 controllers.

An more detailed discussion of each parameter in the GUI can be found in the application report *Using the Fusion Digital Power Designer for TPS536xx VR13 Multiphase Solutions*. Contact vr@list.ti.com for more information.

1. Connect the USB-GPIO adapter to either a system board, or socketed programming board with a TI VR13 controller on-board.
2. Apply 3.3 V to the V3P3 pins of the controller devices. It is not necessary to apply 5-V and 12-V power. Make sure to disable power conversion. You may need to pull the AVR_EN/BEN pins low to stop power conversion.
3. Start the *Fusion Digital Power Designer* software.
4. Click **Change Scanning Modes**, then select **DEVICE_ID & DEVICE_CODE & IC_DEVICE_ID** if the software does not recognize your device.

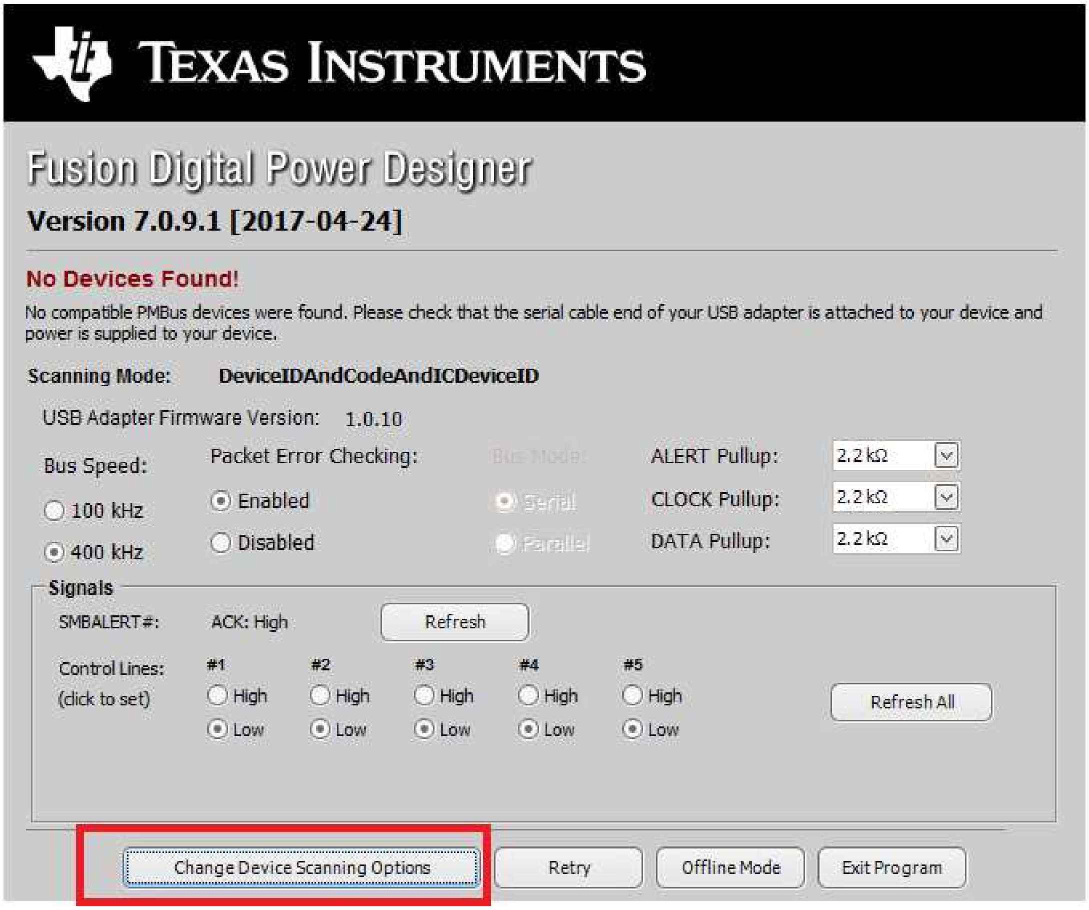

Figure 8. Change Scanning Modes

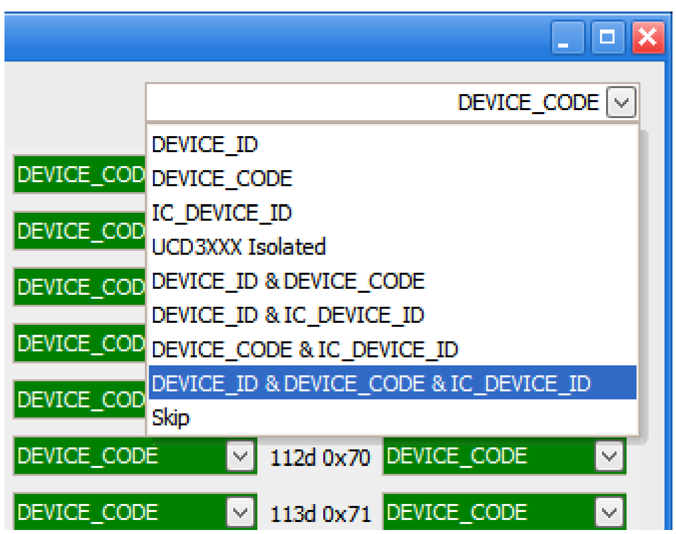

Figure 9. Scanning Mode Selections

5. From the System View, click **Click to Configure Device** to view individual controller devices.

   The application report, *Using the Fusion Digital Power Designer for TPS536xx VR13 Multiphase Solutions* contains detailed information about NVM configuration for TI VR13 multiphase controllers.

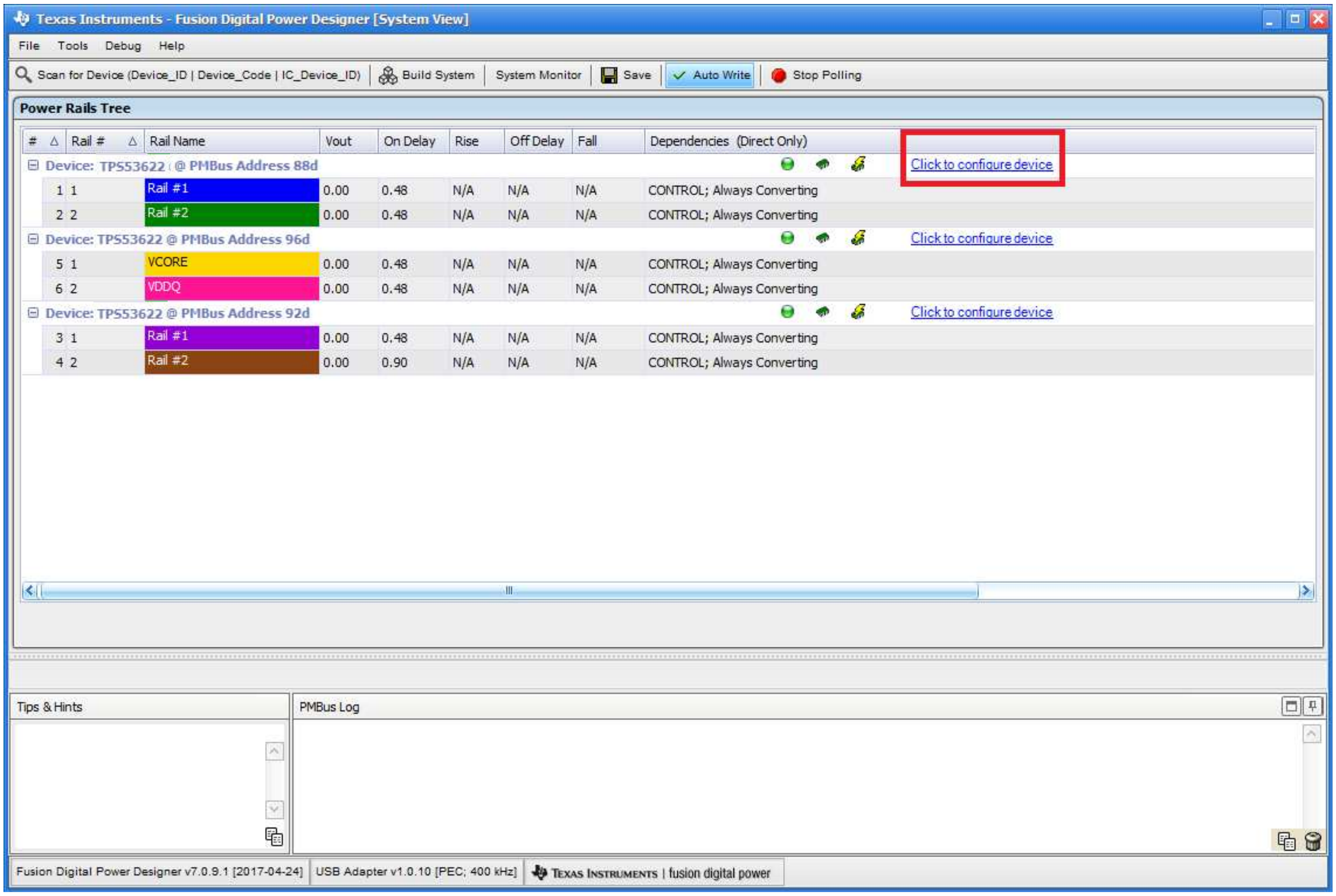

Figure 10. System View

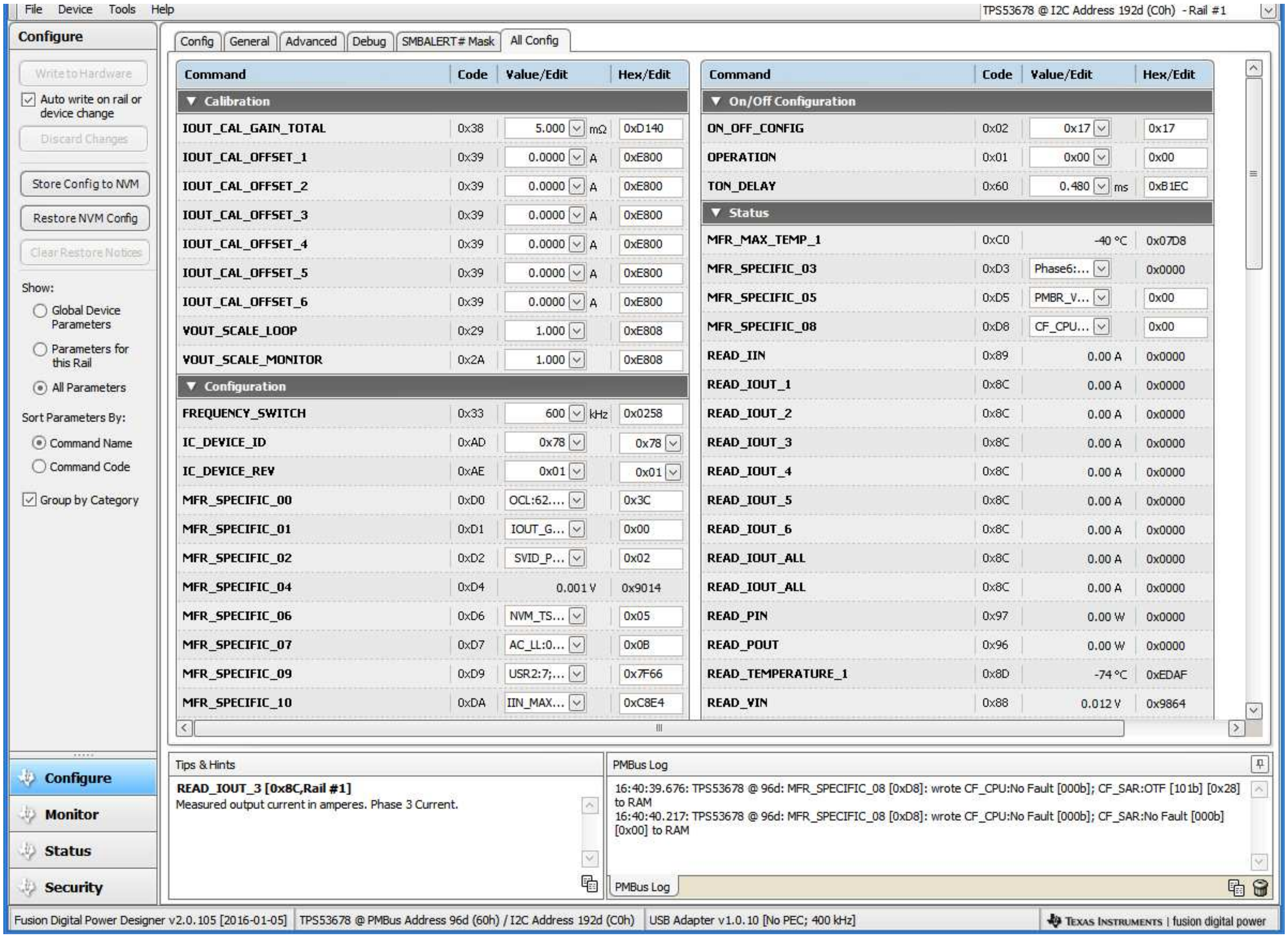

Figure 11. Settings for Individual Device

6. Click on **Write to Hardware** to set write the values to the device

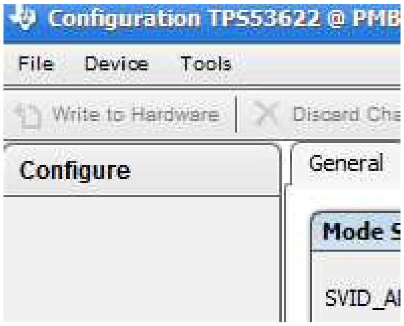

Figure 12. Write to Hardware

7. Click **Store Config. to NVM** to store the NVM settings.
8. Click **Device Menu**
9. Click **Refresh All Parameters** to update USER_DATA commands in the GUI.

