Application Report

SLUA890B–August 2018–Revised May 2019

Programming the NVM of PMBus Buck Converters to Support Mass-Production Needs

Shawn Nie

1 Introduction

For mass production, it is necessary to select suitable software and program devices through simpler steps. This application report shows you how to program integrated PMBus converters with customized configurations to support mass production in various applications.

Some of the devices to which this guide applies include the following:

- TPS544B25: A 4.5-V to 18-V VIN, 20-A SWIFT™ synchronous buck converter with PMBus and frequency synchronization.
- TPS544C25: A 4.5-V to 18-V VIN, 20/30-A voltage-mode PMBus SWIFT™ step-down DC/DC converter with FSYNC.
- TPS544B20: A 4.5-V to 18-V VIN, 20-A SWIFT synchronous buck converter with PMBus programmability and monitoring.
- TPS544C20: A 4.5-V to 18-V VIN, 30-A SWIFT with PMBus programmability and voltage, current and temperature monitoring.
- TPS546C20A: A 4.5-V to 18-V VIN, 35-A, 2x stackable synchronous buck converter with PMBus supporting telemetry.
- TPS546C23: A 4.5-V to 18-V VIN, 35-A, 2x stackable synchronous buck converter with PMBus supporting telemetry.

2 Programming Hardware Requirements

The Fusion Digital Power (GUI) configures the devices. The application uses PMBus protocol to communicate with the device over a serial bus by using a TI USB adapter. The GUI launches into online mode if the devices are connected to a PC with a USB adapter as Figure 1 shows. Otherwise, the GUI launches into offline mode.

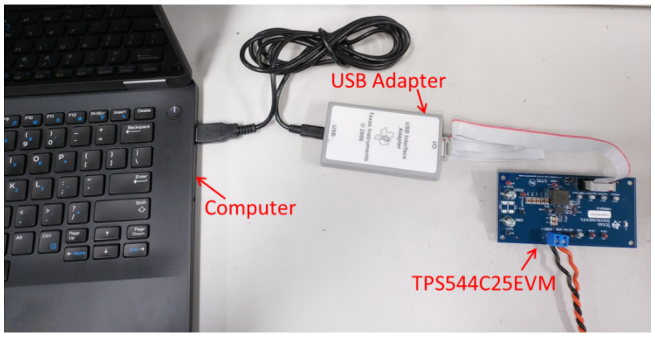


Figure 1. Hardware Requirements

To use any third party programmer or other I2C master to write device configurations to nonvolatile memory of above devices, a corresponding adapter that is compatible with a third party programmer must be used.

The parameters of the devices can be programmed only if the programmer can communicate with device through an I2C. There are three methods to program the parameters:

1. Program before assembling the device on the motherboard. Figure 2 shows the programming socket board for the TPS544C25. Consider the following before designing the schematic:
   - A 5 V (J2) is required to power the AVIN to communicate with the device.
   - The CNTL pin must be pulled to low.
   - See of the PMBus Address section in the TPS544x25 4.5-V to 18-V, 20-A and 30-A SWIFT™ Synchronous Buck Converters with PMBus™ and Frequency Synchronization data sheet to configure the device address. For this design, R7 and R9 are 51.1 k and the address is 36d.
   - For a different PMBus converter, the bypass capacitors from VDD, BP3, and BP6 to GND and resistors between ADDR0 or ADDR1 with GND are different. The values must be selected according to the corresponding data sheet.
   - If a TI USB-To-GPIO adapter is used, R4 and R5 marked by the red fork are not needed, because CLK and DATA are internally pulled up by TI adapter. If you are using the third party programmer and the pull-up resistors cannot be configured by the corresponding adapter, R4 and R5 are required to connect to BP3 (output of the 3.3-V onboard regulator) or an external supply.

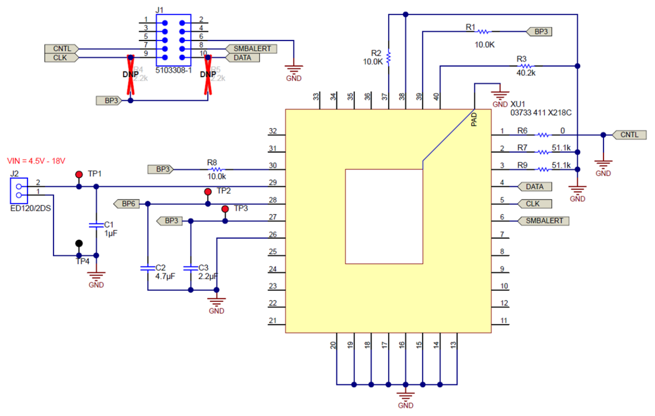


Figure 2. Schematic of The Programming Socket Board for TPS544C25

2. If the devices are already assembled to the motherboard, send the command and data to the device with the onboard controller (such as MCU or FPGA).
3. If the devices are assembled to the motherboard and there are in-circuit test points on the mother board, send the command and data with a different controller board.

Follow these steps to program the overcurrent limit for TPS544C25 to 30 A :

1. Write the IOUT_OC_FAULT_LIMIT command “0x46h” with data “0xF83C”.
2. Write the STORE_USER_ALL command “15h” to store all the data to the NVM (nonvolatile memory).

3 Programming Software Requirements

This document details how to use software to support mass production. It mainly contains Fusion GUI and Fusion Digital Power Manufacturing Tool which are developed by TI.

3.1 Applications of The Fusion Digital Power GUI

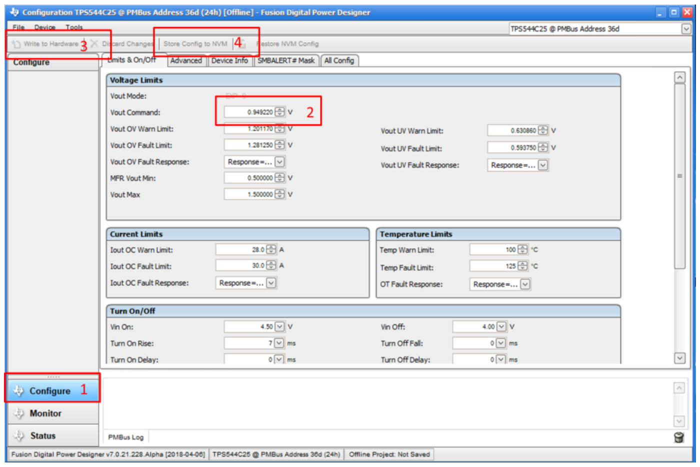
Fusion Digital Power is the Graphical User Interface (GUI) used to configure and monitor select Texas Instruments digital power controllers and sequencer/health monitors.

3.1.1 Configuring The Device

Configure the parameters in the configure window of the Fusion GUI as shown in Figure 3. Follow the steps in the following order:

1. Click Configure.

2. Adjust the voltage limits for the Vout Command.
3. Click Write to Hardware.
4. Click Store Configuration to NVM.
5. When the configuration is completed, save the configuration to the file.



Voltage Limits

Vout Mode:
Vout Command: 0.949220 V
Vout OV Warn Limit: 1.201170 V
Vout OV Fault Limit: 1.281250 V
Vout OV Fault Response: Response=...
MFR Vout Min: 0.500000 V
Vout Max 1.500000 V
Vout UV Warn Limit: 0.630860 V
Vout UV Fault Limit: 0.593750 V
Vout UV Fault Response: Response=...

Current Limits

Iout OC Warn Limit: 28.0 A
Iout OC Fault Limit: 30.0 A
Iout OC Fault Response: Response=...

Temperature Limits

Temp Warn Limit: 100 °C
Temp Fault Limit: 125 °C
OT Fault Response: Response=...

Turn On/Off

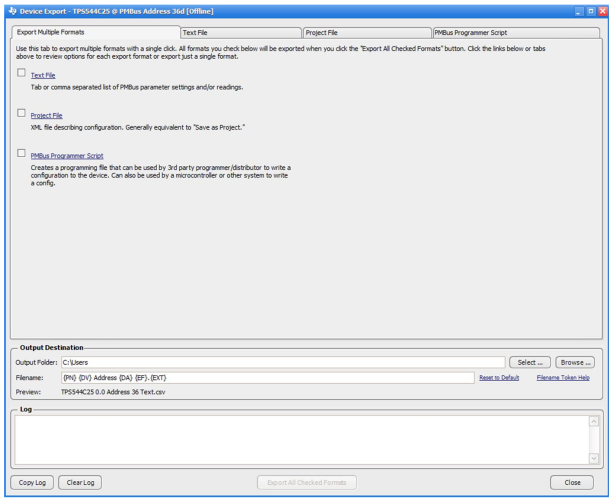
Vin On: 4.50 V
Turn On Rise: 7 ms
Turn On Delay: 0 ms
Vin Off: 4.00 V
Turn Off Fall: 0 ms
Turn Off Delay: 0 ms

Fusion Digital Power Designer v7.0.21.228.Alpha [2018-04-06] | TPS544C25 @ PMBus Address 36d (24h) | Offline Project: Not Saved

Figure 3. Labeled Steps to How to Configure A Device

3.1.2 Exporting A File

To export a file, click on File, then Export. There are multiple export format options to choose from. To save a configuration, click in one or multiple boxes for the following options you want the file to be saved as: Text File, Project File, and PMBus Programmer Script.



Use this tab to export multiple formats with a single click. All formats you check below will be exported when you click the "Export All Checked Formats" button. Click the links below or tabs above to review options for each export format or export just a single format.

☐ Text File
Tab or comma separated list of PMBus parameter settings and/or readings.

☐ Project File
XML file describing configuration. Generally equivalent to "Save as Project."

☐ PMBus Programmer Script
Creates a programming file that can be used by 3rd party programmer/distributor to write a configuration to the device. Can also be used by a microcontroller or other system to write a config.

Output Folder: C:\Users
Filename: {PN} {DV} Address {DA} {EF}.{EXT}
Preview: TPS544C25 0.0 Address 36 Text.csv

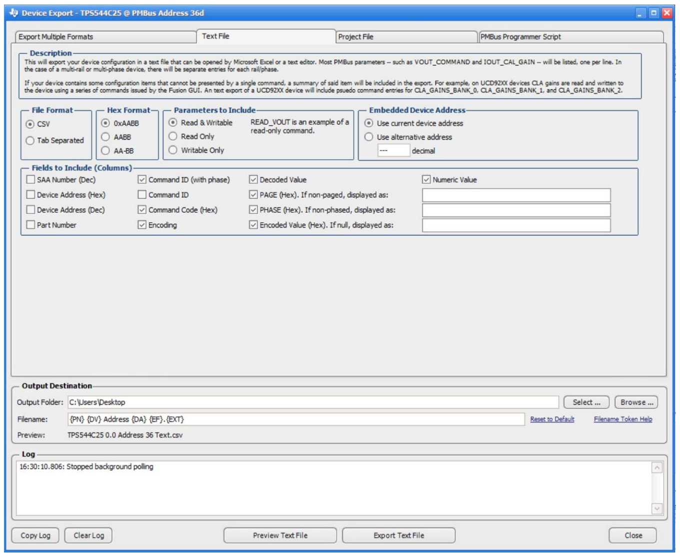
Figure 4. Device Export

Figure 4 shows three different export file formats:

- **Text File (.csv)**: This exports the device configuration in a text file that can be opened with Microsoft Excel or a text editor. You can personalize the format of the text file. Once the format of the text file is established, click Export Text File (shown in Figure 5) to save the text file.
  – **File Format**: Select CSV format to save the file as a Microsoft Excel spreadsheet. Select "Tab Separated" to save the file as a text file.
  – **Hex Format:** Three hex formats (0xAABB, AABB, AA-BB) can be selected.
  – **Parameters to Include:** Select “Read and Writable, Read Only, Writable Only” to decide which will

be saved.

- **Embedded Device Address:** Use “current device address” or enter a decimal number between 1 and 127 as the address.
- **Fields to Include:** Select the fields to be saved in every column.
  - **SAA Number (Dec):** The number of the USB adapters this device is attached to. This is useful if the user is exporting device data for two devices with the same address but on different adapters.
  - **Device Address:** (hexadecimal or decimal) The PMBus address the device is using (or was using at the time of project export in offline mode).
  - **Part Number:** The Texas Instruments' device part number.
  - **Command ID (with phase):** The same command ID in the advanced configure form of the GUI. This is a command ID with the phase number optionally appended. For example, IOUT_CAL_GAIN_1 or READ-TEMPERATURE_2MAX. Phase numbering starts at 1.
  - **Command ID:** A phase number is not appended to the Command ID. For example, there are multiple entries for IOUT_CAL_GAIN_1 for each phase. The phase field determines which phase the record corresponds to . The phase field uses zero-based numbering.
  - **Command Code (Hex):** The PMBus or SMBus command code for the parameter. If the parameter is a meta command, the record does not correspond directly with a PMBus command that the device supports. In this case, this field is empty.
  - **Encoding:** The data type for this parameter. Select "Byte", "Word", or "Block".
  - **Decoded Value:** A textual summary of the value of the parameter. For numerical parameters such as VOUT_COMMAND, this includes units. An example is “1.25 V”.
  - **PAGE (Hex):** The page that the parameter corresponds to. Page numbering starts at 0. If the parameter is global to the device and not PAGED, then the field is empty.
  - **PHASE (Hex):** The phase that the parameter corresponds to. Phase numbering starts at 0. If the parameter never varies when PHASE changes, the field is empty. Some parameters support a special PHASE = 0xFF, such as READ_IOUT.
  - **Encoded Value (Hex):** This is raw data sent to and from the device when the device last read the parameter. This value is displayed in hex.



