UM12015

KITFS24SKTFDMEVM user manual

Rev. 1 — 1 February 2024 User manual

1 Introduction

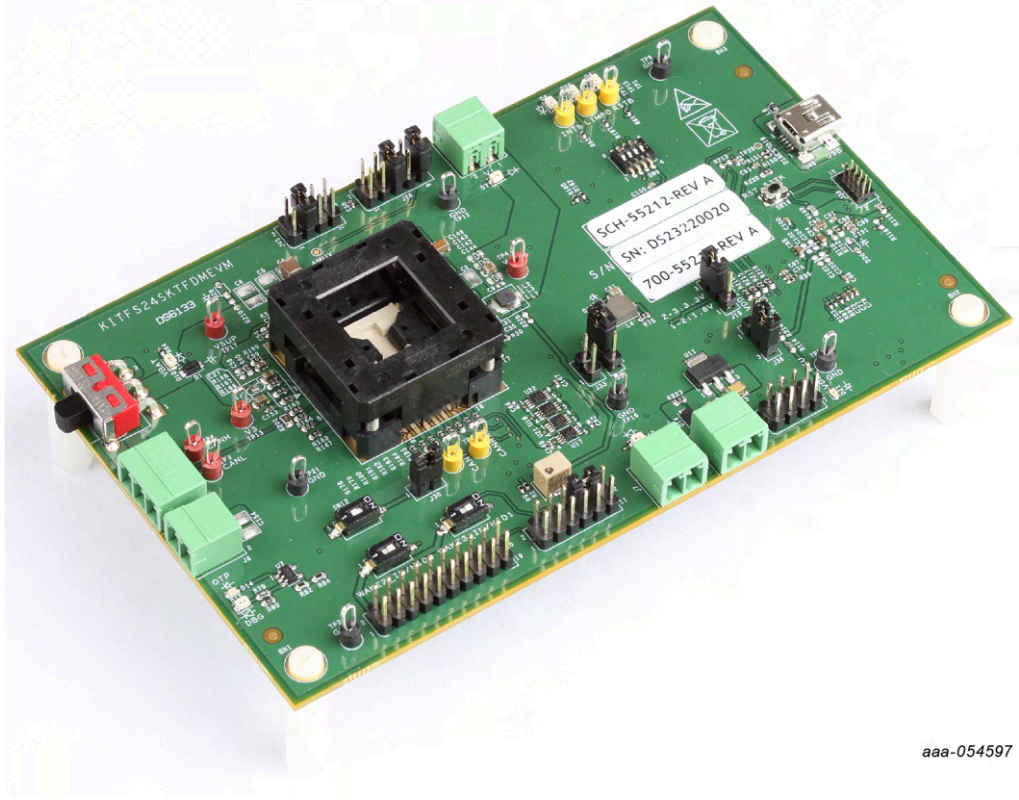


Figure 1. KITFS24SKTFDMEVM block diagram

This document is the user manual for the KITFS24SKTFDMEVM evaluation board. This document is intended for the engineers involved in the evaluation, design, implementation, and validation of the FS2400 fail-safe system basis chip (SBC) with switch mode power supply (SMPS) and low dropout (LDO).

The scope of this document is to provide the user with information to evaluate the FS2400 fail-safe system basis chip with SMPS and LDO. This document covers connecting the hardware, installing the software and tools, configuring the environment, and using the kit.

The KITFS24SKTFDMEVM enables development on the FS2400 device. The kit can be connected to the NXP GUI software that allows the user to interact with registers, try one time configurations (OTP), and burn the part.

The FS2400 devices can be placed and removed easily from the board by using the socket. The device OTP can be burned twice. This board supports the FS2400 device.

Important notice

IMPORTANT NOTICE

For engineering development or evaluation purposes only

NXP provides the product under the following conditions:

This evaluation kit is 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 evaluation board may be used with any development system or other source of I/O signals by connecting it to the host MCU computer board via off-the-shelf cables. This evaluation board is not a Reference Design and is not intended to represent a final design recommendation for any particular application. Final device in an application heavily depends 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 product 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 device incorporating the product. Due to the open construction of the product, it is the responsibility of the user to take all appropriate precautions for electric 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 NXP sales and technical support services.

2 Finding kit resources and information on the NXP web site

NXP Semiconductors provides online resources for this evaluation board and its supported device(s) on http://www.nxp.com.

The information page for KITFS24SKTFDMEVM evaluation board is at http://www.nxp.com/KITFS24SKTFDMEVM. The information page provides overview information, documentation, software and tools, parametric, ordering information, and a Getting Started tab. The Getting Started tab provides quick-reference information applicable to using the KITFS24SKTFDMEVM evaluation board, including the downloadable assets referenced in this document.

2.1 Collaborate in the NXP community

The NXP community is for sharing ideas and tips, asking and answering technical questions, and receiving input on just about any embedded design topic.

The NXP community is at http://community.nxp.com.

3 Getting ready

Working with the KITFS24SKTFDMEVM requires the kit contents, additional hardware, and a Windows PC workstation with installed software.

3.1 Kit contents

- Assembled and tested evaluation board, with preprogrammed S32K144 microcontroller embedded, placed in an anti-static bag
- USB-STD A to USB-B-mini cable
- Four connectors, terminal block plug, two positions, 3.81 mm pitch between pins
- One connector, terminal block plug, three positions, 3.81 mm pitch between pins
- Jumpers mounted on the board

3.2 Additional hardware

In addition to the kit contents, the following hardware is necessary or beneficial when working with this kit.

- A power supply with a range up to 40 V and a current limit set initially to 1 A

3.3 Windows PC workstation

This evaluation board requires a Windows PC workstation. Meeting the following minimum specifications is required:

- USB-enabled computer with Windows 7 or Windows 10
- FTDI USB serial port driver (for FT230X basic UART device)

3.4 Software

Installing the latest version of the NXP GUI software is necessary to work with this evaluation board. All listed software is available on NXP secure files system https://www.nxp.com/mynxp/secure-files

4 Getting to know the hardware

4.1 Kit overview

The KITFS24SKTFDMEVM kit provides an integrated platform for evaluating designs based on NXP's FS2400 SBC.

The kit hardware consists of the KITFS24SKTFDMEVM evaluation board with an embedded S32K144 microcontroller, and the USB cable required to connect the board to the PC.

The KITFS24SKTFDMEVM evaluation board features a socket that allows the user to fuse an individual FS2400 using the device's OTP feature. Connectors, jumpers, and switches on the board can be used to configure an evaluation environment that meets specific design requirements. The board also contains LEDs and test points that provide a means of monitoring performance in real time.

The S32K144 is soldered on the bottom side of the KITFS24SKTFDMEVM board. The role of the S32K144 is to interface the FS2400 device with the GUI installed on the connected computer. The S32K144 draws power either from the USB cable connected to the PC or from the battery supply (when not connected to the GUI).

Note: *Because of the socket, this kit is not optimized for performance measurement or current higher than 1 A.*

4.1.1 KITFS24SKTFDMEVM features

- Phoenix (3.81 mm) male connector for power supply input
- Phoenix (3.81 mm) male connectors for HVBUCK and HVLDO
- Phoenix (3.81 mm) male connectors for CAN communication
- Ignition key switch
- Embedded USB connection for easy connection to the NXP GUI (access to SPI/CAN bus, I/Os, RSTB, LIMP0, INTB, Debug, AMUX, regulators, registers, OTP emulation, and programming)
- LEDs that indicate signal or regulator status
- Support OTP fuse capabilities

Note: *Because of the socket, all current capabilities are limited to 1 A.*

4.1.2 Communication and I/Os between FS2400 and S32K144 MCU

The SPI bus, CAN Tx and Rx, along with the safety I/Os are connected to the S32K144 MCU.

This kit uses an S32K144 MCU to communicate with the NXP GUI. However, if the user wants to connect these I/Os to another MCU, this is possible. In this case, open the SW18 switch to disconnect the S32K144 MCU, see Figure 2, and connect the external MCU using J44 and J46 connectors, as shown in Figure 11. In addition to this change, make sure that the VDDIO voltage domain is the same on the MCU side and the SBC side.

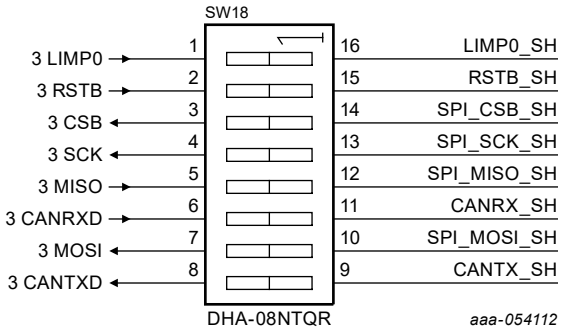


Figure 2. Communication/I/Os connection to S32K144

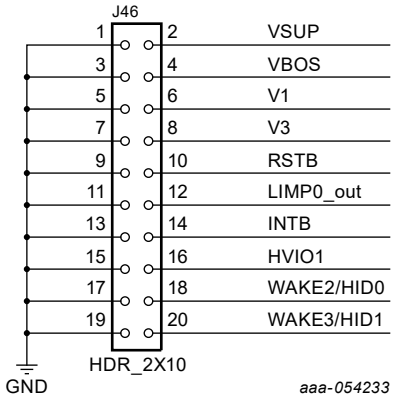


Figure 3. Power up debug signals

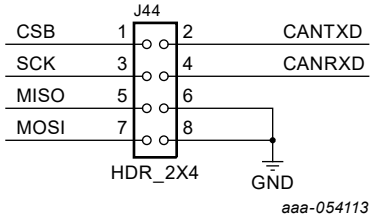


Figure 4. J44-46 Communication/IOs connection

4.1.3 VDDIO selection

The VDDIO pin is powered through the VDDIO net and is used to supply FS2400 internal buffers and SPI communication.

The selection of VDDIO is made using J26 connector. As an option, external LDOs are provided to feed VDDIO through the P_VAR net.

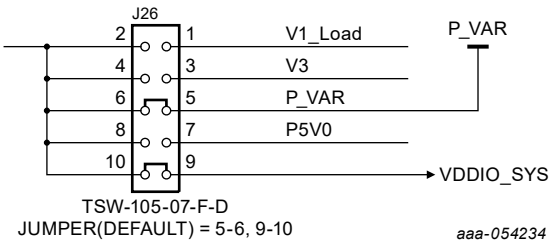


Figure 5. VDDIO selection

VDDIO can also be powered from V1, V3 internal regulators, or from P5V0 net, which can be 5 V from a USB or 5 V coming from an external LDO. Figure 6 shows all the different ways the VDDIO can be supplied. The default configuration is represented by the blue jumpers.

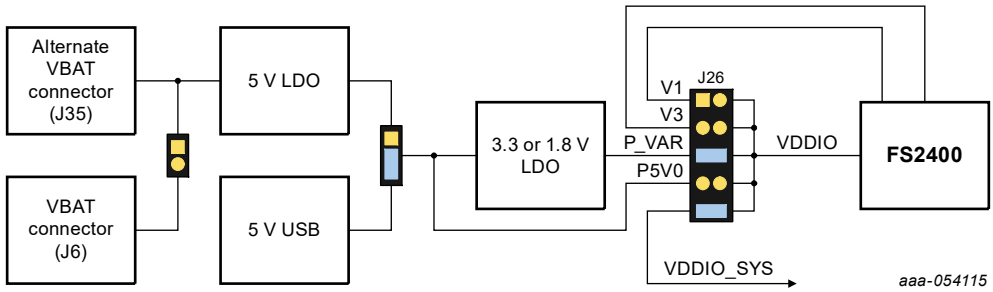


Figure 6. VDDIO power supply selection

VDDIO is compatible with 1.8 V, 3.3 V, or 5 V. Therefore, VDDIO voltage is configurable between 1.8 V, 3.3 V, or 5 V using J26 and J19 connectors (3.3 V by default). As J26 allows the user to choose between 5 V and P_VAR, J19 allows the user to configure P_VAR to 1.8 V or 3.3 V based on the jumper position. Figure 7 shows the schematics for P_VAR voltage selection.

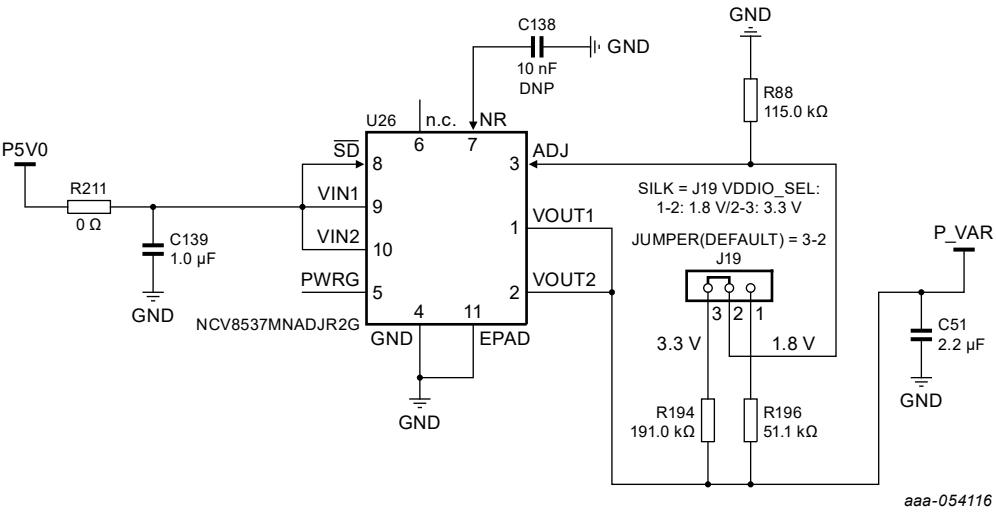


Figure 7. P_VAR supply configuration (LDO P_VAR (P3V3 or P1V8))

The VDDIO_SYS net is used to power nonapplicative components on the board, such as level shifters (to adjust voltage domain between FS2400 and S32K144) or LEDs. To get an accurate FS2400 current consumption measurement, remove the jumper 9-10 on J26.

4.1.4 S32K144 ADC

The S32K144 is configured to read some voltages from the FS2400 device using the circuits shown in Figure 8.

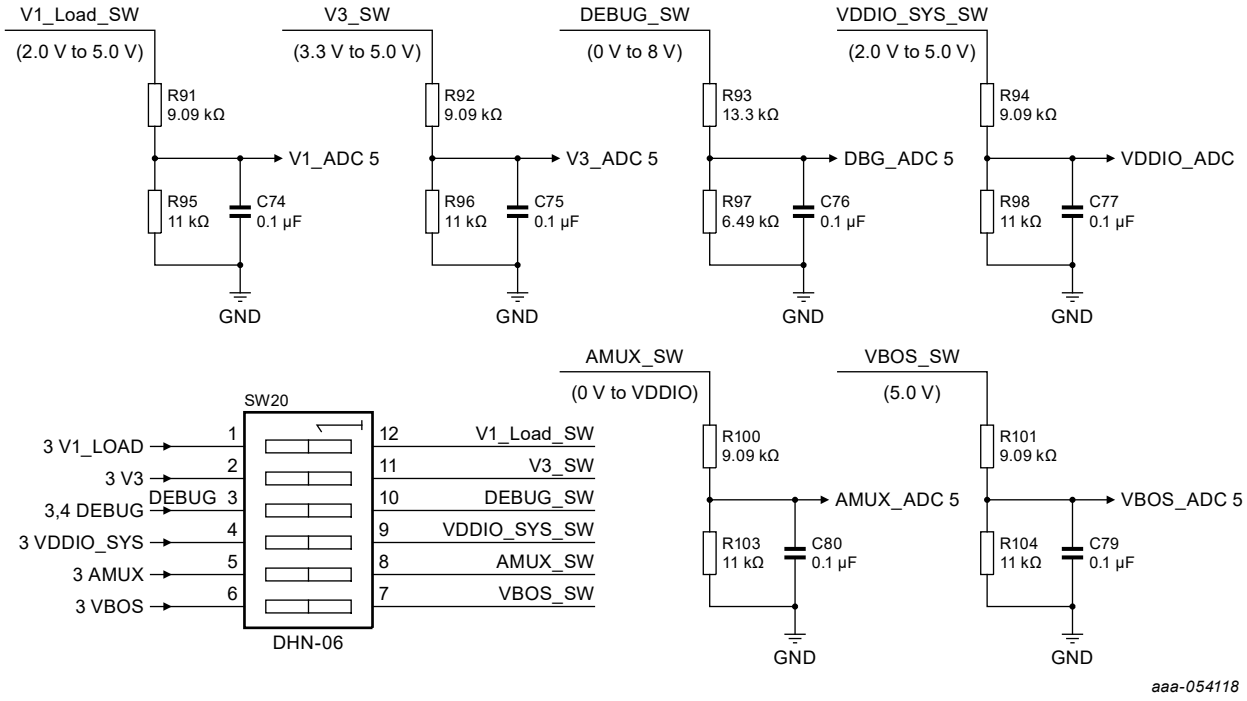


Figure 8. Analog inputs

This allows measurements of the FS2400 characteristic voltages using the AMUX tab of the NXP GUI.

For an accurate FS2400 current consumption measurement, open SW20.

4.2 Device OTP user configuration

NXP recommends learning about the OTP before operating the device. The FS2400 has a high level of flexibility because of its parameter configuration available in the OTP. The OTP affects the functionality of the device. In that sense, it is critical to understand how the OTP parameters are programmed, how to interact with the mirror registers, and the FS2400 SoC.

The OTP-related operations can be performed only in OTP mode, which allows:

- **Emulation:** The product uses a given configuration, as long as the power supply is not switched off
- **Programming:** The product will use the OTP-fused content that is valid even after the supply has been switched off/on

4.2.1 OTP and mirrors registers

The OTP configuration scheme is shown in Figure 9.

The device can be fused two times using mirror registers. The user can first load the mirror register content with the desired contents, then decide either to use the device in Emulation mode or to burn the next sector. The first sector burned is S1, the second is S1bis. The NXP GUI automatically manages the next sector to be burned. It is not possible to revert to the previous sector. When the user reaches the sector S1bis, there is no other possibility for burn. However, emulation of a different configuration in the mirror registers is still possible.

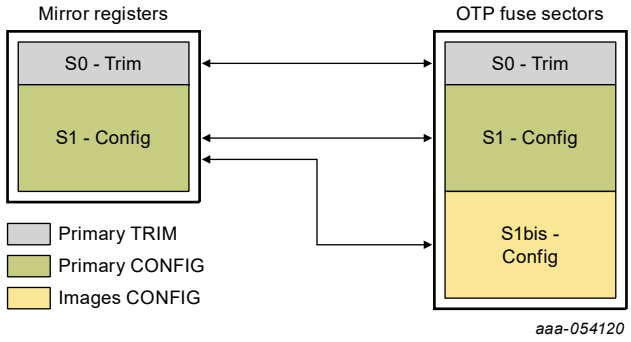


Figure 9. OTP block diagram

At boot, the content of the valid OTP fuse sector is loaded into the Mirror Register Sector 1. The content of the mirror registers is accessible from the NXP GUI by using specific SPI commands.

4.2.2 OTP hardware implementation

To work in OTP emulation or OTP programming, start the device in OTP mode.

Figure 10 shows the sequence to be followed to enter OTP mode. The voltage sequence on the kit is done using switches and a jumper installed on the board, while the OTP registers configuration is managed by the NXP GUI. This is described in detail in Section 6.6 and Section 6.7.

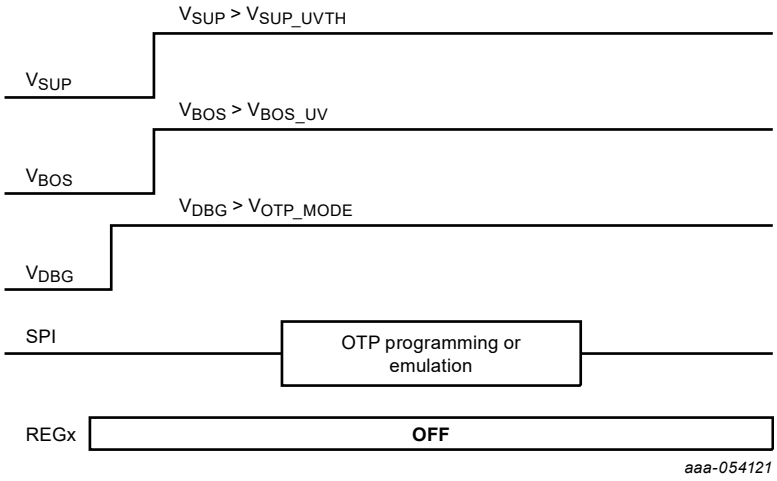


Figure 10. OTP mode entry

Figure 11 illustrates the hardware implementation.

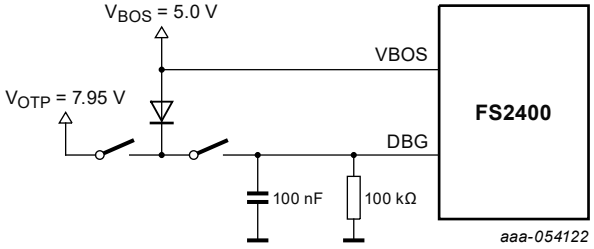


Figure 11. OTP hardware implementation

4.3 Kit featured components

Figure 12 identifies important components on the board and Table 1 provides additional details about these components.

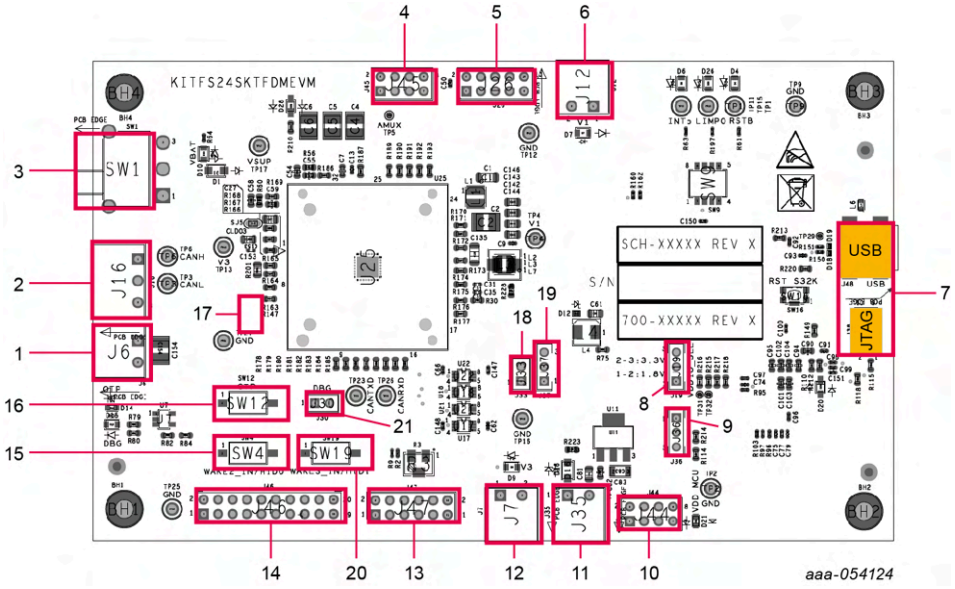


Figure 12. Evaluation board featured components location

Table 1. Evaluation board components description

Number	Description
1	VBAT Phoenix connector
2	CAN Phoenix connector
3	VBAT three position switch • Left position: Board supplied by USB • Middle position: Board not supplied • Right position: Board supplied by Phoenix connector
4	LIMP0 pullup selection
5	VDDIO supply selection
6	V1 power supply Phoenix connector
7	USB connectors (JTAG for MCU flash; S32K144 for NXP GUI control)
8	P_VAR net voltage selection (1.8 V or 3.3 V)
9	S32K144 supply selection
10	Communication debug connector (SPI and CAN)
11	ALT_VBAT Phoenix connector (to supply MCU or VDDIO from a supply other than USB)
12	V3 power supply Phoenix connector
13	FS2400 signals debug connector
14	FS2400 power-up signals debug connector
15	WAKE2/HID0 control switch
16	OTP mode control switch

Table 1. Evaluation board components description*...continued*

Number	Description
17	HVIO1 external pull up to VSUP (when configured as global input)
18	Jumper to supply VBAT to the external LDO providing 5 V
19	5 V selection (USB or external LDO)
20	WAKE3/HID1 control switch
21	Jumper to set DBG pin voltage to 0 V (if opened) or to 4.7 V/7.95 V (if closed)

4.3.1 FS2400: Fail-safe system basis chip with SMPS and LDO

4.3.1.1 General description

The FS2400 is a family of automotive safety SBC devices with multiple power supplies designed to support secure car access applications using ultra-wide band (UWB), near-field communication (NFC), and Bluetooth Low Energy devices while maintaining flexibility to fit other small applications requiring low power and CANFD communication.

This family of devices supports a wide range of applications, offering a choice in output voltage setting, physical interface, integrated system-level features to address low-power and noise-sensitive applications up to automotive safety integrity level (ASIL) B.

The FS2400 integrates a battery-connected switched mode regulator (V1) and a battery-connected linear regulator (V3) to supply the microcontroller, communication devices, and so on. V1 offers a high-performance switching regulator capable of operating in pulse frequency modulation (PFM) mode and forced pulse width modulation (FPWM) mode.

The FS2400 is developed in compliance with the ISO 26262 standard. The FS2400 includes enhanced safety features with fail-safe output, becoming part of a full safety-oriented system, covering up to ASIL B.

4.3.1.2 Features

Operating range

- 40 V DC maximum input voltage
- Low-power off mode with low sleep current and multiple wake-up sources
- Low-power on mode with HVBUCK (V1) active, HVLDO (V3) selectable by OTP, and multiple wake-up sources

Power supplies

- V1: High-voltage synchronous buck converter with integrated FETs. Configurable output voltage (2.0 V to 5.0 V) and switching frequency, output DC current capability up to 400 mA and PFM mode for Low-power on mode operation
- V3: High-voltage LDO regulator for microcontroller I/O support with selectable output voltage between 3.3 V and 5.0 V and up to 150 mA current capability

System support

- One CAN FD 5M following IEC 62228-3 2019 edition
- Four wake-up inputs (40 V capable): WAKEx pins, HVIO pins, CANFD or SPI command
- Hardware ID detection capability
- One high-voltage I/O with wake-up capability (40 V capable)
- Device control via 32-bit SPI interface with CRC
- Integrated long duration timer (LDT) and analog multiplexer (AMUX)

Functional safety

- Developed following ISO 26262:2018 standard to fit for ASIL B applications
- Internal monitoring circuitry with its own reference
- Additional input for external voltage monitoring
- Window or timeout watchdog function to monitor the MCU software failure
- Analog built-in self-test (ABIST) on demand
- Safety outputs (RSTB, LIMP0)
- Safety input to monitor external IC state (ERRMON)

Configuration and enablement

- HVQFN32EP: QFN, 32 pins with exposed pad for optimized thermal management, wettable flanks, 5 mm x 5 mm x 0.85 mm, 0.5 mm pitch
- Permanent device customization via OTP fuse memory
- OTP emulation mode for system development and evaluation

4.3.2 LED signaling

Figure 13 shows the LEDs provided as visual output devices for the evaluation board:

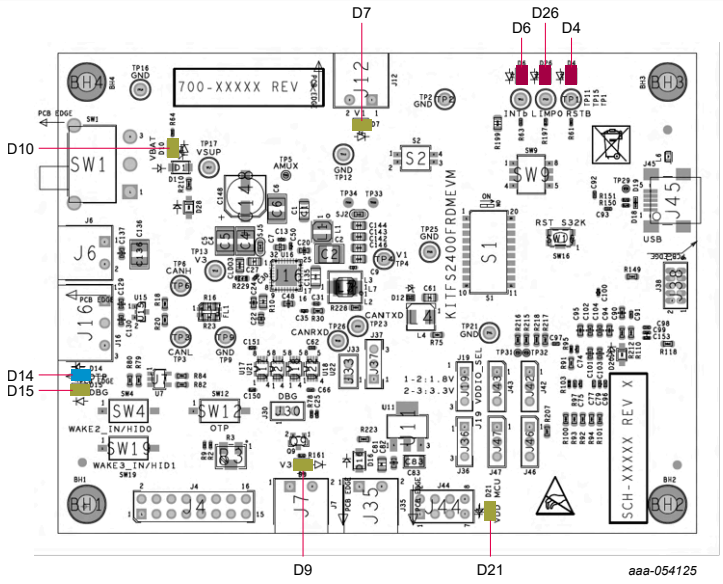


Figure 13. Evaluation board LED signaling location

Table 2. Evaluation board LED signaling description

Label	Name	Color	Description
D4	RSTB	Red	RSTB asserted (logic level = 0)
D6	INTB	Red	INTB asserted (logic level = 0)
D7	V1	Green	V1 Power supply on
D9	V3	Green	V3 Power supply on
D10	VBAT	Green	VBAT on
D14	OTP mode	Blue	OTP mode threshold crossed
D15	DBG mode	Green	DBG mode threshold crossed

Table 2. Evaluation board LED signaling description...*continued*

Label	Name	Color	Description
D21	VDD MCU	Green	MCU supplied
D26	LIMP0	Red	LIMP0 asserted (logic level = 0)

4.3.3 Connectors

Figure 14 shows the location of the connectors on the board.