10. Select **Save Project As** from the **File** menu to export a *Project* file for a single device.

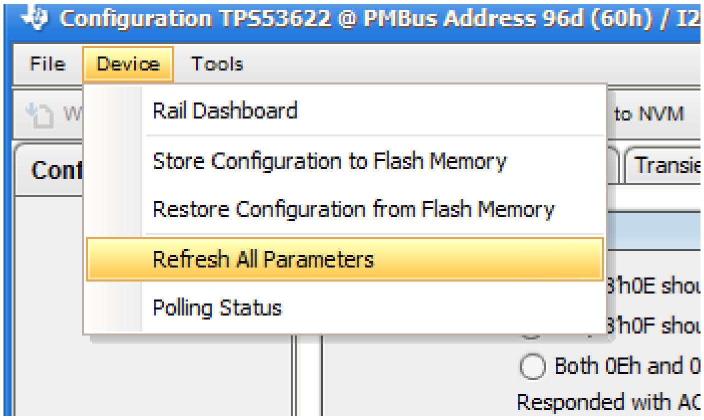

Figure 13. Refresh All Parameters

11. Repeat steps 5 through 14 for each device.
12. To set NVM values for another device, return to the **System View** page and **Click to Configure Device**.

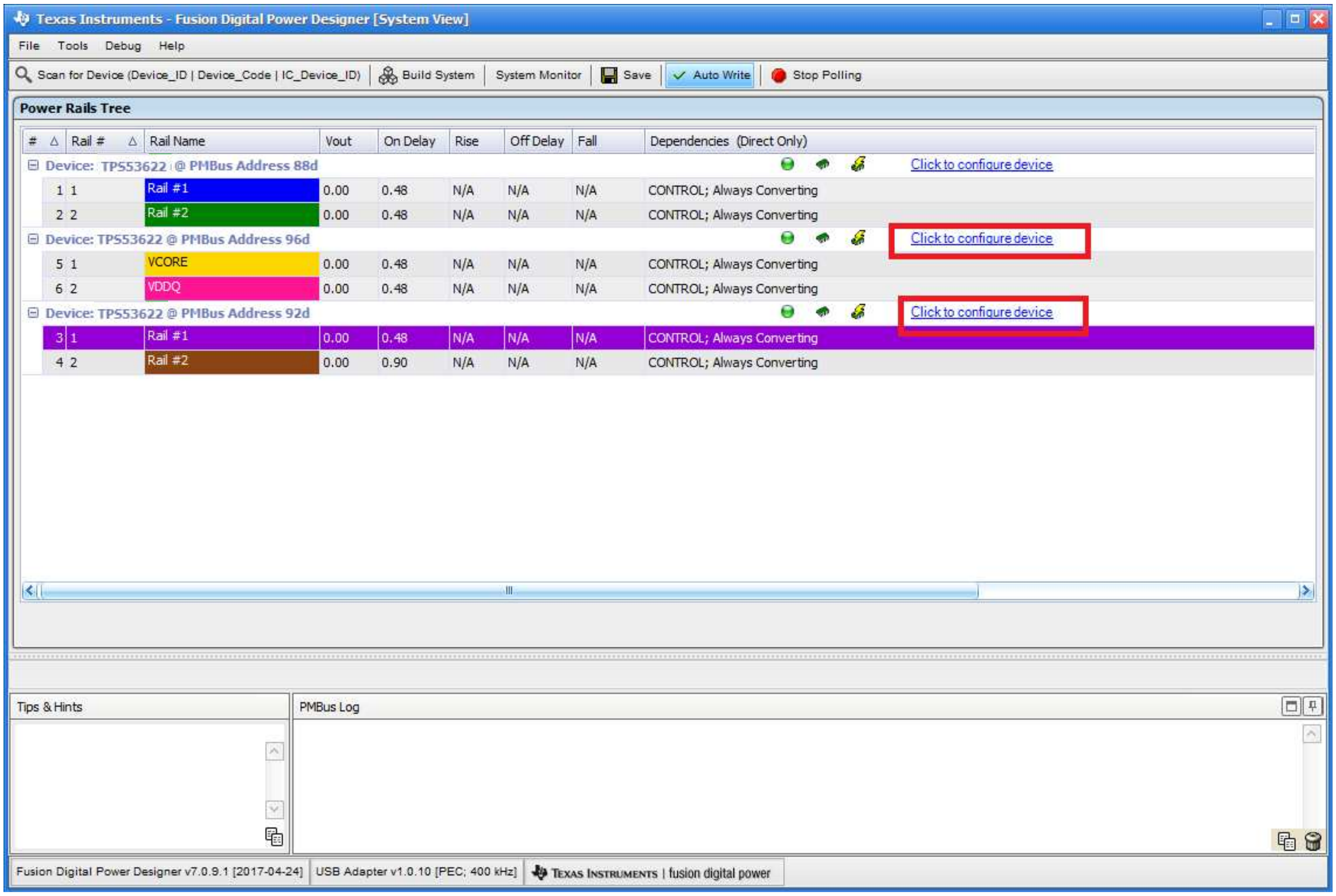

Figure 14. Switch between Multiple Devices

13. To save a *System* file after all devices are configured, return to the **System View** click the **File** menu and select, **Save System File As**.

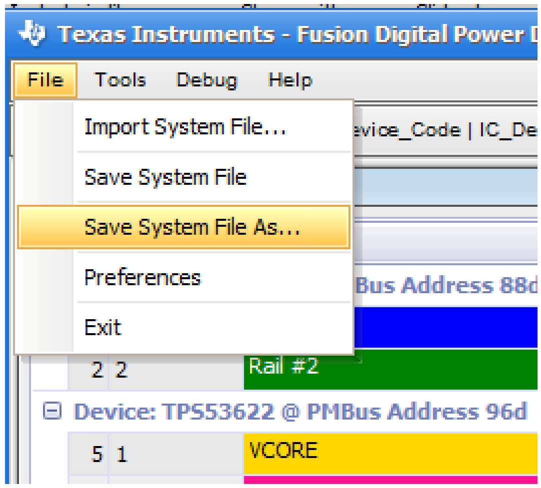

Figure 15. Save a *System* File

3.1.2 View a *Project* File or *System* File Offline

When the software operates in offline mode, you cannot update any NVM settings. Use offline mode only to view *System* files.

> **WARNING**
>
> ***Fusion Digital Power Designer*** **offline mode is for viewing of files only. To edit NVM settings, use online mode, which requires a controller device to be connected.** ***System*** **files that users edit offline may not contain the correct USER_DATA and checksum values stored in the** ***System*** **file, and can lead to unpredictable results.**

1. Open *Fusion Digital Power Designer* in offline mode.
2. Select **Create offline system by opening existing system file**.
3. Click **Next**