Description

This will export your device configuration in a text file that can be opened by Microsoft Excel or a text editor. Most PMBus parameters -- such as VOUT_COMMAND and IOUT_CAL_GAIN -- will be listed, one per line. In the case of a multi-rail or multi-phase device, there will be separate entries for each rail/phase.

If your device contains some configuration items that cannot be presented by a single command, a summary of said item will be included in the export. For example, on UCD92XX devices CLA gains are read and written to the device using a series of commands issued by the Fusion GUI. An text export of a UCD92XX device will include psuedo command entries for CLA_GAINS_BANK_0, CLA_GAINS_BANK_1, and CLA_GAINS_BANK_2.

File Format: [✓ CSV] Tab Separated

Hex Format: [✓ 0xAABB] AABB AA-BB

Parameters to Include: [✓ Read & Writable] Read Only Writable Only — READ_VOUT is an example of a read-only command.

Embedded Device Address: [✓ Use current device address] Use alternative address --- decimal

Fields to Include (Columns):

☐ SAA Number (Dec)	✓ Command ID (with phase)	✓ Decoded Value	✓ Numeric Value
☐ Device Address (Hex)	☐ Command ID	✓ PAGE (Hex). If non-paged, displayed as:	
☐ Device Address (Dec)	✓ Command Code (Hex)	✓ PHASE (Hex). If non-phased, displayed as:	
☐ Part Number	✓ Encoding	✓ Encoded Value (Hex). If null, displayed as:	

Output Destination

Output Folder: C:\Users\Desktop

Filename: {PN} {DV} Address {DA} {EF}.{EXT}

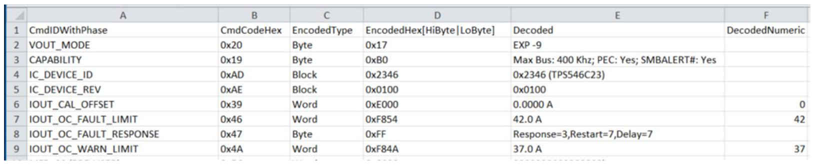
Preview: TPS544C25 0.0 Address 36 Text.csv

Log

```
16:30:10.806: Stopped background polling
```

Figure 5. Exporting Text File

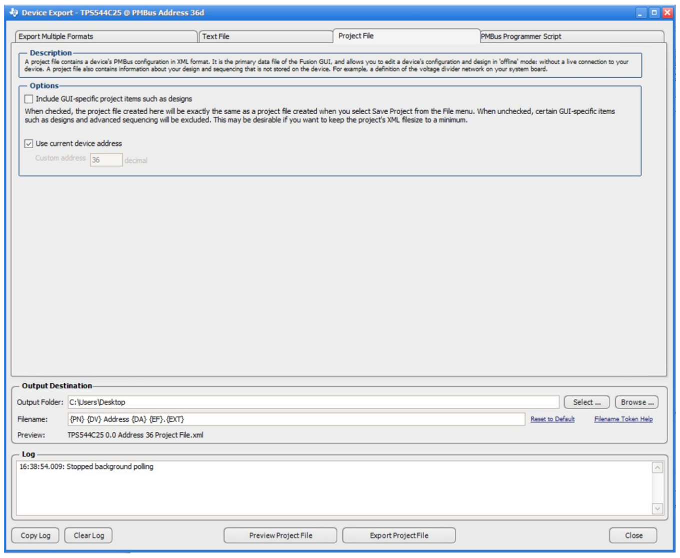
Figure 5 is a saved .csv file. Most PMBus parameters such as VOUT_MODE and IOUT_OC_FAULT_LIMIT are listed one parameter per line. In the case of a multirail or multiphase device, there are separate entries for each rail or phase. For example, “IOUT_OC_FAULT_LIMIT” is the Command ID and 0x46 is the register address of this command. 0xF854 is the code data written to this register, which corresponds to 42A. The "Encoded Type" column lists the type of 0xF854.



	A	B	C	D	E	F
1	CmdIDWithPhase	CmdCodeHex	EncodedType	EncodedHex[HiByte|LoByte]	Decoded	DecodedNumeric
2	VOUT_MODE	0x20	Byte	0x17	EXP -9	
3	CAPABILITY	0x19	Byte	0xB0	Max Bus: 400 Khz; PEC: Yes; SMBALERT#: Yes	
4	IC_DEVICE_ID	0xAD	Block	0x2346	0x2346 (TPS546C23)	
5	IC_DEVICE_REV	0xAE	Block	0x0100	0x0100	
6	IOUT_CAL_OFFSET	0x39	Word	0xE000	0.0000 A	0
7	IOUT_OC_FAULT_LIMIT	0x46	Word	0xF854	42.0 A	42
8	IOUT_OC_FAULT_RESPONSE	0x47	Byte	0xFF	Response=3,Restart=7,Delay=7	
9	IOUT_OC_WARN_LIMIT	0x4A	Word	0xF84A	37.0 A	37

Figure 6. Text File

- **Project File (.xml)**: A project file contains the PMBus configuration of a device in XML format. It is the primary data file of the fusion GUI, and allows the user to edit the configuration of the device and design in offline mode without a live connection to the device. A project file contains information about the design and sequencing that is not stored on the device. For example, a definition of the voltage divider network on the system board. Click "Export Project File" to save the project file (see Figure 7).



Device Export - TPS544C25 @ PMBus Address 36d

Export Multiple Formats | Text File | Project File | PMBus Programmer Script

Description
A project file contains a device's PMBus configuration in XML format. It is the primary data file of the Fusion GUI, and allows you to edit a device's configuration and design in 'offline' mode: without a live connection to your device. A project file also contains information about your design and sequencing that is not stored on the device. For example, a definition of the voltage divider network on your system board.

Options
☐ Include GUI-specific project items such as designs
When checked, the project file created here will be exactly the same as a project file created when you select Save Project from the File menu. When unchecked, certain GUI-specific items such as designs and advanced sequencing will be excluded. This may be desirable if you want to keep the project's XML filesize to a minimum.

☑ Use current device address
Custom address 36 decimal

Output Destination
Output Folder: C:\Users\Desktop
Filename: {PN} {DV} Address {DA} {EF}.{EXT}
Preview: TPS544C25 0.0 Address 36 Project File.xml

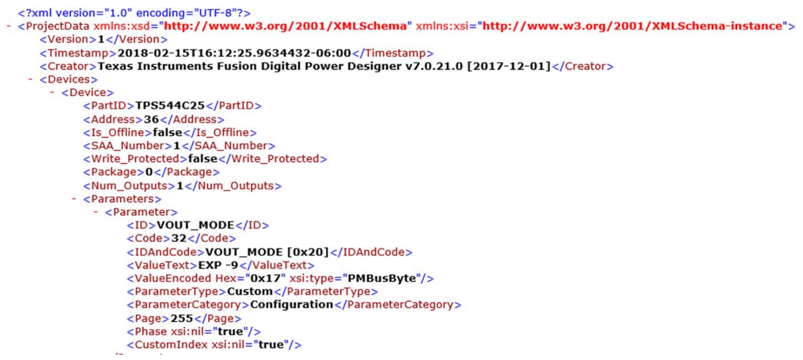
Log
```
16:38:54.009: Stopped background polling
```

Figure 7. Exporting Project File

When you check "Include GUI-specific project items such as designs", the project file created here can be exactly the same as a project file created when you select "Save Project" from the "File" menu. When unchecked, GUI-specific items such as designs and advanced sequencing, can be excluded.

Figure 8 is a part of project file content. The following numbers correspond to specific project information.

1. **Version:** XML version= "1.0"
2. **Timestamp:** The time the project file was created.
3. **Creator:** The software that created the project file.
4. **Part ID:** The part number
5. **Address:** The device address
6. **Is offline:** Whether it is online or offline
7. **Write Protected:** If the device is write protected, the data cannot be written.

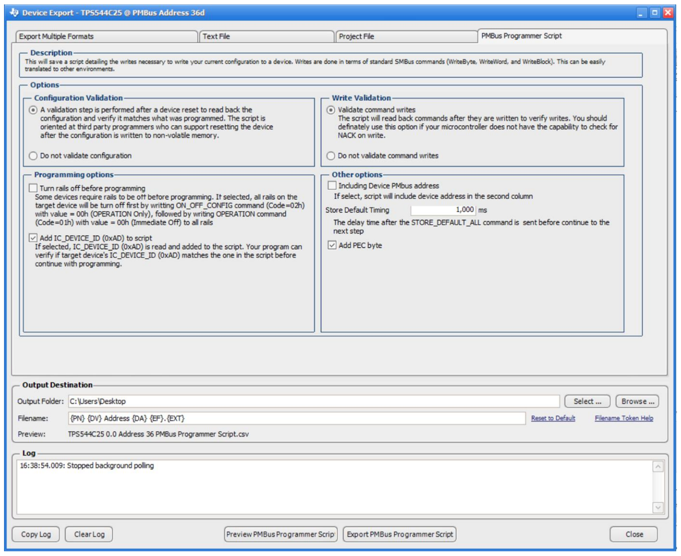


```
  <?xml version="1.0" encoding="UTF-8"?>
- <ProjectData xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <Version>1</Version>
     <Timestamp>2018-02-15T16:12:25.9634432-06:00</Timestamp>
     <Creator>Texas Instruments Fusion Digital Power Designer v7.0.21.0 [2017-12-01]</Creator>
   - <Devices>
      - <Device>
           <PartID>TPS544C25</PartID>
           <Address>36</Address>
           <Is_Offline>false</Is_Offline>
           <SAA_Number>1</SAA_Number>
           <Write_Protected>false</Write_Protected>
           <Package>0</Package>
           <Num_Outputs>1</Num_Outputs>
         - <Parameters>
            - <Parameter>
                 <ID>VOUT_MODE</ID>
                 <Code>32</Code>
                 <IDAndCode>VOUT_MODE [0x20]</IDAndCode>
                 <ValueText>EXP -9</ValueText>
                 <ValueEncoded Hex="0x17" xsi:type="PMBusByte"/>
                 <ParameterType>Custom</ParameterType>
                 <ParameterCategory>Configuration</ParameterCategory>
                 <Page>255</Page>
                 <Phase xsi:nil="true"/>
                 <CustomIndex xsi:nil="true"/>
```

Figure 8. Project File

- **Script File**: A setup of I2C, SMBus and PMBus commands that configure the device. These commands are shown in the script file. The formats are text (.txt) and Excel (.csv). Fusion GUI and third party programmers can use a script file. The file contains steps to program a single device using PMBus commands. Click "Export PMBus Programmer Script" to save the file as Figure 9 shows.

  Figure 9 shows four options that can be selected when the script file is saved.



Device Export - TPS544C25 @ PMBus Address 36d

Description
This will save a script detailing the writes necessary to write your current configuration to a device. Writes are done in terms of standard SMBus commands (WriteByte, WriteWord, and WriteBlock). This can be easily translated to other environments.

Options

Configuration Validation
[✓ A validation step is performed after a device reset to read back the configuration and verify it matches what was programmed. The script is oriented at third party programmers who can support resetting the device after the configuration is written to non-volatile memory.]
Do not validate configuration

Write Validation
[✓ Validate command writes] The script will read back commands after they are written to verify writes. You should definately use this option if your microcontroller does not have the capability to check for NACK on write.
Do not validate command writes

Programming options
☐ Turn rails off before programming
Some devices require rails to be off before programming. If selected, all rails on the target device will be turn off first by writting ON_OFF_CONFIG command (Code=02h) with value = 00h (OPERATION Only), followed by writing OPERATION command (Code=01h) with value = 00h (Immediate Off) to all rails
☑ Add IC_DEVICE_ID (0xAD) to script
If selected, IC_DEVICE_ID (0xAD) is read and added to the script. Your program can verify if target device's IC_DEVICE_ID (0xAD) matches the one in the script before continue with programming.

