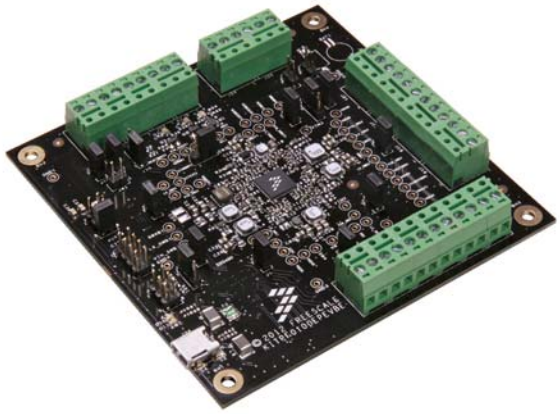
Freescale Semiconductor
User's Guide

Document Number: KTPFSWUG4
Rev. 1.0, 12/2013

KITPFGUI 4.0 Graphical User Interface

for PF Power Management Development Tools



Contents

1 Jump Start

- Go to www.freescale.com/analogtools
- Locate your kit
- Review your Tool Summary Page
- Look for

- Download documents, software and other information

2 Important Notice

Freescale provides the enclosed product(s) under the following conditions:

This evaluation kit is intended for use of ENGINEERING DEVELOPMENT OR EVALUATION PURPOSES ONLY. It is provided as a sample IC pre-soldered to a printed circuit board to make it easier to access inputs, outputs, and supply terminals. This EVB may be used with any development system or other source of I/O signals by simply connecting it to the host MCU or computer board via off-the-shelf cables. This EVB is not a Reference Design and is not intended to represent a final design recommendation for any particular application. Final device in an application will be heavily dependent on proper printed circuit board layout and heat sinking design as well as attention to supply filtering, transient suppression, and I/O signal quality.

The goods provided may not be complete in terms of required design, marketing, and or manufacturing related protective considerations, including product safety measures typically found in the end product incorporating the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge. In order to minimize risks associated with the customers applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards. For any safety concerns, contact Freescale sales and technical support services.

Should this evaluation kit not meet the specifications indicated in the kit, it may be returned within 30 days from the date of delivery and will be replaced by a new kit.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typical”, must be validated for each customer application by customer’s technical experts.

Freescale does not convey any license under its patent rights nor the rights of others. Freescale products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use Freescale products for any such unintended or unauthorized application, the Buyer shall indemnify and hold Freescale and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale was negligent regarding the design or manufacture of the part. Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

3 Introduction

The KITPFGUI 4.0 is a flexible and easy-to-use Graphical User Interface (GUI), created to control and configure the customer evaluation boards and development tools provided by Freescale to support the PF series of Power Management Integrated Circuits (PMIC) powered by SMARTMOS technology.

The new “driverless” environment allows to automatically detect and recognize the board connected through the USB port, enabling the specific features and controls for each board. The official boards supported by the KITPFGUI 4.0 are listed below:

- KITPFPGMEVME (Interfacing with PF0100 or PF0200)
- KITPF0100EPEVBE
- KITPF0200EPEVBE (coming up soon)

This document is intended to provide a detailed description of all the features of the KITPFGUI 4.0, when operating any of the development tools mentioned above.

Note: The KITPFGUI 4.0 is prepared to operate with the new generation of development boards and does not support older versions as stated in Table 1.

4 Software and Drivers Installation

4.1 Hardware Requirements

- PC with Windows XP or Windows 7 operating system
- Standard USB port

4.2 Installing the KITPFGUI 4.0

1. Create a directory on a safe location of your PC. For example: C:\Freescale\KITPFGUI 4.0
2. Extract the KITPFGUI 4.0.zip file into that directory.
3. Launch the "setup.exe" program.
4. When the following pop-up dialog appears, press the “Install” button.

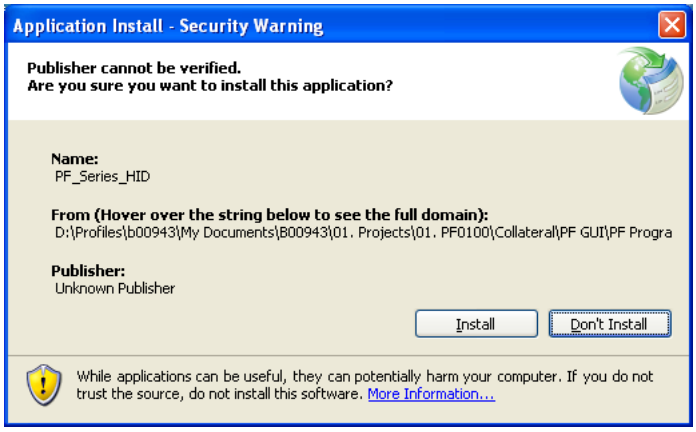


Figure 1. KITPFGUI 4.0 Installation Window

If the installation is successful, the application GUI displays the screen shown in Figure 2.

5 KITPFGUI 4.0 Description

The KITPFGUI 4.0 is a flexible and easy-to-use Graphical User Interface (GUI), created to control and configure the customer evaluation boards and development tools provided by Freescale to support the PF series of Power Management Integrated Circuits (PMIC).

The main features of the KITPFGUI 4.0 are:

1. Automatic detection of the PF development tool connected
2. Read/write access to PF Device registers
3. Intuitive interface for controlling the PF device during operating mode
4. Monitoring all interrupts manually or continuously
5. Manual or imported OTP configuration
6. Single and multi-part programming
7. Scrip editor for prototyping, test emulation, or customized operation of the PF device
8. Saving and recalling customized scripts and configuration files

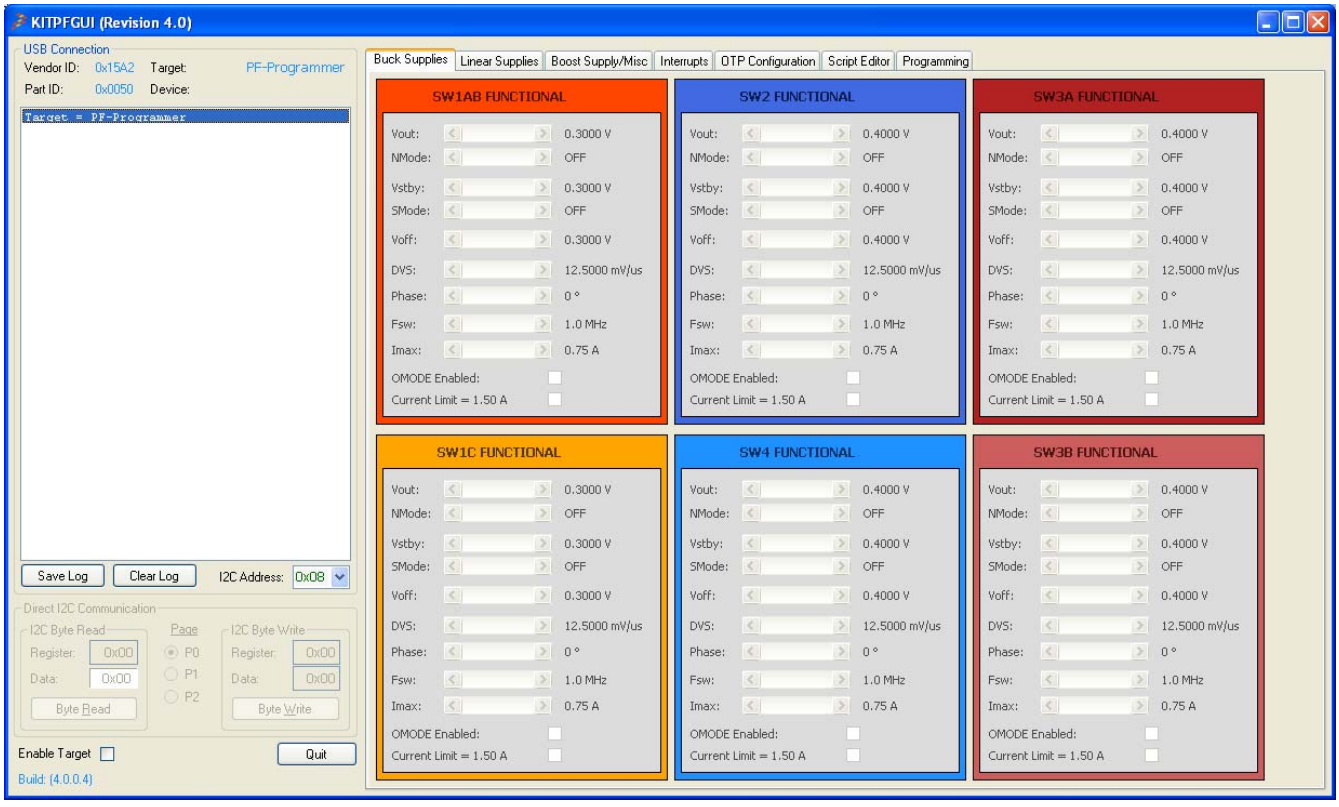


Figure 2. KITPFGUI 4.0 Graphical User Interface

5.1 USB Connection

Connect the device under control to the USB port and wait until the KITPFGUI 4.0 detects and recognizes the board connected as shown in Figure 3.

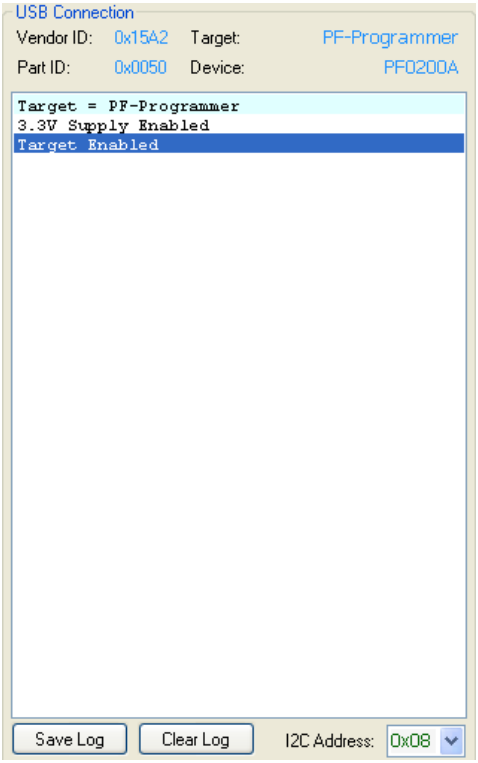


Figure 3. Device Detection and Recognition

Each Freescale PF Development Tool (Target) has a unique vendor ID and Part ID. Proper Vendor ID, Part ID, Target, and Device information is shown on the top section of the USB connection area. Table 1 presents the current boards supported by the KITPFGUI 4.0, and their respective identification values.

Table 1. Freescale PF Development Tool ID

Kit name	Hardware REV(2)	Vendor ID	Part ID	Target	Device
KITPFPGMEVME	Rev B	0x15A2	0x0050	PF-Programmer	(1)
KITPF0100EPEVBE	Rev D(3)	0x15A2	0x00F5	PF0100 EVK	PF0100
KITPF0200EPEVBE	Rev D	0x15A2	0x00F6	PF0200 EVK	PF0200

Notes:

1. Device is dependent on the PMIC to be programmed on the target application board or programming socket.
2. KITPFGUI 4.0 support is guaranteed for the mentioned board revision and higher, unless specified.
3. KITPF0100EPEVBE REV B boards labeled TDA4772 are also supported on the KITPFGUI 4.0

The main log area reports all the activity during the session, which can be saved or cleared at any time with the “Save Log” or “Clear Log” commands respectively. Finally, the KITPFGUI 4.0 allows changing the I2C address to

communicate with a PF device whose I2C address was changed during OTP programming. By default, the KITPFGUI 4.0 enables the default address (0x08) used in all the standard part numbers provided by Freescale.

5.2 Single Data I²C Communication

Use the “Byte Write” button to write one byte of data to any given register of the PF device and use the “Byte Read” button to read back the register contents at the given address. To simplify the commands to write onto the different register pages, use the P0, P1, and P2 options to automatically switch to the specific page, just provide the correct address and data to write or read.

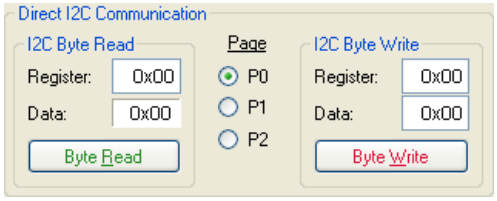


Figure 4. Single data Write/Read

5.3 Enabling the Target Board

To avoid false programming and undesired configuration loaded upon connection, the KITPFGUI 4.0 interface does not communicate to the PMIC when the target board is connected, therefore, the user must enable the target by checking the “Enable Target” box in the bottom left corner of the GUI. This enables the KITPFGUI 4.0 to automatically load the current content of the PF device and fill in the configuration tabs accordingly. Note that the KITPFGUI 4.0 disables the target every time the board is disconnected, hence the process must be repeated to gain control over the PMIC device every time. Figure 5. shows the general controls to enable communication with the PMIC and exit the KITPFGUI 4.0, as well as the current software release version.

Figure 5. General controls

5.4 Buck Supplies Control

The KITPFGUI 4.0 provides simplified access to the configuration parameter of all the Buck converters in the PF devices as depicted in Table 2.

Table 2. Buck Converter Availability on PF PMICS

Register	PF0100	PF0200
SW1AB	√	√
SW1C	√	N/A
SW2	√	√
SW3A	√	√
SW3B	√	√
SW4	√	N/A

Each regulator is controlled through an independent control box with a graphical representation of all the parameters that can be modified via I2C. Figure 6 shows an example of the control box for any of the buck regulators on the PF PMIC. When the user modifies any parameter on one of the control box, the changes do not take effect until the **Update** button is pressed, as shown in Figure 6.

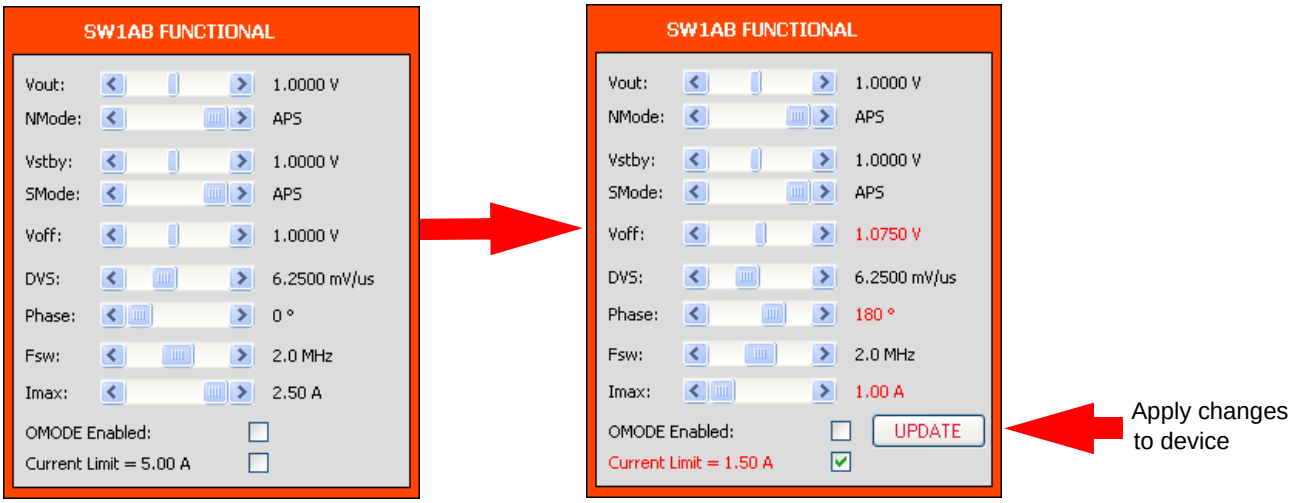


Figure 6. Switching Regulator Control Box

Table 3 describes each one of the functions listed in the Buck regulator control box.