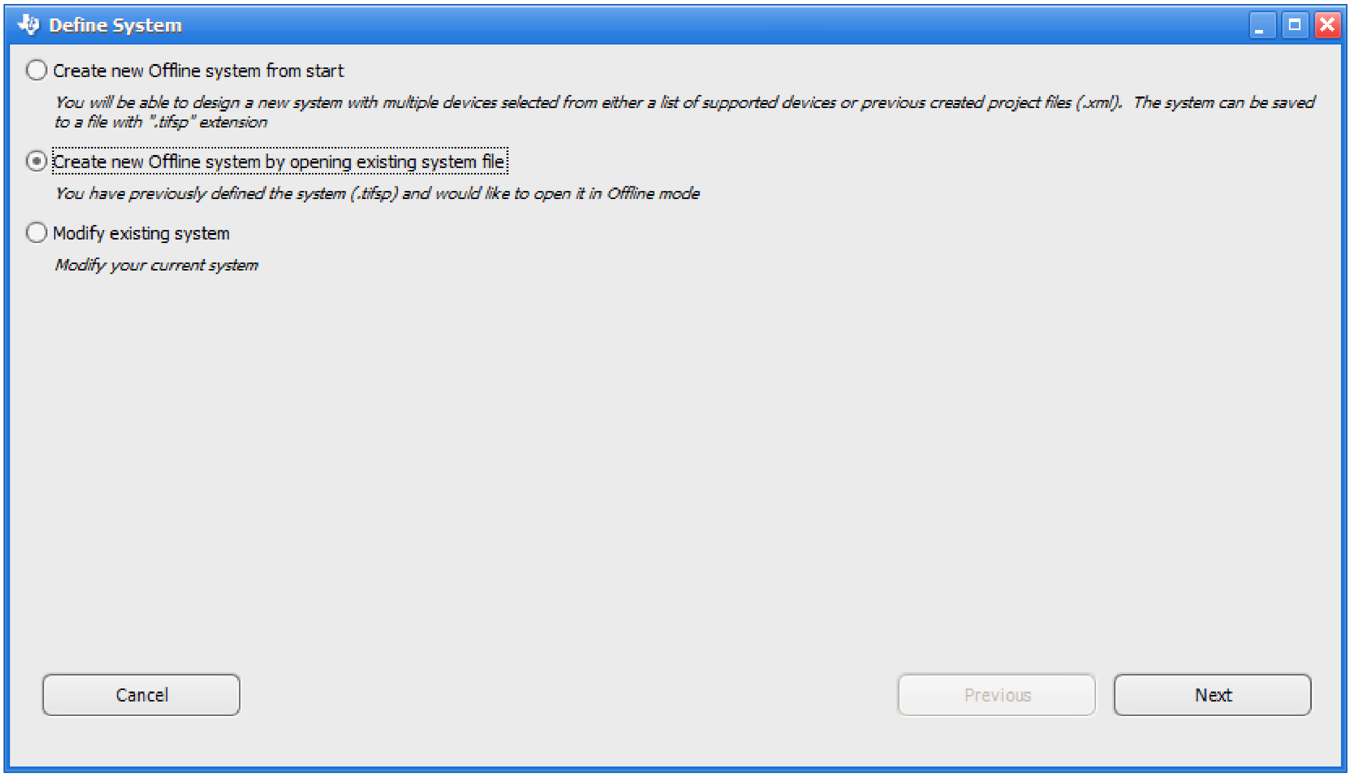

Figure 16. Select *System* File

4. Browse to the file when prompted, and click **Next**.

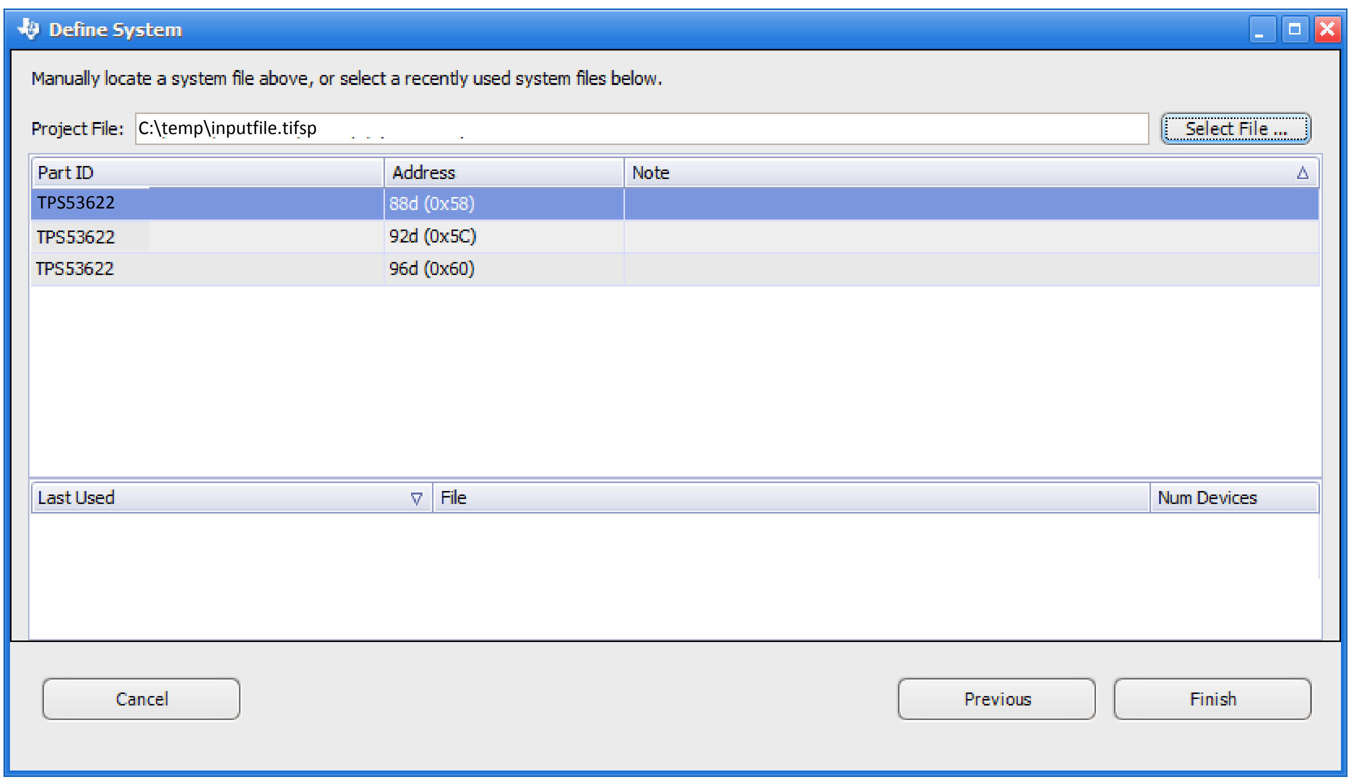

Figure 17. Select Offline *System* File

5. Click **Finish** to open the *System* file to review.

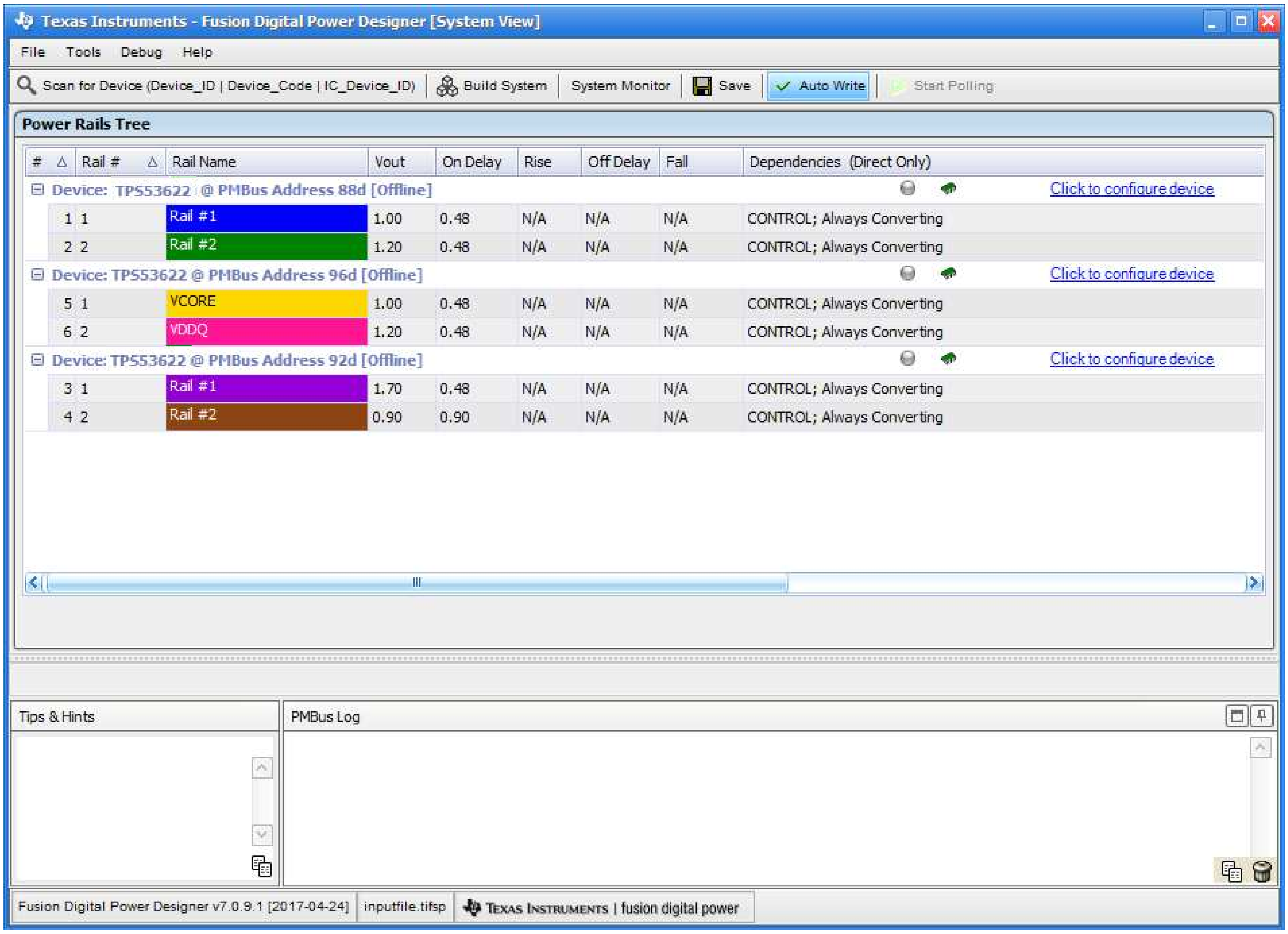

Figure 18. Offline System View

3.1.3 Import a *Project* File to a Single Online Device

1. Connect the USB-GPIO to a board.
2. Apply 3.3 V to the V3P3 pin of the controller device.
3. Start the *Fusion Digital Power Designer* software.
4. From the **System View** window, **Click to configure device**
5. From the **File** menu, select **Import to device**
6. Select **Project File**
7. Click **Next**.