Other options
☐ Including Device PMbus address
If select, script will include device address in the second column
Store Default Timing 1,000 ms
The delay time after the STORE_DEFAULT_ALL command is sent before continue to the next step
☑ Add PEC byte

Output Destination
Output Folder: C:\Users\Desktop
Filename: {PN} {DV} Address {DA} {EF}.{EXT}
Preview: TPS544C25 0.0 Address 36 PMBus Programmer Script.csv

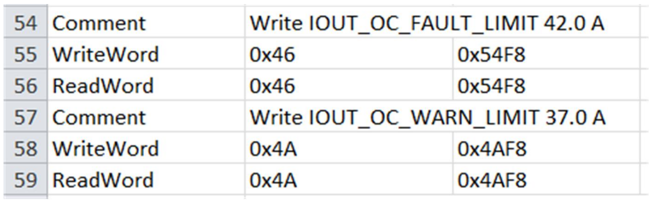
Log
16:38:54.009: Stopped background polling

Figure 9. Exporting Script File

– **Configuration Validation:** A validation step is performed after a device reset to read back the configuration and verify that the configuration matches what was programmed. The script is designed for third party programmers who can support resetting the device after the configuration is written to nonvolatile memory.

– **Write Validation:** When checking the validate command write, the script reads back commands after they are written to verify writes. Use this option if the microcontroller cannot check NACK on write.

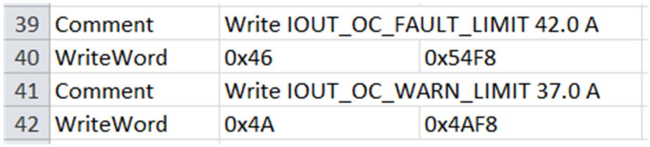
  When checking the validate command write, the user can view the read back commands in the saved script file. The type of the saved code in the script file is changed from 0xABCD to 0xCDAB. Figure 10 shows the IOUT_OC_FAULT_LIMIT command. The code of 42A is 0xF854, but the code is saved as 0x54F8.



54	Comment	Write IOUT_OC_FAULT_LIMIT 42.0 A	
55	WriteWord	0x46	0x54F8
56	ReadWord	0x46	0x54F8
57	Comment	Write IOUT_OC_WARN_LIMIT 37.0 A	
58	WriteWord	0x4A	0x4AF8
59	ReadWord	0x4A	0x4AF8

Figure 10. Checking "Validate Command Write"

If unchecked, the readback commands are not saved in the script file as Figure 11 shows.

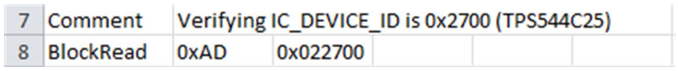


39	Comment	Write IOUT_OC_FAULT_LIMIT 42.0 A	
40	WriteWord	0x46	0x54F8
41	Comment	Write IOUT_OC_WARN_LIMIT 37.0 A	
42	WriteWord	0x4A	0x4AF8

Figure 11. Unchecking "Validate Command Write"

– **Programming Option:**
  - Turn rails off before programming. Some devices require that the rails are turned off before programming. If the rails are selected, all rails on the target devices are turned off first by writing ON_OFF_CONFIG command (Code = 02h) with value = 00h (Operation Only), followed by writing OPERATION command (Code = 01h) with value = 01h (Immediate Off) to all rails.
  - Add IC_DEVICE_ID (0xAD) to script. If selected, IC_DEVICE_ID (0xAD) is read and added to the script. The program can verify if the IC_DEVICE_ID (0xAD) of the target device matches the script before continuing with programming.

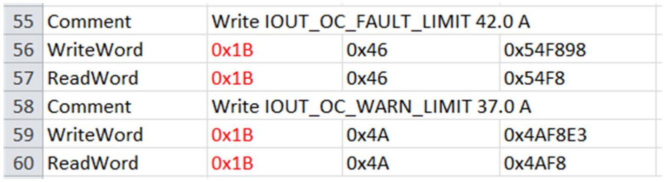
    For example, Figure 12 shows the IC_DEVICE_ID (0xAD) of the TPS544C25 device.



7	Comment	Verifying IC_DEVICE_ID is 0x2700 (TPS544C25)			
8	BlockRead	0xAD	0x022700		

Figure 12. Verifying IC_DEVICE_ID

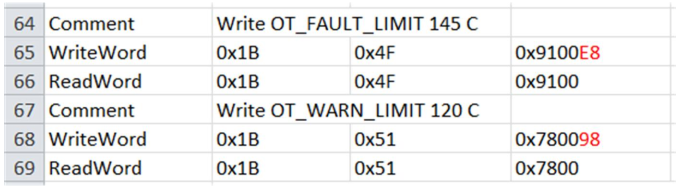
– **Other Options:**
  - **Including Device PMBus Address:** If this option is selected, the script includes the device address in the second column. The red 'w' text in Figure 13 is the device address.



55	Comment	Write IOUT_OC_FAULT_LIMIT 42.0 A		
56	WriteWord	0x1B	0x46	0x54F898
57	ReadWord	0x1B	0x46	0x54F8
58	Comment	Write IOUT_OC_WARN_LIMIT 37.0 A		
59	WriteWord	0x1B	0x4A	0x4AF8E3
60	ReadWord	0x1B	0x4A	0x4AF8

Figure 13. Including Device PMBus Address

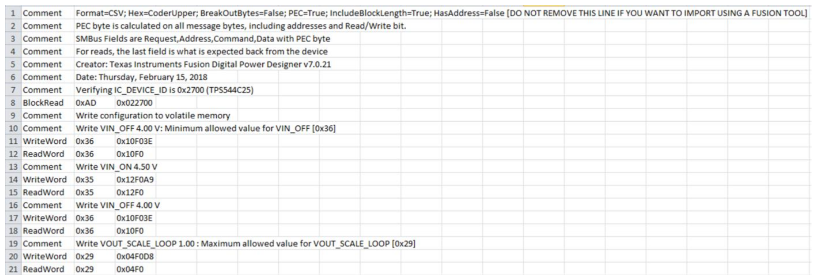
  - **Store Default Timing:** The delay time after the STORE_DEFAULT_ALL command is sent before continuing to the next step.
  - **Add PEC Byte:** The PEC byte can be added of the code as Figure 14 shows.



64	Comment	Write OT_FAULT_LIMIT 145 C		
65	WriteWord	0x1B	0x4F	0x9100E8
66	ReadWord	0x1B	0x4F	0x9100
67	Comment	Write OT_WARN_LIMIT 120 C		
68	WriteWord	0x1B	0x51	0x780098
69	ReadWord	0x1B	0x51	0x7800

Figure 14. Adding PEC Byte

Figure 15 shows an example of a saved script file. Rows 1-6 show important information about script files such as formats, options, PEC, version, and time.



1	Comment	Format=CSV; Hex=CoderUpper; BreakOutBytes=False; PEC=True; IncludeBlockLength=True; HasAddress=False [DO NOT REMOVE THIS LINE IF YOU WANT TO IMPORT USING A FUSION TOOL]	
2	Comment	PEC byte is calculated on all message bytes, including addresses and Read/Write bit.	
3	Comment	SMBus Fields are Request,Address,Command,Data with PEC byte	
4	Comment	For reads, the last field is what is expected back from the device	
5	Comment	Creator: Texas Instruments Fusion Digital Power Designer v7.0.21	
6	Comment	Date: Thursday, February 15, 2018	
7	Comment	Verifying IC_DEVICE_ID is 0x2700 (TPS544C25)	
8	BlockRead	0xAD	0x022700
9	Comment	Write configuration to volatile memory	
10	Comment	Write VIN_OFF 4.00 V: Minimum allowed value for VIN_OFF [0x36]	
11	WriteWord	0x36	0x10F03E
12	ReadWord	0x36	0x10F0
13	Comment	Write VIN_ON 4.50 V	
14	WriteWord	0x35	0x12F0A9
15	ReadWord	0x35	0x12F0
16	Comment	Write VIN_OFF 4.00 V	
17	WriteWord	0x36	0x10F03E
18	ReadWord	0x36	0x10F0
19	Comment	Write VOUT_SCALE_LOOP 1.00 : Maximum allowed value for VOUT_SCALE_LOOP [0x29]	
20	WriteWord	0x29	0x04F0D8
21	ReadWord	0x29	0x04F0

(1) Row 1 shows the format and options you have when the script file is saved.

(2) Row 2 describes the PEC byte calculation.

(3) Row 3 describes the SMBus Fields.

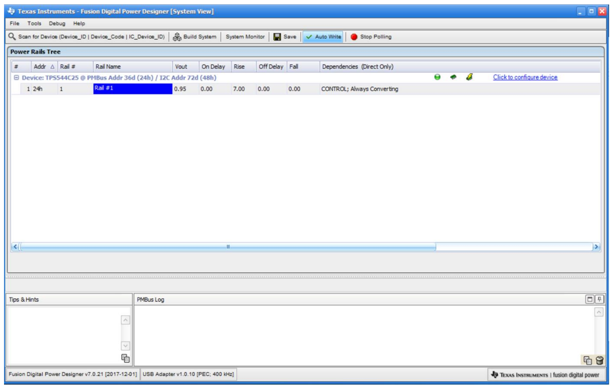
(4) Row 4 is for reads.

(5) Row 5 shows the software and version used to create the script file.

(6) Row 6 records the time of when the script file is created.

Figure 15. Script File

The configuration can be saved as an individual device configuration in a project file (.xml), but also can be saved as a system file (.tifsp) by clicking “File -> Save System File” in the system window (shown in Figure 16).


Texas Instruments - Fusion Digital Power Designer [System View]
File
Tools
Debug
Help
Scan for Device (Device_ID | Device_Code | IC_Device_ID)
Build System
System Monitor
Save
Auto Write
Stop Polling
Power Rails Tree
Tips & Hints
PMBus Log

#	Addr	Rail #	Rail Name	Vout	On Delay	Rise	Off Delay	Fall	Dependencies (Direct Only)
Device: TPS544C25 @ PMBus Addr 36d (24h) / I2C Addr 72d (48h)									Click to configure device
1	24h	1	Rail #1	0.95	0.00	7.00	0.00	0.00	CONTROL; Always Converting

Fusion Digital Power Designer v7.0.21 [2017-12-01] | USB Adapter v1.0.10 [PEC; 400 kHz] | Texas Instruments | fusion digital power

Figure 16. Export System File in System Window

There are some differences between system files and project files. For example, only a system file can be used with TI digital power manufacturing tools.

System file (.tifsp): stores multiple device configurations in a single file. System files can be saved in the system configuration window by using the Save button or File menu.

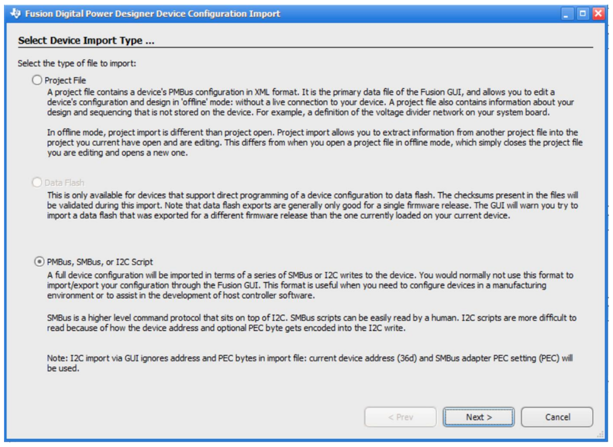
Project file (.xml): stores a single configuration definition of a device and non-device, such as rail names. Project files can be saved in the configuration window of the device by clicking File > Save Project As.

3.1.3 Importing A File

The system file (.tifsp) or the configuration file of the device saved in the previous step can be imported to the device.

To import the project file or script file, follow these steps:

1. Click File, then click Import to Device. Choose the preferred file type (as shown in Figure 17) and click the Next button to continue.



Fusion Digital Power Designer Device Configuration Import

Select Device Import Type ...

Select the type of file to import:

( ) Project File

A project file contains a device's PMBus configuration in XML format. It is the primary data file of the Fusion GUI, and allows you to edit a device's configuration and design in 'offline' mode: without a live connection to your device. A project file also contains information about your design and sequencing that is not stored on the device. For example, a definition of the voltage divider network on your system board.

In offline mode, project import is different than project open. Project import allows you to extract information from another project file into the project you current have open and are editing. This differs from when you open a project file in offline mode, which simply closes the project file you are editing and opens a new one.

( ) Data Flash

This is only available for devices that support direct programming of a device configuration to data flash. The checksums present in the files will be validated during this import. Note that data flash exports are generally only good for a single firmware release. The GUI will warn you try to import a data flash that was exported for a different firmware release than the one currently loaded on your current device.