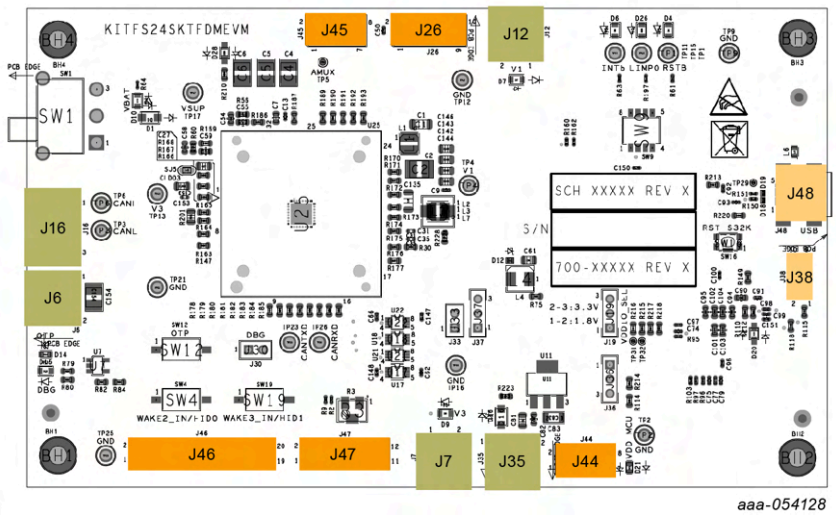


Figure 14. Evaluation board connectors

4.3.3.1 VBAT connectors

VBAT connects to the board through the Phoenix connector (J6).

Table 3. Main VBAT Phoenix connector (J6)

Schematic label	Signal name	Description
J6-1	VBAT	Main battery voltage supply input for FS2400
J6-2	GND	Ground

Table 4. Alternative VBAT Phoenix connector (J35)

Schematic label	Signal name	Description
J35-1	VBAT_ALT	Alternative battery voltage supply input for external LDOs and MCU
J35-2	GND	Ground

4.3.3.2 Output power supply connectors

Table 5. V3 (HVLDO) connector (J7)

Schematic label	Signal name	Description
J7-1	V3	V3 power supply output
J7-2	GND	Ground

Table 6. V1 (HVBUCK) connector (J12)

Schematic label	Signal name	Description
J12-1	V1_load	V1 power supply output
J12-2	GND	Ground

4.3.3.3 CAN bus connector

Table 7. V3 CAN bus connector (J16)

Schematic label	Signal name	Description
J16-1	CANH	CAN bus high
J16-2	CANL	CAN bus low
J16-3	GND	Ground

4.3.3.4 Debug connectors

Table 8. Power-up debug connector (J46)

Schematic label	Signal name	Description
J46-(odd)	GND	Ground
J46-2	VSUP	VSUP pin
J46-4	VBOS	Best of supply
J46-6	V1	V1 power supply output
J46-8	V3	V3 power supply
J46-10	RSTB	Reset pin (active low)
J46-12	LIMP0_out	Fail-safe pin (active low)
J46-14	INTB	Interruption pin (active low)
J46-16	HVIO1	High-voltage I/O pin
J46-18	WAKE2/HID0	Wake pin or HW ID depending on configuration
J46-20	WAKE3/HID1	Wake pin or HW ID depending on configuration

Table 9. Communication debug connector (J44)

Schematic label	Signal name	Description
J44-1	CSB	SPI chip select (active low)
J44-2	CANTXD	CAN transmitter data
J44-3	SCK	SPI clock
J44-4	CANRXD	CAN receiver data
J44-5	MISO	SPI MISO
J44-6	GND	Ground
J44-7	MOSI	SPI MOSI
J44-8	GND	Ground

Table 10. FS2400 signals debug connector (J47)

Schematic label	Signal name	Description
J47-1	WAKE2_IN/HID0	Global input wake pin or HW ID depending on configuration
J47-2	WAKE2/HID0	Wake pin or HW ID depending on configuration
J47-3	WAKE3_IN/HID1	Global input wake pin or HW ID depending on configuration
J47-4	WAKE3/HID1	Wake pin or HW ID depending on configuration
J47-5	HVIO1_IN	Global input HVIO1 pin
J47-6	AMUX	Analog multiplexer output
J47-7	VDDIO	VDDIO pin
J47-8	VMON_EXT	VMON_EXT pin
J47-9	DEBUG	Debug pin
J47-10	P3V3	External 3.3 V power supply
J47-11	VDIG	VDIG pin
J47-12	GND	Ground

4.3.3.5 VDDIO selection connector

Table 11. VDDIO selection connector (J26)

Schematic label	Signal name	Description
J26-1	V1_Load	V1 power supply output
J26-2	VDDIO	VDDIO pin
J26-3	V3	V3 power supply output
J26-4	VDDIO	VDDIO pin
J26-5	P_VAR	External LDO variable voltage (1.8 V or 3.3 V) output
J26-6	VDDIO	VDDIO pin
J26-7	P5V0	External LDO 5 V output
J26-8	VDDIO	VDDIO pin
J26-9	VDDIO_SYS	Net supplying nonapplicative components (LEDs, …)
J26-10	VDDIO	VDDIO pin

4.3.3.6 LIMP0 pullup selection connector

Table 12. LIMP0 pullup selection connector (J45)

Schematic label	Signal name	Description
J26-1	LIMP0	LIMP0 pin
J26-2	VSUP	VSUP pin
J26-3	LIMP0	LIMP0 pin
J26-4	VDDIO	VDDIO pin
J26-5	LIMP0	LIMP0 pin
J26-6	V1	V1 power supply output

Table 12. LIMP0 pullup selection connector (J45)*...continued*

Schematic label	Signal name	Description
J26-7	LIMP0	LIMP0 pin
J26-8	V3	V3 power supply output

4.3.3.7 S32K144 communication connectors

Table 13. USB connector (J48)

Schematic label	Signal name	Description
J48	NA	USB connector to interface GUI with S32K144

Table 14. JTAG connector (J38)

Schematic label	Signal name	Description
J38	NA	JTAG connector to flash S32K144 SW

4.3.4 Test points

Figure 15 shows the test points that provide access to various signals to and from the boards.

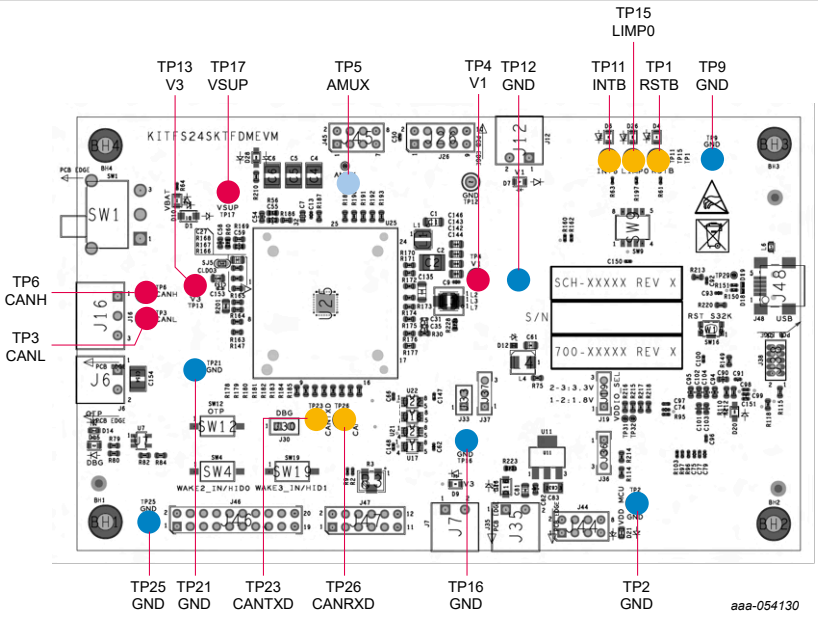


Figure 15. Evaluation board test points

Table 15. Evaluation board test points description

Test point name	Signal name	Description
TP1	RSTB	Reset pin (active low)
TP2	GND	Ground
TP3	CANL	CAN bus low
TP4	V1_load	V1 power supply output
TP5	AMUX	Analog multiplexer pin

Table 15. Evaluation board test points description*...continued*

Test point name	Signal name	Description
TP6	CANH	CAN bus high
TP9	GND	Ground
TP11	INTB	Interruption pin
TP12	GND	Ground
TP13	V3	V3 power supply output
TP15	LIMP0	LIMP0 pin
TP16	GND	Ground
TP17	VSUP	VSUP pin
TP21	GND	Ground
TP23	CANTXD	CAN transmitter
TP25	GND	Ground
TP26	CANRXD	CAN receiver

4.3.5 Jumpers

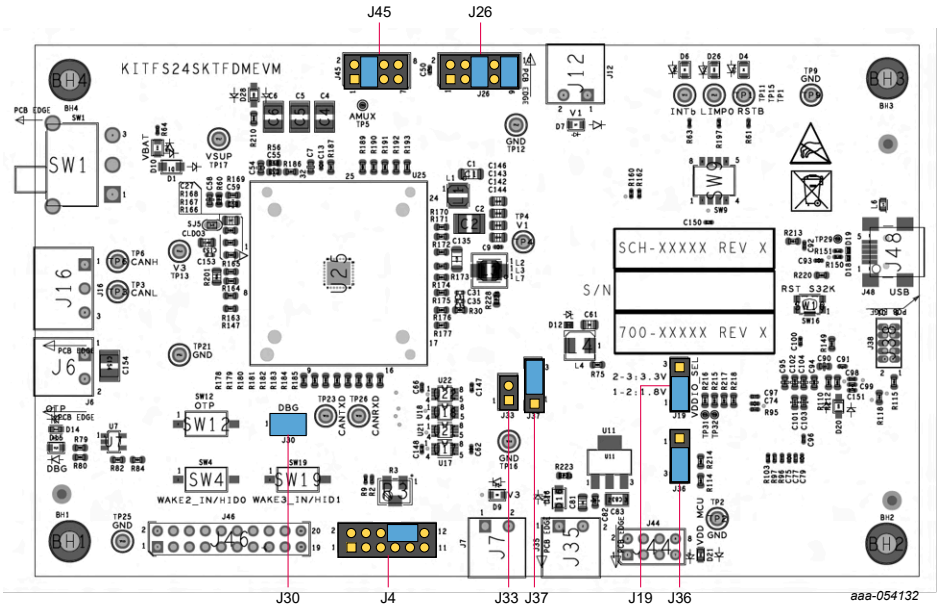


Figure 16. Evaluation board jumpers location (with default position)

Table 16. Evaluation board jumpers description

Name	Function	Pin number	Jumper/pin function
J4	VMON_EXT easy evaluation	8–10	3.3 V to VMON_EXT resistor bridge input
J19	P_VAR output voltage selection	1−2	P_VAR is 1.8 V
		2–3	P_VAR is 3.3 V
J26	VDDIO selection	1−2	VDDIO is supplied by V1
		3−4	VDDIO is supplied by V3
		5–6	VDDIO is supplied by P_VAR external LDO
		7–8	VDDIO is supplied by 5 V external LDO
		9–10	VDDIO_SYS net is supplied from VDDIO net
J30	Apply voltage to DBG pin	1–2	Either 5 V (Debug mode) or 8 V (OTP mode) depending on SW12 position
J33	Power 5 V LDO from main VBAT supply	1−2	5 V LDO input supplied by main VBAT
J36	Select S32K144 supply	1−2	S32K144 is supplied from an external 5V LDO
		2−3	S32K144 is supplied from an external P_VAR LDO
J37	Select P5V0 net supply	1−2	P5V0 net is supplied by an external 5 V LDO
		2–3	P5V0 net is supplied by 5 V from USB
J45	Select LIMP0 pull up	1−2	LIMP0 pulled up to VSUP
		3–4	LIMP0 pulled up to VDDIO
		5–6	LIMP0 pulled up to V1
		7–8	LIMP0 pulled up to V3

4.3.6 Switches

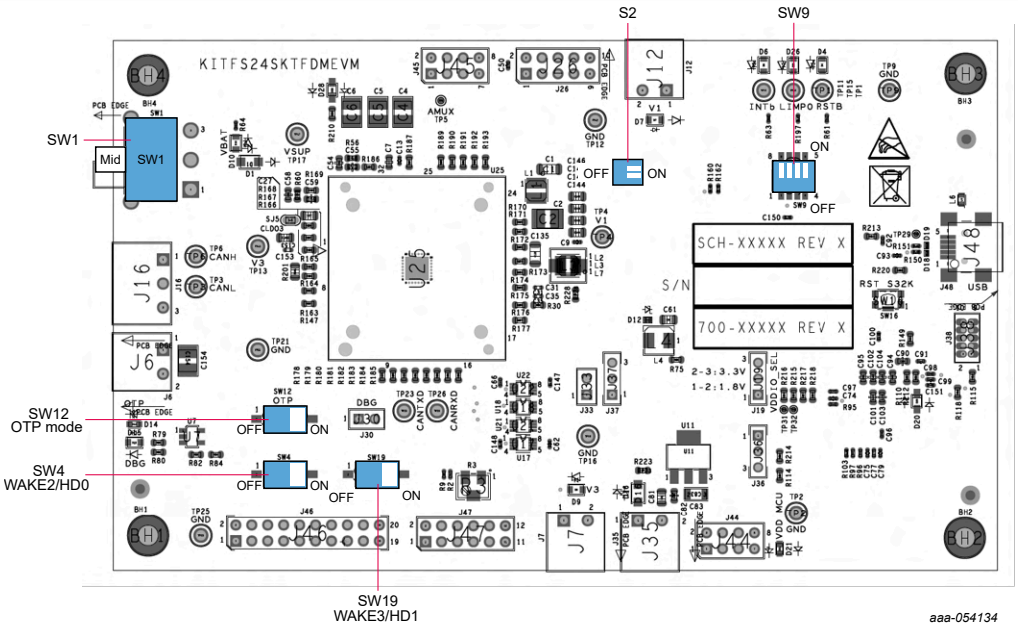


Figure 17. Switches location

Table 17. SW1 description

Position	Function	Description
Up (2–3)	VBAT on	Powered by external supply through J6
Mid	VBAT off	—
Down (1–2)	VBAT on	Powered by board embedded boost (8 V): Only for quick debug, don't load FS2400 using this supply

Table 18. S2 description

Position	Function	Description
Right	Corresponding LED on	Each LED is controlled by an independent switch; disconnecting them allows more accurate current consumption measurement
Left	Corresponding LED off	

Table 19. SW4 description

Position	Function	Description
Left	WAKE2/HID0 off	WAKE2/HID0 pin tied to VSUP
Right	WAKE2/HID0 on	WAKE2/HID0 pin opened

Table 20. SW9 description

Position	Function	Description
Top	Corresponding LED on	Each LED is controlled by an independent switch; disconnecting them allows more accurate current consumption measurement
Bottom	Corresponding LED off	

Table 21. SW12 description

Position	Function	Description
Left	OTP mode off	FS2400 starts in Debug (if J30 1-2 jumper is placed) or in Applicative mode (if J30 jumper is off)
Right	OTP mode on	FS2400 starts in OTP mode (if J30 1-2 jumper is placed) or in Applicative mode (if J30 jumper is off)

Table 22. SW19 description

Position	Function	Description
Left	WAKE3/HID1 off	WAKE3/HID1 pin tied to VSUP
Right	WAKE3/HID1 on	WAKE3/HID1 pin opened

4.4 Schematic, board layout and bill of materials

The schematic, board layout, and bill of materials for the KITFS24SKTFDMEVM evaluation board are available at http://www.nxp.com/KITFS24SKTFDMEVM.

5 Installing and configuring software and tools

The KITFS24SKTFDMEVM is always delivered with the GUI firmware already flashed. If the MCU firmware is already flashed, the user may ignore this section. If it is specified that the user must update the firmware or it is malfunctioning, follow the instructions in Section 5.1.

5.1 Flashing or updating the GUI firmware

In the event the user must flash the MCU S32K144, this section explains how to proceed.

5.1.1 Hardware setup

First, get a compatible debugger (the provided firmware is compatible with PEmicro debuggers):

U-MULTILINK NXP Semiconductors | Mouser France

Figure 18. Example debugger

Make sure to properly connect the debugger:

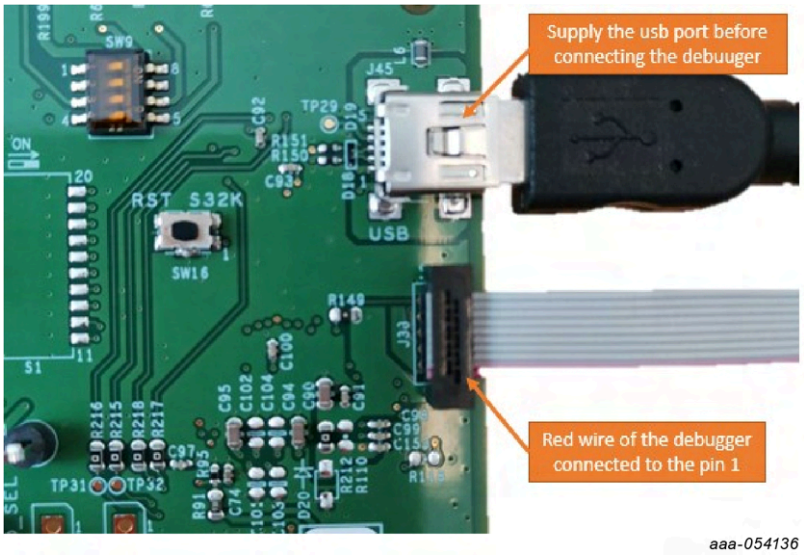
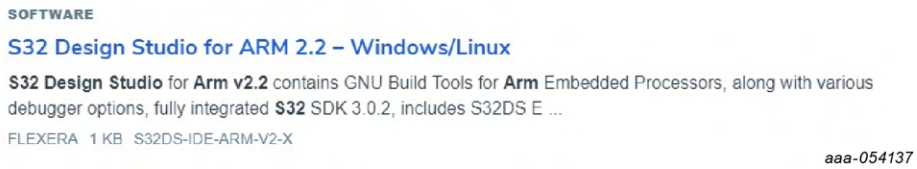


Figure 19. Proper connection

1. Connect the mini USB port to the computer to supply the microcontroller.
2. Connect the debugger to the J38 connector and be careful to connect the red wire to Pin 1.

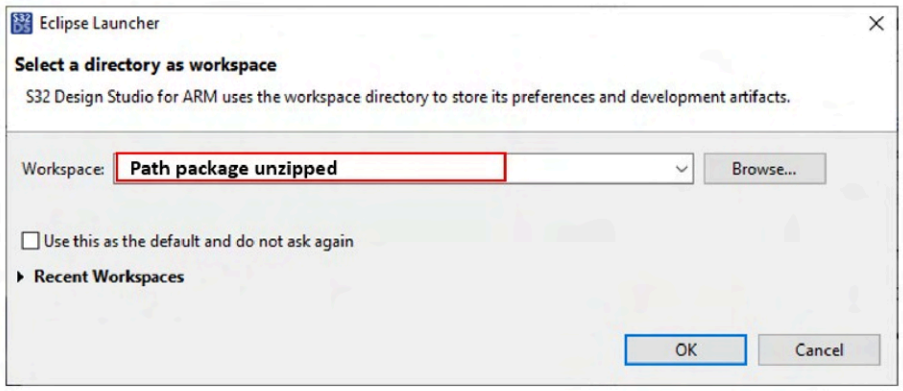
5.1.2 Software setup

1. Download Design Studio : https://www.nxp.com/webapp/swlicensing/sso/downloadSoftware.sp?catid=S32DS-IDE-ARM-V2-X



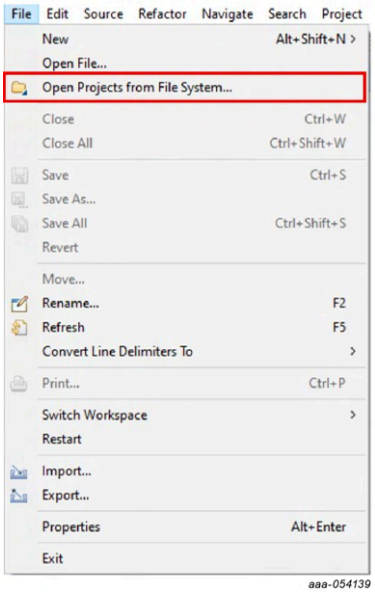
aaa-054137

2. Once downloaded, use the Design Studio installation guide included inside the package provided: “KITFS2400FRDMEVM_HW_Test_Package_W20.zip”.
3. Launch Design Studio



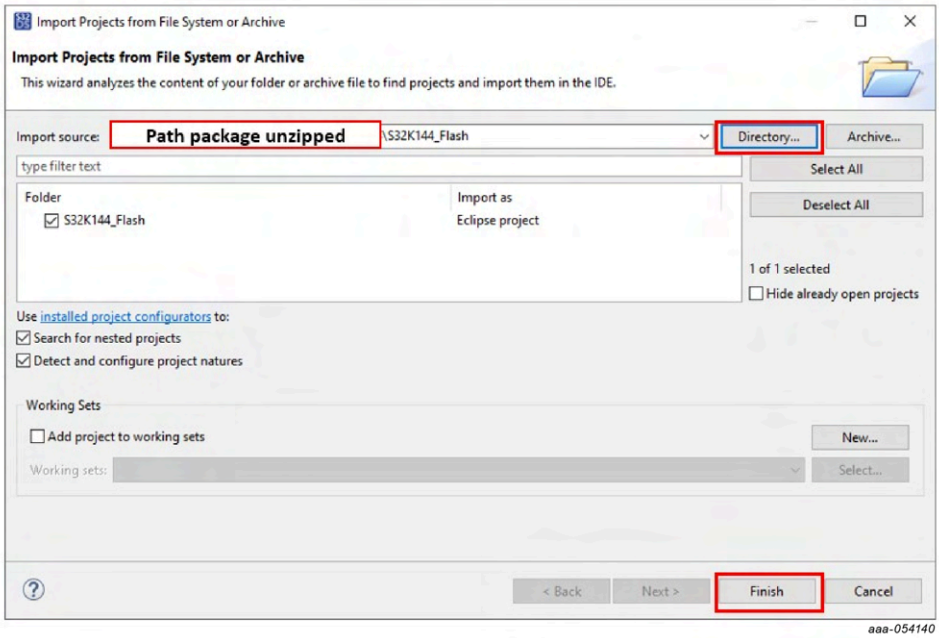
aaa-054138

4. Open the project provided

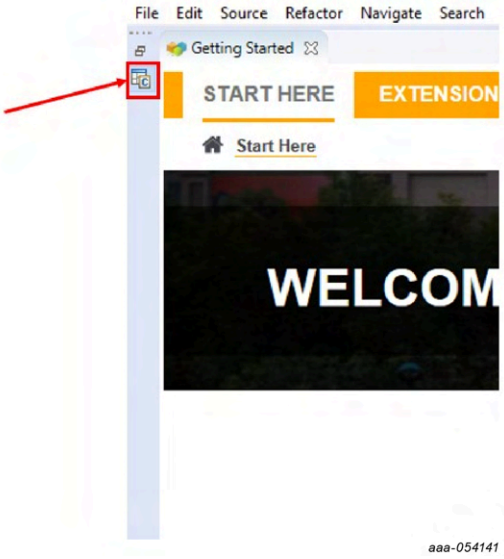


aaa-054139

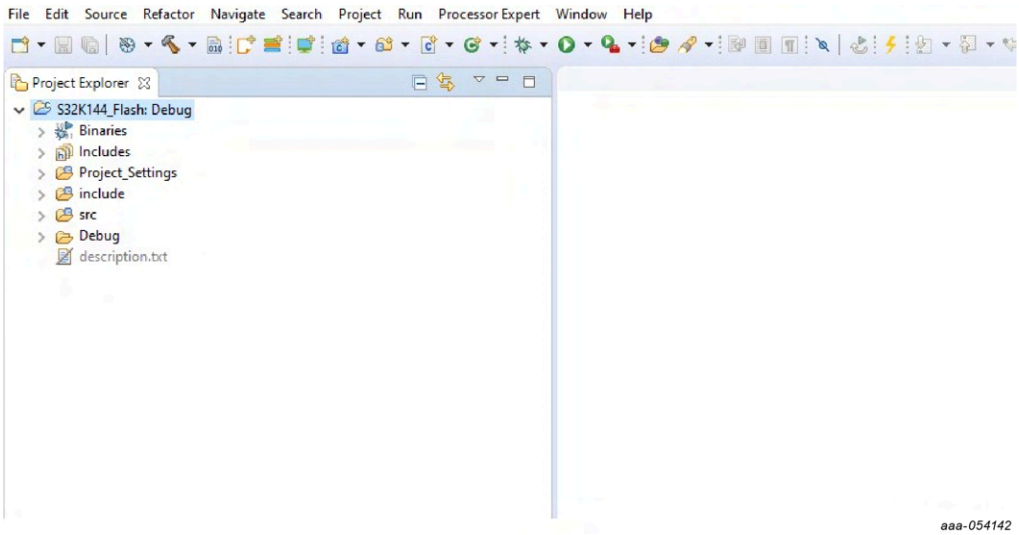
Go to the Directory and the location of the unzipped folder. Select the S32K144_Flash folder:



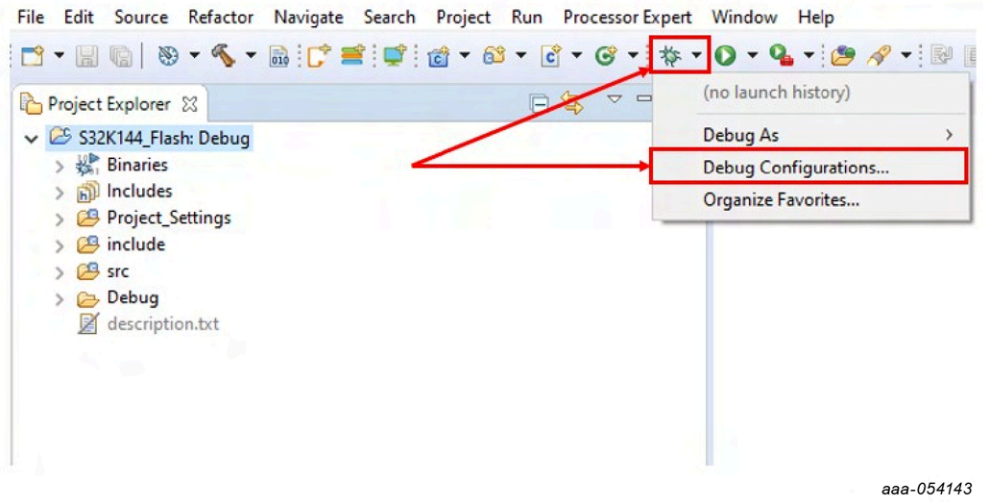
In the following window, click the button identified in the red box to access of the project view.