Table 3. SWx Control Box description.[4]

Function	Description
Output Voltage Range (read only)	Displays the current output voltage operating range set by the bit SWx[6] during OTP configuration.
Vout	Sets the voltage of the regulator during normal operation. • Refer to device specification for all possible values • **I2C Register:** SWxVOLT • **I2C Bits Modified:** SWx[5:0][5]
NMode	Set the operating mode during normal operation. • Operating Modes: OFF, PWM, PFM and APS • **I2C Register:** SWxMODE • **I2C Bits Modified:** SWxMODE[3:0]
VSTBY	Sets the voltage of the regulator during standby operation. • Refer to device specification for all possible values • **I2C Register:** SWxSTBY • **I2C Bits Modified:** SWxSTBY[5:0][5]
SMode	Sets the operating mode during standby operation. • Operating Modes: OFF, PWM, PFM and APS • **I2C Register:** SWxMODE • **I2C Bits Modified:** SWxMODE[3:0]
Voff	Sets the voltage of the regulator during OFF operation. • Refer to device specification for all possible values • **I2C Register:** SWxOFF • **I2C Bits Modified:** SWxOFF[5:0][5]

Table 3. SWx Control Box description.[4] (continued)

Function	Description

DVS	Sets the DVS speed for the Buck regulator[5].
	
	• DVS speed in Low Output Voltage Range (SWx[6] or SWxSTBY[6] = 0)
	  • 25 mV / 2.0 µs = 12.5000 mV/µs
	  • 25 mV / 4.0 µS = 6.2500 mV/µs
	  • 25 mV / 8.0 µS = 3.12500 mV/µs
	  • 25 mV / 16 µS = 1.56250 mV/µs
	
	• DVS speed in High Output Voltage Range (SWx[6] or SWxSTBY[6] = 1)
	  • 50 mV / 4.0 µs = 12.5000 mV/µs
	  • 50 mV / 8.0 µS = 6.2500 mV/µs
	  • 50 mV / 16 µS = 3.12500 mV/µs
	  • 50 mV / 32 µS = 1.56250 mV/µs
	
	• **I2C Register:** SWxCONF
	• **I2C Bits Modified:** SWxDVSSPEED[1:0]

Phase	Sets the switching phase of the Buck regulator. • **Phase shift options:** 0°, 90°, 180° and 270° • **I2C Register:** SWxCONF • **I2C Bits Modified:** SWxPHASE[1:0]
Fsw	Sets the operating frequency of the Buck regulator. • **Frequency selection:** 1.0 MHz, 2.0 MHz and 4.0 MHz • **I2C Register:** SWxCONF • **I2C Bits Modified:** SWxFREQ[1:0]
Imax	Programs the maximum operating current. • Refer to device specification for all possible values • **I2C Register:** SWx PWRSTG - Extended Page 2 • **I2C Bits Modified:** SWx_PWRSTG[2:0]

OMODE Enabled	Enable the part to operate in OFF mode.
	• Check box option:
	  • Unchecked = Off mode disabled
	  • Checked = Off mode enabled
	• **I2C Register:** SWxMODE
	• **I2C Bits Modified:** SWxOMODE

Current Limit	Select current limit level.
	• Check box option:
	  • Unchecked = High level current limit
	  • Checked = Low level current limit
	• **I2C Register:** SWxCONF
	• **I2C Bits Modified:** SWxILIM

Notes:

4. "x" stands for the number of regulator to be configured. I.e., SW1AB, SW1C, SW2, SW3A, SW3B, or SW4.
5. SW2, SW3A/B and SW4 are capable of operating in low or high output voltage range. The operating mode is selected by writing to bit SWx[6] and SWxSTBY[6] during OTP configuration. This bit is read only during normal system operation.

5.5 Linear Supplies Control

The KITPFGUI 4.0 provides simplified access to the configuration parameter of all the LDO supplies in the PF devices, as shown in Figure 7.

Figure 7. LDO Control Box

Table 4 and Table 6 describe each one of the functions listed in the control boxes for all LDOs.

Table 4. VGENx LDO Control Box description(6)

Function	Description
Vout	Sets the voltage of the regulator during normal operation. • Refer to device specification for all possible values • **I2C Register:** VGENxCTL • **I2C Bits Modified:** VGENx[3:0]

Supply Enabled	Enables or disables the linear regulator.
	• Check box option:
	  • Unchecked = Disabled
	  • Checked = Enabled.
	• **I2C Register:** VGENxCTL
	• **I2C Bits Modified:** VGENxEN

Table 4. VGENx LDO Control Box description[6] (continued)

Function	Description

Low Power	Enables low power operation for the linear regulator.[7]
	• Check box option:
	  • Unchecked = Low power mode disabled
	  • Checked = Low power mode enabled
	• **I2C Register:** VGENxCTL
	• **I2C Bits Modified:** VGENxLPWR

Standby	Enables regulator during STANDBY operation[7].
	• Check box option:
	  • Unchecked = No regulator control during Standby
	  • Checked = Regulator controlled on Standby
	• **I2C Register:** VGENxCTL
	• **I2C Bits Modified:** VGENxSTBY

Notes:

6. "x" stands for the number of regulators to be configured. I.e., VGEN1, VGEN2, VGEN3, VGEN4, VGEN5, or VGEN6.
7. See Table 5 for detail description of LDO functionality.

Table 5. LDO Control

VGENxEN	VGENxLPWR	VGENxSTBY	STANDBY Event[8]	VGENxOUT
0	X	X	X	Off
1	0	0	X	On
1	1	0	X	Low Power
1	X	1	NO	On
1	0	1	YES	Off
1	1	1	YES	Low Power

Notes:

8. STANDBY event is triggered by the STANDBY pin as describe in the PF device datasheet.

Table 6. VSNVS and VREFDDR Control Box description

Function	Description
VSNVS Control Box	
Vout	Sets the voltage of VSNVS regulator. • Refer to device specification for all possible values • **I2C Register:** VSNVSCTL • **I2C Bits Modified:** VSNVSVOLT[2:0]
VREFDDR Control Box	

Supply Enabled	Enables VREFDDR during STANDBY operation.
	• Check box option:
	  • Unchecked = VREFDDR is off
	  • Checked = VREFDDR is on
	• **I2C Register:** VREFDDRCTL
	• **I2C Bits Modified:** VREFDDREN

Changes to the configuration in the control boxes do not take place until the **Update** button is pressed, as shown in Figure 8.

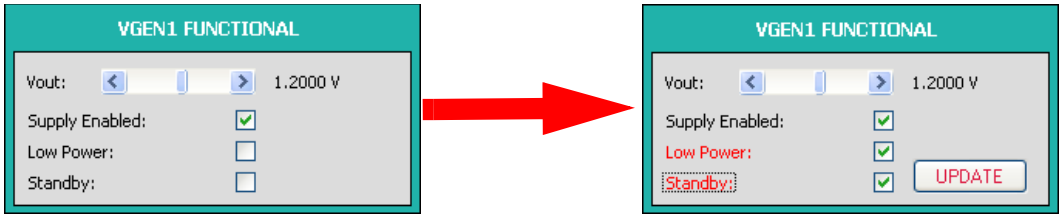


Figure 8. LDO Control Box

5.6 Boost Supply/Misc Control

The **Boost Supply/Misc** tab provides access to the Boost converter control box as well as various general registers on the PF device. Figure 9, shows the **Boost Supply/Misc** tab.

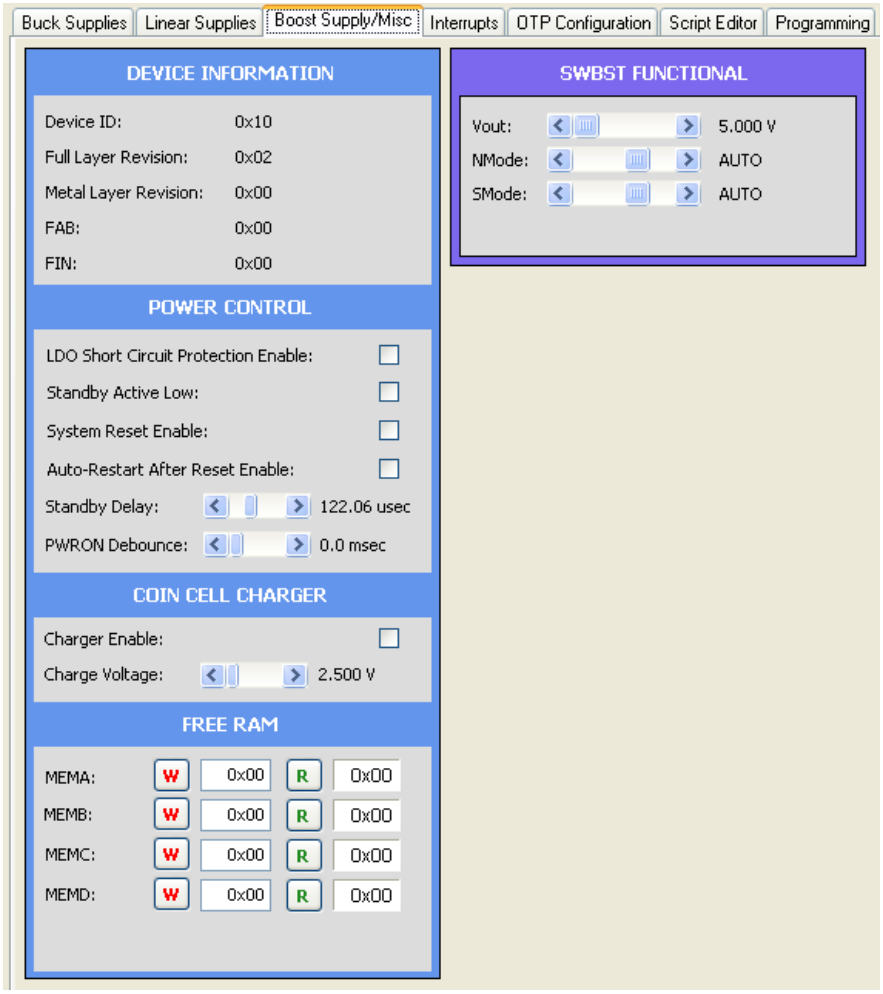


Figure 9. Boost Supply/Misc Control Tab

Table 7 and Table 8 describe all functions within the BOOST SUPPLY/MISC tab.

Table 7. Boost Supply Control Box Description

Function	Description
SWBST Regulator	

Vout	Sets the voltage of the Boost regulator.
	• Possible output voltage
	  • 5.00 V
	  • 5.05 V
	  • 5.10 V
	  • 5.15 V
	• **I2C Register:** SWBSTCTL
	• **I2C Bits Modified:** SWBST1VOLT[1:0]

NMode	Sets the SWBST operating mode in normal operation.
	• Mode selection:
	  • OFF
	  • PFM
	  • AUTO
	  • APS
	• **I2C Register:** SWBSTCTL
	• **I2C Bits Modified:** SWBST1MODE[1:0]

SMode	Sets the SWBST operating mode in STANDBY operation.
	• **Mode selection:**
	  • OFF
	  • PFM
	  • AUTO
	  • APS
	• **I2C Register:** SWBSTCTL
	• **I2C Bits Modified:** SWBST1STBYMODE[1:0]

Table 8. Miscellaneous Control Box Description

Device Information (read only)	
Device ID	Returns the PF device ID.
Full Layer Revision	Returns the full layer revision for the PF device connected.
Metal Layer Revision	Returns the metal layer revision for the PF device connected.
FAB	Freescale internal use
FIN	Freescale internal use
Power Control	

LDO Short Circuit Protection Enable	Describes LDOs behavior during a short circuit event.
	• **Check box option:**
	  • Unchecked = Current Limit
	  • Checked = Shutdown
	• **I2C Register:** PWRCTL
	• **I2C Bits Modified:** REGSCPEN[0]

Table 8. Miscellaneous Control Box Description (continued)

Standby Active Low	Configures the operating level of STANDBY pin.
	• Check box option:
	  • Unchecked = STANDBY pin is Active HIGH
	  • Checked = STANDBY pin is Active LOW
	• **I2C Register:** PWRCTL
	• **I2C Bits Modified:** STANDBYINV[0]

System Reset Enable	Enables a “long press” on PWRON pin to reset the part.
	• Check box option:
	  • Unchecked = No reset possible with PWRON pin
	  • Checked = Reset enabled with long PWRON press
	• **I2C Register:** PWRCTL
	• **I2C Bits Modified:** PWRONRSTEN[0]

Auto-Restart After Reset Enable	Enables an auto-restart after a long PWRON press reset.
	• Check box option:
	  • Unchecked = Auto -restart disabled
	  • Checked = Auto-restart enabled
	• **I2C Register:** PWRCTL
	• **I2C Bits Modified:** RESTARTEN[0]

Standby Delay	Sets the delay to act upon a STANDBY event.
	• Delay times:
	  • 91.55 µs
	  • 122.06 µs
	  • 152.59 µs
	  • 183.10 µs
	• **I2C Register:** PWRCTL
	• **I2C Bits Modified:** STBYDLY[1:0]

PWRON Debounce	Sets the debounce time for PWRON pin.
	• Debounce time:
	  • Turn on: 0.000 ms - Falling: 31.25 ms - Rising: 31.25 ms
	  • Turn on: 31.25 ms - Falling: 31.25 ms - Rising: 31.25 ms
	  • Turn on: 125.0 ms - Falling: 125.0 ms - Rising: 31.25 ms
	  • Turn on: 750.0 ms - Falling: 750.0 ms - Rising: 31.25 ms
	• **I2C Register:** PWRCTL
	• **I2C Bits Modified:** PWRONDBNC[1:0]

Coin Cell Charger	

Charger Enable	Enables the coin cell charger.
	• Check box option:
	  • Unchecked = Disabled
	  • Checked = Enabled
	• **I2C Register:** COINCTL
	• **I2C Bits Modified:** COINCHEN[0]

Charger Voltage	Sets the coin cell charger output voltage. • See device specification for all selectable values • **I2C Register:** COINCTL • **I2C Bits Modified:** VCOIN[2:0]
Ram Memory Slots	
MEMA	Write or Read from Memory register A
MEMB	Write or Read from Memory register B
MEMC	Write or Read from Memory register C
MEMD	Write or Read from Memory register D

5.7 Interrupt Monitoring

The **Interrupts** tab provides access to the four interrupt registers in the functional register map of the PF device. The user can choose to read the interrupts by pressing the **Read Interrupt x** button.

Each interrupt is latched so that even if the interrupt source becomes inactive, the interrupt remains set until cleared. Each interrupt can be cleared by checking the box for the appropriate bit in the Interrupt status register. This also causes the INTB pin to go high.

Each interrupt can be masked by setting the corresponding mask bit to a 1. As a result, when a masked interrupt bit goes high, the INTB pin does not go low. A masked interrupt can still be read from the Interrupt status register.

The sense registers contain status and input sense bits so the system processor can poll the current state of interrupt sources. They are read only, and neither latchable nor clearable.

The user may choose to select the **Poll Interrupt** control to read the corresponding interrupt control box every 250ms. Figure 10 shows the 4 interrupt control boxes contained in the **Interrupts** tab.

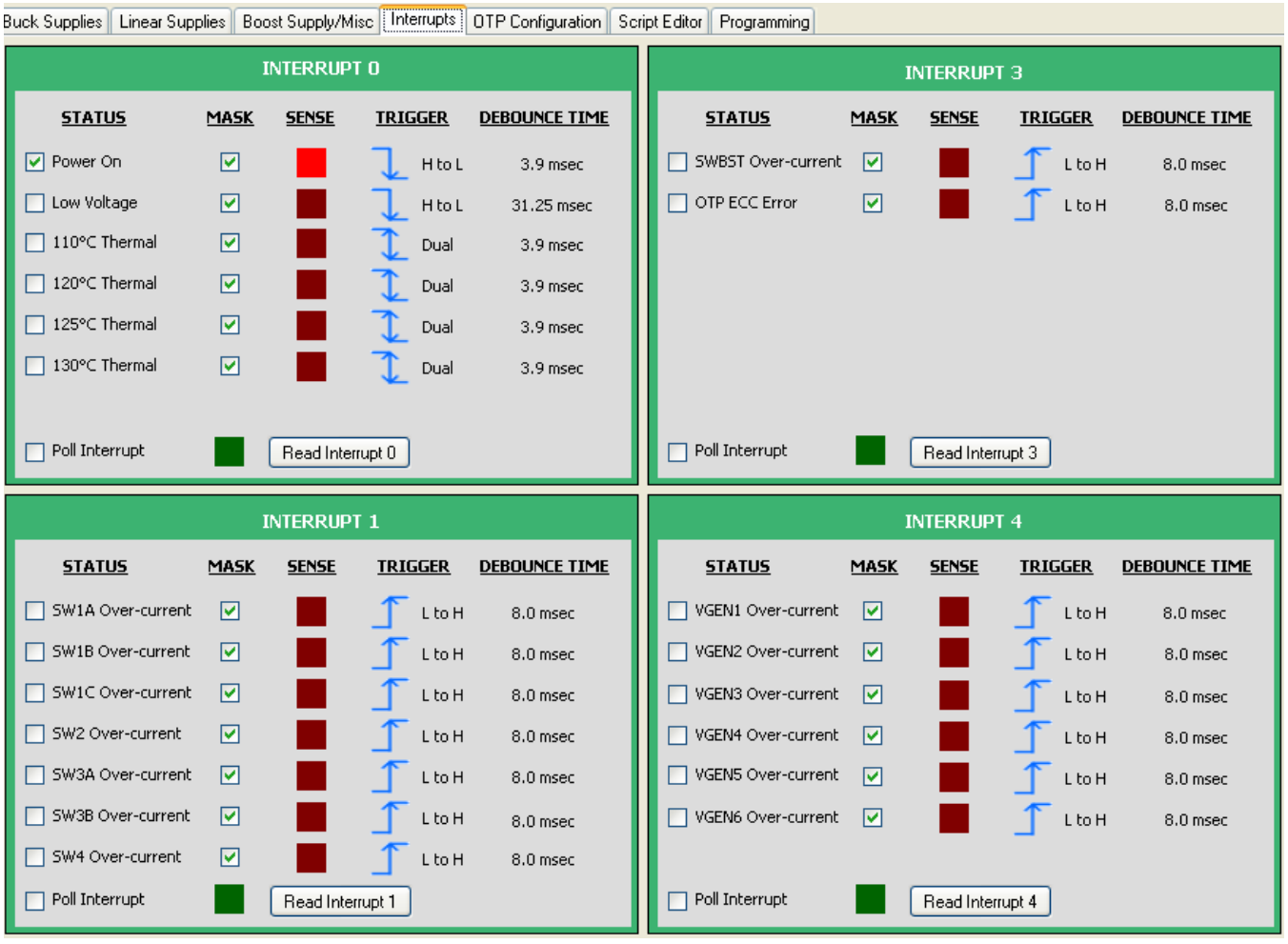


Figure 10. Interrupts Tab

5.8 OTP Configuration

The **OTP Configuration** tab shows the user to graphically inspect and modify the OTP configuration in the PF device. By default, the KITPFGUI 4.0 is set to Run mode, therefore, the OTP configuration cannot be modified until the TBB Mode is enabled. Figure 11 shows the complete view of the **OTP Configuration** tab during Run mode.

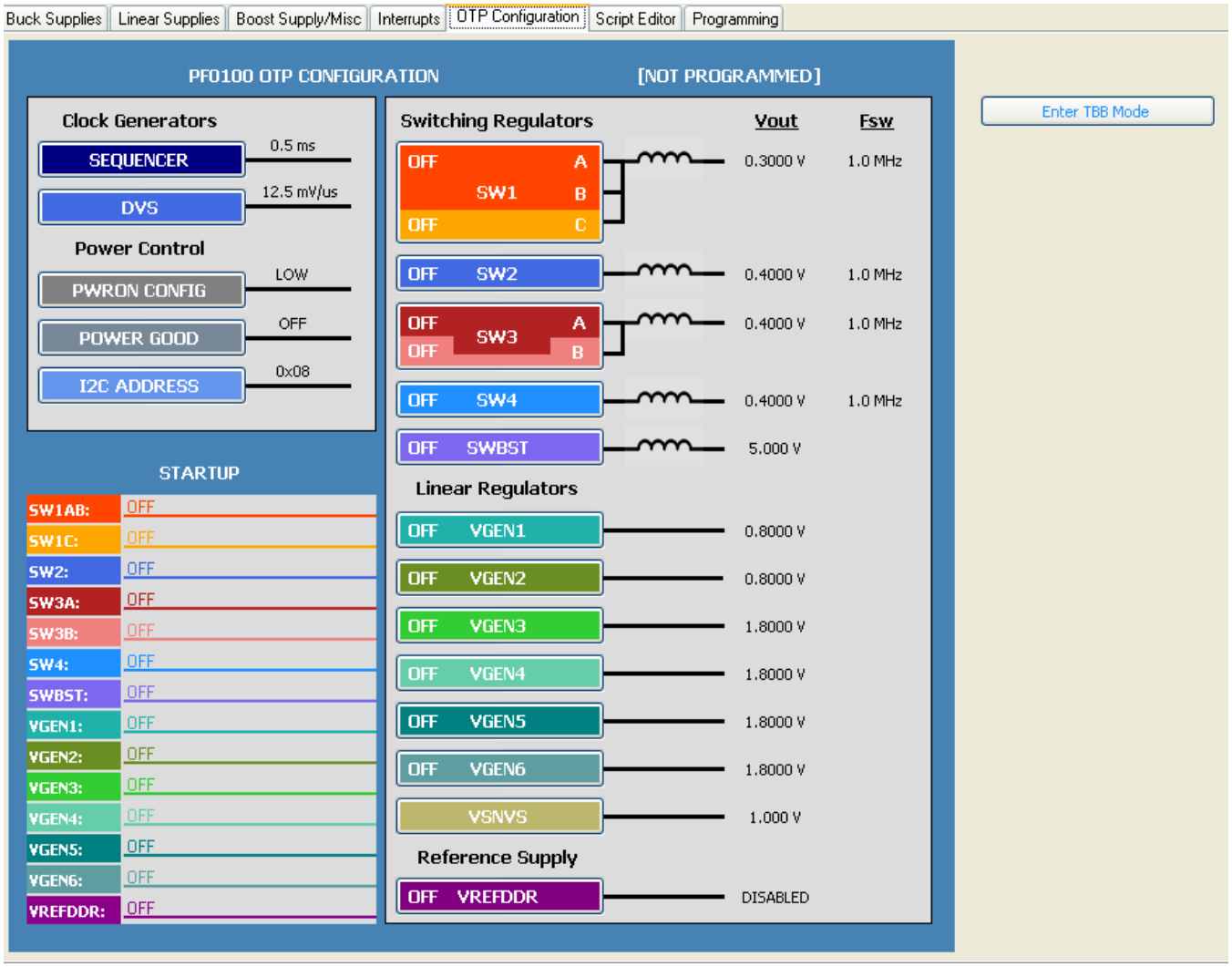


Figure 11. OTP Configuration Tab in "Run Mode"

5.8.1 Enable TBB Mode

When the TBB mode is enabled, the **OTP Configuration** tab changes colors from blue to yellow, and it allows manual configuration of all OTP options in a friendly and intuitive way.

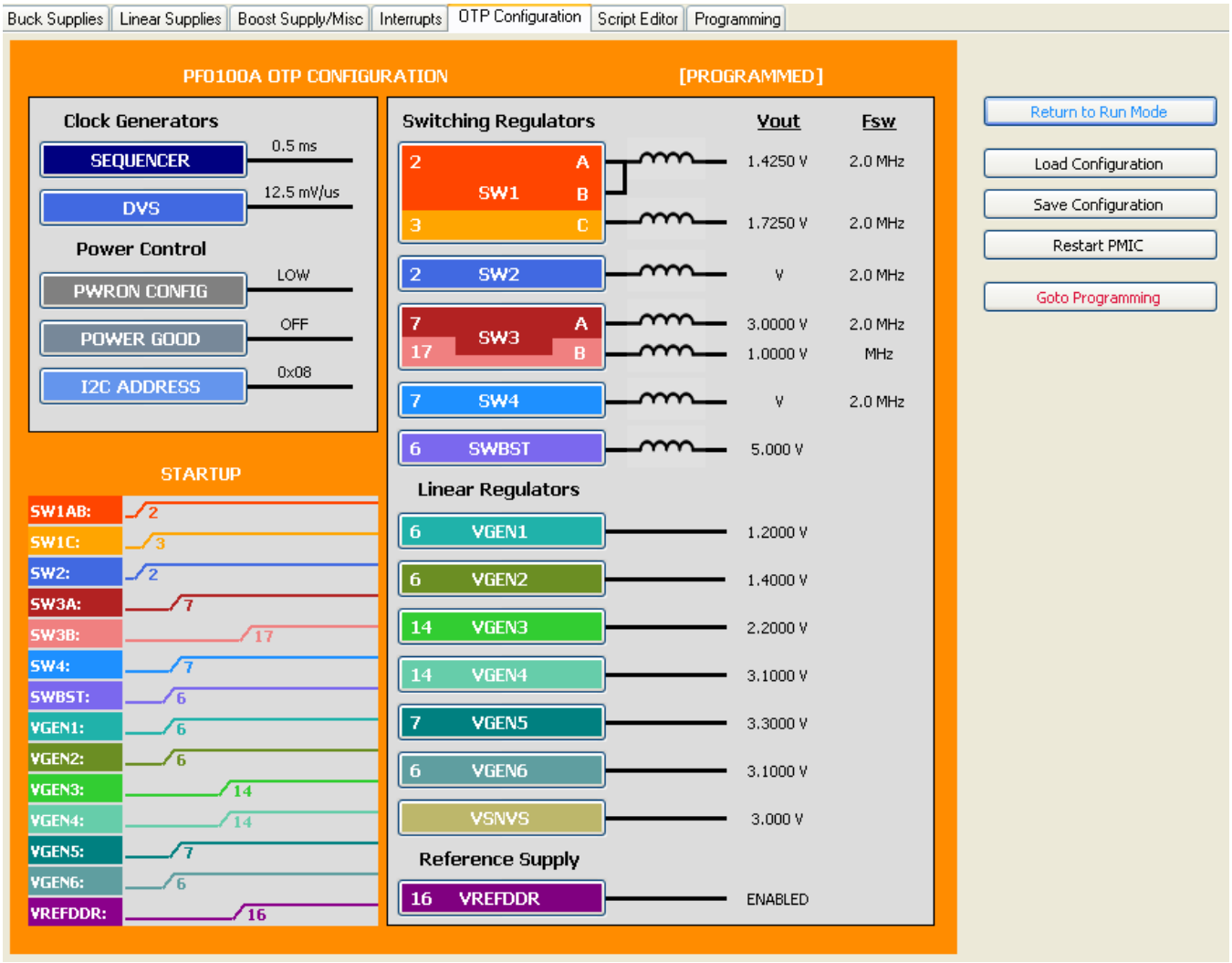


Figure 12. OTP Configuration Tab in “TBB Mode”

5.8.1.1 Manual OTP Configuration

To manually modify the OTP configuration, click on a specific block and a configuration box displays all the configuration options for the respective OTP function.

CLOCK GENERATORS:

- **SEQUENCER**: Sets the SEQ_CLK_SPEED time between regulators turn-on during the power up. The user must select a value of 0.5 ms, 1.0 ms, 2.0 ms, or 4.0 ms during OTP configuration and device uses this value during the power up sequence every time.

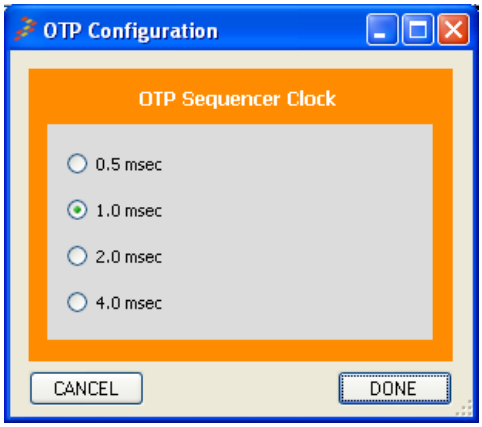


Figure 13. OTP Sequence Clock

- **DVS:** Sets the default DVS configuration to be used during power up from the OTP fuses. The selected DVS configuration is applied to all the switching regulators. However, the step size and time are defined by the output voltage range set by bit SWx[6] or SWxSTBY[6] as shown in Table 9.

Table 9. OTP Programmable DVS values

DVS speed in Low Output Voltage Range	DVS speed in High Output Voltage Range
25 mV / 2.0 µs = 12.5000 mV/µs	50 mV / 4.0 µs = 12.5000 mV/µs
25 mV / 4.0 µS = 6.2500 mV/µs	50 mV / 8.0 µS = 6.2500 mV/µs
25 mV / 8.0 µS = 3.12500 mV/µs	50 mV / 16 µS = 3.12500 mV/µs
25 mV / 16 µS = 1.56250 mV/µs	50 mV / 32 µS = 1.56250 mV/µs

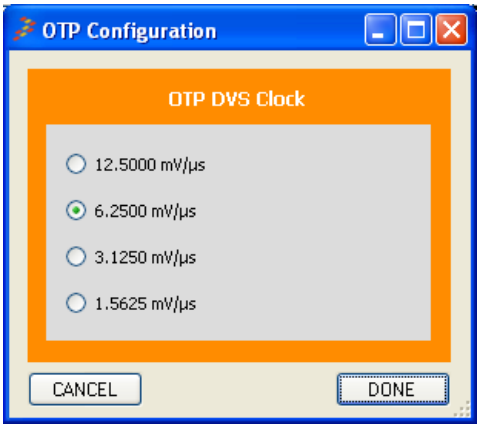


Figure 14. OTP DVS Clock

POWER CONTROL:

- **PWRON CONFIG**: Sets the behavior of the PWRON pin as level sensitive or edge sensitive, as shown in Table 10.

Table 10. PWRON configuration

PWRON CONFIG	Mode
LOW	PWRON pin HIGH = ON PWRON pin LOW = OFF or Sleep mode
HIGH	PWRON pin pulled LOW momentarily = ON PWRON pin LOW for 4.0 seconds = OFF or Sleep mode

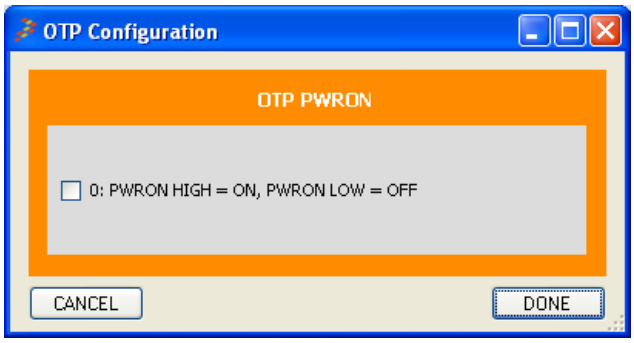


Figure 15. OTP PWRON

- **POWER GOOD**: Sets the configuration of the RESETBMCU pin as the “Power up OK” indicator or as a power fault indicator as described in Table 11.

Table 11. POWER GOOD configuration

POWER GOOD	Function	Description
LOW	Power up OK Indicator	In this default mode, RESETBMCU is deasserted 2.0 to 4.0 ms after the last regulator in the start-up sequence is enabled. RESETBMCU can be used to bring the processor out of reset, or as an indicator that all supplies have been enabled; it is only asserted for a turn-off event.
HIGH	Power Fault indicator	RESETBMCU is deasserted after the start-up sequence is completed only if no faults occurred during start-up. At anytime, if a fault occurs and persists for 1.8 ms typically, RESETBMCU is asserted LOW. The PF device is turned off if the fault persists for more than 100 ms typically. The PWRON signal restarts the part, though if the fault persists, the sequence described above is repeated.

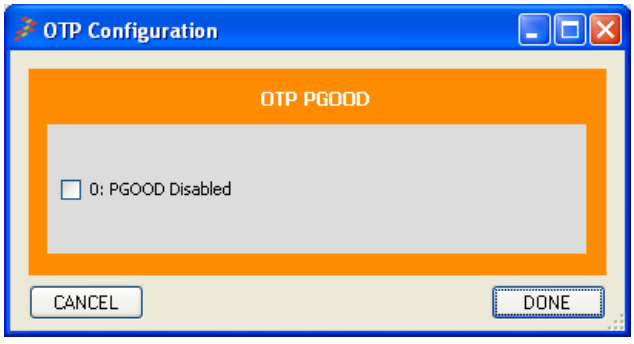


Figure 16. OTP PGOOD

- **I2C Address**: Allows to program the physical I^2C address for the PF device in the range of 0x08 to 0x0F.

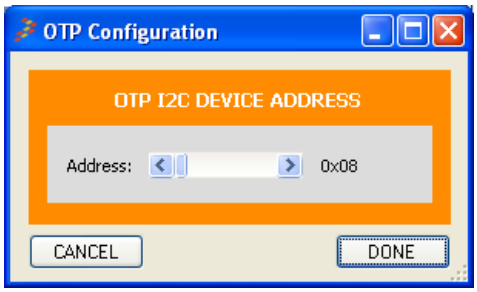


Figure 17. I²C Device Address

STARTUP

Shows the graphical representation of the turn-on position for each regulator during the startup sequence.

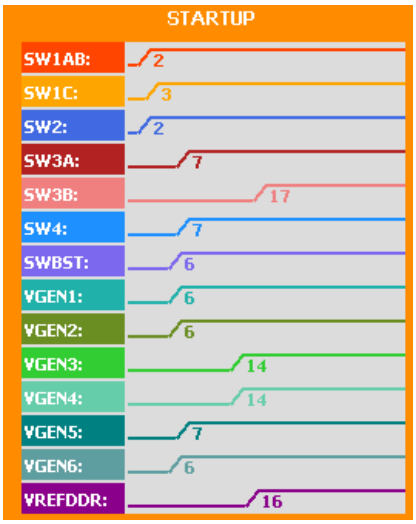


Figure 18. OTP Startup Sequence

SWITCHING REGULATORS

Shows graphically the latest configuration of all switching regulators including the following parameters:

- Turn on place during power up
- Hardware configuration (SW1A/B/C and SW3A/B, SW4)
- Default output voltage
- Output voltage range (SW2, SW3A/B, SW4)
- Default switching frequency

Parameters for each regulator can be modified by clicking on the respective block and a pop-up window with all the configurable options appears. Figure 19 and Figure 20 show examples of the pop-up window to set all the OTP parameter for multi-channel or single channel regulators respectively.

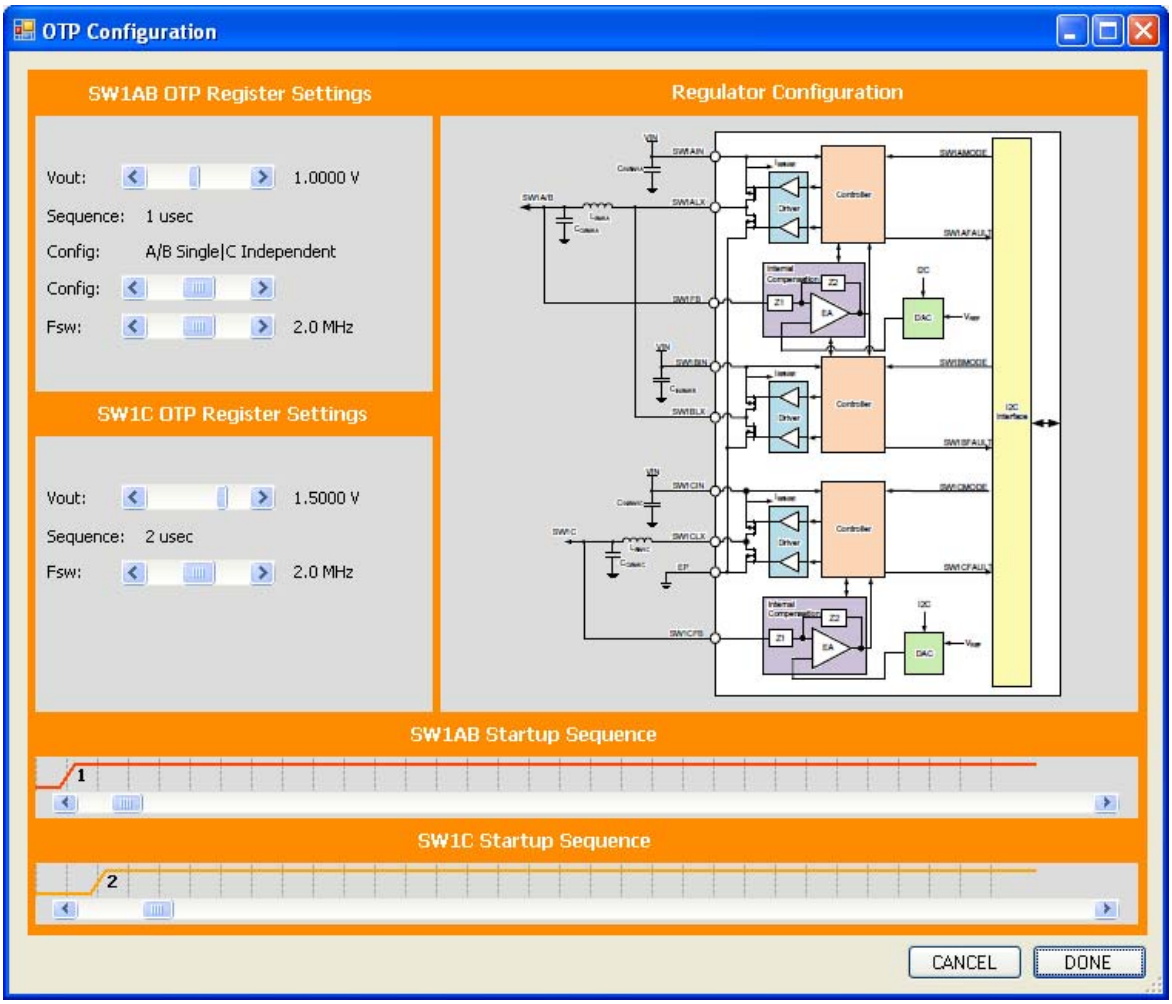


Figure 19. Multi-channel OTP Configuration Pop-up Window

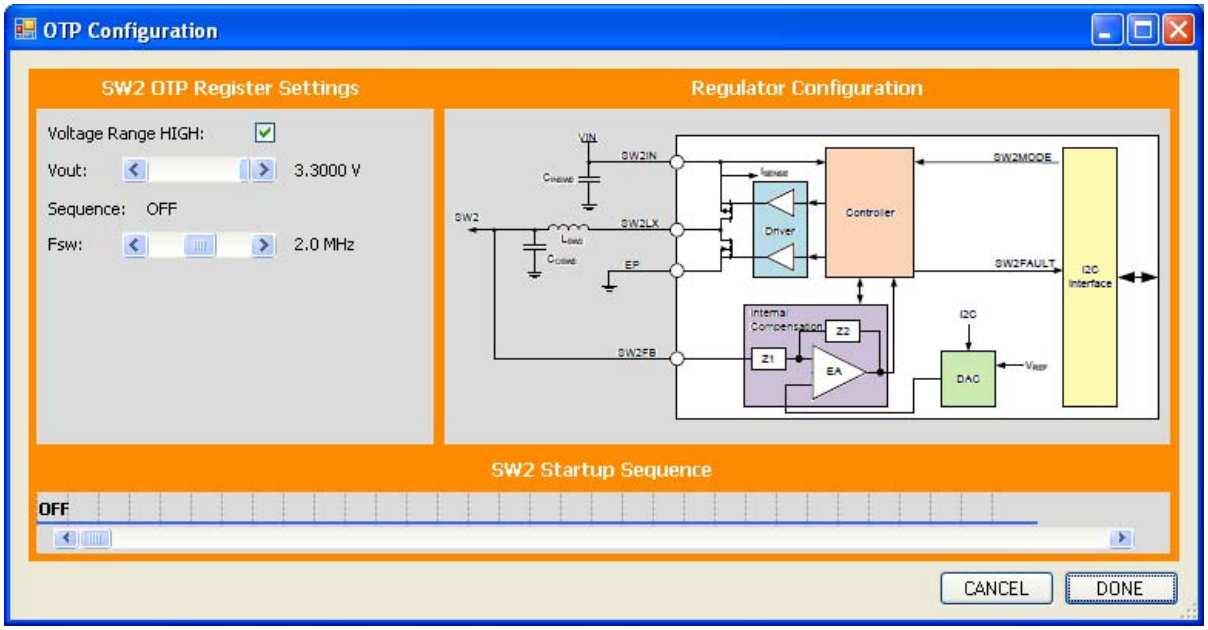


Figure 20. Single Channel OTP Configuration Pop-up Window

LINEAR REGULATORS AND REFERENCE SUPPLIES

Shows a graphical representation of the latest configuration of all linear regulators and reference supplies, including the following parameters:

- Turn on place during power up
- Default output voltage
- Default On/Off state

Parameters for each regulator can be modified by clicking on the respective block and a pop-up window with all the configurable options appears. Figure 21 shows an example of the pop-up window to set all the OTP parameters for any given regulator.

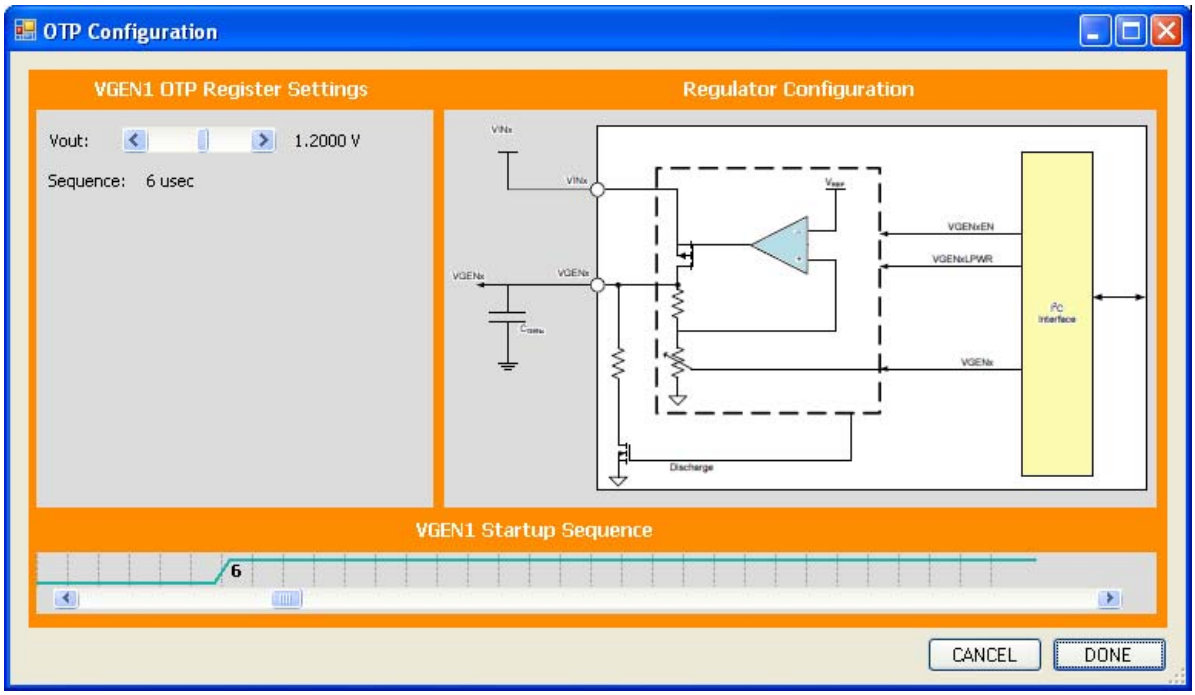


Figure 21. LDO OTP Configuration Pop-up Window

5.8.1.2 Save an OTP Configuration

To save an OTP configuration:

- Click on the **Save Configuration** button.
- Select the directory in which the new file is created.
- Type the name of the new configuration file, and then click **Save**.

5.8.1.3 Load OTP Configuration

To load an OTP configuration from a .cfg file:

- Click the **Load Configuration** button.
- Select the configuration file and then click **Open**. This loads the configuration from the file into the **OTP Configuration** tab.

5.8.1.4 Test the Power Up Sequence in TBB Mode

When working with a Freescale customer kit such as KITPF0100EPEVBE or KITPF0200EPEVBE; or programming a full featured application board using the KITPFPGMEVME, the KITPFGUI 4.0 allows a user to test the OTP power up sequence in TBB mode to emulate the final configuration after OTP programming.

- Click on the **Restart PMIC** button. This toggles the PWRON pin forcing a power restart.
- PMIC powers up with the new configuration.
- Optionally, the PMIC can be restarted by toggling the VIN supply with a valid voltage on LICELL.

5.8.1.5 Programming a Part with the Current TBB Configuration

Once the manual OTP configuration is finalized, the data is kept in RAM memory so the user can use it to program a new PF device.