(●) PMBus, SMBus, or I2C Script

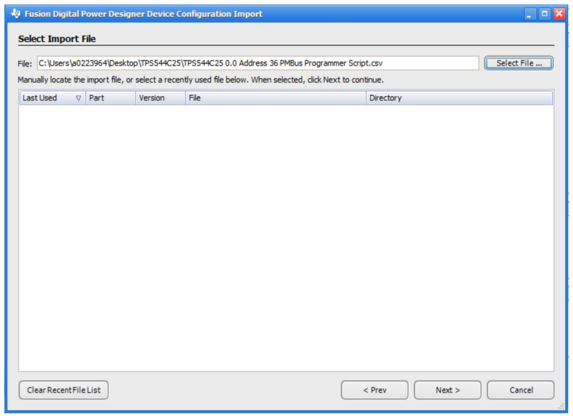
A full device configuration will be imported in terms of a series of SMBus or I2C writes to the device. You would normally not use this format to import/export your configuration through the Fusion GUI. This format is useful when you need to configure devices in a manufacturing environment or to assist in the development of host controller software.

SMBus is a higher level command protocol that sits on top of I2C. SMBus scripts can be easily read by a human. I2C scripts are more difficult to read because of how the device address and optional PEC byte gets encoded into the I2C write.

Note: I2C import via GUI ignores address and PEC bytes in import file: current device address (36d) and SMBus adapter PEC setting (PEC) will be used.

Figure 17. Selecting Device Import Type

2. Click Select File, and select the file you want to import. Click Next to continue (as shown in Figure 18).



Fusion Digital Power Designer Device Configuration Import

Select Import File

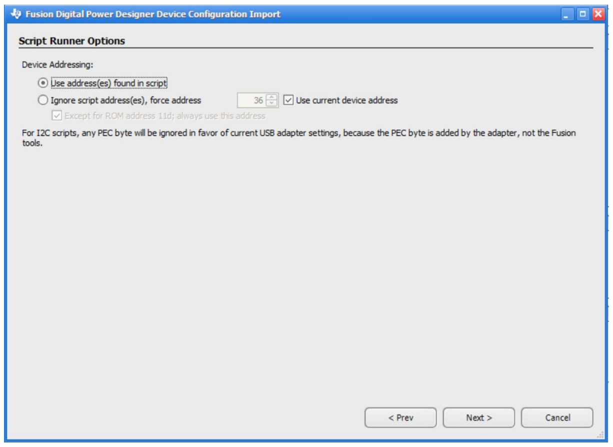
File: C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 PMBus Programmer Script.csv

Manually locate the import file, or select a recently used file below. When selected, click Next to continue.

Last Used	Part	Version	File	Directory

Figure 18. Selecting Import File

3. Set the address by clicking Use Address(es) found in script, and click the Next button (as shown in Figure 19).



Fusion Digital Power Designer Device Configuration Import

Script Runner Options

Device Addressing:

(●) Use address(es) found in script

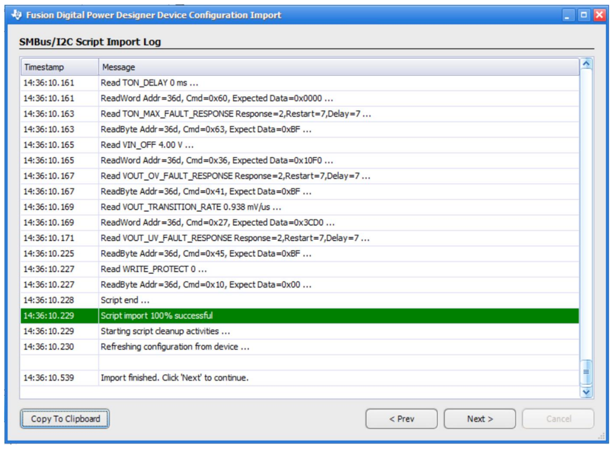
( ) Ignore script address(es), force address 36 [✓] Use current device address

[✓] Except for ROM address 11d; always use this address

For I2C scripts, any PEC byte will be ignored in favor of current USB adapter settings, because the PEC byte is added by the adapter, not the Fusion tools.

Figure 19. Script Runner Options

4. The Fusion GUI then begins to import the script. After the import is completed, the text **Script import 100% successful** appears and is highlighted in green (as shown in Figure 20)

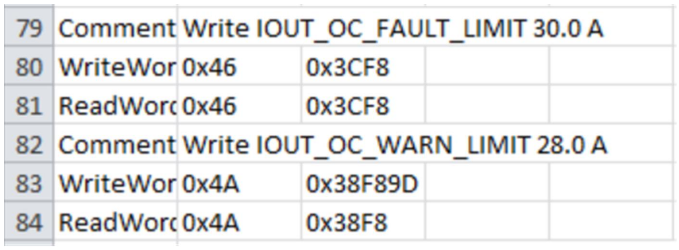


SMBus/I2C Script Import Log

Timestamp	Message
14:36:10.161	Read TON_DELAY 0 ms ...
14:36:10.161	ReadWord Addr=36d, Cmd=0x60, Expected Data=0x0000 ...
14:36:10.163	Read TON_MAX_FAULT_RESPONSE Response=2,Restart=7,Delay=7 ...
14:36:10.163	ReadByte Addr=36d, Cmd=0x63, Expect Data=0xBF ...
14:36:10.165	Read VIN_OFF 4.00 V ...
14:36:10.165	ReadWord Addr=36d, Cmd=0x36, Expected Data=0x10F0 ...
14:36:10.167	Read VOUT_OV_FAULT_RESPONSE Response=2,Restart=7,Delay=7 ...
14:36:10.167	ReadByte Addr=36d, Cmd=0x41, Expect Data=0xBF ...
14:36:10.169	Read VOUT_TRANSITION_RATE 0.938 mV/us ...
14:36:10.169	ReadWord Addr=36d, Cmd=0x27, Expected Data=0x3CD0 ...
14:36:10.171	Read VOUT_UV_FAULT_RESPONSE Response=2,Restart=7,Delay=7 ...
14:36:10.225	ReadByte Addr=36d, Cmd=0x45, Expect Data=0xBF ...
14:36:10.227	Read WRITE_PROTECT 0 ...
14:36:10.227	ReadByte Addr=36d, Cmd=0x10, Expect Data=0x00 ...
14:36:10.228	Script end ...
14:36:10.229	Script import 100% successful
14:36:10.229	Starting script cleanup activities ...
14:36:10.230	Refreshing configuration from device ...
14:36:10.539	Import finished. Click 'Next' to continue.

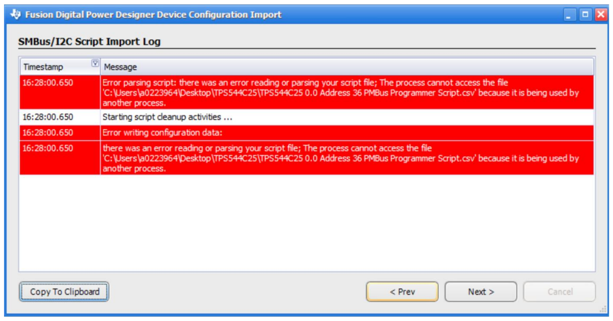
Figure 20. SMBus/I2C Script Import Log

NOTE: When there are errors in the script file during an import, the errors are indicated by the text **0x3CF833** (as shown in Figure 21), followed by an error parsing script message (as shown in Figure 22), indicating that there are errors in the script during a file import.



79	Comment	Write IOUT_OC_FAULT_LIMIT 30.0 A			
80	WriteWor	0x46	0x3CF8		
81	ReadWor	0x46	0x3CF8		
82	Comment	Write IOUT_OC_WARN_LIMIT 28.0 A			
83	WriteWor	0x4A	0x38F89D		
84	ReadWor	0x4A	0x38F8		

Figure 21. Script File With An Error

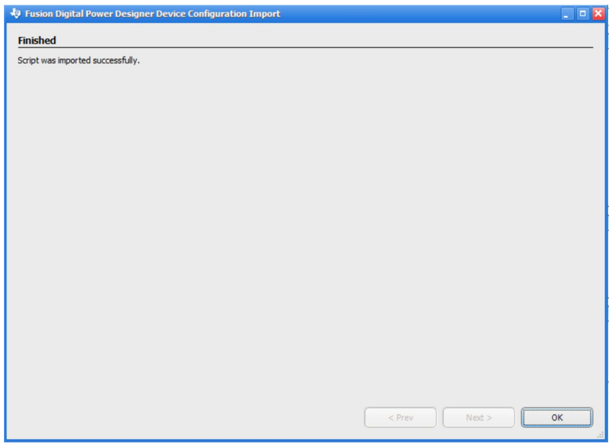


SMBus/I2C Script Import Log

Timestamp	Message
16:28:00.650	Error parsing script: there was an error reading or parsing your script file; The process cannot access the file 'C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 PMBus Programmer Script.csv' because it is being used by another process.
16:28:00.650	Starting script cleanup activities ...
16:28:00.650	Error writing configuration data:
16:28:00.650	there was an error reading or parsing your script file; The process cannot access the file 'C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 PMBus Programmer Script.csv' because it is being used by another process.

Figure 22. Reporting The Error

5. Click the OK button to complete the importing (shown in Figure 23).



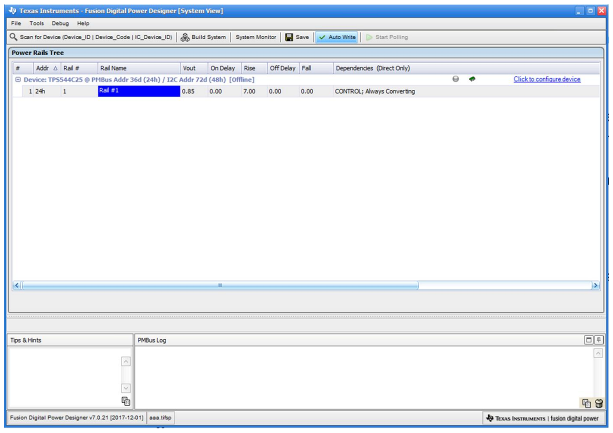
Finished

Script was imported successfully.

Figure 23. Finishing Importing A Script File

A system file can be imported to a device in the system window.

1. Click File then click Import System File to select the file you want to import (as shown in Figure 24).

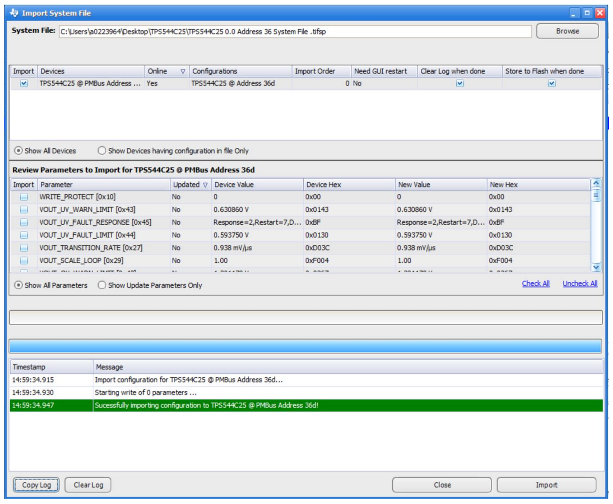

Texas Instruments - Fusion Digital Power Designer [System View]
File
Tools
Debug
Help
Scan for Device (Device_ID | Device_Code | IC_Device_ID)
Build System
System Monitor
Save
Auto Write
Start Polling
Power Rails Tree
Tips & Hints
PMBus Log
Texas Instruments | fusion digital power

#	Addr	Rail #	Rail Name	Vout	On Delay	Rise	Off Delay	Fall	Dependencies (Direct Only)
Device: TPS544C25 @ PMBus Addr 36d (24h) / I2C Addr 72d (48h) [Offline]									Click to configure device
1	24h	1	Rail #1	0.85	0.00	7.00	0.00	0.00	CONTROL; Always Converting

Fusion Digital Power Designer v7.0.21 [2017-12-01] | aaa.tifsp

Figure 24. System Window

2. The import dialog box appears after you select a file to import. You can review the parameters using the import dialog box by clicking on Show Update Parameters Only. Once you are done reviewing the updated parameters, click the Import button to import the system file to the device (as shown in Figure 25).



System File: C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 System File .tifsp

Import	Devices	Online	Configurations	Import Order	Need GUI restart	Clear Log when done	Store to Flash when done
✓	TPS544C25 @ PMBus Address ...	Yes	TPS544C25 @ Address 36d	0	No	✓	✓

Review Parameters to Import for TPS544C25 @ PMBus Address 36d

Import	Parameter	Updated	Device Value	Device Hex	New Value	New Hex
☐	WRITE_PROTECT [0x10]	No	0	0x00	0	0x00
☐	VOUT_UV_WARN_LIMIT [0x43]	No	0.630860 V	0x0143	0.630860 V	0x0143
☐	VOUT_UV_FAULT_RESPONSE [0x45]	No	Response=2,Restart=7,D...	0xBF	Response=2,Restart=7,D...	0xBF
☐	VOUT_UV_FAULT_LIMIT [0x44]	No	0.593750 V	0x0130	0.593750 V	0x0130
☐	VOUT_TRANSITION_RATE [0x27]	No	0.938 mV/µs	0xD03C	0.938 mV/µs	0xD03C
☐	VOUT_SCALE_LOOP [0x29]	No	1.00	0xF004	1.00	0xF004

Timestamp	Message
14:59:34.915	Import configuration for TPS544C25 @ PMBus Address 36d...
14:59:34.930	Starting write of 0 parameters ...
14:59:34.947	Sucessfully importing configuration to TPS544C25 @ PMBus Address 36d!

