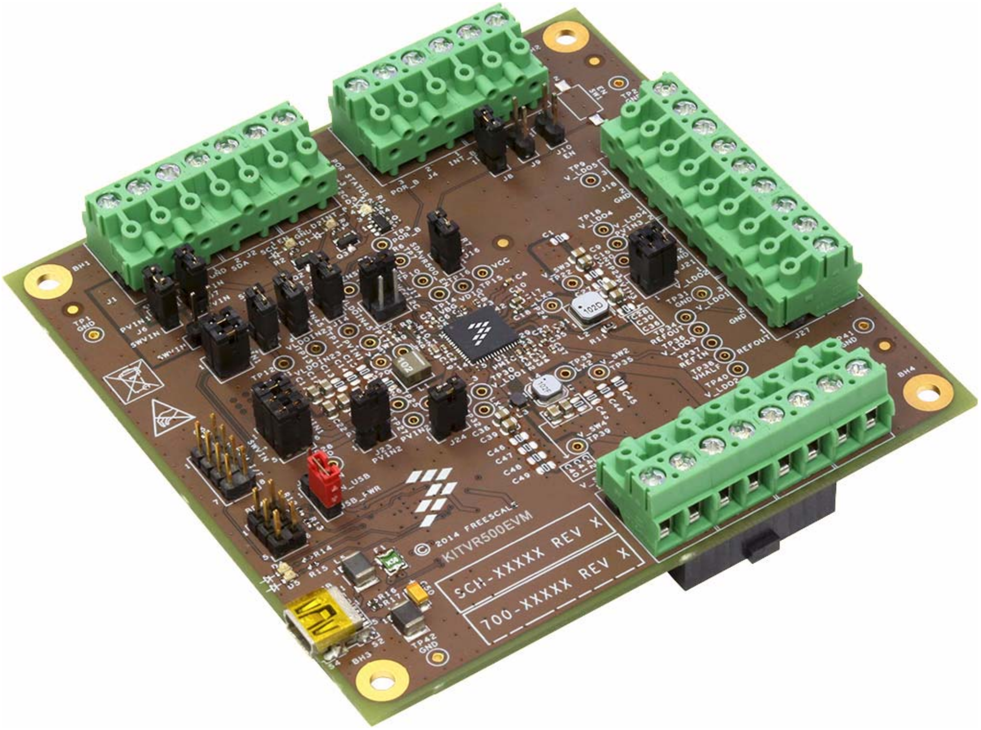
cale Semiconductor
User's Guide

Document Number: KTVR500SWUG
Rev. 1.0, 8/2014

KITVR500GUI Graphical User Interface



Contents

1 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.

2 Jump Start

Freescale's analog product development boards help to easily evaluate Freescale products. These tools support analog mixed signal and power solutions including monolithic ICs using proven high-volume SMARTMOS mixed signal technology, and system-in-package devices utilizing power, SMARTMOS, and MCU dies. Freescale products enable longer battery life, smaller form factor, component count reduction, ease of design, lower system cost and improved performance in powering state of the art systems.

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

  Jump Start Your Design

- Download documents, software and other information

Once the files are downloaded, review the user guide in the bundle. The user guide includes setup instructions, BOM, and schematics. Jump start bundles are available on each tool summary page with the most relevant and current information. The information includes everything needed for design.

3 Introduction

The KITVR500GUI 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 34VR500 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 KITVR500GUI is:

- KITVR500EVM

This document is intended to provide a detailed description of all the features of the KITVR500GUI, when operating any of the development tools mentioned above.

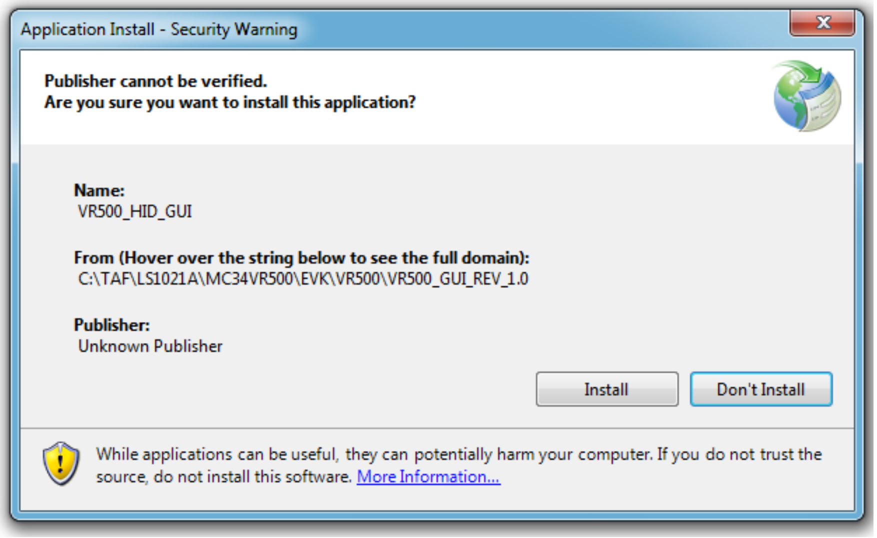
4 Software and Drivers Installation

4.1 Hardware Requirements

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

4.2 Installing the KITVR500GUI

1. Create a directory on a safe location of your PC. For example: C:\Freescale\KITVR500GUI
2. Extract the KITVR500GUI.zip file into that directory
3. Launch the “setup.exe” program
4. When the following pop-up dialog appears, press the “Install” button



Application Install - Security Warning

Publisher cannot be verified.
Are you sure you want to install this application?

Name:
VR500_HID_GUI

From (Hover over the string below to see the full domain):
C:\TAF\LS1021A\MC34VR500\EVK\VR500\VR500_GUI_REV_1.0

Publisher:
Unknown Publisher

While applications can be useful, they can potentially harm your computer. If you do not trust the source, do not install this software. More Information...

Figure 1. KITVR500GUI Installation Window

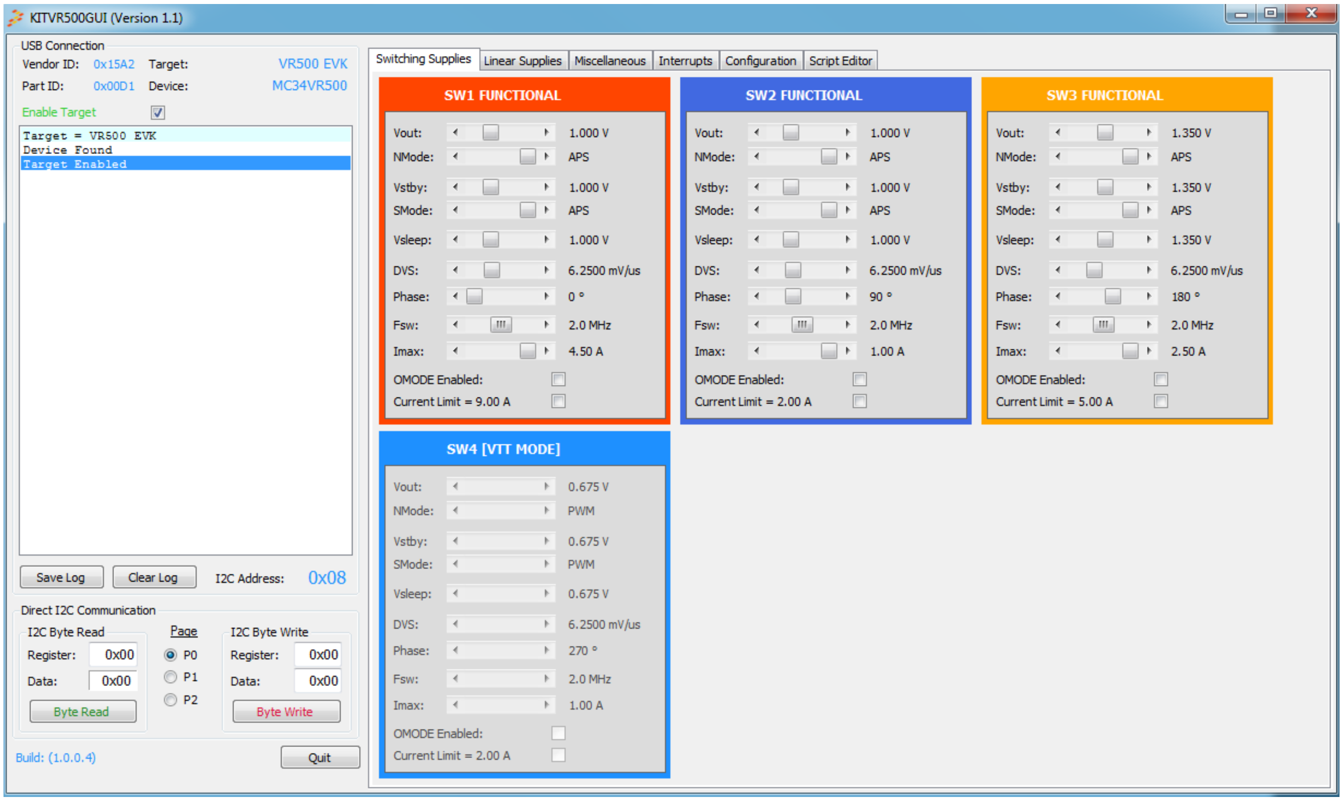
If the installation is successful, the application GUI displays the screen shown in Figure 2.

5 KITVR500GUI Description

The KITVR500GUI 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 34VR500 Power Management.

The main features of the KITVR500GUI are:

1. Automatic detection of the KITVR500EVM
2. Read/write access to the 34VR500 PMIC
3. Intuitive interface for controlling the 34VR500
4. Monitoring all interrupts manually or continuously
5. Scrip editor for prototyping, test emulation, or customized operation of the 34VR500 device
6. Saving and recalling customized script files



USB Connection

Vendor ID: 0x15A2 Target: VR500 EVK

Part ID: 0x00D1 Device: MC34VR500

Enable Target ☑

```
Target = VR500 EVK
Device Found
Target Enabled
```

I2C Address: 0x08

Direct I2C Communication

I2C Byte Read: Register: 0x00 Data: 0x00

Page: ● P0 ○ P1 ○ P2

I2C Byte Write: Register: 0x00 Data: 0x00

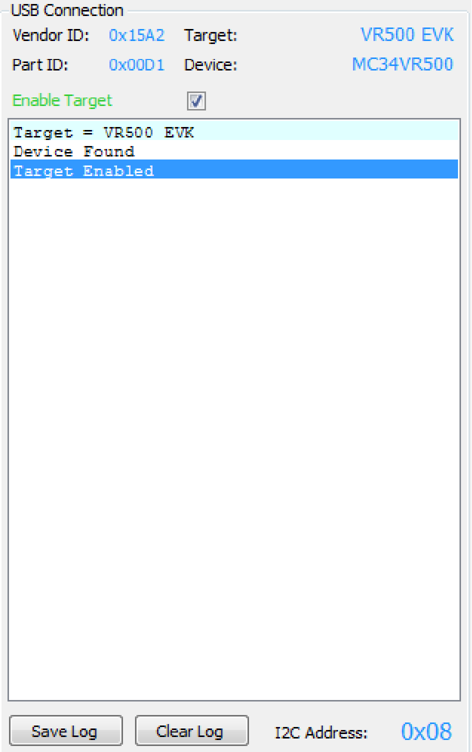
Build: (1.0.0.4)

	SW1 FUNCTIONAL	SW2 FUNCTIONAL	SW3 FUNCTIONAL	SW4 [VTT MODE]
Vout:	1.000 V	1.000 V	1.350 V	0.675 V
NMode:	APS	APS	APS	PWM
Vstby:	1.000 V	1.000 V	1.350 V	0.675 V
SMode:	APS	APS	APS	PWM
Vsleep:	1.000 V	1.000 V	1.350 V	0.675 V
DVS:	6.2500 mV/us	6.2500 mV/us	6.2500 mV/us	6.2500 mV/us
Phase:	0 °	90 °	180 °	270 °
Fsw:	2.0 MHz	2.0 MHz	2.0 MHz	2.0 MHz
Imax:	4.50 A	1.00 A	2.50 A	1.00 A
OMODE Enabled:	□	□	□	□
Current Limit	= 9.00 A □	= 2.00 A □	= 5.00 A □	= 2.00 A □

Figure 2. KITVR500GUI Graphical User Interface

5.1 USB Connection

Connect the device under control to the USB port and wait until the KITVR500GUI detects and recognizes the board connected as shown in Figure 3.


USB Connection

Vendor ID: 0x15A2 Target: VR500 EVK

Part ID: 0x00D1 Device: MC34VR500

Enable Target ☑

```
Target = VR500 EVK
Device Found
Target Enabled
```

I2C Address: 0x08

Figure 3. Device Detection and Recognition

Each Freescale 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 KITVR500GUI.

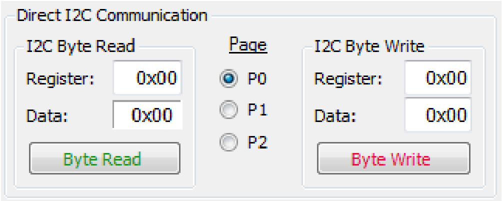
Table 1. Freescale PF Development Tool ID

Kit name	Hardware REV	Vendor ID	Part ID	Device
KITVR500EVM	Rev A	0x15A2	0x00D1	34VR500

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. By default, the KITVR500GUI enables the default address (0x08) used in all the standard part numbers provided by Freescale.

5.2 Single Data I^{2}C Communication

Use the “Byte Write” button to write one byte of data to any given register of the 34VR500 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 and P1 options to automatically switch to the specific page, just provide the correct address and data to write or read.



Direct I2C Communication

I2C Byte Read
Register: 0x00
Data: 0x00

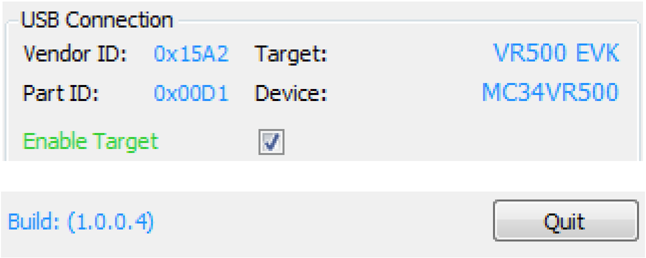
Page
● P0
○ P1
○ P2

I2C Byte Write
Register: 0x00
Data: 0x00

Figure 4. Single data Write/Read

5.3 Enabling the Target Board

To avoid false programming and undesired configuration loaded upon connection, the KITVR500GUI 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 KITVR500GUI to automatically load the current content of the 34VR500 device and fill in the configuration tabs accordingly. Note that the KITVR500GUI 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 KITVR500GUI, as well as the current software release version.



USB Connection
Vendor ID: 0x15A2
Target: VR500 EVK
Part ID: 0x00D1
Device: MC34VR500
Enable Target ☑

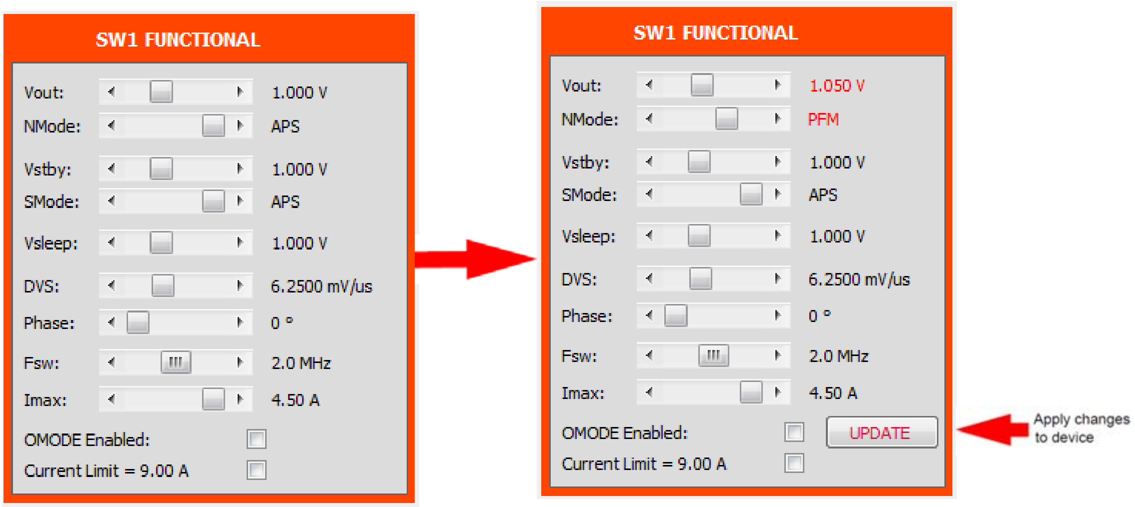
Build: (1.0.0.4)

Figure 5. General controls

5.4 Buck Supplies Control

The KITVR500GUI provides simplified access to the configuration parameter of all the Buck converters in the 34VR500 devices.

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 34VR500 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.



SW1 FUNCTIONAL

Vout: 1.000 V
NMode: APS
Vstby: 1.000 V
SMode: APS
Vsleep: 1.000 V
DVS: 6.2500 mV/us
Phase: 0 °
Fsw: 2.0 MHz
Imax: 4.50 A
OMODE Enabled: □
Current Limit = 9.00 A □

SW1 FUNCTIONAL

Vout: 1.050 V
NMode: PFM
Vstby: 1.000 V
SMode: APS
Vsleep: 1.000 V
DVS: 6.2500 mV/us
Phase: 0 °
Fsw: 2.0 MHz
Imax: 4.50 A
OMODE Enabled: □ UPDATE
Current Limit = 9.00 A □

Apply changes to device

Figure 6. Switching Regulator Control Box

Table 2 describes each one of the functions listed in the Buck regulator control box.

Table 2. SWx Control Box description.(1)

Function	Description
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]
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]
SMode	Sets the operating mode during standby operation. • Operating Modes: OFF, PWM, PFM and APS • **I²C Register:** SWxMODE • **I²C Bits Modified:** SWxMODE[3:0]
Vsleep	Sets the voltage of the regulator during Sleep operation. • Refer to device specification for all possible values • **I²C Register:** SWxOFF • **I²C Bits Modified:** SWxOFF[5:0]
DVS	Sets the DVS speed for the Buck regulator. • 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 • **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]

Table 2. SWx Control Box description.[1] (continued)

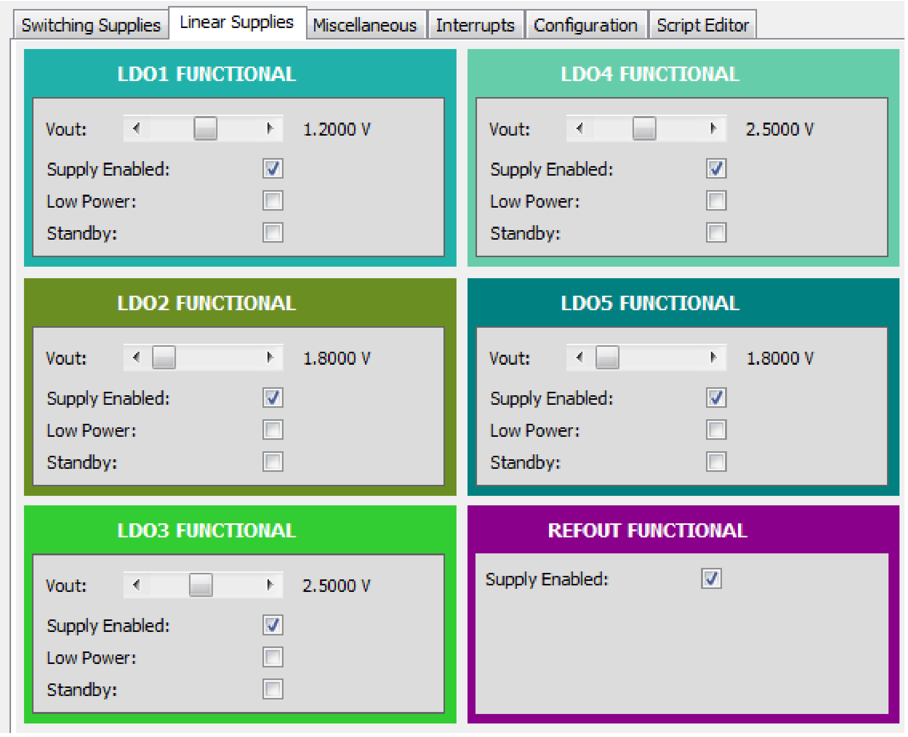
Function	Description
Fsw	Sets the operating frequency of the Buck regulator. • **Frequency selection:** 1.0 MHz, 2.0 MHz and 4.0 MHz • **I^{2}C Register:** SWxCONF • **I^{2}C Bits Modified:** SWxFREQ[1:0]
Imax	Programs the maximum operating current. • Refer to device specification for all possible values • **I^{2}C Register:** SWx PWRSTG - Extended Page 2 • **I^{2}C Bits Modified:** SWx_PWRSTG[2:0]
OMODE Enabled	Enable the part to operate in Sleep mode. • Check box option: • Unchecked = Sleep mode disabled • Checked = Sleep mode enabled • **I^{2}C Register:** SWxMODE • **I^{2}C Bits Modified:** SWxOMODE
Current Limit	Select current limit level. • Check box option: • Unchecked = High level current limit • Checked = Low level current limit • **I^{2}C Register:** SWxCONF • **I^{2}C Bits Modified:** SWxILIM

Notes:

[1] "x" stands for the number of regulator to be configured. I.e., SW1, SW2, SW3, or SW4.

5.5 Linear Supplies Control

The KITVR500GUI provides simplified access to the configuration parameter of all the LDO supplies in the 34VR500 devices, as shown in Figure 7.



LDO1 FUNCTIONAL
Vout: 1.2000 V
Supply Enabled: ☑
Low Power: □
Standby: □

LDO4 FUNCTIONAL
Vout: 2.5000 V
Supply Enabled: ☑
Low Power: □
Standby: □

LDO2 FUNCTIONAL
Vout: 1.8000 V
Supply Enabled: ☑
Low Power: □
Standby: □

LDO5 FUNCTIONAL
Vout: 1.8000 V
Supply Enabled: ☑
Low Power: □
Standby: □

LDO3 FUNCTIONAL
Vout: 2.5000 V
Supply Enabled: ☑
Low Power: □
Standby: □

REFOUT FUNCTIONAL
Supply Enabled: ☑

Figure 7. LDO Control Box

Table 3 and Table 5 describe each one of the functions listed in the control boxes for all LDOs.

Table 3. LDOx Control Box description[2]

Function	Description
Vout	Sets the voltage of the regulator during normal operation. • Refer to device specification for all possible values • **I^2C Register:** LDOxCTL • **I^2C Bits Modified:** LDOx[3:0]
Supply Enabled	Enables or disables the linear regulator. • Check box option: • Unchecked = Disabled • Checked = Enabled. • **I^2C Register:** LDOxCTL • **I^2C Bits Modified:** LDOxEN
Low Power	Enables low power operation for the linear regulator.[3] • Check box option: • Unchecked = Low power mode disabled • Checked = Low power mode enabled • **I^2C Register:** LDOxCTL • **I^2C Bits Modified:** LDOxLPWR
Standby	Enables regulator during STANDBY operation[3]. • Check box option: • Unchecked = No regulator control during Standby • Checked = Regulator controlled on Standby • **I^2C Register:** LDOxCTL • **I^2C Bits Modified:** LDOxSTBY

Notes:

2. “x” stands for the number of regulators to be configured. I.e., LDO1, LDO2, LDO3, LDO4, LDO5.
3. See Table 4 for detail description of LDO functionality.

Table 4. LDO Control

LDOxEN	LDOxLPWR	LDOxSTBY	STANDBY [4]	LDOxOUT
0	X	X	X	Off
1	0	0	X	On
1	1	0	X	Low Power
1	X	1	0	On
1	0	1	0	Off
1	1	1	1	Low Power

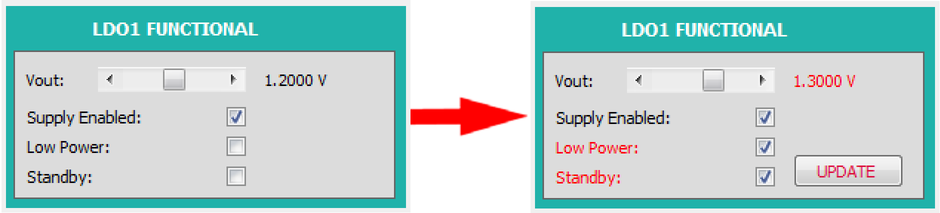
Notes:

4. STANDBY event is triggered by the STBY pin as describe in the 34VR500 datasheet.

Table 5. REFOUT Control Box description

Function	Description
REFOUT Control Box	
Supply Enabled	• Check box option: • Unchecked = REFOUT is off • Checked = REFOUT is on • **I^2C Register:** REFOUTCTRL • **I^2C Bits Modified:** REFOUTEN

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


LDO1 FUNCTIONAL

Vout: 1.2000 V

Supply Enabled: ☑

Low Power: □

Standby: □

LDO1 FUNCTIONAL

Vout: 1.3000 V

Supply Enabled: ☑

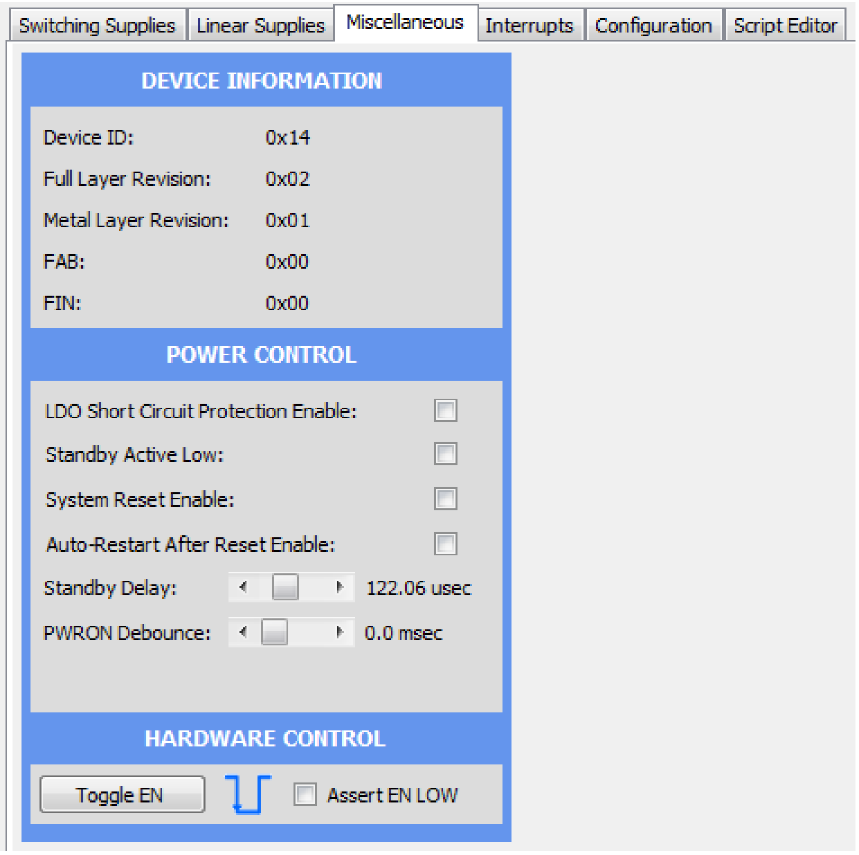
Low Power: ☑

Standby: ☑

Figure 8. LDO Control Box

5.6 Miscellaneous Control

The **Miscellaneous** tab provides access to various general registers on the 34VR500. Figure 9, shows the **Miscellaneous** tab.



DEVICE INFORMATION

Device ID: 0x14

Full Layer Revision: 0x02

Metal Layer Revision: 0x01

FAB: 0x00

FIN: 0x00

POWER CONTROL

LDO Short Circuit Protection Enable: □

Standby Active Low: □

System Reset Enable: □

Auto-Restart After Reset Enable: □

Standby Delay: 122.06 usec

PWRON Debounce: 0.0 msec

HARDWARE CONTROL

□ Assert EN LOW

Figure 9. Boost Supply/Misc Control Tab

Table 6 describes all functions within the Miscellaneous tab.

Table 6. Miscellaneous Control Box Description

Device Information (read only)	
Device ID	Returns the 34VR500 device ID.
Full Layer Revision	Returns the full layer revision for the 34VR500 device connected.
Metal Layer Revision	Returns the metal layer revision for the 34VR500 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 • **I^{2}C Register:** PWRCTL • **I^{2}C Bits Modified:** REGSCPEN[0]
Standby Active Low	Configures the operating level of STBY pin. • Check box option: • Unchecked = STBY pin is Active HIGH • Checked = STBY pin is Active LOW • **I^{2}C Register:** PWRCTL • **I^{2}C Bits Modified:** STBYINV[0]
System Reset Enable	Enables a “long press” on EN pin to reset the part. • Check box option: • Unchecked = No reset possible with EN pin • Checked = Reset enabled with long EN press • **I^{2}C Register:** PWRCTL • **I^{2}C Bits Modified:** PWRONRSTEN[0]
Auto-Restart After Reset Enable	Enables an auto-restart after a long EN press reset. • Check box option: • Unchecked = Auto -restart disabled • Checked = Auto-restart enabled • **I^{2}C Register:** PWRCTL • **I^{2}C Bits Modified:** RESTARTEN[0]
Standby Delay	Sets the delay to act upon a STBY event. • Delay times: • 91.55 µs • 122.06 µs • 152.59 µs • 183.10 µs • **I^{2}C Register:** PWRCTL • **I^{2}C Bits Modified:** STBYDLY[1:0]
EN Debounce	Sets the debounce time for EN 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 • **I^{2}C Register:** PWRCTL • **I^{2}C Bits Modified:** ENDBNC[1:0]
Toggle EN	Toggle the EN pin to restart the device

5.7 Interrupt Monitoring

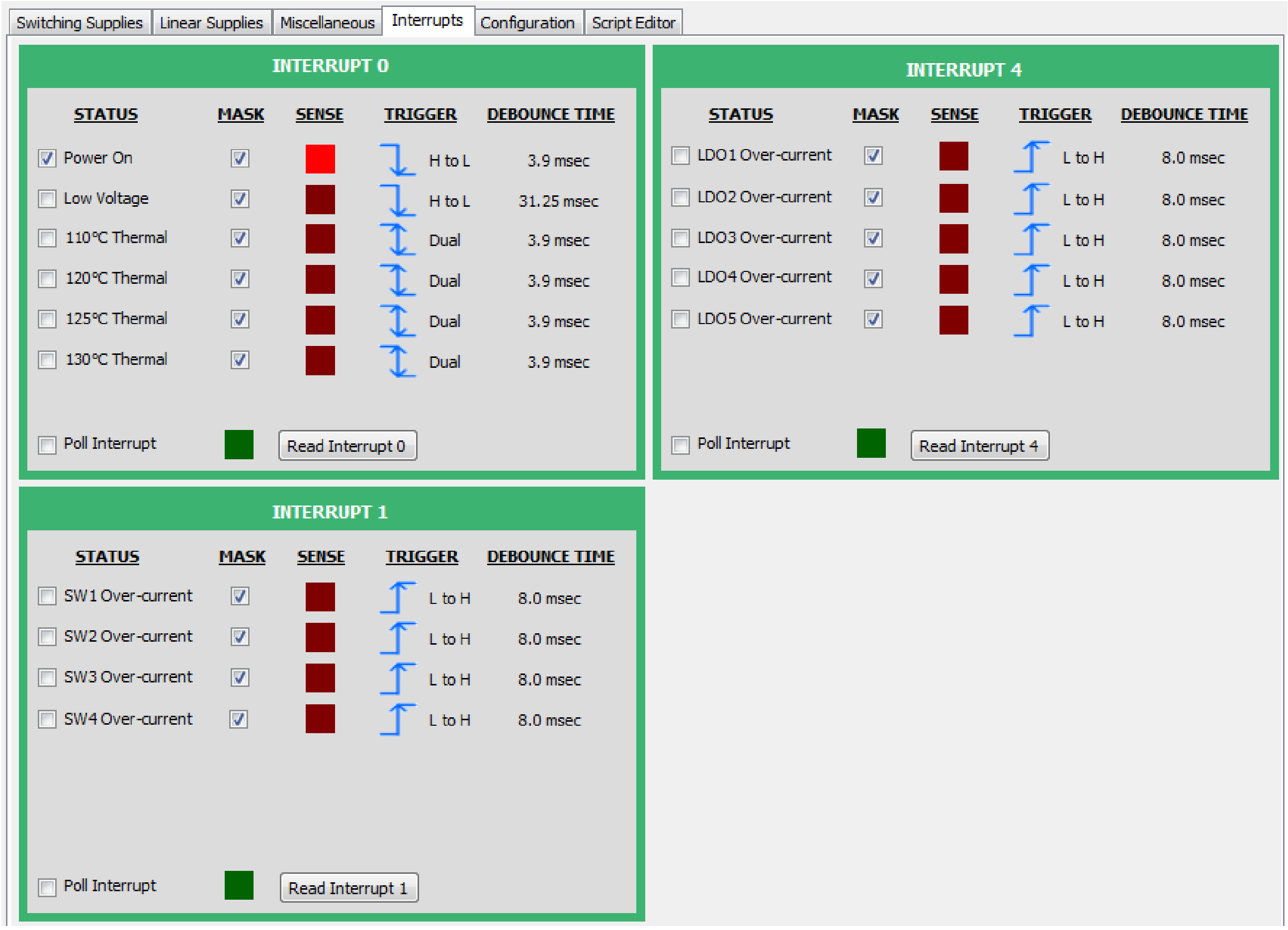
The **Interrupts** tab provides access to the three interrupt registers in the functional register map of the 34VR500. The user can choose to read the interrupts by pressing the **Read Interrupt x** button.

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 three interrupt control boxes contained in the **Interrupts** tab.



INTERRUPT 0

STATUS	MASK	SENSE	TRIGGER	DEBOUNCE TIME
☑ Power On	☑	🟥	H to L	3.9 msec
□ Low Voltage	☑	🟥	H to L	31.25 msec
□ 110°C Thermal	☑	🟥	Dual	3.9 msec
□ 120°C Thermal	☑	🟥	Dual	3.9 msec
□ 125°C Thermal	☑	🟥	Dual	3.9 msec
□ 130°C Thermal	☑	🟥	Dual	3.9 msec

□ Poll Interrupt 🟩 Read Interrupt 0

INTERRUPT 4

STATUS	MASK	SENSE	TRIGGER	DEBOUNCE TIME
□ LDO1 Over-current	☑	🟥	L to H	8.0 msec
□ LDO2 Over-current	☑	🟥	L to H	8.0 msec
□ LDO3 Over-current	☑	🟥	L to H	8.0 msec
□ LDO4 Over-current	☑	🟥	L to H	8.0 msec
□ LDO5 Over-current	☑	🟥	L to H	8.0 msec

□ Poll Interrupt 🟩 Read Interrupt 4

INTERRUPT 1

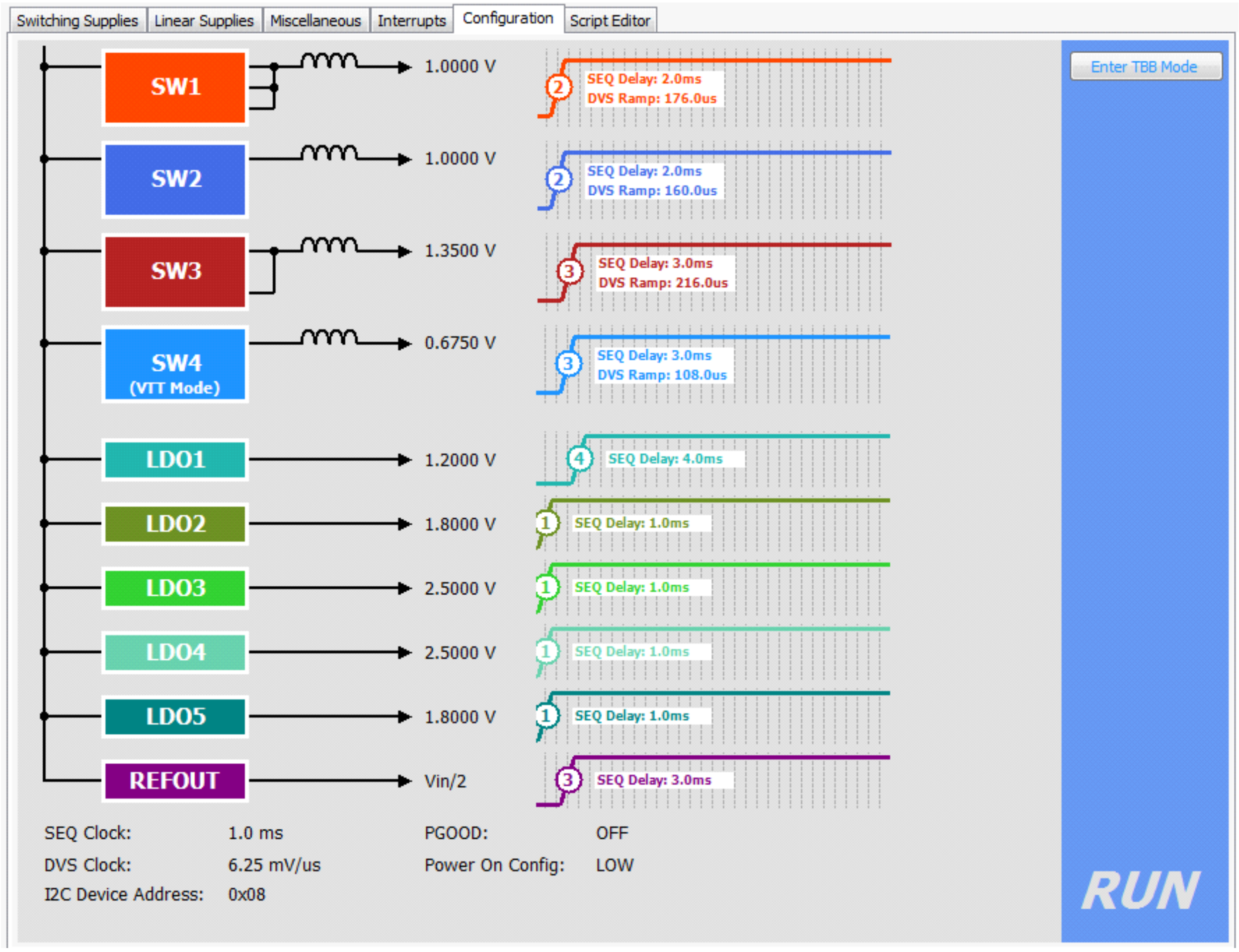
STATUS	MASK	SENSE	TRIGGER	DEBOUNCE TIME
□ SW1 Over-current	☑	🟥	L to H	8.0 msec
□ SW2 Over-current	☑	🟥	L to H	8.0 msec
□ SW3 Over-current	☑	🟥	L to H	8.0 msec
□ SW4 Over-current	☑	🟥	L to H	8.0 msec

□ Poll Interrupt 🟩 Read Interrupt 1

Figure 10. Interrupts Tab

5.8 Configuration Tab

The **Configuration** tab shows the user to graphically inspect and modify the Starting Sequence configuration in the 34VR500 device. By default, the KITVR500GUI is set to Run mode, therefore, the Starting Sequence configuration cannot be modified until the TBB mode is enabled. Figure 11 shows the complete view of the **Configuration** tab during Run mode.



SW1 1.0000 V SEQ Delay: 2.0ms DVS Ramp: 176.0us

SW2 1.0000 V SEQ Delay: 2.0ms DVS Ramp: 160.0us

SW3 1.3500 V SEQ Delay: 3.0ms DVS Ramp: 216.0us

SW4 (VTT Mode) 0.6750 V SEQ Delay: 3.0ms DVS Ramp: 108.0us

LDO1 1.2000 V SEQ Delay: 4.0ms

LDO2 1.8000 V SEQ Delay: 1.0ms

LDO3 2.5000 V SEQ Delay: 1.0ms

LDO4 2.5000 V SEQ Delay: 1.0ms

LDO5 1.8000 V SEQ Delay: 1.0ms

REFOUT Vin/2 SEQ Delay: 3.0ms

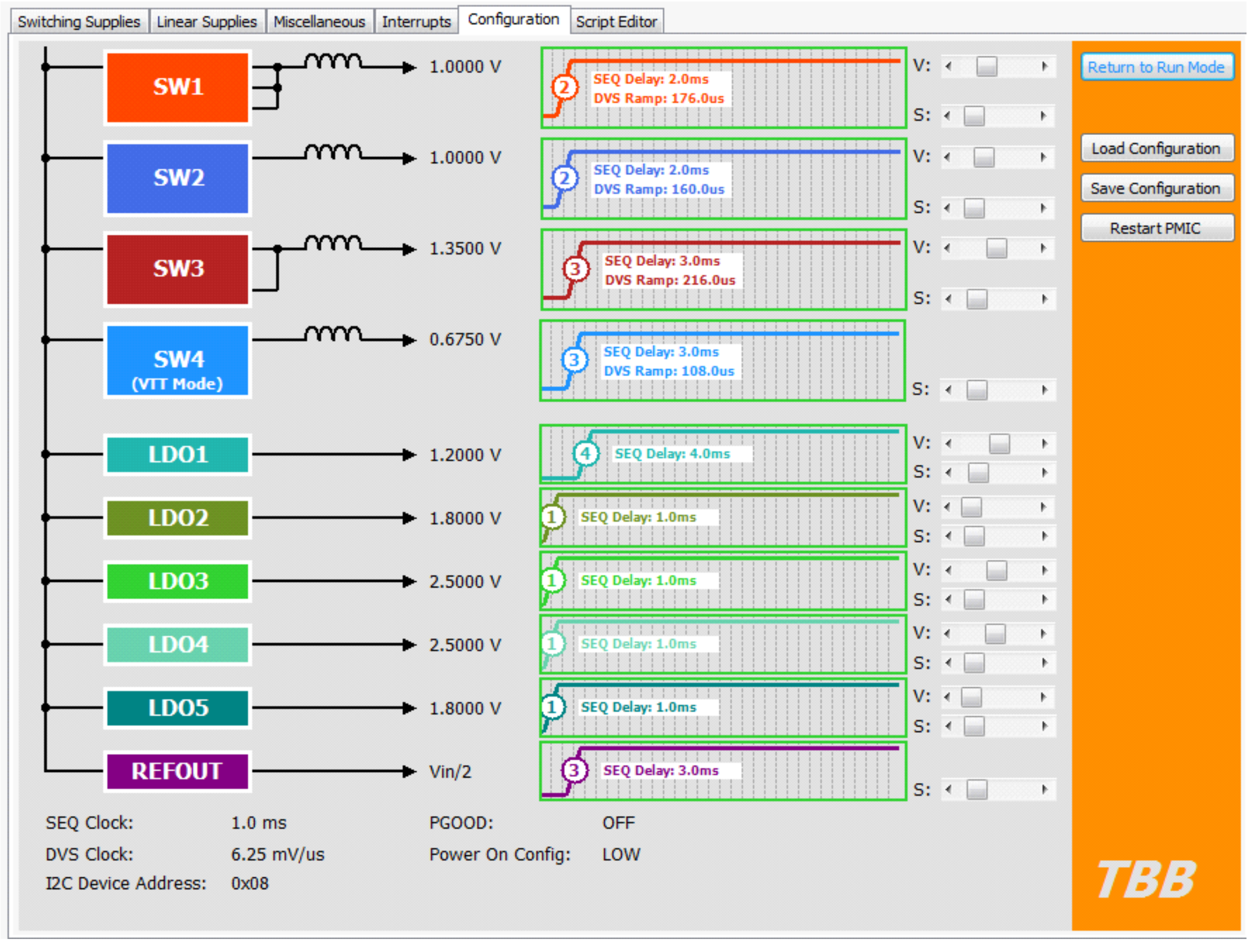
SEQ Clock: 1.0 ms
DVS Clock: 6.25 mV/us
I2C Device Address: 0x08

PGOOD: OFF
Power On Config: LOW

Figure 11. Configuration Tab in “Run Mode”

5.8.1 Enable TBB Mode

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



SW1 1.0000 V — 2 SEQ Delay: 2.0ms DVS Ramp: 176.0us

SW2 1.0000 V — 2 SEQ Delay: 2.0ms DVS Ramp: 160.0us

SW3 1.3500 V — 3 SEQ Delay: 3.0ms DVS Ramp: 216.0us

SW4 (VTT Mode) 0.6750 V — 3 SEQ Delay: 3.0ms DVS Ramp: 108.0us

LDO1 1.2000 V — 4 SEQ Delay: 4.0ms

LDO2 1.8000 V — 1 SEQ Delay: 1.0ms

LDO3 2.5000 V — 1 SEQ Delay: 1.0ms

LDO4 2.5000 V — 1 SEQ Delay: 1.0ms

LDO5 1.8000 V — 1 SEQ Delay: 1.0ms

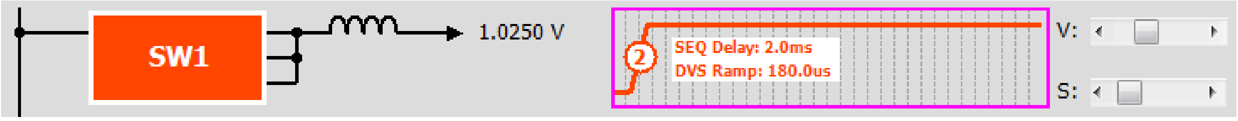
REFOUT Vin/2 — 3 SEQ Delay: 3.0ms

SEQ Clock: 1.0 ms
DVS Clock: 6.25 mV/us
I2C Device Address: 0x08
PGOOD: OFF
Power On Config: LOW

Figure 12. Configuration Tab in "TBB Mode"

- Manual Starting Sequence Configuration:

To manually modify the Starting Sequence Configuration, click on the specific arrow on the right of the regulator as shown in Figure 13.

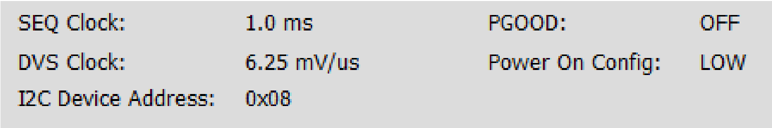


SW1 1.0250 V — 2 SEQ Delay: 2.0ms DVS Ramp: 180.0us

Figure 13. Modify the Starting Sequence

The V arrow corresponds to the Voltage Output of the selected regulator and the S arrow modifies the Start-up Sequence timing.

It’s not possible to modify the following parameters:

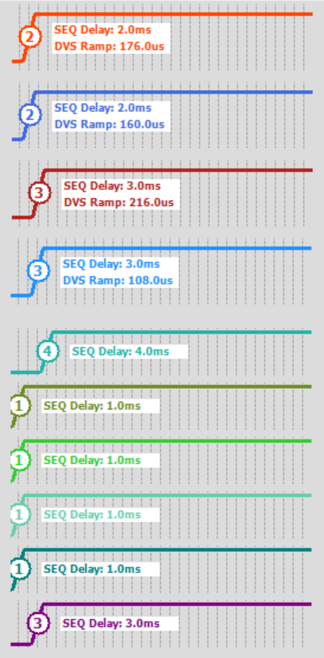


SEQ Clock:	1.0 ms	PGOOD:	OFF
DVS Clock:	6.25 mV/us	Power On Config:	LOW
I2C Device Address:	0x08		

Figure 14. Clock Generators and Power Controls

- Startup:

Figure 15 shows the graphical representation of the turn-on position for each regulator during the startup sequence.



2 SEQ Delay: 2.0ms
DVS Ramp: 176.0us

2 SEQ Delay: 2.0ms
DVS Ramp: 160.0us

3 SEQ Delay: 3.0ms
DVS Ramp: 216.0us

3 SEQ Delay: 3.0ms
DVS Ramp: 108.0us

4 SEQ Delay: 4.0ms

1 SEQ Delay: 1.0ms

1 SEQ Delay: 1.0ms

1 SEQ Delay: 1.0ms

1 SEQ Delay: 1.0ms

3 SEQ Delay: 3.0ms

Figure 15. Startup Sequence

- Save Configuration:

To save a Starting Sequence 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
- Load Configuration:

To load a Starting Sequence 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.
- Test a Starting Sequence Configuration:
  - Click on the Restart PMIC button. This toggles the EN pin forcing a power restart.

PMIC powers up with the new configuration.

5.9 Using the Script Editor

The Script Editor is a powerful tool that automates the development process when using a 34VR500 device. Scripts are groups of commands executed sequentially. They can quickly load 34VR500 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 processor.

The Script Editor work area is shown in Figure 16. 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.

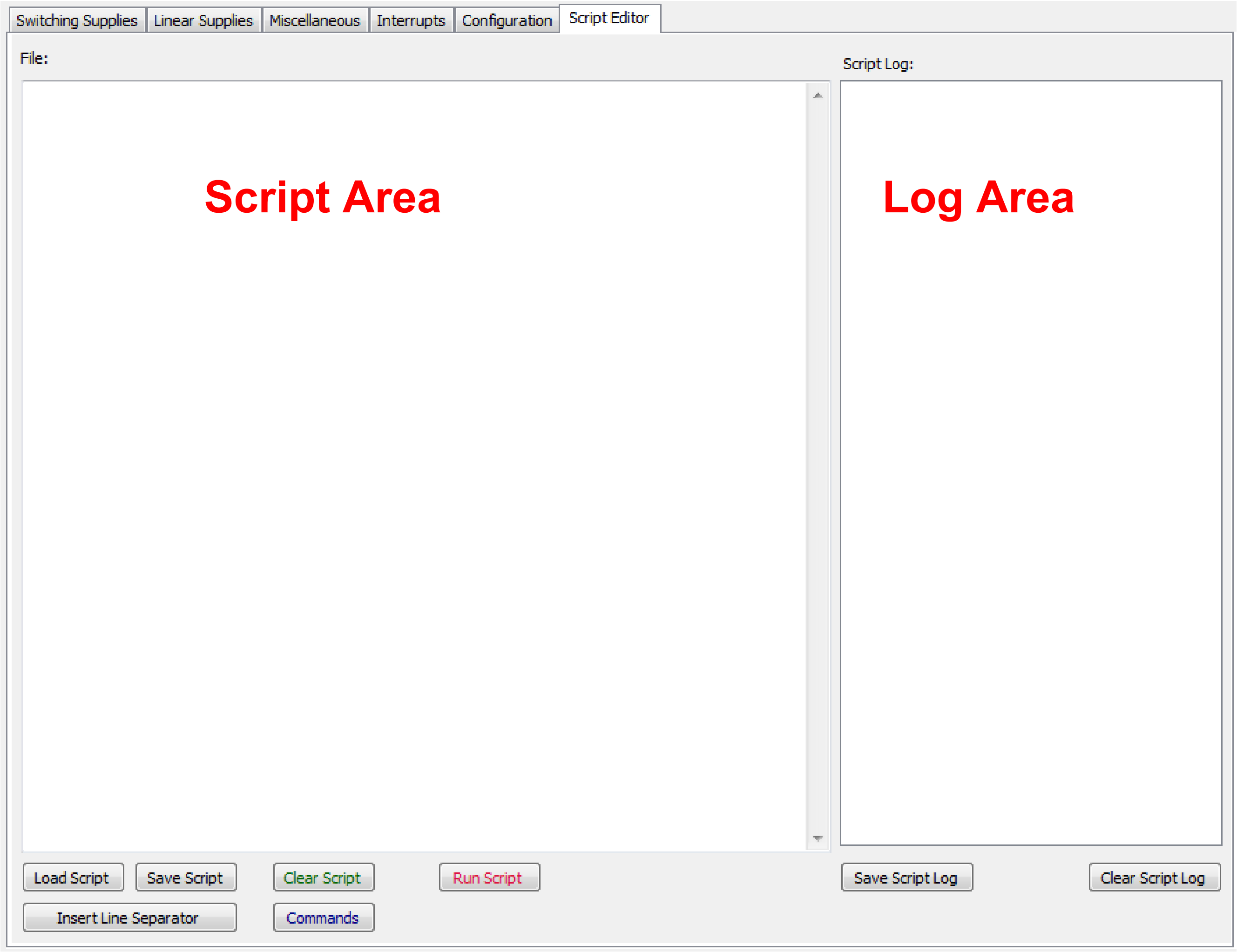


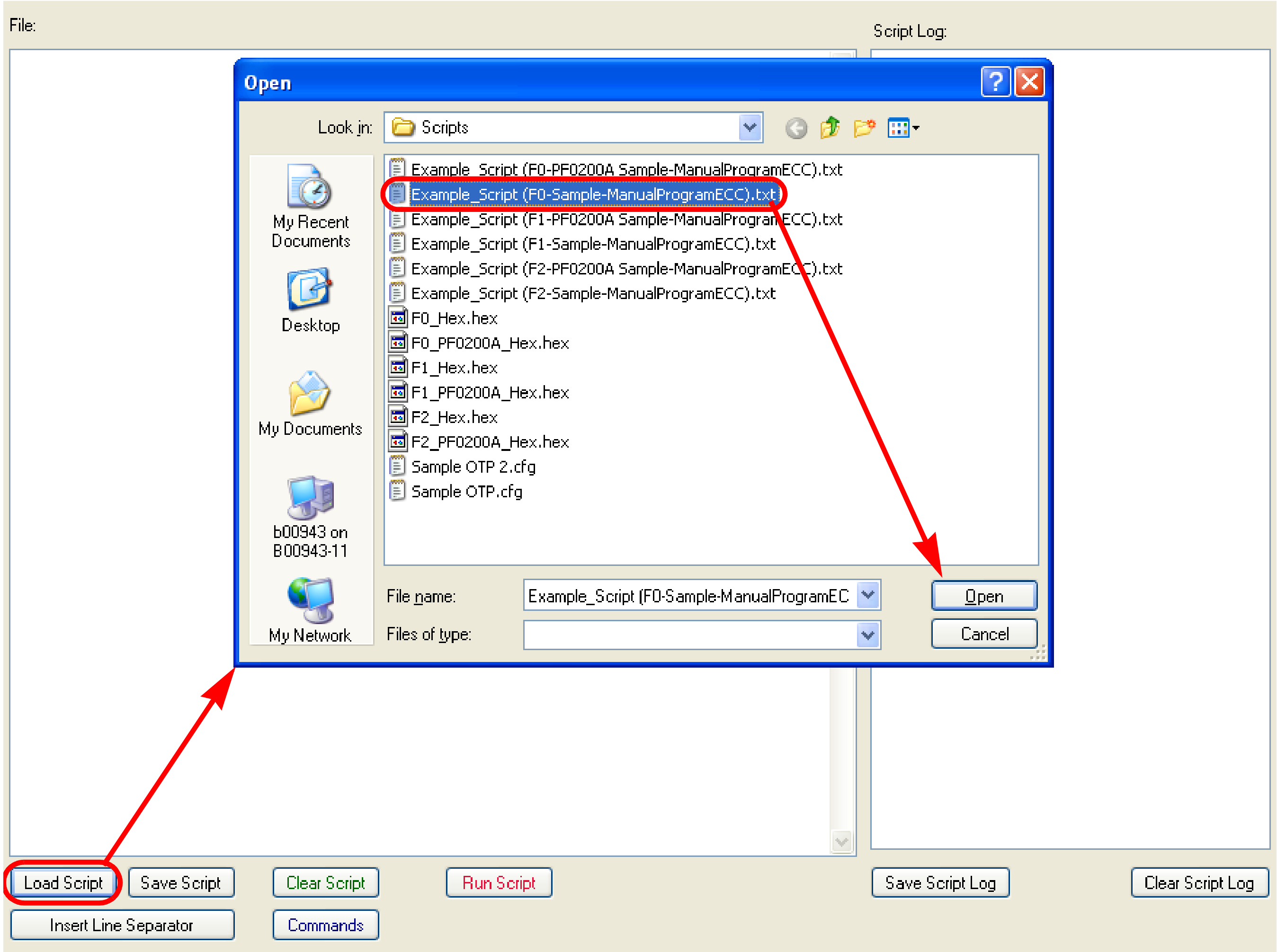
Figure 16. Script Editor Tab

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 Script Log:** Launches the File Save dialog box, allowing users to save the Script Log to a file.
- **Clear Script Log:** Clears the Script Log.
- **Commands**: Displays a pop-up window shown in Figure 20, with a graphical set of commands to add to the script.

5.9.1 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.



Example_Script (F0-PF0200A Sample-ManualProgramECC).txt
Example_Script (F0-Sample-ManualProgramECC).txt
Example_Script (F1-PF0200A Sample-ManualProgramECC).txt
Example_Script (F1-Sample-ManualProgramECC).txt
Example_Script (F2-PF0200A Sample-ManualProgramECC).txt
Example_Script (F2-Sample-ManualProgramECC).txt
F0_Hex.hex
F0_PF0200A_Hex.hex
F1_Hex.hex
F1_PF0200A_Hex.hex
F2_Hex.hex
F2_PF0200A_Hex.hex
Sample OTP 2.cfg
Sample OTP.cfg

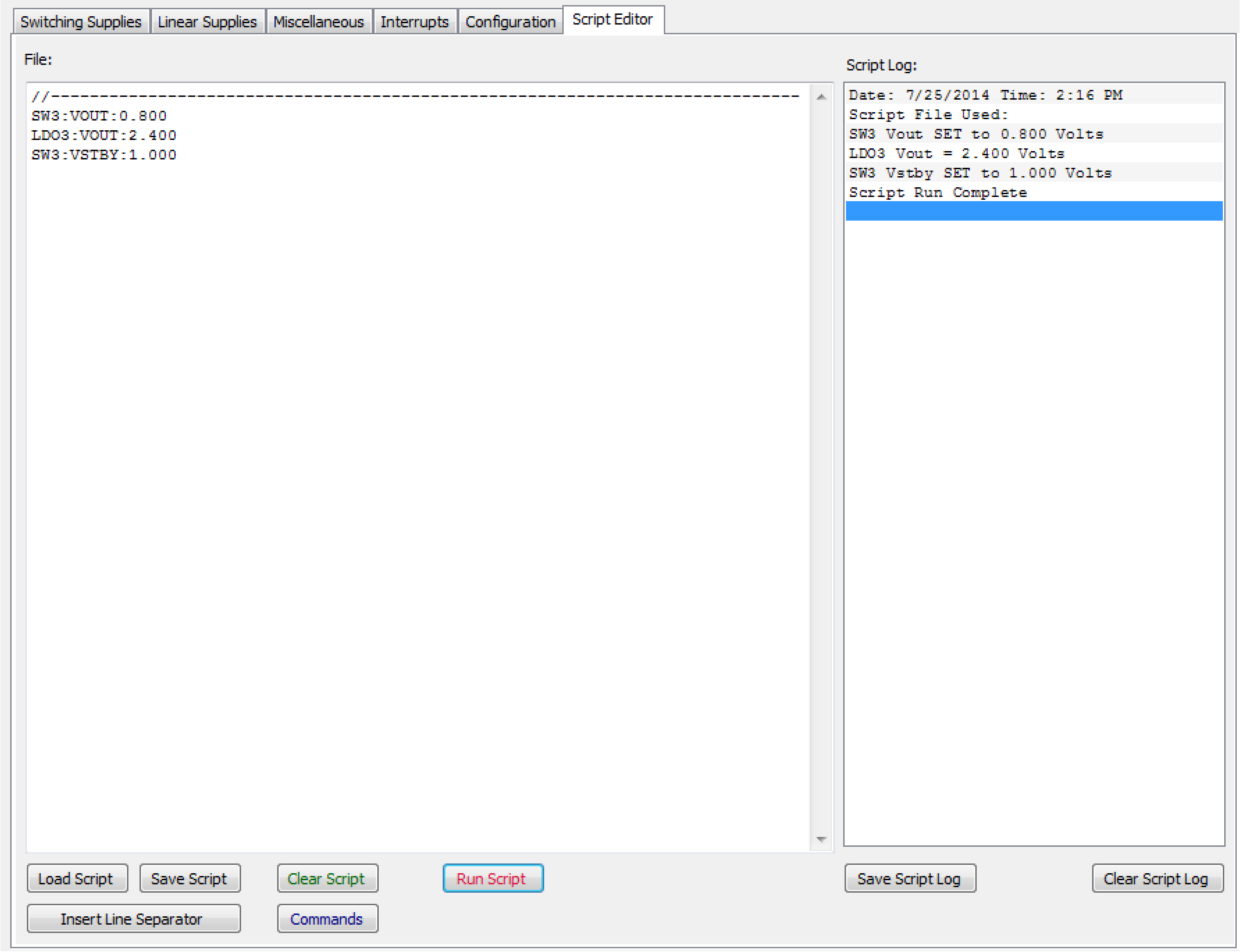
File name: Example_Script (F0-Sample-ManualProgramEC

Files of type:

Figure 17. 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 18.



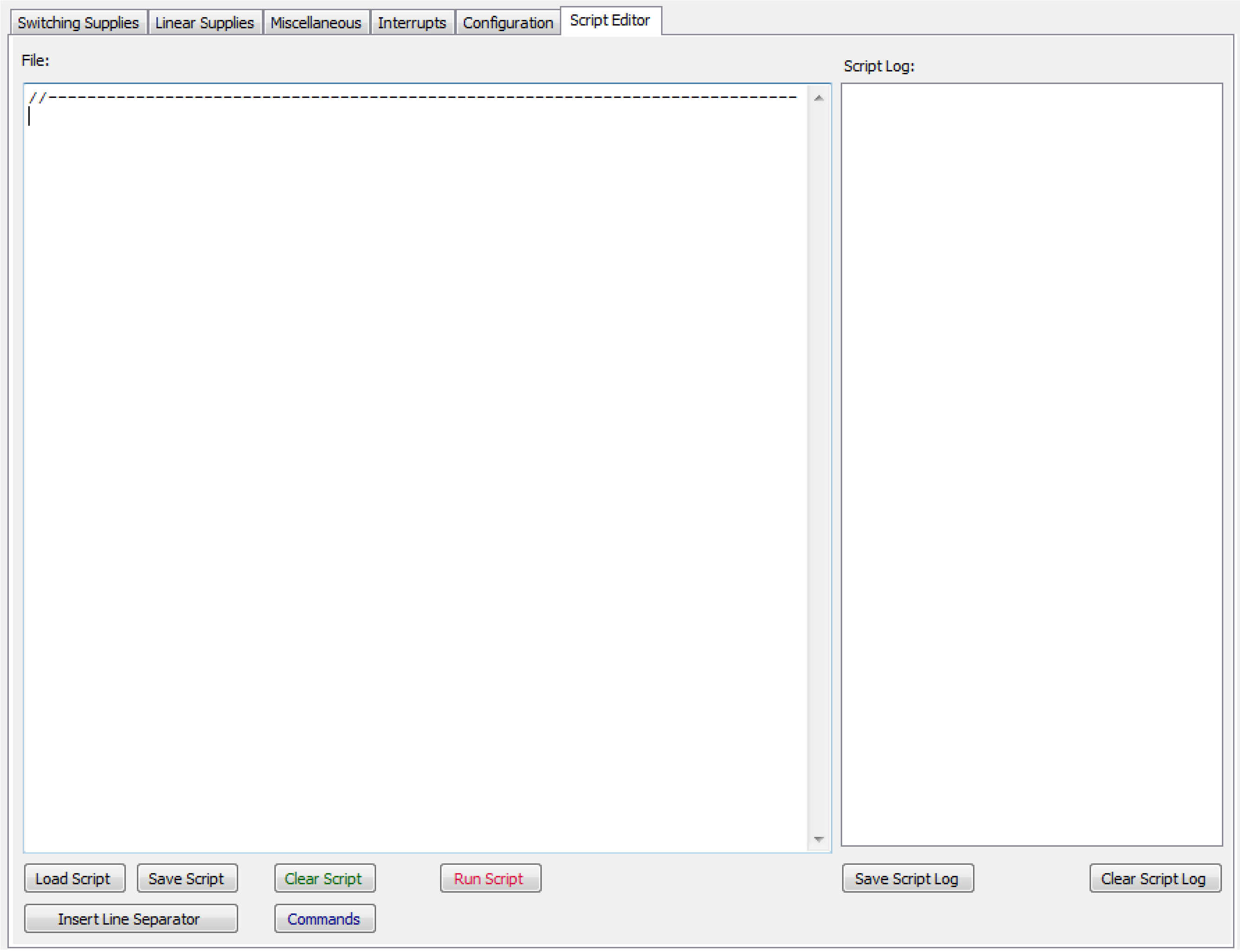
```
//------------------------------------------------------------------------------
SW3:VOUT:0.800
LDO3:VOUT:2.400
SW3:VSTBY:1.000
```

```
Date: 7/25/2014 Time: 2:16 PM
Script File Used:
SW3 Vout SET to 0.800 Volts
LDO3 Vout = 2.400 Volts
SW3 Vstby SET to 1.000 Volts
Script Run Complete
```

Figure 18. Running the Script

5.9.2 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 19. 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 20.

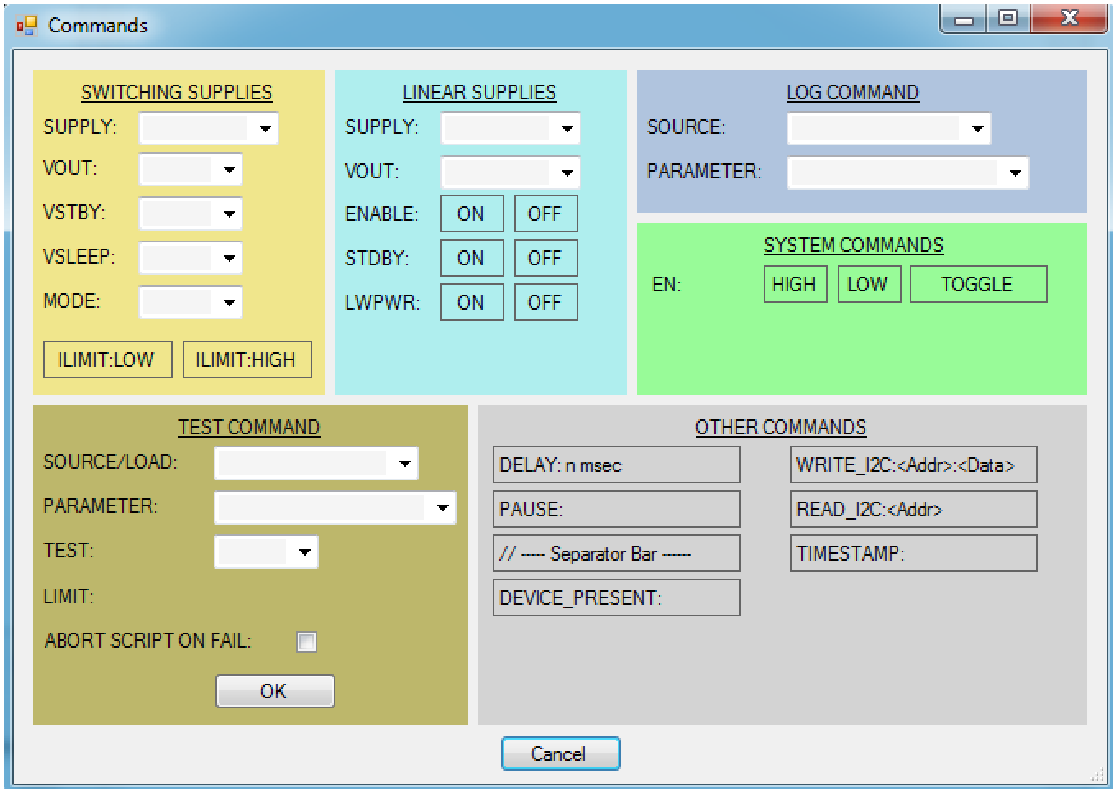


Figure 20. Command Selector Window

The **Command** window contains a set of commands useful for automatically sequencing the 34VR500 device power supplies, thereby emulating system behavior. The **Command** window is organized in six sections:

- 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 Sleep voltage setpoint, and current limit levels selection.
- 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.
- System Commands> Allows to control EN 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.
- 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 7.
- Test Command> Provides logic comparators to test the INT pin.

5.9.2.1 Syntax and Command Set

Delimiters

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

Table 7. Command List[5]

Command	Description
Hardware Control Commands	
WRITE_I2C:<Addr>:<Data>	Sends <Data> to I^{2}C register <Addr>[6]
READ_I2C:<Addr>	Reads the value of <Addr> and displays it in the Script Log [6]
V3V3:ON	Enables the 3.3 V system supply
V3V3:OFF	Disables the 3.3 V system supply
EN:HIGH	Releases the EN signal to a high-impedance state, allowing the 34VR500 to start up
EN:LOW	Asserts the EN signal Low, forcing the 34VR500 to shutdown
EN:TOGGLE	Asserts the EN signal Low, and then releases it to a high-impedance state, generating a power on event
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.
DEVICE_PRESENT:	Verify the presence of a 34VR500 device in the I^{2}C Bus
TIMESTAMP:	Prints out the current date and time
Switching Supplies Commands	
SWx:MODE:<operator>	Sets the mode of operation of the SWx regulator. The valid operators are as follows: • OFF • PFM • PWM • APS
SWx:VOUT:<value>	Sets the SWx output voltage in normal operation.
SWx:VSTBY:<value>	Sets the SWx output voltage to the STANDBY mode.
SWSleep:<value>	Sets the SWx output voltage to the Sleep mode.
SWx:ILIM:<operator>	Sets the SWx current limit level low or high. Valid operators: • LOW • HIGH
Linear Supply Commands	
ENABLEx:ON	Enables the LDOx supply
ENABLEx:OFF	Disables the LDOx supply
LDOx:STBY:<operator>	Enables the LDOx supply to stay ON or OFF during Standby mode. Valid operators: • ON • OFF
LDOx:LOWPWR:<operator>	Enables the low power bit for LDOx supply. Valid operators: • ON • OFF
LDOx:VOUT:<value>	Sets the output voltage for LDOx supply.
REFOUT:ON	Enables the REFOUT supply
REFOUT:OFF	Disables the REFOUT supply

Table 7. Command List[5] (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 • VSLEEP = 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 • DEFAULT_VOUT = Default power up voltage • DEFAULT_SEQUENCE = Default power up sequence of regulator • DEFAULT_CONFIG_REGISTER = Returns the SWx_CONF register value • DEFAULT_SWCONFIG= hardware configuration on SW1x and SW3x • DEFAULT_FSW = Switching Frequency set on registers
LOG:LDOx:<log operator>	Shows the current value of the <log operator> for the LDOx regulator. Log operators: • VOUT = Output voltage. • ENABLE = supply is ENABLED/DISABLED • STBY = Regulator status during Standby operation • LOWPWR = Low power mode enable or disabled • DEFAULT_VOUT = Default power up voltage • DEFAULT_SEQUENCE = Default power up sequence of regulator
LOG:REFOUT:<log operator>	Shows the current value of the <log operator> for the REFOUT regulator. Log operators: • ENABLE = supply is ENABLED/DISABLED • DEFAULT_SEQUENCE = Default power up sequence of regulator
LOG:DEFAULT_PU_CONFIG:<log operator>	Shows the current value set as default by DEFAULT. • SEQ_CLK_SPEED = programmed power up sequencing speed • DVS_CLK_SPEED = programmed DVS speed • EN_MODE = programmed EN 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 • SW1_OVERCURRENT • SW2_OVERCURRENT • SW3_OVERCURRENT • SW4_OVERCURRENT • LDO1_OVERCURRENT • LDO2_OVERCURRENT • LDO3_OVERCURRENT • LDO4_OVERCURRENT • LDO5_OVERCURRENT

Notes:

[5] All characters have to be entered in uppercase.

[6] The register and data values should be entered as hexadecimal numbers, for example: 0x20 is entered as 20.

5.9.3 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.

6 References

Document Number	Description	URL
KITVR500EVM	Tool Summary Page	http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=KITVR500EVM
MC34VR500	Product Summary Page	http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC34VR500
	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	8/2014	• 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.