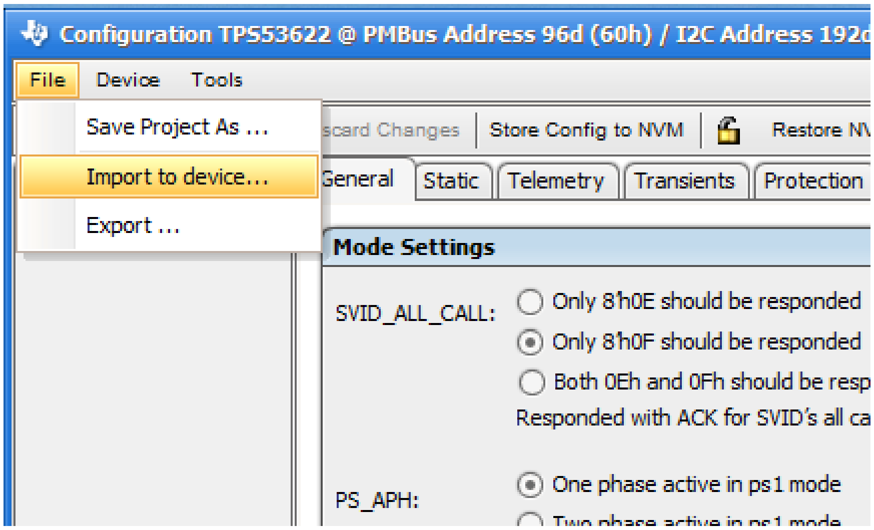

Figure 19. Import to Devcie

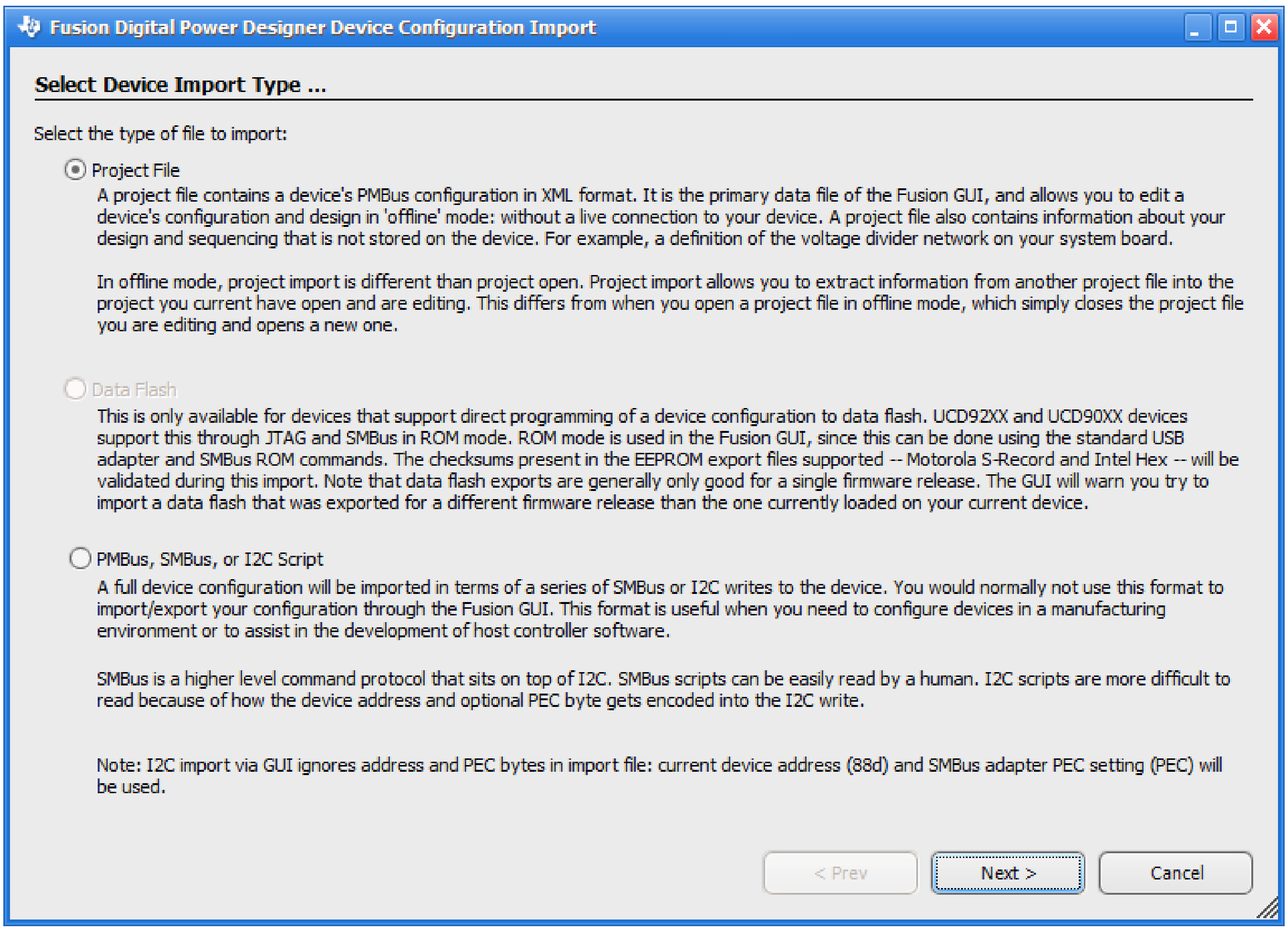

Figure 20. Import *Project* File

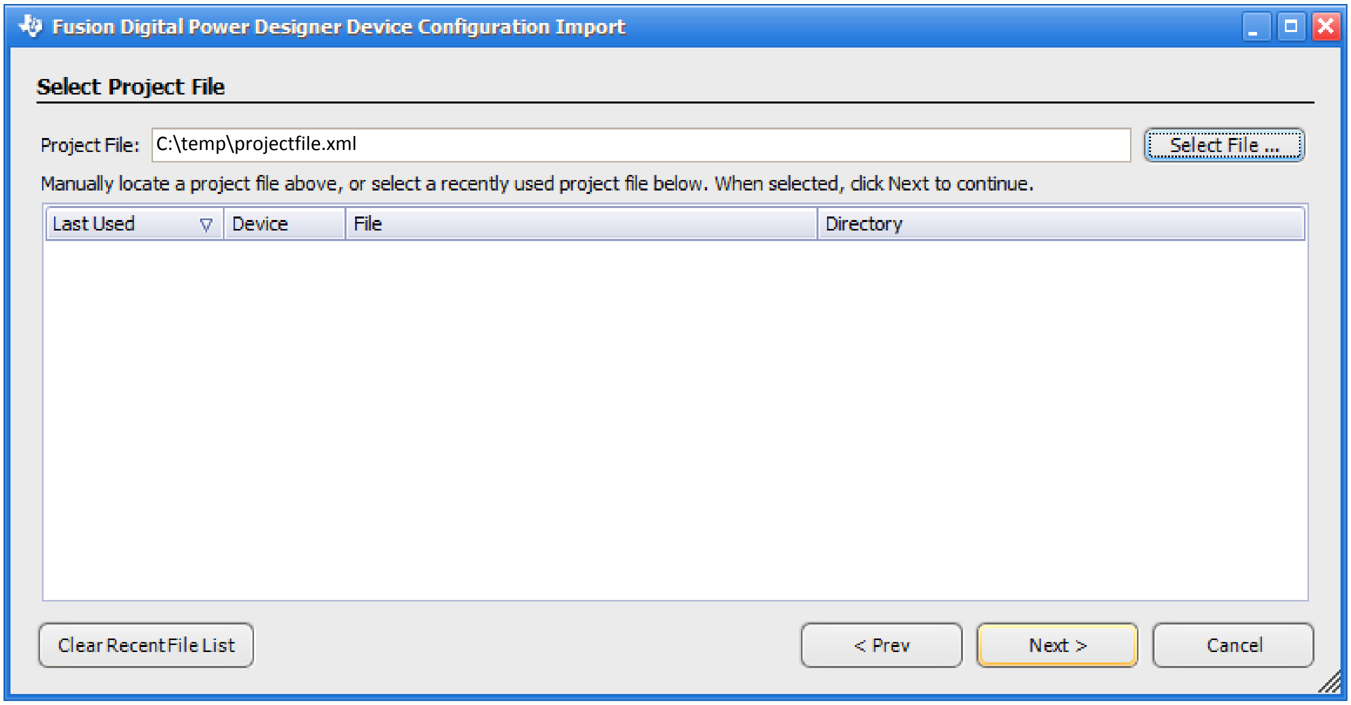

Figure 21. Select *Project* File

8. Click **Select All** to import all parameters of the device.
9. Click **Write Checked**.

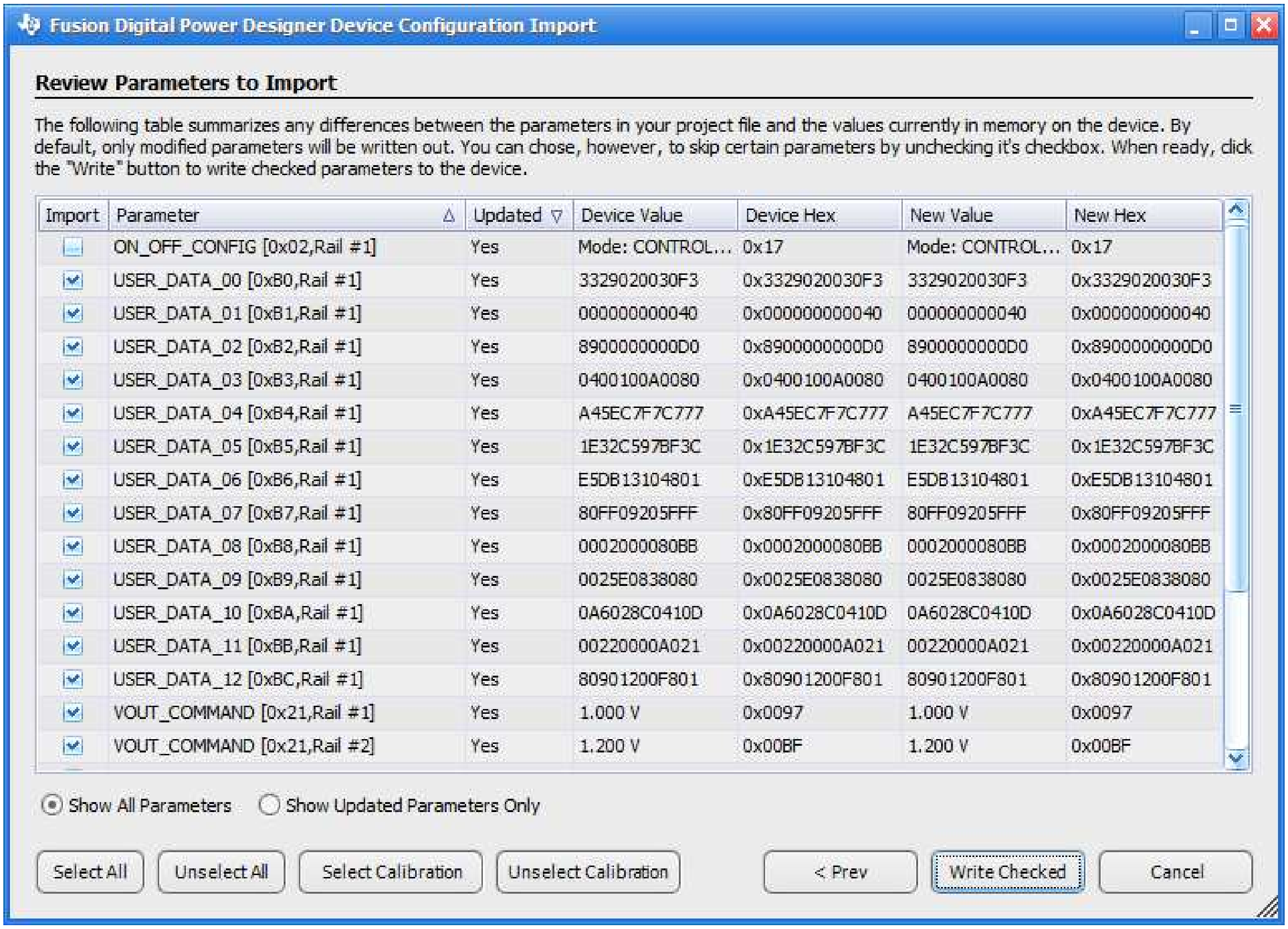

Figure 22. Select All and Write Checked Parameters

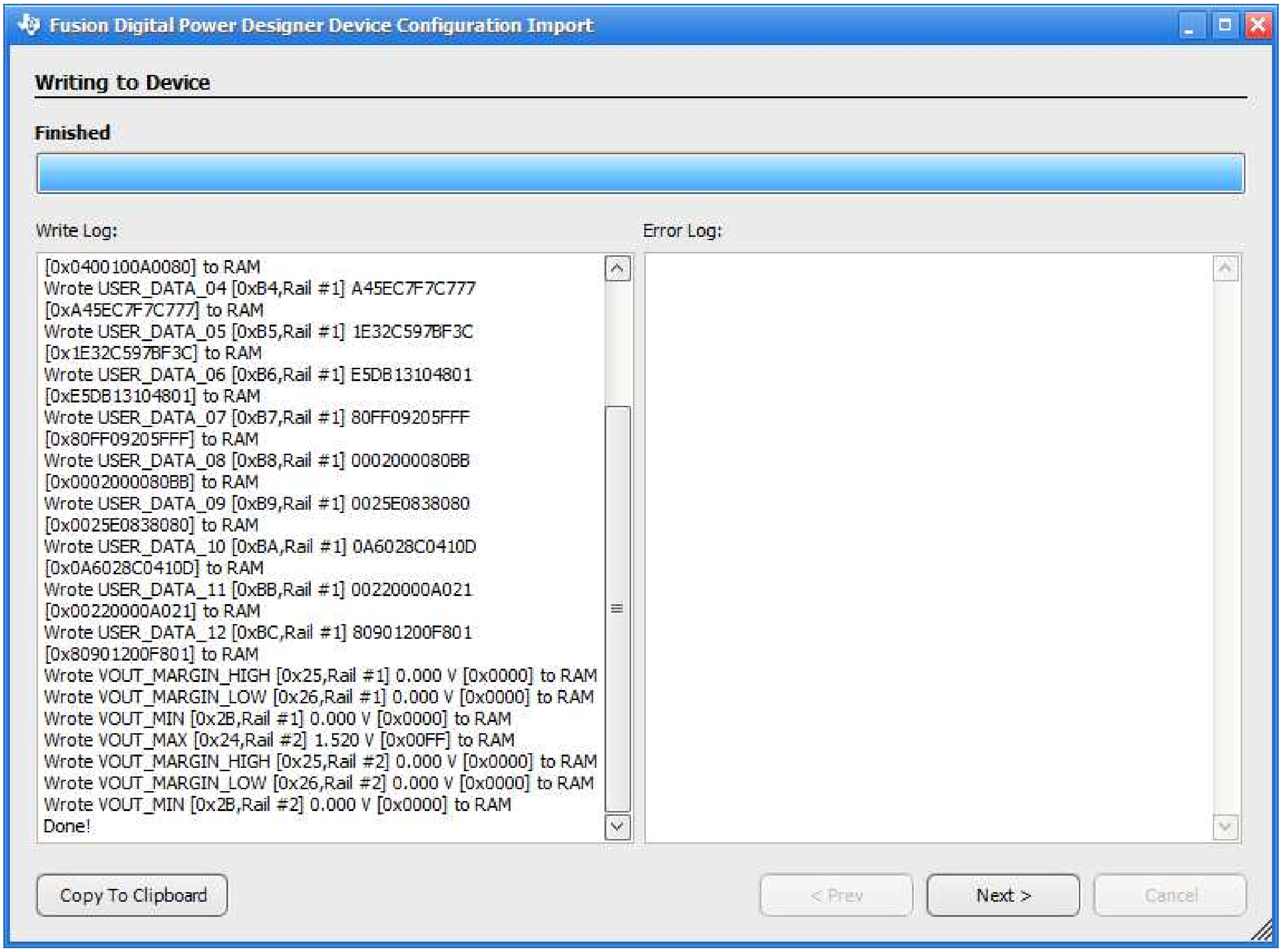

Figure 23. Successful *Project* File Import

10. When the *Project* file import finishes, the software **does not automatically issue an NVM store operation** command. You must click **Store Config to NVM** to update NVM settings after you import a *Project* file.

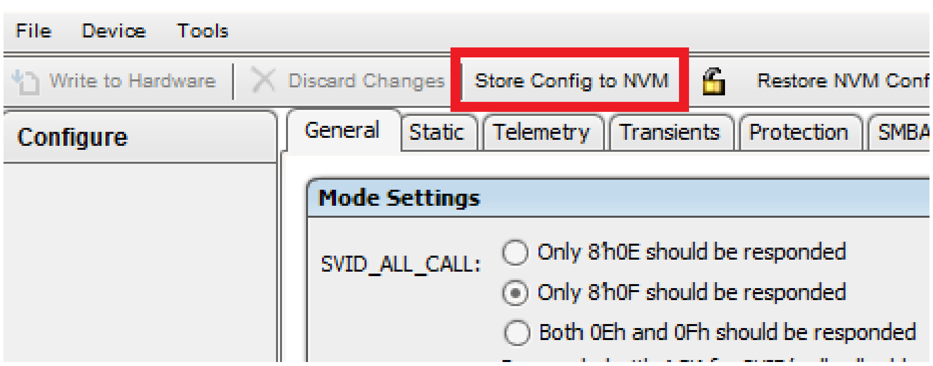

Figure 24. Store to NVM

3.1.4 Import a *System* File to an Online System

1. Connect the USB-GPIO interface to the board
2. Apply 3.3 V to the V3P3 pins of the controller devices.

3. Start the *Fusion Digital Power Designer* software.
4. From **File** menu on the **System View** page, select **Import System File...**

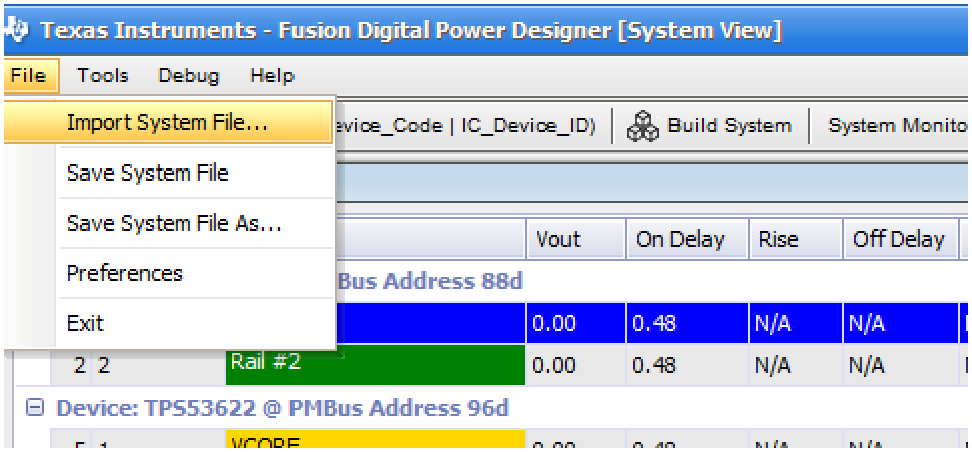

Figure 25. Import *System* **File**

5. The default setting is that all of the devices are selected but none of the parameters are selected. Check the box in the **Import** column for each device.
6. Click **Check All** to review all stored parameters.