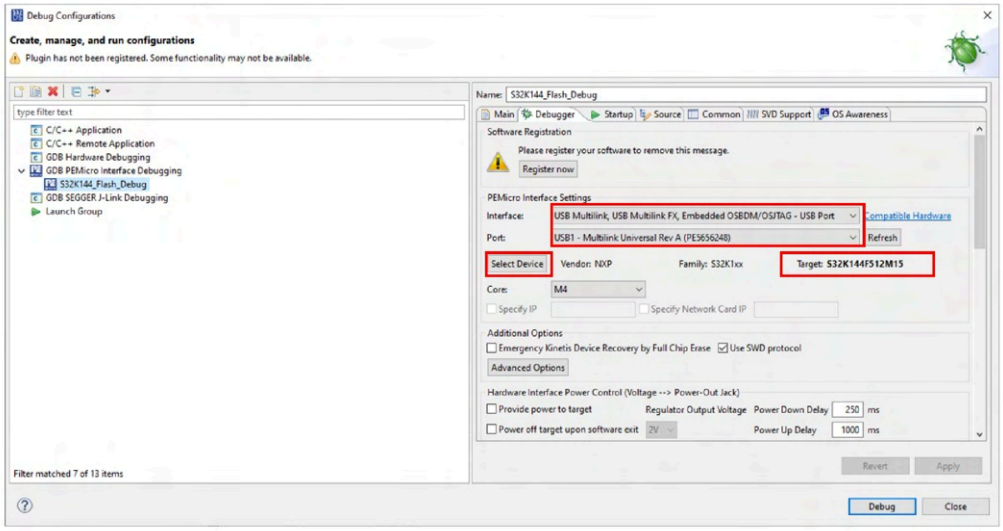
The Project Explorer view appears:



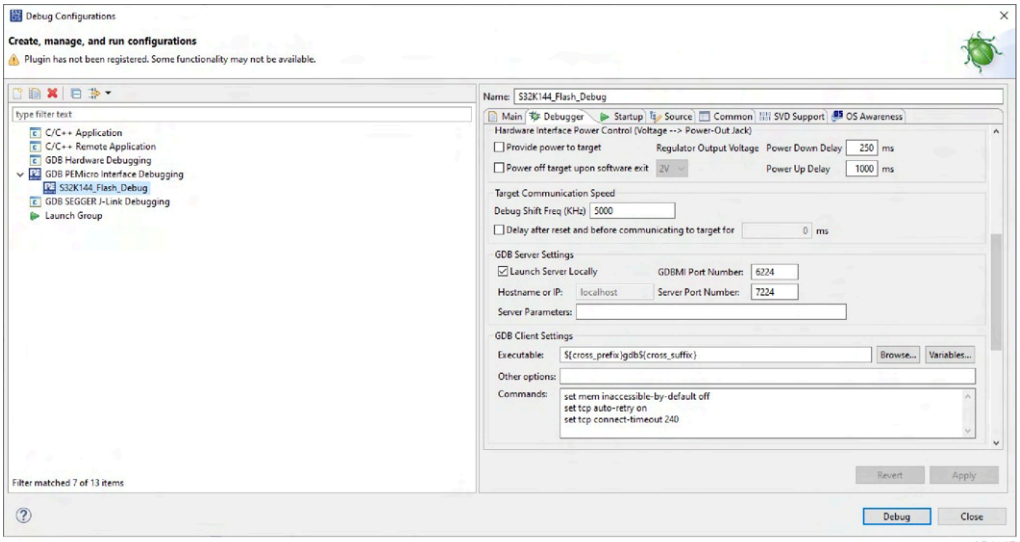
5. Configure the debug parameters:
   Go to the symbol inside the red box shown below and choose Debug Configurations.



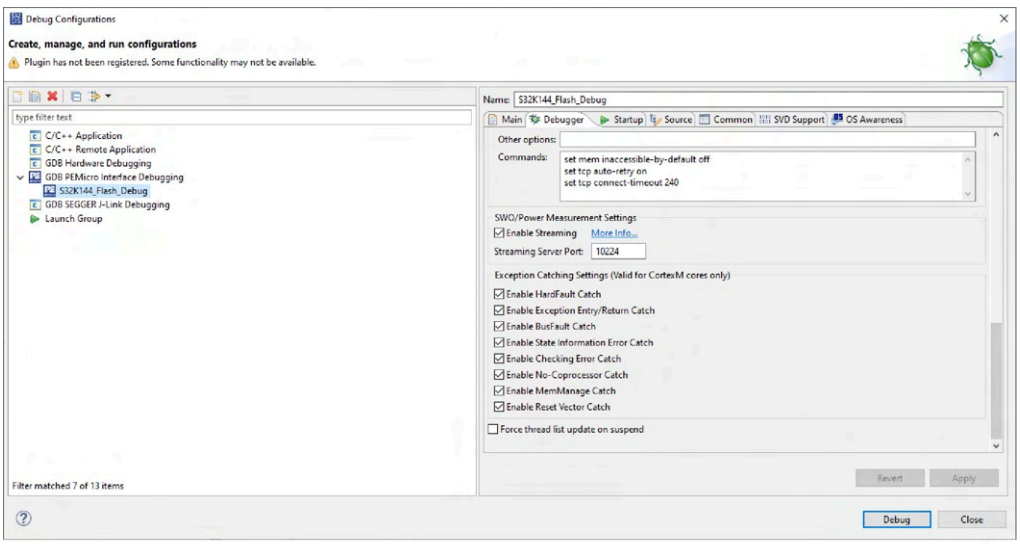
The debugger configuration should be the same as below:



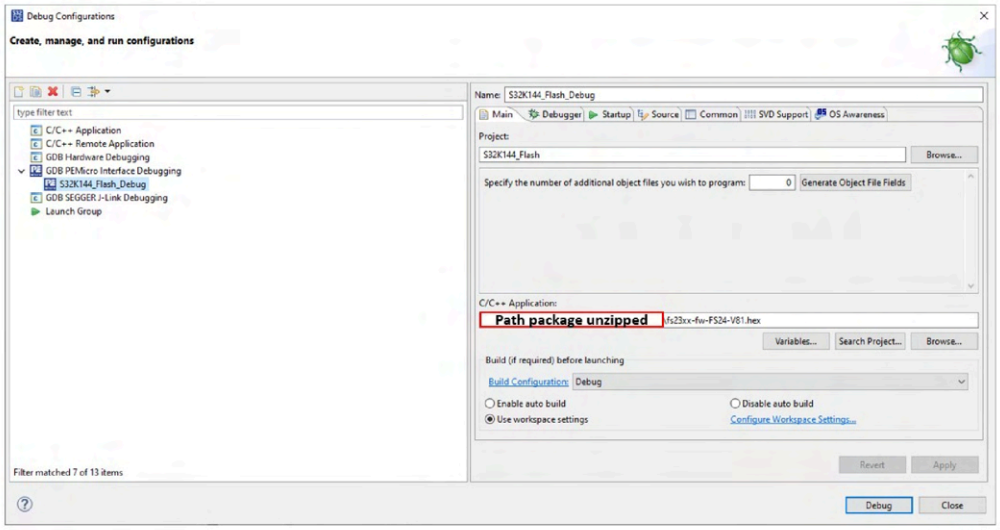
aaa-054144



aaa-054145

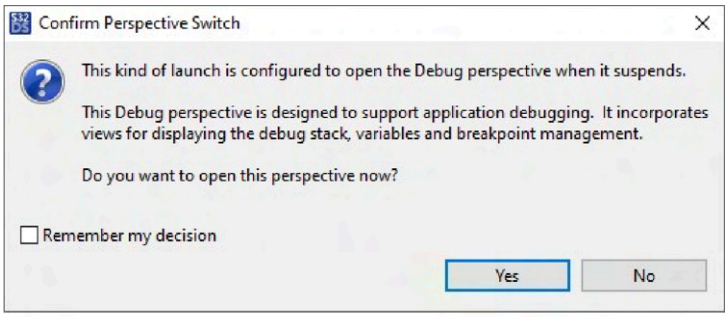


aaa-054146



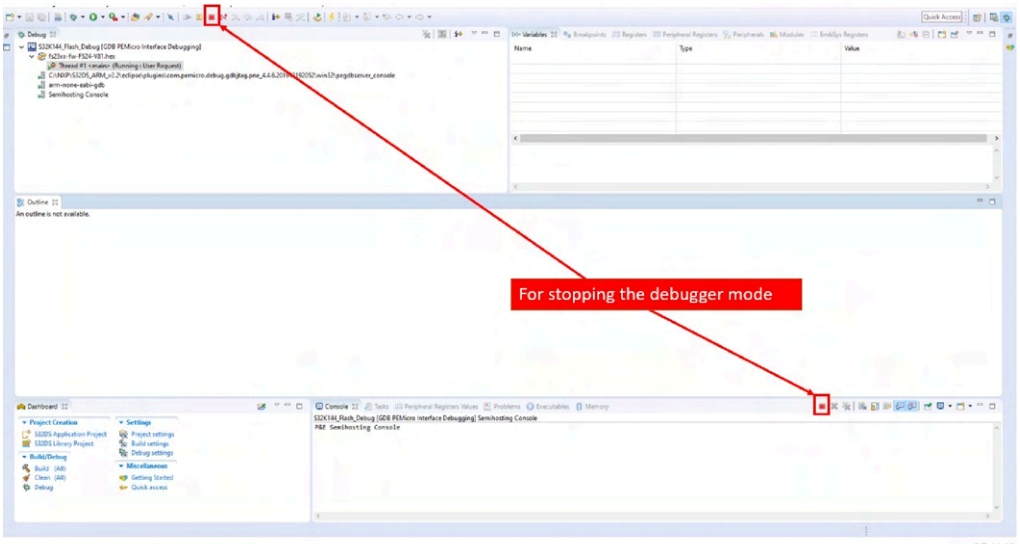
aaa-054147

When the window below appears, click **Yes**.



aaa-054148

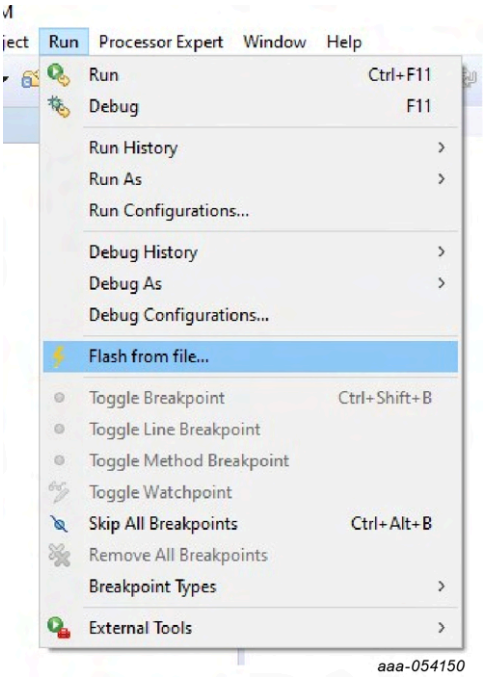
If the debug configuration is properly set, the following window should appear:



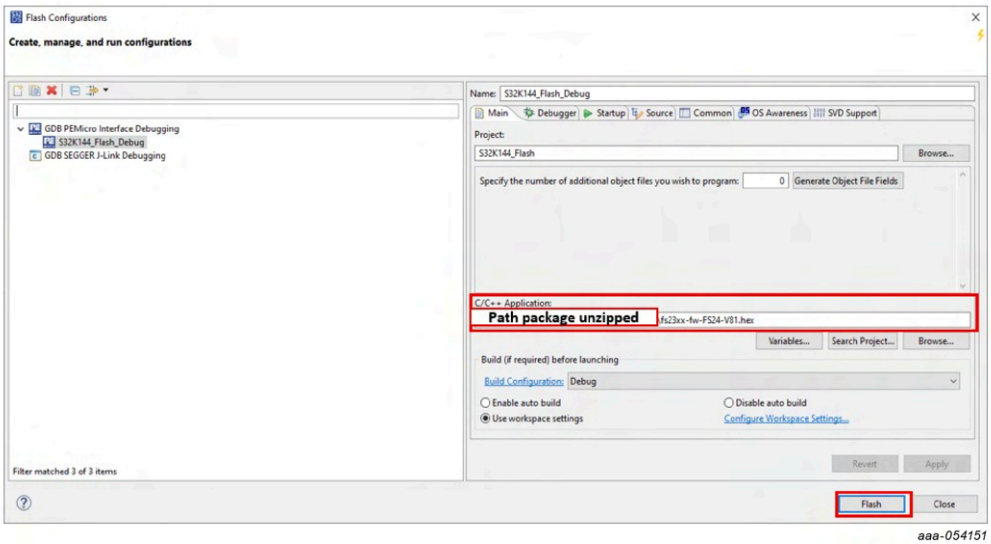
aaa-054149

Click either **Stop** button to stop the debugger.

6. Go to the Run menu and click **Flash from file** as shown below:



7. Select the file “fs23xx-fw-FS24-Vxx.hex” provided in the installation package.



Click **Flash** once the .hex file is selected.
At the end, if the microcontroller is correctly flashed, the following message appears:

```
   Checksum Verification Successful. (Cumulative CRC-16=$2705)
Application verified in memory. No need to reprogram.

CMD>RE

Initializing.
Target has been RESET and is active.
```

aaa-054152

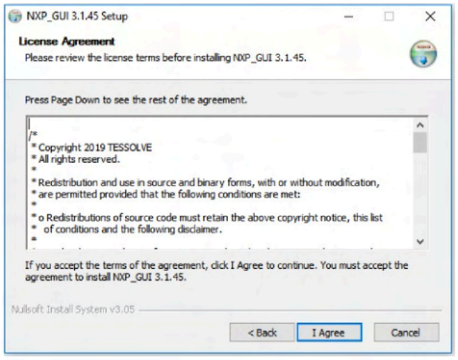
8. Install the NXP GUI.

9. Connect the board to the computer via the mini USB port. Click **Start** (at the upper left of the interface). If the NXP user interface recognizes the microcontroller, the version of the firmware will appear at the bottom of the window.

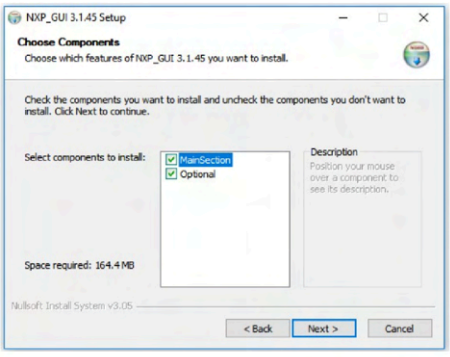
5.2 Installing NXP GUI software package

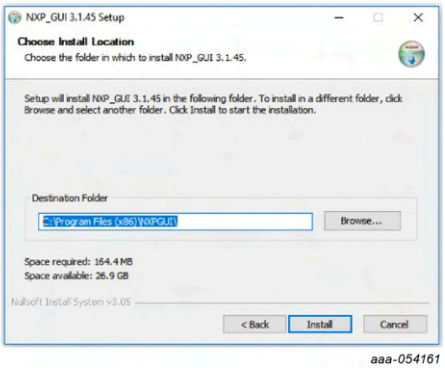
To install the FS2400 NXP GUI, download or obtain the NXP GUI package, unzip the package and double click the NXP_GUI_version-Setup.exe and follow the instructions. Proceed with the following pop-up windows to install the application on a Windows PC:

aaa-054159



aaa-054160



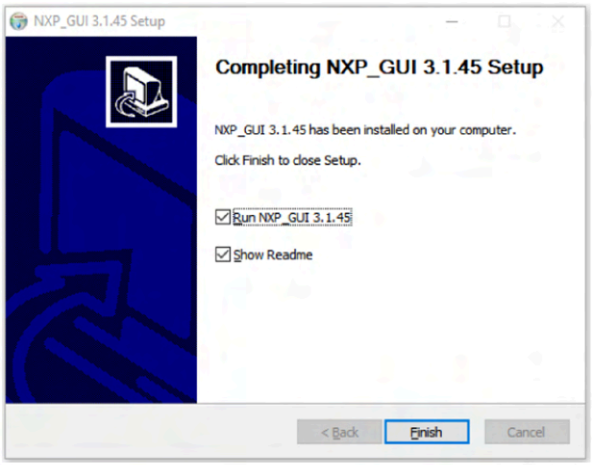


aaa-054161

Select the following options before completing the installation of the setup:

- Run NXP_GUI
- Show Readme

Click **Finish** to complete the installation



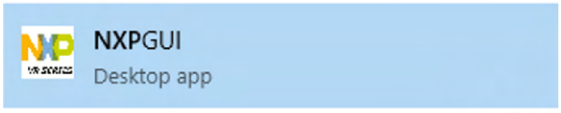
aaa-054162

When the installation is finished, search for "NXPGUI" in the windows search bar. Click to launch.

5.3 Launching the FS24 NXP GUI

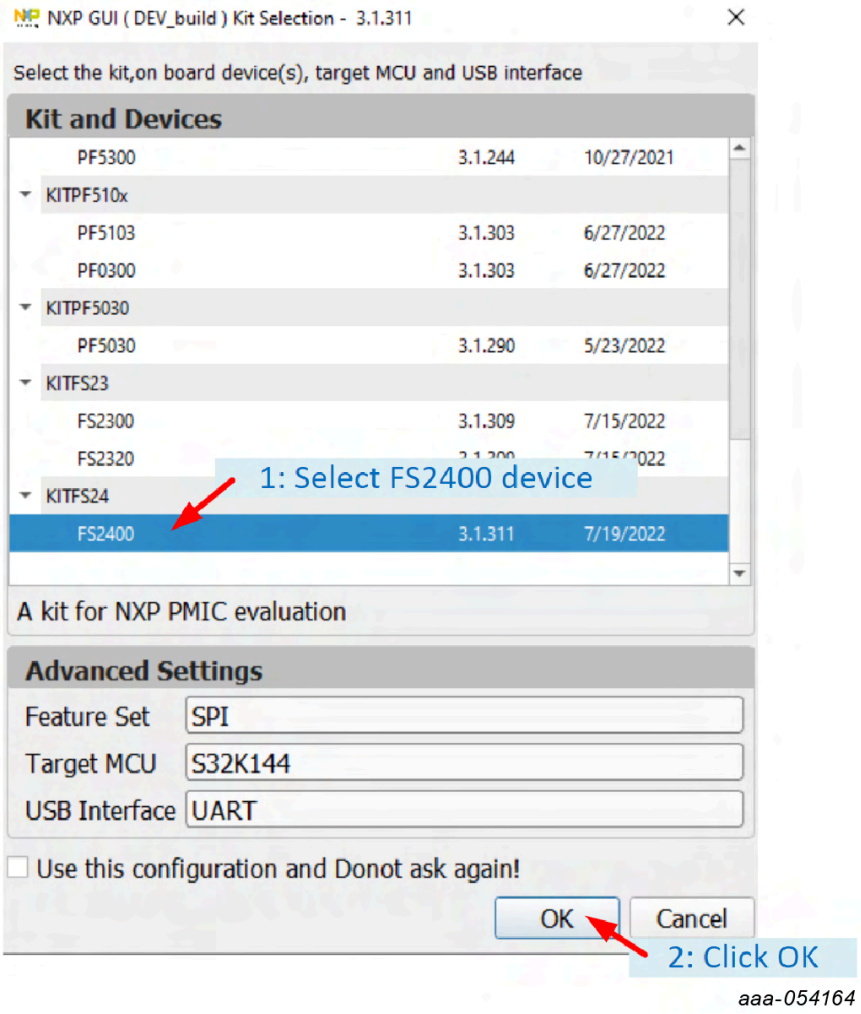
When the KITFS24SKTFDMEVM kit is set up and the GUI installed, follow the steps below to launch the GUI:

1. Click the **Windows** icon (bottom left corner) and locate "NXPGUI" in the Windows All Apps bar, then click the NXPGUI icon to launch the GUI.



aaa-054163

2. When the GUI opens, the first window to appear is the Kit Selection window. In the Kit Selection window, select the settings shown below. When finished selecting the settings, click **OK**.
   To avoid the Kit Selection window on every launch, check the box “Use this configuration and do not ask again".



Note: *The Kit Selection window can be enabled/disabled through the File main menu item once the GUI is launched by checking/unchecking the "Do not display GUI Kit Selection at Start" box.*

6 Configuring the hardware and software

6.1 Setting up the KITFS24SKTFDMEVM

The procedure for setting up the KITFS24SKTFDMEVM board is as follows:

1. Make sure the board has the jumpers and switches configured in their default positions. The default debug configuration enables the board to be fully controlled by the S32K144 MCU (via SPI) and the GUI. Section 4.3 shows the default jumpers and switches configurations.
2. Connect the power supply to J6 (Phoenix connector - 3.81 mm). The power supply should be set to a nominal value of 14 V.
3. Make sure that the USB cable between the board and the PC is securely connected. This connection is critical because the USB port serves as a communication channel between the PC and the S32K144 MCU onboard and provides voltages and references to some onboard circuits.
4. Place SW1 in top position (2-3)

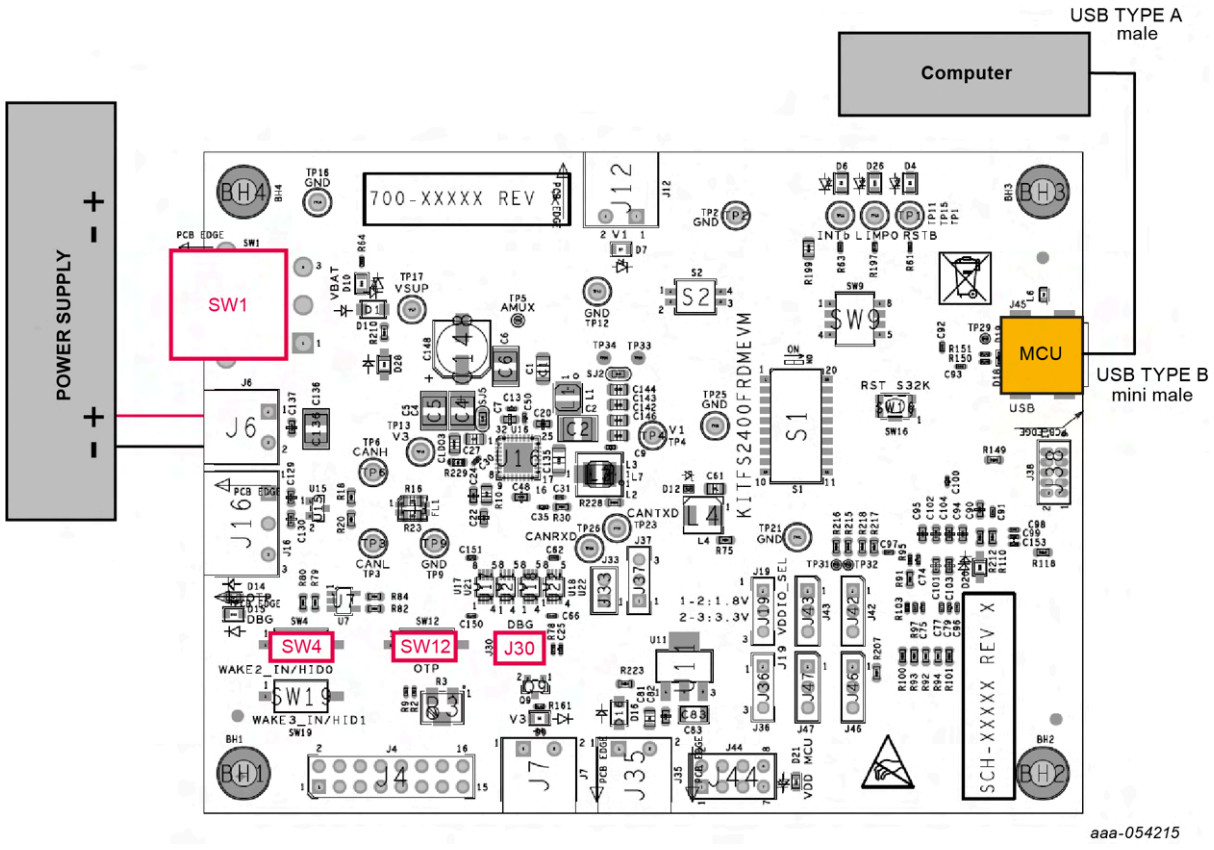


Figure 20. Power supply

6.2 Connecting the KITFS24SKTFDMEVM to the GUI

The procedure for connecting the KITFS24SKTFDMEVM to the GUI is as follows:

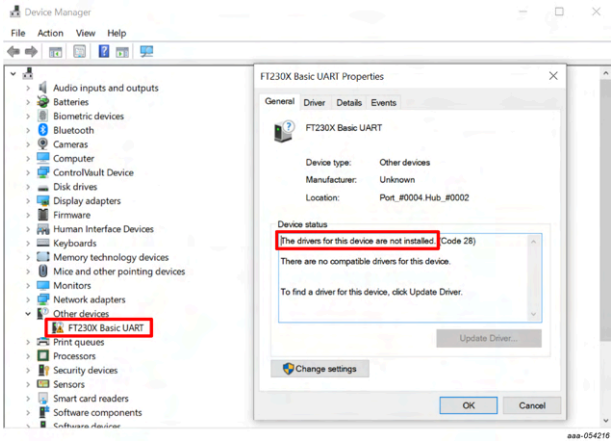
1. To launch the NXP GUI application, see Section 5.3.
2. In the USB and Device Status bar at the bottom of the GUI window, the State message should display "DISCONNECTED" when the USB cable is plugged in but communication has not yet been established between the S32K144 MCU and the FS24xx SBC.

USB cable plugged in, communication not established yet	 aaa-049530
USB cable not plugged in	 aaa-049531

3. To establish communication between the S32K144 MCU and the FS24xx SBC and allow the GUI to take control, click **Start** in the Connection Toolbar in the top left corner of the GUI window. Once the communication is established, the State message becomes "CONNECTED".

USB cable plugged in, communication established	 aaa-049528

Note: *If **Start** does not light and stays gray, the FTDI driver might be missing. To confirm this, access the Windows Device Manager and look for FT230X Basic UART Device.*



aaa-054216

Note: *To solve this issue, look for the FTDI USB Serial Port driver corresponding to the PC and install it. For Dell computers, the package can be downloaded here.*

6.3 Operation modes

The KITFS24SKTFDMEVM provides three distinct operation modes with direct impact on the device's functionalities. Understanding these modes helps use the GUI properly.

Note: *When using the KITFS24SKTFDMEVM for the first time, it is recommended to start the device in Debug mode. Indeed, Debug mode disables the watchdog, making engineering and debugging easier. To start the device in Debug mode, the hardware should be configured accordingly, see Section 6.3.1.2. Once the watchdog is set to infinite window, Debug mode can be exited from the GUI.*

The voltage level on the FS2400 Debug pin is one condition for entering a given operation mode.

Figure 21 gives an overview of the device modes and actions to perform in the GUI or on the EVB to enter or exit modes.