- Click on the **Goto Programming** button.
- The KITPFGUI 4.0 automatically changes to the **Programming** tab and prepares the information in the file area.
- Follow steps 3 to 5 in section Programming Steps.

When programming an OTP configuration from TBB mode, the KITPFGUI 4.0 automatically disables the TBB mode.

5.9 Programming PF Device

The **Programming** tab allows a number of options to import pre-set configurations from various types of sources; once the desired configuration is properly loaded. It provides full control to program a single or multiple parts, verifying proper programming, and also exporting the current configuration into various formats of script files. Figure 22 shows the **Programming** tab.

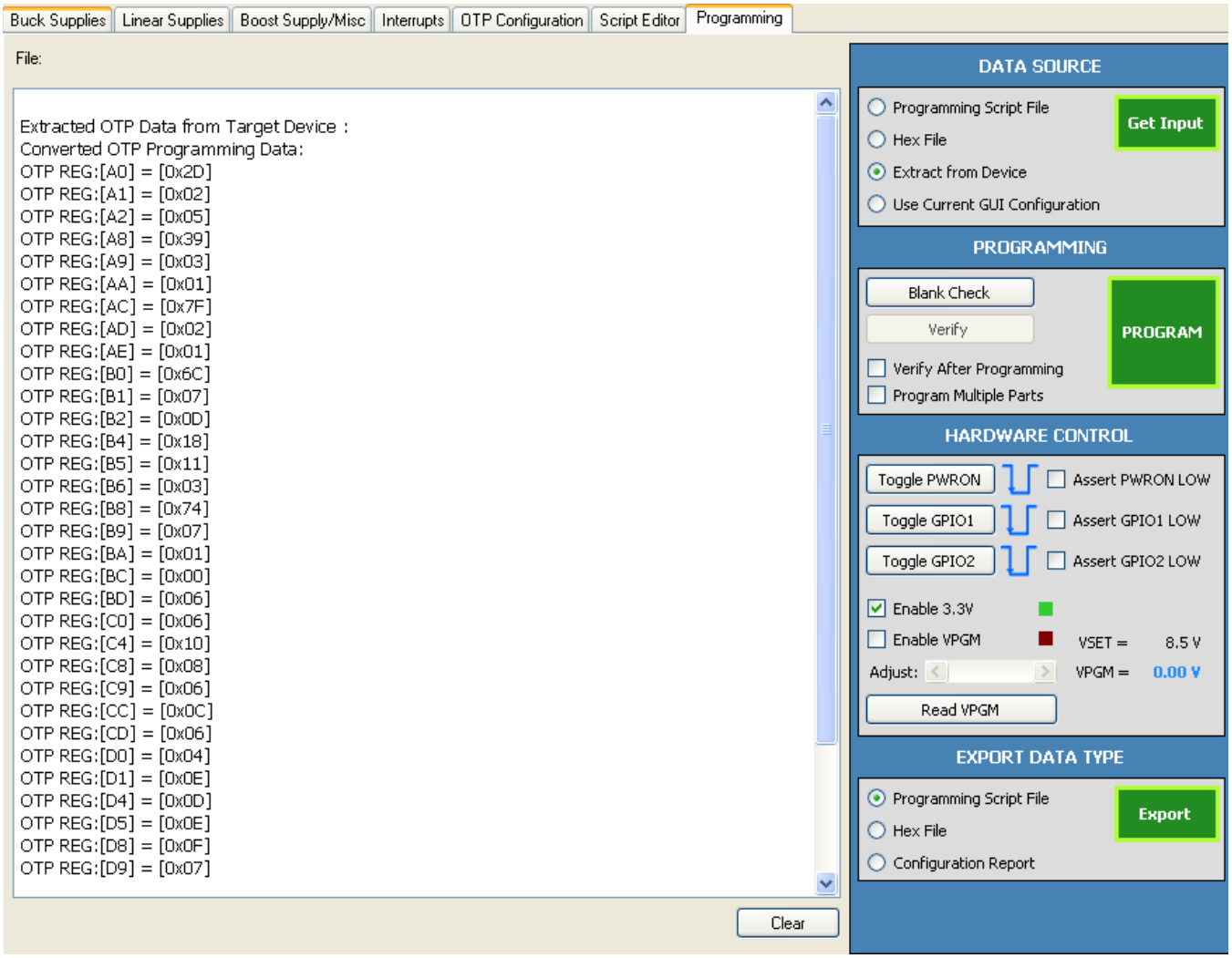


Figure 22. Programming Tab

The **Programming** tab is divided in five sections described below:

Table 12. Programming Tab Description

Section	Feature	Description
File Area	Shows the sequential data to be programmed on the PF device. Data is traduced from the selected data source and prepared for programming into a new device.	
DATA SOURCE	Selects the source of the OTP configuration.	
	Programming script file	Uses a .txt script file generated through the script editor as the source for OTP programming
	Hex File	Uses a custom .hex file as the source for OTP programming
	Extract from Device	Extracts the OTP configuration from a physical device. The data is store in RAM memory and prepared to be programmed on a new device.
	Current OTP configuration	The KITPFGUI 4.0 uses the information from the OTP **Configuration** tab as the source for the OTP programming.
	Get Input Button	Translates the source information into a suitable format to program the OTP register in a new PF device
PROGRAMMING	Blank Check	Allows to verify if the current PF device has been programmed already or if it is a black device and thus ready to be programmed
	Verify	Enables manual verification after programming a device
	Verify after Programming	Check this box for automatic verification after programming a device
	Program Multiple parts	Allows programming of multiple files with the same OTP configuration
HARDWARE CONTROL	Toggle Power On	Toggle the PWRON pin to restart the device with the new OTP configuration or in TBB mode
	Assert PWRON Low	When checked, the PWRON pin is set low, and released when the box is unchecked
	Toggle GPIO1	Toggle the GPIO1 pin provided for general purpose control
	Assert GPIO1 Low	When checked, the GPIO1 pin is set low, and released when the box is unchecked
	Toggle GPIO2	Toggle the GPIO1 pin provided for general purpose control.
	Assert GPIO2 Low	When checked, the GPIO1 pin is set low, and released when the box is unchecked
	Enable 3.3 V	Manually enables the 3.3 V supply provided by the programming device. Typically tied to VIN on the PF device for OTP programming.
	Enable VPGM	Manually enables the 8.5 V supply used to program the OTP registers.
	Adjust bar	Allows fine adjustment of the VPGM voltage from 7.7 V to 9.5 V
	Read VPGM	Reads the current voltage level of VPGM
EXPORT DATA TYPE	Programming Script File	Exports the current configuration to the Scrip Editor
	Hex File	Exports the current configuration to the Scrip Editor
	Configuration Report	Exports a configuration report to the Scrip Editor

5.9.1 Programming Steps

1. Select the data source and press **Get Input**.

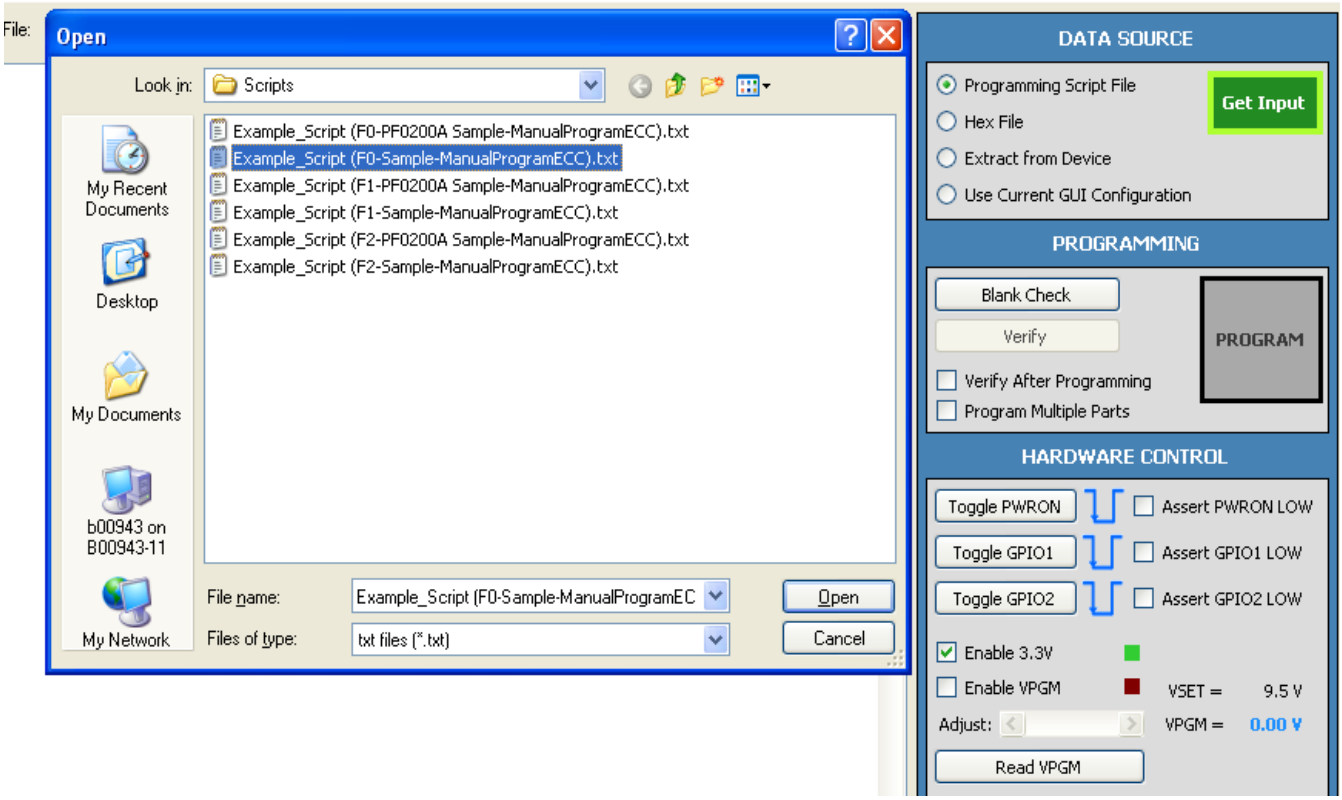


Figure 23. Select Data Source

2. Convert Script File to register data prior to programming.

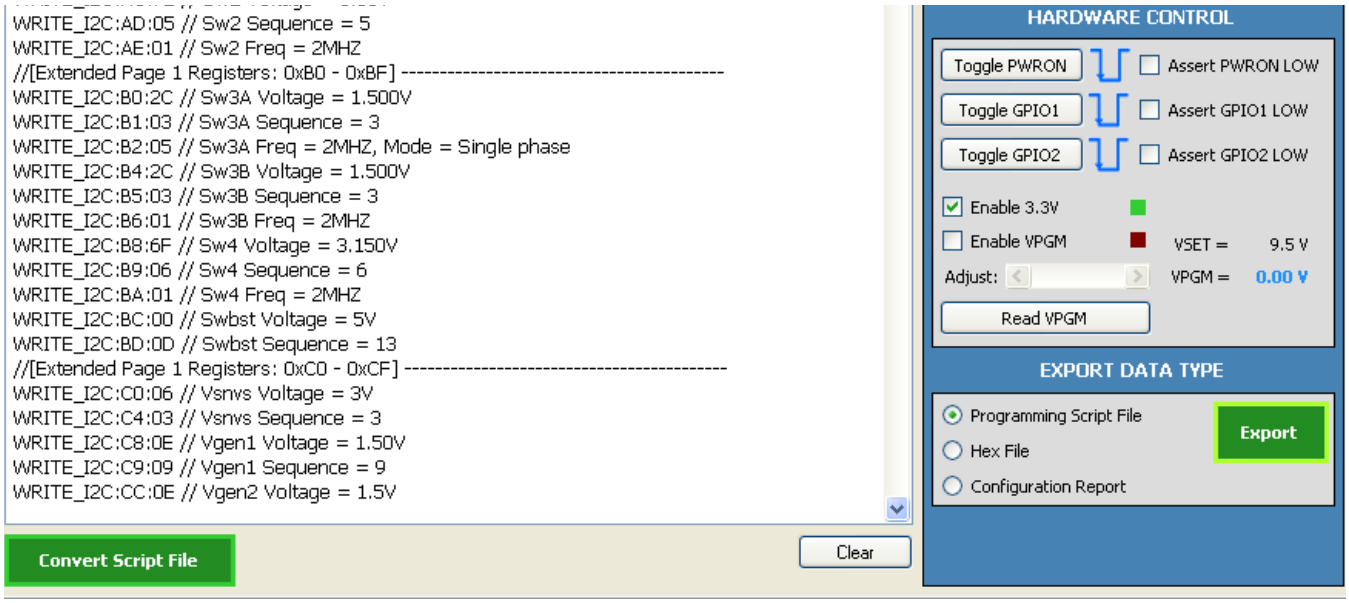


Figure 24. Convert Script File

3. Check if the PF device is a “Blank” device or if it has been programmed already.

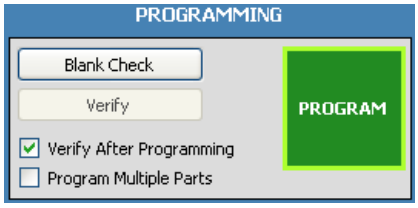


Figure 25. Blank Check

4. If device is “Programmed”, disable the 3.3 V supply, replace the device with a blank device and enable the 3.3 V supply.

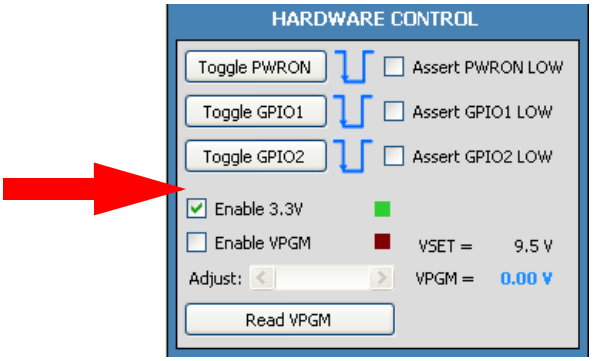


Figure 26. Disable 3.3 V Supply to Replace PF Device

5. Select the **Verify After Programming** option for automatic verification, or else verify manually after programming. Multiple parts can be programmed by selecting the **Program Multiple Parts** option. Click the **Program** button and wait until the part is programmed.

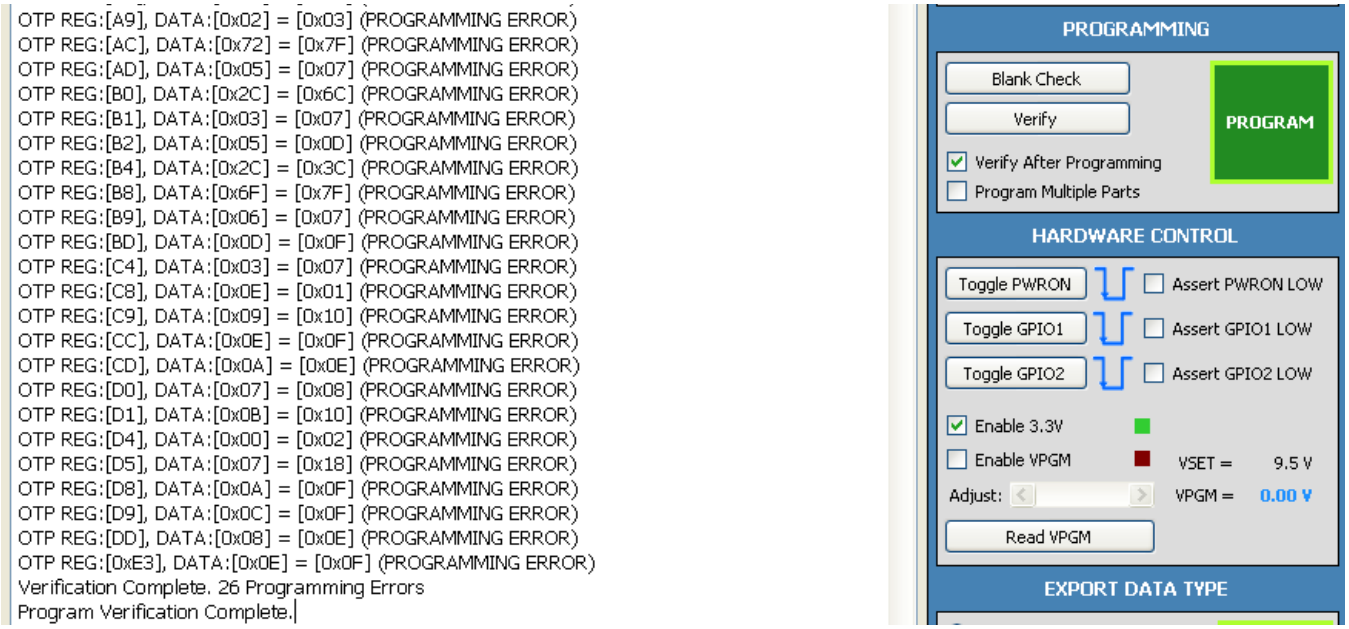


Figure 27. Program and Verify

5.9.2 Exporting Current Data to a File

To export the current configuration to a file, select the desired type of file from the **Export Data Type** section in the **Programming** tab and click **Export**. The data is ported on to the Script editor in the corresponding format ready to be saved. For more details on the Script Editor and script types, refer to section Using the Script Editor.

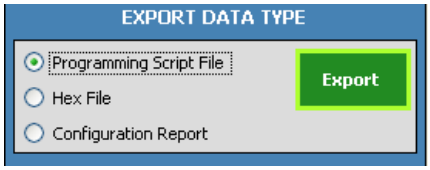


Figure 28. Export Data Type Section

5.10 Using the Script Editor

The Script Editor is a powerful tool that automates the development process when using a PF device. Scripts are groups of commands executed sequentially. They can quickly load PF device registers with your desired configuration, or they can help you determine the correct power up sequence for your design. Scripts are stored as simple text files, and as such, can be edited with any text editor. Since scripts are driven by your PC, PMIC configurations can be explored and validated prior to connecting to a host i.MX processor.

The Script Editor work area is shown in Figure 29. Script files are created in the large script area to the left side. The blank area in the right side is the Script Log, which displays the script output as it steps sequentially.

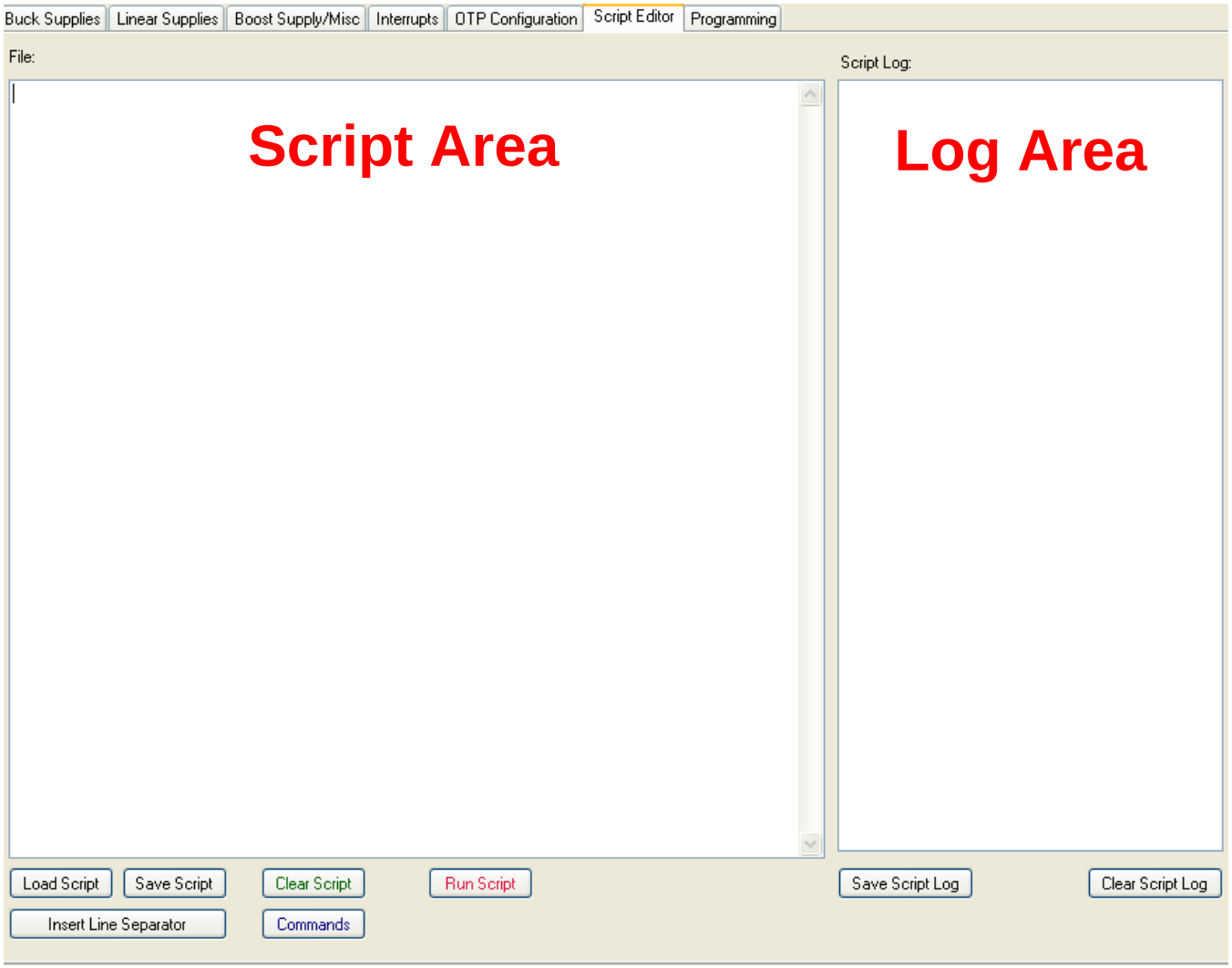


Figure 29. Script Editor Tab

The following list describes all the available buttons on the **Script Editor** tab.

- **Load Script:** Launches the File Load dialog box, allowing the user to select and load a stored script file.
- **Save Script:** Launches the File Save dialog box, allowing the user to save a script file to storage.
- **Clear Script:** Clears the current Script Editor work area to prepare for writing a new script.
- **Run Script:** Begins execution of the currently loaded script. Execution runs sequentially.
- **Insert Line Separator:** Inserts a comment at the current cursor position that represents a separating line. Used to organize long scripts.
- **Save Log:** Launches the File Save dialog box, allowing users to save the Script Log to a file.
- **Clear Log:** Clears the Script Log.
- **Commands**: Displays a pop-up window shown in Figure 33, with a graphical set of commands to add to the script.

5.10.1 Loading and Running a Script

To load a pre-existing script file, press the **Load Script** button. The **File Load** dialog box appears, allowing you to navigate to the directory where your script file is located. Select the file you want and click on the **Open** button.

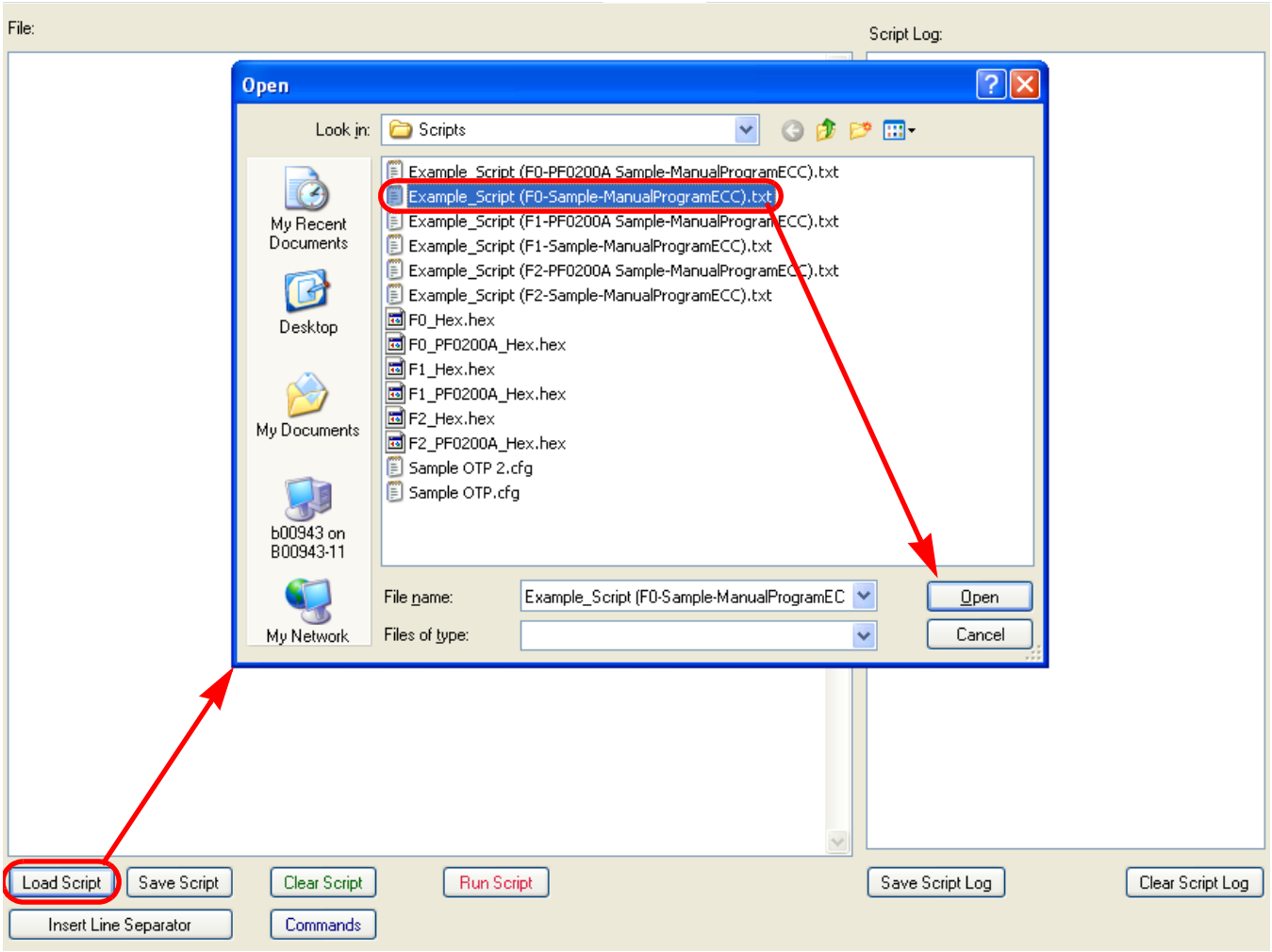


Figure 30. Loading a Script File

The Script Editor script area is now filled with the file content, and the file name appears next to the file label, and also as an entry in the Script Log.

Once the Script has been loaded, click the **Run Script** button to execute the script. As the script executes, each command appears sequentially in the Script Log. Commented lines with "//" are ignored during the script execution. When the script has completed, an entry in the Script Log is made, as shown in Figure 31.

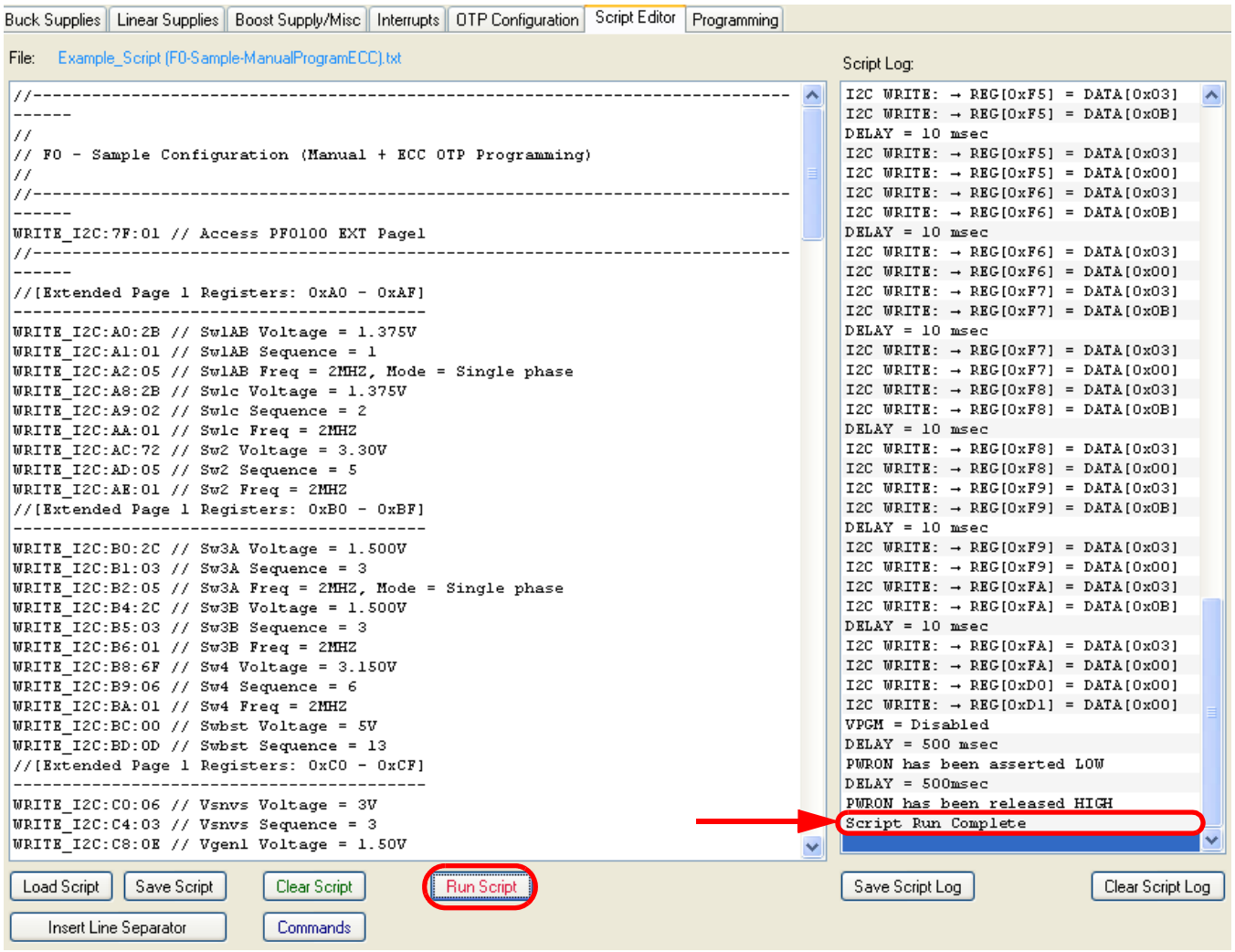


Figure 31. Running the Script

5.10.2 Writing a New Script

When writing a new script, it is recommended to use line separators to keep the code clean and organized. Create a comment header using the **Insert Line Separator** button.

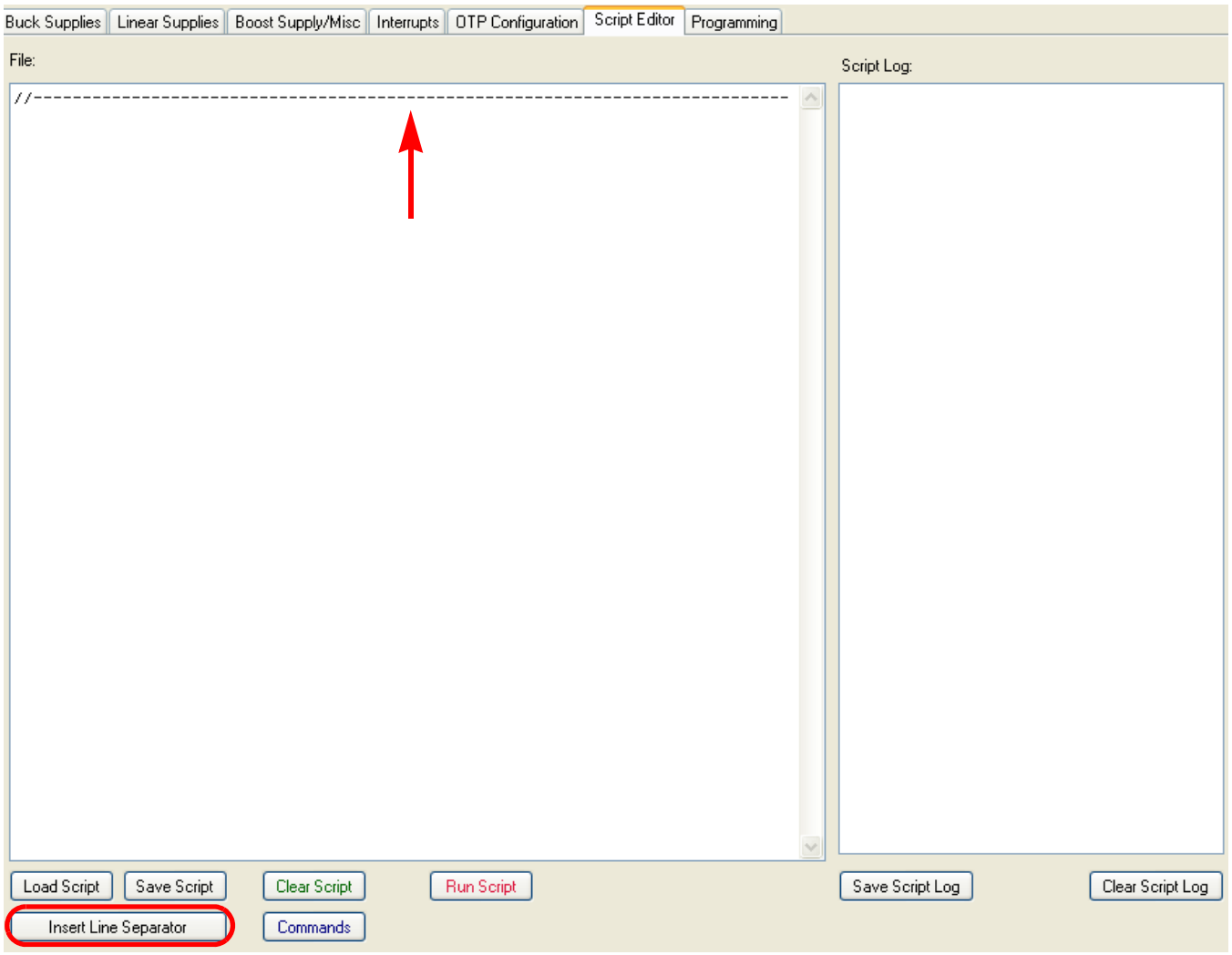


Figure 32. Inserting Line Separators

Proceed by manually writing the desired commands or use the **Command** button to display a graphical command selector in a new window, as shown in Figure 33.

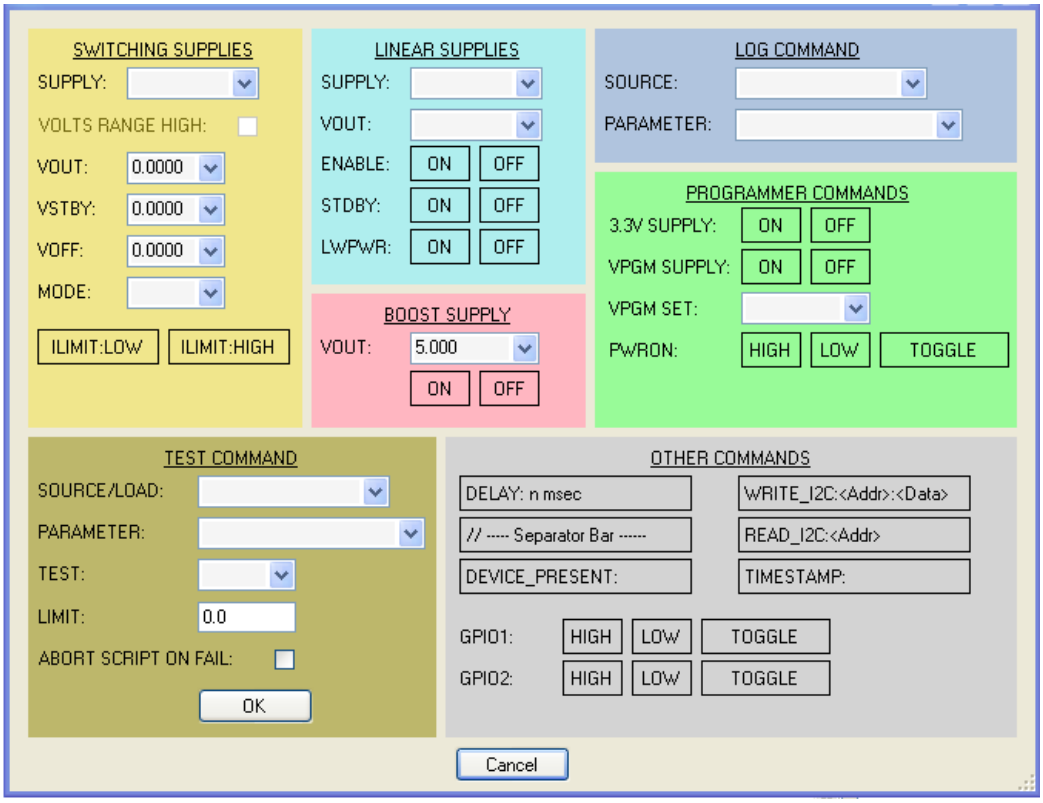


Figure 33. Command Selector Window

The **Command** window contains a set of commands useful for automatically sequencing the PF device power supplies, thereby emulating system behavior. The **Command** window is organized in six sections:

- PF Device - Switching Supplies> Place a single script command with the selected Buck regulator and the desired function. The available functions are: mode selection, normal operation, Standby and OFF voltage setpoint, and current limit levels selection. Note that for SW2, SW3A/B and SW4, the **Volts Range High** box reflects the operating voltage range bit in the OTP configuration at power up. If working in the lower voltage range, box is unchecked.
- PF Device - Linear Supplies> Place a single script command with the selected LDO regulator and the desired function. The available functions are: operating voltage setpoint, enable and disable outputs in normal operation or Standby operation, and enable or disable the low power mode.
- PF Device - SWBST> Allows changing the operating voltage and enabling and disabling the SWBST output.
- Programmer Commands> Allows to control the 3.3 V supply, the ~8.5 V boost supply for OTP programming, as well as control the PWRON pin to trigger a Power-on event.
- Other Commands> Provides access to common instructions initiated by the control MCU. The possible commands include delay, add separator bar, generic I^2C write/read, and toggle GPIO1 and GPIO2.
- Log Command> Provides a log report of the actual status of a specific configuration on the PMIC. Syntax for the log commands are shown in Table 13.
- Test Command> Provides logic comparators to test the input/output voltage of the 8.5~ regulator in the PF development tool (VUSB/VPGM), as well as a logic True/False comparison with the interrupts in the PF device.

5.10.2.1 Syntax and Command Set

Delimiters

- ':' - Is used as a separator
- '//' - Anything after a '//' is ignored
- White spaces are truncated

Table 13. Command List[9]

Command	Description
Hardware Control Commands	
WRITE_I2C:<Addr>:<Data>	Sends <Data> to I^2C register <Addr>[10]
READ_I2C:<Addr>	Reads the value of <Addr> and displays it in the Script Log [10]
VPGM:ON	Enables the 8.0 V OTP programming supply
VPGM:OFF	Disables the 8.0 V OTP programming supply
VPGM:VSET:<V>	Sets the OTP programming voltage (VPGM) to the selected <V> value.
V3V3:ON	Enables the 3.3 V system supply
V3V3:OFF	Disables the 3.3 V system supply
PWRON:HIGH	Releases the PWRON signal to a high-impedance state, allowing the PF0100 to start up
PWRON:LOW	Asserts the PWRON signal Low, forcing the PF0100 to shutdown
PWRON:TOGGLE	Asserts the PWRON signal Low, and then releases it to a high-impedance state, generating a power on event on the PF0100
DELAY:<value>	Adds delay between script commands. Note that delays are cumulative with the Script Delay set on the Editor. Delay is set in ms.
GPIO1:HIGH	Releases the GPIO1 signal to a high-impedance state
GPIO1:LOW	Asserts the GPIO1 signal Low
GPIO1:TOGGLE	Asserts the GPIO1 signal LOW, and then releases it to a high-impedance state
GPIO2:HIGH	Releases the GPIO2 signal to a high-impedance state
GPIO2:LOW	Asserts the GPIO2 signal LOW
GPIO2:TOGGLE	Asserts the GPIO2 signal LOW, and then releases it to a high-impedance state
DEVICE_PRESENT:	Verify the presence of a PF device in the I^2C Bus
TIMESTAMP:	Prints out the current date and time

Table 13. Command List[9] (continued)

Command	Description
Switching Supplies Commands	
SW1x:MODE:<operator>	Sets the mode of operation of the SW1x regulator. The valid operators are as follows: • OFF • PFM • PWM • APS
SW1x:VOUT:<value>	Sets the SW1x output voltage in normal operation. Operating range from 0.300 V to 1.875 V in 0.025 V steps
SW1x:VSTBY:<value>	Sets the SW1x output voltage to the STANDBY mode. Operating range from 0.300 V to 1.875 V in 0.025 V steps.
SW1x:OFF:<value>	Sets the SW1x output voltage to the OFF Mode. Operating range from 0.300 V to 1.875 V in 0.025 V steps.
SW1x:ILIM:<operator>	Sets the SW1x current limit level low or high. Valid operators: • LOW • HIGH
SWx:MODE:<operator>	Sets the mode of operation of the SWx regulator. Following are valid operators: • OFF • PFM • PWM • APS
SWx:VOUT:<value>	Sets the SWx output voltage to normal operation. Full operating range from 0.300 V to 3.300 V divided into two operating ranges[11]: • Low voltage range > 0.300 V to 1.875 V in 0.025 V steps • High voltage range > 0.800 V to 3.300 V in 0.050 V steps
SWx:VSTBY:<value>	Sets the SWx output voltage to the STANDBY mode. Full operating range from 0.300 V to 3.300 V divided in two operating ranges[11]: • Low voltage range > 0.300 V to 1.875 V in 0.025 V steps • High voltage range > 0.800 V to 3.300 V in 0.050 V steps
SWx:OFF:<value>	Sets the SWx output voltage to the OFF mode. Full operating range from 0.300 V to 3.300 V divided into two operating ranges[11]: • Low voltage range > 0.300 V to 1.875 V in 0.025 V steps • High voltage range > 0.800 V to 3.300 V in 0.050 V steps
SWx:ILIM:<operator>	Sets the SWx current limit level low or high. Valid operators: • LOW • HIGH

Table 13. Command List[9] (continued)

Command	Description
SWBST:VOUT:<value>	Sets the output voltage of the SWBST regulator. Valid output voltage: • 5.000 • 5.050 • 5.100 • 5.150
SWBST:ON	Enables SWBST regulator
SWBST:OFF	Disables SWBST regulator
Linear Supply Commands	
VGENx:ON	Enables the VGENx supply
VGENx:OFF	Disables the VGENx supply
VGENx:STBY:<operator>	Enables the VGENx supply to stay ON or OFF during Standby mode. Valid operators: • ON • OFF
VGENx:LOWPWR:<operator>	Enables the low power bit for VGENx supply. Valid operators: • ON • OFF
VGENx:VOUT:<value>	Sets the output voltage for VGENx supply. • VGEN1/2 operating range: 0.800 V to 1.550 V with 50 mV steps • VGEN3/4/5/6 operating range: 1.800 V to 3.3 V with 100 mV steps
VREFDDR:ON	Enables the VREFDDR supply
VREFDDR:OFF	Disables the VREFDDR supply
VSNVS:ON	Enables the VSNVS supply
VSNVS:OFF	Disables the VSNVS supply

Table 13. Command List[9] (continued)

Command	Description
LOG Commands	
LOG:SWx:<log operator>	Shows the current value of the <log operator> for the SWx regulator. Log operators: • VOUT = Output voltage in normal operation • VSTBY = Output voltage in STANDBY mode • VOFF = Output voltage in OFF mode. • MODE_REGISTER = Returns SWxMODE register value • MODE_NORM = Switching mode in normal operation • MODE_STBY = Switching mode in Standby operation • OMODE = Regulator status during OFF mode • CONFIG_REGISTER = Returns the SWxCONF register value. • FSW = Regulator Switching frequency • PHASE = Regulator phase • DVS = DVS speed set • ILIMIT = Current limit enabled or disabled • OTP_VOUT = Default power up voltage set through OTP • OTP_SEQUENCE = Default power up sequence of regulator • OTP_CONFIG_REGISTER = Returns the OTP_SWx_CONF register value • OTP_SWCONFIG= hardware configuration on SW1x and SW3x • OTP_FSW = Switching Frequency set on OTP registers
LOG:VGENx:<log operator>	Shows the current value of the <log operator> for the VGENx regulator. Log operators: • VOUT = Output voltage. • ENABLE = supply is ENABLED/DISABLED • STBY = Regulator status during Standby operation • LOWPWR = Low power mode enable or disabled • OTP_VOUT = Default power up voltage set through OTP • OTP_SEQUENCE = Default power up sequence of regulator
LOG:VSWBST:<log operator>	Shows the current value of the <log operator> for the VSWBST regulator. Log operators: • CONTROL_REGISTER = Returns the SWBSTCTL register value • VOUT = Output voltage • MODE_NORM = Regulator ON/OFF status in normal operation • MODE_STBY = Regulator ON/OFF status in Standby operation • OTP_VOUT = Default power up voltage set through OTP • OTP_SEQUENCE = Default power up sequence of regulator
LOG:VREFDDR:<log operator>	Shows the current value of the <log operator> for the VREFDDR regulator. Log operators: • ENABLE = supply is ENABLED/DISABLED • OTP_SEQUENCE = Default power up sequence of regulator
LOG:VSNVS:<log operator>	Shows the current value of the <log operator> for the VSNVS regulator. Log operators: • VOUT = Output voltage • OTP_VOUT = Default power up voltage set through OTP

Table 13. Command List[9] (continued)

Command	Description
LOG:OTP_PU_CONFIG:<log operator>	Shows the current value set as default by OTP. • SEQ_CLK_SPEED = programmed power up sequencing speed • DVS_CLK_SPEED = programmed DVS speed • PWRON_MODE = programmed PWRON pin active level • PGOOD_ENABLE = Power good mode is on/off
LOG:INT:<Int operator>	Shows the status of the corresponding interrupt bit. Interrupt operators are as follows: • 110_DEGREES • 120_DEGREES • 125_DEGREES • 130_DEGREES • SW1A_OVERCURRENT • SW1C_OVERCURRENT[12] • SW2_OVERCURRENT • SW3A_OVERCURRENT • SW3B_OVERCURRENT • SW4_OVERCURRENT[12] • SWBST_OVERCURRENT • VGEN1_OVERCURRENT • VGEN2_OVERCURRENT • VGEN3_OVERCURRENT • VGEN4_OVERCURRENT • VGEN5_OVERCURRENT • VGEN6_OVERCURRENT
LOG:USB:VOUT	Shows the voltage at the input of the VPGM boost regulator [13]
LOG:VPGM:VOUT	Shows the voltage of the VPGME output

Notes:

9. All characters have to be entered in uppercase.
10. The register and data values should be entered as hexadecimal numbers, for example: 0x20 is entered as 20.
11. The output voltage operating range is set during OTP programming and cannot be changed under normal PMIC control.
12. Only on PF0100 devices.
13. Available only on KITPFPGMEVME programming device.

5.10.3 Saving a script file

To save the script file, press the “Save Script” Button, then the Save File dialog box appears. Enter the script file name, including the .txt file extension, then click the **Save** button.

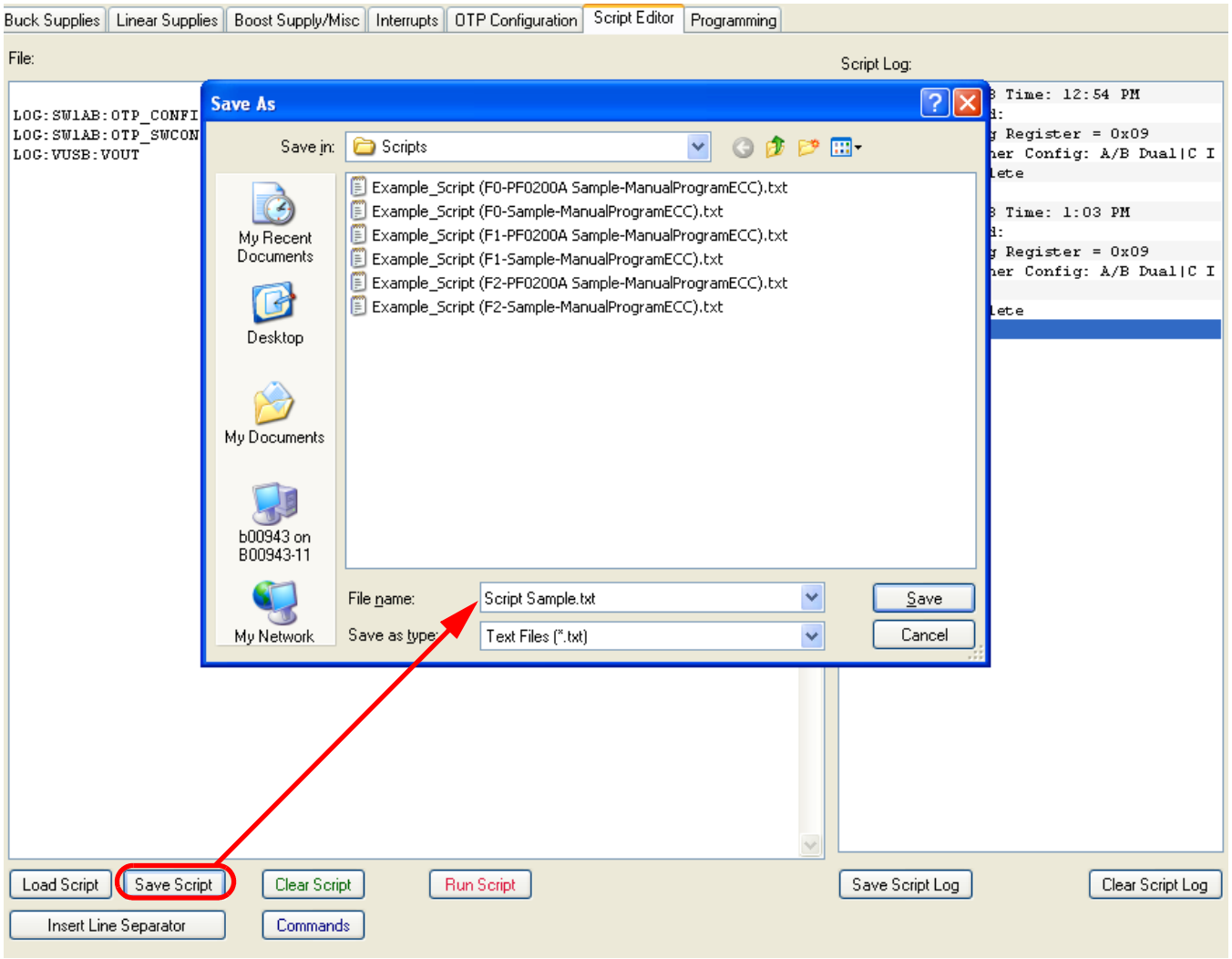


Figure 34. Saving a Script

5.10.4 Types of Files used on the KITPFGUI 4.0

The KITPFGUI 4.0 recognizes four types of file listed below:

1. Script file (.txt)
2. Hex configuration file (.hex)
3. Configuration file (.cfg)
4. Configuration report (.txt)

5.10.4.1 Script File

Scripts are groups of commands executed sequentially. They can quickly load PF device registers with your desired configuration, or they can help you determine the correct power up sequence for your design. Scripts are stored as simple text files, and as such, can be edited with any text editor. In the KITPFGUI 4.0 the script file is generated using the Script Editor and the commands explained in section Syntax and Command Set.

The following code shows an example of the structure of a script file. Note that it is not necessary to follow the same structure, but is recommended to use separator and commenting within the code to keep a clean and organized file for future reference.

```
//-------------------------------------------------------------------------
//                        PF0100 Sample Script
//-------------------------------------------------------------------------


PWRON:TOGGLE
DEVICE_PRESENT:      // Check for device connected


LOG:SW1AB:VOUT       // Display SW1AB output voltage
SW1AB:VOUT:1.0000    // Change SW1AB output voltage to 1.000V
LOG:SW1AB:VOUT       // Display SW1AB output voltage
SW1AB:VOUT:1.5000    // Change SW1AB output voltage to 1.500V
LOG:SW1AB:VOUT       // Display SW1AB output voltage

DELAY:500            // Add a 500ms Delay
TIMESTAMP:           // Display the DAte and time of execution of the script

//   END OF SCRIPT
//-------------------------------------------------------------------------
```

5.10.4.2 Hex Configuration File

The Hex configuration file use the extension .hex, and it contains the configuration coded packed in the standard Intel HEX format. Each line contains hexadecimal values encoding a sequence of data and their starting offset or absolute address.

Each line of Intel HEX file consists of six parts:

- **Start code**, one character, an ASCII colon ':'
- **Byte count**, two hex digits, a number of bytes (hex digit pairs) in the data field. 16 (0x10) or 32 (0x20) bytes of data are the usual compromise values between line length and address overhead.
- **Address**, four hex digits, a 16-bit address of the beginning of the memory position for the data. Limited to 64 kilobytes, the limit is worked around by specifying higher bits via additional record types.
- **Record type**, two hex digits, 00 to 05, defining the type of the data field.
- **Data**, a sequence of n bytes of the data themselves, represented by 2n hex digits.
- **Checksum**, two hex digits - the least significant byte of the two's complement of the sum of the values of all fields except fields 1 and 6 (Start code ":" byte and two hex digits of the Checksum).

Since this file is meant to be used as a way to store the OTP configuration values, the KITPFGUI 4.0 starts allocating the register data starting on address line 0x00A0 and uses the record type 0x00 (data record) to store the information related to the OTP configuration.

The code below shows a sample of the Hex configuration file for a PF device. The OTP register data is stored in the blue portion of the file.

```
:10000000000000000000000000000000000000000000F0
:10001000000000000000000000000000000000000000E0
:10002000000000000000000000000000000000000000D0
:10003000000000000000000000000000000000000000C0
:10004000000000000000000000000000000000000000B0
:10005000000000000000000000000000000000000000A0
:1000600000000000000000000000000000000000000090
:1000700000000000000000000000000000000000000080
:1000800000000000000000000000000000000000000070
:1000900000000000000000000000000000000000000060
:1000A0002B01050000000002B0201007205010079
:1000B0002C0305002C0301006F060100000D0000059
:1000C000060000000300000000E0900000E0A0000F8
:1000D000070B0000000700000A0C00000F080000DA
:1000E0000E0E0E000202020000000000000000000E0
:1000F0001F1F000000000000000000000000008BA
:00000001FF
```

5.10.4.3 Configuration file

This type of file is generated in the **OTP Configuration** tab during TBB mode, it comprises an identifier of the PF device, and a sequential array of all register values for the OTP configuration starting on address 0xA0 to address 0xFF. It is used to load predefine data in the **OTP Configuration** tab during TBB mode.

The following code presents an example of the contents of a .cfg file created by the KITPFGUI 4.0.

```
DEVICE:PF0100A
ADDR:A0:DATA:2D
ADDR:A1:DATA:02
ADDR:A2:DATA:2D
ADDR:A8:DATA:39
ADDR:A9:DATA:03
ADDR:AA:DATA:39
ADDR:AC:DATA:7F
ADDR:AD:DATA:02
ADDR:AE:DATA:7F
ADDR:B0:DATA:6C
ADDR:B1:DATA:07
ADDR:B2:DATA:6C
ADDR:B4:DATA:18
ADDR:B5:DATA:11
ADDR:B6:DATA:18
ADDR:B8:DATA:74
ADDR:B9:DATA:07
ADDR:BA:DATA:74
ADDR:BC:DATA:00
ADDR:BD:DATA:06
ADDR:C0:DATA:06
ADDR:C4:DATA:10
ADDR:C8:DATA:08
ADDR:C9:DATA:06
ADDR:CC:DATA:0C
ADDR:CD:DATA:06
ADDR:D0:DATA:04
ADDR:D1:DATA:0E
ADDR:D4:DATA:0D
ADDR:D5:DATA:0E
ADDR:D8:DATA:0F
ADDR:D9:DATA:07
ADDR:DC:DATA:0D
ADDR:DD:DATA:06
ADDR:E0:DATA:01
ADDR:E8:DATA:00
ADDR:FF:DATA:08
```

5.10.4.4 Configuration Report

The configuration report is a plain .txt file used to display the OTP configuration in a readable format. It translates register addresses and data, into a meaningful value for each regulator and modifies the function in the PF device.

```
--------------------------------------------------------------------------------

    OTP Configuration Report
    Device: PF0200A

--------------------------------------------------------------------------------

SW1AB OTP Configuration:
------------------------
  Vout = 0.5750V
  Startup Sequence = 11
  Switcher Configuration = A/B Dual|C Independent
  Fsw = MHz

SW1C OTP Configuration:
-----------------------
  Vout = 0.5750V
  Startup Sequence = 11
  Fsw = MHz

SW2 OTP Configuration:
----------------------
  Vout = 0.6750V
  Startup Sequence = 11
  Fsw = MHz

SW3A OTP Configuration:
-----------------------
  Vout = 0.6750V
  Startup Sequence = 11
  Switcher Configuration = A/B Dual Phase
  Fsw = MHz

SW3B OTP Configuration:
-----------------------
  Vout = 0.6750V
  Startup Sequence = 11
  Fsw = MHz

SW4 OTP Configuration:
----------------------
  Vout = 0.6750V
  Startup Sequence = 11
  VTT Mode = Disabled|
  Fsw = MHz

SWBST OTP Configuration:
------------------------
  Vout = 5.150V
  Startup Sequence = 11
```

```
VSNVS OTP Configuration:
------------------------
  Vout = 1.300V

VREFDDR OTP Configuration:
--------------------------
  Startup Sequence = 11

VGEN1 OTP Configuration:
------------------------
  Vout = 1.3500V
  Startup Sequence = 11

VGEN2 OTP Configuration:
------------------------
  Vout = 1.3500V
  Startup Sequence = 11

VGEN3 OTP Configuration:
------------------------
  Vout = 2.9000V
  Startup Sequence = 11

VGEN4 OTP Configuration:
------------------------
  Vout = 2.9000V
  Startup Sequence = 11

VGEN5 OTP Configuration:
------------------------
  Vout = 2.9000V
  Startup Sequence = 11

VGEN6 OTP Configuration:
------------------------
  Vout = 2.9000V
  Startup Sequence = 11

Device OTP Configuration:
-------------------------
  DVS Clock Speed = 3.1250mV/µs
  Sequencer Clock Speed = 4.0msec
  PWRON Pin Configuration: 0:PWRON HIGH = ON, PWRON LOW = OFF
  Power Good = Enabled
  I2C Device Address = 0x0B
```

6 References

Document Number	Description	URL
KITPFGUI 4.0 Graphical User Interface		
KITPFGUI_4.0	Software Download	http://www.freescale.com/webapp/sps/download/license.jsp?colCode=KITPFGUI_4.0.ZIP
PF PMIC		
MMPF0100	Data Sheet	http://cache.freescale.com/files/analog/doc/data_sheet/MMPF0100.pdf
MMPF0100ER	Errata	http://cache.freescale.com/files/analog/doc/errata/MMPF0100ER.pdf
MMPF0200	Datasheet	Page under development
MMPF0200ER	Errata	Page under development
PFSERIESFS	Fact Sheet	http://cache.freescale.com/files/analog/doc/fact_sheet/PFSeriesFS.pdf
AN4622	Layout Application Note	http://cache.freescale.com/files/analog/doc/app_note/AN4622.pdf
	Product Summary Page	http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMPF0100
Development Tools		
KITPFPGMEVME	PF Programming Device	http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=KITPFPGMEVME
KITPF0100SKTEVBE	PF Programming Socket	http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=KITPF0100SKTEVBE
KITPF0100EPEVBE	PF0100 Customer Evaluation Kit	http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=KITPF0100EPEVBE
KITPF0200EPEVBE	PF0200 Customer Evaluation Kit	Page under development
Other		
	Power Management Home Page	http://www.freescale.com/PMIC
	Analog Home Page	http://www.freescale.com/analog

6.1 Support

Visit Freescale.com/support for a list of phone numbers within your region.

6.2 Warranty

Visit Freescale.com/warranty for a list of phone numbers within your region.

7 Revision History

Revision	Date	Description of Changes
1.0	12/2013	• Initial Release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: KTPFSWUG4
Rev. 1.0
12/2013