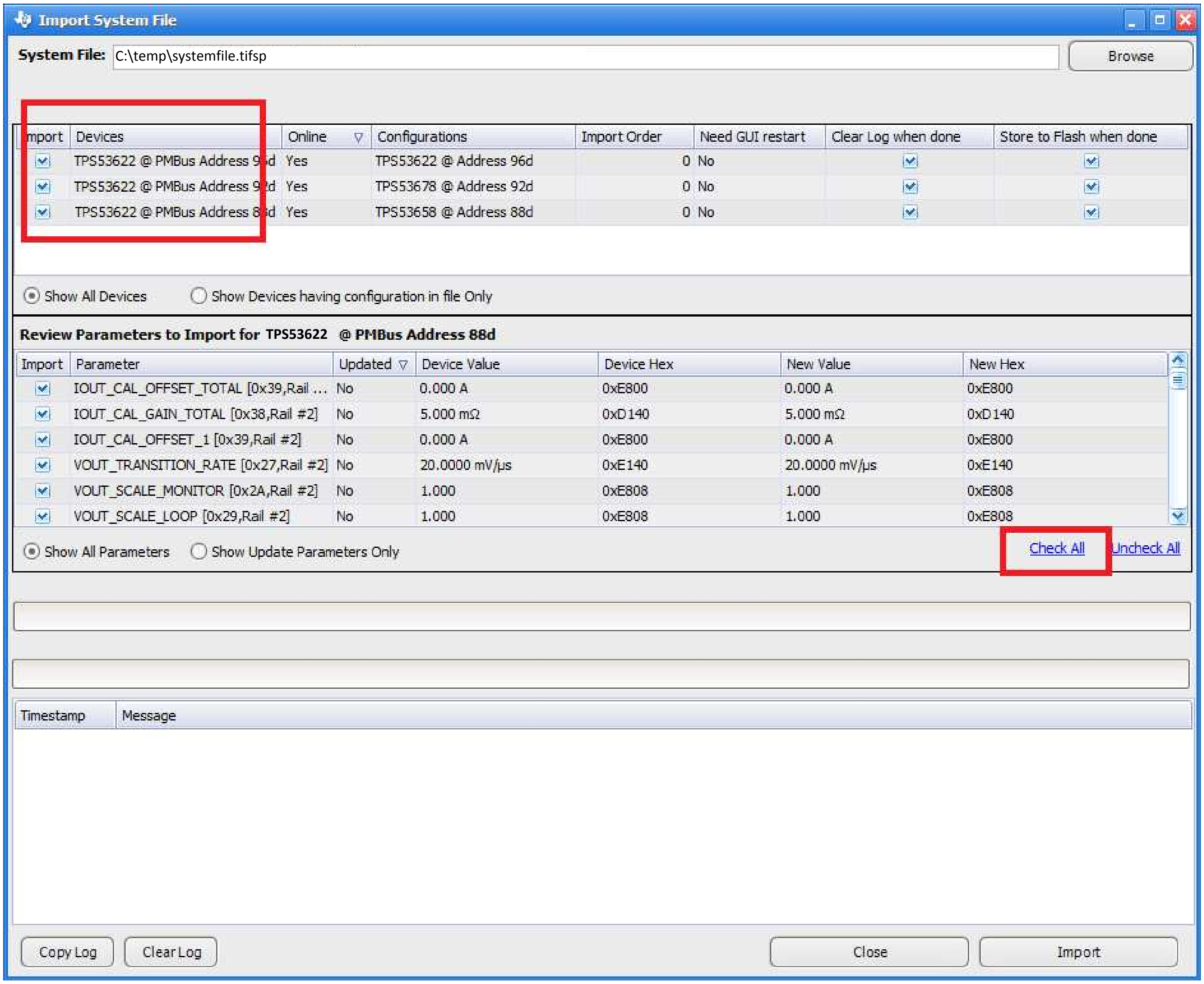

Figure 26. Check All Parameters

7. Click **Import**.
8. Review the messages to make sure each device program is correctly. Imported *System* files contain an NVM store operation for each device by default. They need no separate NVM store command.

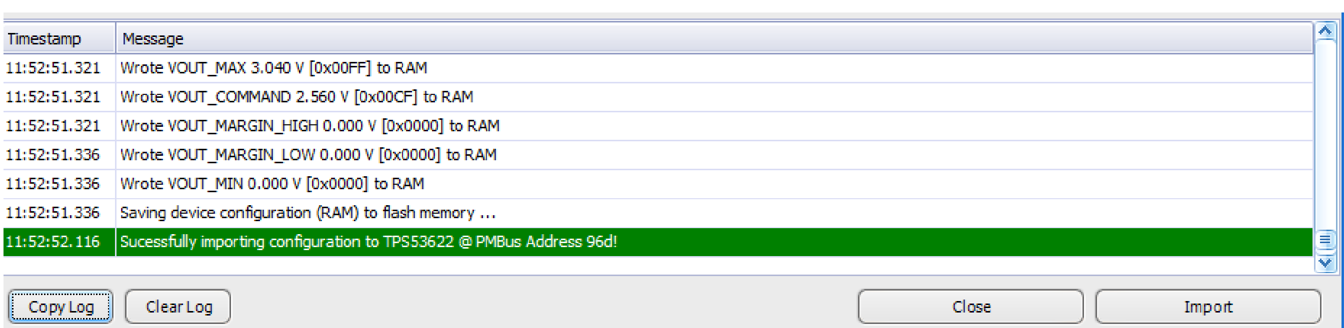

Figure 27. System File Import Messages

3.1.5 Export CSV Script for Third Party Tools

The *Fusion Digital Power Designer* software also exports configuration data to a comma-separated text file (.csv). Third-party software or programming vendors usually require the .csv format because it is easy to adapt to existing software.

1. To export data from a single device, click **Click to Configure Device** from the **System View**.
2. From the **File** menu, select **Export...**

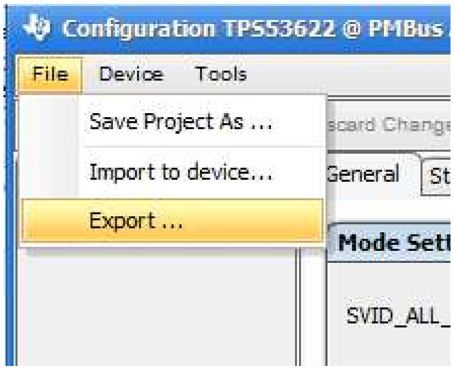

Figure 28. Device File Export

3. From the **Device Export** page, click **PMBus Programmer Script** tab.
4. Use the settings shown in Figure 29 for most situations. Third-party software sometimes has other requirements.

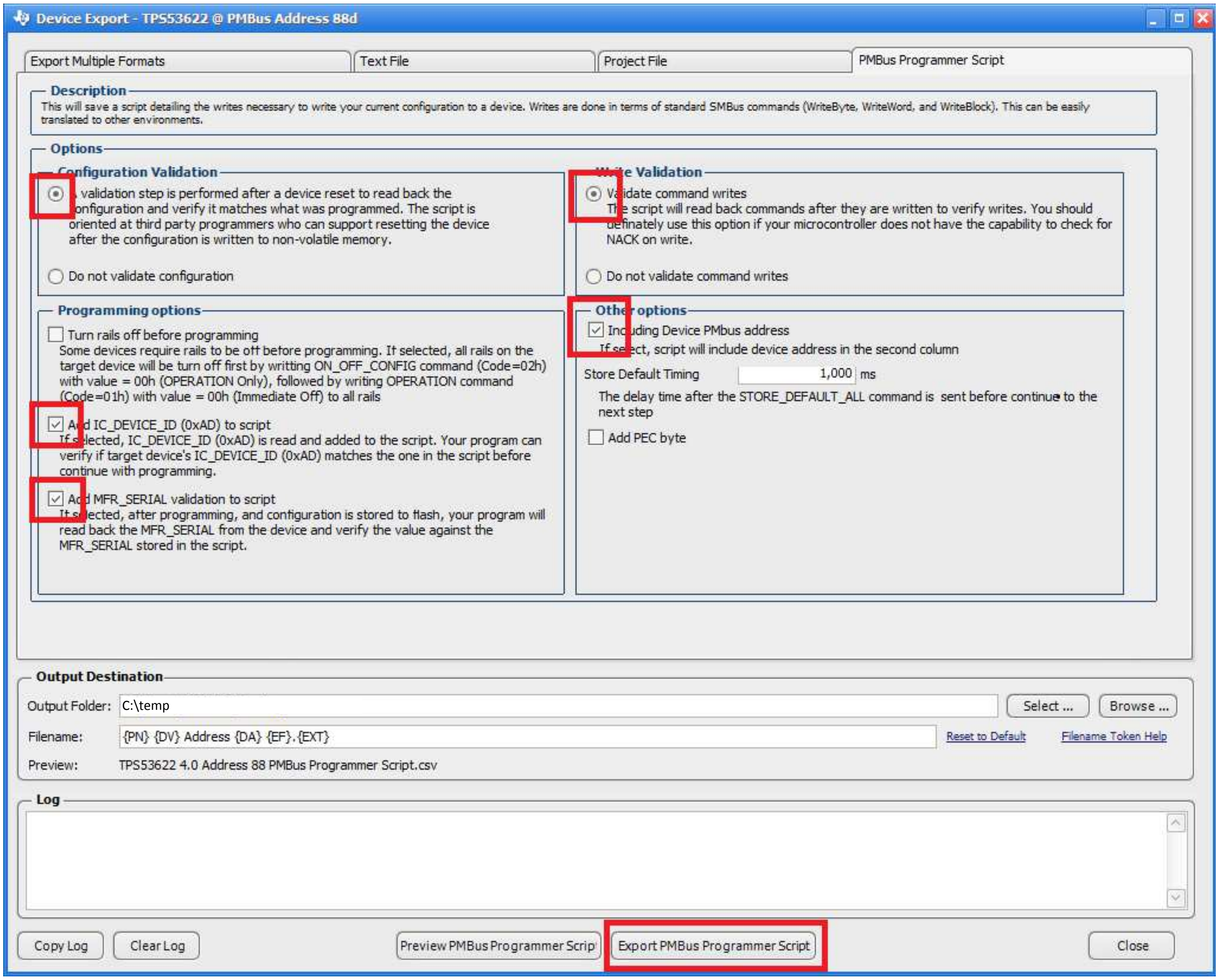

Figure 29. Recommended CSV Export Settings

5. After you complete the settings, click **Export PMBus Programmer Script** .
6. The system generates a text file, and saves it to the path you select as the output folder.

3.1.6 Compare Two *Project* Files

This procedure applies to *Project* files created in *Fusion Digital Power Designer 7.0* and later. This software compares *Project* files only. It does not compare *Fusion Manufacturing* scripts.