Figure 25. .Importing System File

3.2 Applications of Fusion Digital Power Manufacturing Tool

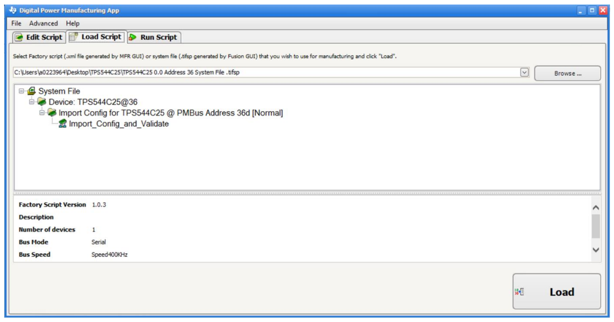
The Fusion Digital Power Manufacturing Tool (MFR GUI) is a Graphical User Interface used to download firmware, upgrade firmware, and to configure, calibrate, and test firmware. The MFR GUI supports the same devices as the Fusion Digital Power Designer software. This tool is typically used in manufacturing environments.

With the Fusion Manufacturing Tool, mass production is very convenient. After the device is connected to a PC, click the Scan button to find a device, then click the Start button so the system file can be loaded to the device.

3.2.1 Import System File

Once the manufacturing tool is launched, three tabs appear within the Digital Power Manufacturing App window. The tabs are titled Edit Script, Load Script, and Run Script (as shown in Figure 26). Proceed with the following steps:

1. Click on the Load Script tab, and under that tab, click on the Browse button to find a script to load.
2. Select a script.
3. Click the Load button once the script is displayed.



Select Factory script (.xml file generated by MFR GUI) or system file (.tifsp generated by Fusion GUI) that you wish to use for manufacturing and click "Load".

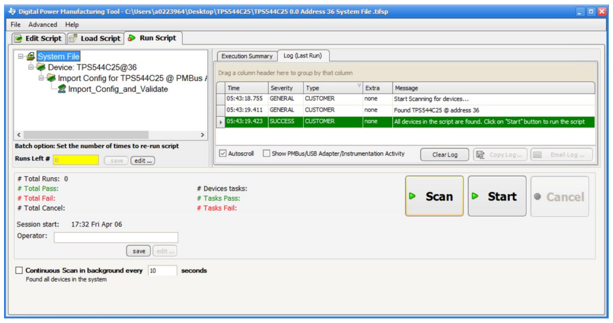
C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 System File .tifsp

```
System File
  Device: TPS544C25@36
    Import Config for TPS544C25 @ PMBus Address 36d [Normal]
      Import_Config_and_Validate
```

Factory Script Version	1.0.3
Description	
Number of devices	1
Bus Mode	Serial
Bus Speed	Speed400KHz

Figure 26. Loading Script

4. After the "Run Script" tab appears, click the Scan button to find the device (shown in Figure 27).


Digital Power Manufacturing Tool - C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 System File .tifsp

System File
Device: TPS544C25@36
Import Config for TPS544C25 @ PMBus
Import_Config_and_Validate

Batch option: Set the number of times to re-run script
Runs Left # 0

Drag a column header here to group by that column

Time	Severity	Type	Extra	Message
05:43:18.755	GENERAL	CUSTOMER	none	Start Scanning for devices...
05:43:19.411	GENERAL	CUSTOMER	none	Found TPS544C25 @ address 36
05:43:19.423	SUCCESS	CUSTOMER	none	All devices in the script are found. Click on "Start" button to run the script

# Total Runs: 0
# Total Pass:
# Total Fail:
# Total Cancel:

# Devices tasks:
# Tasks Pass:
# Tasks Fail:

Session start: 17:32 Fri Apr 06
Operator:

Continuous Scan in background every 10 seconds
Found all devices in the system

Figure 27. Running Script

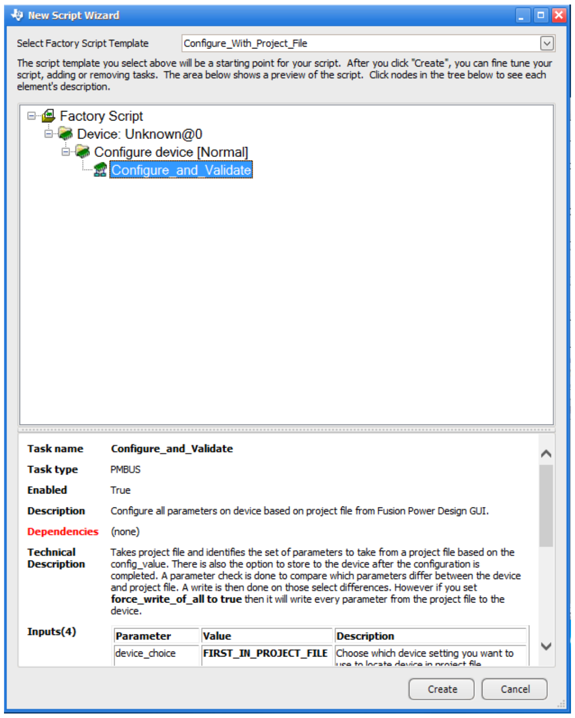
5. After the scan is finished, click the Start button to run the script.

3.2.2 Creating A Script

Sometimes, you just want to change a few parameters, so there is no need to export a system file from the Fusion Digital Power GUI and load the script to the device using the manufacturing tool. It is more convenient to create script directly using the manufacturing tool.

In the Edit Script window shown in Figure 28, click on File, then click on New Script Wizard. When the New Script Wizard dialog box appears, you have the option of selecting some factory script templates:

- **Basic:** Contains the device and three generally used activities (initialization, normal and shutdown)
- **Blank:** Empty Device
- **Configure_ With_Project_File:** Configures a device based on a project file
- **Firmware_Download_via_Rom:** Downloads firmware to a device in read-only memory mode and sends it to program mode. The activity type of downloading firmware to a blank device is the Initialization type.
- **Firmware_Update:** Updates firmware on a device in program mode. The activity type is Normal, since it is in program mode.


New Script Wizard

Select Factory Script Template: Configure_With_Project_File

The script template you select above will be a starting point for your script. After you click "Create", you can fine tune your script, adding or removing tasks. The area below shows a preview of the script. Click nodes in the tree below to see each element's description.

Factory Script
Device: Unknown@0
Configure device [Normal]
Configure_and_Validate

Task name	Configure_and_Validate
Task type	PMBUS
Enabled	True
Description	Configure all parameters on device based on project file from Fusion Power Design GUI.
Dependencies	(none)
Technical Description	Takes project file and identifies the set of parameters to take from a project file based on the config_value. There is also the option to store to the device after the configuration is completed. A parameter check is done to compare which parameters differ between the device and project file. A write is then done on those select differences. However if you set **force_write_of_all to true** then it will write every parameter from the project file to the device.

Inputs(4)

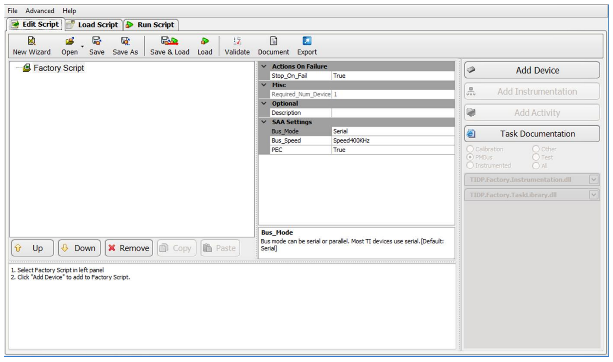
Parameter	Value	Description
device_choice	FIRST_IN_PROJECT_FILE	Choose which device setting you want to use to locate device in project file

Figure 28. New Script Wizard

You can click nodes in the tree to see each element description. After selecting the factory script template, click Create to finish script creation.

Alternatively, you can create the script directly from the Edit Script window instead of selecting the factory script template to Blank. For example, you can change the IOUT_OC_FAULT_LIMIT from 30 A to 20 A by following these seven steps:

1. After launching the manufacturing tool, the Load Script screen appears, as shown in Figure 29. To create a new script or edit an existing script, you must enable a special script editor mode. Click File in the top menu and then click Admin; enter the password “texas” A new Edit Script tab appears in the GUI.



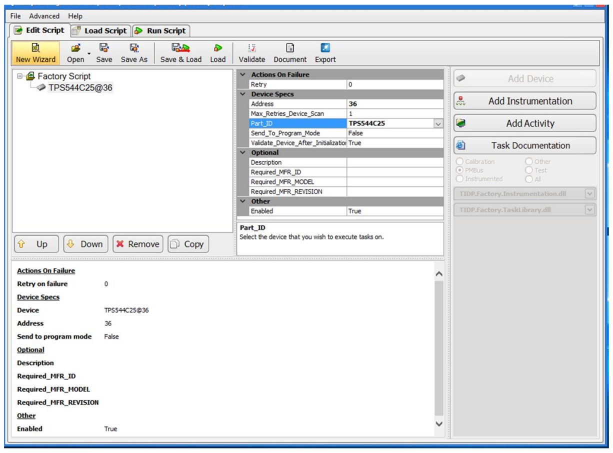
Actions On Failure	
Stop_On_Fail	True
Misc	
Required_Num_Device	1
Optional	
Description	
SAA Settings	
Bus_Mode	Serial
Bus_Speed	Speed400KHz
PEC	True

Bus_Mode
Bus mode can be serial or parallel. Most TI devices use serial.[Default: Serial]

1. Select Factory Script in left panel
2. Click "Add Device" to add to Factory Script.

Figure 29. Load Script Screen

2. Click Add Device and select Part_ID TPS544C25. Then write the device address 36d (shown in Figure 30).



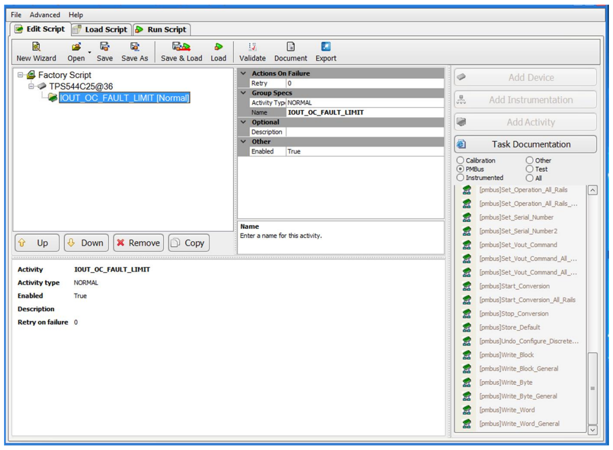
Actions On Failure	
Retry	0
Device Specs	
Address	36
Max_Retries_Device_Scan	1
Part_ID	TPS544C25
Send_To_Program_Mode	False
Validate_Device_After_Initializatio	True
Optional	
Description	
Required_MFR_ID	
Required_MFR_MODEL	
Required_MFR_REVISION	
Other	
Enabled	True

Part_ID
Select the device that you wish to execute tasks on.

Actions On Failure	
Retry on failure	0
Device Specs	
Device	TPS544C25@36
Address	36
Send to program mode	False
Optional	
Description	
Required_MFR_ID	
Required_MFR_MODEL	
Required_MFR_REVISION	
Other	
Enabled	True

Figure 30. Adding The Device

3. Click Add Activity and change the activity name to IOUT_OC_FAULT_LIMIT. Select activity type Normal (shown in Figure 31).



Factory Script
TPS544C25@36
IOUT_OC_FAULT_LIMIT [Normal]

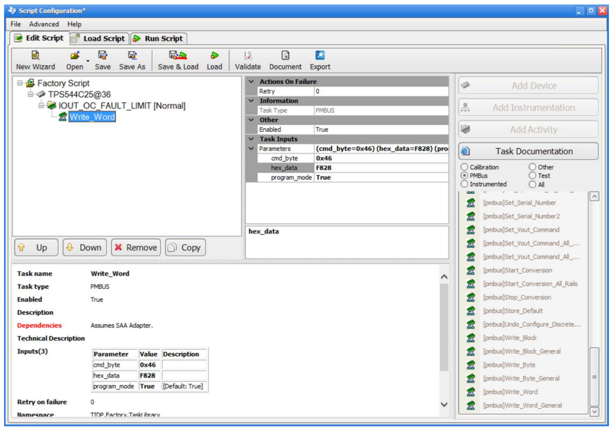
Actions On Failure	
Retry	0
Group Specs	
Activity Typ	NORMAL
Name	IOUT_OC_FAULT_LIMIT
Optional	
Description	
Other	
Enabled	True

Name
Enter a name for this activity.