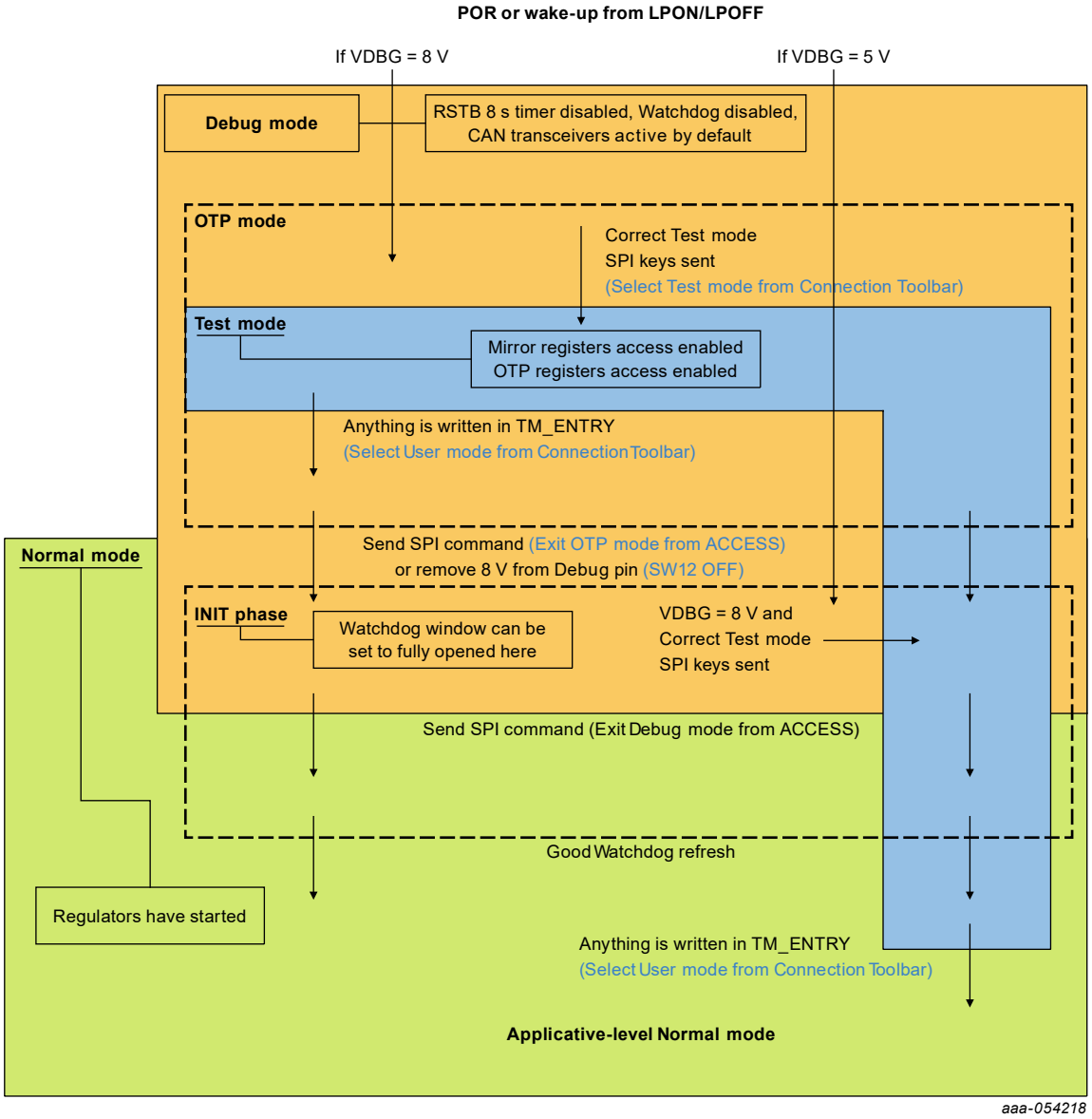


Figure 21. Operation modes

6.3.1 Debug mode

6.3.1.1 Debug mode definition

The device starts in Debug mode when the hardware is configured accordingly. This mode requires the Debug pin to be connected to VBOS through a diode to enter Debug mode automatically at startup. On the board, jumper J30 on.

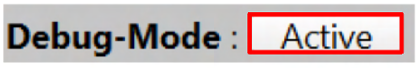
Debug mode works in parallel with Normal mode or Test mode. When Debug mode is active, the FS24 will run with limited functionalities as some safety functions are disabled.

Note: *RSTB 8 s timer is disabled, Watchdog is configured with infinite timeout and window fully opened (equivalent to disabled), CAN transceiver is set in Active mode by default.*

Debug mode is helpful during software development if the device has those functions enabled by OTP.

6.3.1.2 Debug mode activation

1. Set the default jumper configuration, see Section 6.1.
2. Verify that jumper J30 is on (allows to apply 5 V or 8 V on the pin).
3. Plug the USB cable between the PC and the board. Blue LED D14 for OTP 8 V generation is off. Green LED D15 for debug is on.
4. Apply power supply VBAT.
5. Open the GUI and start the connection.
6. Check that the GUI has detected Debug mode in the USB and device status bar:



aaa-054217

7. The equipment is now set to Debug mode.

6.3.1.3 Debug mode deactivation

To get out of Debug mode, send a DEBUG EXIT SPI request by either using the ACCESS>Main Tab>Device State>"Exit Debug Mode" checkbox, or using the DBG_EXIT bit from the M_SYS1_CFG register of the Register Map tab.

Note: *Removing the Debug Selection jumper J30 will not automatically deactivate Debug mode, as DBG 5 V on FS2400 Debug pin is only a condition for Debug mode entry, not a condition to remain in Debug mode.*

6.3.2 Test mode

6.3.2.1 Test mode definition

Test mode allows the user to write in the Mirror registers to configure or reconfigure the device for customer evaluation and to burn OTP fuses.

Note: *In case of a POR, the Mirror registers will be reset to the default OTP configuration (empty if OTP not burned).*

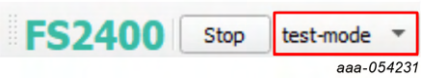
Test mode requires jumper J30 on and switch SW12 on (OTP mode is therefore enabled when Test mode is activated) and valid Test mode keys to be sent by SPI. Permanently fusing all the data stored in Mirror registers to the OTP sectors necessitates an extra key command, available in the PROG tool. The OTP fuse burning process is explained in Section 6.7.

When the Mirror registers configuration is done, the user can move to User mode to power up the device with the given configuration.

6.3.2.2 Test mode activation: hardware and software

In order to start the device with a configuration loaded in the Mirror registers, follow the instructions below:

1. Set the default jumper configuration, see Section 6.1.
2. Set switch SW12 on (apply 8 V on the debug pin).
3. Verify that jumper J30 is on (allows the user to apply 5 V or 8 V on the pin).
4. Plug the USB cable between the PC and the board. Blue LED D14 for OTP 8 V generation and green LED D15 for debug is on.
5. Apply power supply VBAT. This action loads the Mirror registers with burned OTP configuration if present.
6. Open the GUI and start the connection.
7. Select "test-mode" from Connection Toolbar:



6.3.2.3 Test mode deactivation

Once activated, Test mode can be deactivated using the GUI by selecting User mode from the Connection Toolbar. The user must apply 8 V on the Debug pin to exit Test mode.

Note: *Only switching off "OTP Mode Enable SW12" without changing mode in GUI will not automatically deactivate Test mode, as OTP 8 V on FS2400 debug pin is only a condition for Test mode entry/exit and not a condition to remain in Test mode.*

6.3.2.4 Test mode operation

Operating in Test mode allows the user to create and test a preliminary version of a desired configuration in the Mirror registers, prior to submitting the configuration to the OTP fuse burning process.

There are two ways to load Mirror registers:

- With a TBB script via SCRIPT tool (see Section 6.6.1),
- With the MIRROR tool (see Section 6.6.2).

These methods are functional with an empty or burned part.

6.3.3 User mode

6.3.3.1 User mode definition

The User mode operation is used for final product test in an automotive environment.

User mode entry at power-on reset requires 0 V on the FS2400 debug pin.

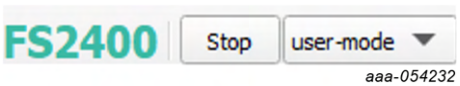
When User mode is active, the device cannot access mirror registers content or perform OTP-related operations (emulation or programming).

6.3.3.2 User mode activation

Section 6.3.3.2.1 shows how to activate User mode:

6.3.3.2.1 With the GUI using mode selection in Connection Toolbar

1. Set the default jumper configuration, see Section 6.1.
2. Plug the USB cable between the PC and the board.
3. Apply power supply VBAT.
4. Open the GUI and start the connection.
5. Check that the GUI started in User mode in the Connection Toolbar:



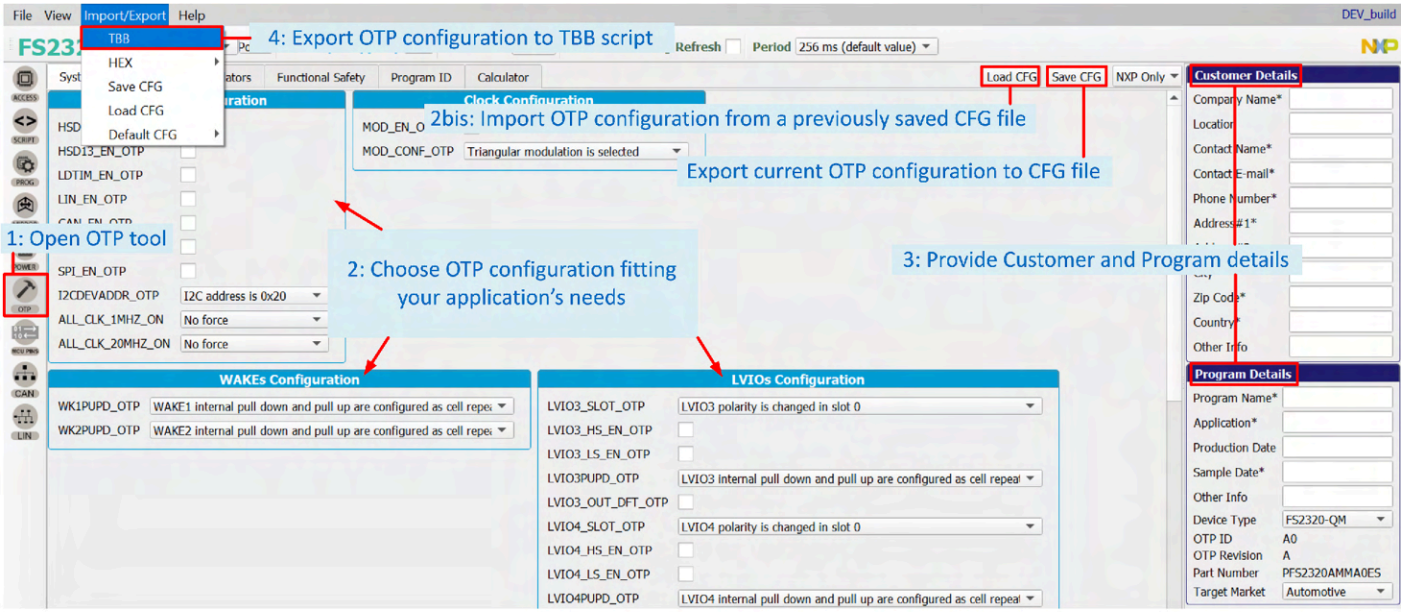
6. The equipment is set to User mode.

6.4 Generate a TBB script

A TBB file is needed to burn an OTP fuse or to emulate an OTP configuration by filling the Mirror registers through the Script tool.

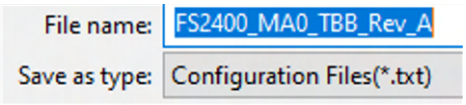
To generate a TBB file, use the following procedure:

1. Go to the OTP tool from the Tools Access Bar:



aaa-054219

2. Configure the OTP parameters to fit the application's needs or import an OTP configuration from a previously saved CFG file.
3. Enter customer and program details in the window on the right side.
4. When done, export the OTP configuration directly to a TBB script by clicking **Export** in the Framework settings bar and choosing **TBB**:
5. Select the desired folder to save the TBB script:



aaa-054220

6. TBB is now generated and saved, see Section 6.9.

6.5 First-start procedure: configure Watchdog as Infinite Time Out

This procedure is to be used as a first-start and allows the user to enter system-level Normal mode execution with the Watchdog configured with window fully opened (equivalent to disabled).

1. Set up and connect the KITFS24SKTFDMEVM to the GUI using Section 6.1 and Section 6.2.
2. If the FS2400 part is empty or if the configuration loaded from OTP should be modified:
   a. Enter Test mode using Section 6.3.2.2.
   b. Load a configuration in the MIRROR tool, then write the configuration in the Mirror registers.
   c. Select User mode from Connection Toolbar to exit Test mode.
3. Exit the OTP mode by switching OFF SW12.
4. Go to ACCESS Main Tab and click **Read all** to check that the device is in Normal and Debug mode.
5. Go to ACCESS WatchDog and configure watchdog with Infinite Time Out.

6. Exit Debug mode from ACCESS Main Tab in Device State box.
7. Under the WatchDog tab beside Good watchdog refresh, write "WD Answer Good" to get out of INIT state.

Figure 22 shows the sequence of steps (from step A to step I) to perform in ACCESS tool:

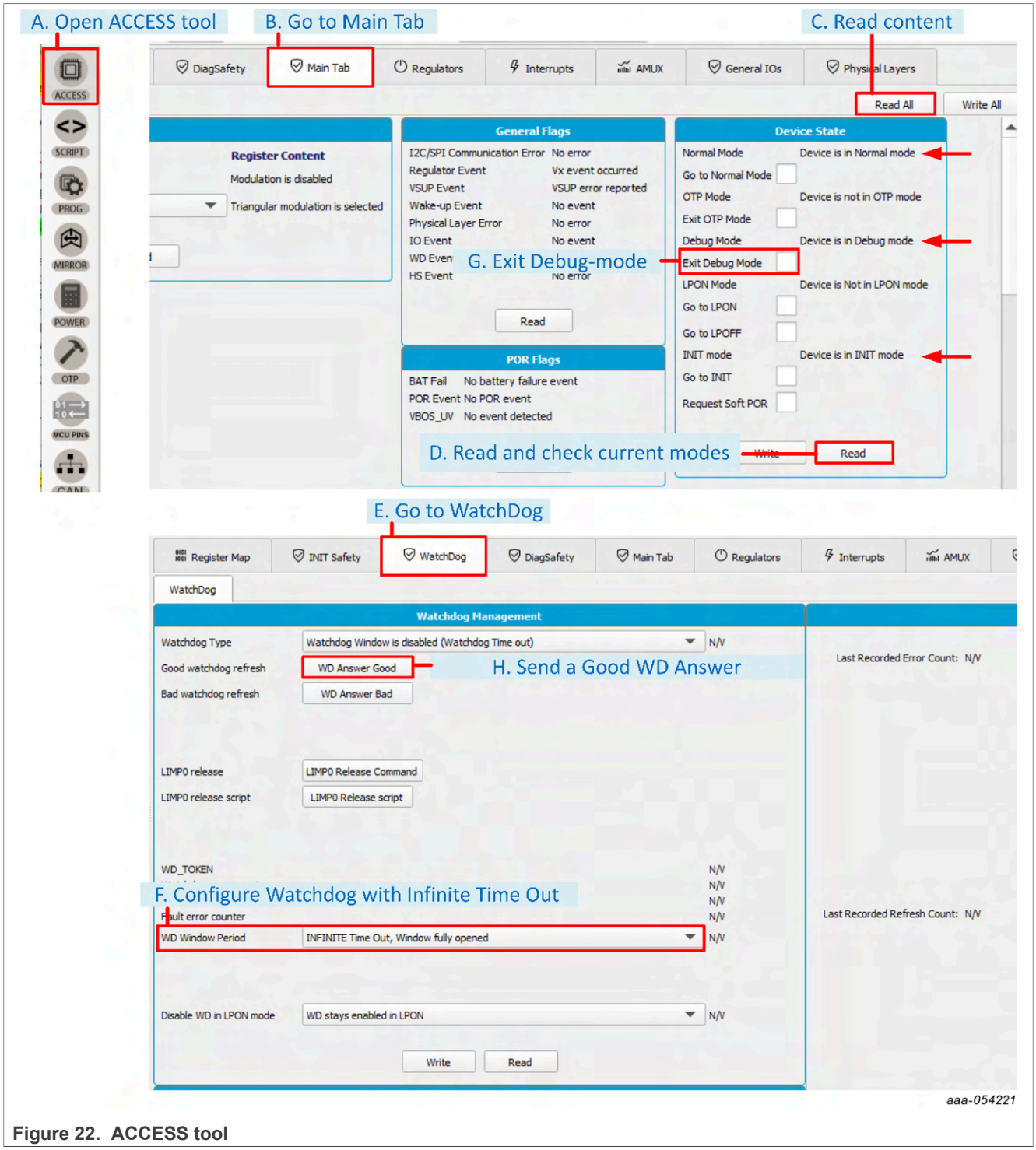


Figure 22. ACCESS tool

6.6 Operate OTP emulation by loading configuration in the Mirror registers

Mirror registers are where the OTP-fused registers are loaded at each FS2400 boot up. OTP emulation mode allows the user to directly write to Mirror registers to emulate any OTP-fused configuration. Mirror registers can be read/written multiple times, whereas OTP registers can only be burned once.

Mirror registers can be read and written in Test mode only, see Section 6.3.2.

In case of a power-on reset, the Mirror registers are reset to the default OTP configuration (default if OTP sectors are not burned).

The MIRROR tool provides access to all Mirror registers with a similar disposition to the OTP tool.

6.6.1 Modify the Mirror registers with a TBB script

From an active Test mode environment, the Mirror registers can be configured using the SCRIPT tool, see Section 7.5.4.

1. In the SCRIPT tool, click **OPEN** in the Script bar to load the TBB script file into the Script Command Window.
   Note: *The script file must have an FS240x prefix, with x being the part version.*

aaa-049539

2. To execute the TBB script, click **RUN** in the bar at the bottom of the Script Command Window:

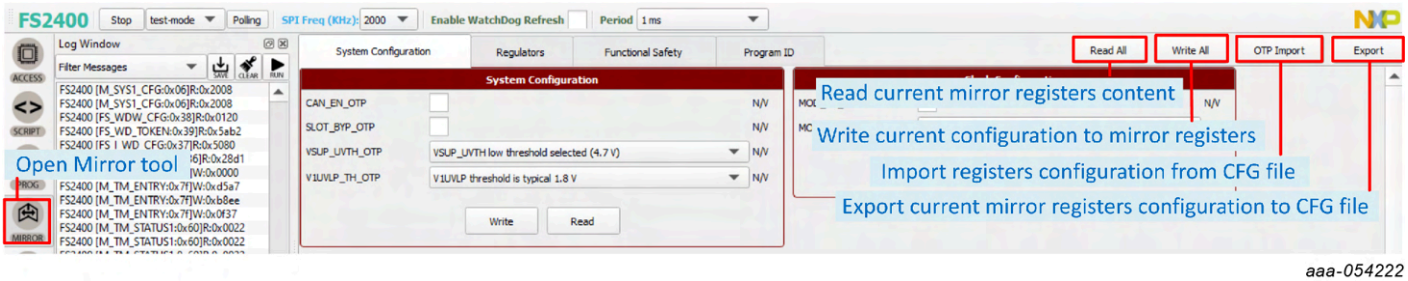
aaa-049540

3. Deactivate Test mode to start the device with a given configuration and load the SPI functional registers with the Mirror registers content.
   As a cross-check, for example, open the ACCESS tool and check the fields in the Regulators tab. If the operation completed without a problem, all fields should display the expected configuration.

6.6.2 Modify the Mirror registers with the MIRROR tool

To configure the Mirror registers, the MIRROR tool can be used to write/read directly into Mirror registers, see Section 7.5.5). This requires Test mode to be activated.

1. Open the MIRROR tool from the Tools Access Bar:



aaa-054222

2. Configure Mirror registers to fit the application or load an existing configuration by importing a CFG file.
   Note: *It is possible to load a configuration from a previously made configuration saved as a CFG file using* ***Import from CFG*** *at the top right of the Mirror tool.*
3. Once done, click **Write All**. As a cross-check, click **Read all**. If the operation completed without a problem, all fields should display the expected configuration.

4. Exit OTP mode to start the device.
   Note: *It is possible to save a current mirror registers configuration as a CFG file using* ***Export from CFG*** *at the top right of the Mirror tool.*

6.7 Programming an OTP operation

The PROG tool is used to permanently burn the FS2400 fuses with the customer's OTP configuration from a TBB script file. See Section 7.5.3 for further information.

The TBB script can be generated from an OTP configuration. See Section 6.4.

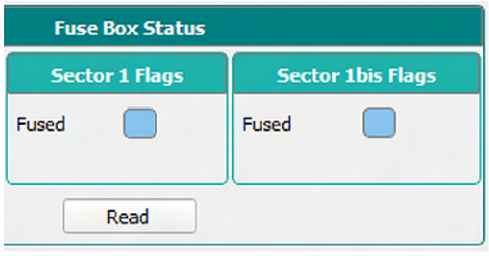
Note: *The TBB file must have FS240x as a prefix, with x corresponding to part version.*

An FS2400 device can be burned just twice.

Requirement: Make sure the socket contains a device that has not yet been burned before starting the burning process.

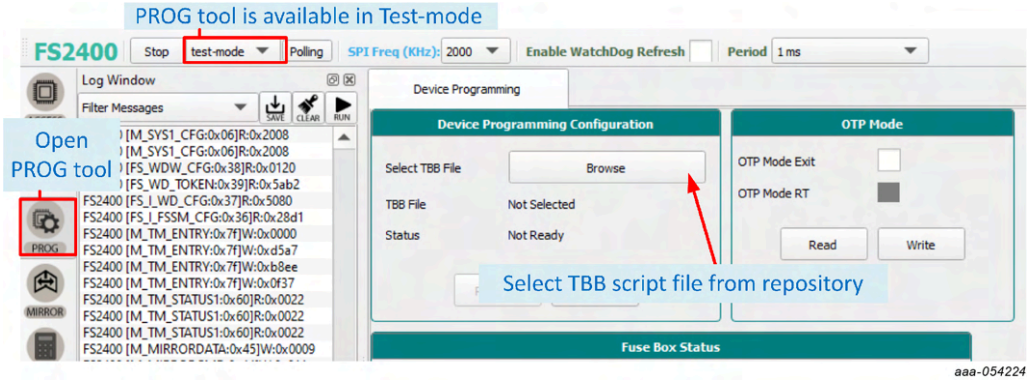
Follow this procedure to verify that the part is empty:

1. Set the default jumper configuration, see Section 6.1.
2. Set switch SW12 on (applies 8 V to debug pin).
3. Verify that jumper J30 is on (enables Debug mode/OTP mode entry).
4. Plug the USB cable between the PC and the board. Blue LED D14 for OTP 8 V generation and green LED D15 for debug is on.
5. Apply power supply VBAT (this action loads the mirror registers with burned OTP configuration if present).
6. Open the GUI and start the connection.
7. Switch to Test mode.
8. Open the PROG tool from the Tools Access Bar.
9. Check the flags in the Sector Flags section of the Fuse Box Status window, see Section 7.5.3.3 for further information.
   The device is empty and can still be programmed two times when Sector 1 and 1bis flags are set to 0 (blue).



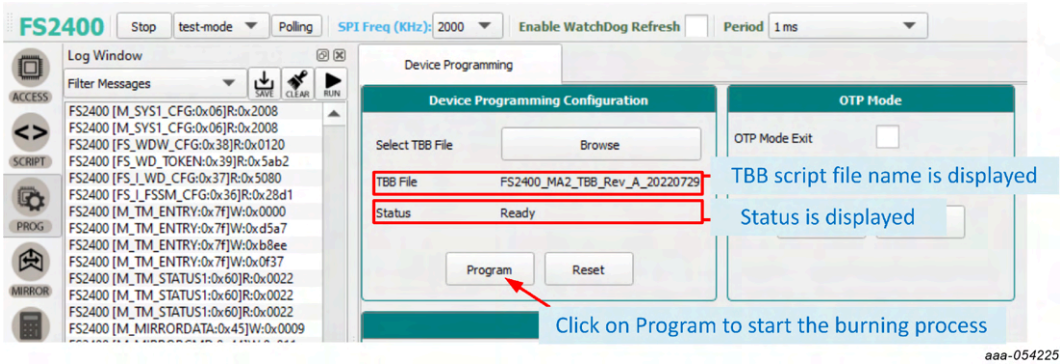
aaa-054223

   Sector flags are set to 1 (orange) once the given sector is burned.



aaa-054224

10. Load the TBB file.



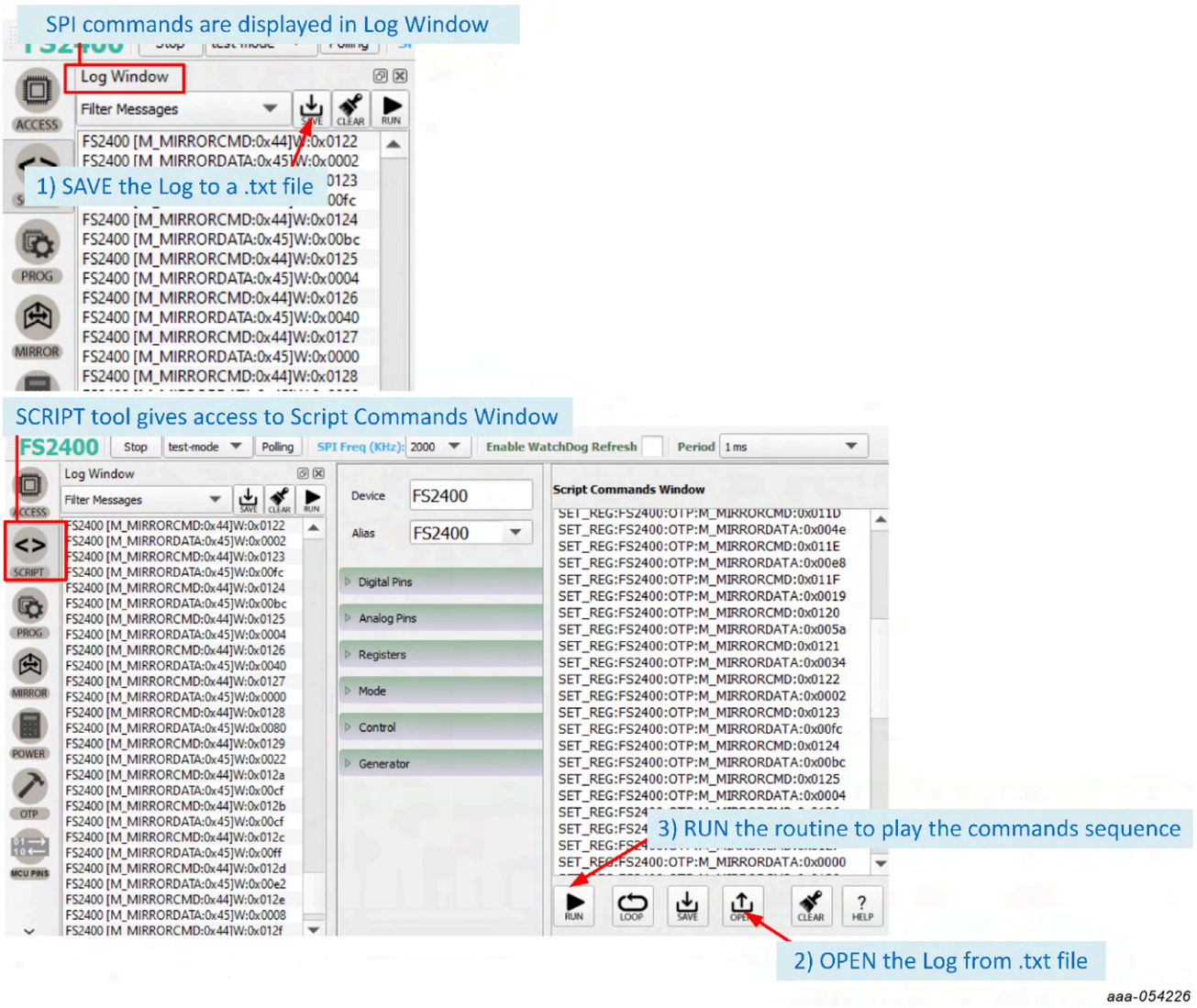
11. Click **Program**.
12. One pop-up window will appear asking for confirmation of burning. Confirm to launch the burning process. At the end of OTP programming, change the jumper configuration and restart the device completely before going to User mode.
    A good check is to use the AMUX panel in the ACCESS tool to see if all regulator voltages are correct.