1. From the **System View**, click on a device name, and click **Configure Device**.
2. From the **Tools** menu, select **Device/Project Configuration Compare**

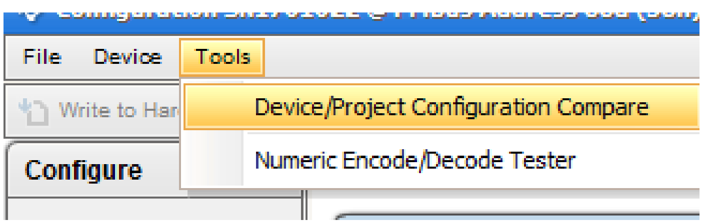

Figure 30. Device/Project Configuration Compare

3. Select two or more *Project* files to compare.

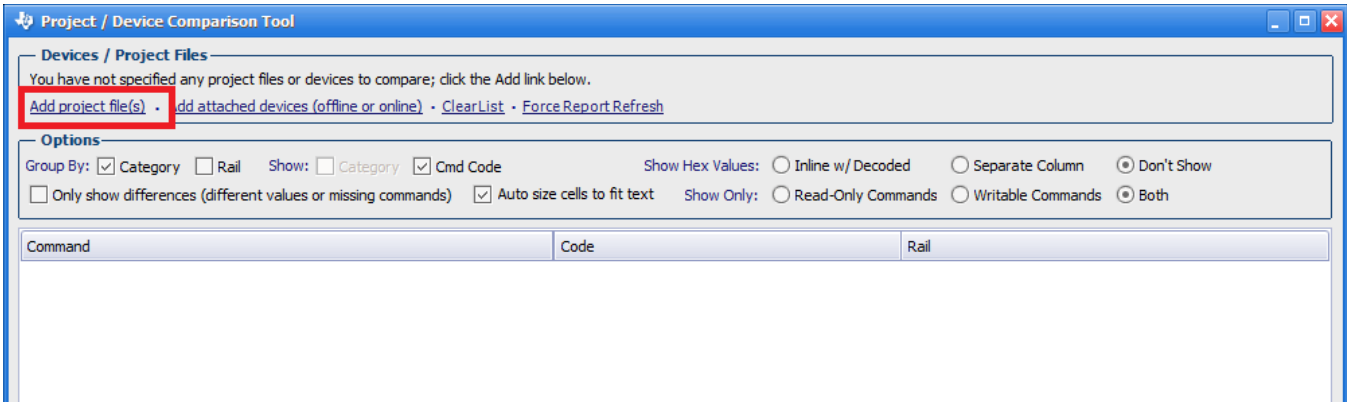

Figure 31. Compare Project Files

4. The **Results** window shows the differences between the commands in a column for each device.

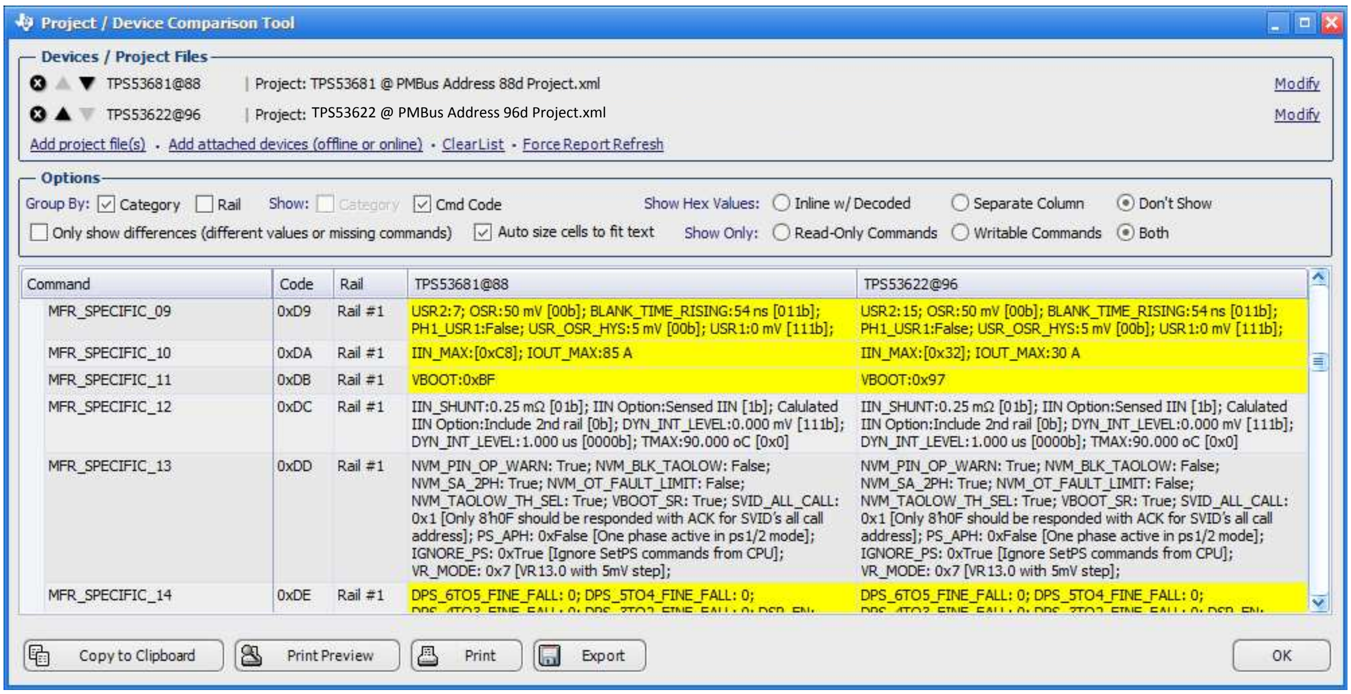

Figure 32. ***Project*** **File Compare**

3.1.7 Build a System File from Existing *Project* Files

New projects often share components with previous projects. With the *Fusion Digital Power Designer* software, designers can assemble a *System* (.tifsp) file using controller devices from existing *Project* files or *System* files. Individual projects must not be modified offline. However you can safely assemble existing *Project* files into a *System* file when the software operates in offline mode.

1. Start *Fusion Digital Power Designer* in offline mode.
2. Open an existing *System* file.
3. From the **System View**, select **Build System**.

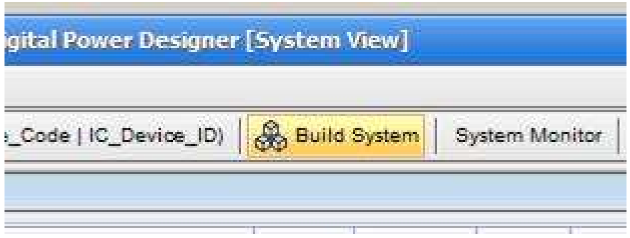

Figure 33. Build System

4. From the **Define System** window, select **Modify an existing system**.

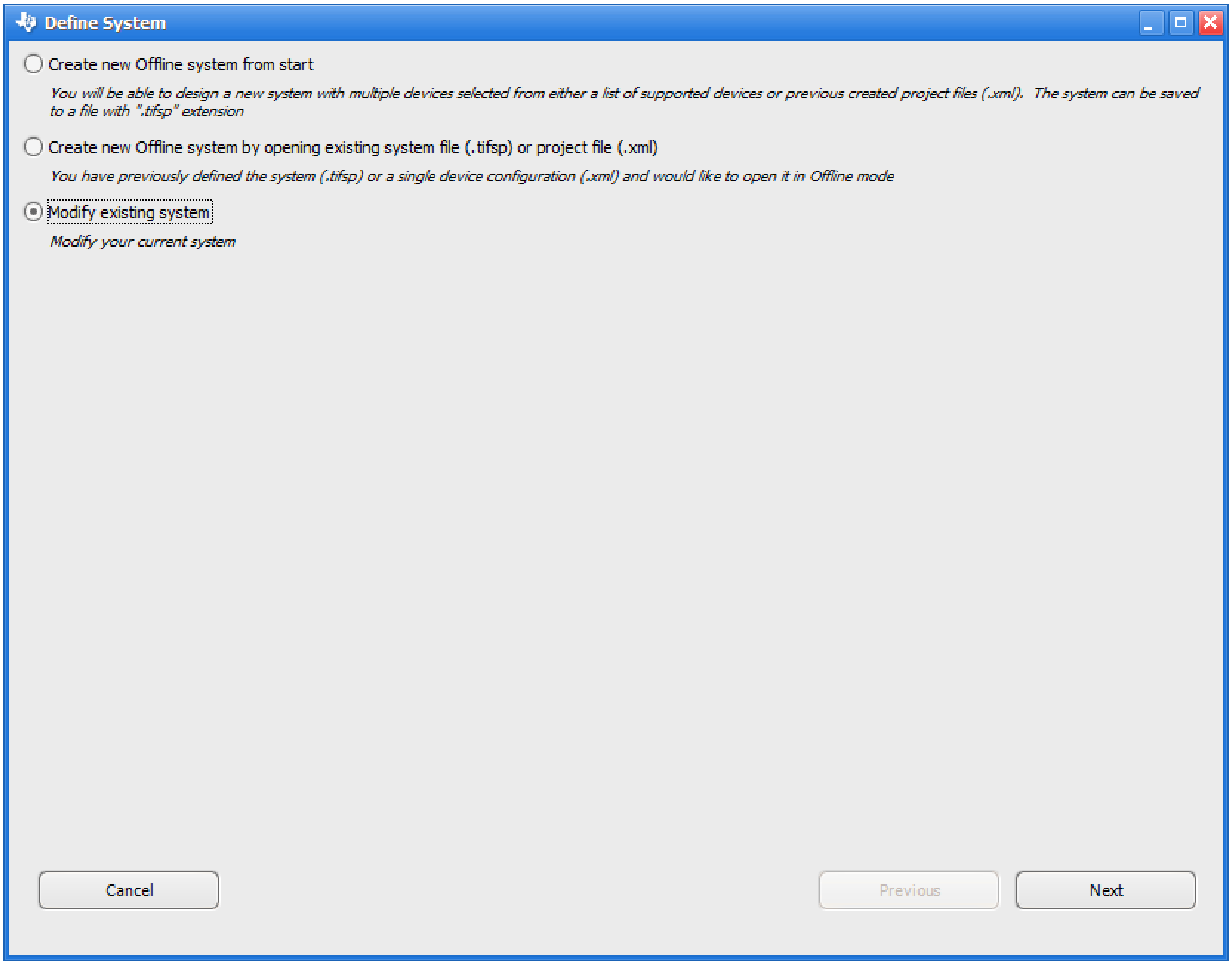

Figure 34. Modify Existing System

5. Click **Next**.
6. The software displays the selected *Project* file names, the components and the addresses.
7. Click **Remove** to delete a device if needed.

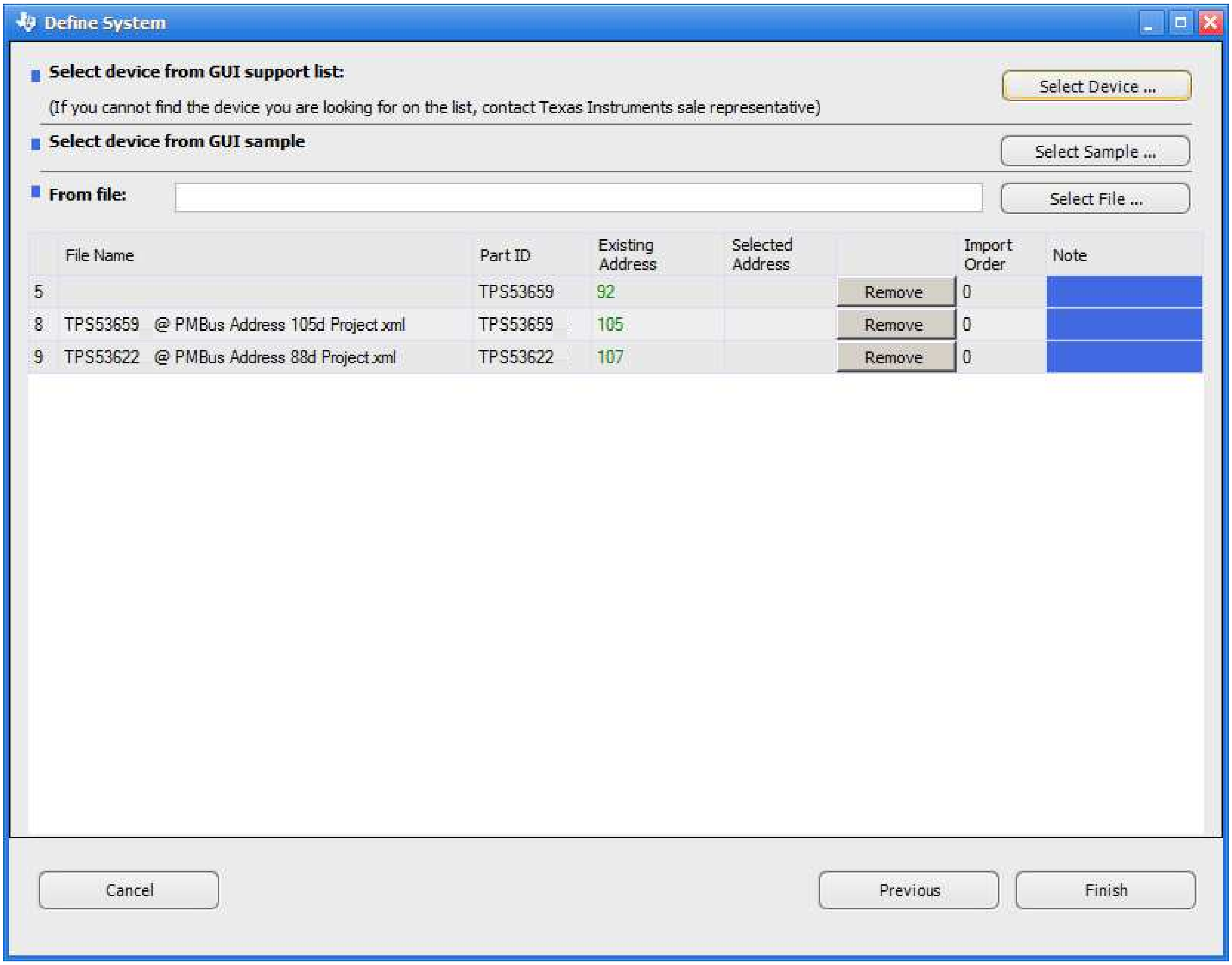

Figure 35. Existing *Project* File Devices

8. To add a *Project* file, click **Select File**.
9. Browse to the *Project* file.
10. Update the PMBus address if necessary by clicking in the **Selected Addresses** cell.
11. Repeat steps 8 through 10 to add more *Project* files to the system.
12. Click **Finish**.