Activity	IOUT_OC_FAULT_LIMIT
Activity type	NORMAL
Enabled	True
Description	
Retry on failure	0

Figure 31. Adding An Activity

4. To add a task to modify the value of IOUT_OC_FAULT_LIMIT, click Write Word and write cmd_byte=0x46, hex_data=0xF828 (shown in Figure 32).



Factory Script
TPS544C25@36
IOUT_OC_FAULT_LIMIT [Normal]
Write_Word

Actions On Failure	
Retry	0
Information	
Task Type	PMBUS
Other	
Enabled	True
Task Inputs	
Parameters	(cmd_byte=0x46) (hex_data=F828) (pro
cmd_byte	0x46
hex_data	F828
program_mode	True

hex_data

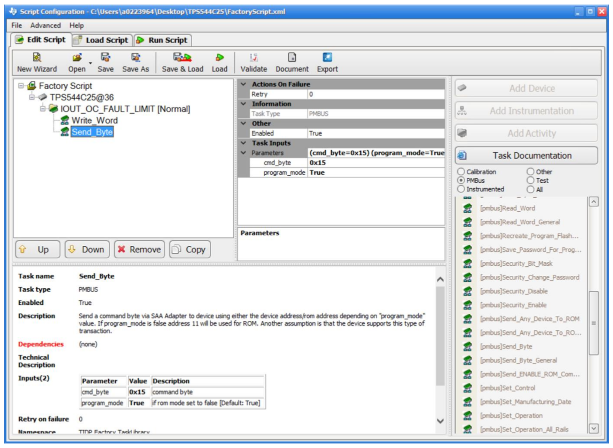
Task name Write_Word
Task type PMBUS
Enabled True
Description
Dependencies Assumes SAA Adapter.
Technical Description
Inputs(3)

Parameter	Value	Description
cmd_byte	0x46	
hex_data	F828	
program_mode	True	[Default: True]

Retry on failure 0
Namespace TIDP.Factory.TaskLibrary

Figure 32. Adding A Task to Modify The Value of IOUT_OC_FAULT_LIMIT

5. To add a task to store all data to a non-volatile User Store memory, click Send Byte and write cmd_byte=0x15 (shown in Figure 33).



Factory Script
TPS544C25@36
IOUT_OC_FAULT_LIMIT [Normal]
Write_Word
Send_Byte

Actions On Failure	
Retry	0
Information	
Task Type	PMBUS
Other	
Enabled	True
Task Inputs	
Parameters	(cmd_byte=0x15) (program_mode=True
cmd_byte	0x15
program_mode	True

Parameters

Task name Send_Byte
Task type PMBUS
Enabled True
Description Send a command byte via SAA Adapter to device using either the device address/rom address depending on "program_mode" value. If program_mode is false address 11 will be used for ROM. Another assumption is that the device supports this type of transaction.
Dependencies (none)
Technical Description
Inputs(2)

Parameter	Value	Description
cmd_byte	0x15	command byte
program_mode	True	if rom mode set to false [Default: True]

Retry on failure 0
Namespace TIDP.Factory.TaskLibrary

Figure 33. Adding A Task to Store All Data to NVM

6. Click Save and Load to save the file and load it to the device (shown in Figure 34).

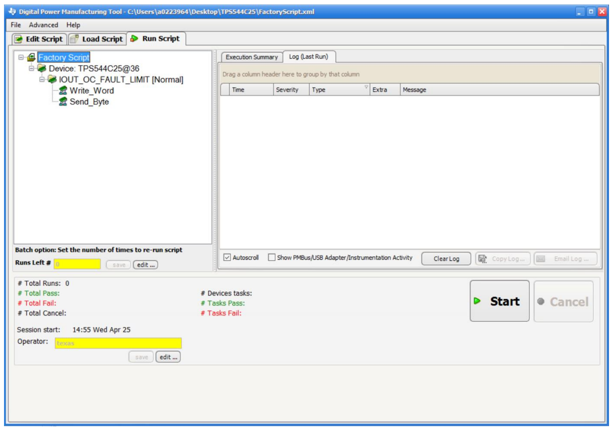
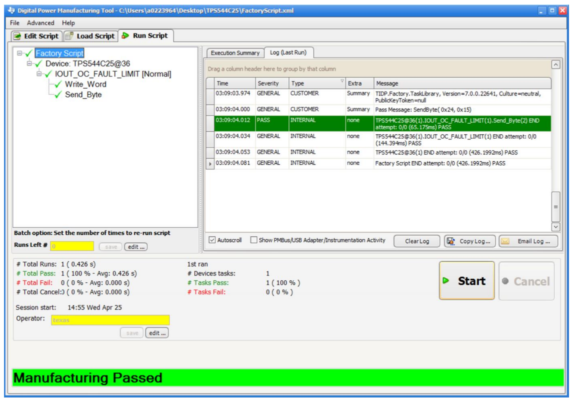

Digital Power Manufacturing Tool - C:\Users\a0223964\Desktop\TPS544C25\FactoryScript.xml
File Advanced Help
Edit Script
Load Script
Run Script
Factory Script
Device: TPS544C25@36
IOUT_OC_FAULT_LIMIT [Normal]
Write_Word
Send_Byte
Execution Summary
Log (Last Run)
Drag a column header here to group by that column
Time Severity Type Extra Message
Batch option: Set the number of times to re-run script
Runs Left #
save edit ...
Autoscroll
Show PMBus/USB Adapter/Instrumentation Activity
Clear Log
Copy Log ...
Email Log ...
# Total Runs: 0
# Total Pass:
# Total Fail:
# Total Cancel:
# Devices tasks:
# Tasks Pass:
# Tasks Fail:
Session start: 14:55 Wed Apr 25
Operator: texas
Start
Cancel

Figure 34. Saving and Loading Script

7. When the Run Script tab appears, click the Start button to begin script execution (shown in Figure 35).


Digital Power Manufacturing Tool - C:\Users\a0223964\Desktop\TPS544C25\FactoryScript.xml
File Advanced Help
Edit Script
Load Script
Run Script
Factory Script
Device: TPS544C25@36
IOUT_OC_FAULT_LIMIT [Normal]
Write_Word
Send_Byte
Execution Summary
Log (Last Run)
Drag a column header here to group by that column
Autoscroll
Show PMBus/USB Adapter/Instrumentation Activity
Clear Log
Copy Log ...
Email Log ...
Batch option: Set the number of times to re-run script
Runs Left #
save edit ...
Start
Cancel

Time	Severity	Type	Extra	Message
03:09:03.974	GENERAL	CUSTOMER	Summary	TIDP.Factory.TaskLibrary, Version=7.0.0.22641, Culture=neutral, PublicKeyToken=null
03:09:04.000	GENERAL	CUSTOMER	Summary	Pass Message: SendByte(0x24, 0x15)
03:09:04.012	PASS	INTERNAL	none	TPS544C25@36(1).IOUT_OC_FAULT_LIMIT(1).Send_Byte(2) END attempt: 0/0 (65.175ms) PASS
03:09:04.034	GENERAL	INTERNAL	none	TPS544C25@36(1).IOUT_OC_FAULT_LIMIT(1) END attempt: 0/0 (144.394ms) PASS
03:09:04.053	GENERAL	INTERNAL	none	TPS544C25@36(1) END attempt: 0/0 (426.1992ms) PASS
03:09:04.081	GENERAL	INTERNAL	none	Factory Script END attempt: 0/0 (426.1992ms) PASS

```
# Total Runs:  1 ( 0.426 s)                  1st ran
# Total Pass:  1 ( 100 % - Avg: 0.426 s)     # Devices tasks:    1
# Total Fail:  0 ( 0 % - Avg: 0.000 s)       # Tasks Pass:       1 ( 100 % )
# Total Cancel:0 ( 0 % - Avg: 0.000 s)       # Tasks Fail:       0 ( 0 % )

Session start:   14:55 Wed Apr 25
Operator:        texas

Manufacturing Passed
```

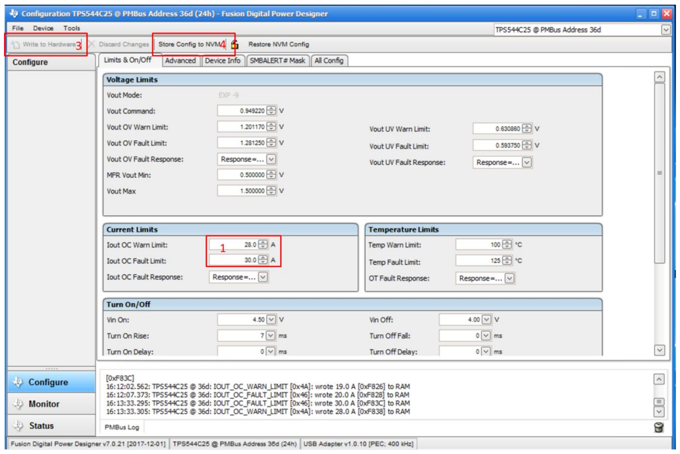
Figure 35. Running Script

4 Examples to Support Mass Production

The following examples show how to export a system file from the Fusion Digital Power GUI and load that file to the TPS544C25 using a manufacturing tool to support mass production. The purpose is to configure the IOUT_OC_WARN_LIMIT to 28 A and IOUT_OC_FAULT_LIMIT to 30 A.

4.1 Configuring The Device

1. After launching the Fusion Digital Power GUI, click the Configure button. Change the IOUT_OC_WARN_LIMIT to 28 A and the IOUT_OC_FAULT_LIMIT to 30 A, as shown in Figure 36.


Configuration TPS544C25 @ PMBus Address 36d (24h) - Fusion Digital Power Designer

TPS544C25 @ PMBus Address 36d

Voltage Limits
- Vout Mode: EXP -9
- Vout Command: 0.949220 V
- Vout OV Warn Limit: 1.201170 V
- Vout OV Fault Limit: 1.281250 V
- Vout OV Fault Response: Response=...
- MFR Vout Min: 0.500000 V
- Vout Max: 1.500000 V
- Vout UV Warn Limit: 0.630860 V
- Vout UV Fault Limit: 0.593750 V
- Vout UV Fault Response: Response=...

Current Limits
- Iout OC Warn Limit: 28.0 A
- Iout OC Fault Limit: 30.0 A
- Iout OC Fault Response: Response=...

Temperature Limits
- Temp Warn Limit: 100 °C
- Temp Fault Limit: 125 °C
- OT Fault Response: Response=...

Turn On/Off
- Vin On: 4.50 V
- Turn On Rise: 7 ms
- Turn On Delay: 0 ms
- Vin Off: 4.00 V
- Turn Off Fall: 0 ms
- Turn Off Delay: 0 ms

```
[0xF83C]
16:12:02.562: TPS544C25 @ 36d: IOUT_OC_WARN_LIMIT [0x4A]: wrote 19.0 A [0xF826] to RAM
16:12:07.373: TPS544C25 @ 36d: IOUT_OC_FAULT_LIMIT [0x46]: wrote 20.0 A [0xF828] to RAM
16:13:33.295: TPS544C25 @ 36d: IOUT_OC_FAULT_LIMIT [0x46]: wrote 30.0 A [0xF83C] to RAM
16:13:33.305: TPS544C25 @ 36d: IOUT_OC_WARN_LIMIT [0x4A]: wrote 28.0 A [0xF838] to RAM
```

Fusion Digital Power Designer v7.0.21 [2017-12-01] | TPS544C25 @ PMBus Address 36d (24h) | USB Adapter v1.0.10 [PEC; 400 kHz]

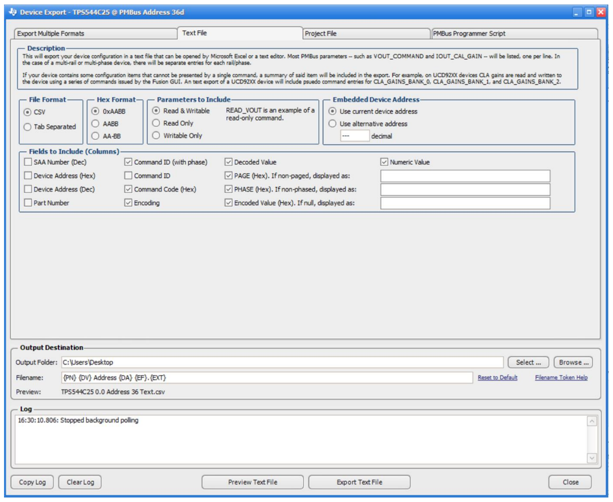
Figure 36. Configuring The Device

2. Click Write to Hardware.
3. Click Write Store Configuration to NVM.

4.2 Exporting A File

4.2.1 Exporting A Text File

1. Click File > Export > Text File and choose the file format (shown in Figure 37).


Device Export - TPS544C25 @ PMBus Address 36d

Description

This will export your device configuration in a text file that can be opened by Microsoft Excel or a text editor. Most PMBus parameters -- such as VOUT_COMMAND and IOUT_CAL_GAIN -- will be listed, one per line. In the case of a multi-rail or multi-phase device, there will be separate entries for each rail/phase.