6.8 Save a routine from Log window then Run it as a Script

The Log window shows the requests and answers transiting between the S32K144 MCU and the FS24 device. The commands are sent using SPI protocol depending on the user's choice.

Requests are sent to the FS24 after a user action on the graphical interface and answers are received by the MCU, then displayed to the user in the graphical interface. These exchanges are stored in the Log file.

When the user needs to operate the same routine multiple times on the device (for example, to automate loading the Mirror registers, as shown in the example below), it is possible to record the actions from the Log file, then run the routine later as a Script.



In the .txt file, it is possible to add:

- A delay between successive commands using DELAY:xx command with xx the delay in milliseconds
- A pause in the script (for example to have time to change hardware configuration between two commands) using PAUSE command
- A comment in the script using // at line start

6.9 TBB script example

This TBB script corresponds to FS2400 HVBUCK ASIL B version an is given as an example code.

```
//FS2400 - OTP Editor
//file generated on Thu Oct 19 13:57:03 2023
//Device Type : QM
//OTP ID : A1
//OTP Revision : B
//Part Marking : PFS2400AVMA1ES
//Customer : NXP
//Write main registers
//Test mode entry
SET_MODE:FS2400:test-mode
//Verify test mode entry
GET_REG:FS2400:M_TestMode:M_TM_STATUS1
//Configure OTP Mirror Registers
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0008
SET_REG:FS2400:OTP:M_MIRRORCMD:0x011C
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0001
SET_REG:FS2400:OTP:M_MIRRORCMD:0x011D
SET_REG:FS2400:OTP:M_MIRRORDATA:0x003a
SET_REG:FS2400:OTP:M_MIRRORCMD:0x011E
SET_REG:FS2400:OTP:M_MIRRORDATA:0x00a8
SET_REG:FS2400:OTP:M_MIRRORCMD:0x011F
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0017
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0120
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0057
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0121
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0034
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0122
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0012
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0123
SET_REG:FS2400:OTP:M_MIRRORDATA:0x00fc
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0124
SET_REG:FS2400:OTP:M_MIRRORDATA:0x00bc
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0125
SET_REG:FS2400:OTP:M_MIRRORDATA:0x000c
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0126
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0040
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0127
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0000
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0128
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0080
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0129
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0022
SET_REG:FS2400:OTP:M_MIRRORCMD:0x012A
SET_REG:FS2400:OTP:M_MIRRORDATA:0x00cf
SET_REG:FS2400:OTP:M_MIRRORCMD:0x012B
SET_REG:FS2400:OTP:M_MIRRORDATA:0x00cf
SET_REG:FS2400:OTP:M_MIRRORCMD:0x012C
SET_REG:FS2400:OTP:M_MIRRORDATA:0x00f0
SET_REG:FS2400:OTP:M_MIRRORCMD:0x012D
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0022
SET_REG:FS2400:OTP:M_MIRRORCMD:0x012E
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0000
SET_REG:FS2400:OTP:M_MIRRORCMD:0x012F
SET_REG:FS2400:OTP:M_MIRRORDATA:0x0030
SET_REG:FS2400:OTP:M_MIRRORCMD:0x0130
//OTP Command CRC_Fill + GO
SET_REG:FS2400:M_OTP:M_OTPCMD:0x0125
//OTP Command CRC_Check + GO
SET_REG:FS2400:M_OTP:M_OTPCMD:0x0124
//Verify test mode entry
GET_REG:FS2400:M_TestMode:M_TM_STATUS1
//----------------------------- END MAIN -----------------------------
```

7 Tool interface (GUI) description

7.1 Framework window

The Framework window consists of the following sections:

- **Connection Toolbar:** Used to start communication with the device, enter or exit User/Debug/Test modes, fix SPI frequency and main address, enable watchdog.
- **Framework settings:** Manages file import/export and Framework configuration.
- **Window log:** Reports USB and Device communication events.
- **USB and Device status:** Indicates if USB or Device is connected or disconnected, shows communication protocol (SPI), shows firmware and GUI version, displays current device mode.
- **Tools access bar:** Provides quick access to the FS24 evaluation tools and features.
- **Tab content:** Shows the content of each tool or tab. There may be several tabs, boxes, or windows inside.

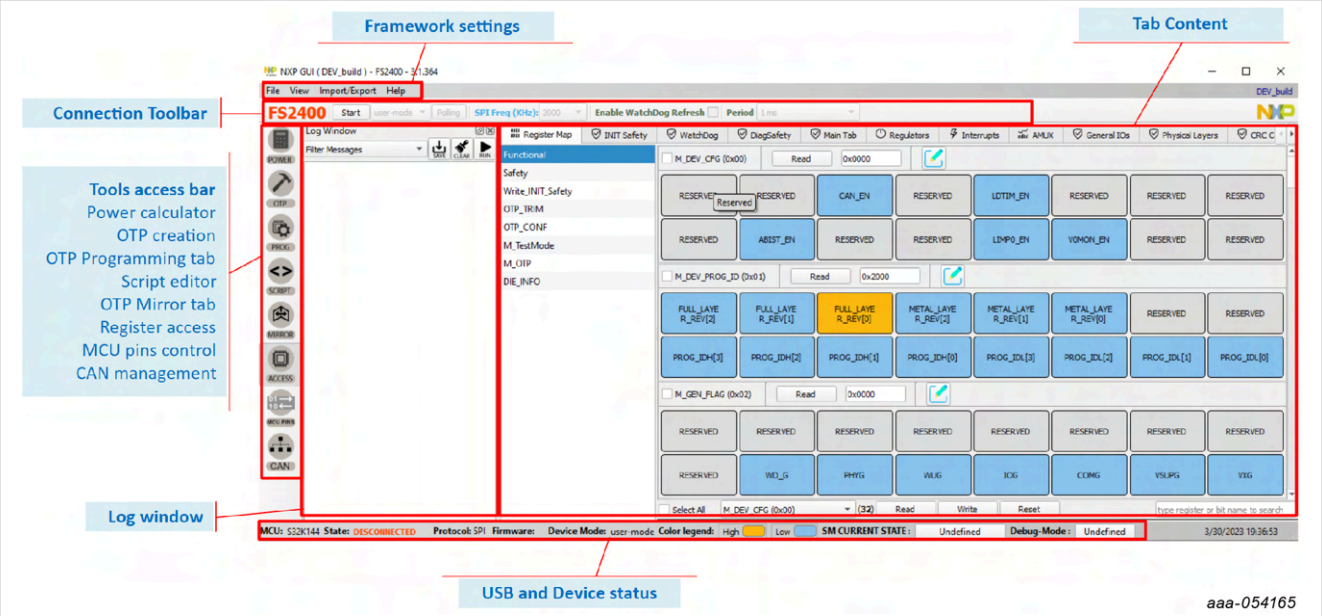
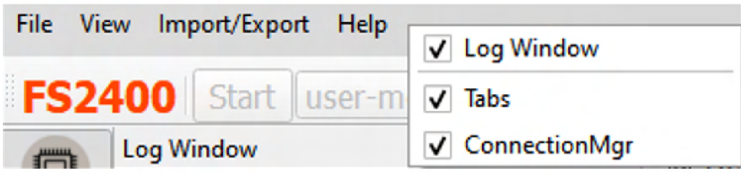


Figure 23. Framework window

If Log window, Tools Access Bar, or Connection Toolbar do not appear on the screen, right-click on the Framework settings bar. This action displays three selection boxes (checked if display is active):

- Log window
- Tabs correspond to the Tools Access Bar
- ConnectionMgr



7.2 Framework settings

The Framework settings section appears at the top left corner of the Framework window. It consists of four items:

- File
- View
- Import/Export
- Help

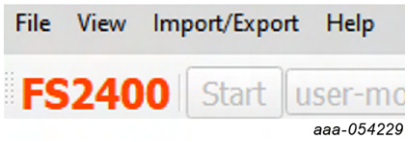


Figure 24. Framework setings

7.2.1 File menu item

The file menu item is used to set the current GUI as default or exit the application.

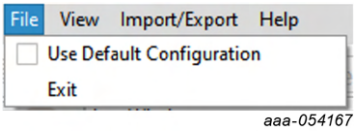


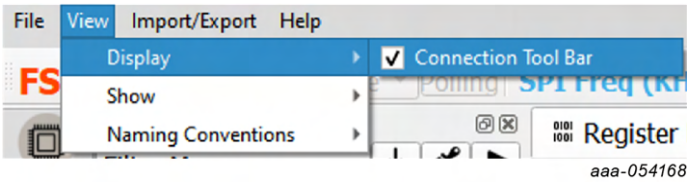
Figure 25. File menu item

- **Use Default Configuration:** Check the box to always open the NXP GUI as the current GUI version. Uncheck the box to display the product selection box at the NXP GUI startup.
- **Exit:** Exits the NXP GUI application.

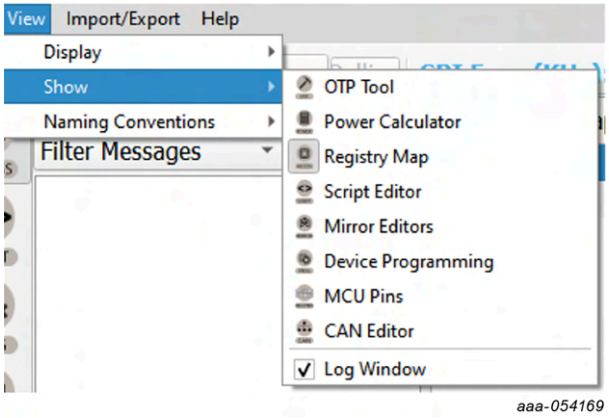
7.2.2 View menu item

The View menu item contains the following options:

- **Display**
- **Show**
- **Naming Conventions**
  - **Display:** Allows the user to enable or disable the Connection Toolbar (enabled by default).

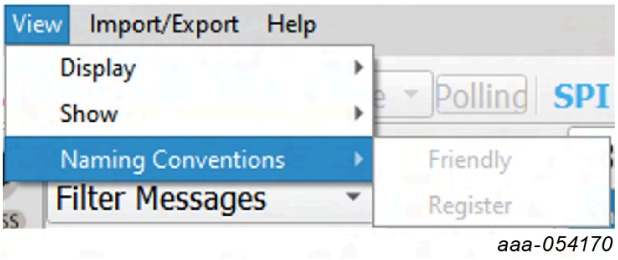


  - **Show:** Allows the user access to various sections of the GUI and displays the Log window.



aaa-054169

– **Naming Conventions:** Allows the user to select Friendly or Register name display for the OTP tool. This option is enabled only when the OTP tool is active.

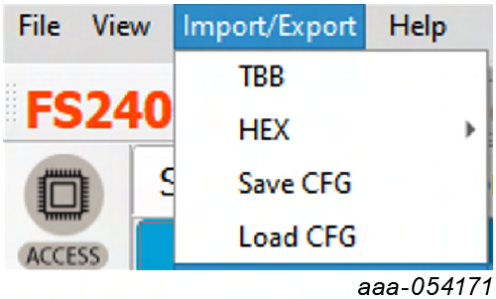


aaa-054170

– The naming convention options are:
  – **Friendly:** Register names are displayed as user-friendly names in the OTP tool.
  – **Register:** Register names are displayed by their technical names in the OTP tool.

7.2.3 Import/Export menu item

The Import/Export menu item allows the user to manage the files needed for Mirror emulation, for the PROG tool, and for GUI configuration. This menu item is only active when the OTP tool has been selected. See Section 7.5.2 for details.



aaa-054171

- **TBB:** Exports the OTP configuration into a TBB script file that can be used to load the Mirror registers using the SCRIPT tool. The same file can be used by the PROG tool to burn OTP fuses.
- **HEX:** Outputs the OTP configuration in I-HEX (Intel Hex) or S-HEX (Simple Hex) script file format.
- **Save CFG:** Used to save the current configuration as a CFG file.
- **Load CFG:** Used to load a previously saved configuration from a CFG file.
- **Default CFG:** Used to load a predefined OTP configuration in QM or ASIL B to use as a starting point.

7.2.4 Help menu item

The Help menu item contains links to additional information, allows the user to display GUI and firmware version numbers, and contains a glossary for acronyms.

- **Documentation:** Lists online NXP documentation related the FS2400 GUI.
- **About:** Displays the version number of the GUI currently installed.
- **Glossary:** Lists expanded names for acronyms used in the GUI.

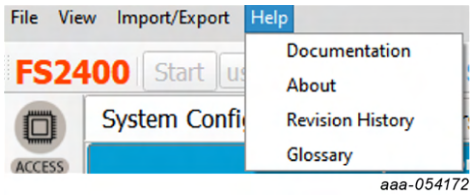


Figure 26. Help menu item

7.3 Connection Toolbar

The Connection Toolbar menu is directly below the Framework settings menu at the top left of the Framework window.

Note: *The Connection Toolbar is not displayed if not selected in the Frameworks settings>View>Display menu item.*

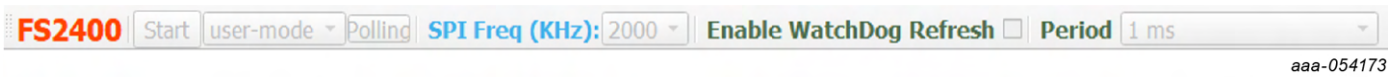
The Connection Toolbar allows the user to start or stop communication with the device, enter or exit User/Debug/Test modes, fix the SPI frequency (depending on the chosen MCU communication protocol at start), and enable watchdog.

The Connection Toolbar consists of the following items:

- **Start/Stop**: Open or close communication with the device
- **Mode:** Select between Debug mode, Test mode, and User mode
- **Polling:** When on, the GUI detects a mode change by polling mode status
- **SPI Frequency:** Set the SPI frequency in kHz
- **Enable Watchdog Refresh:** Enable/disable the Watchdog refresh
- **Period:** Set the Watchdog period

7.3.1 Device connection

The device connection boxes appear first in the Connection Toolbar menu.



When the S32K144 MCU is not connected through the USB port, the State indicator in the USB and Status bar shows "NOT DETECTED", the FS2400 header text appears red, and the **Start** button is not available.

After the USB cable is connected, the State indicator displays "DISCONNECTED" and the **Start** button becomes available.

aaa-054175

Click **Start** to start communication with the FS2400.

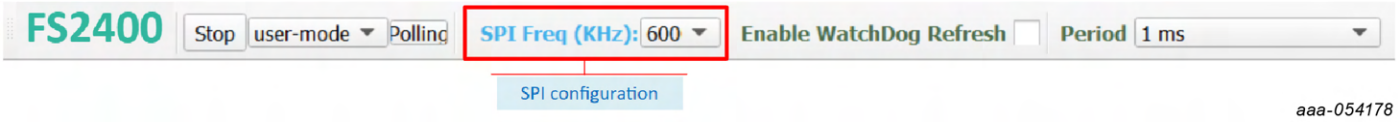
At this point, the State indicator displays "CONNECTED" and the FS2400 header text changes from red to green.

aaa-054177

Usually, once connected, the next step is to load an OTP configuration and write it in the Mirror registers.

7.3.2 SPI communication configuration

The FS2400 supports SPI communication protocol. SPI configuration settings for the MCU side appear second in the Connection Toolbar, allowing the user to choose the protocol frequency and device address, if applicable.



aaa-054178

7.3.3 Watchdog management

The FS2400 provides watchdog monitoring with keys as a functional safety feature. The watchdog feature can be disabled only in the QM version. The watchdog is always enabled for an ASIL B part.

7.3.3.1 Watchdog enablement and configuration on FS2400 side

On the FS2400 side, the watchdog can be disabled by OTP. Watchdog monitoring default configuration is done via SPI via the ACCESS tool.

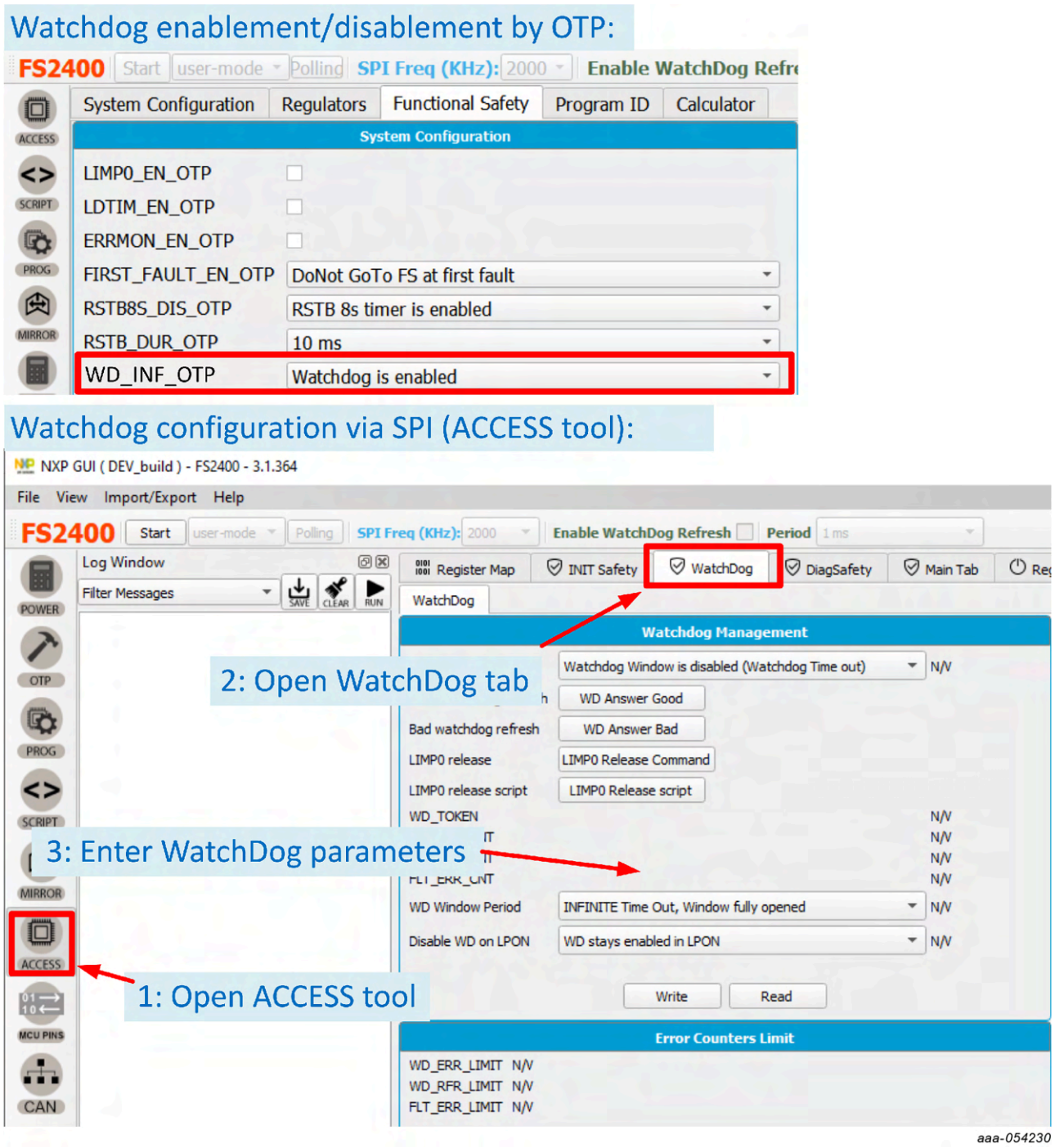


Figure 27. Watchdog enablement/disablement/configuration

7.3.3.2 Watchdog configuration on MCU side

On the S32K144 MCU side, actions are configured with the watchdog management boxes in the Connection Toolbar menu.

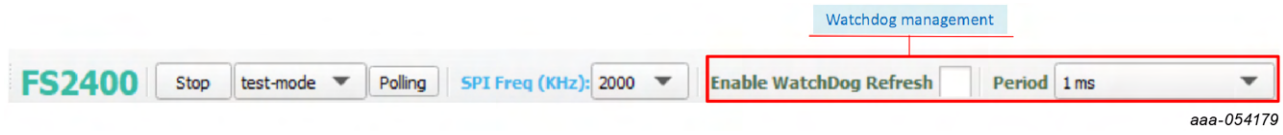


Figure 28. Watchdog management boxes

The mechanism is fully operational when both SBC and MCU actions are enabled and have matching configurations.

The steps for watchdog enablement are described in Figure 29:

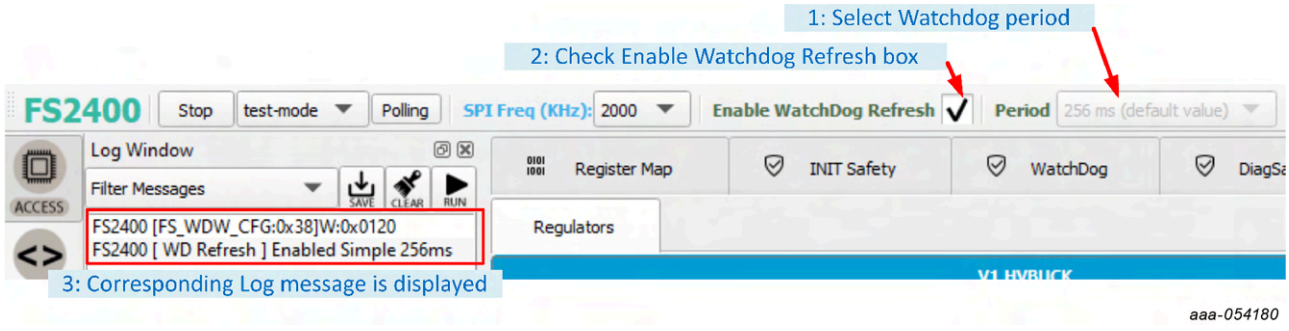


Figure 29. Watchdog enablement

1. Select the watchdog period via the Period selection box in the Connection Toolbar.
2. Enable WatchDog Refresh by checking corresponding box to enable watchdog monitoring on the MCU side.
3. A message is displayed in the Log window with the selected period and type values.

If the Period selection box is unavailable, verify that Enable WatchDog Refresh is unchecked.

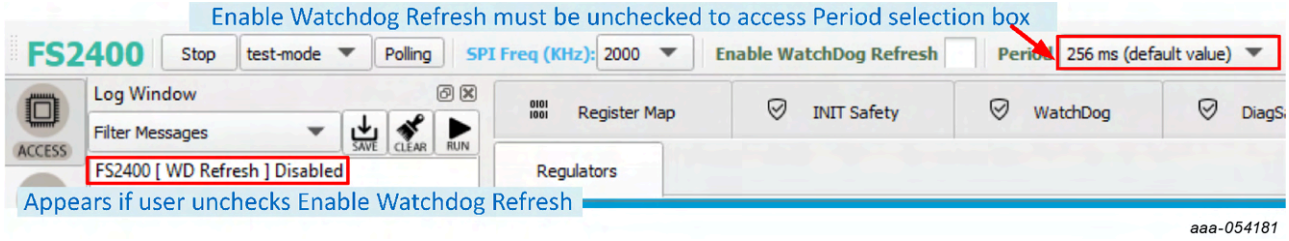


Figure 30. Enable WatchDog Refresh unchecked

7.4 USB and Device status bar

The USB and Device status bar is at the bottom of the Framework window. This toolbar gives information about the GUI and the firmware version, about the USB connection status, and about FS24 active modes.

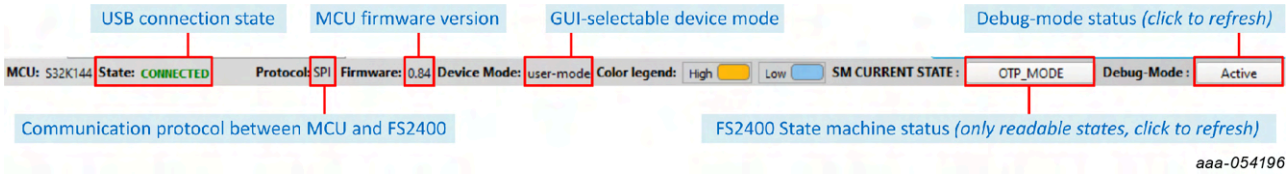


Figure 31. USB and Device status bar

7.5 Tools access bar

aaa-054182

The Tools access bar appears in a vertical row along the left side of the Framework window. The bar provides access to tools that implement various GUI functions. The Tools access bar consists of nine items:

- **POWER:** Provides a power dissipation analysis of the FS2400
- **OTP:** Used to create and save OTP configuration files, with customer and program details
- **PROG:** Used to manage OTP fuse-burning process
- **SCRIPT:** Used to create, open, save, and run scripts
- **MIRROR:** Provides access to Mirror registers
- **ACCESS:** Provides access to I^2C/SPI registers via Register Map or thematic tabs
- **MCU PINS:** Used to read and set MCU I/O pins
- **CAN:** Used to manage CAN communication

7.5.1 POWER

The POWER tool is used to compute the power dissipation of the device in a given application.

Figure 32 shows how to use the POWER tool. Input values are selected on the interface from keyboard inputs or selection boxes. Results are shown in the green boxes. Power dissipation graphs can be displayed from the interface using **System Power Dissipation** or **IC Power Dissipation** buttons. **Load Power Config** and **Save Power Config** buttons are used to resume the power dissipation calculation later, by saving and loading a text file.

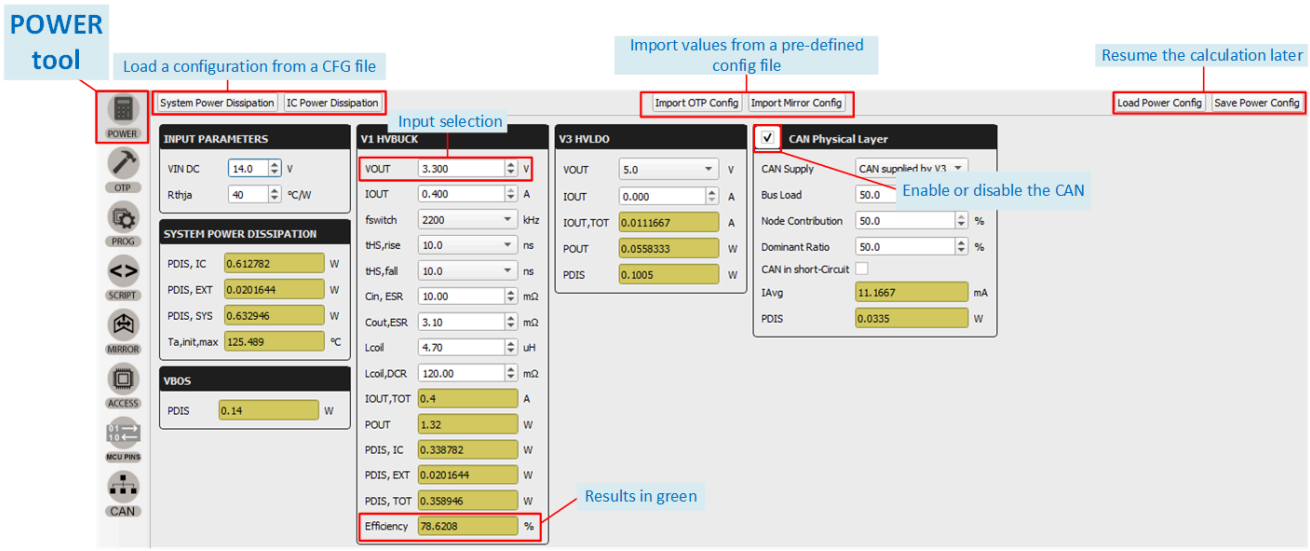


Figure 32. POWER tool

7.5.2 OTP

The OTP tool allows the user to enter an OTP configuration. The OTP configuration can be saved as a CFG file or exported as a TBB file:

- **CFG file:** Used by the GUI to log all of the configuration information. The CFG file can be used to save an OTP configuration or load Mirror registers with the MIRROR tool in OTP mode.
- **TBB file**: Can be created with a .txt extension. The TBB file can be used to load the Mirror registers with the SCRIPT tool in OTP mode or to burn OTP fuses using the PROG tool.