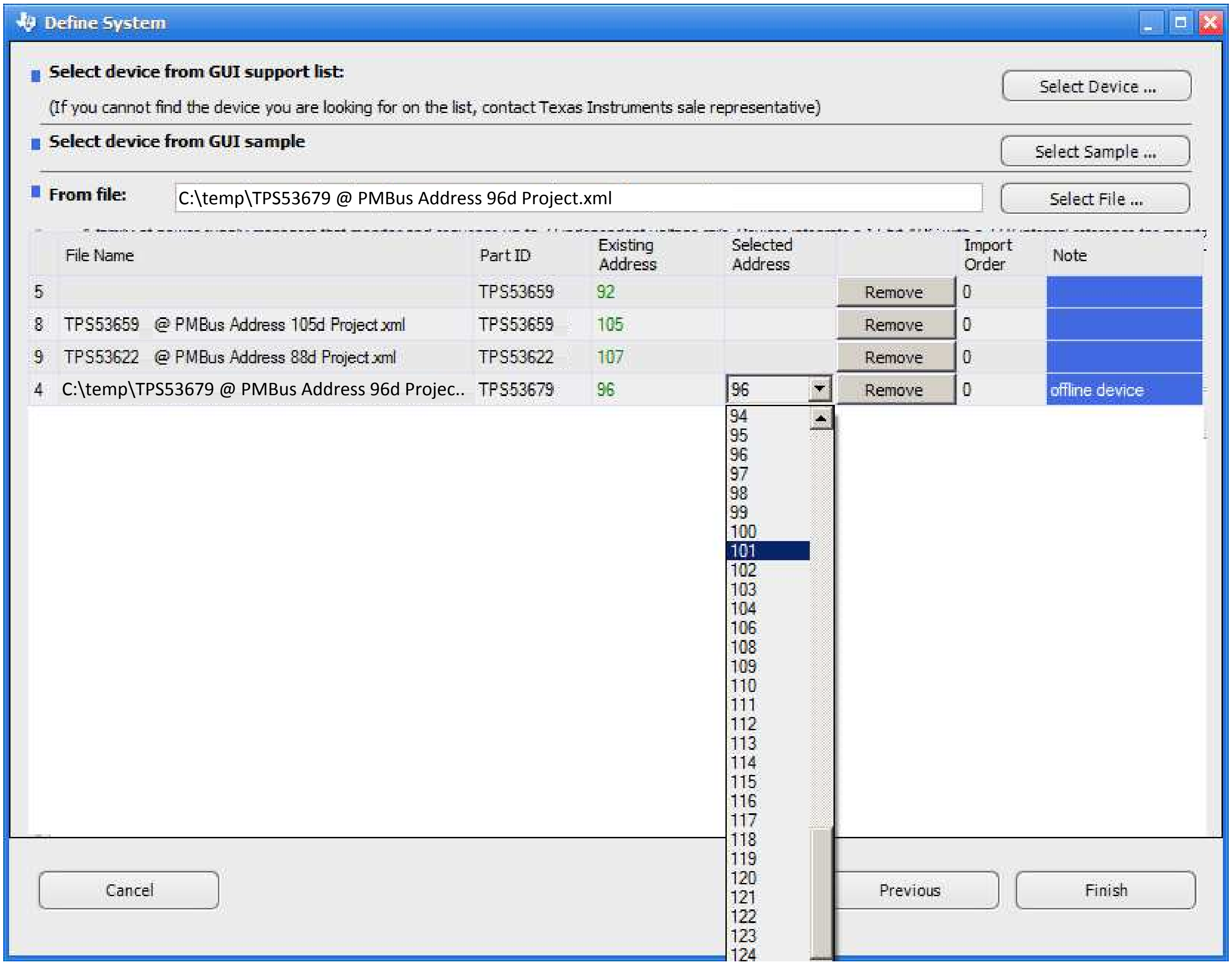

Figure 36. Add a *Project* File to a *System* File

3.2 *Fusion Manufacturing Tool*

3.2.1 Import a *System* File

Use the *Fusion Manufacturing Tool* in mass-production environments to import a *System* file (.tifsp) to an online board.

To download the *Fusion Manufacturing Tool* , click here: http://www.ti.com/tool/FUSION_MFR_GUI. .

1. Build a Fusion System File (.tifsp) using *Fusion Digital Power Designer*
2. Apply power to the target board
3. Connect the USB-to-GPIO interface adapter
4. Start the *Fusion Manufacturing Tool*
5. Click the **Load Script** tab.
6. Click **Browse** to select a **.tifsp** file.

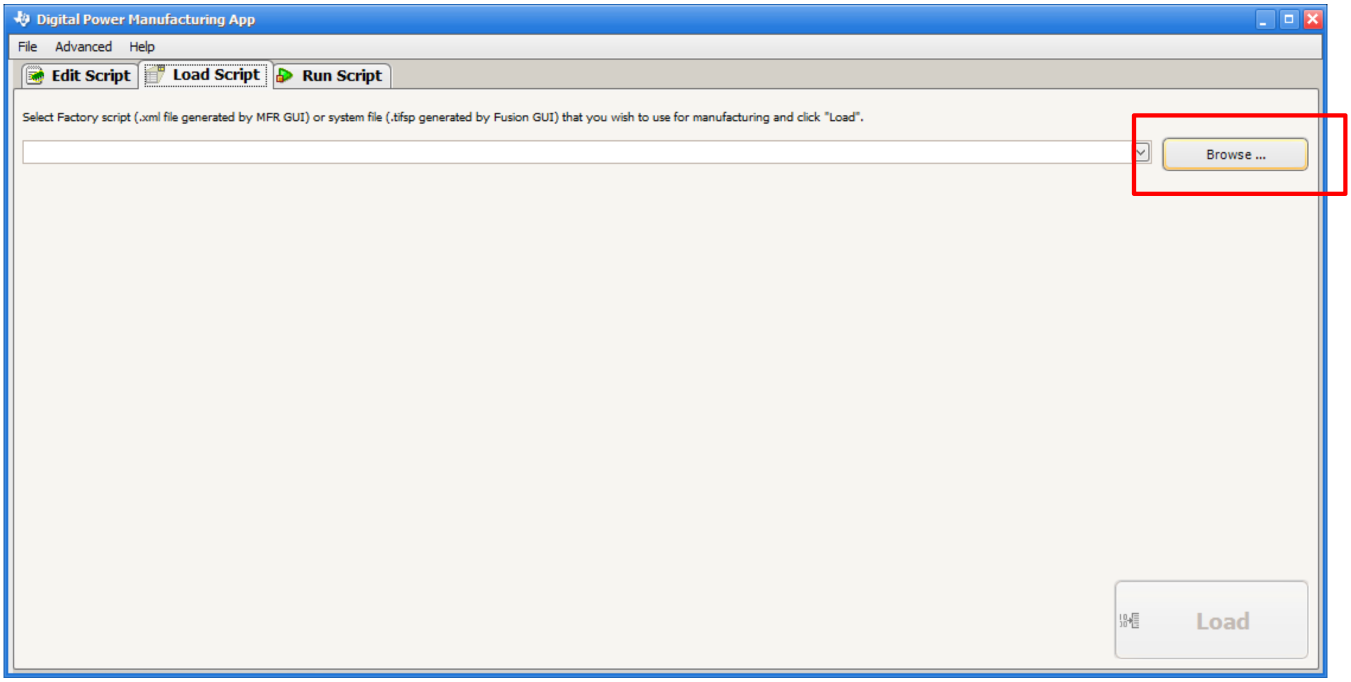

Figure 37. Browse for *System* File

7. After the software loads the *System* file, it shows a list of available devices.
8. Review this list to make sure the correct *System* files are selected.
9. Click **Load Script**.

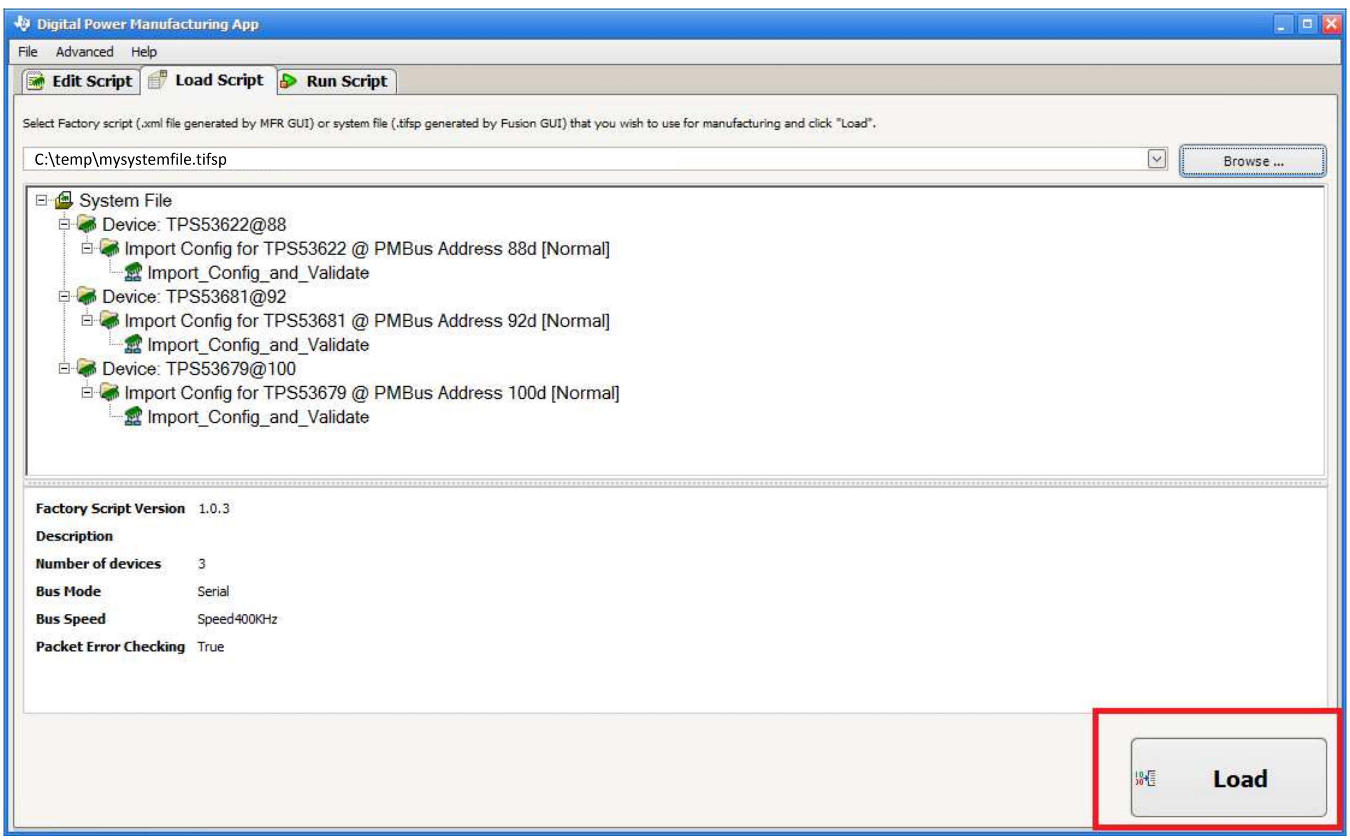

Figure 38. Load the Script

10. Click **Scan** to ensure that the USB-to-GPIO interface adapter recognizes each device in the *System* file.
11. Click **Start** to upload the file to the target board.