If your device contains some configuration items that cannot be presented by a single command, a summary of said item will be included in the export. For example, on UCD92XX devices CLA gains are read and written to the device using a series of commands issued by the Fusion GUI. An text export of a UCD92XX device will include psuedo command entries for CLA_GAINS_BANK_0, CLA_GAINS_BANK_1, and CLA_GAINS_BANK_2.

File Format: [✓ CSV] Tab Separated

Hex Format: [✓ 0xAABB] AABB AA-BB

Parameters to Include: [✓ Read & Writable] Read Only Writable Only — READ_VOUT is an example of a read-only command.

Embedded Device Address: [✓ Use current device address] Use alternative address --- decimal

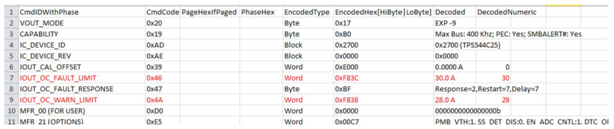
Fields to Include (Columns):
- ☐ SAA Number (Dec)
- ☐ Device Address (Hex)
- ☐ Device Address (Dec)
- ☐ Part Number
- ✓ Command ID (with phase)
- ☐ Command ID
- ✓ Command Code (Hex)
- ✓ Encoding
- ✓ Decoded Value
- ✓ PAGE (Hex). If non-paged, displayed as:
- ✓ PHASE (Hex). If non-phased, displayed as:
- ✓ Encoded Value (Hex). If null, displayed as:
- ✓ Numeric Value

Output Destination
- Output Folder: C:\Users\Desktop
- Filename: {PN} {DV} Address {DA} {EF}.{EXT}
- Preview: TPS544C25 0.0 Address 36 Text.csv

Log
```
16:30:10.806: Stopped background polling
```

Figure 37. Exporting A Text File

2. 2. Click Export Text File to create a file named TPS544C25 0.0 Address 36 Text.csv (shown in Figure 38).

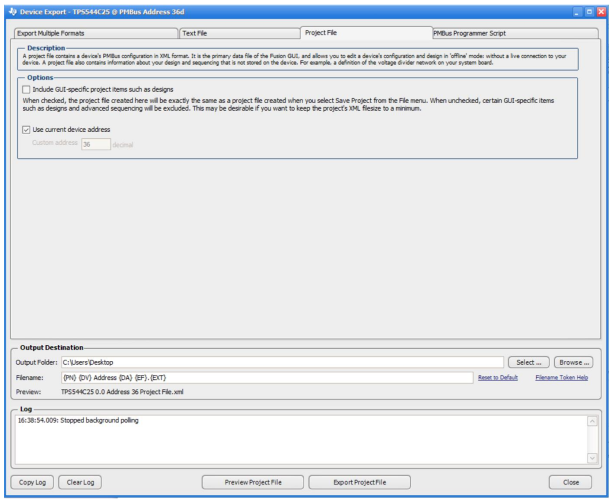


	CmdIDWithPhase	CmdCode	PageHexIfPaged	PhaseHex	EncodedType	EncodedHex[HiByte|LoByte]	Decoded	DecodedNumeric
2	VOUT_MODE	0x20			Byte	0x17	EXP -9	
3	CAPABILITY	0x19			Byte	0xB0	Max Bus: 400 Khz; PEC: Yes; SMBALERT#: Yes	
4	IC_DEVICE_ID	0xAD			Block	0x2700	0x2700 (TPS544C25)	
5	IC_DEVICE_REV	0xAE			Block	0x0000	0x0000	
6	IOUT_CAL_OFFSET	0x39			Word	0xE000	0.0000 A	0
7	IOUT_OC_FAULT_LIMIT	0x46			Word	0xF83C	30.0 A	30
8	IOUT_OC_FAULT_RESPONSE	0x47			Byte	0xBF	Response=2,Restart=7,Delay=7	
9	IOUT_OC_WARN_LIMIT	0x4A			Word	0xF838	28.0 A	28
10	MFR_00 (FOR USER)	0xD0			Word	0x0000	0000000000000000b	
11	MFR_21 (OPTIONS)	0xE5			Word	0x00C7	PMB_VTH:1, SS_DET_DIS:0, EN_ADC_CNTL:1, DTC_OI	

Figure 38. Text file

4.2.2 Exporting A Project File

1. Click File > Export > Project File and choose the file format (shown in Figure 39).



Device Export - TPS544C25 @ PMBus Address 36d

Description

A project file contains a device's PMBus configuration in XML format. It is the primary data file of the Fusion GUI, and allows you to edit a device's configuration and design in 'offline' mode: without a live connection to your device. A project file also contains information about your design and sequencing that is not stored on the device. For example, a definition of the voltage divider network on your system board.

Options

☐ Include GUI-specific project items such as designs

When checked, the project file created here will be exactly the same as a project file created when you select Save Project from the File menu. When unchecked, certain GUI-specific items such as designs and advanced sequencing will be excluded. This may be desirable if you want to keep the project's XML filesize to a minimum.

☑ Use current device address

Custom address: 36 decimal

Output Destination

Output Folder: C:\Users\Desktop

Filename: {PN} {DV} Address {DA} {EF}.{EXT}

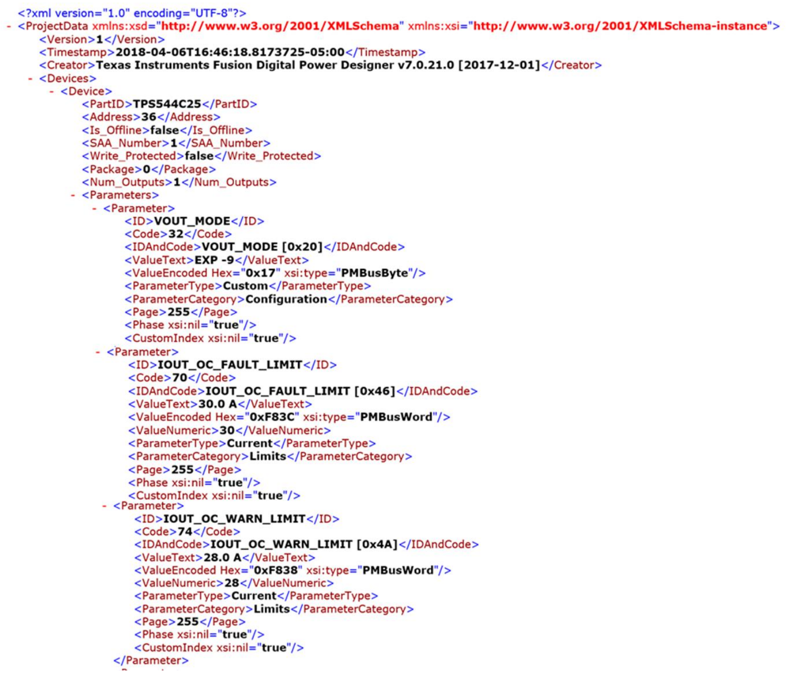
Preview: TPS544C25 0.0 Address 36 Project File.xml

Log

```
16:38:54.009: Stopped background polling
```

Figure 39. Exporting A Project File

2. Click Export Project File to create a file named TPS544C25 0.0 Address 36 Project File.xml (shown in Figure 40).

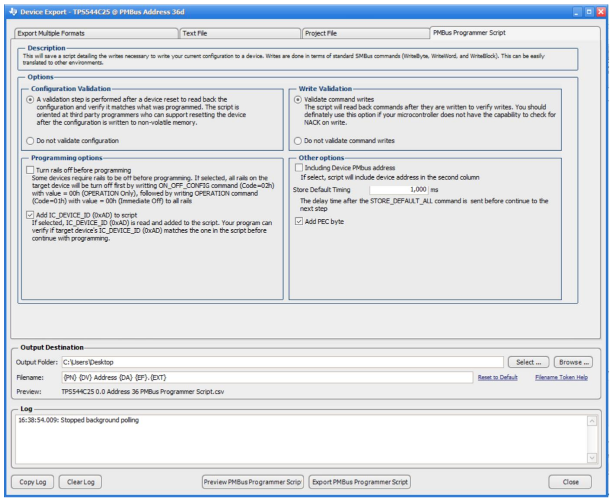


```
  <?xml version="1.0" encoding="UTF-8"?>
- <ProjectData xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <Version>1</Version>
     <Timestamp>2018-04-06T16:46:18.8173725-05:00</Timestamp>
     <Creator>Texas Instruments Fusion Digital Power Designer v7.0.21.0 [2017-12-01]</Creator>
   - <Devices>
      - <Device>
           <PartID>TPS544C25</PartID>
           <Address>36</Address>
           <Is_Offline>false</Is_Offline>
           <SAA_Number>1</SAA_Number>
           <Write_Protected>false</Write_Protected>
           <Package>0</Package>
           <Num_Outputs>1</Num_Outputs>
         - <Parameters>
            - <Parameter>
                 <ID>VOUT_MODE</ID>
                 <Code>32</Code>
                 <IDAndCode>VOUT_MODE [0x20]</IDAndCode>
                 <ValueText>EXP -9</ValueText>
                 <ValueEncoded Hex="0x17" xsi:type="PMBusByte"/>
                 <ParameterType>Custom</ParameterType>
                 <ParameterCategory>Configuration</ParameterCategory>
                 <Page>255</Page>
                 <Phase xsi:nil="true"/>
                 <CustomIndex xsi:nil="true"/>
            - <Parameter>
                 <ID>IOUT_OC_FAULT_LIMIT</ID>
                 <Code>70</Code>
                 <IDAndCode>IOUT_OC_FAULT_LIMIT [0x46]</IDAndCode>
                 <ValueText>30.0 A</ValueText>
                 <ValueEncoded Hex="0xF83C" xsi:type="PMBusWord"/>
                 <ValueNumeric>30</ValueNumeric>
                 <ParameterType>Current</ParameterType>
                 <ParameterCategory>Limits</ParameterCategory>
                 <Page>255</Page>
                 <Phase xsi:nil="true"/>
                 <CustomIndex xsi:nil="true"/>
            - <Parameter>
                 <ID>IOUT_OC_WARN_LIMIT</ID>
                 <Code>74</Code>
                 <IDAndCode>IOUT_OC_WARN_LIMIT [0x4A]</IDAndCode>
                 <ValueText>28.0 A</ValueText>
                 <ValueEncoded Hex="0xF838" xsi:type="PMBusWord"/>
                 <ValueNumeric>28</ValueNumeric>
                 <ParameterType>Current</ParameterType>
                 <ParameterCategory>Limits</ParameterCategory>
                 <Page>255</Page>
                 <Phase xsi:nil="true"/>
                 <CustomIndex xsi:nil="true"/>
              </Parameter>
```

Figure 40. Project File

4.2.3 Exporting A Script File

1. Click File > Export > PMBus Programmer Script and choose the file format (shown in Figure 41).



Device Export - TPS544C25 @ PMBus Address 36d

Export Multiple Formats | Text File | Project File | PMBus Programmer Script

Description
This will save a script detailing the writes necessary to write your current configuration to a device. Writes are done in terms of standard SMBus commands (WriteByte, WriteWord, and WriteBlock). This can be easily translated to other environments.

Options

Configuration Validation
[✓ A validation step is performed after a device reset to read back the configuration and verify it matches what was programmed. The script is oriented at third party programmers who can support resetting the device after the configuration is written to non-volatile memory.]
Do not validate configuration

Write Validation
[✓ Validate command writes]
The script will read back commands after they are written to verify writes. You should definately use this option if your microcontroller does not have the capability to check for NACK on write.
Do not validate command writes

Programming options
☐ Turn rails off before programming
Some devices require rails to be off before programming. If selected, all rails on the target device will be turn off first by writting ON_OFF_CONFIG command (Code=02h) with value = 00h (OPERATION Only), followed by writing OPERATION command (Code=01h) with value = 00h (Immediate Off) to all rails
✓ Add IC_DEVICE_ID (0xAD) to script
If selected, IC_DEVICE_ID (0xAD) is read and added to the script. Your program can verify if target device's IC_DEVICE_ID (0xAD) matches the one in the script before continue with programming.

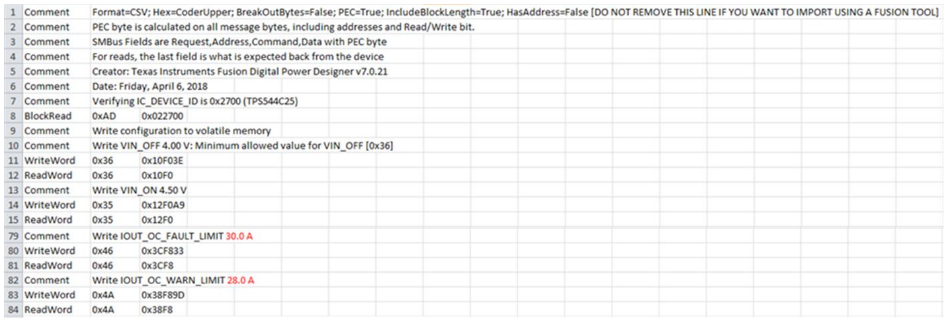
Other options
☐ Including Device PMbus address
If select, script will include device address in the second column
Store Default Timing 1,000 ms
The delay time after the STORE_DEFAULT_ALL command is sent before continue to the next step
✓ Add PEC byte

Output Destination
Output Folder: C:\Users\Desktop
Filename: {PN} {DV} Address {DA} {EF}.{EXT}
Preview: TPS544C25 0.0 Address 36 PMBus Programmer Script.csv

Log
```
16:38:54.009: Stopped background polling
```

Figure 41. Exporting A Script File

2. Click Export PMBus Programmer Script to create a script file named TPS544C25 0.0 Address 36 PMBus Programmer Script.csv (shown in Figure 42).

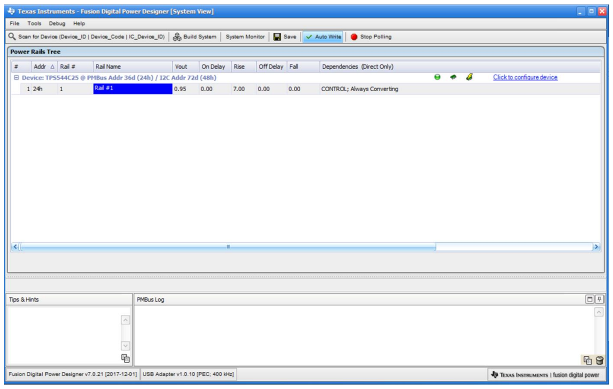


1	Comment	Format=CSV; Hex=CoderUpper; BreakOutBytes=False; PEC=True; IncludeBlockLength=True; HasAddress=False [DO NOT REMOVE THIS LINE IF YOU WANT TO IMPORT USING A FUSION TOOL]	
2	Comment	PEC byte is calculated on all message bytes, including addresses and Read/Write bit.	
3	Comment	SMBus Fields are Request,Address,Command,Data with PEC byte	
4	Comment	For reads, the last field is what is expected back from the device	
5	Comment	Creator: Texas Instruments Fusion Digital Power Designer v7.0.21	
6	Comment	Date: Friday, April 6, 2018	
7	Comment	Verifying IC_DEVICE_ID is 0x2700 (TPS544C25)	
8	BlockRead	0xAD	0x022700
9	Comment	Write configuration to volatile memory	
10	Comment	Write VIN_OFF 4.00 V: Minimum allowed value for VIN_OFF [0x36]	
11	WriteWord	0x36	0x10F03E
12	ReadWord	0x36	0x10F0
13	Comment	Write VIN_ON 4.50 V	
14	WriteWord	0x35	0x12F0A9
15	ReadWord	0x35	0x12F0
79	Comment	Write IOUT_OC_FAULT_LIMIT 30.0 A	
80	WriteWord	0x46	0x3CF833
81	ReadWord	0x46	0x3CF8
82	Comment	Write IOUT_OC_WARN_LIMIT 28.0 A	
83	WriteWord	0x4A	0x38F89D
84	ReadWord	0x4A	0x38F8

Figure 42. Script File

4.2.4 Exporting A System File

In the system window shown in Figure 43, click File > Save System File.



Texas Instruments - Fusion Digital Power Designer [System View]
File Tools Debug Help
Scan for Device (Device_ID | Device_Code | IC_Device_ID)
Build System
System Monitor
Save
Auto Write
Stop Polling
Power Rails Tree
Click to configure device
Tips & Hints
PMBus Log
Texas Instruments | fusion digital power

Device: TPS544C25 @ PMBus Addr 36d (24h) / I2C Addr 72d (48h)

#	Addr	Rail #	Rail Name	Vout	On Delay	Rise	Off Delay	Fall	Dependencies (Direct Only)
1	24h	1	Rail #1	0.95	0.00	7.00	0.00	0.00	CONTROL; Always Converting

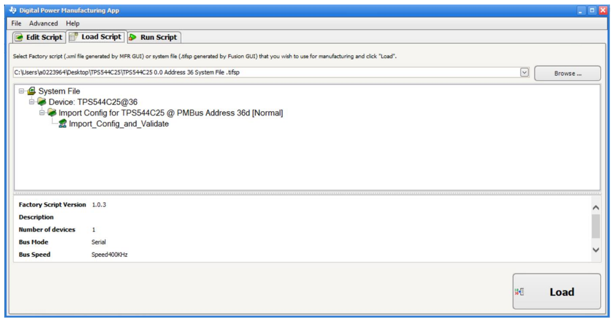
Fusion Digital Power Designer v7.0.21 [2017-12-01] | USB Adapter v1.0.10 [PEC; 400 kHz]

Figure 43. Exporting A System File In System Window

4.3 Loading A System File

After launching the manufacturing tool, the Edit Script tab appears, as shown in Figure 44.

1. Click Load Script. The Load Script tab appears.
2. Click Browse and select the system file TPS544C25 0.0 Address 36 System File.tifsp. After the script is displayed in the Load Script tab, click Load.



Select Factory script (.xml file generated by MFR GUI) or system file (.tifsp generated by Fusion GUI) that you wish to use for manufacturing and click "Load".

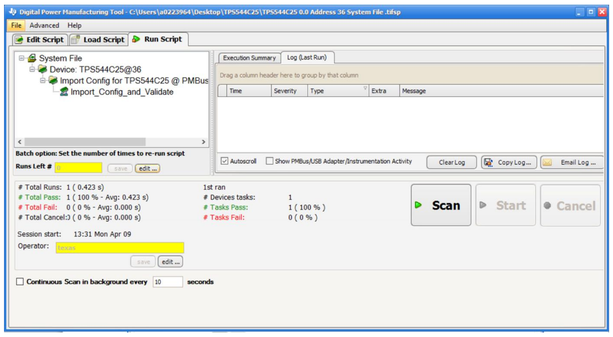
C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 System File .tifsp

```
System File
  Device: TPS544C25@36
    Import Config for TPS544C25 @ PMBus Address 36d [Normal]
      Import_Config_and_Validate
```

Factory Script Version	1.0.3
Description	
Number of devices	1
Bus Mode	Serial
Bus Speed	Speed400KHz

Figure 44. Loading A Script

3. After the Run Script tab appears, click the Scan button (shown in Figure 45).



```
System File
  Device: TPS544C25@36
    Import Config for TPS544C25 @ PMBus
      Import_Config_and_Validate
```

Drag a column header here to group by that column

Time	Severity	Type	Extra	Message

Autoscroll [✓] Show PMBus/USB Adapter/Instrumentation Activity [ ]

Batch option: Set the number of times to re-run script
Runs Left # 0

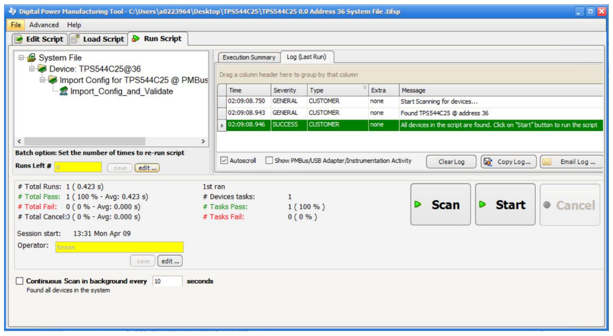
```
# Total Runs:   1 ( 0.423 s)                 1st ran
# Total Pass:   1 ( 100 % - Avg: 0.423 s)    # Devices tasks:   1
# Total Fail:   0 ( 0 % - Avg: 0.000 s)      # Tasks Pass:      1 ( 100 % )
# Total Cancel:0 ( 0 % - Avg: 0.000 s)       # Tasks Fail:      0 ( 0 % )

Session start:    13:31 Mon Apr 09
Operator:  texas
```

[ ] Continuous Scan in background every 10 seconds

Figure 45. Scanning For Devices

4. After the scan is finished, click the Start button to run the script (shown in Figure 46).



```
System File
  Device: TPS544C25@36
    Import Config for TPS544C25 @ PMBus
      Import_Config_and_Validate
```

Drag a column header here to group by that column

Time	Severity	Type	Extra	Message
02:09:08.750	GENERAL	CUSTOMER	none	Start Scanning for devices...
02:09:08.943	GENERAL	CUSTOMER	none	Found TPS544C25 @ address 36
02:09:08.946	SUCCESS	CUSTOMER	none	All devices in the script are found. Click on "Start" button to run the script

Autoscroll [✓] Show PMBus/USB Adapter/Instrumentation Activity [ ]

Batch option: Set the number of times to re-run script
Runs Left # 0

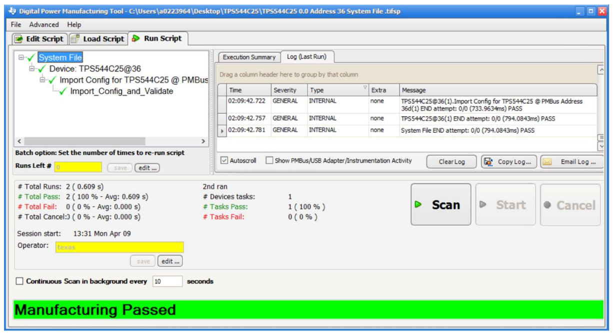
```
# Total Runs:   1 ( 0.423 s)                 1st ran
# Total Pass:   1 ( 100 % - Avg: 0.423 s)    # Devices tasks:   1
# Total Fail:   0 ( 0 % - Avg: 0.000 s)      # Tasks Pass:      1 ( 100 % )
# Total Cancel:0 ( 0 % - Avg: 0.000 s)       # Tasks Fail:      0 ( 0 % )

Session start:    13:31 Mon Apr 09
Operator:  texas
```

[ ] Continuous Scan in background every 10 seconds
Found all devices in the system

Figure 46. Running The Script

5. When the script is finished running, you see Manufacturing Passed on the bottom of the screen (shown in Figure 47).


Digital Power Manufacturing Tool - C:\Users\a0223964\Desktop\TPS544C25\TPS544C25 0.0 Address 36 System File .tifsp
File
Advanced
Help
Edit Script
Load Script
Run Script
System File
Device: TPS544C25@36
Import Config for TPS544C25 @ PMBus
Import_Config_and_Validate
Execution Summary
Log (Last Run)
Clear Log
Copy Log ...
Email Log ...
Scan
Start
Cancel
save
edit ...

Drag a column header here to group by that column

Time	Severity	Type	Extra	Message
02:09:42.722	GENERAL	INTERNAL	none	TPS544C25@36(1).Import Config for TPS544C25 @ PMBus Address 36d(1) END attempt: 0/0 (733.9634ms) PASS
02:09:42.757	GENERAL	INTERNAL	none	TPS544C25@36(1) END attempt: 0/0 (794.0843ms) PASS
02:09:42.781	GENERAL	INTERNAL	none	System File END attempt: 0/0 (794.0843ms) PASS

✓ Autoscroll ☐ Show PMBus/USB Adapter/Instrumentation Activity

Batch option: Set the number of times to re-run script
Runs Left # 0

```
# Total Runs:   2 ( 0.609 s)                 2nd ran
# Total Pass:   2 ( 100 % - Avg: 0.609 s)    # Devices tasks:   1
# Total Fail:   0 ( 0 % - Avg: 0.000 s)      # Tasks Pass:      1 ( 100 % )
# Total Cancel:0 ( 0 % - Avg: 0.000 s)       # Tasks Fail:      0 ( 0 % )

Session start:    13:31 Mon Apr 09
Operator:  texas
```

☐ Continuous Scan in background every 10 seconds

Manufacturing Passed

Figure 47. Manufacturing Passed Alert

5 References

1. Fusion Digital Power Manufacturing GUI User's Guide, Script Runner, February 2014.
2. Fusion Digital Power Manufacturing Tool User's Guide, Script Writer, April 2010.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from A Revision (March 2019) to B Revision	Page
• Replaced figure 2.	2

Changes from Original (May 2018) to A Revision	Page
• Edited application report for clarity.	1
• Changed images to be more uniform in size.	1