The OTP configuration is not addressed here. For information about OTP configuration, refer to the FS2400 data sheet.

The OTP tool's main panel is divided into two windows:

- **OTP Parameters Setting window**
- **OTP Details window**

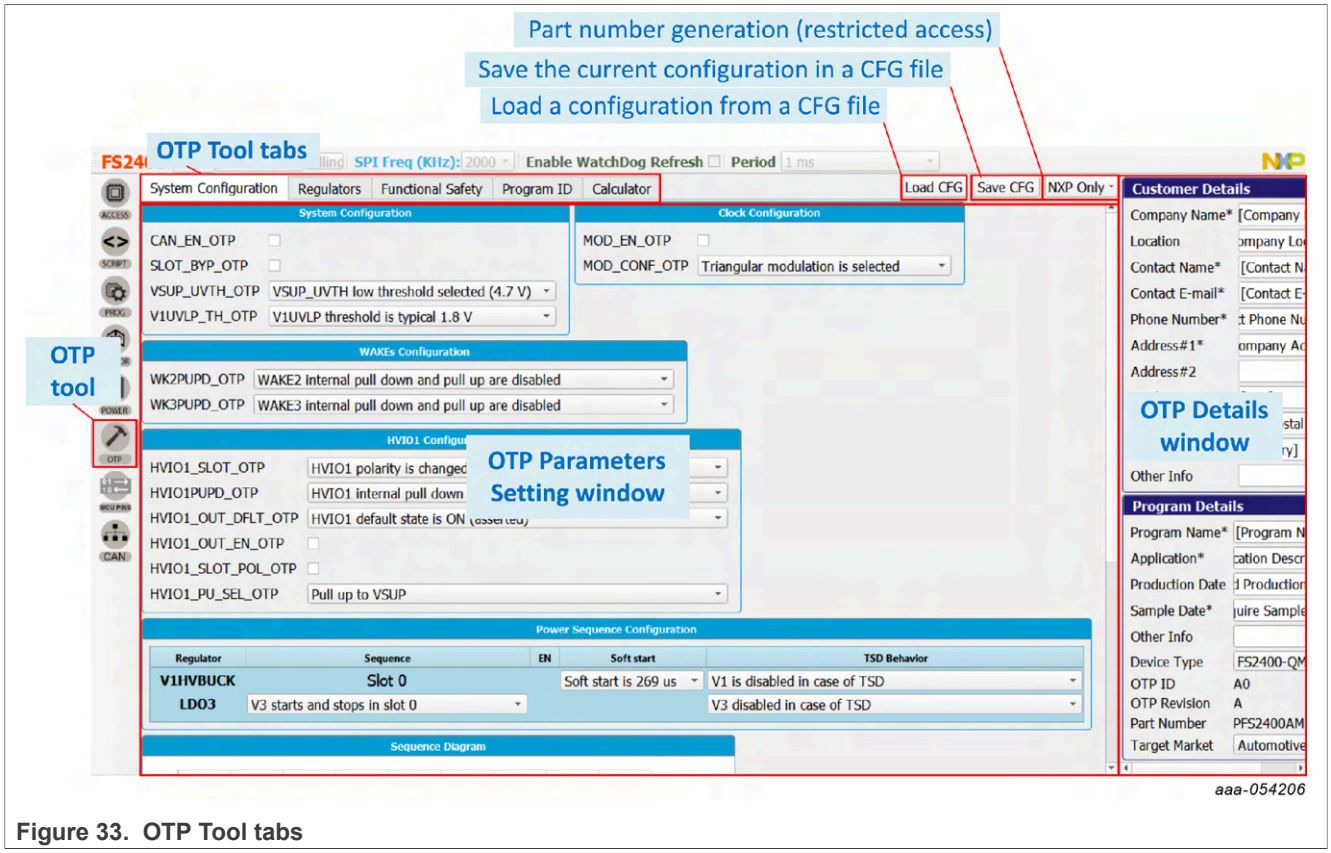


Figure 33. OTP Tool tabs

7.5.2.1 OTP Parameters Setting window

The OTP Parameters Setting section is organized into four tabs.

7.5.2.1.1 System Configuration tab

The System Configuration tab provides a means of setting miscellaneous FS2400 system configuration parameters, including clock, I/Os, and power-up sequence configuration. The tab displays the set power-up sequence as a graph in the Sequence Diagram box.

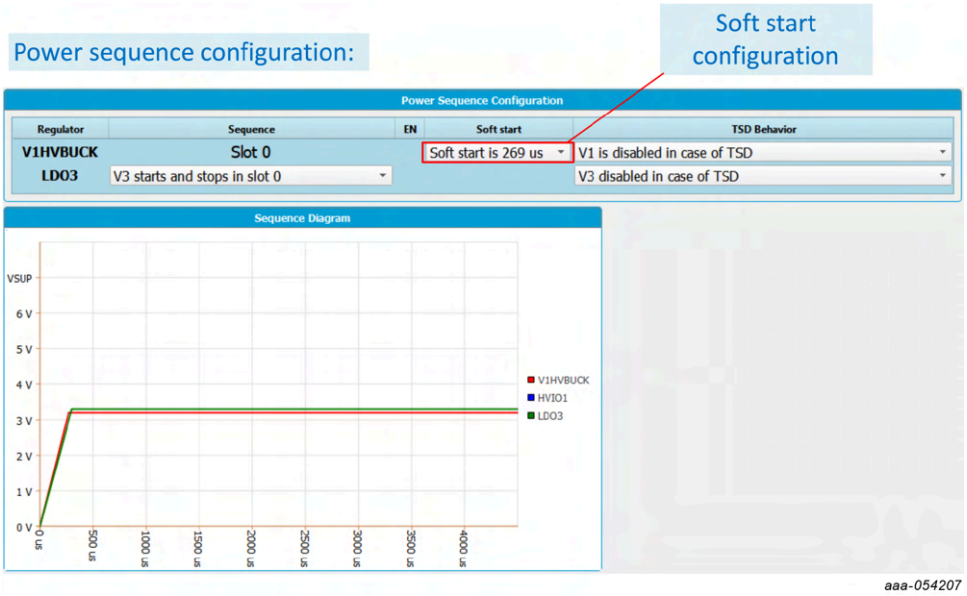


Figure 34. Power sequence configuration

7.5.2.1.2 Regulators tab

The Regulators tab allows the user to set OTP parameters for V1 - HVBUCK and V3 - HVLDO. The tab displays a simplified diagram of the selected configuration as a visual cross-check.

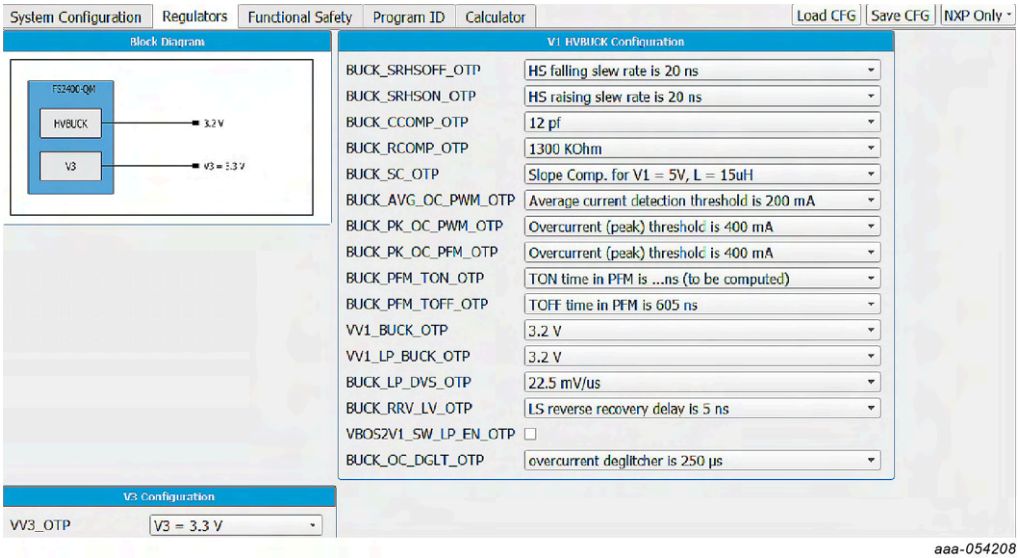


Figure 35. Regulators tab

7.5.2.1.3 Functional Safety tab

The Functional Safety tab allows the user to set OTP safety-related parameters, such as voltage monitoring, LIMP function, and other safety-related system configurations. This tab displays a different window depending on the device's safety level (QM or ASIL B) selected in the Program Details panel, see Section 7.5.2.2.

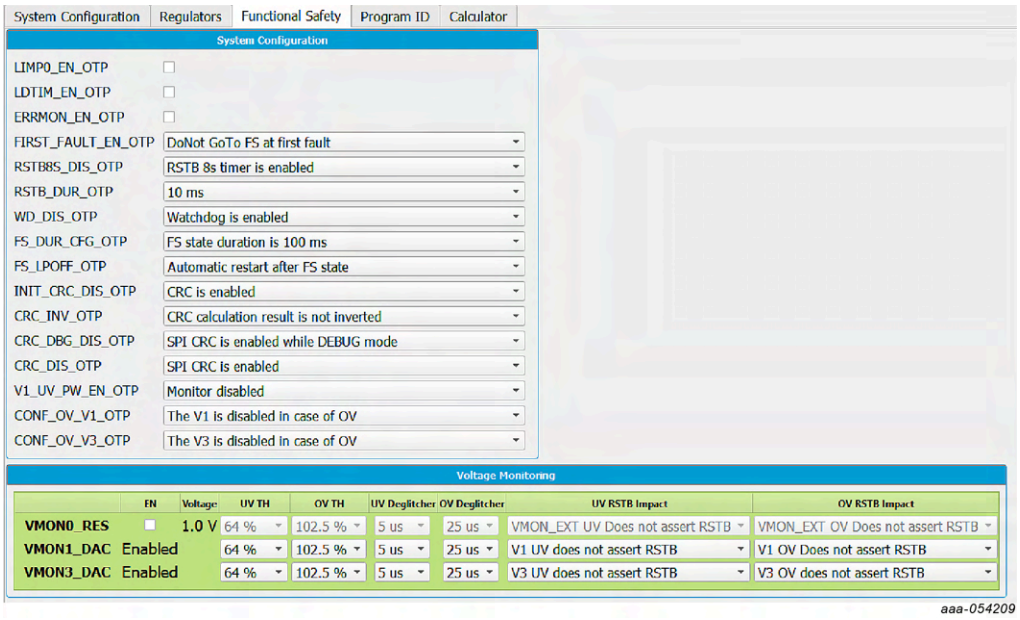


Figure 36. Functional Safety tab

7.5.2.1.4 Program ID tab

The Program ID tab displays the OTP ID. Only NXP users can create a new OTP ID.

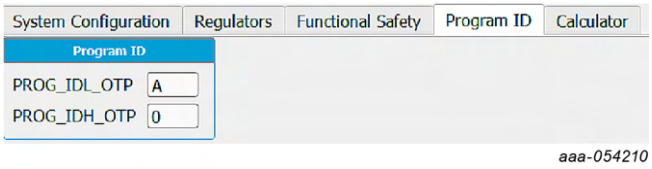


Figure 37. Program ID tab

7.5.2.2 OTP Details window

The OTP Details window collects and stores information about the customer and OTP version. All the information entered in this section will be part of the CFG and TBB files.

This section is organized into two boxes:

- **Customer Details box:** Collects and displays customer contact information.

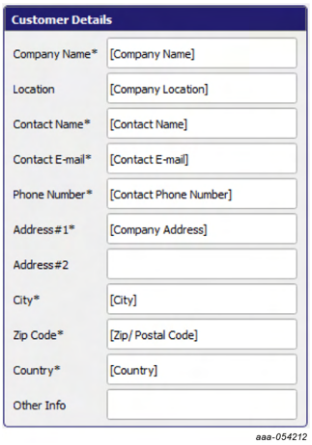


Figure 38. Customer Details box

- **Program Details box:** In addition to comments related to the application, this window allows the user to:
  - Select the device type (see Section 7.5.2.1.3 for details)
  - Display the OTP ID (set by NXP only)
  - Display the build part number (set by NXP only)
  - Select the target market, either automotive or industrial

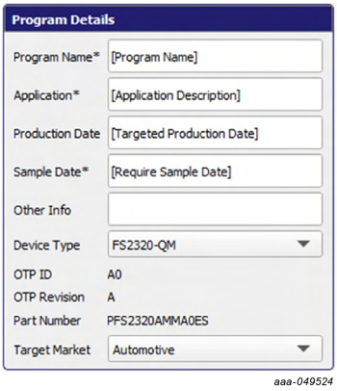


Figure 39. Program Details box

7.5.3 PROG

The PROG tool provides access to device programming configuration and tools for the OTP burning process.

Note: *A device can be programmed once. Use this tab cautiously.*

The programming tool is available only in Test mode.

The programming screen consists of three sections:

- **Device Programming Configuration window**
- **OTP Mode window**
- **Fuse Box Status window**

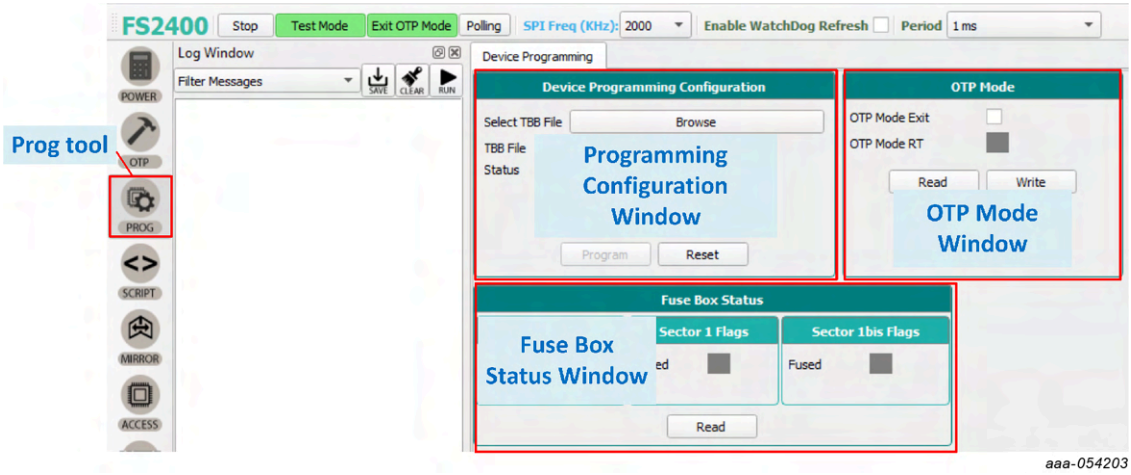


Figure 40. PROG tool

7.5.3.1 Device Programming Configuration window

This window allows the user to program an OTP operation from a saved TBB file.

To program the part by fusing OTP, see Section 6.7.

7.5.3.2 OTP mode window

The OTP mode window allows the user to:

- Exit OTP mode by sending a SPI frame.
- Verify that the FS2400 device is in OTP mode (OTP 8 V applied to DBG pin).

7.5.3.3 Fuse Box Status window

The Fuse Box Status window gives status about which sector has already been fused/burned. In the indicator boxes, blue indicates low state, orange indicates high state.

- Sector 0 is NXP proprietary.
- Sector 1 is customer OTP configuration.
- Sector 1bis is a duplicate of Sector 1 for second-time programming.

Notice that an empty part has the following fuse box status, see Figure 41.

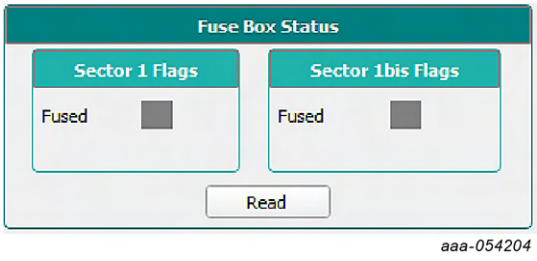


Figure 41. Fuse Box Status

7.5.4 SCRIPT

The Script editor allows the user to create new script sequences or to send existing sequences to the device. Commands include reading/writing individually to a register, to a digital pin, or to an analog pin. This tool allows the emulation of an OTP configuration.

The Script editor window consists of four sections:

- **Log Window**
- **Script Commands**
- **Script Window and its Script bar**
- **Script Results Window**

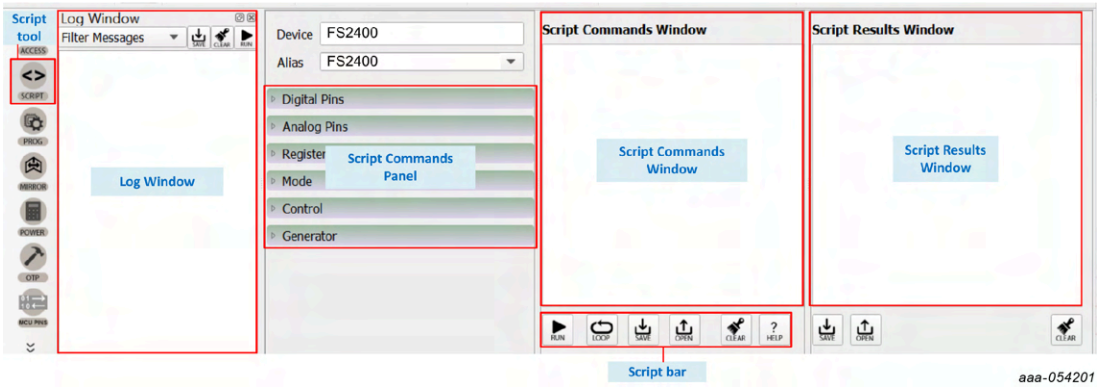


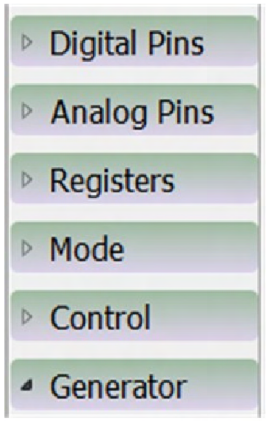
Figure 42. SCRIPT tool

7.5.4.1 Log window

The Log window lists events as they occur in real time when the script is executing. The Filter Messages box allows the user to limit log messages to certain events (Register Read, Register Write, Pin Read, Pin Write). The Log window menu bar also contains buttons for saving the log contents to a file, clearing the log, or running the script in the Script Command window.

7.5.4.2 Script Commands panel

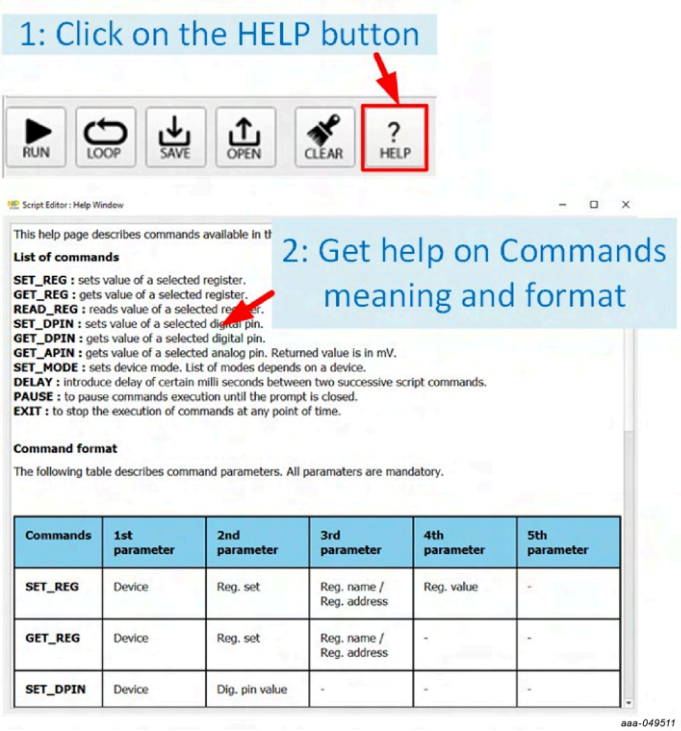
The Script Commands panel allows the user to enter commands into the Script Command window by clicking on the appropriate command. Facilitated command entry ensures error-free syntax in the command. The commands are organized into functional categories. Opening a panel tab and selecting a specific pin or register makes the associated command appear in the Script Command window.



aaa-049510

- **Digital Pins:** Used to enter a script command to read or writing the value of the selected digital pin
- **Analog Pins:** Used to enter a script command to read the value of the selected analog pin
- **Registers:** Used to enter a script command to read or write functional and safety registers
- **Mode:** Used to enter a script command setting the desired mode.
- **Control:** Used to enter a script command to pause script execution (execution halts when the Pause command is encountered and a pop-up window appears prompting to continue execution), to delay script execution (default is 300 ms, editable to any ms value in the Script Commands window), or to exit script execution
- **Generator:** Used to clear the content of the Script Commands window and enters a prepared script sequence

Note: *The **HELP** button in the Script bar gives information about Commands.*



aaa-049511

All menu items work in a similar way. Figure 43 shows a typical process using the Registers menu tab.

Clicking on the Register tab brings up the parameters panel shown in Figure 43. A Write operation to the FS_LIMP0_REL register in the Safety register group has been selected. The value 0x5401 is selected as the value to be written to the register. Hitting **Enter** with the cursor in the value field enters the command in the Script Commands window.

To apply the commands on the Script Commands window, click **RUN** in the Script bar. The sent commands are displayed in the Log window and command results are displayed in the Script Results window.

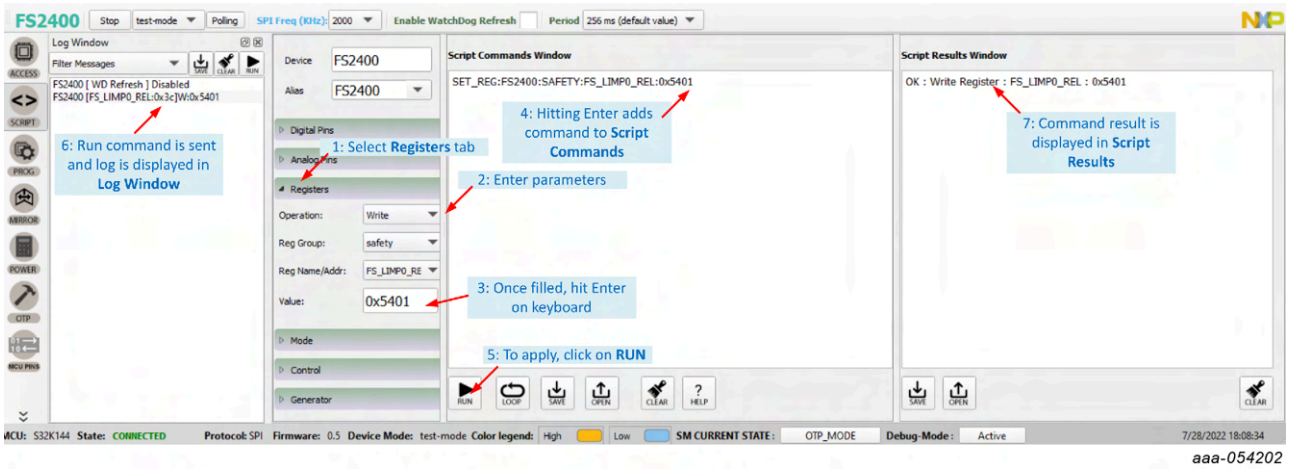


Figure 43. Parameters panel

7.5.4.3 Script Commands window

The Script Commands window is the area where existing script files can be loaded in Test mode only and where script commands are entered, edited, and executed.

Another use of the SCRIPT tool is to replay a Log command sequence, for example to run a routine. This specific use is detailed in Section 6.8.

The Script bar at the bottom of the window contains the following six buttons:

- **RUN:** Initiates execution of the script sequence in the Script Commands window.
- **LOOP:** Executes the script as a loop. Click **LOOP**, then click **RUN**.
- **SAVE:** Saves the content of the Script Commands window as a .txt file that can be subsequently reloaded.
- **OPEN:** Clears the current content and loads a previously saved script into the Script Commands window.

Note: *The loaded file has to be a TBB file with .txt extension.*

- **CLEAR:** Clears the current content of the Script Commands window.
- **HELP:** Shows a list of all script editor commands with their formats.

Figure 44. Script Commands window

7.5.4.4 Script Results window

The Script Results window displays the results of an executed script. The menu bar at the bottom of the window contains three buttons, shown in Figure 45:

- **SAVE:** Saves the content of the Script Results window as a .txt file that can be subsequently reloaded.
- **OPEN:** Clears the current content and loads a previously saved results file into the Script Results window.
- **CLEAR:** Clears the current content of the Script Results window.

Figure 45. Script Results window

7.5.5 MIRROR

The MIRROR tool provides access to all Mirror registers. Mirror registers are an emulation of OTP registers. Mirror registers can be read/written multiple times, whereas OTP registers can be burned once. In case of a power-on reset, the Mirror registers will be reset to the default OTP configuration (empty if OTP sectors not burned). Mirror registers can be read and written in Test mode only, see Section 6.3.2.

Note: *The MIRROR tool display tabs are similar to the corresponding OTP tool tab. To avoid confusion, the tab header background colors are different. The MIRROR tool tab header is red. The OTP tool tab headers are blue.*

The MIRROR tool is available only in Test mode. Test mode loads required keys in order to have full access to Mirror registers. This tool allows the user to:

- Read and write Mirror register values.
- Load a CFG file into the Mirror registers and debug the configuration. For further details on CFG file preparation and parameters configuration, see Section 7.5.2.

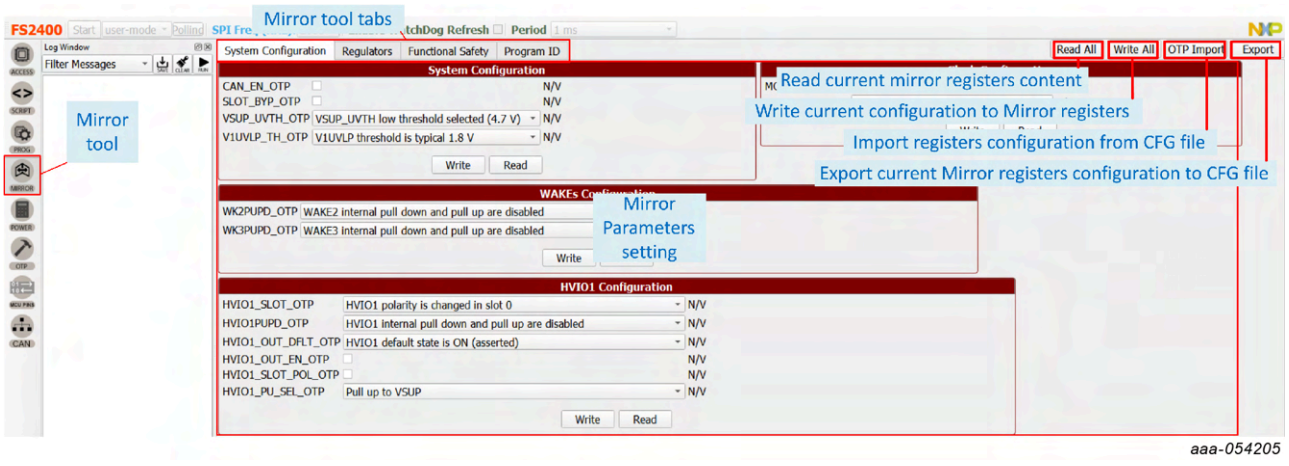


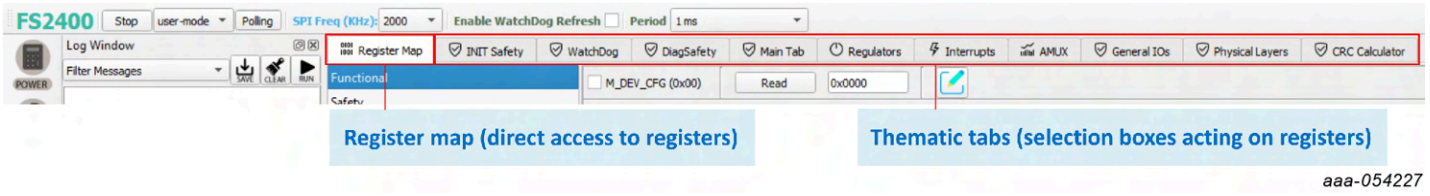
Figure 46. MIRROR tool

7.5.6 ACCESS

The ACCESS tool is the central tool for an evaluation session. This tool gives access to GUI functions that configure, monitor, and control the FS2400 device during the evaluation session.

The ACCESS tool provides access to all SPI registers, displayed either … :

- ... in a Register Map format with direct access to register bits.
- ... in thematic tabs with a graphical and more readable view.



aaa-054227

The Tab Content contains 11 tabs:

- **Register Map**
- **INIT Safety**
- **WatchDog**
- **DiagSafety**
- **Main Tab**
- **Regulators**
- **Interrupts**
- **AMUX**
- **General IOs**
- **Physical Layers**
- **CRC Calculator**

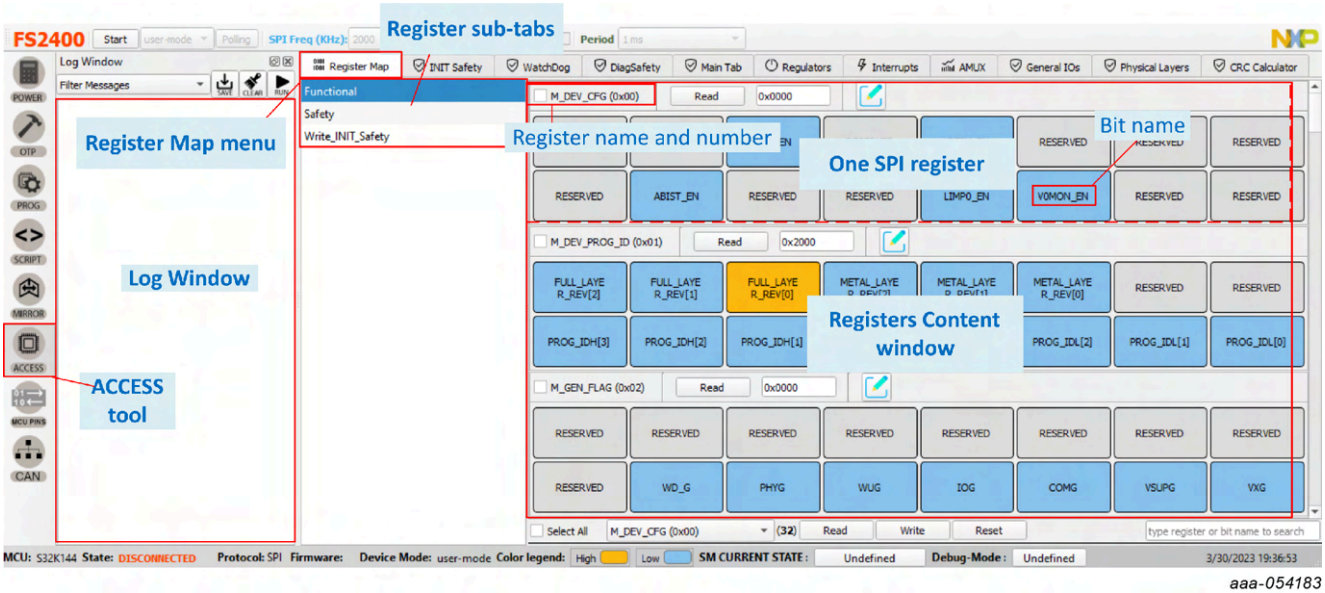
7.5.6.1 Register Map

The Register Map tab allows the user to read or write the FS2400 SPI registers, bit per bit.

Note: *SPI registers are responsible for FS2400 flexible device configuration, in opposition to OTP configuration, which is permanent once fuses are burned. The two levels of configuration work together and should be defined accordingly.*

The Register Map is composed of three sub-tabs:

- **Functional**: Access to SPI functional registers for system configuration.
- **Safety**: Access to SPI registers relative to Safety functions.
- **Write_INIT_Safety**: Access to SPI registers relative to Safety behavior configuration.



aaa-054183

7.5.6.1.1 Modifying one bit by clicking on the bit name

In the Registers Content window, the user can access the FS2400 SPI registers. Two types of registers exist:

- Read Only registers (for example, Status) appear with a **Read** button only.
- Read/Write registers (for example, System configuration, Regulators Control, and so on) appear with both a **Read** button and a **Write** button.

When the bit is writable, the user can click on a bit name to change its value.

aaa-054184

7.5.6.1.2 Accessing one register content using Read and Write buttons

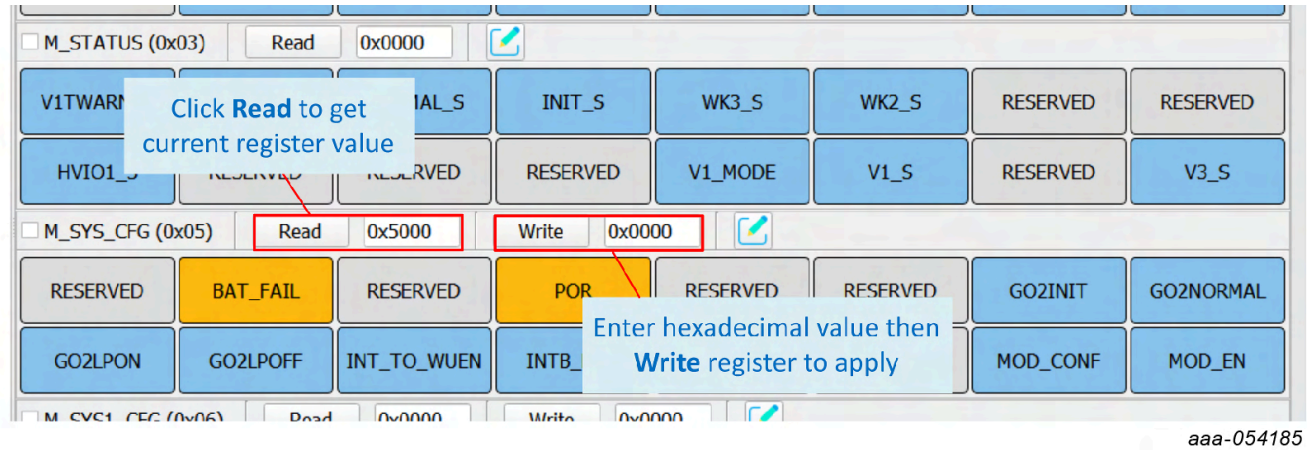
In the Registers Content window, the user can read and/or write the FS2400 SPI registers using the **Read** and **Write** buttons.

To read the content of one register:

- Click **Read** next to the corresponding register name.
- The current register content is displayed as an hexadecimal value next to the **Read** button.

To write the content of one register:

- Enter the desired register value as an hexadecimal value in the box next to the **Write** button.
- Click **Write** next to the corresponding register name.
- As a cross-check, the current register value can be accessed by clicking **Read**.



aaa-054185

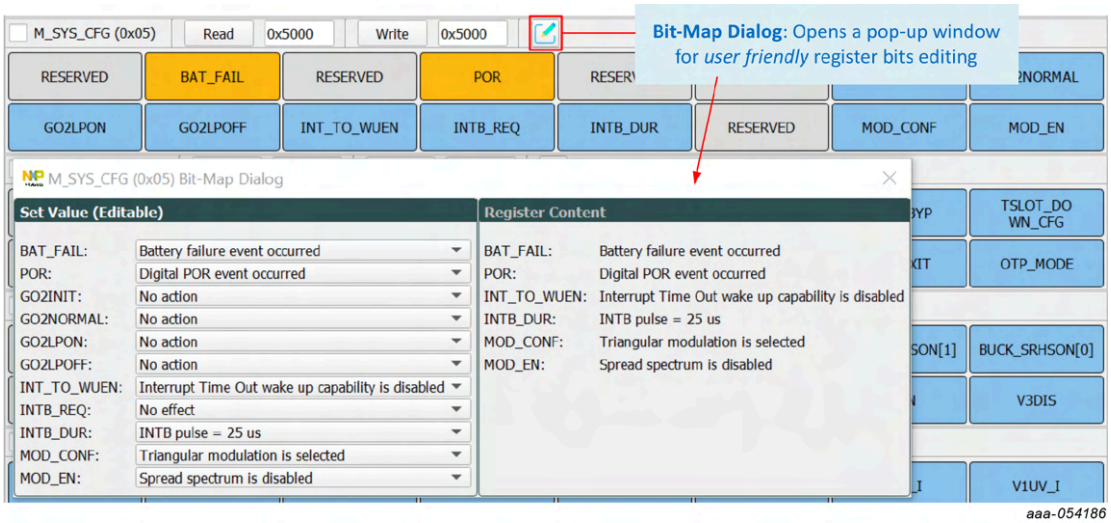
7.5.6.1.3 Accessing one register content using Bit-Map Dialog

In the Registers Content window, the user can read and/or write the FS2400 SPI registers using the Bit-Map Dialog window.

Clicking the **Green Pencil** next to the corresponding register name opens the Bit-Map Dialog window and shows the name and description of all of the register's bitfields.

The Bit-Map Dialog window:

- … allows the user to change bit values from selection boxes on the left side.
- … displays the current register content on the right side.



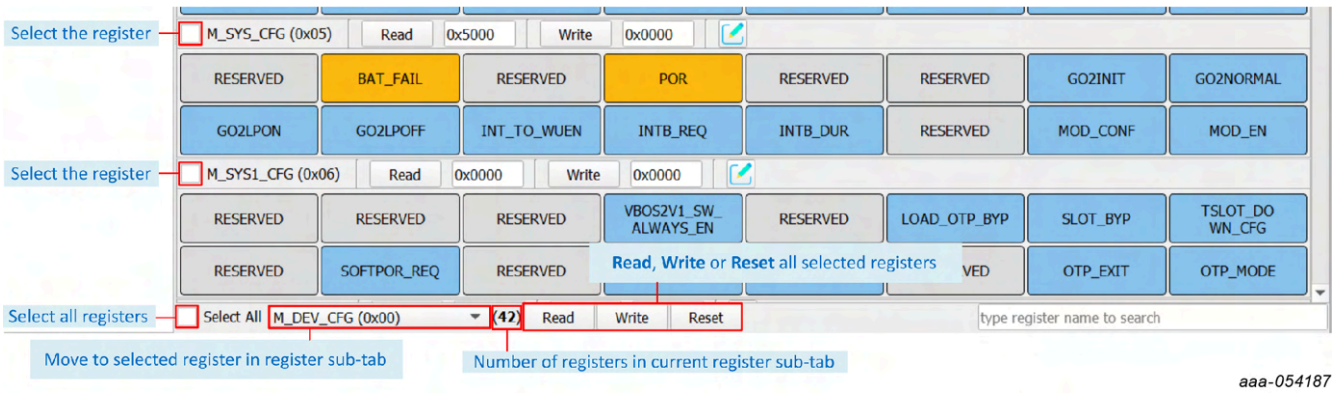
aaa-054186

7.5.6.1.4 Modifying multiple registers using the lower Read/Write/Reset bar

In the Registers Content window, the user can read, write, and/or reset multiple FS2400 SPI registers at a time using the lower Read/Write/Reset bar.

- Checkboxes allow the user to select the registers to be read or written.
- Ticking the Select All checkbox allows the user to simultaneously act on all the registers in the Register sub-tab.
- **Read**, **Write,** and **Reset** buttons allow the user to act on the previously selected registers. **Reset** switches all the bits in the register to 0.

Navigating in the Registers content in the Register sub-tab can be facilitated using the selection box in the bar.

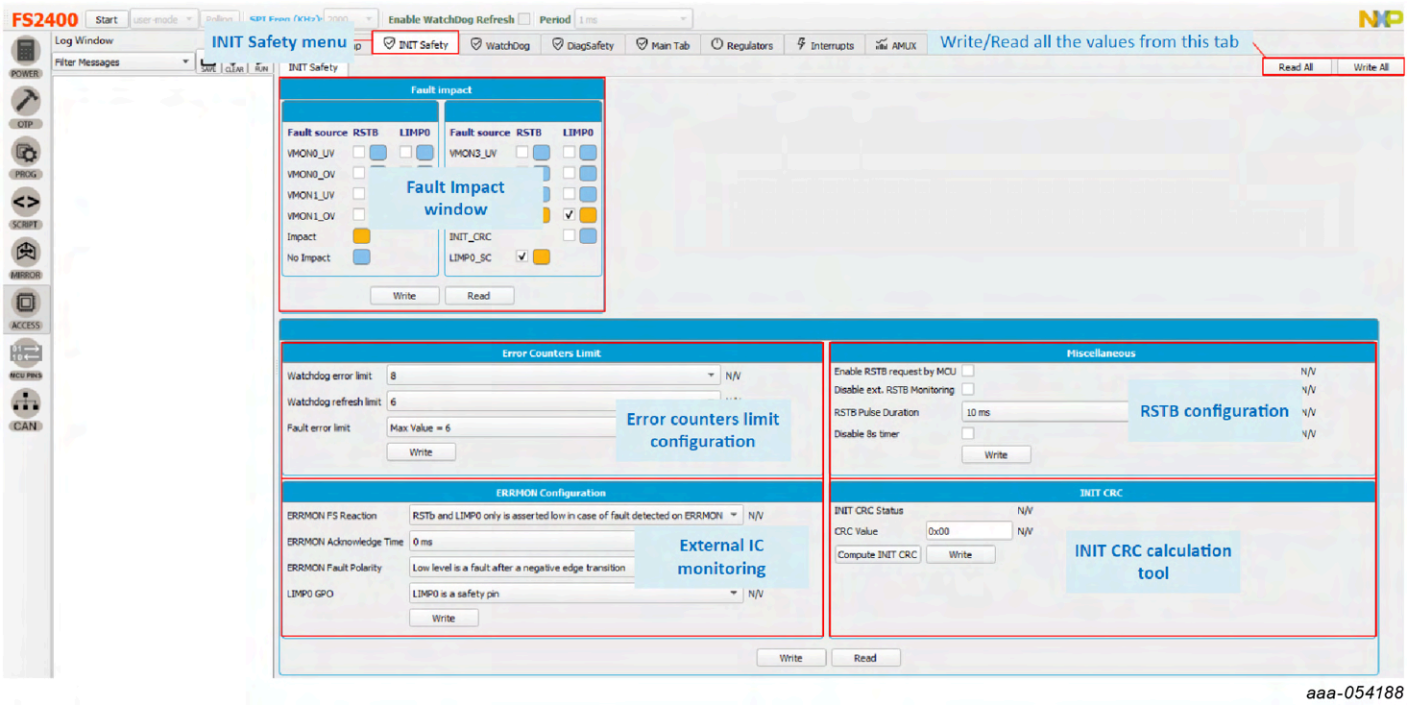


aaa-054187

7.5.6.2 INIT Safety

The INIT Safety tab allows the user to read or write the parameters related to the Safety functions initialization and Safety behavior configuration.

- **Fault Impact:** Used to choose Safety pin assertion response to each fault source.
- **Safety Behavior Config:** Used to initialize and configure Safety-related functions such as error counters limits, RSTB, ERRMON.
- **INIT CRC:** Used to compute the INIT CRC (FS_CRC register) based on the connected device INIT registers (FS_I_XXX) content. When no board is connected, the tab data is used.

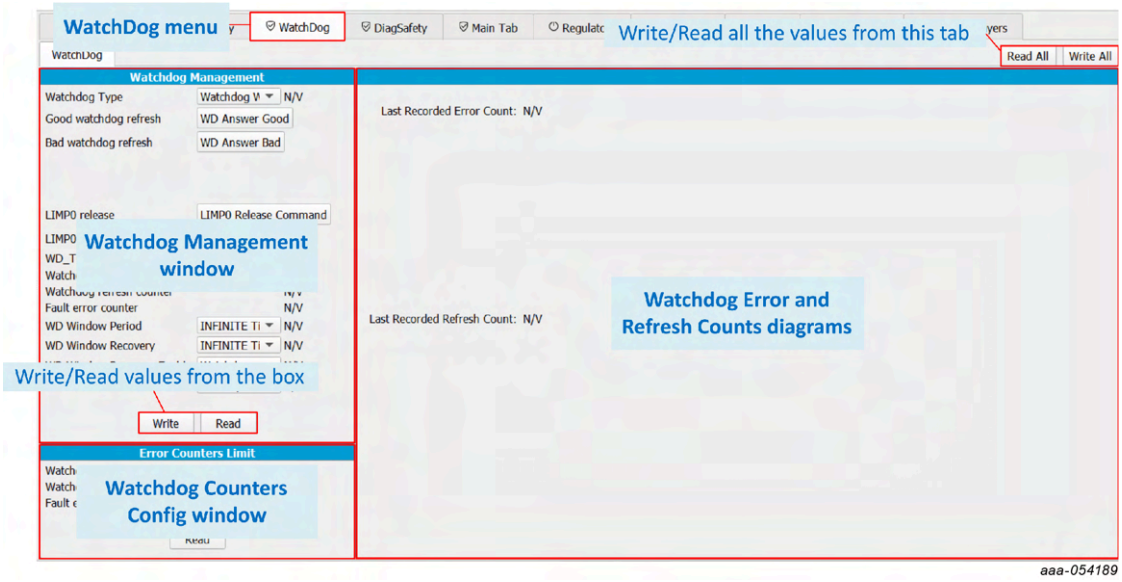


aaa-054188

7.5.6.3 Watchdog

The Watchdog tab allows the user to read or write the parameters related to the Watchdog management and to observe watchdog error and refresh counters evolution.

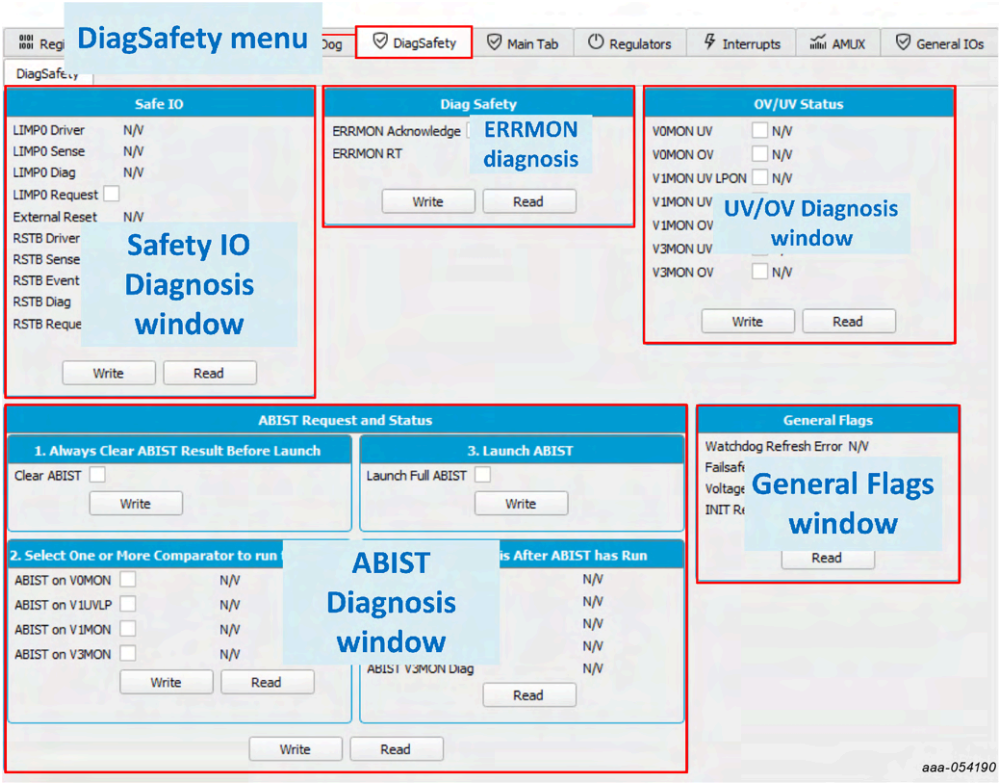
- **Watchdog Management:** Used to manage watchdog operation by sending SPI commands and set watchdog functional parameters, such as window timing.
- **Watchdog Counters Config:** Used to read the previously set watchdog error and refresh counters' limit values and fault error limit action. See Section 7.5.6.2 tab for parameter settings.
- **Watchdog Error and Refresh Counts diagrams:** Used to display the diagrams of watchdog error count and watchdog refresh count relative to the previously set watchdog error and refresh counters limit values.



7.5.6.4 DiagSafety

The DiagSafety tab allows the user to carry out the safety diagnosis by reading or clearing the interrupt and status bits of the Safety pins and the UV/OV monitoring bits. The DiagSafety tab also allows the user to launch ABIST and to observe the ABIST status.

- **Safety IO Diagnosis:** Used to request and observe the diagnosis on the Safety I/O pins.
- **ABIST Diagnosis:** Used to operate ABIST on the device.
- **ERRMON Diagnosis:** Used to carry out the diagnosis on external IC error monitoring.
- **UV/OV Diagnosis:** Used to carry out the diagnosis on UV/OV monitoring by exploiting interrupt flags and real-time status bits.
- **General Flags:** Displays general error flags states.



7.5.6.5 Main Tab

The Main Tab allows the user to manage the Clock Modulation, to monitor occurrence of events, operate the Device States, and to configure the FS2400 LDT.

- **Clock Management:** Used to enable and set 20 MHz clock modulation.
- **General Flags:** Used to monitor occurrence of events on FS2400 functions.
- **Device State:** Used to monitor the device's state and request mode change.
- **LDT Configurations:** Used to enable, configure, and monitor the LDT.
- **LDT Calibration:** Used to emulate the LDT calibration procedure using the device's embedded S32K144. The MCU software timing measurements are not accurate, the GUI tool is only intended to help understand the procedure.
- **Miscellaneous management:** Used to monitor FS2400 external events and FS2400 VBOS, or to monitor time slots' attribution and interrupt settings.

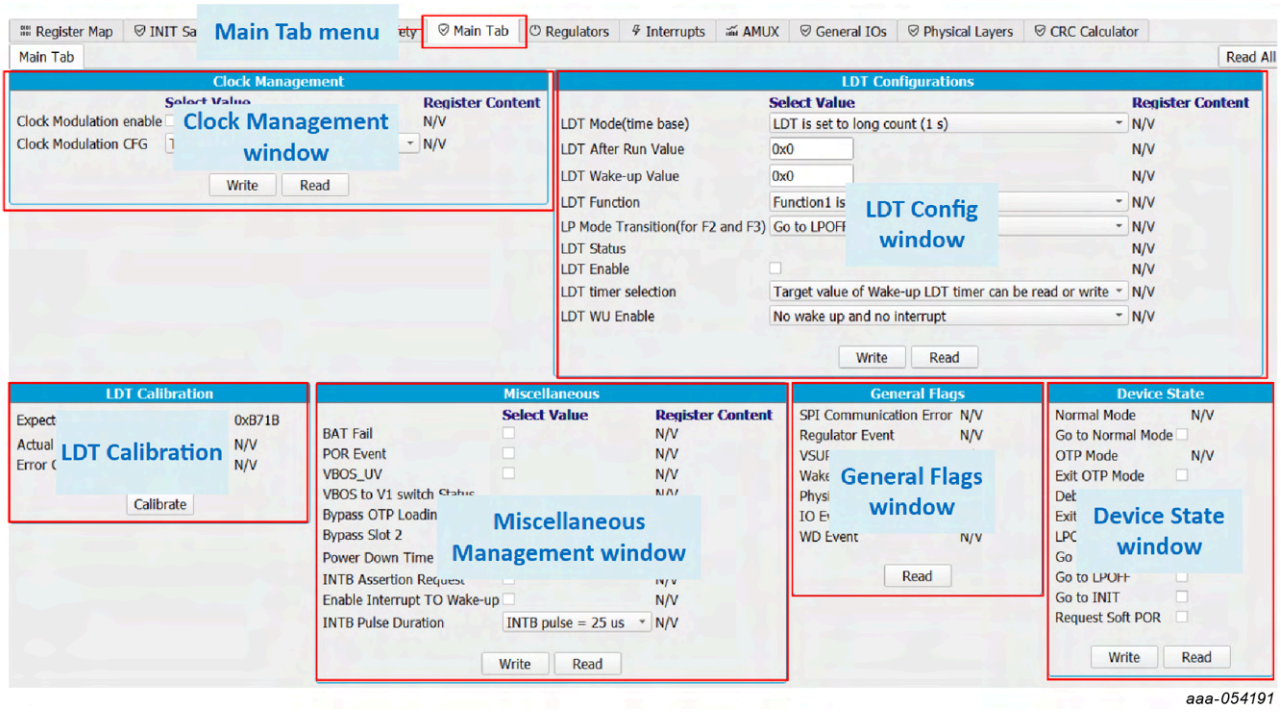


Figure 47. Main Tab

7.5.6.6 Regulators

The Regulators tab allows the user to control and read status related to HVBUCK (V1) and HVLDO3 (V3).

The Regulators tab gives access to the real-time status of each regulator for monitoring, and allows the user to control the regulators' enablement in specific modes.

- **V1 HVBUCK:** Used to configure V1 slew rate and DVS parameters
- **Regulators Control:** Used to control the regulators' enablement and V1 DVS transition

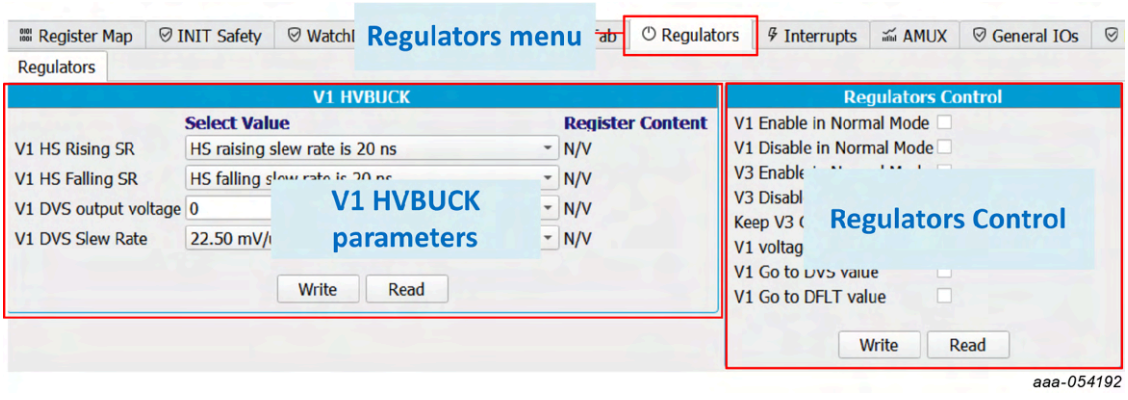


Figure 48. Regulators menu

7.5.6.7 Interrupts

The Interrupts tab is arranged as an Interrupt board with two thematic sub-menus displaying one Interrupt box per function:

- **Main Interrupt Configuration:** Used to monitor events related to FS2400 functional features, such as regulators' temperature and current, SPI communication, physical layers, timers, high-side drivers, I/Os, and wake-up events.
- **FailSafe Interrupt Configuration:** Used to monitor safety-related events, such as undervoltage/overvoltage or open load on regulators, and events on safety pins or functions.

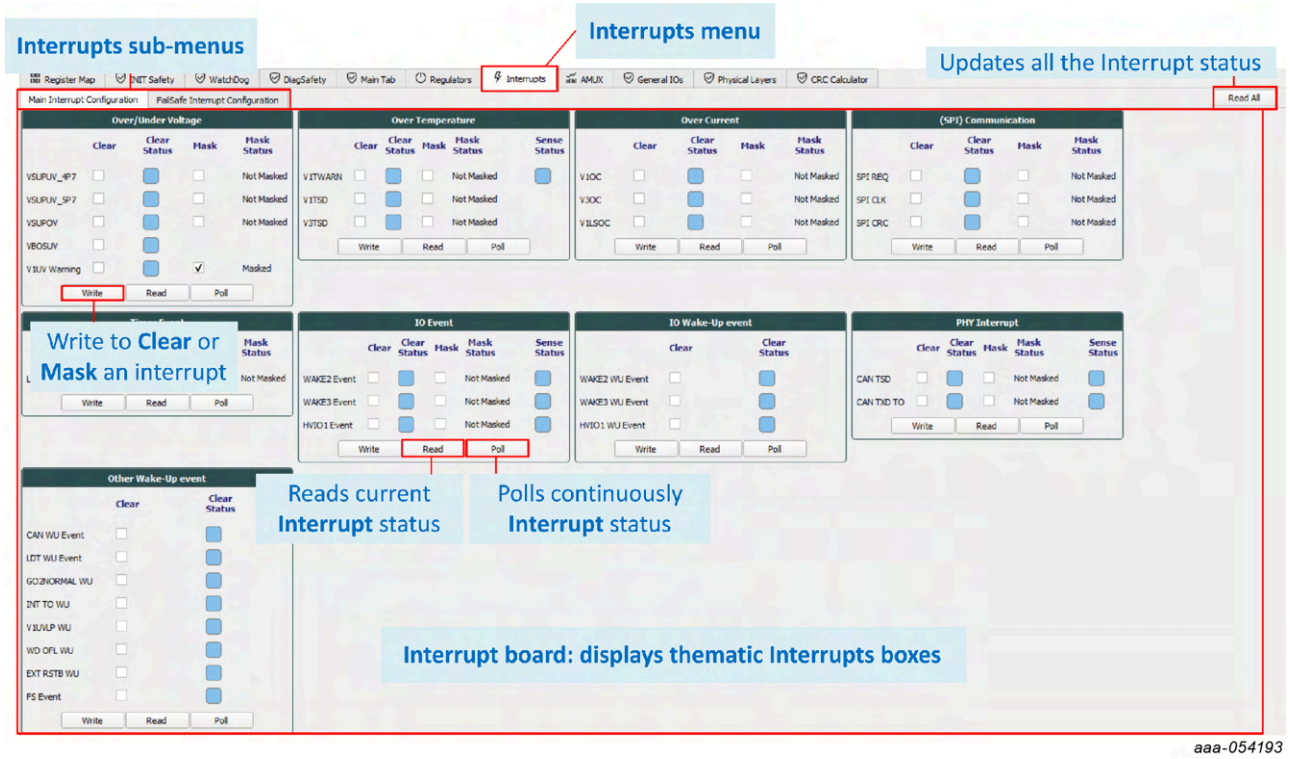


Figure 49. Interrupts menu

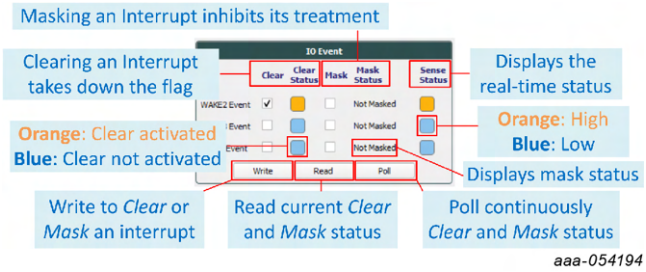


Figure 50. Masking an interrupt

In each Interrupt box, the interruption can be:

- **Read**: Click **Read** on each box to read the current status or Read all to update the whole Interrupt board.
- **Polled**: Click **Poll** on each box to read the current status periodically.
- **Cleared**: Select each check box from the Clear column, then click **Write** to apply.
- **Masked**: Select each check box from the Mask column, then click **Write** to apply.

7.5.6.8 AMUX

The AMUX tab allows the user to measure voltage levels and temperature values of multiple key locations from onboard sensors. The AMUX feature uses one ADC to successively measure multiple points. The AMUX feature must be enabled beforehand by clicking **Enable AMUX**.

- **AMUX Measurements:** Displays the measurements of voltage levels and temperature values from sensors placed at key locations. Click **Read** to obtain or refresh the measurements.
- **ADC Measurements:** Displays the measurements of valuable voltage levels sensed by an ADC outside of the AMUX feature. Click **Read** to obtain or refresh the measurements.
- **Voltage Measurement graph:** Displays the selected voltage measurement values as a function of time in a graph.
- **Temperature Measurement graph:** Displays the selected temperature measurement values as a function of time in a graph.

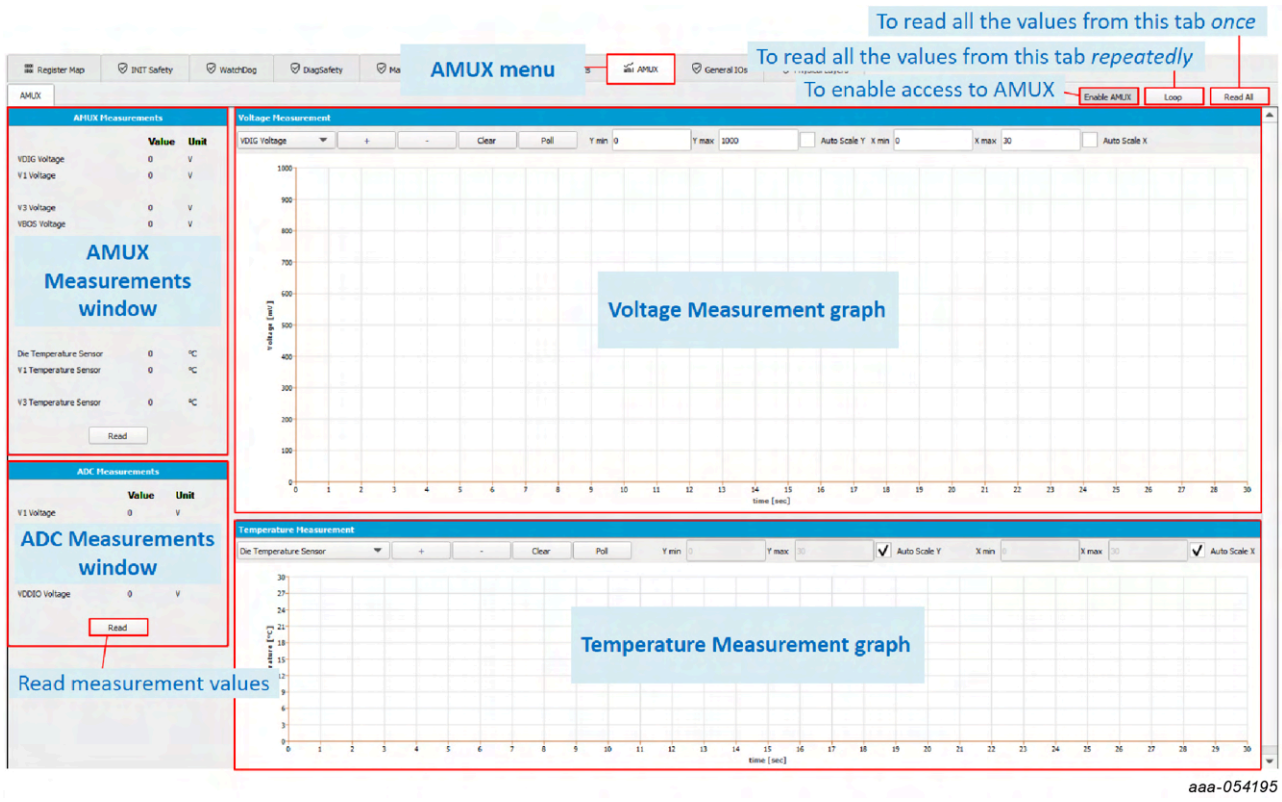


Figure 51. AMUX menu

Measurement graphs can be edited using the editing bar, as shown in Figure 52:

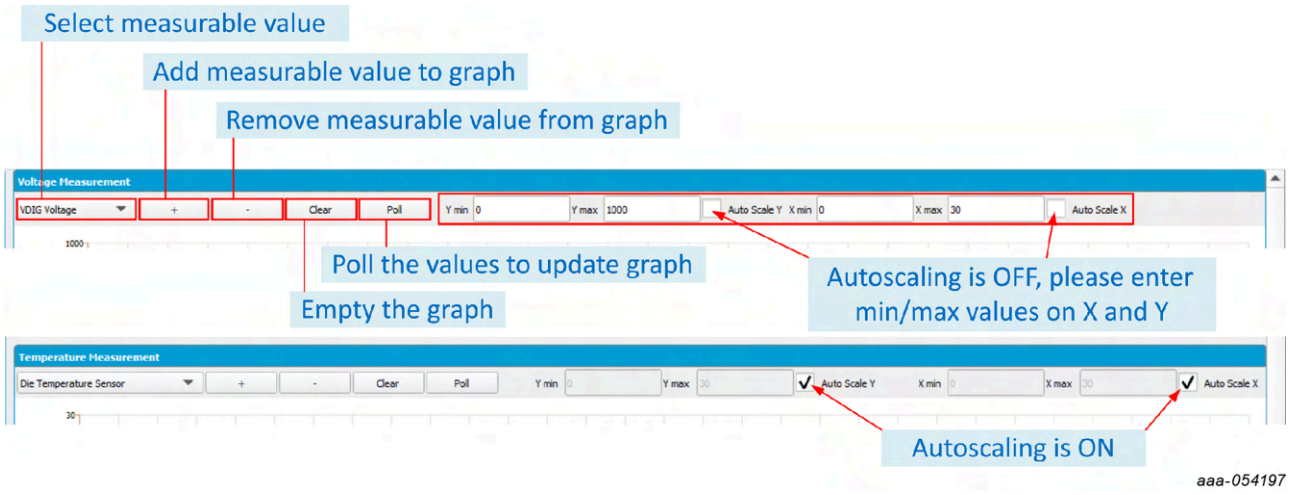


Figure 52. Measurement graph

7.5.6.9 General IOs

The General IOs tab allows the user to control I/O levels, to configure Wake-Up/Interrupt features, and to operate the Cyclic Sense feature depending on the I/Os desired configuration. A pre-configuration of the I/Os is done by OTP.

- **IO Control:** Used to control I/O output levels when configured as outputs
- **IO Wake-up Config:** Used to set I/O wake-up parameters when the Wake-Up feature is enabled on I/O
- **IO Wake-up/Interrupt Enable:** Used to enable Wake-Up and/or Interrupt feature on I/O
- **HW ID Config:** Used to enable and disable pullup and pulldown current sources on FS2400 WAKE2/HID0 and WAKE3/HID1 pins

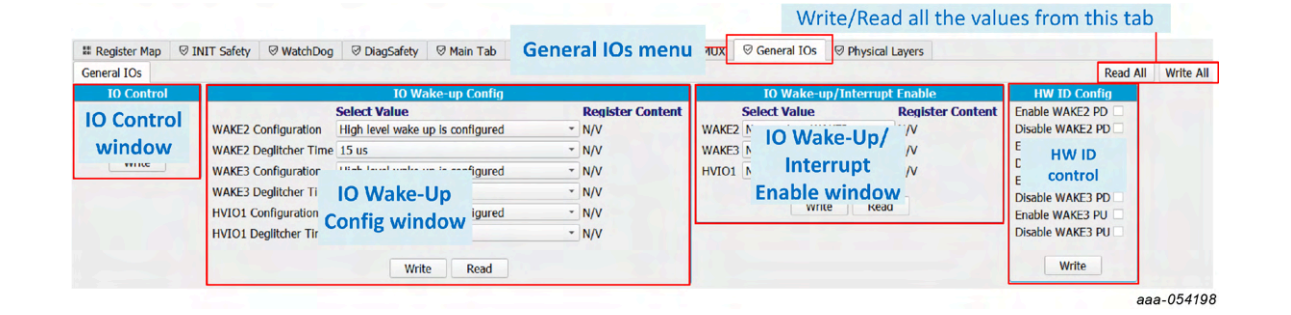


Figure 53. General IOs tab

7.5.6.10 Physical Layers

The Physical Layers tab allows the user to configure the CAN and LIN transceiver and monitor status.

- **CAN Config/Status:** Used to configure the CAN transceiver and monitor status. The CAN setting on the MCU side and the CAN frames sending is done using the CAN tool.

Figure 54. Physical Layers tab

7.5.6.11 CRC Calculator

The CRC Calculator tab allows the user to compute the FS2400 SPI CRC using the specified polynomial. It can be useful while debugging SPI communication.

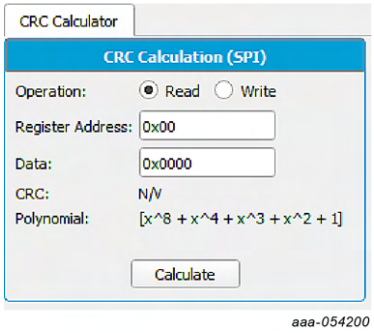


Figure 55. CRC Calculator

7.5.7 MCU PINS

The MCU PINS tool provides a means of reading and setting the S32K144 input and for signals RSTB and LIMP0.

The MCU PINS panel consists of three sections:

- **Log window:** Maintains a running log of events initiated during the current session. A drop-down menu in the upper left allows the log to be filtered by register read, register write, pin read, and pin write. Buttons in the upper right allow the Log window contents to be saved, cleared, or run.
- **MCU Input Pins Reading window:** Allows the listed MCU pins to be read or polled during a selected time duration.

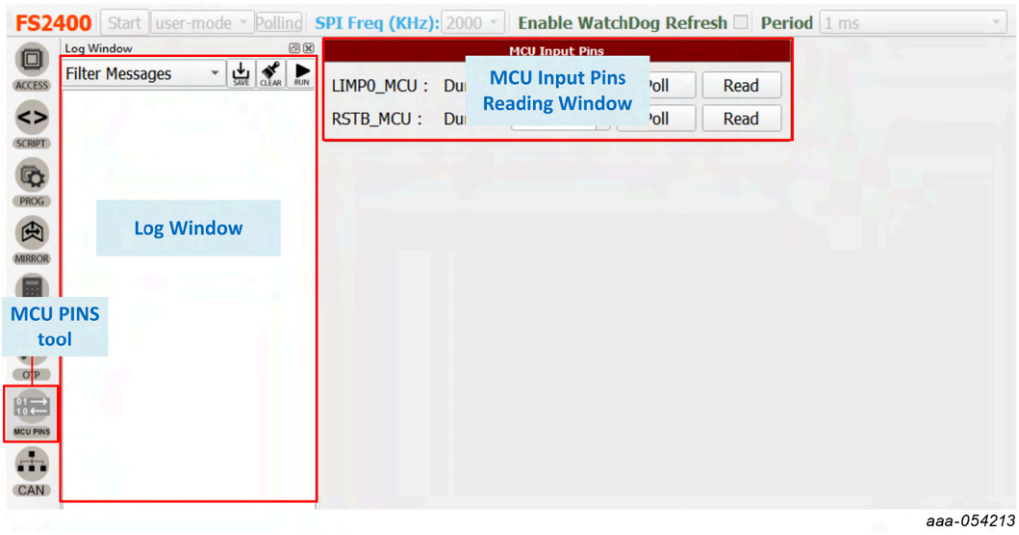


Figure 56. MCU PINS tools

7.5.8 CAN

The CAN tool provides a means to use the CAN transceiver on FS2400. The CAN tool allows the user to configure the CAN bus on the MCU side and send edited frames on the bus sporadically or periodically with a selectable repeat rate. CAN logs are displayed.

The CAN tool panel consists of four sections:

- **MCU CAN Configuration window:** Used to configure the CAN on the MCU S32K144 side by choosing the baud rate, CAN ID, and payload size
- **CAN Send window:** Used to send CAN frames sporadically or periodically with a selectable repeat rate from 200 ms to 100 s
- **CAN Frames Editing window:** Used to edit the frames to be sent on the CAN bus (up to 64 bytes) with hexadecimal values. CAN frames can be imported and saved as CSV files
- **CAN Log window:** Displays exchanged frames
- **CAN Tool state machine:** Helps with understanding how to use the CAN tool

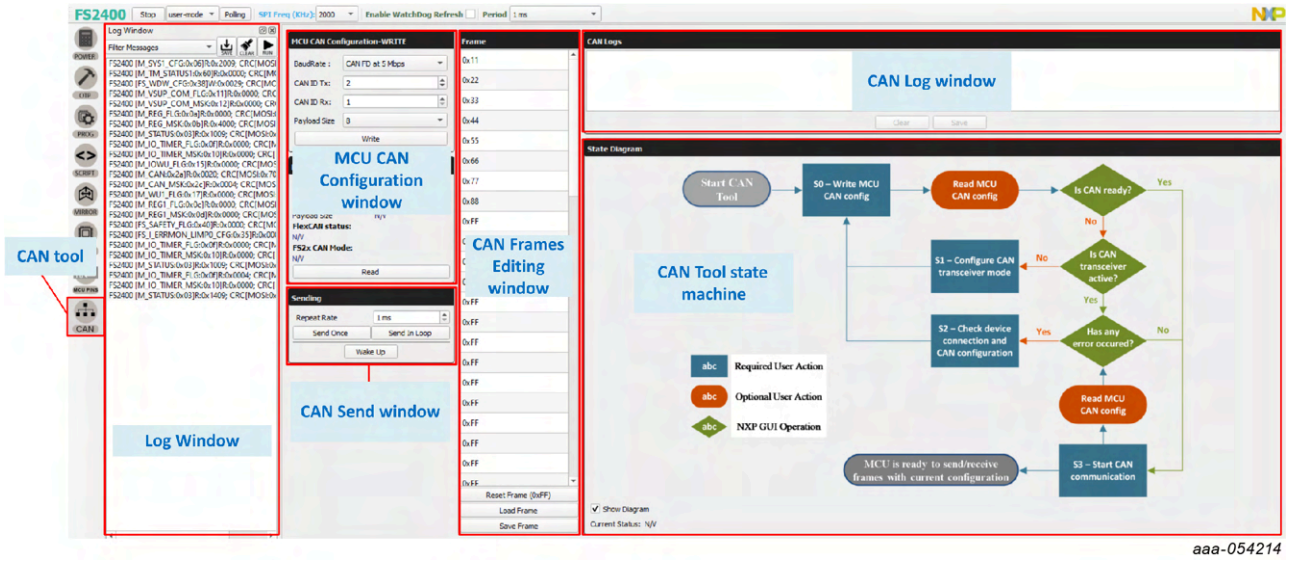


Figure 57. CAN tool

8 References

1. **FS2400** — product information on FS2400, Safety system basis chip, fit for ASIL D

9 Revision history

Table 23. Revision history

Document ID	Release date	Description
UM12015 v.1.0	01 February 2024	Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Suitability for use in automotive applications (functional safety) — This NXP product has been qualified for use in automotive applications. It has been developed in accordance with ISO 26262, and has been ASIL classified accordingly. If this product is used by customer in the development of, or for incorporation into, products or services (a) used in safety critical applications or (b) in which failure could lead to death, personal injury, or severe physical or environmental damage (such products and services hereinafter referred to as "Critical Applications"), then customer makes the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, safety, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. As such, customer assumes all risk related to use of any products in Critical Applications and NXP and its suppliers shall not be liable for any such use by customer. Accordingly, customer will indemnify and hold NXP harmless from any claims, liabilities, damages and associated costs and expenses (including attorneys' fees) that NXP may incur related to customer's incorporation of any product in a Critical Application.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Tab. 1. Evaluation board components description 10
Tab. 2. Evaluation board LED signaling description 12
Tab. 3. Main VBAT Phoenix connector (J6) 13
Tab. 4. Alternative VBAT Phoenix connector (J35) 13
Tab. 5. V3 (HVLDO) connector (J7) 13
Tab. 6. V1 (HVBUCK) connector (J12) 14
Tab. 7. V3 CAN bus connector (J16) 14
Tab. 8. Power-up debug connector (J46) 14
Tab. 9. Communication debug connector (J44) 14
Tab. 10. FS2400 signals debug connector (J47) 15
Tab. 11. VDDIO selection connector (J26) 15
Tab. 12. LIMP0 pullup selection connector (J45) 15
Tab. 13. USB connector (J48) 16
Tab. 14. JTAG connector (J38) 16
Tab. 15. Evaluation board test points description 16
Tab. 16. Evaluation board jumpers description 18
Tab. 17. SW1 description 19
Tab. 18. S2 description 19
Tab. 19. SW4 description 19
Tab. 20. SW9 description 20
Tab. 21. SW12 description 20
Tab. 22. SW19 description 20
Tab. 23. Revision history 75

Figures

Fig. 1. KITFS24SKTFDMEVM block diagram 1
Fig. 2. Communication/I/Os connection to S32K144 5
Fig. 3. Power up debug signals 6
Fig. 4. J44-46 Communication/IOs connection 6
Fig. 5. VDDIO selection 6
Fig. 6. VDDIO power supply selection 7
Fig. 7. P_VAR supply configuration (LDO P_VAR (P3V3 or P1V8)) 7
Fig. 8. Analog inputs 8
Fig. 9. OTP block diagram 9
Fig. 10. OTP mode entry 9
Fig. 11. OTP hardware implementation 9
Fig. 12. Evaluation board featured components location 10
Fig. 13. Evaluation board LED signaling location 12
Fig. 14. Evaluation board connectors 13
Fig. 15. Evaluation board test points 16
Fig. 16. Evaluation board jumpers location (with default position) 17
Fig. 17. Switches location 19
Fig. 18. Example debugger 21
Fig. 19. Proper connection 21
Fig. 20. Power supply 31
Fig. 21. Operation modes 33
Fig. 22. ACCESS tool 37
Fig. 23. Framework window 43
Fig. 24. Framework setings 44
Fig. 25. File menu item 44
Fig. 26. Help menu item 46
Fig. 27. Watchdog enablement/disablement/ configuration 48
Fig. 28. Watchdog management boxes 49
Fig. 29. Watchdog enablement 49
Fig. 30. Enable WatchDog Refresh unchecked 49
Fig. 31. USB and Device status bar 49
Fig. 32. POWER tool 51
Fig. 33. OTP Tool tabs 52
Fig. 34. Power sequence configuration 53
Fig. 35. Regulators tab 53
Fig. 36. Functional Safety tab 54
Fig. 37. Program ID tab 54
Fig. 38. Customer Details box 55
Fig. 39. Program Details box 55
Fig. 40. PROG tool 56
Fig. 41. Fuse Box Status 56
Fig. 42. SCRIPT tool 57
Fig. 43. Parameters panel 59
Fig. 44. Script Commands window 59
Fig. 45. Script Results window 60
Fig. 46. MIRROR tool 60
Fig. 47. Main Tab 67
Fig. 48. Regulators menu 67
Fig. 49. Interrupts menu 68
Fig. 50. Masking an interrupt 68
Fig. 51. AMUX menu 69
Fig. 52. Measurement graph 70
Fig. 53. General IOs tab 70
Fig. 54. Physical Layers tab 71
Fig. 55. CRC Calculator 71
Fig. 56. MCU PINS tools 72
Fig. 57. CAN tool 73

Contents

1 Introduction 1
2 Finding kit resources and information on the NXP web site 3
2.1 Collaborate in the NXP community 3
3 Getting ready 4
3.1 Kit contents 4
3.2 Additional hardware 4
3.3 Windows PC workstation 4
3.4 Software 4
4 Getting to know the hardware 5
4.1 Kit overview 5
4.1.1 KITFS24SKTFDMEVM features 5
4.1.2 Communication and I/Os between FS2400 and S32K144 MCU 5
4.1.3 VDDIO selection 6
4.1.4 S32K144 ADC 7
4.2 Device OTP user configuration 8
4.2.1 OTP and mirrors registers 8
4.2.2 OTP hardware implementation 9
4.3 Kit featured components 10
4.3.1 FS2400: Fail-safe system basis chip with SMPS and LDO 11
4.3.1.1 General description 11
4.3.1.2 Features 11
4.3.2 LED signaling 12
4.3.3 Connectors 13
4.3.3.1 VBAT connectors 13
4.3.3.2 Output power supply connectors 13
4.3.3.3 CAN bus connector 14
4.3.3.4 Debug connectors 14
4.3.3.5 VDDIO selection connector 15
4.3.3.6 LIMP0 pullup selection connector 15
4.3.3.7 S32K144 communication connectors 16
4.3.4 Test points 16
4.3.5 Jumpers 17
4.3.6 Switches 19
4.4 Schematic, board layout and bill of materials 20
5 Installing and configuring software and tools 21
5.1 Flashing or updating the GUI firmware 21
5.1.1 Hardware setup 21
5.1.2 Software setup 22
5.2 Installing NXP GUI software package 28
5.3 Launching the FS24 NXP GUI 29
6 Configuring the hardware and software 31
6.1 Setting up the KITFS24SKTFDMEVM 31
6.2 Connecting the KITFS24SKTFDMEVM to the GUI 31
6.3 Operation modes 32
6.3.1 Debug mode 33
6.3.1.1 Debug mode definition 33
6.3.1.2 Debug mode activation 34
6.3.1.3 Debug mode deactivation 34
6.3.2 Test mode 34
6.3.2.1 Test mode definition 34
6.3.2.2 Test mode activation: hardware and software 34
6.3.2.3 Test mode deactivation 35
6.3.2.4 Test mode operation 35
6.3.3 User mode 35
6.3.3.1 User mode definition 35
6.3.3.2 User mode activation 35
6.4 Generate a TBB script 36
6.5 First-start procedure: configure Watchdog as Infinite Time Out 36
6.6 Operate OTP emulation by loading configuration in the Mirror registers 38
6.6.1 Modify the Mirror registers with a TBB script 38
6.6.2 Modify the Mirror registers with the MIRROR tool 38
6.7 Programming an OTP operation 39
6.8 Save a routine from Log window then Run it as a Script 40
6.9 TBB script example 41
7 Tool interface (GUI) description 43
7.1 Framework window 43
7.2 Framework settings 44
7.2.1 File menu item 44
7.2.2 View menu item 44
7.2.3 Import/Export menu item 45
7.2.4 Help menu item 46
7.3 Connection Toolbar 46
7.3.1 Device connection 46
7.3.2 SPI communication configuration 47
7.3.3 Watchdog management 47
7.3.3.1 Watchdog enablement and configuration on FS2400 side 47
7.3.3.2 Watchdog configuration on MCU side 48
7.4 USB and Device status bar 49
7.5 Tools access bar 50
7.5.1 POWER 50
7.5.2 OTP 51
7.5.2.1 OTP Parameters Setting window 52
7.5.2.2 OTP Details window 54
7.5.3 PROG 55
7.5.3.1 Device Programming Configuration window 56
7.5.3.2 OTP mode window 56
7.5.3.3 Fuse Box Status window 56
7.5.4 SCRIPT 57
7.5.4.1 Log window 57
7.5.4.2 Script Commands panel 57
7.5.4.3 Script Commands window 59
7.5.4.4 Script Results window 59
7.5.5 MIRROR 60
7.5.6 ACCESS 60
7.5.6.1 Register Map 61
7.5.6.2 INIT Safety 64
7.5.6.3 Watchdog 64
7.5.6.4 DiagSafety 65
7.5.6.5 Main Tab 66
7.5.6.6 Regulators 67

7.5.6.7 Interrupts 67
7.5.6.8 AMUX 69
7.5.6.9 General IOs 70
7.5.6.10 Physical Layers 70
7.5.6.11 CRC Calculator 71
7.5.7 MCU PINS 71
7.5.8 CAN 72
8 References 74
9 Revision history 75
Legal information 76

Date of release: 1 February 2024
Document identifier: UM12015