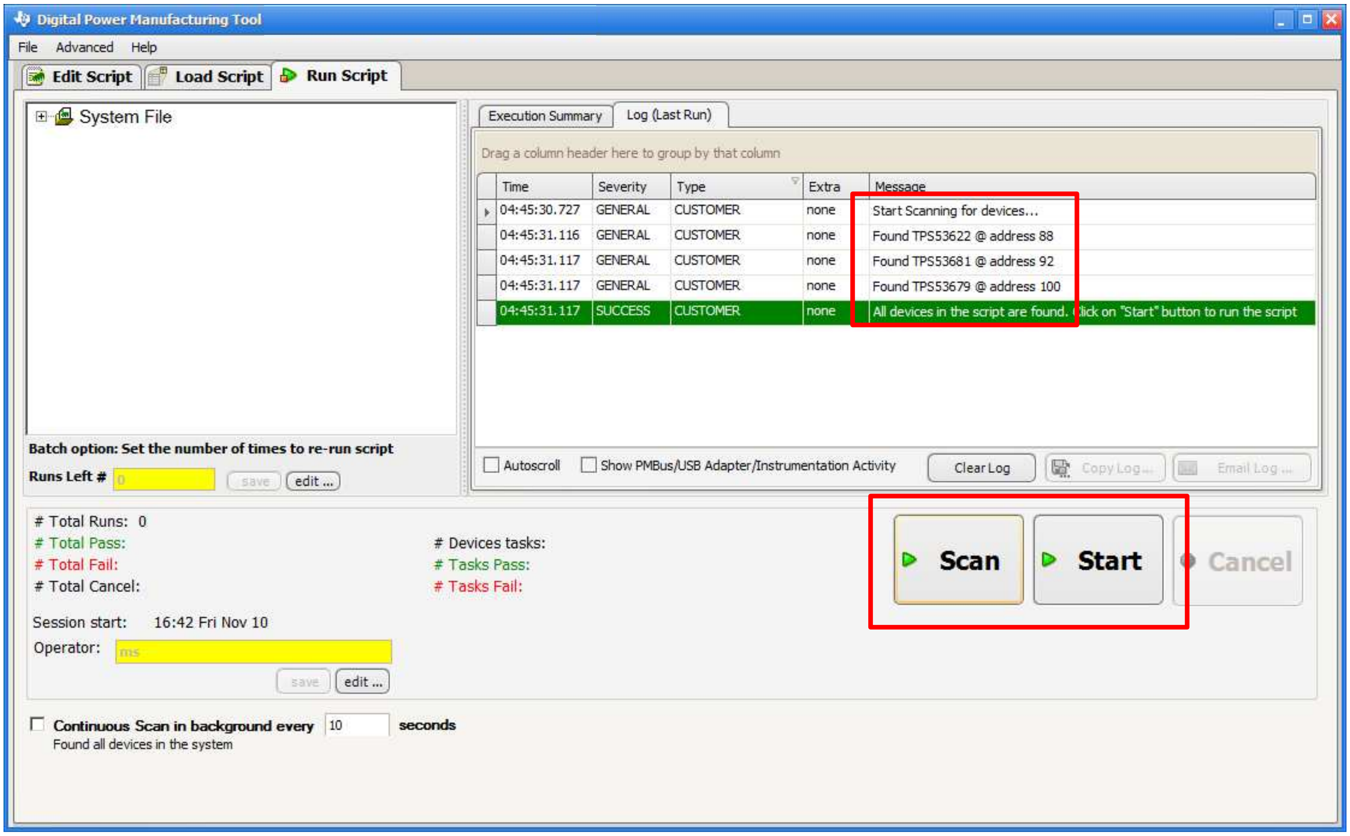

Figure 39. Scan and Start

12. The **Result Summary** dialog box appears. Click **OK**.
13. Disconnect power from the target board.
14. Repeat from step 10 for all remaining boards.

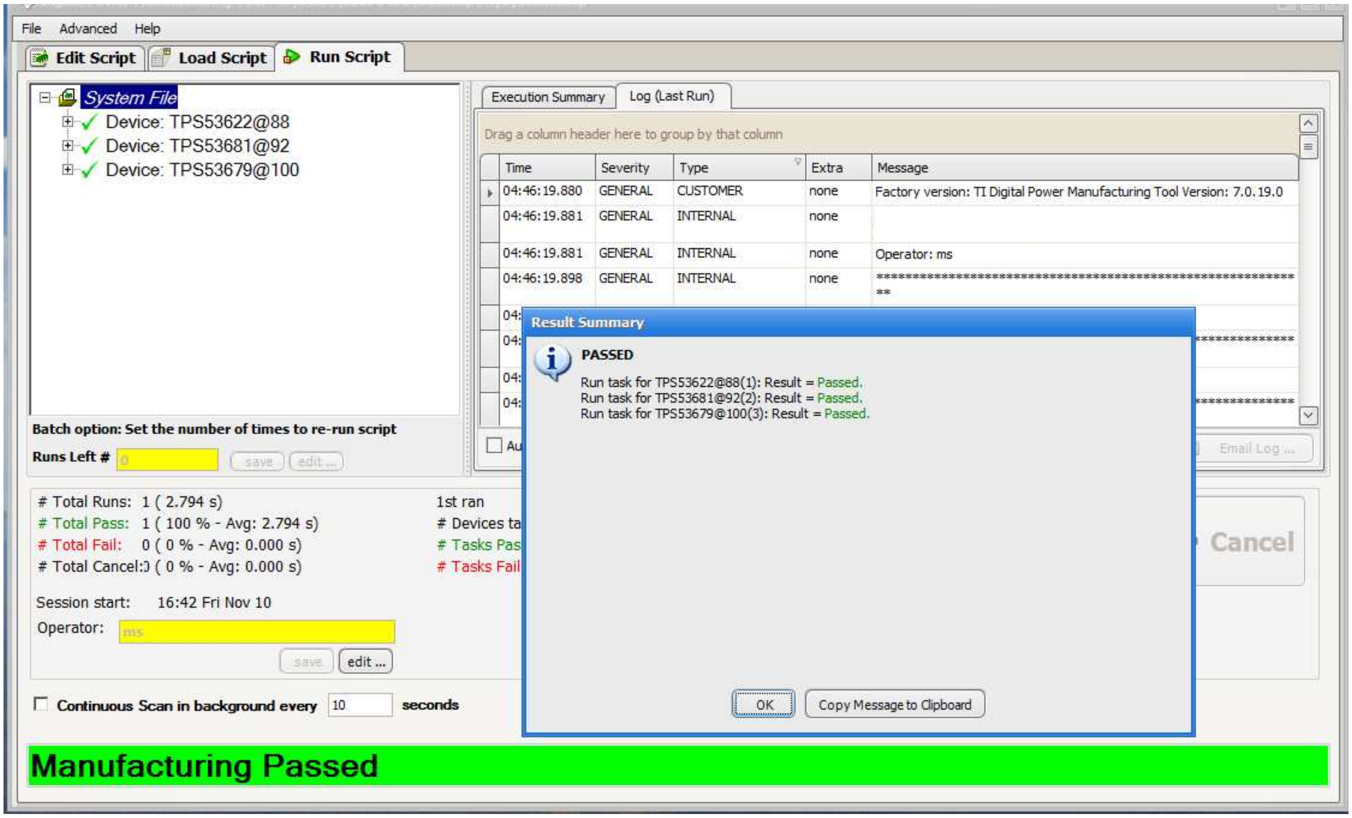

Figure 40. Manufacturing Passed

15.

3.3 TI Programming Board

TI can provide a limited number of socketed programming boards for engineering and pre-production programming use. The programming board allows fast programming of prototyping samples or small production quantities. Boards are available for the unique package and pinout combinations that are supported by TI VR13 controllers. Request a board from your local field representative or sales representative.

This section describes the various connectors and components of the TI socketed programming board, and how to install devices properly to the socket.

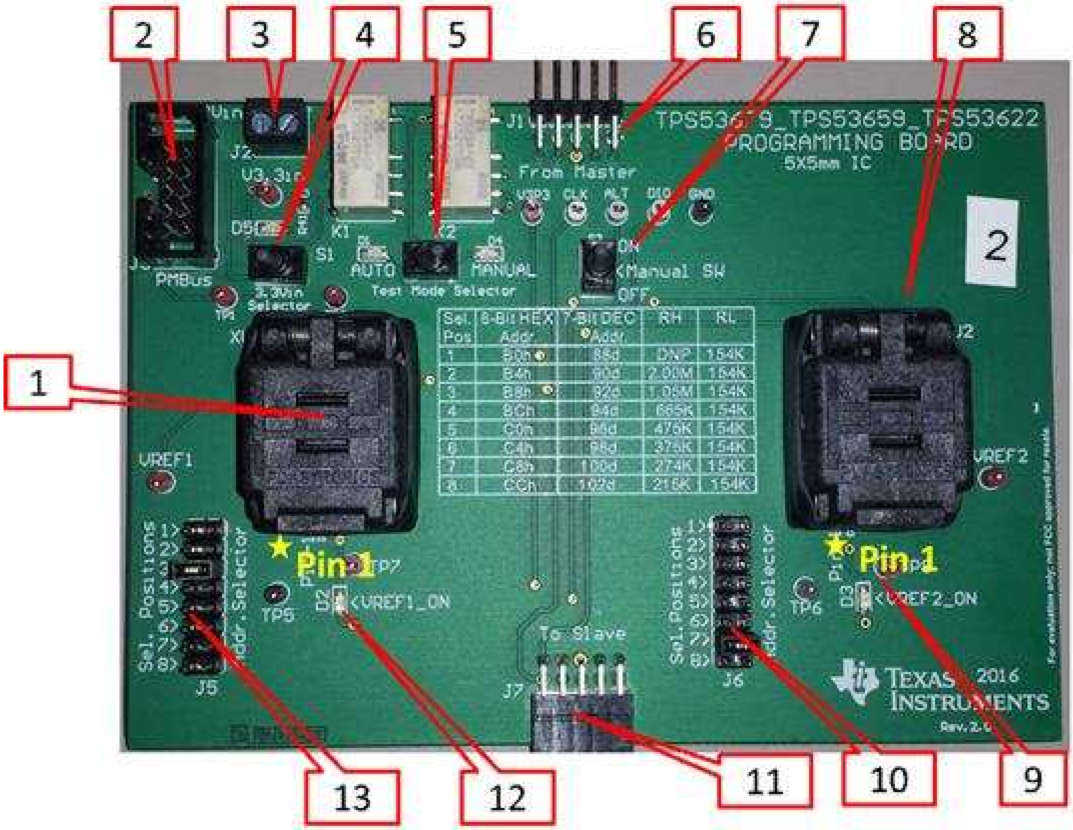

Figure 41. Socketed Programming Board

3.3.1 Board and Connector Description

1. First 40-pin socket. Place DUT #1 in this socket, with pin 1 in the corner marked with a yellow star symbol.
2. 10-pin keyed connector for TI USB-to-GPIO adapter.
3. 2-pin connector (J2) for external 3.3V power supply
4. 3-position switch (S1). Determines whether the 3.3V power supply is sourced from external connector (J2):
   - Left position: sources 3.3-V power supply from PMBus adapter
   - Middle position: disconnected 3.3-V power
   - Right position: sources 3.3-V power supply from J2
5. 2-position switch (test mode selector). Switches between manual power-on and power-off (using S 2), or automatic control.
   - Left position. Automatic control. PMBus CNTL pin controls 3.3-V power applied to DUT #1 and DUT #2.
   - “Right position. Manual control. Switch S2 controls 3.3-V power applied to DUT #1 and DUT #2.

   a.. b.
6. Multiple board male connector (J1). Use J1 and J7 to mate multiple programming boards in series.
7. 2-position switch (S2). Toggles power applied to the socketed devices, when manual power control is selected using the test mode selector switch.
   - Up position: power is applied to DUTs (only in Manual mode) .
   - Down position: Power is not applied to DUTs (only in Manual mode).
8. Second 40-pin socket. Place DUT #2 in this socket, with pin 1 in the corner marked with a yellow star

symbol.

9. VREF2_ON. LED illuminates when DUT#2 successfully powers up (VREF voltage increased to approximately 1.7 V)
10. Address selection for DUT#2. Changing the jumper position according to the table printed on the silkscreen of the board changes the PMBus address that DUT#2 responds to.
11. Multiple board female connector (J7). Use J1 and J7 to mate multiple programming boards in series.
12. VREF1_ON. LED illuminates when DUT#2 successfully powers up (VREF voltage increased to approximately 1.7 V)

3.3.2 Placing devices in the 40-pin socket

1. Use any of these methods to make sure that power is not applied to the sockets before placing or replacing devices into this socket.
   - Move S1 to the middle position, so that neither the PMBus adaptor nor the J2 connector supplies power
   - Remove the power supply to J2 when S1 is in the right position.
   - Remove the PMBus adaptor connection when S1 is in the left position.
   - Switch S2 to the OFF position when the board operates in Manual mode.
   - Pull the PMBus CNTL pin low when the board operated in Automatic control mode.
2. Properly insert the device to match the Pin 1 marking on the board.
3. Close and latch the socket lid firmly before applying power to the device.
4. Make sure that both VREF_ON LEDs illuminate before you continue NVM programming.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES