UM11879

KITFS23LDOEVM evaluation board

Rev. 2 — 25 September 2023

User manual

Document information

Information	Content
Keywords	FS2300, S32K144, FS2300 NXP GUI, SBC, Body and Comfort, ISO 26262
Abstract	The KITFS23LDOEVM evaluation board user guide is intended for engineers involved in the evaluation, design, implementation, and validation of FS2300 fail-safe system basis chips.

Revision history

Rev	Date	Description

v.2	20230925	• Replaced all occurences of "User mode" with "Normal mode"
		• Section 1; Section 6.1; Section 6.3.2; Section 6.5.2.1.5; Section 6.5.3.2; Section 6.5.4.1; Section 6.5.4.2; Section 6.5.6.2; Section 6.5.6.5; Section 6.5.1; Section 7.3.2.3; Section 7.3.3.2.1: Updated text
		• Updated Figure 2; Figure 3; Figure 4; Figure 10; Figure 11; Figure 21; Figure 30; Figure 31; Figure 32; Figure 33; Figure 34; Figure 35; Figure 36; Figure 38; Figure 39; Figure 40; Figure 41; Figure 42; Figure 43; Figure 44; Figure 45; Figure 47; Figure 48; Figure 52; Figure 59; Figure 60; Figure 66; Figure 78; Figure 79; Figure 80; Figure 81; Figure 83; Figure 84; Figure 85; Figure 86; Figure 88; Figure 90
		• Section 2.1; Section 3.4; Section 4.4.14: Updated link to https://www.nxp.com/KITFS23LDOEVM
		• Section 4.4: Updated Table 1; Table 2: Removed J33 row
		• Section 4.4.5
		  – Removed "J33" from section title, updated text
		  – Updated Table 7
		    – Removed "J33" from table title
		    – Removed "J33-1-2" and J33 (OFF) rows
		    – Updated Description of "J35-1"
		• Removed section titled "FS0B pullup configuration (R13/R12/C26)"
		• Updated Table 12
		• Section 4.4.11.2: Updated last sentence of first paragraph; replaced former figures 8, 9 , 10 with tables containing same data.
		• Section 4.4.12: Updated Signals and Power supply bullet items
		• Section 6.3: Updated text; Added Protocol selection bullet item
		• Section 6.5: Reordered sub-sections, updated Power description
		• Section 6.5.3.3: Updated text, removed figure titled "Fuse box status for an empty part"
		• Section 6.5.7: Updated "MCU Input Pins Reading window" and "MCU Output Pins Setting window"
		• Section 6.5.8, Section 6.5.9: Added note
		• Section 7.3: Updated text, added Table 17
		• Section 7.5: Updated set up steps
		• Section 7.7.2: Changed "Import from CFG" to "Load CFG" in step 2, changed "Export from CFG" to "Save CFG" in step 4
		• Section 7.8: Updated text; updated "Programming procedure"; removed figures titled "Programming an SPI part", "Programming a part using PROG tool", and "Loading a TBB file to burn a part"; added Figure 92

v.1	20230223	Initial version

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.

1 Introduction

The KITFS23LDOEVM board user manual is intended for engineers involved in the evaluation, design, implementation, and validation of the FS2300 Body and Comfort system basis chip (SBC).

The KITFS23LDOEVM board is a soldered board supporting the FS2300 fail-safe SBC with three low dropout (LDO) regulators.

The KITFS23LDOEVM enables development on the FS2300 Body and Comfort SBC family of devices. This document covers connecting the hardware, installing the NXP GUI software, configuring the environment, and using the kit. It includes guidance on how to use the registers, try OTP configurations, and burn the part.

The board as delivered includes a soldered FS2320 device with empty OTP content in order to leave the opportunity to the user to burn the OTP configuration.

2 Finding kit resources and information on the NXP website

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

The information page for KITFS23LDOEVM board is at https://www.nxp.com/KITFS23LDOEVM. The information page provides overview information, documentation, software and tools, parametrics, ordering information.

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 KITFS23LDOEVM requires the kit contents, additional hardware, and a Windows PC workstation with installed software.

3.1 Kit contents

The KITFS23LDOEVM kit contains the following items:

• Assembled and tested evaluation board with preprogrammed S32K144 microcontroller in an anti-static bag
• 3.0 ft USB-STD A to USB-B-micro cable
• Six connectors, terminal block plug, two positions, straight, 3.81 mm
• Three connectors, terminal block plug, three positions, straight, 3.81 mm
• Jumpers mounted on board

3.2 Additional hardware

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

• One power supply with a range up to 40 V.

3.3 Minimum system requirements

This evaluation board requires a Windows PC workstation.

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

3.4 Software

Installing software is necessary to work with the KITFS23LDOEVM evaluation board.

All listed software is available on the evaluation board's information page at https://www.nxp.com/KITFS23LDOEVM.

• NXP GUI for automotive SBC family installation package - latest version

4 Getting to know the hardware

The KITFS23LDOEVM kit provides an integrated platform for evaluating designs based on NXP's FS2300 SBC. All FS2300 features can be accessed and monitored in a test environment using NXP GUI software to access the registers in Read and Write mode. All regulators are accessible through connectors. Nonuser signals are mapped on test points. Digital signals (I^2C, RSTB, etc.) are accessible through connectors.

4.1 Kit overview

The kit's hardware consists of the KITFS23LDOEVM evaluation board embedded with an S32K144 microcontroller, and the USB cable required to connect the board to the PC.

The KITFS23LDOEVM evaluation board features a soldered FS2300 part and allows the user to fuse the device's one time programmable (OTP) feature without extra tools. 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. An emulation mode allows the user to test as many configurations as needed before programming the part.

The S32K144 is soldered on the bottom side of the KITFS23LDOEVM board. The role of the S32K144 is to manage SPI or I^2C communication (depending on the user's choice) between the KITFS23LDOEVM board and the GUI installed on the PC. The S32K144 draws power either from the USB cable connected to the PC or from the battery supply (when not connected to the GUI).
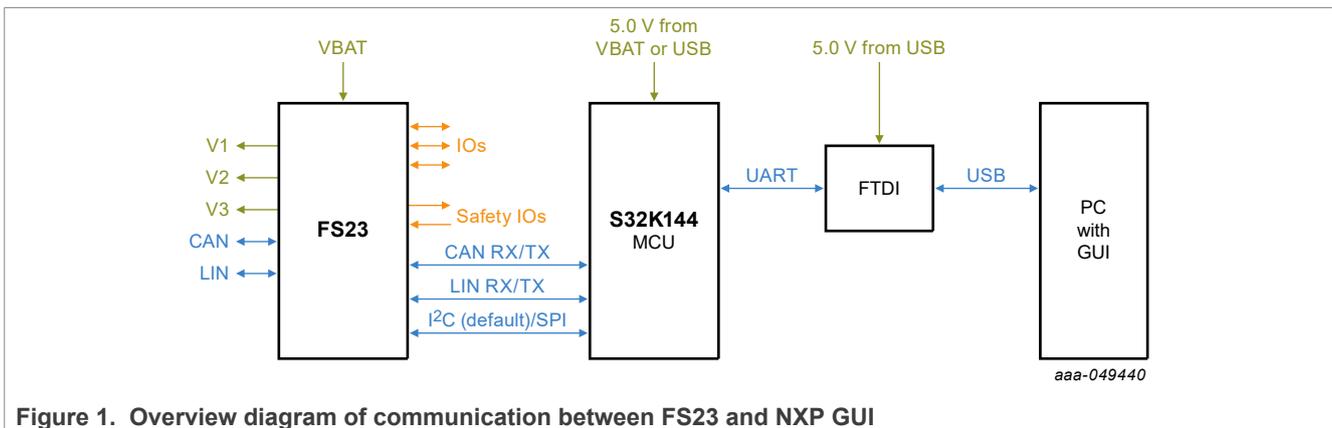

Figure 1. Overview diagram of communication between FS23 and NXP GUI

4.2 Board features

- Phoenix (3.81 mm) male connector or Jack connector for power supply input
- Phoenix (3.81 mm) male connectors for HVLDOs
- Phoenix (3.81 mm) male connectors for CAN and LIN communication
- Phoenix (3.81 mm) male connectors for high-side drivers
- Selectable Debug mode, Test mode, and Normal mode with switches and jumpers to power up the device in different modes
- INTB, RSTB, FS0B, LIMP0, FCCU pins
- Embedded USB to I^2C/SPI protocol for easy connection to software GUI through S32K144 MCU
- Red and green LEDs to indicate signal or regulator status
- Blue LED to indicate that OTP 8 V burning voltage is set
- Header connectors for I/O configuration
- Advanced system monitoring via S32K144 MCU

4.3 KITFS23LDOEVM featured components

Figure 2 shows the placement of connectors, configuration headers, and LEDs signaling. For information on default jumper and switch configuration for FS2300, see Section 7.1.
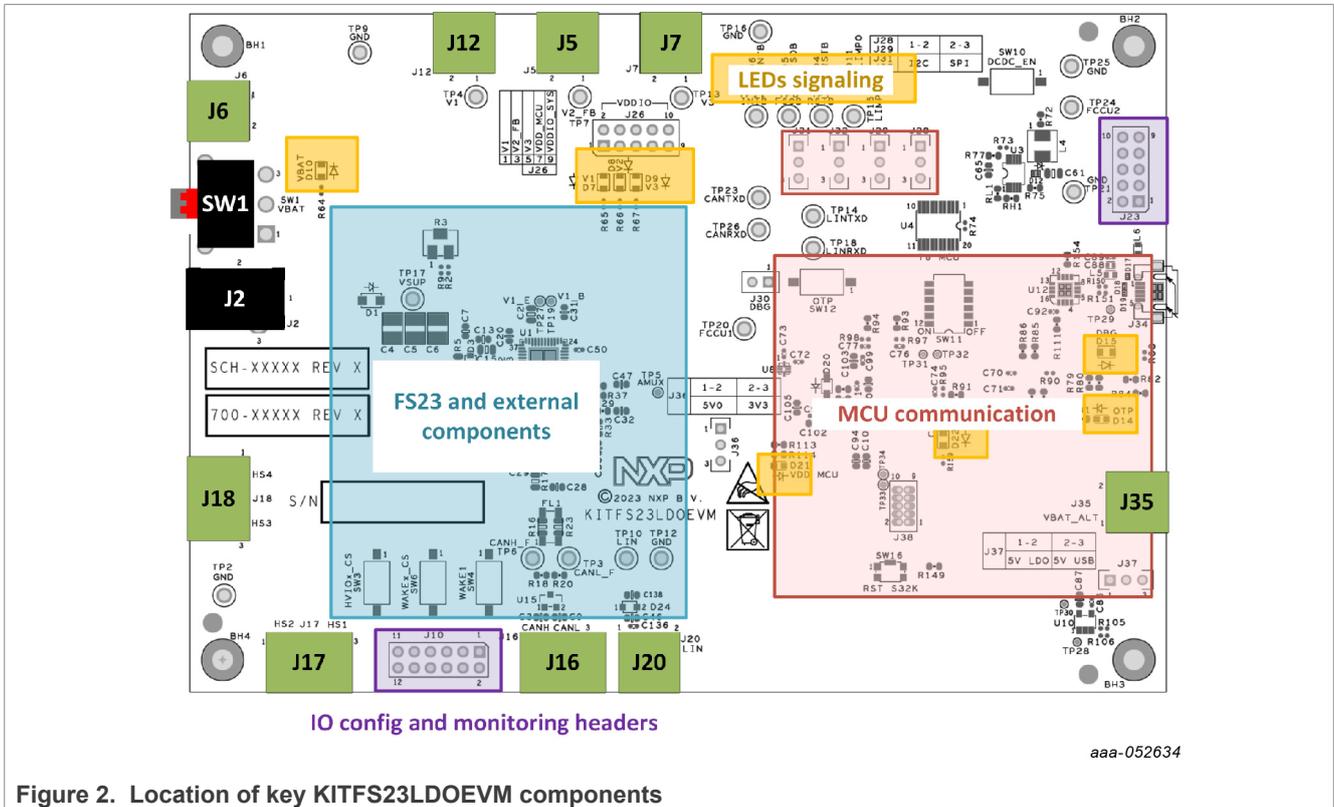

Figure 2. Location of key KITFS23LDOEVM components

4.4 KITFS23LDOEVM evaluation board

This section describes the KITFS23LDOEVM evaluation board by explaining the features associated with each jumper and switch, giving information on hardware configuration to enable these features, and providing advice on LED signaling and test points available on the board.

Note: The silkscreen on the board gives hints on the jumper positioning depending on the functions implemented.

The default debug configuration enables the board to be fully controlled by the S32K144 MCU (via I^{2}C) and the GUI. Figure 3 shows the default jumper and switch configuration for an FS2300 part.
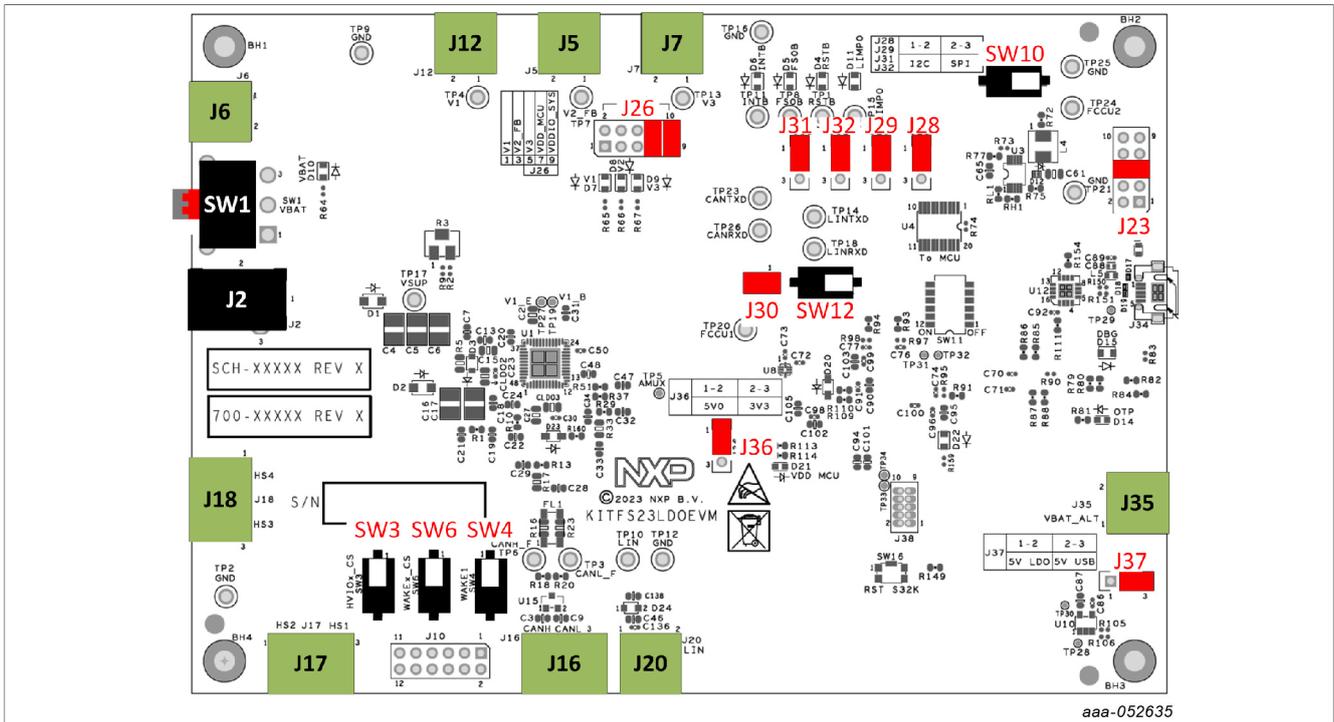
Figure 3. KITFS23LDOEVM default jumpers and switches configuration

Table 1. Switches functions

Position	Function	Description
SW1	VIN_SEL	Input voltage supply can come from battery (VIN_BAT) or from external power supply (VIN_PHX).
SW3	HVIOx_CS	Emulates a door switch as an example in cyclic sense configuration, by default for HVIO1, possible to change to HVIO2.
SW4	VSUP_TO_WAKE1	Configures the VSUP as input for WAKE1 pin (global).
SW6	WAKEx_CS	Emulates a door switch as an example in cyclic sense configuration, by default on WAKE2, possible to change to WAKE1.
SW9	LED_SIG_SAFETY	Enables LED signaling for safety pins values when ON (on bottom side).
SW10	OTP_DCDC_EN	Enables OTP DC-DC 8 V generator (see J30).
SW11	SAFETY_IO_EN	Enables safety I/Os and LVIOs signal propagation between FS23 and MCU when ON (see silkscreen).
SW12	OTP_EN	Enables OTP mode if OTP DC-DC enabled, else enables DBG mode (see J30).
SW16	MCU_RST	Pushing button to reset S32K144 MCU.

UM11879

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

9 / 73

Table 2. Jumpers functions

Position	Function	Description
J2	VIN_BAT	Input voltage from battery via Jack connector (40 V max).
J5	V2_OUT	Output for V2.
J6	VIN_PHX	Input voltage from external power supply via Phoenix connector (40 V max).
J7	V3_OUT	Output for V3.
J10	WAKEx_HVIOx_CONF	Configuration of WAKEx and HVIOx Input/Output modes and monitoring headers.
J12	V1_OUT	Output for V1.
J16	CAN_OUT	Output for CAN.
J17	HS12_OUT	Output for high-side drivers HS1 and HS2.
J18	HS34_OUT	Output for high-side drivers HS3 and HS4.
J20	LIN_OUT	Output for LIN.
J23	FCCU2_SEL	FCCU2 selection between HVIO1 or HVIO2 and monitoring headers.
J26	VDDIO_SEL	VDDIO selection between V1, V2, V3, or VDD_MCU.
J28	SPI_I2C_SEL1	MCU communication can be configured to SPI or I^2C.
J29	SPI_I2C_SEL2	MCU communication can be configured to SPI or I^2C.
J30	DBG_OTP_EN	Enables Debug and OTP modes (see SW10 and SW12.)
J31	SPI_I2C_SEL3	MCU communication can be configured to SPI or I^2C.
J32	SPI_I2C_SEL4	MCU communication can be configured to SPI or I^2C.
J35	5V0_PHX_IN	5.0 V LDO power supply from Phoenix connector.
J36	MCU_SUP_SEL	MCU power supply selection between 3.3 V or 5.0 V.
J37	5V0_SEL	5.0 V power supply either from USB or from LDO.

4.4.1 VIN selection (SW1/J2/J6)

The input voltage supply can come from the battery via a Jack connector (VIN_BAT) or from an external power supply via the Phoenix connector (VIN_PHX). Nominal VIN voltage is 12 V (40 V maximum).

Table 3. VIN selection (SW1/J2/J6)

Schematic label	Signal name	Description
SW1-1	VIN_BAT	VIN input voltage supply via Jack connector.
SW1-2	VIN_OFF	Board not supplied.
SW1-3	VIN_PHX	VIN input voltage supply via Phoenix connector.
J2-1	VIN_BAT	VIN_BAT input supply via Jack connector.
J2-2	GND	Ground.
J6-1	VIN_PHX	VIN_PHX input supply via Phoenix connector.
J6-2	GND	Ground.

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

10 / 73

4.4.2 V1, V2, V3 connectors (J12/J5/J7)

For FS2300, HVLDO1 can be configured to 3.3 V or 5.0 V, up to 100 mA with internal PMOS and up to 250 mA with external PNP.

HVLDO2 can be configured to 3.3 V or 5.0 V, up to 100 mA current capability.

HVLDO3 is either CAN FD block supply or can be configured to 3.3 V or 5.0 V, up to 150 mA current capability.

Table 4. V1, V2, V3 connectors (J12/J15)

Schematic label	Signal name	Description
J12-1	V1	V1 output power supply.
J12-2	GND	Ground.
J5-1	V2_FB	V2 output power supply.
J5-2	GND	Ground.
J7-1	V3	V3 output power supply.
J7-2	GND	Ground.

4.4.3 V2 optional diode bypass (R5)

HVLDO2 has an optional protection diode against short to battery. A 0 ohm resistor R5 ON allows the user to bypass the protection diode. R5 is ON by default.

Table 5. V2 optional diode bypass (J5)

Schematic label	Description
R5-ON	V2 protection diode is bypassed (local).
R5-OFF	V2 is protected against short to battery by an external diode (global).

4.4.4 HVLDO1 PNP configuration (R4/R6/R7/R8)

HVLDO1 can be configured with or without PNP.

V1_E (Pin 34) and V1_B (Pin 32) are either connected to:

- V1_IN if HVLDO1 configured without external PNP
- PNP_E and PNP_B if HVLDO1 configured with external PNP

Table 6. HVLDO1 PNP configuration (R4/R6/R7/R8)

Schematic label	Signal name	Description
R6-ON/R7-OFF	V1_IN	V1_E is connected to V1_IN for HVLDO1 configuration without external PNP.
R6-OFF/R7-ON	PNP_E	V1_E is connected to PNP_E for HVLDO1 configuration with external PNP.
R4-ON/R8-OFF	V1_IN	V1_B is connected to V1_IN for HVLDO1 configuration without external PNP.
R4-OFF/R8-ON	PNP_B	V1_B is connected to PNP_B for HVLDO1 configuration with external PNP.

4.4.5 MCU power supply selection (J36/J37/J35)

The MCU can be supplied by 3.3 V or 5.0 V.

The green LED D21 turns ON when the MCU is supplied by VDD_MCU. The MCU can be reset by pressing SW16.

When VDD_MCU is supplied from onboard 5.0 V LDO (J37 in position 1-2), a distinct power supply must be connected to J35 Phoenix connector. This way, the MCU has a distinct power supply domain from FS23 device under evaluation.

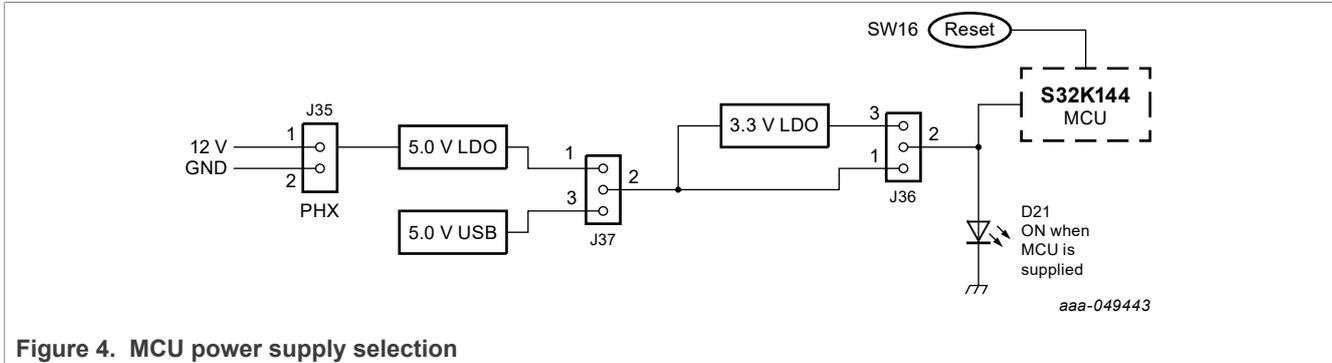Figure 4 describes the MCU power supply selection:

Figure 4. MCU power supply selection

Table 7. MCU power supply selection (J36/J37/J35)

Schematic label	Signal name	Description
J36-1-2	P5V0	VDD_MCU is connected to 5.0 V *(default)*.
J36-2-3	P3V3	VDD_MCU is connected to 3.3 V.
J37-1-2	P5V0_FROM_LDO	5.0 V power supply for VDD_MCU from LDO.
J37-2-3	P5V0_FROM_USB	5.0 V power supply for VDD_MCU from USB.
J35-1	12V_PHX_IN	12.0 V input for 5.0 V LDO supply via Phoenix connector.
J35-2	GND	Ground.

4.4.6 MCU communication SPI/I^{2}C selection (J28/J29/J31/J32)

MCU communication with FS2300 can be configured to I^{2}C *(default)* or SPI.

Table 8. MCU communication SPI/I^{2}C selection (J28/J29/J31/J32)

Schematic label	Signal name	Description
J28-1-2	I2C_SDA	I^{2}C is selected as MCU communication protocol.
J28-2-3	SPI_CSB	SPI is selected as MCU communication protocol.
J29-1-2	LVI5	LVI5 is available when I^{2}C is selected as MCU communication protocol.
J29-2-3	SPI_MOSI	SPI is selected as MCU communication protocol.
J31-1-2	I2C_SCL	I^{2}C is selected as MCU communication protocol.
J31-2-3	SPI_SCK	SPI is selected as MCU communication protocol.
J32-1-2	LVO6	LVO6 is available when I^{2}C is selected as MCU communication protocol.
J32-2-3	SPI_MISO	SPI is selected as MCU communication protocol.

4.4.7 CAN and LIN connectors (J16/J20)

The CAN transceiver is supplied internally by V3 regulator.

The LIN bus driver is supplied by VSHS supply input.

Table 9. CAN and LIN connectors (J16/J20)

Schematic label	Signal name	Description
J16-1	CANH	CAN bus high.
J16-2	CANL	CAN bus low.
J16-3	GND	Ground.
J20-1	LIN	LIN bus.
J20-2	GND	Ground.

4.4.8 Debug and OTP configuration (J30/SW10/SW12)

Debug mode is active when DEBUG pin (Pin 15) is set to 5 V. See Section 7.3.1. Debug mode disables the Watchdog and sets the CAN and LIN transceivers active by default.

OTP mode is active when DEBUG pin is set to 8 V. Once in OTP mode, specific keys are necessary to enter Test mode. See Section 7.3.2, or OTP programming process.

J30, SW10, and SW12 offer an onboard solution to set the necessary hardware context for Debug mode and Test mode entry.
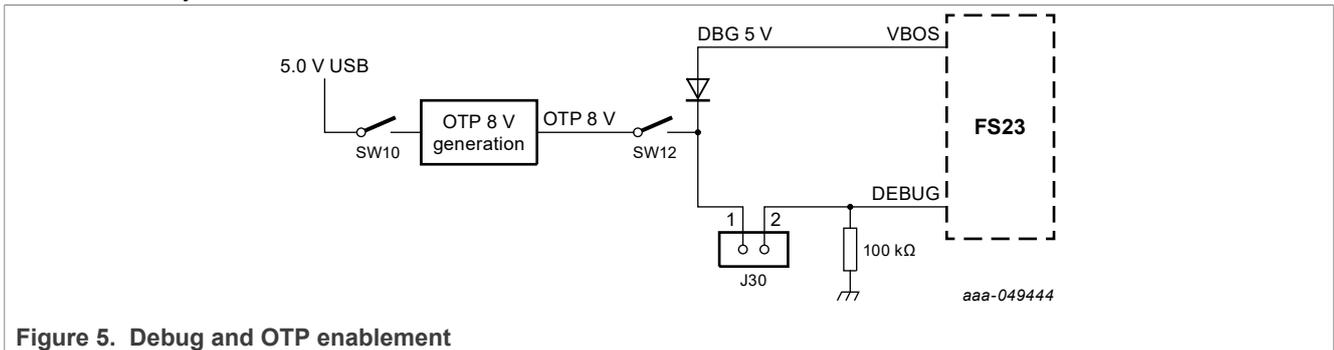
Figure 5. Debug and OTP enablement

Table 10. Debug and OTP configuration (J30/SW10/SW12)

Schematic label	Signal name	Description
J30-1-2	DBG_OTP_EN	Jumper ON enables DBG or OTP mode depending on SW10 and SW12 status.
SW10-1 (OFF)	Open switch	DC-DC disabled, no 8 V OTP generation.
SW10-2 (ON)	OTP_DCDC_EN	DC-DC enabled for 8 V OTP generation.
SW12-1 (OFF)	Open switch	OTP mode is disabled.
SW12-2 (ON)	OTP_EN	OTP mode is enabled if OTP DC-DC is ON. Debug mode is enabled if OTP DC-DC is OFF.

4.4.9 High-side driver connectors (J17/J18)

Each high-side driver (HSx) can be used to drive loads, or to perform cyclic sense when used in combination with a high-voltage input (WAKEx or HVIOx). High-side drivers are supplied by VSHS supply voltage.

Table 11. High-side driver connectors (J17/J18)

Schematic label	Signal name	Description
J17-1	HS2	HS2 output.
J17-2	GND	Ground.
J17-3	HS1	HS1 output.
J18-1	HS4	HS4 output.
J18-2	GND	Ground.
J18-3	HS3	HS3 output.

4.4.10 VDDIO selection jumper (J26)

VDDIO selection jumper selects V1, V2, V3, or VDD_MCU for VDDIO, and links to VDDIO_SYS.

Table 12. VDDIO selection jumper (J26)

Schematic label	Signal name	Description
J26-1-2	V1	Set VDDIO to V1 output voltage.
J26-3-4	V2	Set VDDIO to V2 output voltage.
J26-5-6	V3	Set VDDIO to V3 output voltage.
J26-7-8	VDD_MCU	Set VDDIO to VDD_MCU output voltage.
J26-9-10	VDDIO_SYS	Set VDDIO_SYS to VDDIO.

4.4.11 I/Os configuration and monitoring headers

HVIOx, LVIOx, and WAKEx pins have multiple modes configurable via J23 and J10. Jumpers J23 and J10 can also be used as monitoring headers.
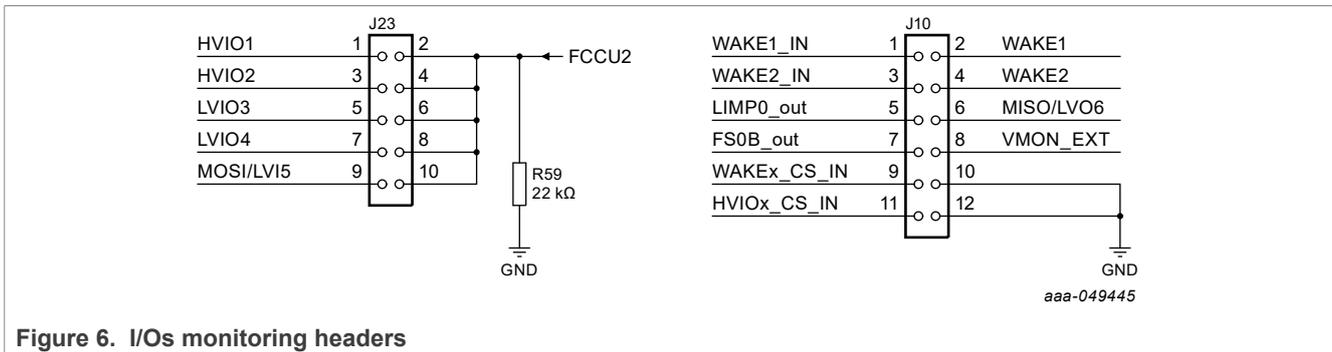
Figure 6. I/Os monitoring headers

Table 13. I/O configuration and monitoring headers (J23/J10)

Schematic label	Signal name	Description
J23-1	HVIO1	HVIO1 signal access.
J23-2/4/6/8/10	FCCU2	FCCU2 signal access.
J23-3	HVIO2	HVIO2 signal access.
J23-5	LVIO3	LVIO3 signal access.
J23-7	LVIO4	LVIO4 signal access.
J23-9	MOSI/LVI5	MOSI signal access in SPI mode. LVI5 in I^2C mode.
J10-1	WAKE1_IN	WAKE1 input signal access.
J10-2	WAKE1	WAKE1 signal access.
J10-3	WAKE2_IN	WAKE2 input signal access.
J10-4	WAKE2	WAKE2 signal access.
J10-5	LIMP0	LIMP0 signal access.
J10-6	MISO/LVO6	MISO signal access in SPI mode. LVO6 in I^2C mode.
J10-7	FS0B	FS0B signal access.
J10-8	VMON_EXT	VMON_EXT signal access.
J10-9	WAKEx_CS_IN	WAKEx cyclic sensing input signal access.
J10-10/12	GND	Ground.
J10-11	HVIOx_CS_IN	HVIOx cyclic sensing input signal access.

4.4.11.1 External voltage monitoring via VMON_EXT

The FS2300 can monitor an external regulator through VMON_EXT monitoring pin, accessible via J10-8 on the KITFS23LDOEVM. The regulator connected to VMON_EXT must be at least 1 V to be compatible with overvoltage and undervoltage monitoring thresholds. An external resistor bridge is used to divide the regulator voltage if higher than 1 V, and set the middle point to 1 V. The resistor bridge value can be adjusted by using a screwdriver on the R3 potentiometer.
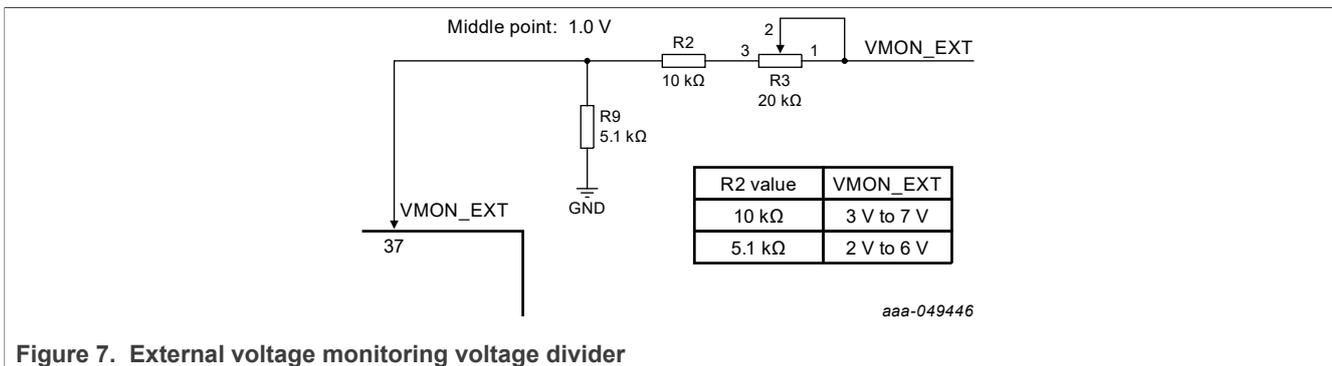
Figure 7. External voltage monitoring voltage divider

4.4.11.2 I/Os jumper configuration

The configuration of HVIOx, LVIOx, and WAKEx pins must be done by OTP. The hardware must be configured depending on the chosen functionality for HVIOx, LVIOx, and WAKEx pins. The following table gives the necessary jumper configuration for each pin-function association. A default configuration of the jumpers and switches is given in Section 4.4, matching the default I/O configuration in blue in the table.

Table 14. HSx configuration with corresponding resistor configuration

		Associated IO	Cyclic sense	Load/LED driver
HS1				Load between J17-3 and J17-2
HS2		HVIO1	Populate R24 and R14	Load btw J17-1 and J17-2
		HVIO2	Populate R24 and R31	
HS3				Load between J18-3 and J18-2
HS4		WAKE1	Populate R27 and R34 DNP R36	Load btw J18-1 and J18-2
		WAKE2	Populate R27 and R32 DNP R35	

		Cyclic sense	Global input	Local input
WAKE1		Populate R27 and R34 DNP R36	Populate R36 and DNP R34 Input via SW4 or J10-1	DNP R34 and R36 Input via J10-2
WAKE2		Populate R27 and R32 DNP R35	Populate R35 & DNP R32 Input via J10-3	DNP R32 and R35 Input via J10-4

	Simple input	Simple output	Cyclic sense	FCCU2 (input)	LIMP1 (output)	LIMP2 (output)
HVIO1	Available	Available	Populate R24 and R14	J23-1-2	Available	
HVIO2	Available	Available	Populate R24 and R31	J23-3-4		Available
LVIO3	Available	Available		J23-5-6	Available	
LVIO4	Available	Available		J23-7-8		Available
LVI5	Available if I²C			Available if I²C J23-9-10		
LVO6		Available if I²C			Available if I²C	

Regarding J23, remove any jumper between J23-1-2 and J23-3-4 when HVIOx are used as output, or when voltage is superior to 5 V.

4.4.11.3 Testing the cyclic sense function with onboard switches

HVIO1, HVIO2, WAKE1, and WAKE2 pins can be configured for cyclic sensing when coupled with their corresponding HSx.

The KITFS23LDOEVM provides a means to emulate door switches sensing with onboard switches, as an application example for the cyclic sense configuration.
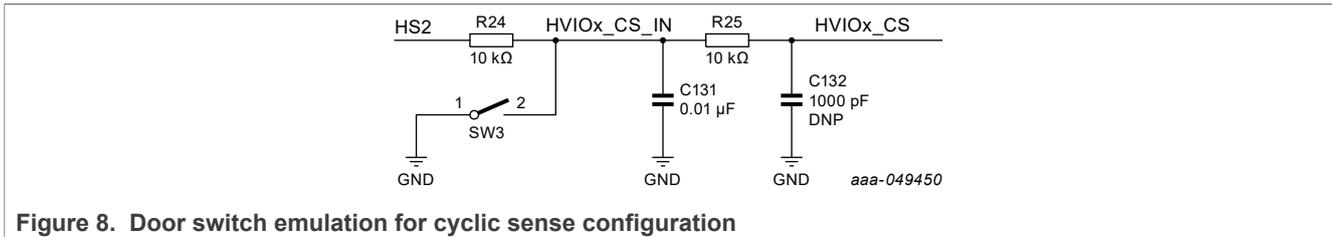
Figure 8. Door switch emulation for cyclic sense configuration

Table 15. Cyclic sense application examples (SW3/SW6)

Schematic label	Signal name	Description
SW3-1 (OFF)	Open switch	HVIOx senses HS2 output.
SW3-2 (ON)	HVIOx_CS	HVIOx senses 0 V (GND).
SW6-1 (OFF)	Open switch	WAKEx senses HS4 output.
SW6-2 (ON)	WAKEx_CS	WAKEx senses 0 V (GND).

4.4.11.4 Setting VSUP as an input for WAKE1 pin (global)

The WAKE1 and WAKE2 pins can be configured as global input pins.

The KITFS23LDOEVM provides a means to test the global input pin configuration with onboard switches.

Switching SW4 ON configures VSUP as input for WAKE1 pin.
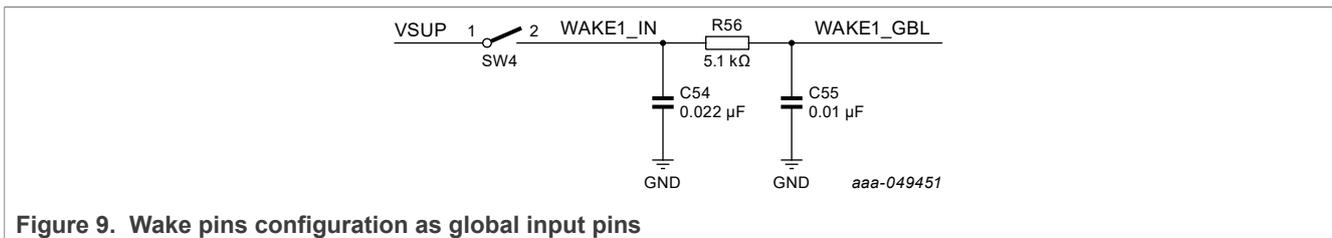
Figure 9. Wake pins configuration as global input pins

Table 16. Wake pins as global input application examples (SW4)

Schematic label	Signal name	Description
SW4-1 (OFF)	Open switch	Place input on WAKE1_IN (J10-1).
SW4-2 (ON)	VSUP_TO_WAKE1	VSUP configured as input for WAKE1 pin (global).

4.4.12 LED signaling

LED signaling is enabled by switches SW8/SW9 and jumper J27. The LED signaling can be turned OFF manually at any time to avoid undesired losses and obtain more accurate current consumption measurements.

LED signaling displays the state of the following signals:

- Signals: RSTB, FS0B, INTB, LIMP0; MCU status pin (to be used during software development for debugging purposes)
- Power: HVBUCK regulator output, HVLDOx regulator outputs
- Power supply: VBAT, MCU supply
- OTP control: Debug 5 V, OTP 8 V
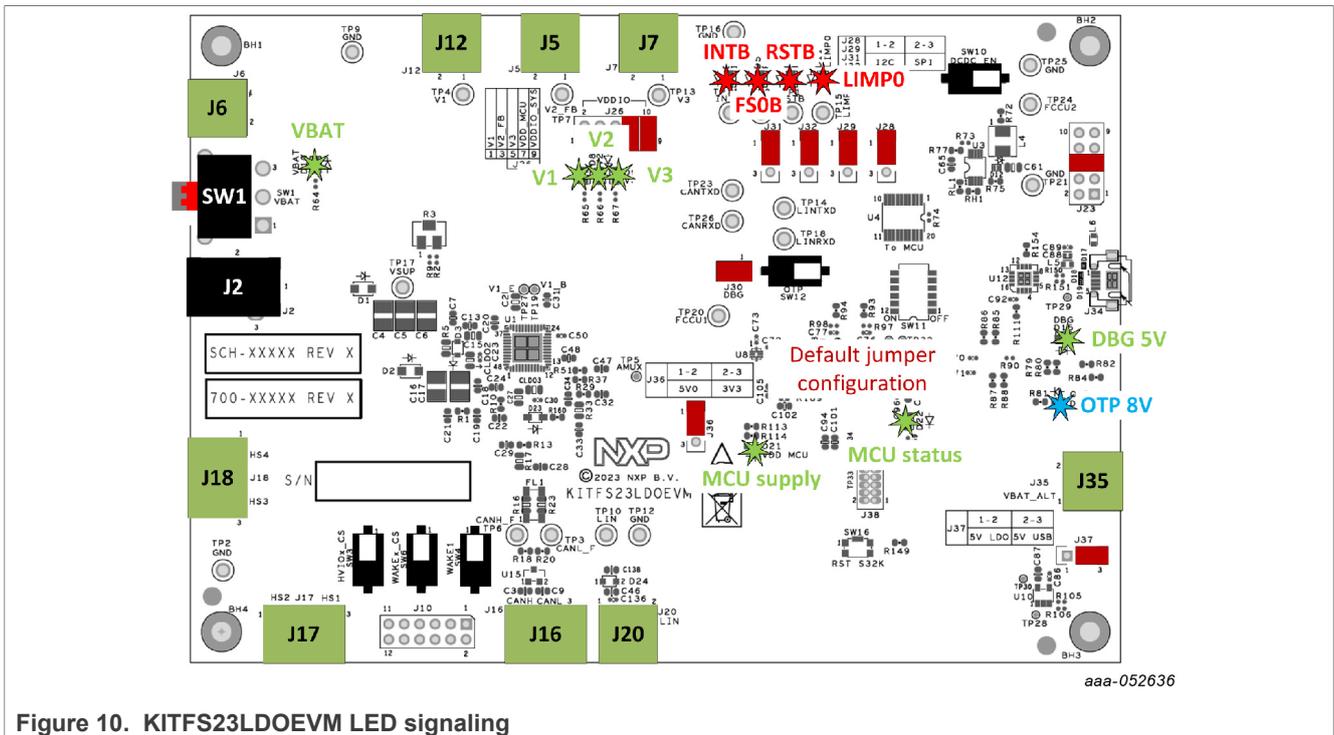
Figure 10. KITFS23LDOEVM LED signaling

UM11879
All information provided in this document is subject to legal disclaimers.
© 2023 NXP B.V. All rights reserved.

User manual
Rev. 2 — 25 September 2023

18 / 73

4.4.13 Test points

The KITFS23LDOEVM board has several test points that facilitate access and measurements. This section summarizes the available test points, how they are color-coded, and whether they are a test point component with a part number or a pad test point with no part number.

Yellow: Test loop access to safety outputs and analog signals

Red: Test loop access for power supplies and CAN/LIN bus

Black: Test loop access to GND
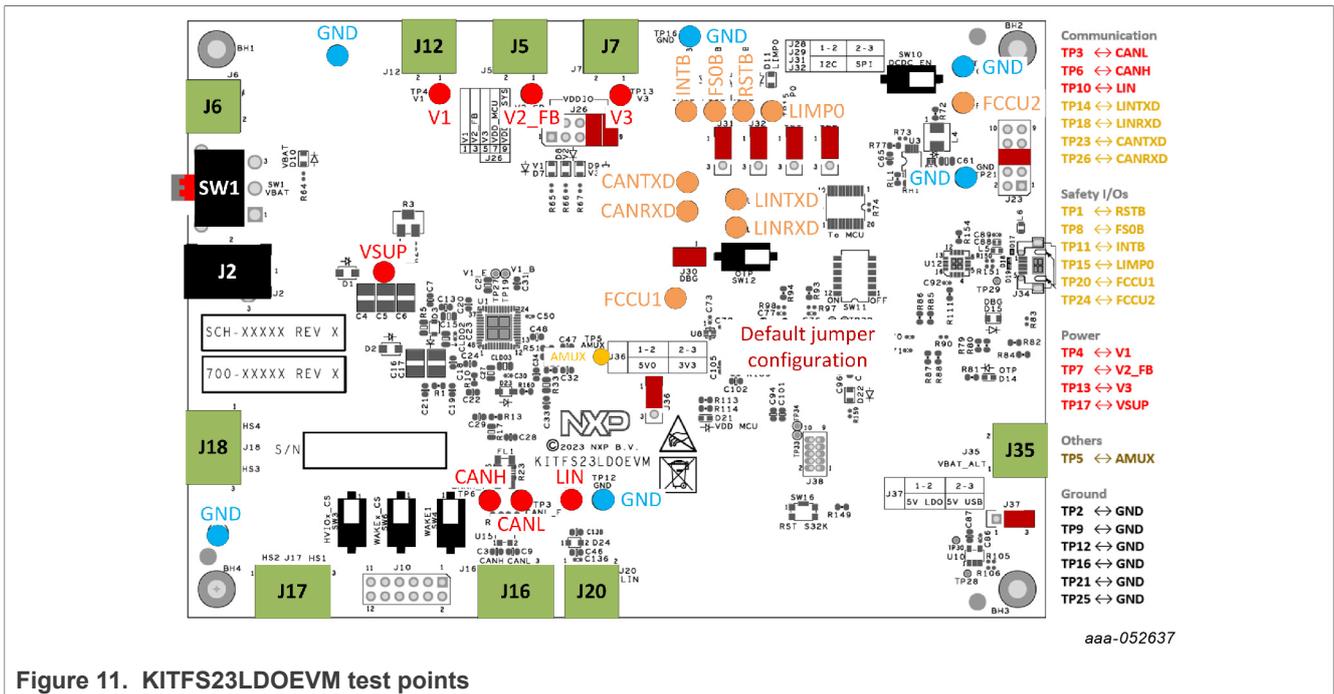
Figure 11. KITFS23LDOEVM test points

4.4.14 Schematic layout and bill of materials

The board layout and bill of materials for the KITFS23LDOEVM evaluation board are available at https://www.nxp.com/KITFS23LDOEVM

5 Installing and configuring software tools

5.1 Flashing S32K144 MCU GUI firmware

The KITFS23LDOEVM is delivered with the S32K144 firmware already flashed.

5.2 Installing the FS23 NXP GUI software package

To install the FS2300 NXP GUI, first download or obtain the NXP GUI package, then follow the instructions below:

1. Open the NXP GUI package. Unzip and open the 1 - NXP_GUI_Setup folder

aaa-049454

Figure 12. GUI folder list

2. Double click **NXP_GUI-version-Setup.exe** to launch the application and follow the instructions.

aaa-049455

Figure 13. NXP_GUI-version-Setup.exe

3. Click **Next** when the initial application setup window appears.

aaa-049456
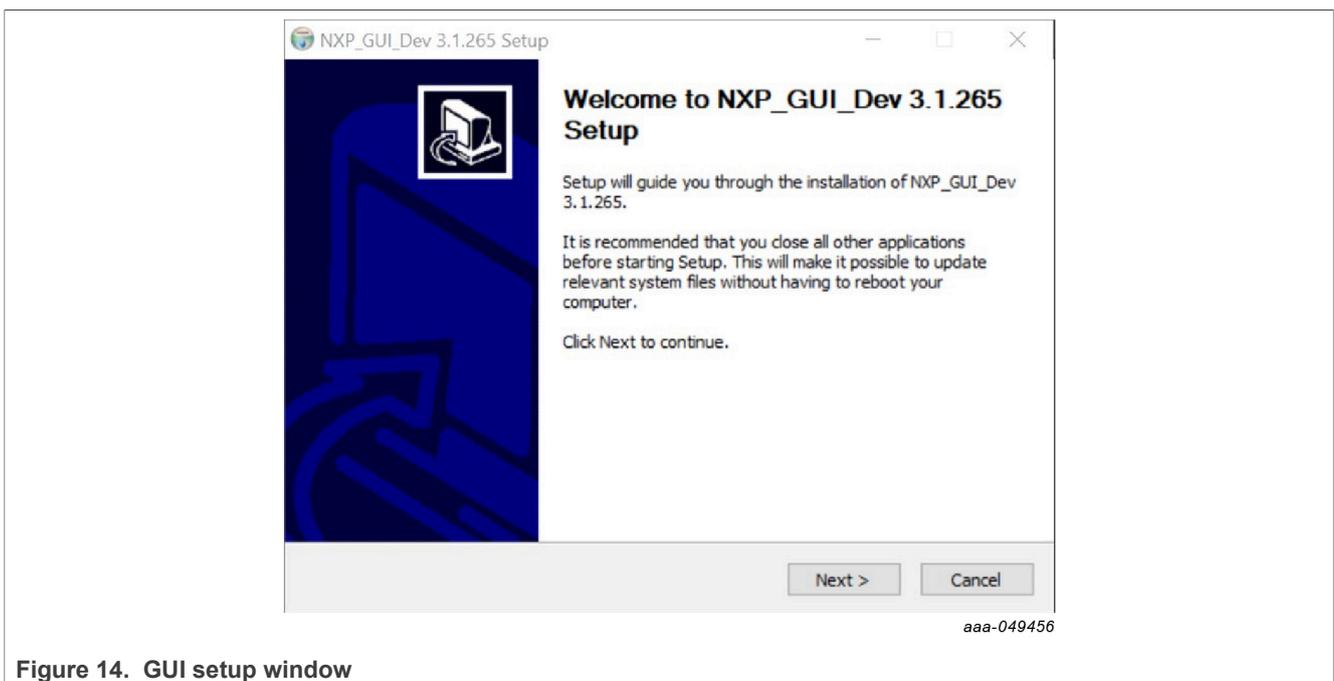
Figure 14. GUI setup window

4. On the License Agreement window, read the license information and click **I Agree**.

aaa-049457

Figure 15. GUI license agreement
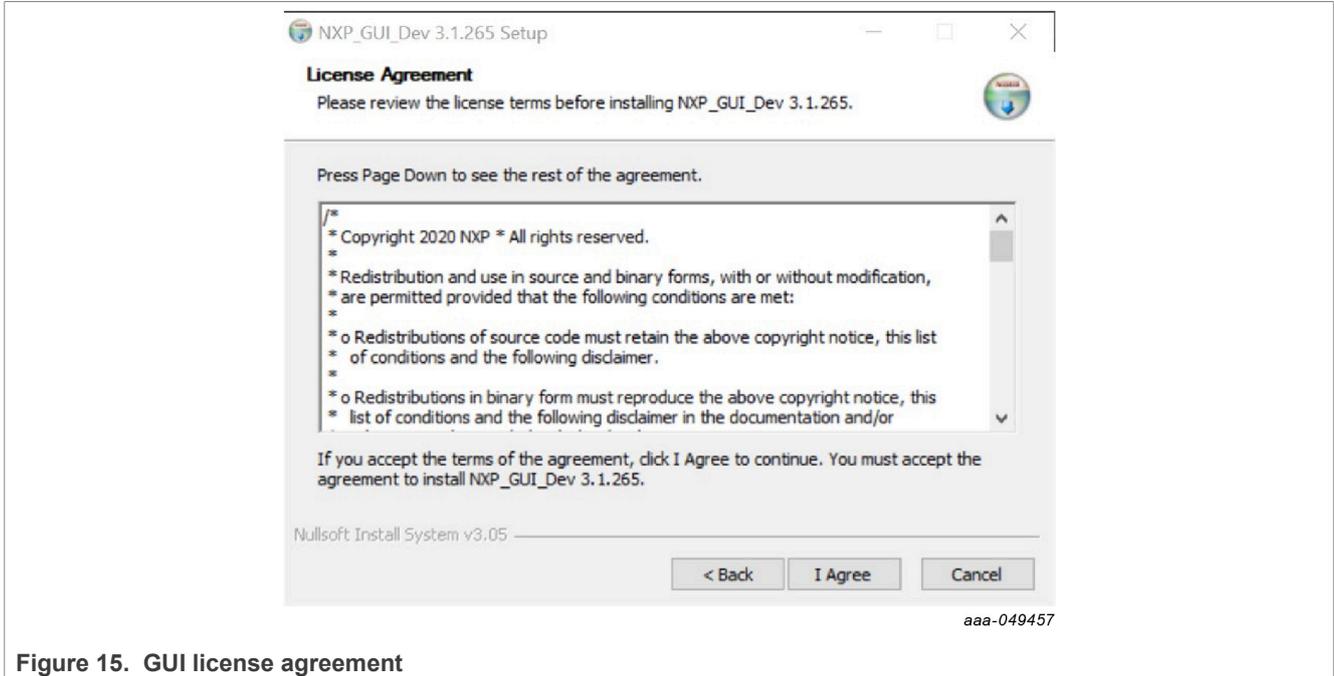
5. In the Choose Components window, select the GUI components to install, then click **Next**.

aaa-049458

Figure 16. GUI choose components to install
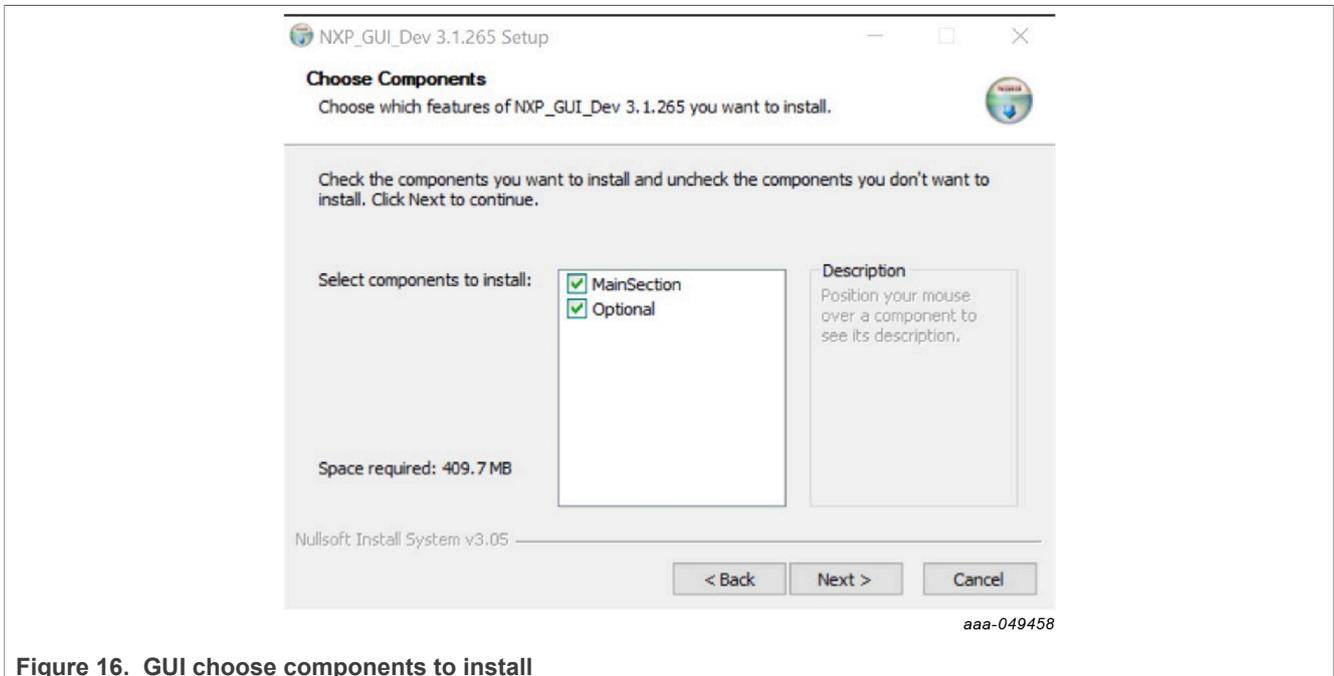
6. In the Choose Install Location window, choose the folder to install the GUI.

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

21 / 73

aaa-049459
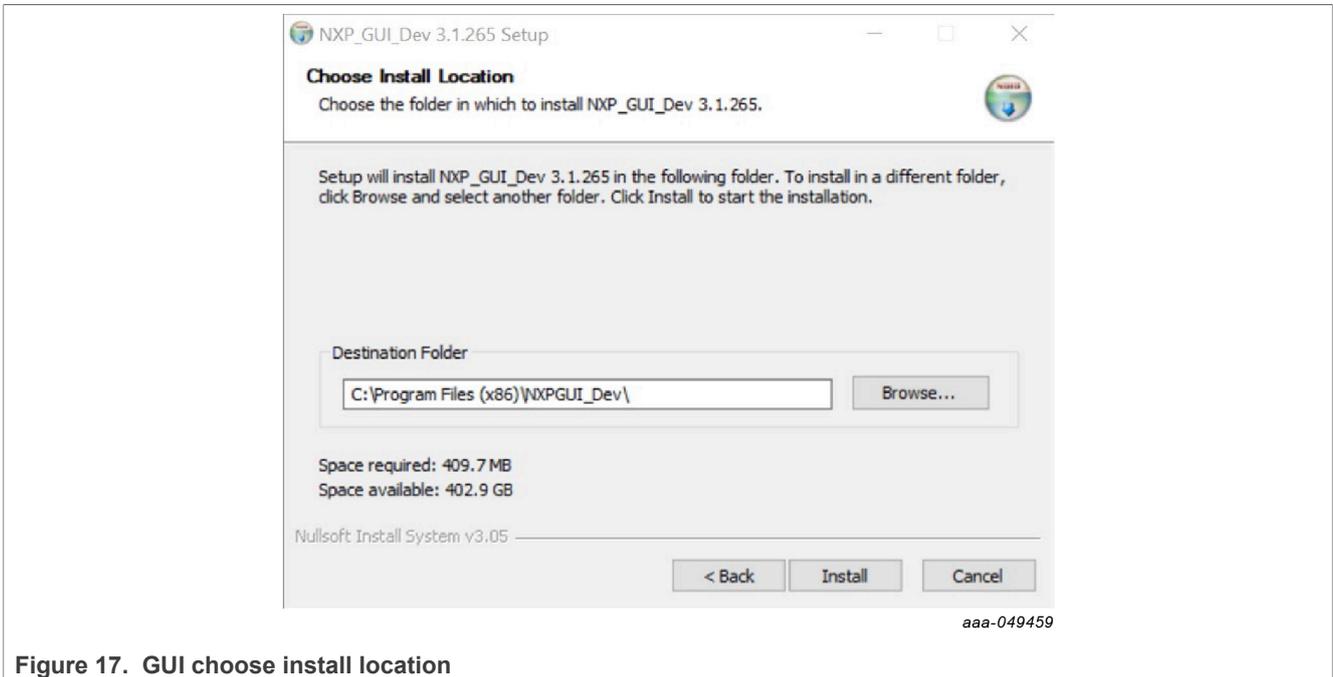
Figure 17. GUI choose install location

7. In the Completing Setup window, select the following options:
   • Run NXP_GUI
   • Show Readme
   Then click **Finish** to complete the installation.

aaa-049460
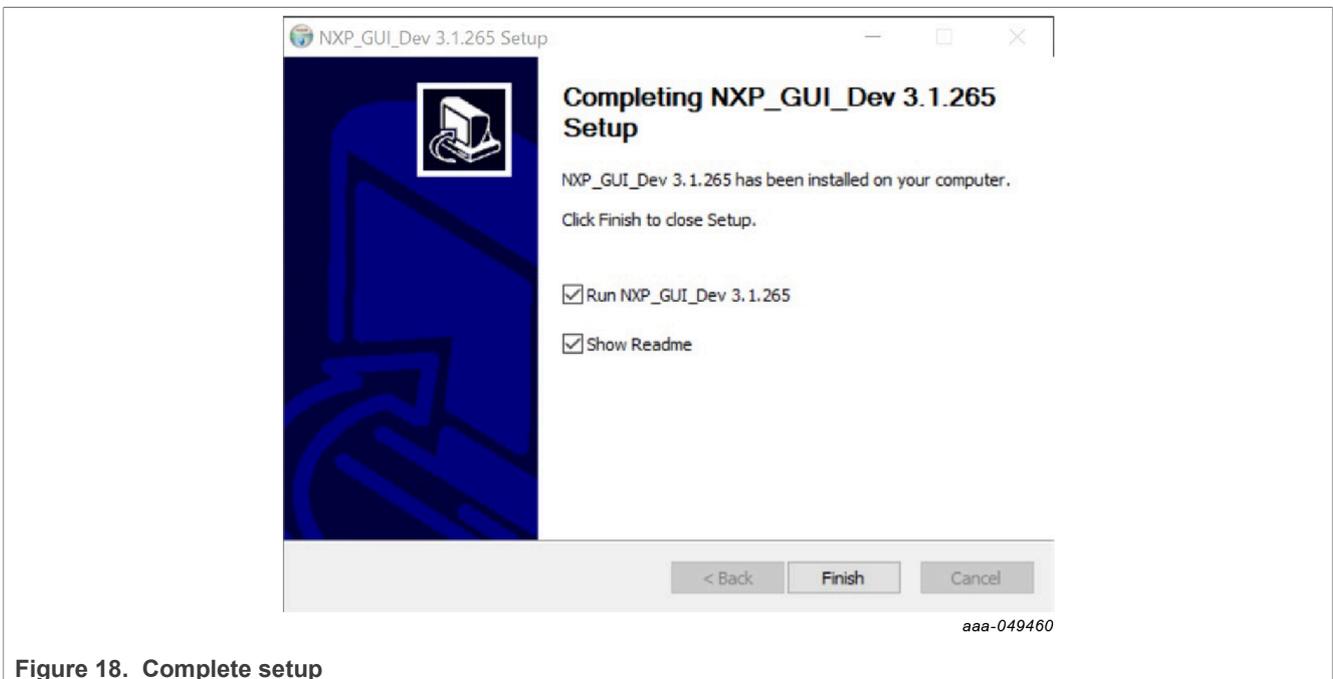
Figure 18. Complete setup

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

22 / 73

5.3 Launching the FS23 NXP GUI

When the KITFS23LDOEVM kit is set up and the GUI is 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 on the NXPGUI icon to launch the GUI.

aaa-049461
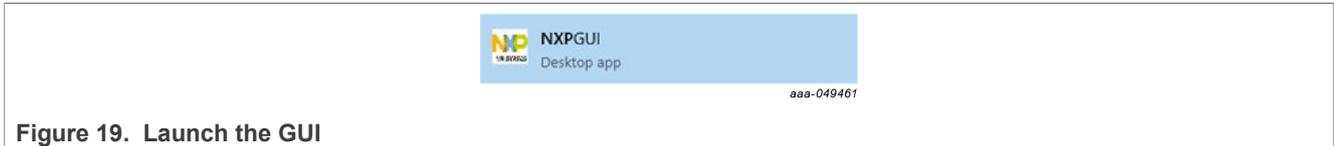
Figure 19. Launch the GUI

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

aaa-049462
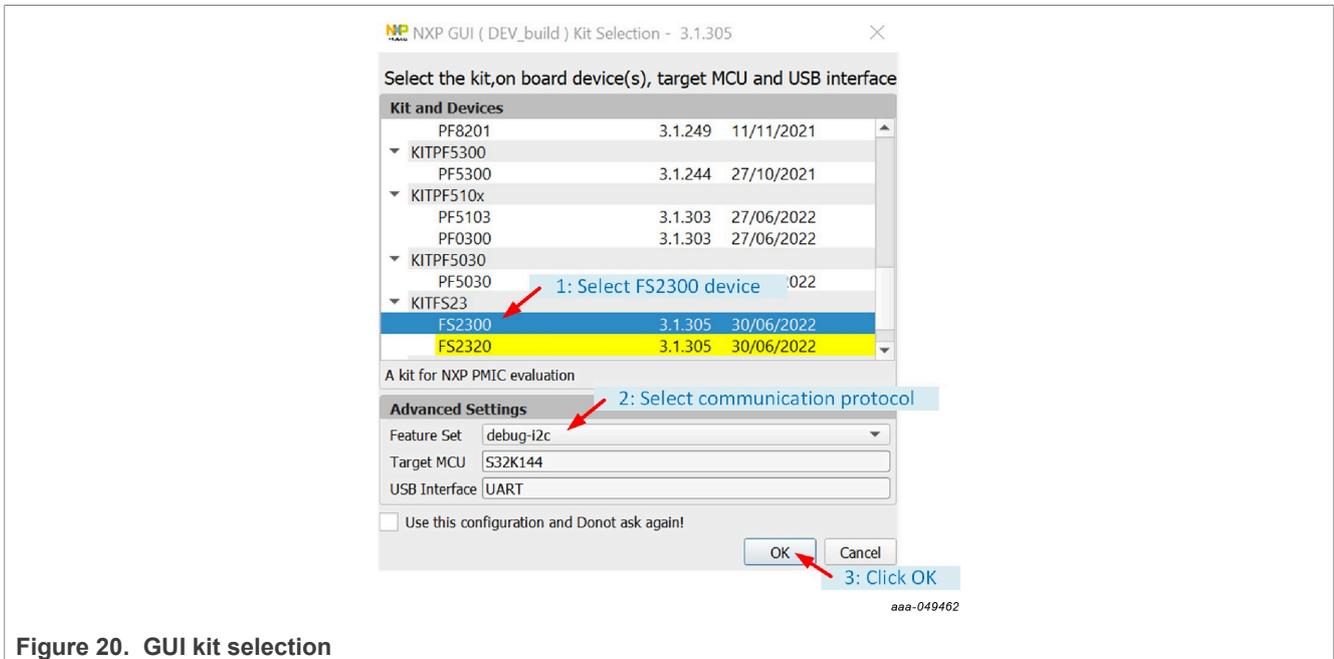
Figure 20. GUI kit selection

If the FS2300 part is empty, the default communication protocol is I^2C, and the GUI should be started with I^2C even if SPI protocol is desired. See Section 6.3.2 for SPI enablement from an empty part.

6 FS23 NXP GUI

This section gives guidance on use of the FS23 NXP GUI.

6.1 NXP GUI framework window

The Framework window consists of the following sections:

- **Connection toolbar:** Used to start communication with device, enter or exit Test/OTP modes, fix I^2C/SPI frequency and I^2C 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 (I^2C or SPI), shows firmware and GUI version, displays current Device mode.
- **Tools access bar:** Provides quick access to the FS23 evaluation tools and features.
  Note: Power tool is unavailable.
- **Tab content:** Shows the content of each tool or tab. There may be several tabs, boxes, or windows inside.

aaa-052487
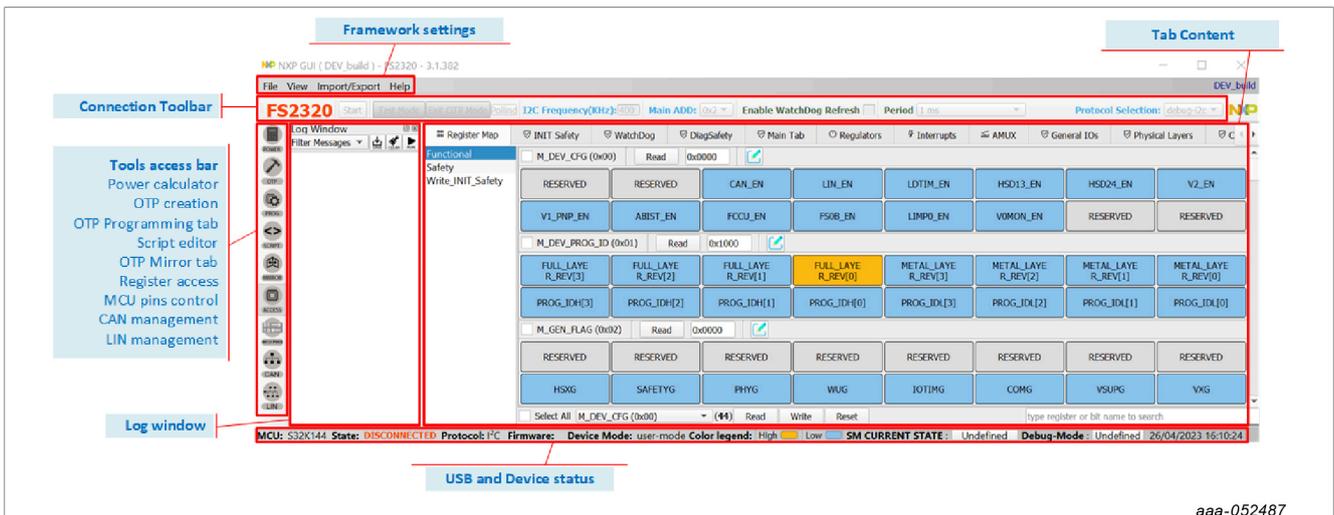
Figure 21. Framework window

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 corresponds to the Tools access bar
- Connection toolbar

aaa-049464
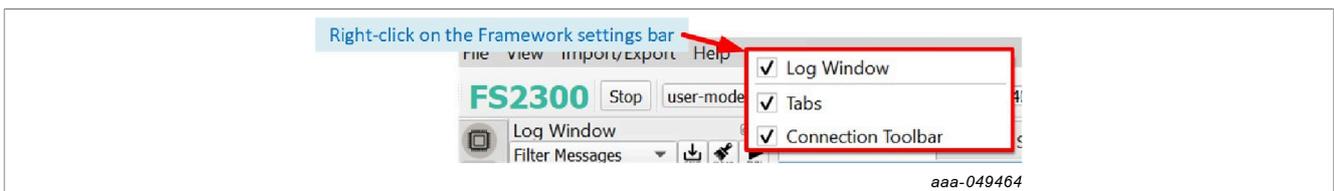
Figure 22. Framework settings bar

6.2 Framework settings bar

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

• File
• View
• Import/Export
• Help

aaa-049465

Figure 23. Framework items

6.2.1 File menu item

To choose to display or not the GUI Kit selection at Start or exit the application.

aaa-049466

Figure 24. File menu

• **Do not display GUI Kit selection at Start**: Check this box to always open NXP GUI as the current GUI version. Uncheck this box to display the product selection box at NXP GUI startup.
• **Exit**: Exits the NXP GUI application.

6.2.2 View menu item

The View menu contains the following options:

• **Display**: To enable or disable the Connection toolbar (enabled by default).

aaa-049467
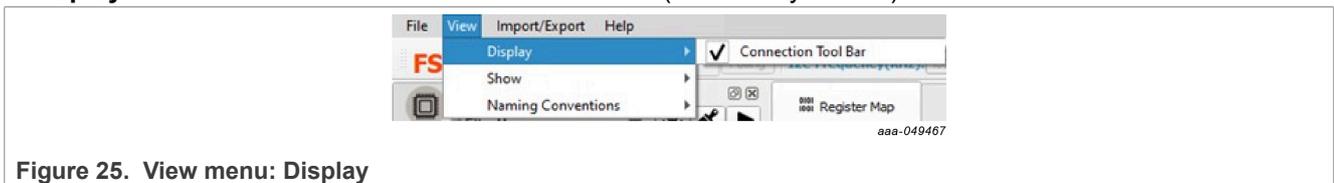
Figure 25. View menu: Display

• **Show**: Allows the user to access various sections of the GUI and display the Log window.
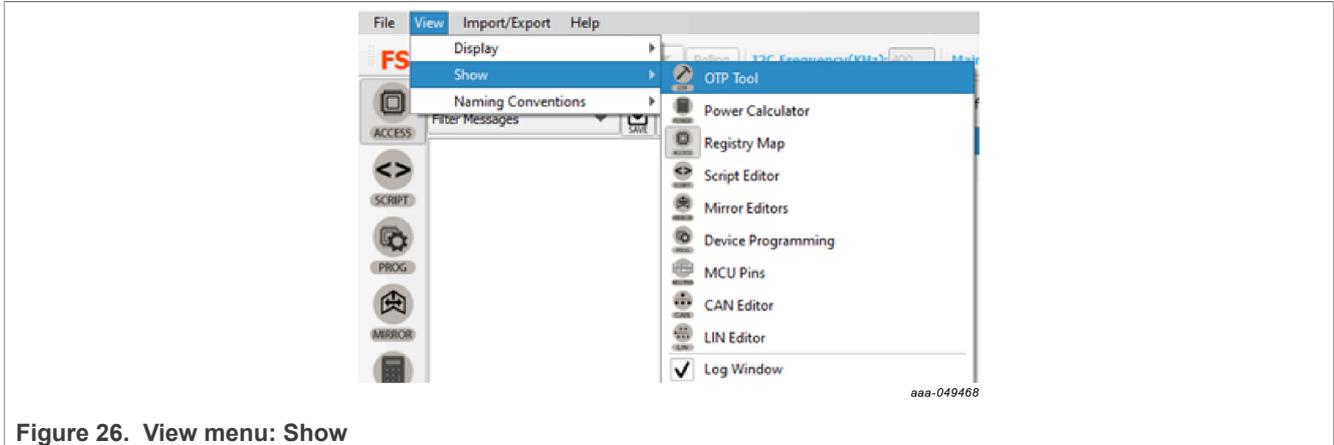
Figure 26. View menu: Show

- **Naming Conventions**: Allows the user to select Friendly or Register name display for OTP tool. Option enabled only when OTP tool is active.
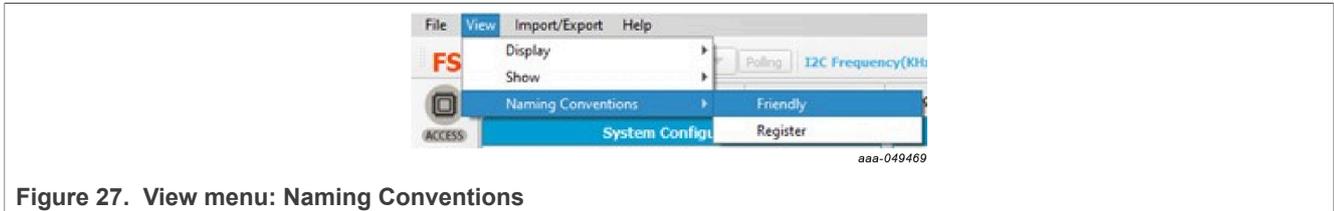
Figure 27. View menu: Naming Conventions

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 name in the OTP tool.
  Example: Power Sequence Slot For LVIO3 ⇔ LVIO3_SLOT_OTP.

6.2.3 Import/Export menu item

The Import/Export menu item allows the user to manage the files needed for Mirror emulation, for PROG tool, and for GUI configuration. This menu item is only active when the OTP tool has been selected. See Section 6.5.2.
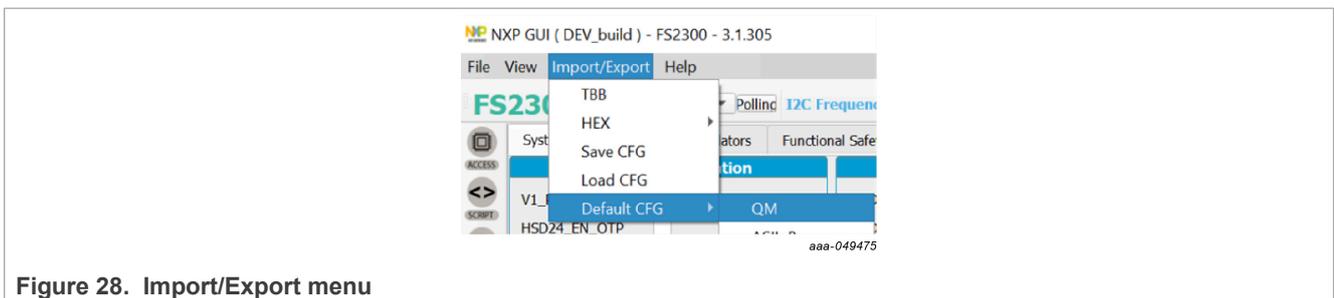
Figure 28. Import/Export menu

- **TBB**: Exports the OTP configuration into a TBB script file that can be used to load the Mirror registers using 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**: To save the current configuration as a CFG file.
- **Load CFG**: To load a previously saved configuration from a CFG file.
- **Default CFG**: To load a predefined OTP configuration in QM or ASIL B to use as a starting point.

6.2.4 Help menu item

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

aaa-049476
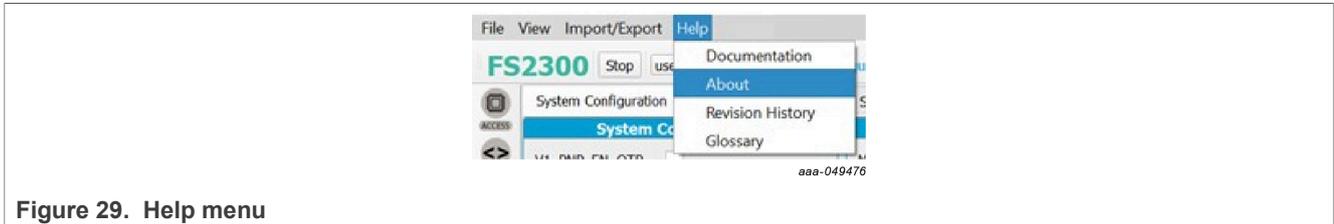
Figure 29. Help menu

- **Documentation:** Lists online NXP documentation related the FS2300 GUI.
- **About:** Displays the version number of the GUI currently installed.
- **Glossary:** Lists expanded names for acronyms used in the GUI.

6.3 Connection toolbar

The Connection toolbar menu is located directly below the Framework settings menu in the top left corner of the Framework window.

aaa-052488
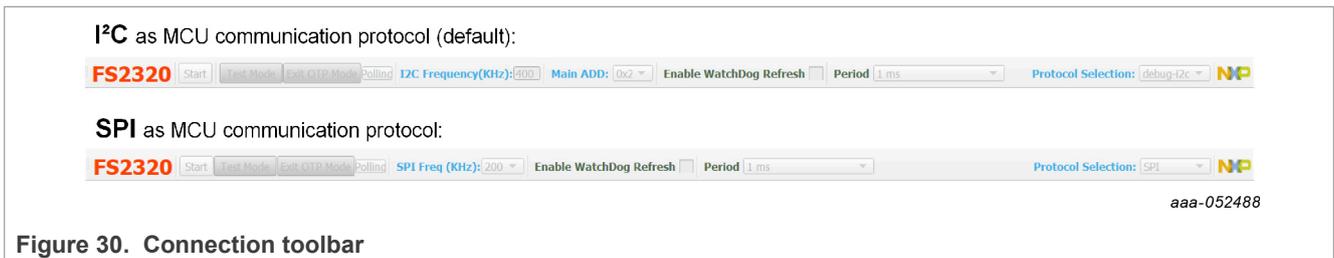
Figure 30. Connection toolbar

Note: The Connection toolbar is not displayed if not selected in the Frameworks setting>View>Display menu item.

The Connection toolbar allows the user to start or stop communication with the device, enter or exit Test mode and OTP mode, fix I^2C/SPI frequency (depending on chosen MCU communication protocol at start) and I^2C main address, enable Watchdog.

The Connection toolbar consists of the following items:

- **Start/Stop**: To open or close communication with the device.
- **Mode**: To enter or exit Test mode, and exit OTP mode.
- **Polling**: When started and connected, the GUI detects a mode change by polling mode status.
- **I^2C Frequency**: To set the I^2C frequency in kHz.
- **Main ADD**: To assign the I^2C address to the device.
- **SPI Frequency**: To set the SPI frequency in kHz.
- **Enable Watchdog Refresh**: To enable/disable the Watchdog refresh.
- **Period**: To set the Watchdog period.
- **Protocol selection:** To select MCU communication protocol between I²C and SPI. Prerequisite: MCU communication jumper positions must be changed on the EVB.

6.3.1 Device connection

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

aaa-052638

Figure 31. Device connection

When the S32K144 MCU is not connected through the USB port, the State indicator in the USB and Status bar shows **NOT DETECTED**, the FS2300 header text appears red, and the **Start** button is not available.

aaa-052639

Figure 32. Device connection: Not detected

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

aaa-052640

Figure 33. Device connection: Disconnected

Click **Start** to start communication with the FS2300.

At this point, the State indicator displays **CONNECTED** and FS2300 header text changes from red to green.

aaa-052641

Figure 34. Device connection: Connected

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

6.3.2 I^{2}C/SPI communication configuration

The FS2300 supports both I^{2}C and SPI communication protocol. I^{2}C is the default communication protocol. I^{2}C or SPI configuration settings for MCU side appear second in the Connection toolbar, allowing to choose the protocol frequency and device address if applicable.

aaa-052642

Figure 35. I^{2}C/SPI communication configuration

The display changes depending on the communication protocol chosen when launching the NXP GUI, see Section 5.3, when enabling SPI communication by OTP, or when selecting MCU communication protocol from Protocol selection box.

On the FS2300 side, the I^{2}C device address is set by OTP in I2CDEVADDR register. See Section 6.5.2.1.1.

If the FS2300 part is empty, the default communication protocol is I^{2}C, and the GUI should be started with I^{2}C even if SPI protocol is desired as outlined in Section 5.3. To enable SPI from an empty part, follow the instructions below:

1. Start the FS23 NXP GUI in I^{2}C mode and connect the KITFS23LDOEVM to the GUI by following the procedure in Section 7.2.
2. Enter Test mode to be able to write into the Mirror registers. See Section 7.3.2.
3. From the MIRROR tool, in Section 6.5.2.1.1, select only SPI_EN bit and Write in the Mirror registers.
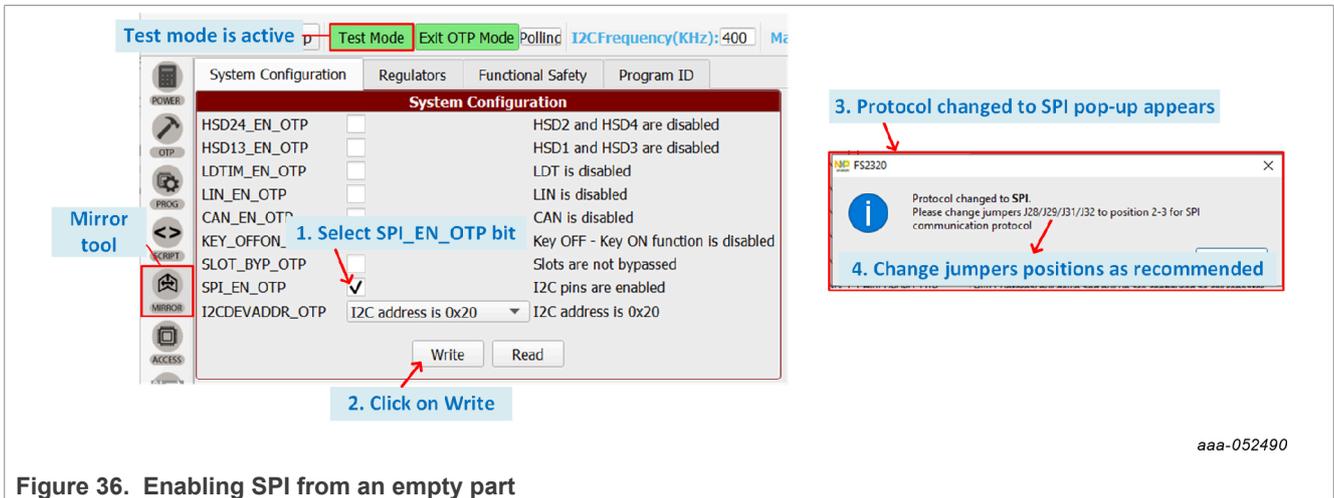
Figure 36. Enabling SPI from an empty part

4. On the KITFS23LDOEVM, change jumpers J28/29/31/32 to position 2-3.
5. The GUI should change to SPI mode. If not, select SPI from the the Protocol selection box. After ensuring that Test mode is still active, the MIRROR tool can be accessed to continue the modification of Mirror registers using SPI communication protocol.

The Protocol Selection box is used to change MCU communication protocol from the GUI. OTP and hardware changes must be made up-front accordingly.

The currently selected protocol can be checked in the USB and Device status bar.

Figure 37 illustrates these check points.
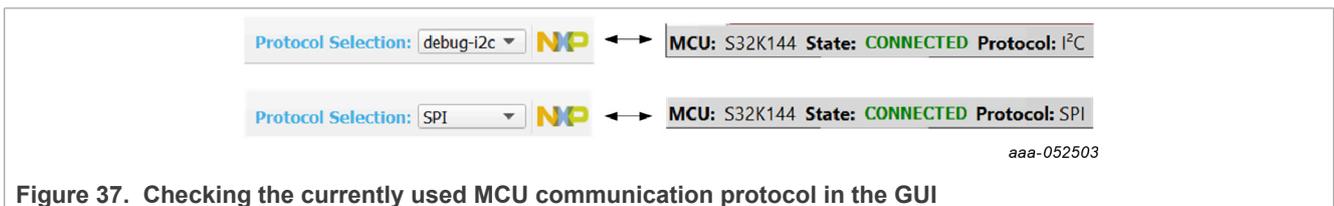
Figure 37. Checking the currently used MCU communication protocol in the GUI

6.3.3 Watchdog management

The FS2300 provides Watchdog monitoring with keys as a functional safety feature. Watchdog monitoring requests synchronize actions from SBC and MCU. The Watchdog feature can be disabled in QM version only. The Watchdog is always enabled for an ASIL B part.

6.3.3.1 Watchdog enablement and configuration on SBC side

On the FS2300 side, when the part is QM, Watchdog disablement is possible by OTP. Watchdog monitoring default configuration is done via SPI/I^{2}C via ACCESS tool.
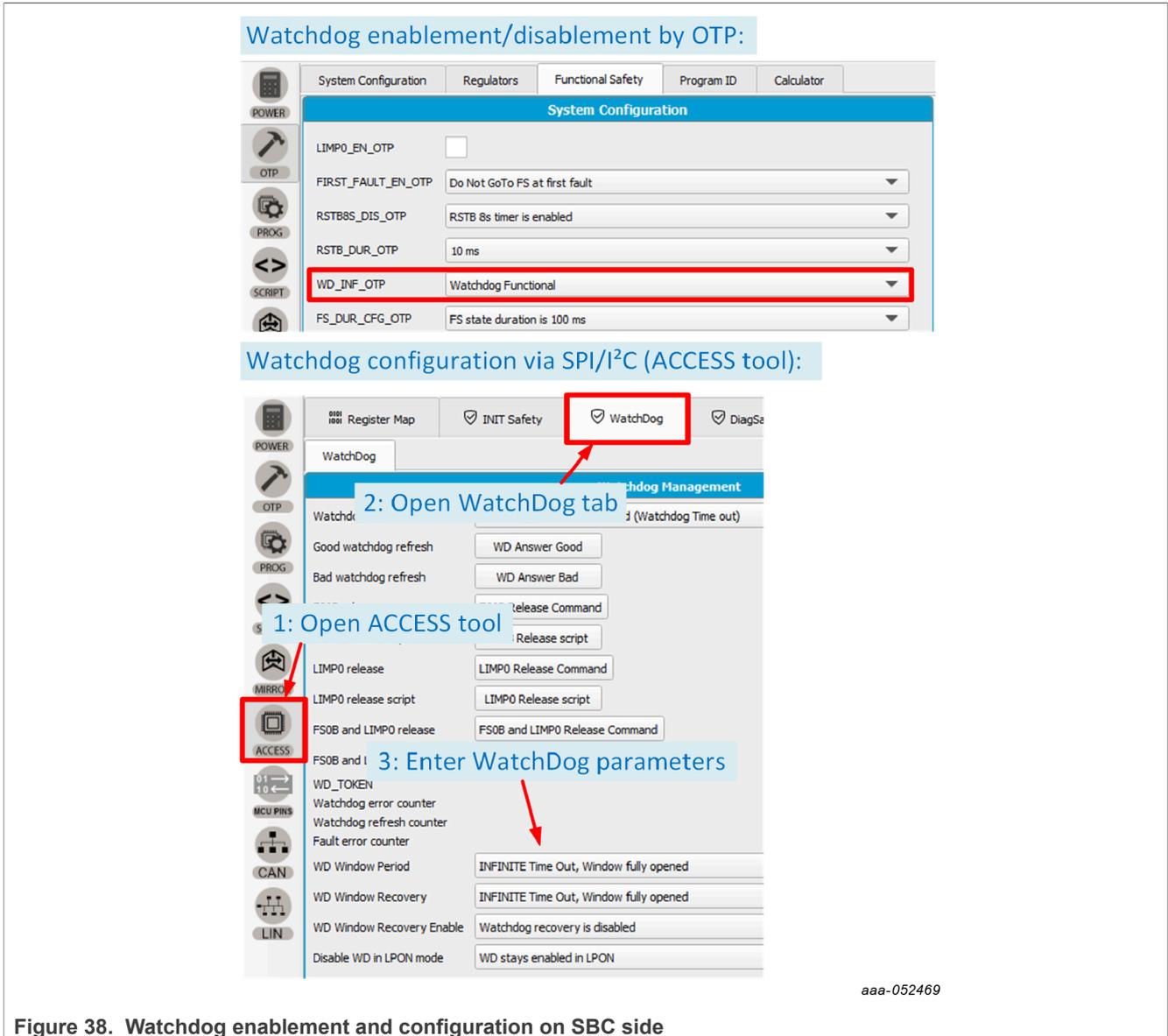
Figure 38. Watchdog enablement and configuration on SBC side

6.3.3.2 Watchdog configuration on MCU side

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

aaa-052643

Figure 39. Enable Watchdog Refresh

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

The steps for Watchdog enablement are described in Figure 40:

aaa-052644
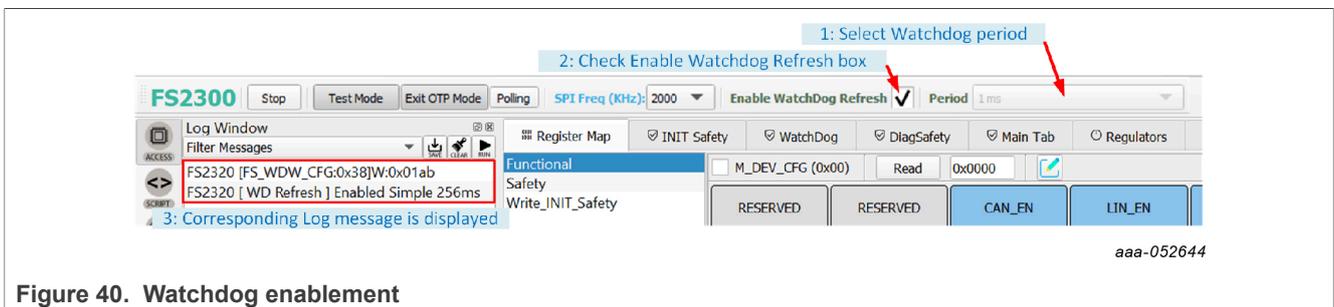
Figure 40. Watchdog enablement

1. Select Watchdog period via the Period selection box in the Connection toolbar.
2. Enable Watchdog Refresh by checking the 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.

aaa-052645
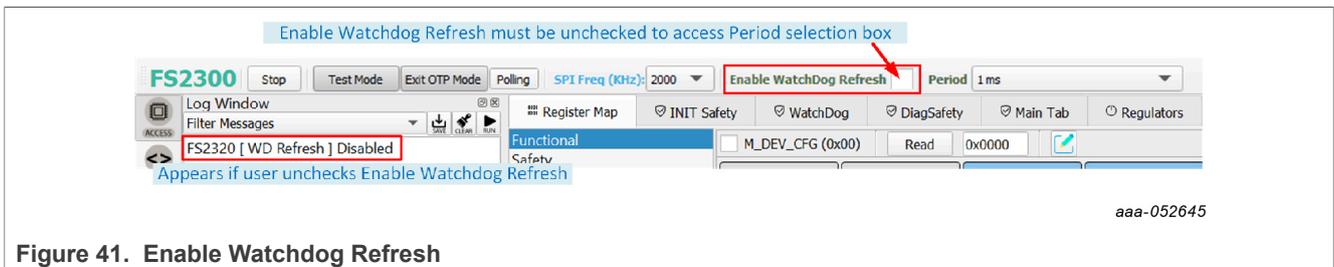
Figure 41. Enable Watchdog Refresh

6.4 USB and Device status bar

The USB and Device status bar is at the bottom of the Framework window. This toolbar gives information on the GUI and firmware version, on the USB connection status, and on FS23 active modes.

aaa-052646
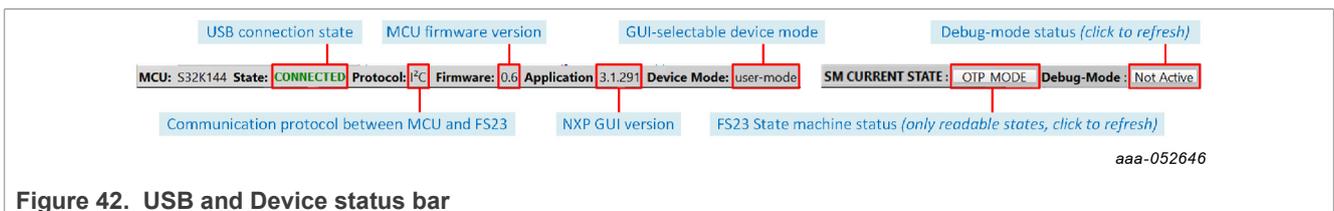
Figure 42. USB and Device status bar

6.5 Tools access bar

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:

Figure 43. Tools access bar
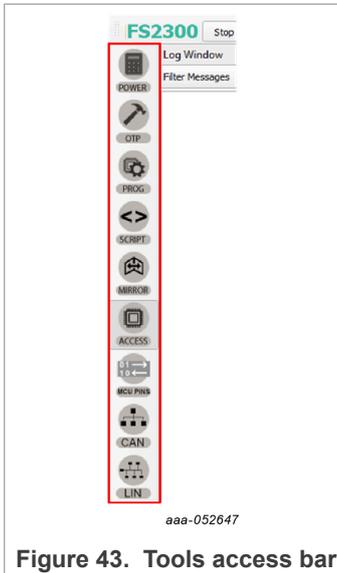
- **POWER**: To compute power dissipation of the device in a given application

- **OTP**: To create and save OTP configuration files, with customer and program details

- **PROG**: To manage OTP fuse-burning process

- **SCRIPT**: To create, open, save, and run scripts

- **MIRROR**: Provides access to Mirror registers

- **ACCESS**: Provides access to I^{2}C/SPI registers via Register Map or thematic tabs

- **MCU PINS**: To read and set MCU I/O pins

- **CAN**: To manage CAN communication

- **LIN**: To manage LIN communication

6.5.1 POWER

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

Figure 44 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 or IC Power Dissipation buttons. Load and Save Power Config buttons are used to resume the power dissipation calculation later, by saving and loading a text file.
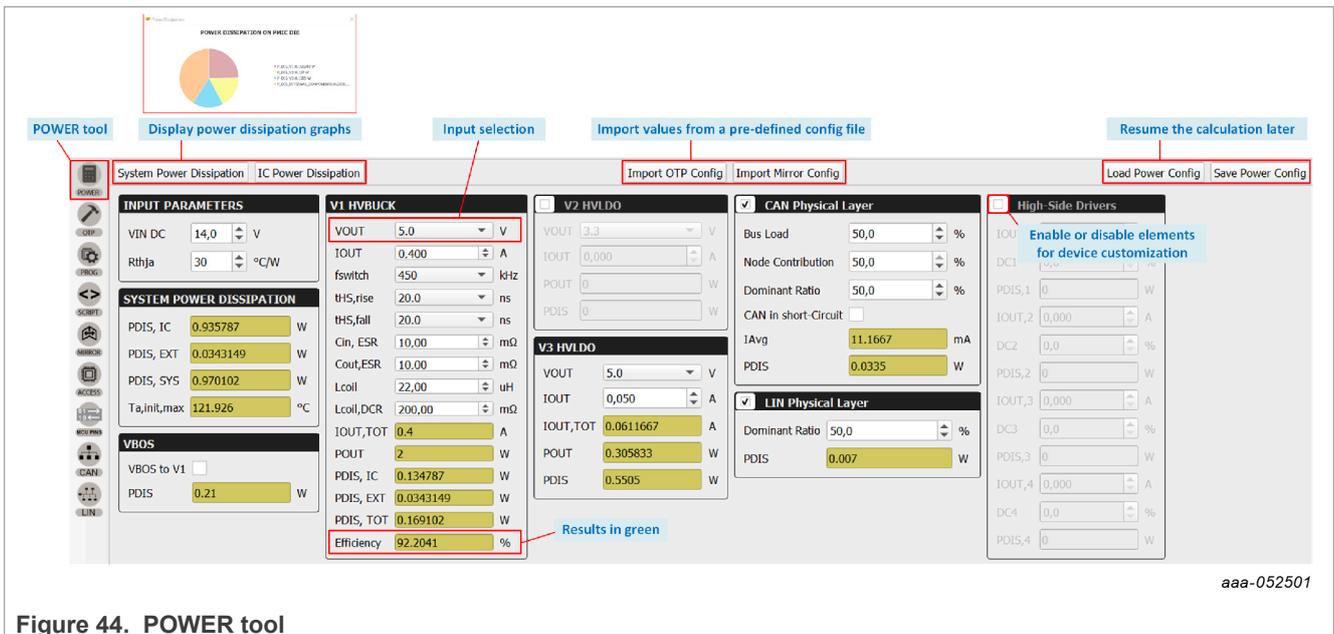
Figure 44. POWER tool

6.5.2 OTP

The OTP tool is used to enter an OTP configuration. The OTP configuration can be saved as a CFG file or exported as a TBB file:

- The CFG file is 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 Debug mode.
- The 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 Debug mode or to burn OTP fuses using the PROG tool.

In the OTP tool, the displayed parameters differ depending on the selected device type (ASIL B or QM) in Program Details box. The OTP configuration is not addressed here. For information on OTP configuration contents, refer to the FS23 data sheet.

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

- OTP Parameters Setting
- OTP Details
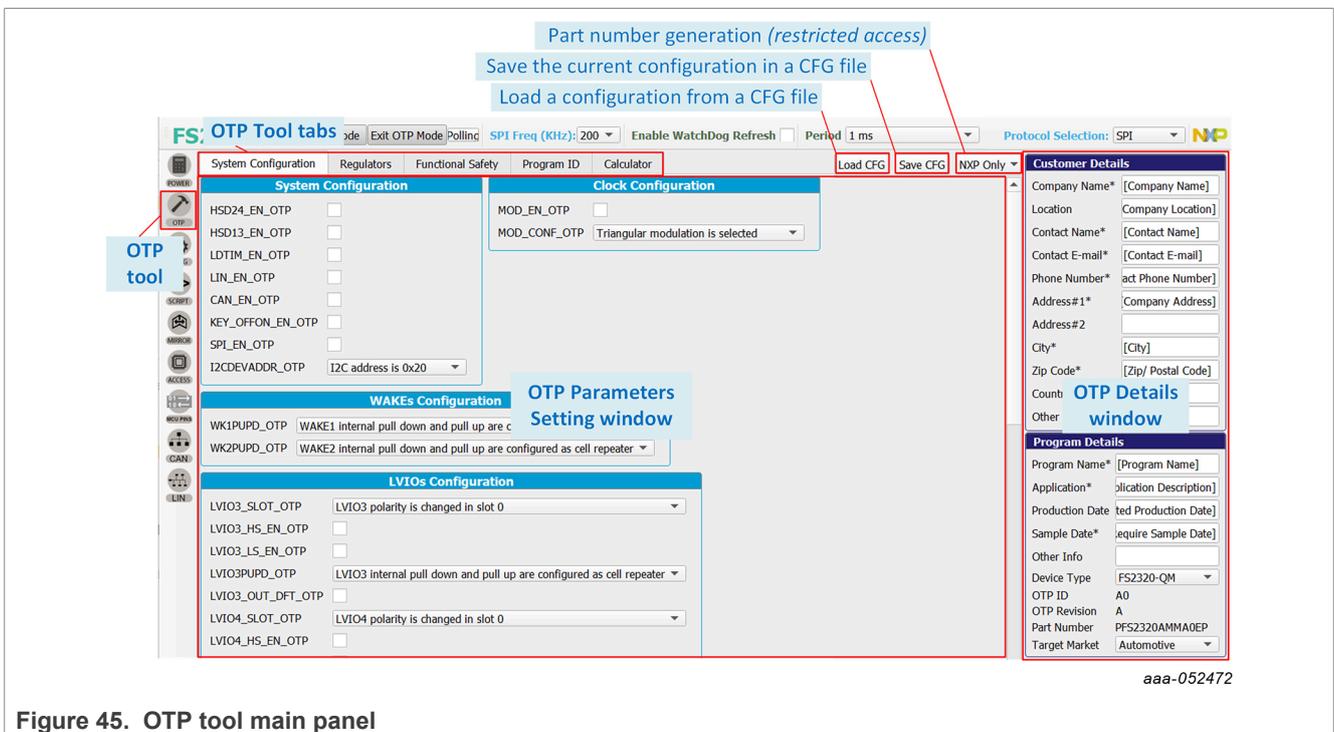
Figure 45. OTP tool main panel

6.5.2.1 OTP Parameters Setting window

The OTP Parameters Setting section is organized into five tabs.

Note: The Functional Safety window differs depending on the selected safety level. Windows are the same for QM and ASIL B parts for all other tabs. Safety level can be selected in the Program Details box of Section 6.5.2.2.

6.5.2.1.1 System Configuration tab

The System Configuration tab provides a means of setting miscellaneous FS2300 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.

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

33 / 73
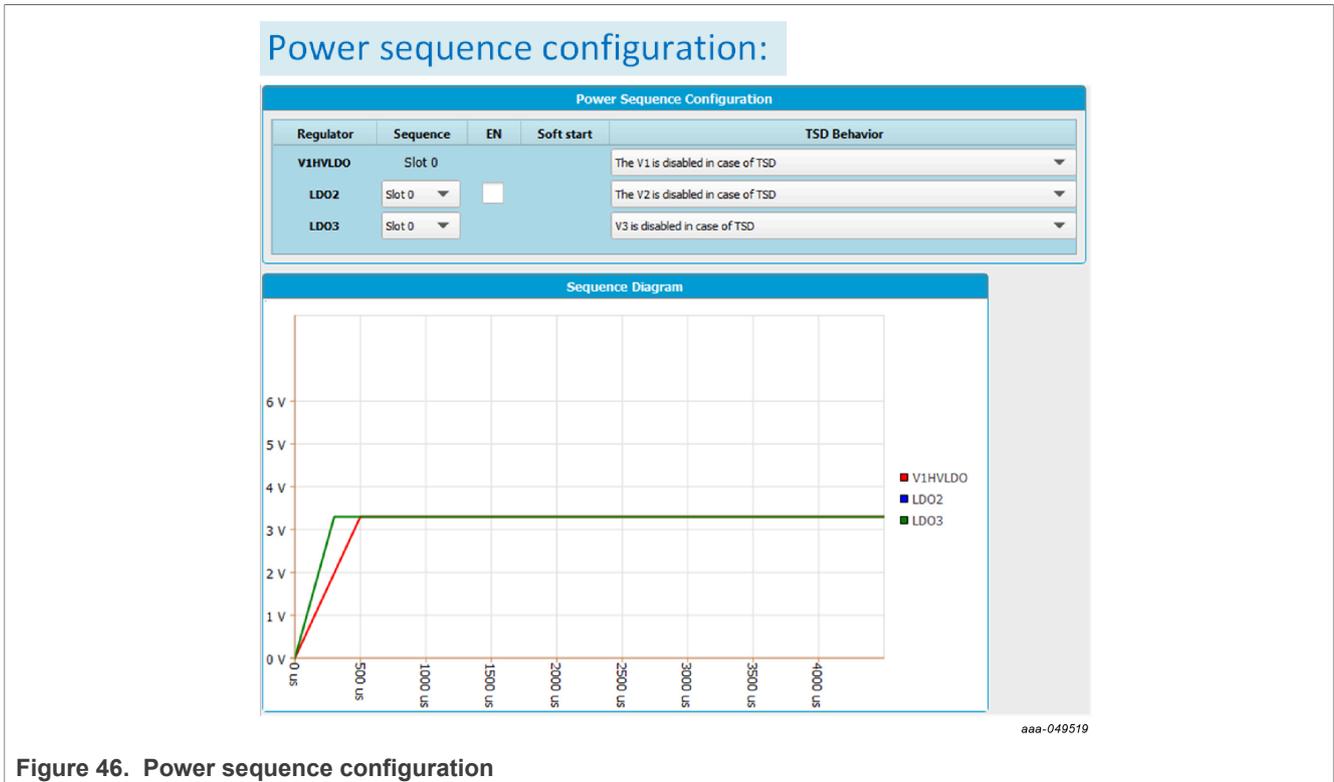
Figure 46. Power sequence configuration

6.5.2.1.2 Regulators tab

The Regulators tab allows the user to set OTP Parameters for HVLDO1, HVLDO2, and HVLDO3. This tab displays a simplified diagram of the selected configuration as a visual crosscheck.
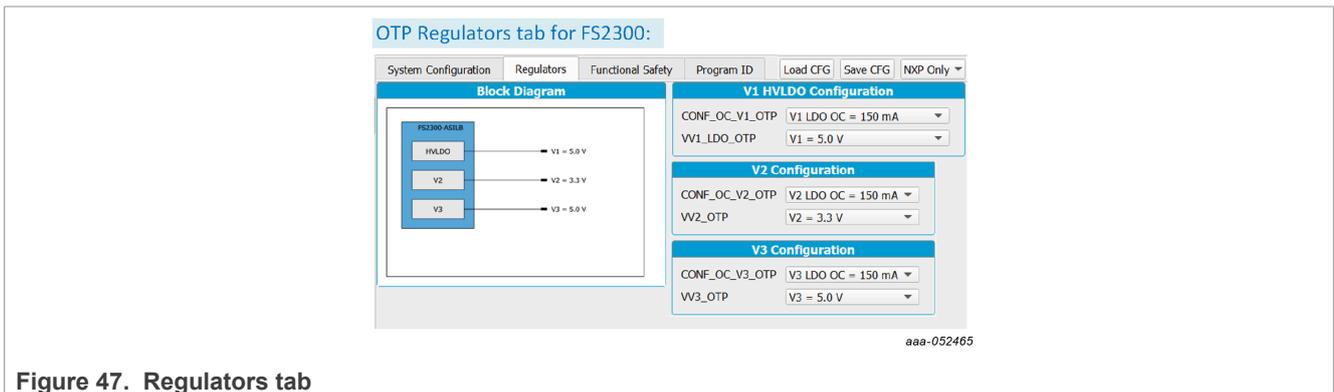
Figure 47. Regulators tab

6.5.2.1.3 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 6.5.2.2.

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

34 / 73
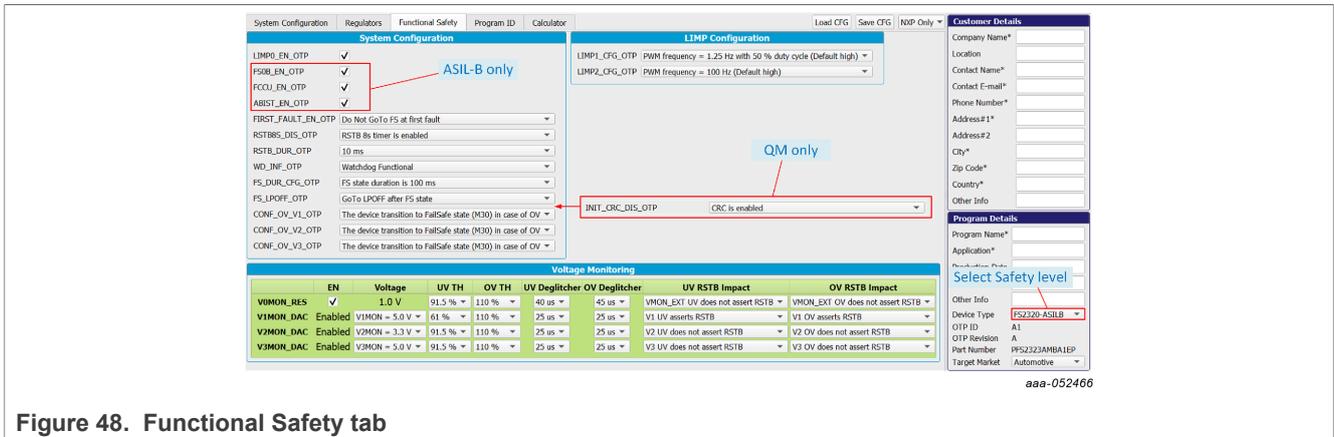
Figure 48. Functional Safety tab

6.5.2.1.4 Program ID tab

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

Figure 49. Program ID tab

6.5.2.1.5 Calculator tab

The Calculator tab is available for FS2320 version only.

6.5.2.2 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.
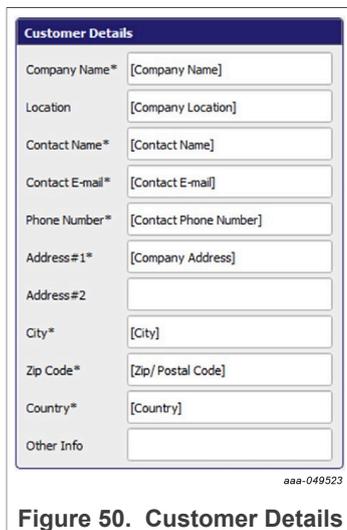
Figure 50. Customer Details

Program Details box: In addition to comments related to the application, this window allows the user to:

□ Select device type. See Section 6.5.2.1.3.

□ Display the OTP ID (set by NXP only)

□ Display the build part number (set by NXP only)

□ Select target market, either automotive or industrial
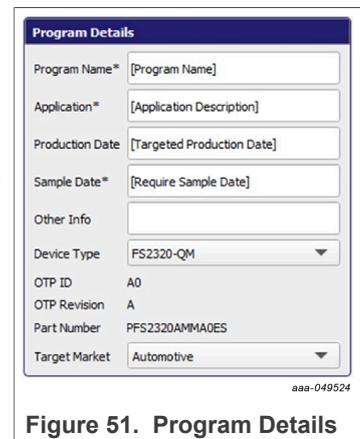
Figure 51. Program Details

6.5.3 PROG

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

Note: Use this tab cautiously, as a device can be programmed just twice.

The programming tool is available only in Test mode.
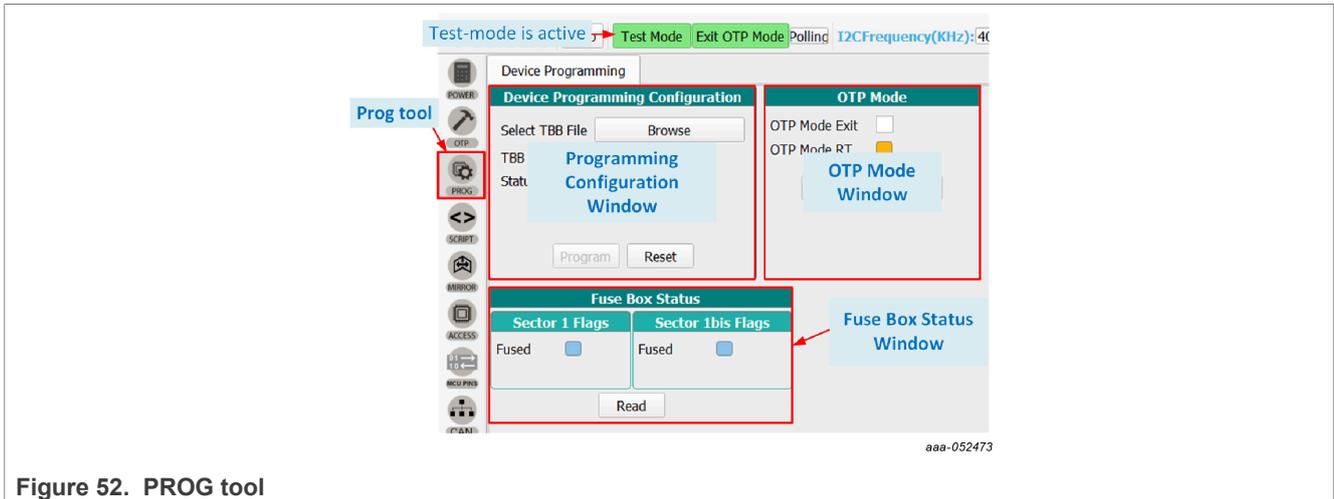
Figure 52. PROG tool

6.5.3.1 Device Programming Configuration window

From this window, the user can program an OTP operation from a saved TBB file.

To program the part by fusing OTP, see Section 7.8.

6.5.3.2 OTP mode window

OTP mode window allows the user to:

- Exit OTP mode by sending an I^2C/SPI frame (by checking OTP mode exit box).
- Verify that the state machine is in OTP mode (Condition: OTP 8 V applied to DBG pin at startup).

6.5.3.3 Fuse Box Status window

The Fuse Box Status window shows the OTP fuse status. In the indicator boxes, blue indicates the section has not been programmed yet, orange indicates the section has been fused already.

As a reference, an empty part has the same Fuse Box status as shown in Figure 52.

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

6.5.4 SCRIPT

The Script editor allows the user to create 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**
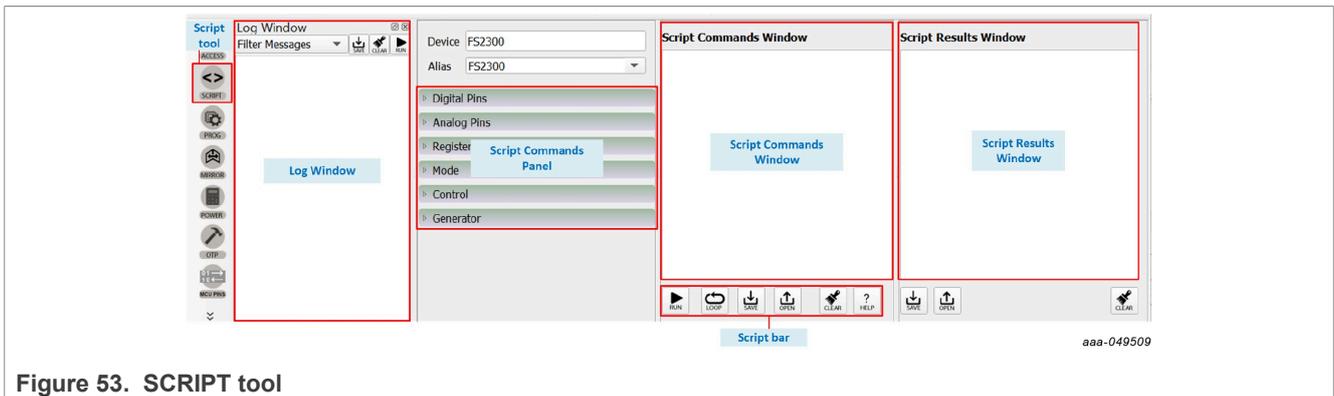
Figure 53. SCRIPT tool

6.5.4.1 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.

The Log window shows the CRC for each sent or received frame (valid for I²C and SPI).

Note: The Log window content can be saved to a file, then opened as a Script to run it. See Section 7.9.

6.5.4.2 Script Commands panel

The Script Commands panel allows the user to enter commands into the Script Command window by clicking 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.

- **Digital Pins:** To enter a script command to read or write the value of the selected digital pin.
- **Analog Pins:** To enter a script command to read the value of the selected analog pin.
- **Registers:** To enter a script command to read or write functional and safety registers.
- **Mode:** To enter a script command setting the desired mode.
- **Control:** To enter a script command to pause script execution (execution halts when the Pause command is encountered and a pop-up window appears, prompting the user 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:** To clear the content of the Script Command window and enters a pre-prepared script sequence.

An INIT script example has been integrated to the Generator tab. The example provides a generic step-by-step procedure to initialize the device with a default configuration and transition to Normal applicative mode.
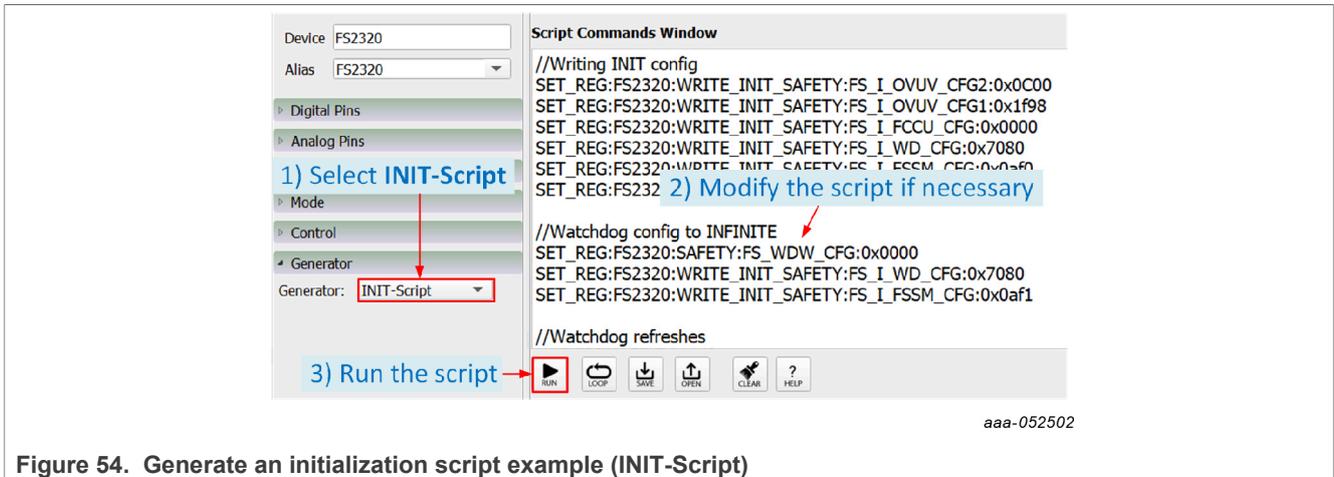
Figure 54. Generate an initialization script example (INIT-Script)

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

Figure 55. Script command panel help

All menu items work in a similar way. The example below shows a typical process using the Registers menu tab.

How to use the script tool: Clicking the Register tab brings up the parameters panel shown in Figure 56. A Write operation to the CTRL1 register in the functional register group has been selected. The value 0x00 is selected as the value to be written to the register. Hitting Enter with the cursor in the value field enters the well-formed 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.
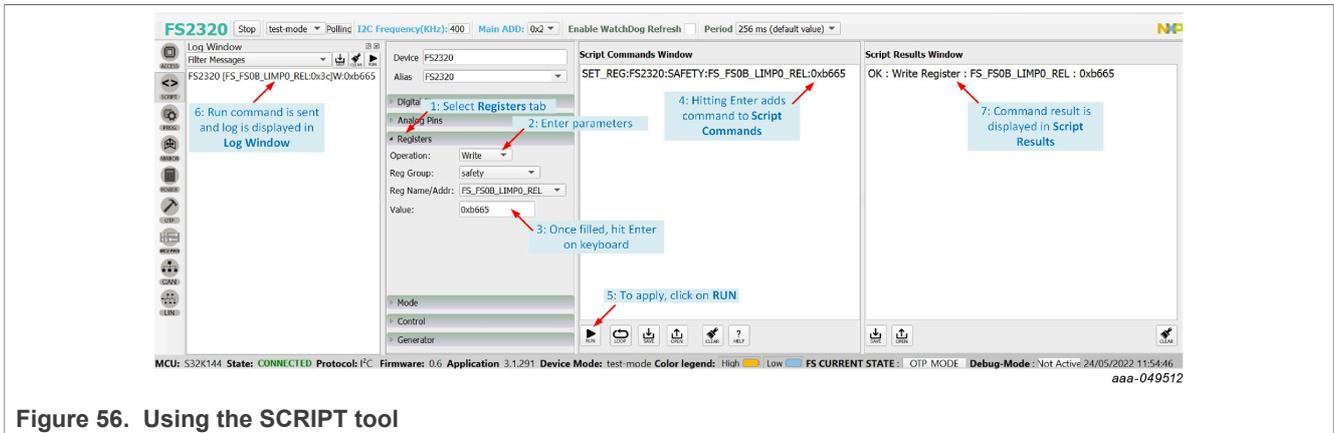
Figure 56. Using the SCRIPT tool

6.5.4.3 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 commands sequence, for example to run a routine. This specific use is detailed in Section 7.9.

Figure 57. Script Commands window

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

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

Note: The loaded file must have a .txt extension.

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

6.5.4.4 Script Results window

The Script Results window displays the results of an executed script. The menu bar at the bottom of the window contains the following three buttons:

Figure 58. Script Results window

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

6.5.5 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 are reset to the default OTP configuration (empty if OTP sectors are not burned). Mirror registers can be read and written in Test mode only. See Section 7.3.2.

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 purple. OTP tool tab headers are blue.

The MIRROR tool is available only in Test mode. Test mode loads require 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 OTP.

Figure 59. MIRROR tool

6.5.6 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 FS2300 device during the evaluation session.

The ACCESS tool provides access all I^{2}C/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.

Figure 60. ACCESS tool

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

40 / 73
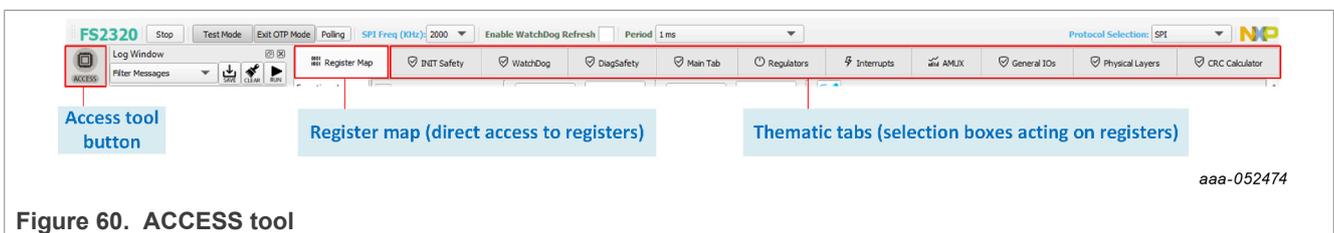
The tab content contains 11 tabs:

- **RegisterMap**
- **INIT Safety**
- **WatchDog**
- **DiagSafety**
- **Main Tab**
- **Regulators**
- **Interrupts**
- **AMUX**
- **General I/Os**
- **Physical Layers**
- **CRC Calculator**

6.5.6.1 Register map

The Register Map tab allows the user to read or write the FS2300 SPI/I^{2}C registers, bit per bit.

Note: *SPI/I^{2}C registers are responsible for FS2300 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 subtabs:

- **Functional**: Access to SPI/I^{2}C functional registers for system configuration.
- **Safety**: Access to SPI/I^{2}C registers relative to safety functions.
- **Write_INIT_Safety**: Access to SPI/I^{2}C registers relative to safety behavior configuration.
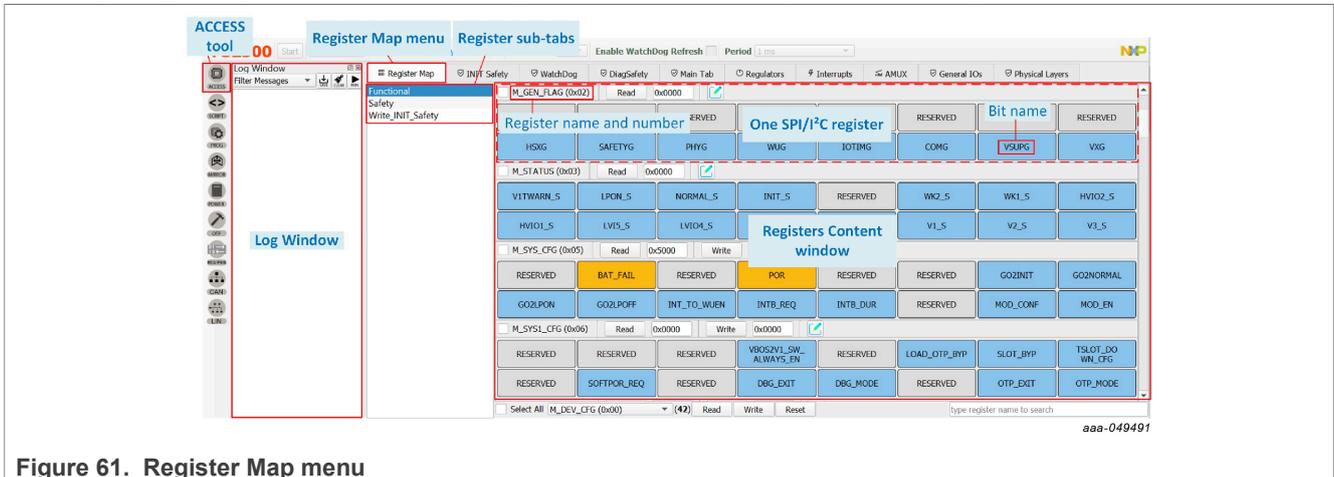
Figure 61. Register Map menu

6.5.6.1.1 Modifying one bit content by clicking on the bit name

In the Registers Content window, the user can access the FS2300 SPI/I^{2}C 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, etc.) appear with both a Read and a Write button.

The user can click a bit name to change its value, when the bit is writable.

aaa-049492
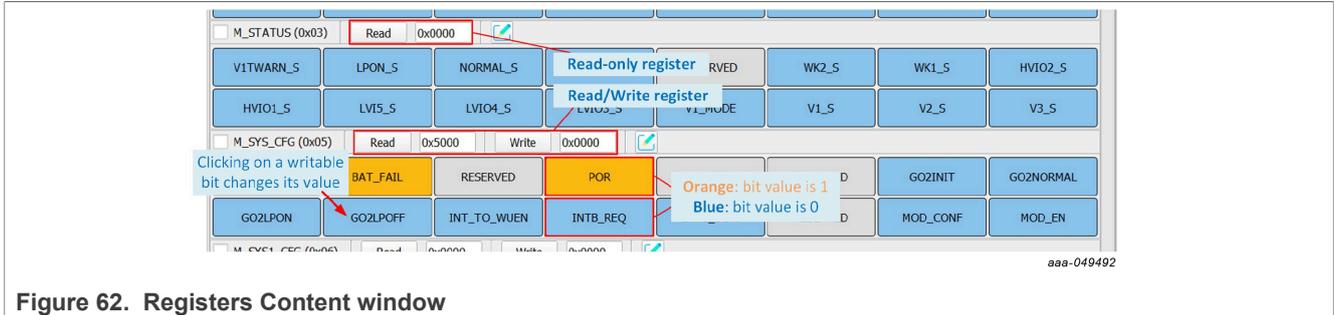
Figure 62. Registers Content window

6.5.6.1.2 Accessing one register content using Read and Write buttons

In the Registers Content window, the user can read and/or write the FS2300 SPI/I^{2}C 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 a hexadecimal value next to the Read button.

To **write** the content of one register:

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

aaa-049493
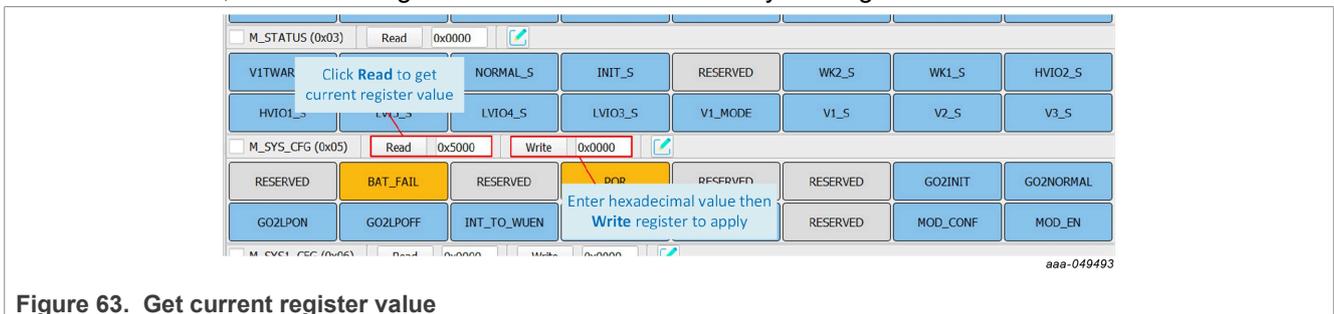
Figure 63. Get current register value

6.5.6.1.3 Accessing one register content using Bit-Map Dialog

In the Registers Content window, the user can read and/or write the FS2300 SPI/I^{2}C 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.

UM11879

User manual

All information provided in this document is subject to legal disclaimers.

Rev. 2 — 25 September 2023

© 2023 NXP B.V. All rights reserved.

42 / 73
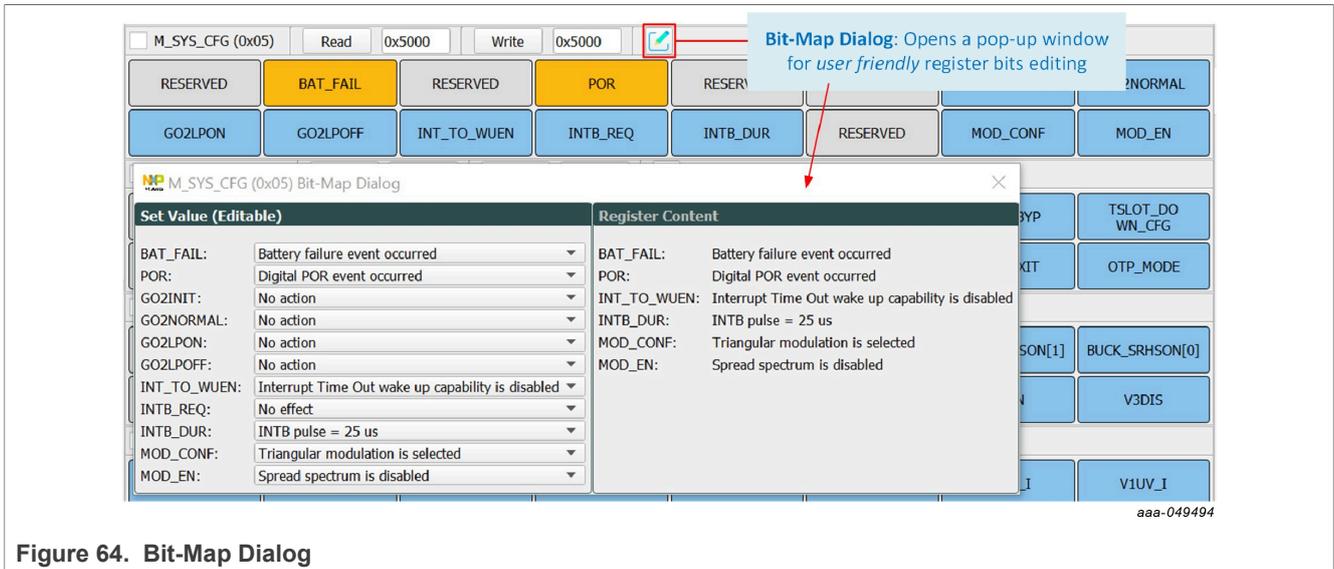
Figure 64. Bit-Map Dialog

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

In the Registers Content window, the user can read, write, and/or reset multiple FS2300 SPI/I^{2}C registers at a time using the lower Read/Write/Reset bar.

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

Navigating in the Registers content in the Register subtab can be facilitated using the selection box in the bar.
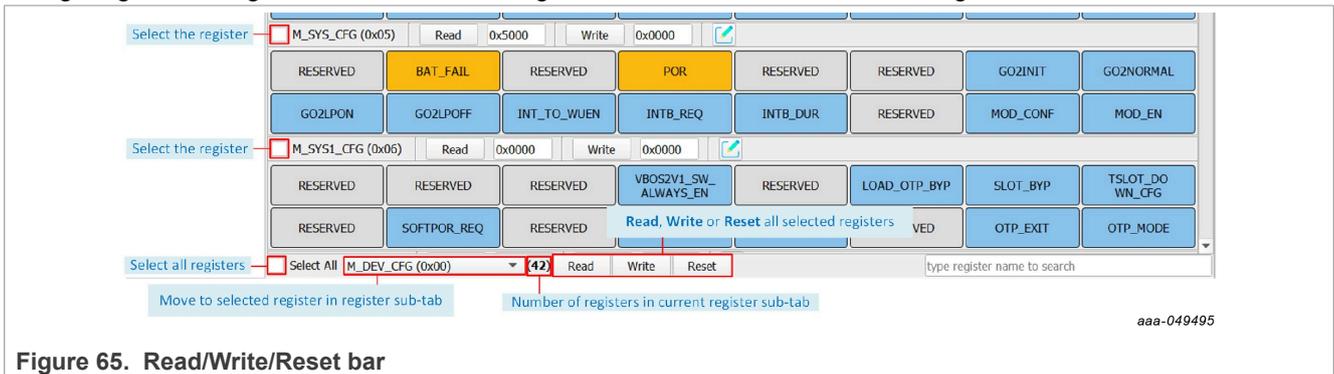
Figure 65. Read/Write/Reset bar

UM11879

User manual

All information provided in this document is subject to legal disclaimers.

Rev. 2 — 25 September 2023

© 2023 NXP B.V. All rights reserved.

43 / 73

6.5.6.2 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.
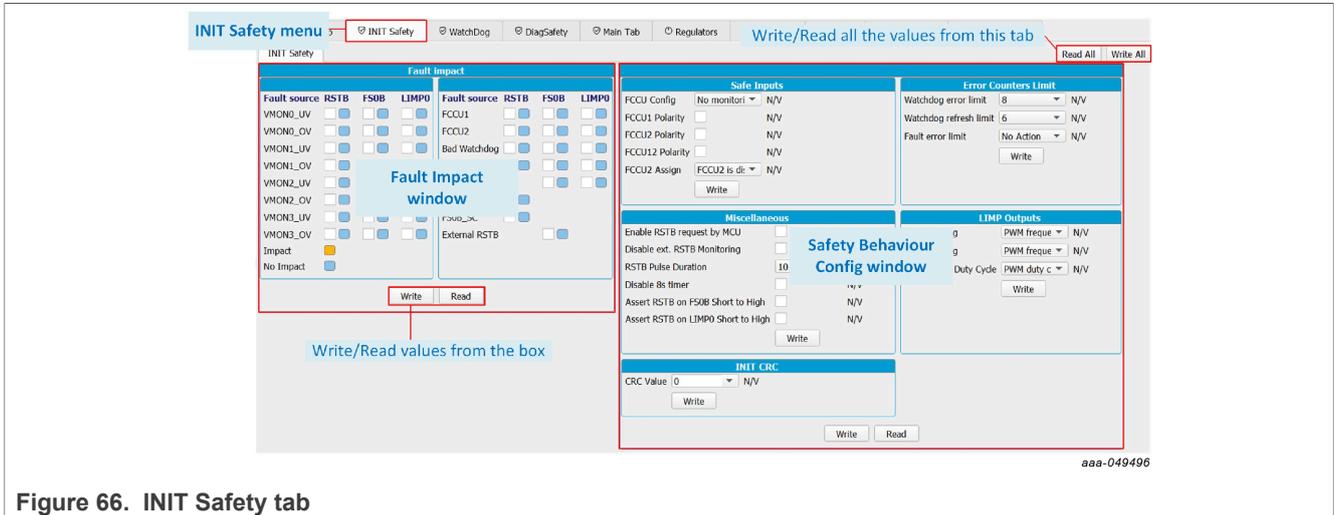
Figure 66. INIT Safety tab

- **Fault Impact:** To choose safety pin assertion response to each fault source.
- **Safety Behavior Config:** To initialize and configure safety-related functions, such as FCCUx assignment, Watchdog error counter, LIMPx mode, or CRC check value.
- **INIT CRC Calculator:** The INIT CRC computation toolbox allows the user to compute INIT CRC from current INIT Safety registers content and write INIT CRC into INIT CRC register.
  Note: INIT CRC computation is also available when the EVB is not connected.
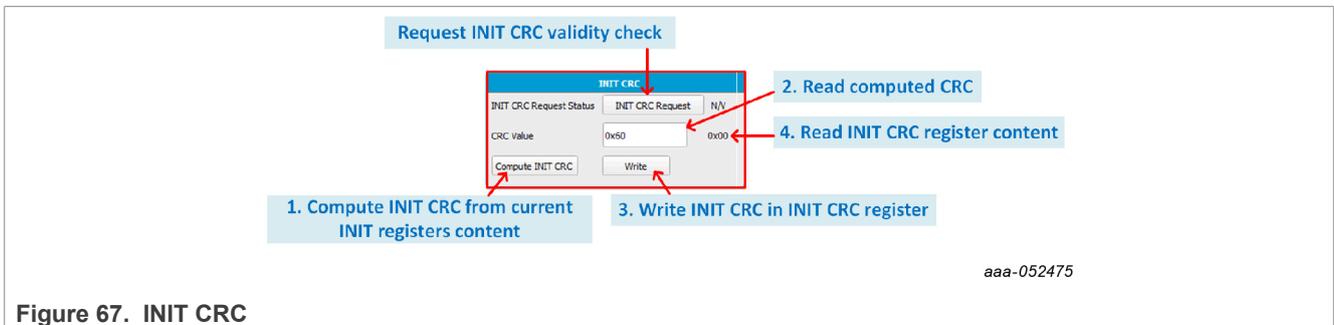
Figure 67. INIT CRC

6.5.6.3 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:** To manage Watchdog operation by sending SPI/I^{2}C commands and set Watchdog functional parameters such as window timing.
- **Watchdog Counters Config:** To read the previously set Watchdog error and refresh counters' limit values and fault error limit action. See Section 6.5.6.2 for parameters setting.
- **Watchdog Error and Refresh Counts diagrams:** Displays the diagrams of Watchdog error count and Watchdog refresh count relative to the previously set Watchdog error and refresh counters' limit values.

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

44 / 73
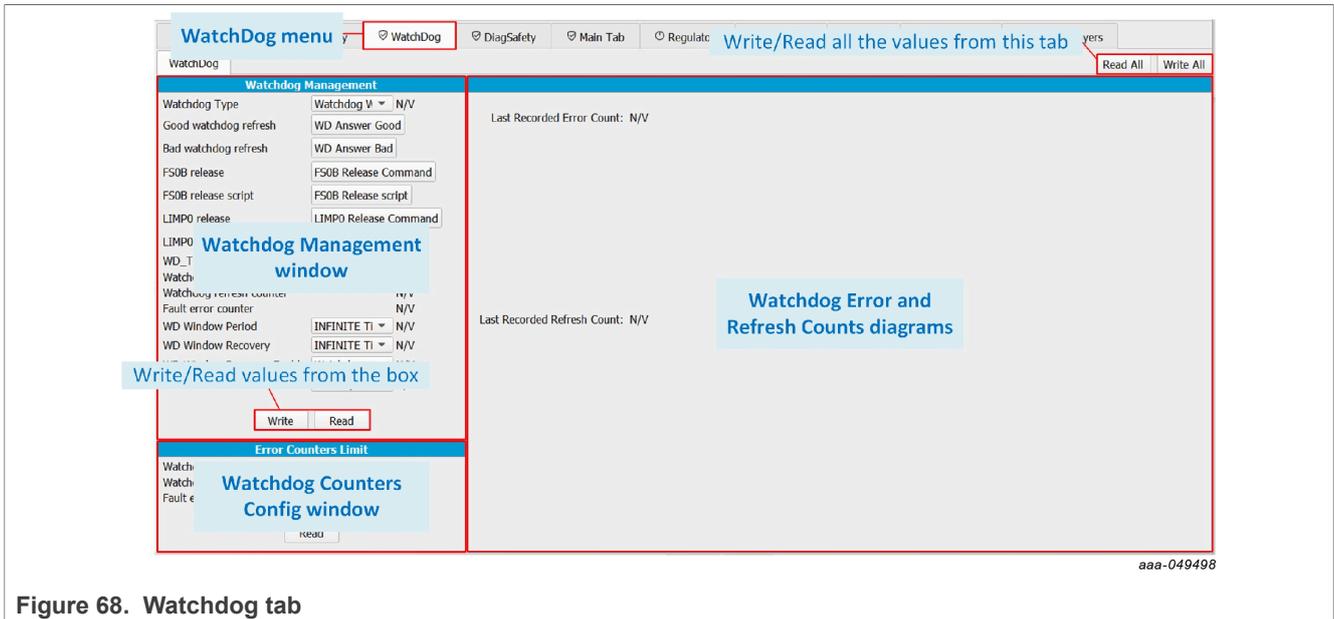
Figure 68. Watchdog tab

6.5.6.4 DiagSafety

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

• **Safety I/O Diagnosis:** To request and observe the diagnosis on safety I/O pins.
• **ABIST Diagnosis:** To operate and observe ABIST on the device.
• **FCCU/WD Diagnosis:** To carry out the diagnosis on FCCUx and WD Refresh by exploiting interrupt flags and real-time status bits.
• **UV/OV Diagnosis:** 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.
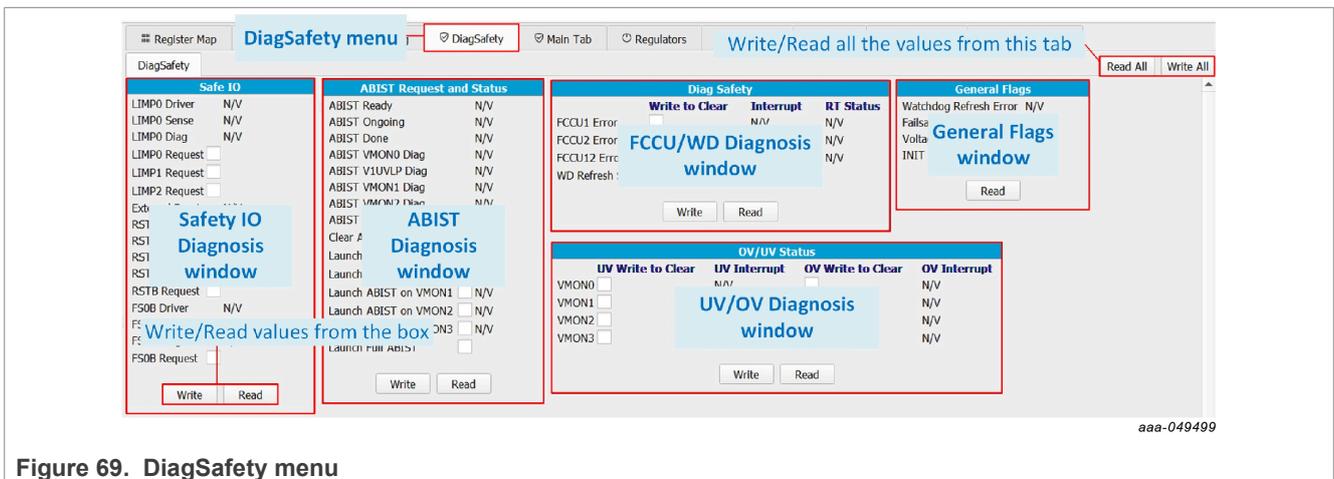
Figure 69. DiagSafety menu

6.5.6.5 Main Tab

The Main Tab allows the user to monitor Events occurrence, to operate the Device States, and to configure FS2300 functions such as Timers, PWMs, or Long Duration Timer (LDT).

- **General Flags:** To monitor occurrence of events on FS2300 functions.
- **Device State:** To monitor the device's state and request mode change.
- **Timers Config:** To enable timers and configure timer parameters.
- **PWMs Config:** To enable PWMs and configure PWMs parameters.
- **LDT Config:** To enable, configure, and monitor the LDT.
- **Miscellaneous Management:** To operate FS2300 external events and FS2300 VBOS, time slots attribution, and interrupt settings.
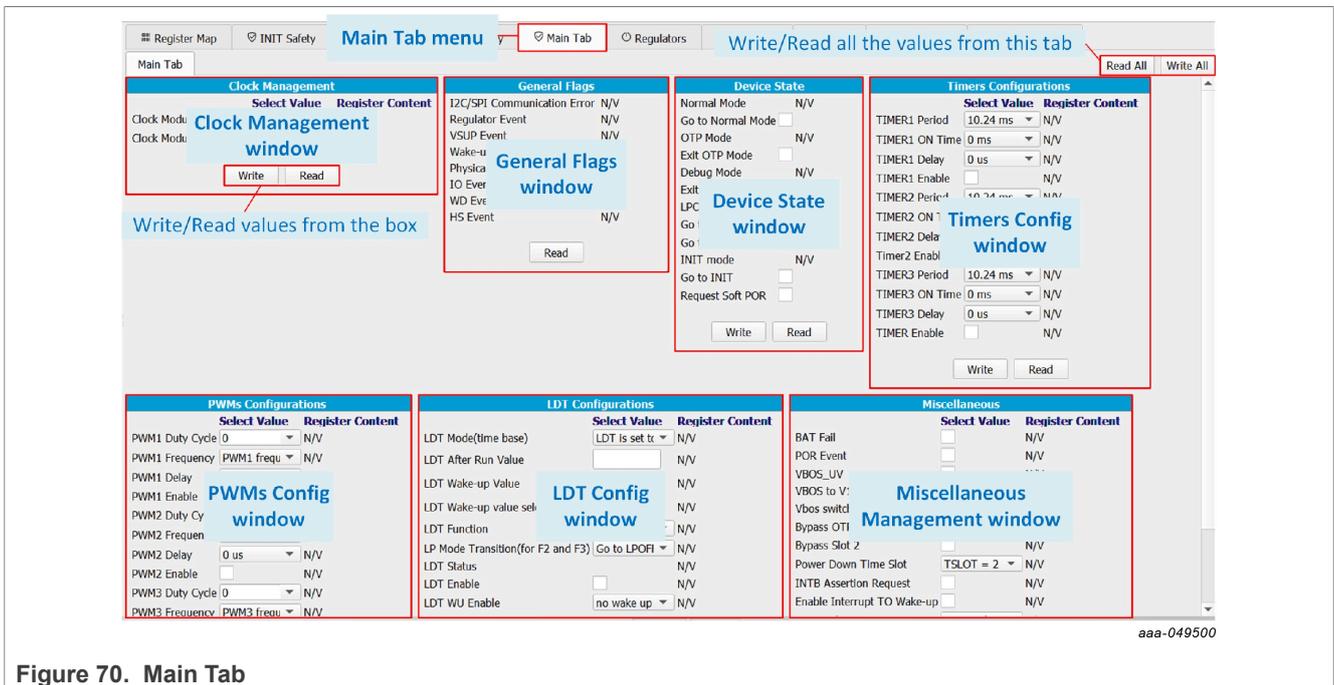
Figure 70. Main Tab

6.5.6.6 Regulators

The Regulators tab allows the user to control and read status related to HVLDO1 (V1), HVLDO2 (V2), and HVLDO3 (V3).

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

- **Regulators Control:** To control regulators' enablement in Normal and LPON modes.
- **Regulators Real-Time Status:** To monitor regulators in real time.
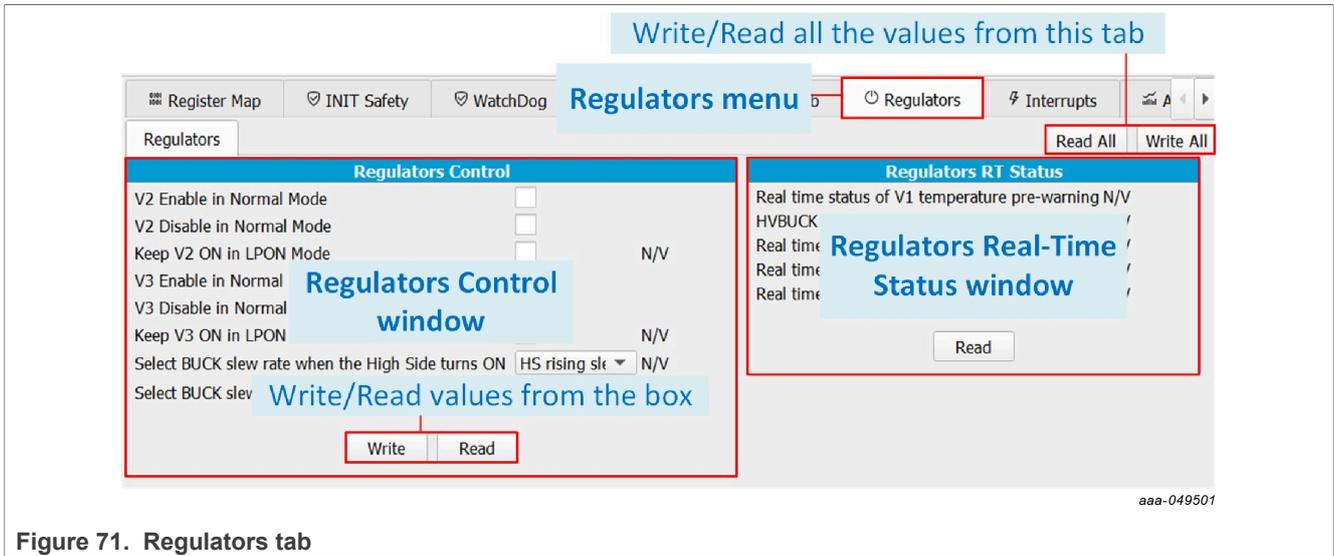
Figure 71. Regulators tab

6.5.6.7 Interrupts

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

- **Main Interrupt Configuration:** To monitor the events related to FS2300 functional features such as Regulators Temperature and Current, SPI/I^{2}C Communication, Physical Layers, Timers, High-Side drivers, I/Os, and Wake-Up events.
- **FailSafe Interrupt Configuration:** To monitor safety-related events such as under-, overvoltage, or open load on regulators, and events on safety pins or functions.
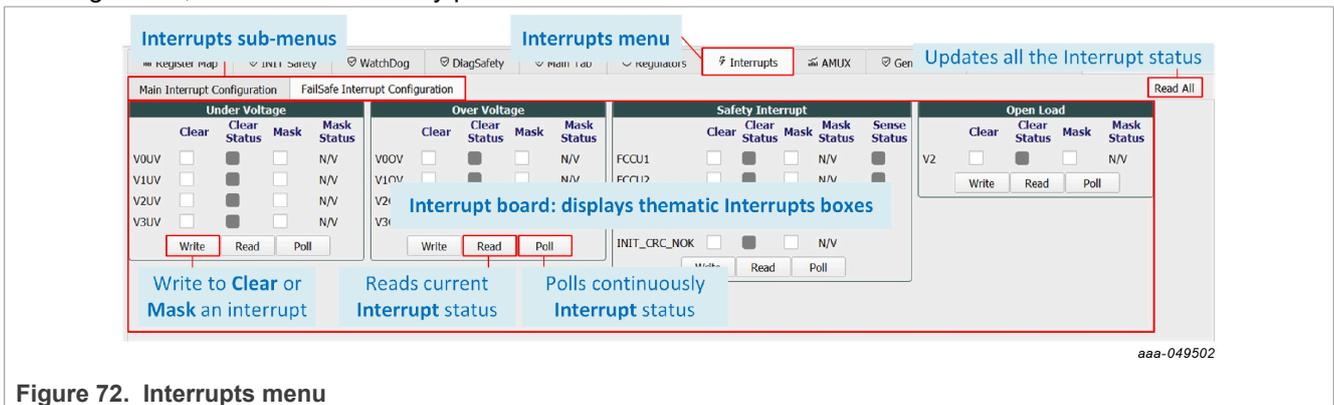
Figure 72. Interrupts menu

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 Write to apply.
- **Masked**: Select each check box from the Mask column then Write to apply.
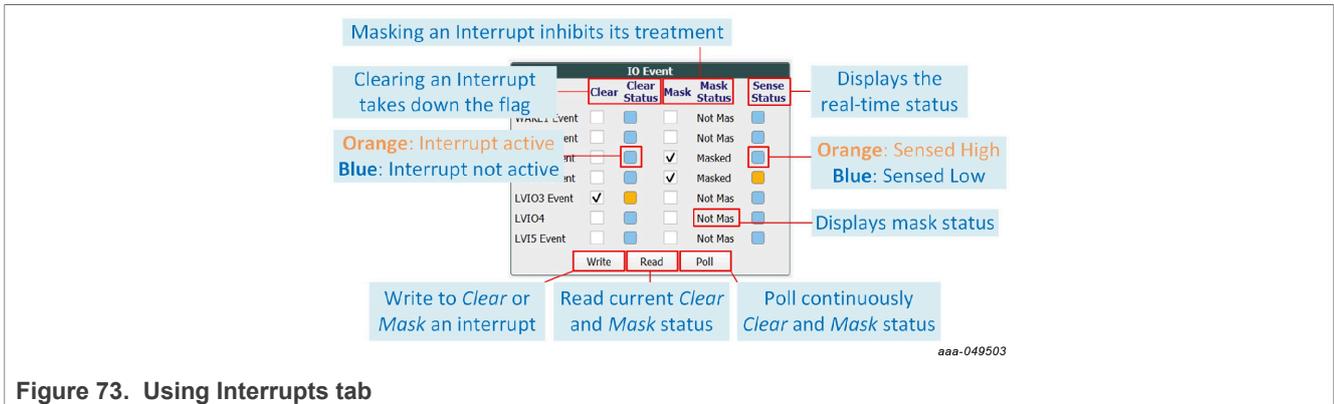
Figure 73. Using Interrupts tab

6.5.6.8 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**. The MCU supply voltage level must be selected when enabling the tool.

- **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.
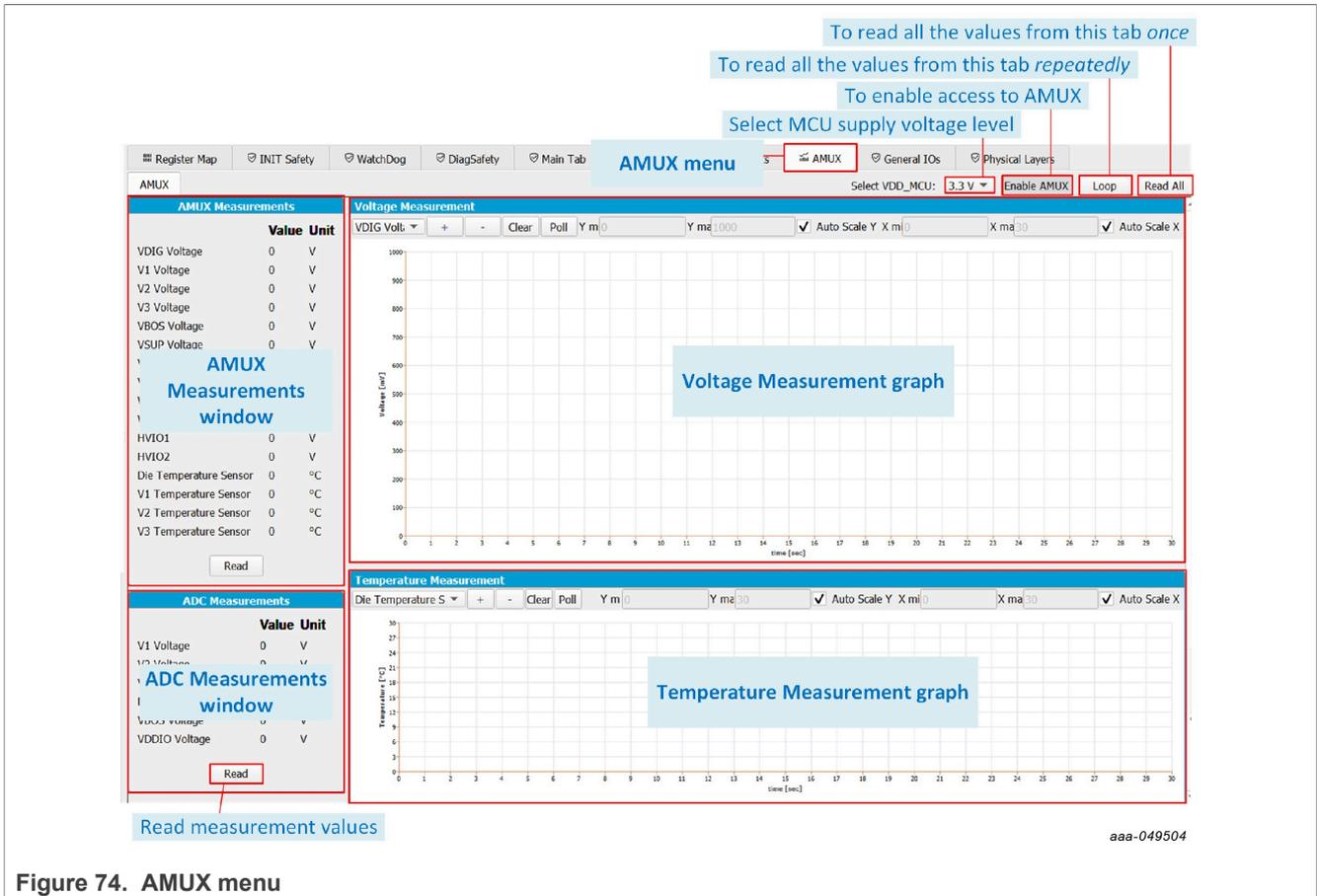
Figure 74. AMUX menu

Measurement graphs can be edited using their editing bar, as shown in [Figure 75](#):
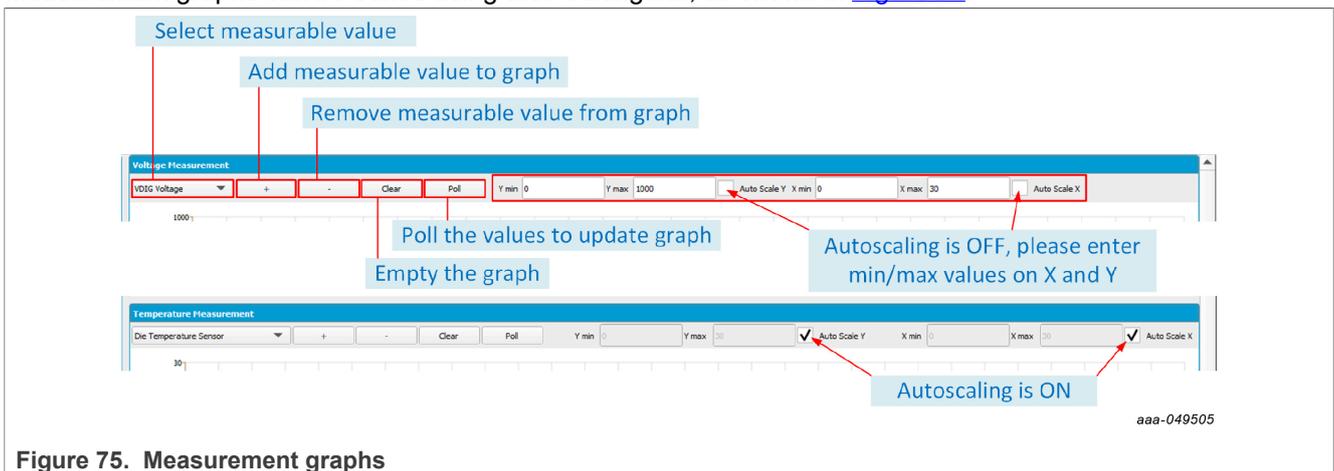
Figure 75. Measurement graphs

6.5.6.9 General I/Os

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

- **I/O Control:** To control I/O output levels when configured as outputs.
- **I/O Wake-up/Interrupt Enable:** To enable the Wake-Up and/or Interrupt feature on I/O.

- **I/O Wake-up Config:** To set I/O Wake-Up parameters when the Wake-Up feature is enabled on I/O.
- **Cyclic Sense:** To enable Cyclic Sense feature, to set Cyclic Sense parameters on I/Os and High-Side pins, and to operate the Cyclic Sensing by monitoring status.
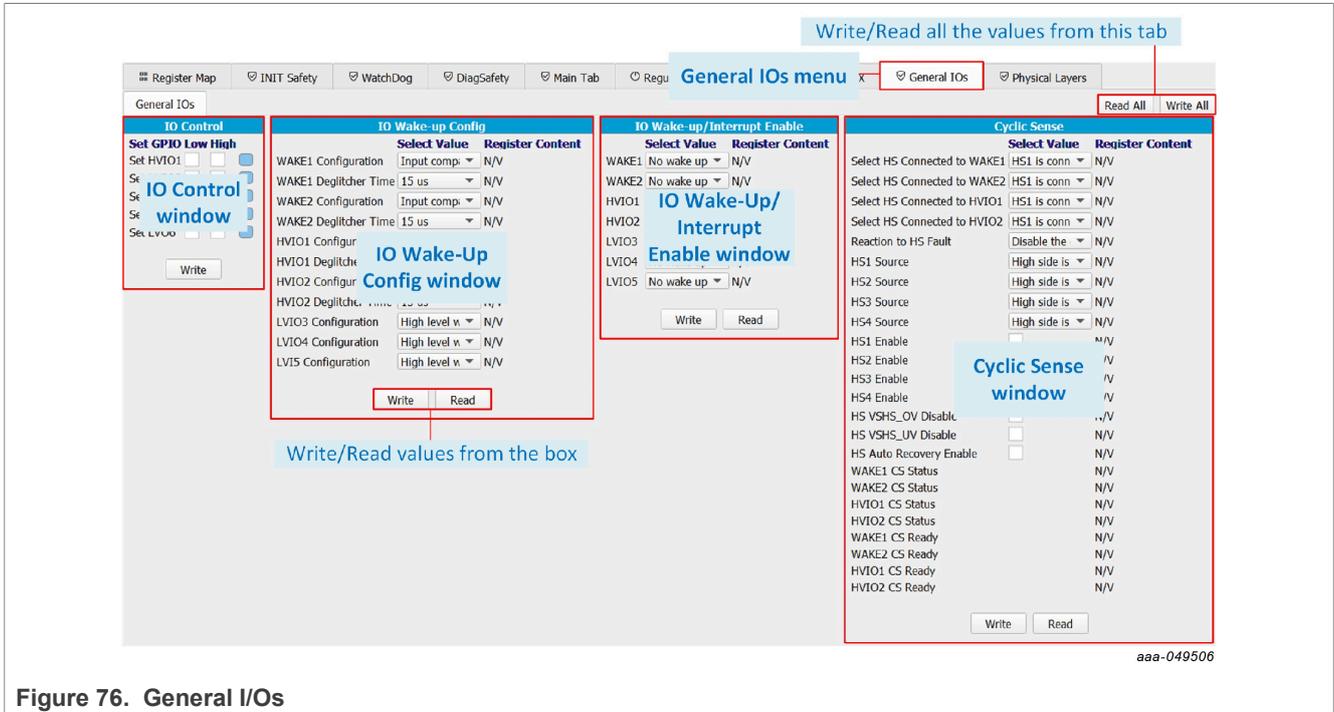
Figure 76. General I/Os

6.5.6.10 Physical Layers menu

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

- **CAN Config/Status:** To configure the CAN transceiver and the monitor status. The CAN setting on the MCU side and the CAN frames sending is done using CAN tool.
- **LIN Config/Status:** To configure the LIN transceiver and the monitor status. The LIN setting on the MCU side and the LIN frames sending is done using LIN tool.
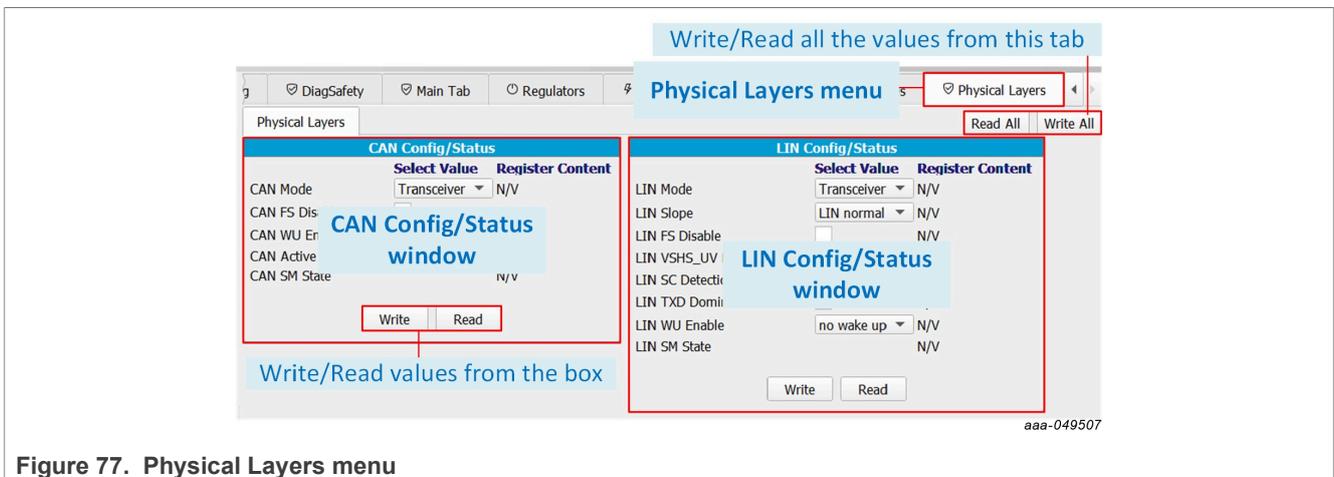
Figure 77. Physical Layers menu

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

50 / 73

6.5.6.11 CRC Calculator

The CRC Calculator is used to compute I^{2}C/SPI protocol CRC from an address and a data field value in hexadecimal format.

The user enters the address field content and the data field content of the I^{2}C/SPI frame to be sent, and reads CRC result.
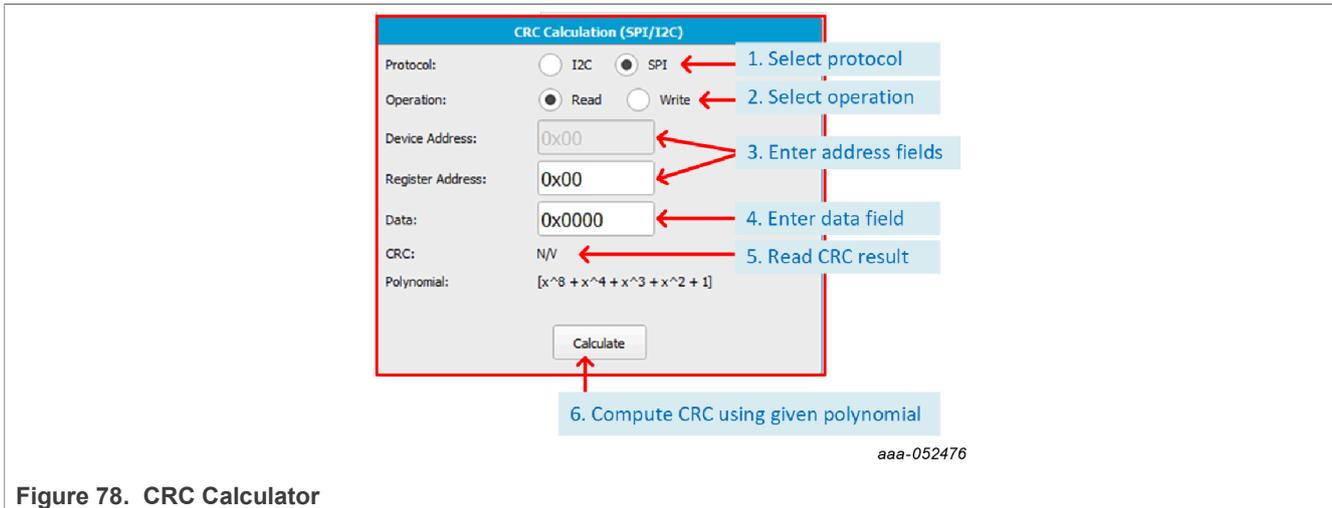
Figure 78. CRC Calculator

6.5.7 MCU PINS

The MCU PINS tool provides a means of reading and setting the S32K144 input and output pins for signals FS0B, INTB, RSTB, FCCUx, LVIOx:

- FS0B, INTB, RSTB are MCU inputs.
- FCCUx are MCU outputs.
- LVIOx can be configured as input or output for the MCU depending on the configuration on the FS2300 device.

Note: *The configuration of LVIOx pins must be done by OTP. Depending on the chosen functionality for LVIOx pins, the hardware must be configured as explained in [Section 4.4.11](#).*

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.
- **I/Os Function Selection window:** To use the MCU PINS tool, it is mandatory to select the FS23 I/Os settings as configured in OTP/Mirror tabs in order to configure MCU pins accordingly. Without this information, MCU pins cannot be configured properly.
- **MCU Input Pins Reading window:** To read or poll the listed MCU pins during a selected time duration.
- **MCU Output Pins Setting window:** To set the FCCUx and LVIOx (when configured as output) pins high or low on MCU side.
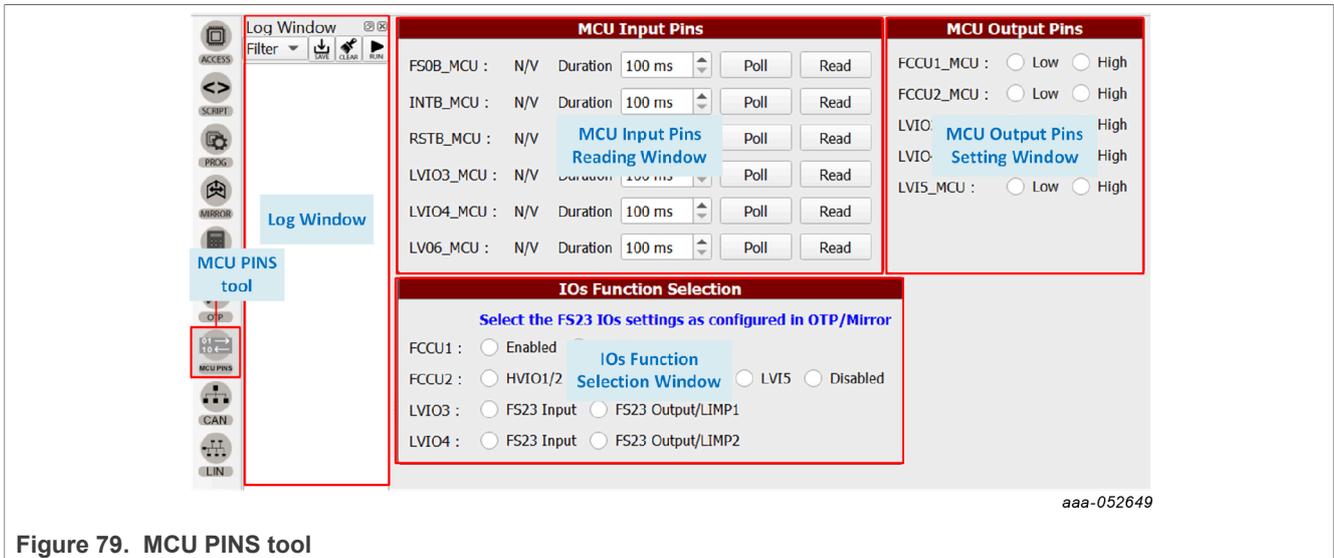
Figure 79. MCU PINS tool

6.5.8 CAN

The CAN tool provides a means to use the CAN transceiver on FS2300. 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.
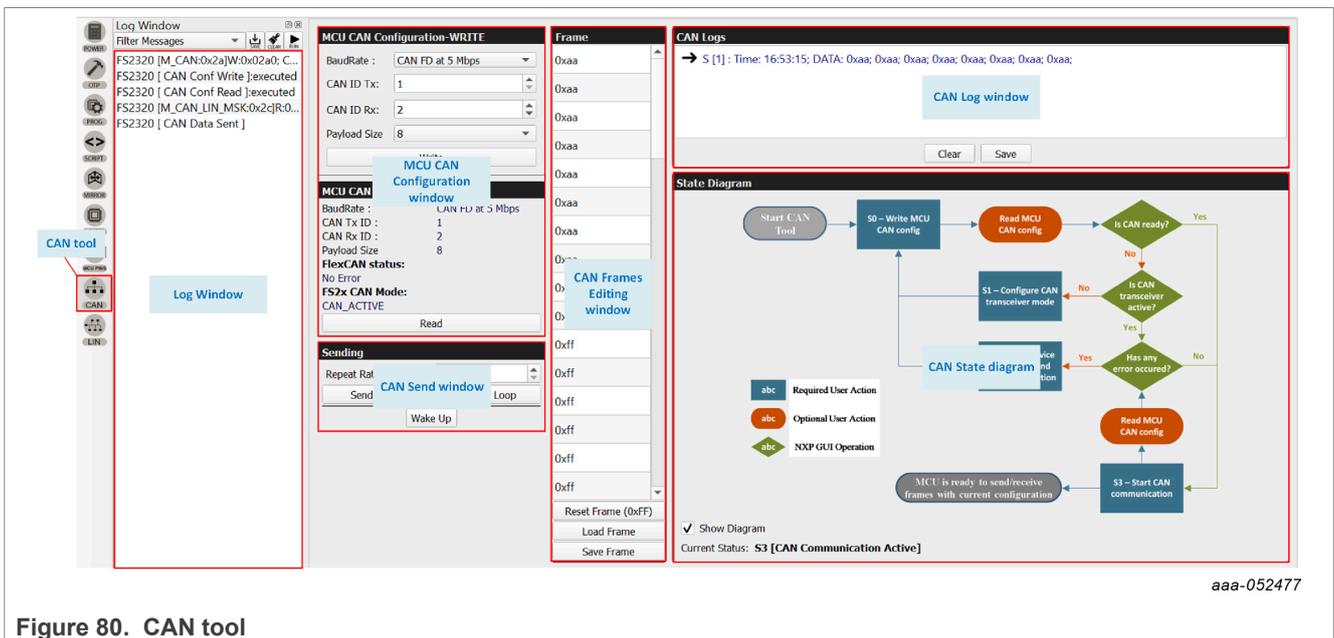
Figure 80. CAN tool

The CAN tool panel consists of four sections:

- **MCU CAN Configuration window:** The WRITE section allows the user to configure the CAN on the MCU S32K144 side by choosing the baud rate, CAN Tx ID (the Device transmits a frame to this ID), and Rx ID (Device's ID), and payload size (data length). The READ section allows the user to read back the MCU FlexCAN module configuration, and check the FlexCAN module status on the MCU side, and the CAN transceiver status on the FS23 side.
- **CAN Send window:** To send wake-up frames and to send CAN frames sporadically or periodically with a selectable repeat rate from 1 ms to 100 s.

- **CAN Frames Editing window:** 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.

Note: *A diagram of the CAN state machine implementation in the MCU can be displayed, to understand the behavior of the CAN on the MCU side.*

6.5.9 LIN

The LIN tool provides a means to use the LIN transceiver on the FS2300. The LIN tool allows the user to configure the LIN bus on the MCU side and send edited frames on the bus sporadically or periodically with a selectable repeat rate. LIN logs are displayed.

Figure 81. LIN tool

The LIN tool panel consists of four sections:

- **MCU LIN Configuration window:** The WRITE section allows the user to configure the CAN on the MCU S32K144 side by choosing the baud rate, LIN ID (ID 0 is defined as Master's ID by default), Master/Slave bus control position, and payload size (data length). The READ section allows the user to read back the MCU LIN module configuration, and check LIN module status on the MCU side and the LIN transceiver status on the FS23 side.
- **LIN Send window:** To send wake-up frames and to send LIN frames sporadically or periodically with a selectable repeat rate from 44 ms to 100 s.
- **LIN Frames Editing window:** To edit the frames to be sent on the LIN bus (up to 8 bytes) with hexadecimal values. LIN frames can be imported and saved as CSV files.
- **LIN Log window:** Displays exchanged frames.

Note: *A diagram of the LIN state machine implementation in the MCU can be displayed, to understand the behavior of the LIN on the MCU side.*

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

53 / 73

7 Setting up and running the KITFS23LDOEVM

This section gives guidance on how to set up and run the KITFS23LDOEVM evaluation board and GUI.

The device has a high level of flexibility thanks to the parameter configuration available by using the OTP registers. The user should learn about OTP and Mirror registers before operating with the device.

7.1 Setting up the KITFS23LDOEVM

The procedure for setting up the KITFS23LDOEVM 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 I^{2}C) and the GUI. Section 4.4 shows the default jumpers and switches configuration for FS2300 part.
2. Connect the power supply to J6 (Phoenix connector - 3.81 mm). The power supply should be set to a nominal value of 14.0 V.
   OR Connect the power supply to J2 (Jack connector) via a 14.0 V battery.
   Note: When connecting loads to the boards, check that the J37 jumper is in position J37-1-2 to avoid drawing current from the USB line. Place jumper in position J37-1-2 to supply 5.0 V from onboard LDO connected to VIN. Place jumper in position J37-2-3 to supply 5 V from the USB line.
3. Make sure 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 also provides voltages and references to some onboard circuits.
4. Place SW1 to position SW1-3 if the power supply is connected to J6.

OR Place SW1 to position SW1-1 if the power supply is connected to J2.

7.2 Connecting the KITFS23LDOEVM to the GUI

The procedure for connecting the KITFS23LDOEVM 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 FS2300 SBC.

USB cable plugged in, communication not established yet

State: DISCONNECTED
aaa-049530

USB cable not plugged in

State: NOT DETECTED
aaa-049531

3. To establish communication between the S32K144 MCU and the FS2300 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

State: CONNECTED
aaa-049528

Note: If the Start button does not light on and stays grey, the FTDI driver might be missing. To confirm this, access to the Windows Device Manager and look for FT230X Basic UART device.
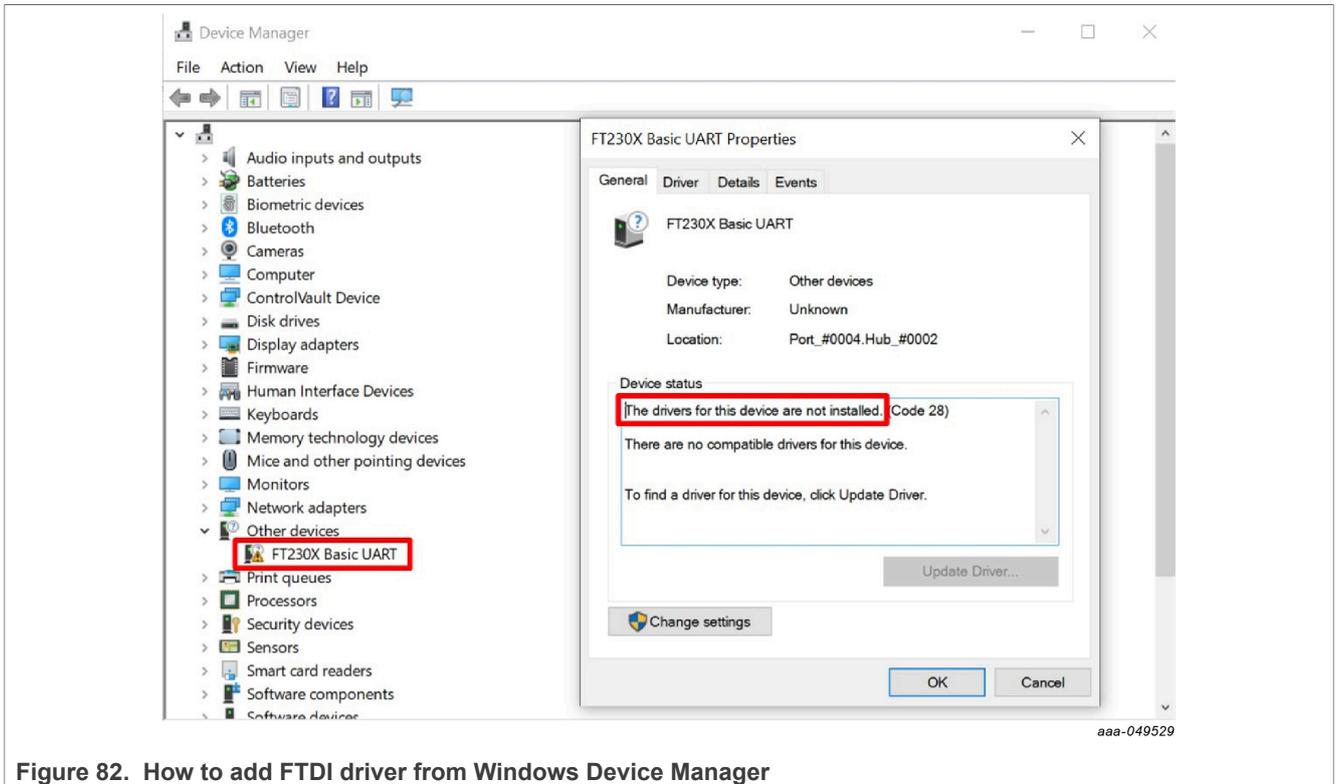
Figure 82. How to add FTDI driver from Windows Device Manager

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 at this address.*

7.3 Operation modes

The KITFS23LDOEVM provides three distinct operation modes with direct impact on device functionalities. Understanding these modes helps the user interact with the GUI properly.

Note: *When using the KITFS23LDOEVM for the first time, it is recommended the user start the device in Debug mode so the Watchdog function is not active. Indeed, the Watchdog must be configured in Debug mode before starting the device in Normal mode for the first time. Otherwise a Watchdog error is detected. To start the device in Debug mode, the hardware should be configured accordingly. See Section 7.3.1.2. Once the Watchdog is configured, Debug mode can be exited from the GUI.*

The voltage level on FS2300 DEBUG pin is one condition for entering a given operation mode. Table 17 gives the hardware configuration conditions to enter Normal, Debug or OTP/Test mode. Debug and OTP configuration (J30/SW10/SW12) gives more information about onboard hardware enablement of Debug mode and Test mode context.

Table 17. Mode entry hardware conditions

Board configuration	Configuration		
	Normal mode	Debug mode entry	OTP/Test mode entry
J10 (DBG)	J10 OFF	J10 ON	J10 ON
SW12 (OTP)	SW12 OFF	SW12 OFF	SW12 ON
SW10 (DCDC_EN)	SW10 OFF	SW10 OFF	SW10 ON

Figure 83 gives an overview of the device modes and actions to do in the GUI or on the EVB to enter or exit modes. Test mode stays active for as long as nothing is written in TM_ENTRY register. Therefore, it is possible to be in Test mode and Normal mode at the same time.
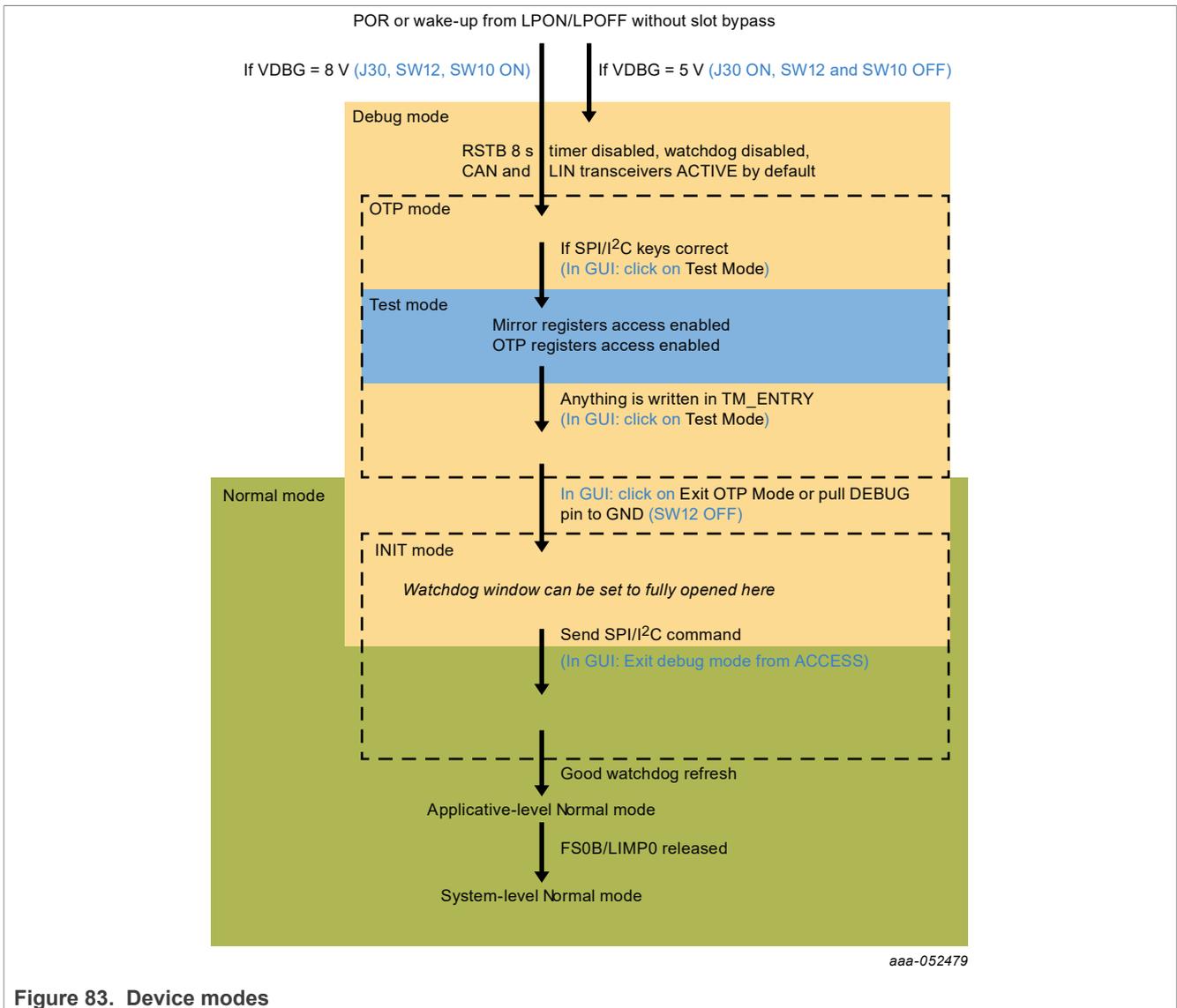
Figure 83. Device modes

7.3.1 Debug mode

7.3.1.1 Debug mode definition

The device starts in Debug mode when the hardware is configured accordingly. This mode requires the DBG pin to be connected to VBOS through a diode to enter Debug mode automatically at startup. On the board, jumper J30 ON and switches SW10 and SW12 OFF connect the DBG pin to VBOS through a diode.

Debug mode works in parallel with Normal mode or Test mode. When Debug mode is active, the device runs with limited functionalities as some functions are disabled.

Note: *In Debug mode, RSTB 8 seconds timer is disabled, Watchdog is configured with infinite timeout, and the window is fully opened (equivalent to disabled), the CAN and LIN transceivers are set in Active mode by default.*

UM11879

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

User manual

Rev. 2 — 25 September 2023

56 / 73

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

7.3.1.2 Debug mode activation

There are two ways to activate Debug mode:

7.3.1.2.1 With the GUI using mode selection in Connection toolbar (recommended)

1. Set the default jumper configuration. See Section 7.1.
2. Verify that switches SW10 and SW12 are OFF (disables OTP 8 V generation).
3. Verify that jumper J30 is ON (enables Debug mode).
4. Plug the USB cable between the PC and the board. The blue LED D14 for OTP 8 V generation is OFF. The green LED D15 for Debug is ON.
5. Apply power supply VBAT.
6. Open the GUI and start the connection.
7. Check that the GUI has detected Debug mode in the USB and device status bar:
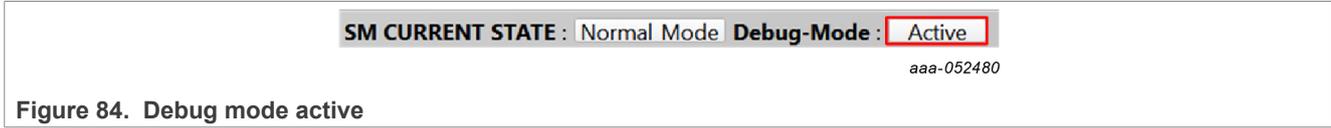
aaa-052480

Figure 84. Debug mode active

8. The equipment is now set to Debug mode.

7.3.1.2.2 Without the GUI (hardware only)

1. Set the default jumper configuration. See Section 7.1.
2. Verify that switches SW10 and SW12 are OFF (disables OTP 8 V generation).
3. Verify that jumper J30 is ON (enables Debug mode).
4. Apply power supply VBAT.
5. The device powers up and operates in Debug mode. The blue LED D14 for OTP 8 V generation is OFF. The green LED D15 for Debug is ON.

7.3.1.3 Debug mode deactivation

Once activated, Debug mode can be deactivated by software by sending an I^2C/SPI command via the GUI with the ACCESS tool:

- With the thematic tabs: in the Main Tab, in the Device State box, select checkbox Exit Debug mode, then click **Write**.
- With the Register Map: in the Functional subtab, set DBG_EXIT bit to 1 in M_SYS1_CFG register.

Note: Taking off Debug Selection jumper J30 will not automatically deactivate Debug mode, as DBG 5 V on FS2300 DEBUG pin is only a condition for Debug mode entry. Therefore, it is not a condition to remain in Debug mode.

7.3.2 Test mode

7.3.2.1 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: Mirror registers are an emulation of OTP registers. In case of a POR, the Mirror registers are reset to the default OTP configuration (empty if OTP not burned).

Test mode requires **DEBUG SELECT** jumper ON, **DEBUG** = +8 V (Debug mode is therefore enabled when Test mode is enabled) and valid Test mode keys sent by SPI/I^{2}C. 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 7.8.

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

7.3.2.2 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 7.1.
2. Set switches SW10 and SW12 ON (enables OTP 8 V generation).
3. Verify that jumper J30 is ON (enables Debug mode).
4. Plug the USB cable between the PC and the board. The blue LED D14 for OTP 8 V generation and the green LED D15 for Debug are 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:
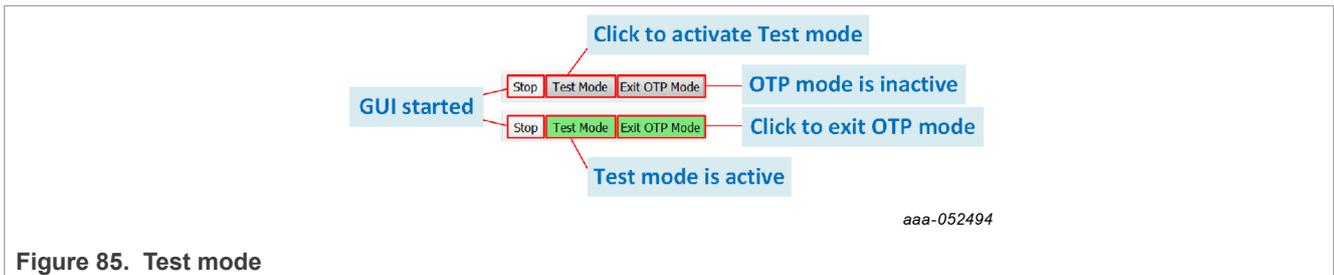
Figure 85. Test mode

7.3.2.3 Test mode deactivation

Once activated, Test mode can be deactivated using the GUI by clicking on the green Test mode button from the Connection toolbar. Once Test mode is exited, the device goes to OTP mode as long as OTP mode exit actions are not completed. When OTP mode is exited, then the device transitions to Normal and Debug mode. See Debug-mode deactivation to learn more about the connection between Debug and Normal 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 FS2300 DEBUG pin is only a condition for Test mode entry. Therefore, it is not a condition to remain in Test mode.*

7.3.2.4 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 6.5.4.
• With the MIRROR tool. See Section 6.5.5.

These methods are functional with an empty or burned part.

The procedure is explained in Section 7.7.

7.3.3 Normal mode

7.3.3.1 Normal mode definition

The Normal mode operation is used for final product test in automotive environment. Normal mode entry at power-on reset requires **DEBUG SELECT** jumper J30 = OFF and **DEBUG** = GND.

When Normal mode is active, the device operates with all the functionalities enabled by the loaded OTP configuration.

7.3.3.2 Normal mode activation

There are two ways to activate Normal mode:

7.3.3.2.1 With the GUI using mode selection in Connection toolbar (recommended)

1. Set the default jumper configuration. See Section 7.1.
2. Verify that switches SW10 and SW12 are OFF (disables OTP 8 V generation).
3. Open jumper J30 (disables Debug mode).
4. Plug the USB cable between the PC and the board. The blue LED D14 for OTP 8 V generation and the green LED D15 for Debug are OFF.
5. Apply power supply VBAT.
6. Open the GUI and start the connection.
7. Check that the GUI started in Normal mode from the USB and device status bar:
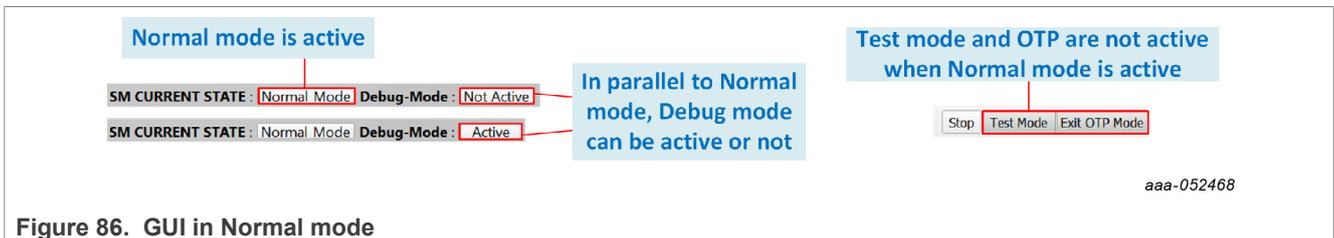
Figure 86. GUI in Normal mode

8. The equipment is now set to Normal mode.

7.3.3.2.2 Without the GUI (hardware only)

1. Set the default jumper configuration. See Section 7.1.
2. Set switches SW10 and SW12 OFF (disables OTP 8 V generation).
3. Open jumper J30 (disables Debug mode).
4. Apply power supply VBAT.
5. The device powers up and operates in Normal mode. The blue LED D14 for OTP 8 V generation and the green LED D15 for Debug are OFF.

7.4 Generate a TBB script

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

To generate a TBB file, follow the procedure:

1. Go to OTP tool from the Tool Access bar:
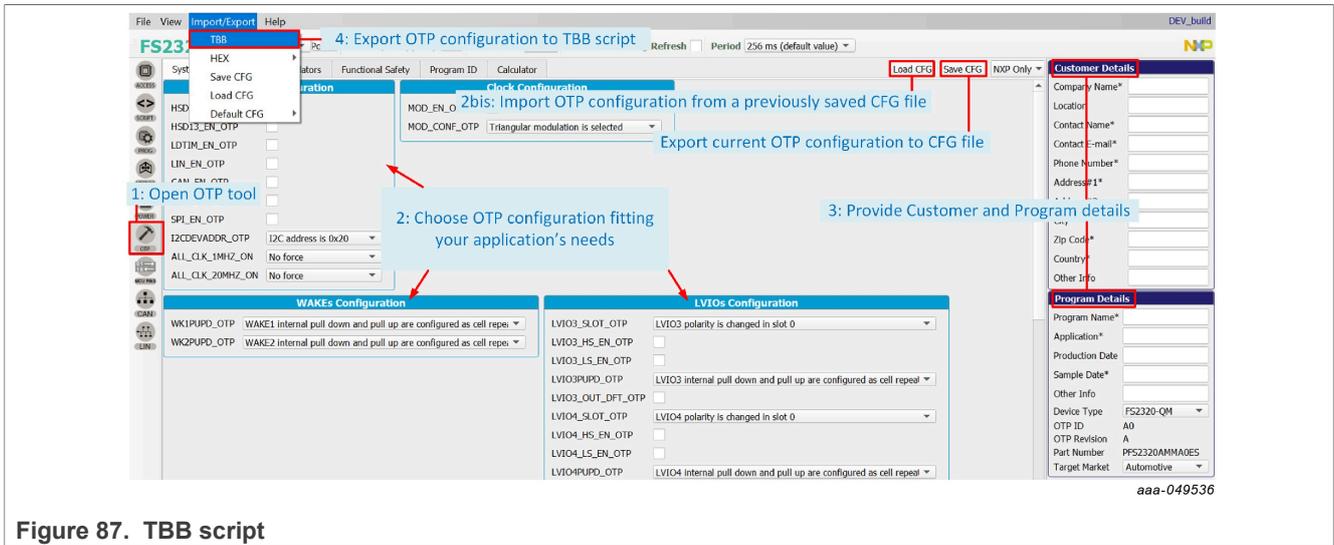
Figure 87. TBB script

2. Configure the OTP parameters to fit the application's needs or import OTP configuration from a previously saved CFG file. Be aware that the displayed parameters differ depending on selected device type (ASIL B or QM) in Program Details box.

3. Enter Customer and Program details in the window on the right side.

4. When done, export directly the OTP configuration 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:

Note: The script file must have a "FS2300" prefix, with xx being the part version.

6. TBB is now generated and saved. See Section 7.11.

7.5 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 infinite timeout and window fully opened (equivalent to disabled).

1. Set up and connect the KITFS23LDOEVM to the GUI using Section 7.1 and Section 7.2.
2. If the FS2300 part is empty or if the configuration loaded from OTP should be modified:
   a. Enter Test mode using Section 7.3.2.2.
   b. Load a configuration in MIRROR tool, then write the configuration in the Mirror registers.
   c. Exit Test mode from the Connection toolbar.
3. Exit the OTP mode by turning SW12 OFF, or clicking the Exit OTP mode button from the Connection toolbar.
4. Go to ACCESS Main Tab and click **Read all** to check that the device is in Normal, Debug, and INIT mode.
5. Go to ACCESS Watchdog and configure Watchdog with Infinite Time Out.
6. Unlock the INIT phase exit by writing '1' in Lock INIT from ACCESS Main Tab in Device State box.
7. Exit Debug mode from ACCESS Main Tab in Device State box.
8. In ACCESS Watchdog, write seven WD Answer Good, then release FS0B and LIMP0.

Figure 88 shows the sequence of steps (from step A to step J) to perform in ACCESS tool:
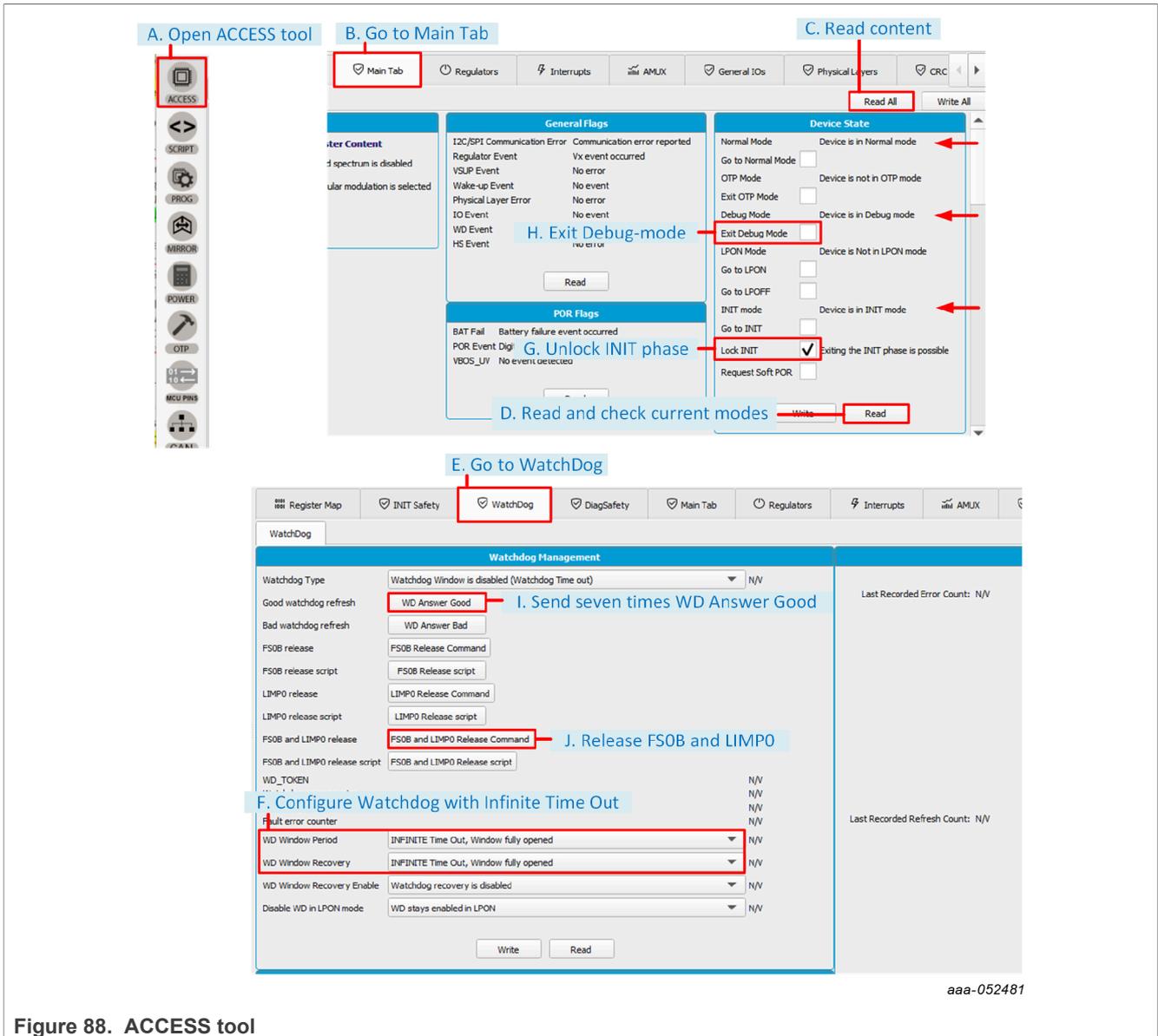
Figure 88. ACCESS tool

Note: *If FS0B and LIMP0 cannot be released after resetting the fault error counter, check that no interrupt is still pending and triggering FS0B or LIMP0 assertion.*

7.6 OTP and Mirror registers

The device incorporates one OTP block for configuration. The OTP block is shared for main parameters and fail-safe parameters. Two sectors, S1 and S1bis, are available for the OTP block so the device can be fused twice. The OTP configuration scheme is shown in Figure 89.

Figure 89. OTP block diagram
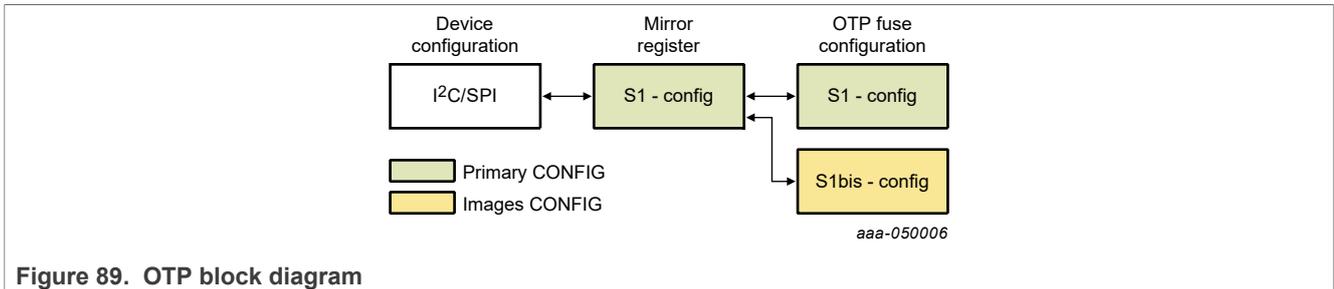
At device start up, the content of the last-programmed sector is loaded into the Mirror registers. The Mirror registers content is accessible from the NXP GUI. See Section 6.5.5. The NXP GUI manages the Mirror configuration, which facilitates access and content manipulation for OTP emulation. See Section 7.7.

The NXP GUI provides a tool to perform OTP programming, see Section 7.8. The first sector to be burned is S1, the second is S1bis. The NXP GUI automatically manages the second sector programming. It is not possible to revert to the previous sector. Once the user has reached S1bis, it is not possible to burn the part again. However, OTP emulation mode is still available.

7.7 Operate OTP emulation by loading configuration in the 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.

Mirror registers can be read and written in Test mode only. See Section 7.3.2.

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

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

7.7.1 Modify the Mirror registers with a TBB script

From a Test mode active environment, the Mirror registers can be configured using the SCRIPT tool. See Section 6.5.4.

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 a "FS2300" prefix.*
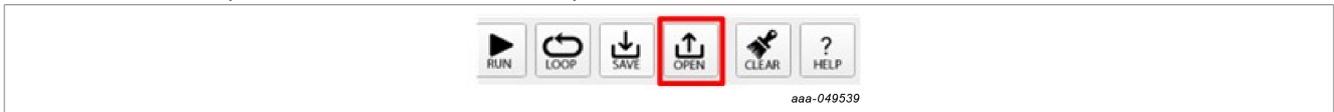
2. To execute the TBB script, click **RUN** in the bar at the bottom of the Script Command window:

3. Deactivate Test mode to start device with given configuration and load the SPI/I^{2}C functional registers with the Mirror registers content.

As a crosscheck, open the ACCESS tool and check the fields in the Regulators tab, for example. If the operation completed without problems, all fields should display expected configuration.

7.7.2 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 6.5.5. This requires Test mode to be activated.

1. Open the MIRROR tool from the Tool Access bar:

aaa-049541
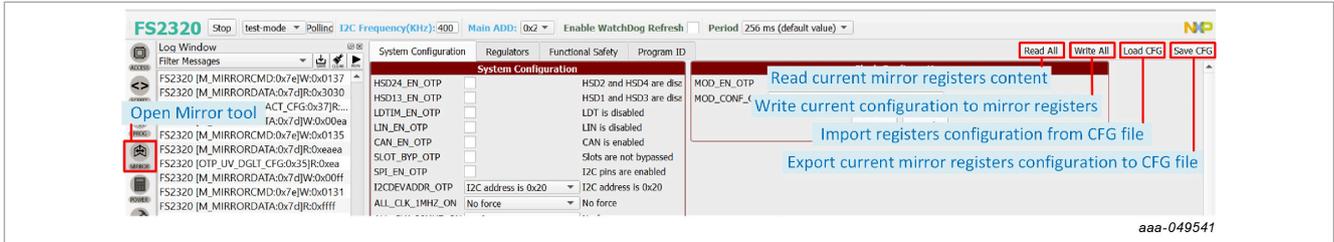
2. Configure Mirror registers to fit the application or load an existing configuration by importing a CFG file.
   Note: It is possible to load a configuration from a previously made configuration saved as a CFG file, using the Load CFG button in the top-right corner of the Mirror tool.
3. Once done, click **Write All**. As a crosscheck, click **Read all**. If the operation completed without problems, all fields should display expected configuration.
4. Deactivate Test mode to load the Mirror registers with the given configuration and start the device.
   Note: It is possible to save a current mirror registers configuration as a CFG file, using the Save CFG button in the top-right corner of the Mirror tool.

7.8 Programming an OTP configuration

The PROG tool is used to permanently burn the FS2300 fuses with the customer's OTP configuration from a TBB script file. See Section 6.5.3.

The TBB script can be generated from an OTP configuration. See Section 7.4.

Note: The TBB file must have "FS2300" as a prefix.

An FS2300 device can be burned just twice.

Requirement: Make sure that 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 7.1.
2. Set switches SW10 and SW12 ON (enables OTP 8 V generation).
3. Verify that jumper J30 is ON (enables Debug mode).
4. Plug the USB cable between the PC and the board. The blue LED D14 for OTP 8 V generation and the green LED D15 for Debug are 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 Tool Access bar.
9. Check the flags in the Sector Flags section of the Fuse Box Status window. See Section 6.5.3.3.

The device is empty and capable of being burned if Sector 1 is set to 0 (blue). Once the sector is burned, the sector flag is set to 1 (orange). The figure to the right shows the Sector Flags status in the Programming interface for an empty part.

Figure 90. Sector flags status for an empty part

The figure to the right shows the Sector Flags status in the Programming interface for a part burned once.
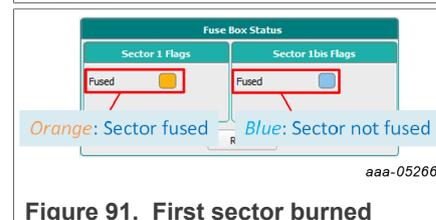
Figure 91. First sector burned

Programming procedure:

1. Set the default jumper configuration. See Section 7.1.
2. Set switches SW10 and SW12 ON (enables OTP 8 V generation).
3. Verify that jumper J30 is ON (enables Debug mode).
4. Plug the USB cable between the PC and the board. The blue LED D14 for OTP 8 V generation and the green LED D15 for Debug are ON.
5. Apply power supply VBAT.
6. Open the GUI in I^{2}C (default) mode, start the connection and switch to Test mode.
   For I^{2}C configurations, jump to step 8. For SPI configurations, follow step 7.
7. Open the MIRROR tool (requires Test mode) from the Tool Access bar, and set SPI_EN_OTP bit to '1'. A popup appears to inform that protocol has been changed. Change jumpers positions as recommended.
8. Open the PROG tool (requires Test mode) from the Tool Access bar, and load the TBB file.
9. Click **Program** to start the burning process.
10. A popup window appears to ask 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 Normal mode.

A good check is to use the AMUX panel in the ACCESS tool to see if all regulators values are correct.
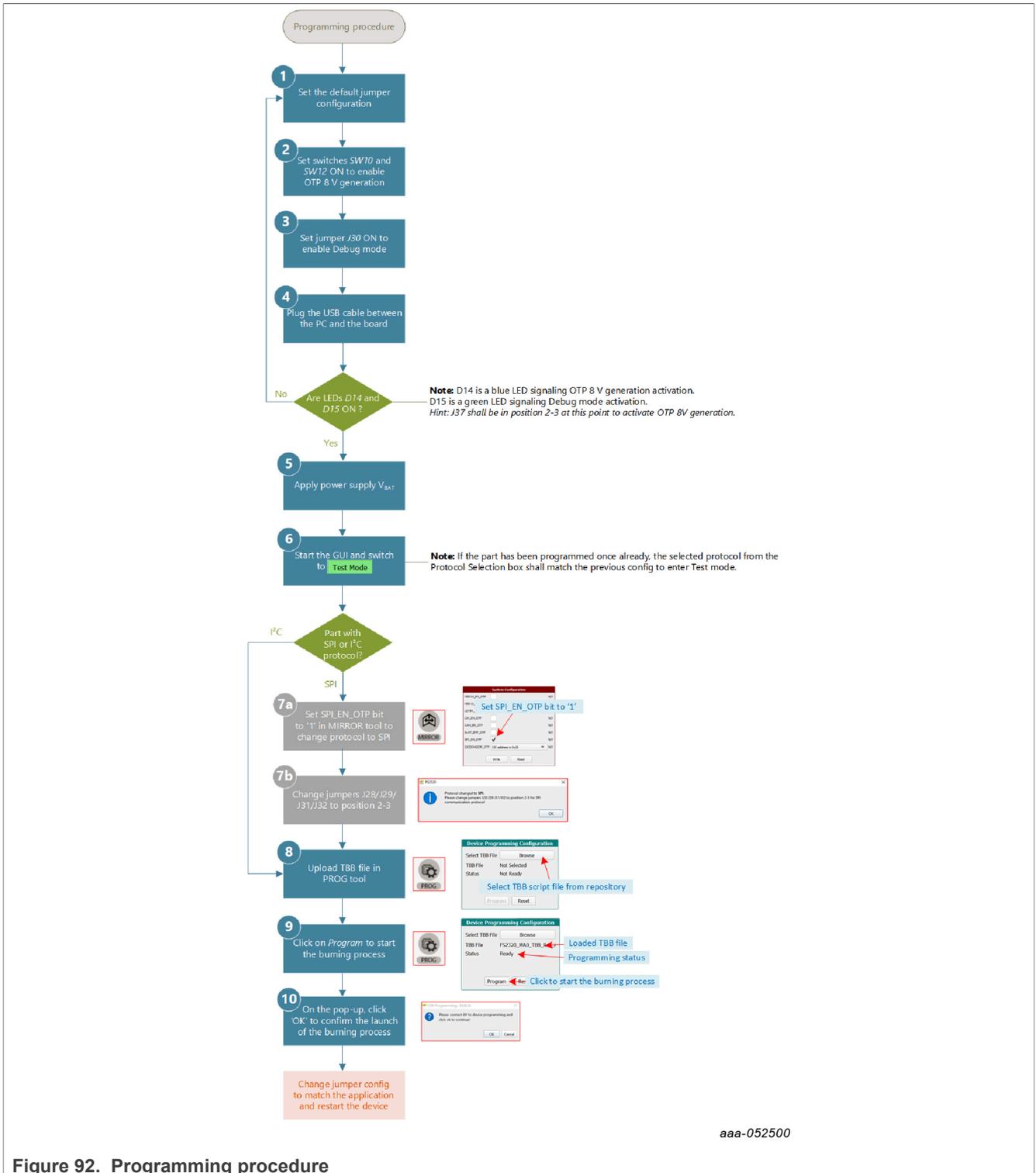
Figure 92. Programming procedure

7.9 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 FS23 device. The commands are sent using I²C or SPI protocol depending on user's choice.

Requests are sent to FS23 after a user action on the graphical interface. Answers are received by MCU, then displayed to the user in the graphical interface. These exchanges are stored in the Log file.

When the user must operate the same routine multiple times on the device, for example to automate loading the Mirror registers, as shown in <u>Figure 93</u>, it is possible to record the actions from the Log file then run the routine later as a Script.
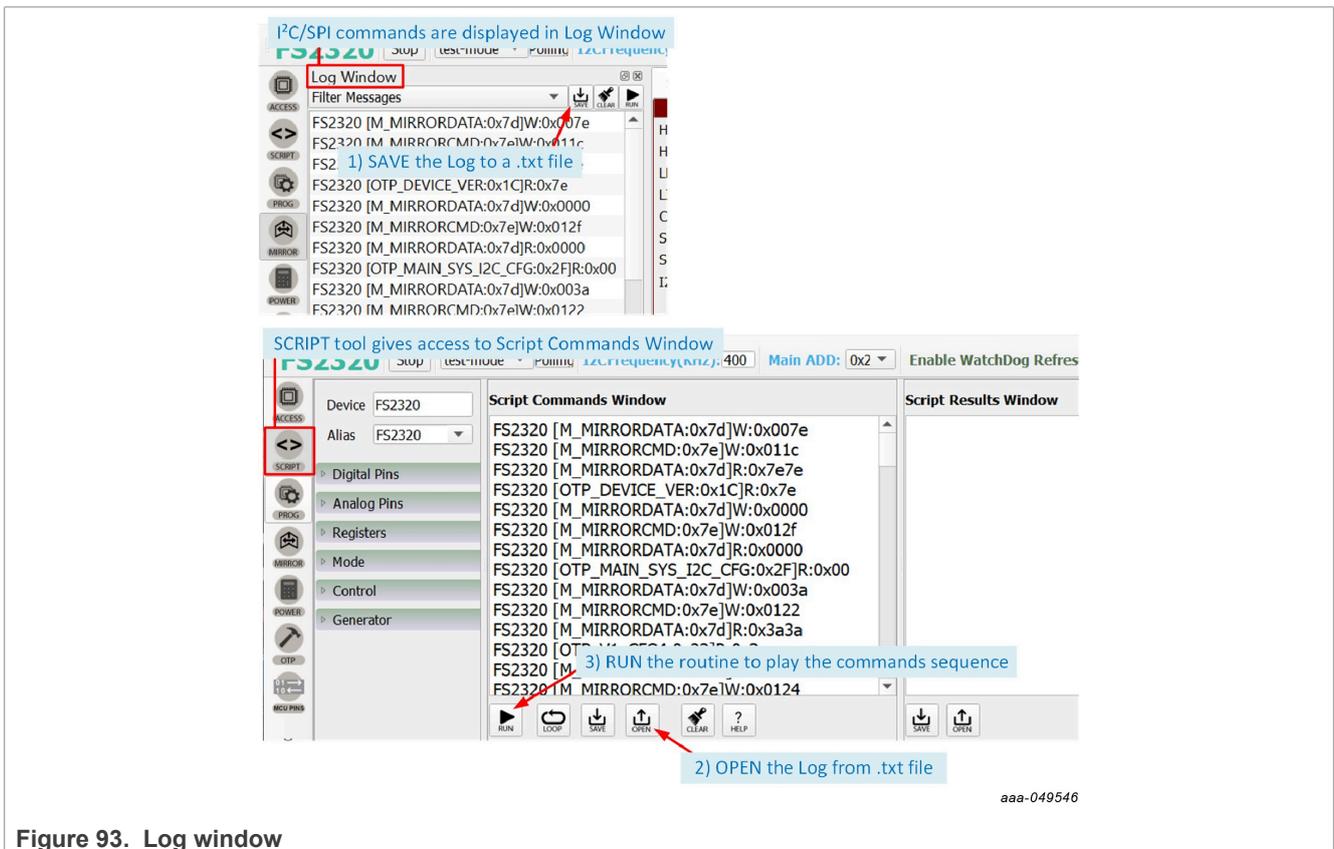
Figure 93. Log window

Note: *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.*

7.10 INIT CRC computation

This section gives guidance on how to compute INIT CRC.

A specific tool for INIT CRC computation is available in the GUI. See Section 6.5.6.11.

Procedure for INIT CRC computation

1. Read the FS configuration registers and extract the following bits.

aaa-049548
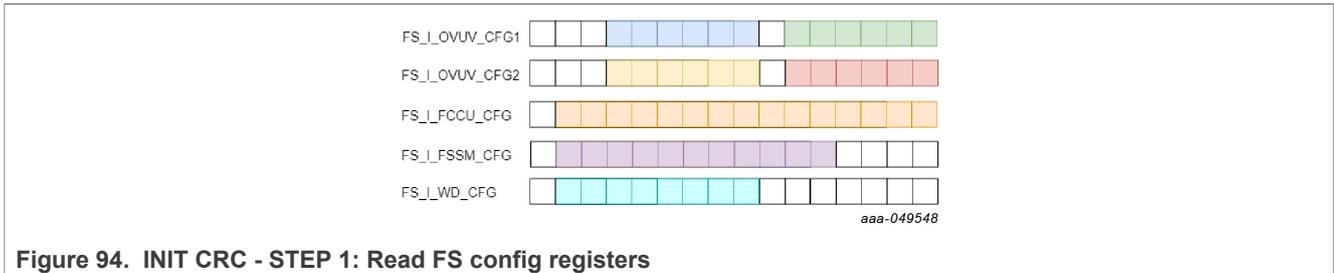
Figure 94. INIT CRC - STEP 1: Read FS config registers

2. Create the 58-bit word by concatenating the 58 bits.

aaa-049549
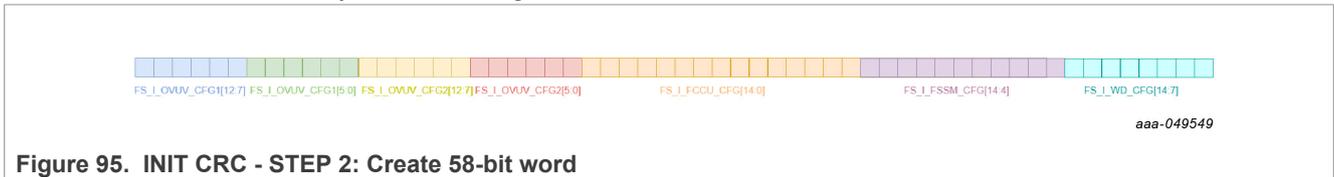
Figure 95. INIT CRC - STEP 2: Create 58-bit word

3. Compute INIT CRC **bitwise** using 0x1D polynomial.

INIT CRC computation algorithm

crc_result = 0xFF

for each of the 58 bits successively (starting with MSB):
    msb = crc_result(7) XOR *BITVALUE*
    crc_result(7) = crc_result(6)
    crc_result(6) = crc_result(5)
    crc_result(5) = crc_result(4)
    crc_result(4) = crc_result(3) XOR msb
    crc_result(3) = crc_result(2) XOR msb
    crc_result(2) = crc_result(1) XOR msb
    crc_result(1) = crc_result(0)
    crc_result(0) = msb

read result in crc_result

aaa-049547

Figure 96. INIT CRC - STEP 3: CRC bitwise computation

7.11 TBB script example

This TBB script, which corresponds to FS2300 HVBUCK ASIL B version, is given as an example code.

```
//FS2300 - OTP Editor
//file generated on jeu. juin 2 11:00:02 2022
//Device Type : FS2300-ASILB
//OTP ID : D0
//OTP Revision : A
//Part Marking : PFS2300AMBD0ES
//Customer : [Company Name]
//Write main registers
//Test mode entry
SET_MODE:FS2300:test-mode
//Verify test mode entry
GET_REG:FS2300:M_TestMode:M_TM_STATUS1
//Configure OTP Mirror Registers SET_REG:FS2300:OTP:M_
MIRRORDATA:0x007e
SET_REG:FS2300:OTP:M_MIRRORCMD:0x011C
SET_REG:FS2300:OTP:M_MIRRORDATA:0x007c
SET_REG:FS2300:OTP:M_MIRRORCMD:0x011D
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0003
SET_REG:FS2300:OTP:M_MIRRORCMD:0x011E
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0074
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0127
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0031
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0128
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0074
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0129
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0040
SET_REG:FS2300:OTP:M_MIRRORCMD:0x012A
SET_REG:FS2300:OTP:M_MIRRORDATA:0x00ff
SET_REG:FS2300:OTP:M_MIRRORCMD:0x012B
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0010

SET_REG:FS2300:OTP:M_MIRRORCMD:0x012C
SET_REG:FS2300:OTP:M_MIRRORDATA:0x007f
SET_REG:FS2300:OTP:M_MIRRORCMD:0x012D
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0003
SET_REG:FS2300:OTP:M_MIRRORCMD:0x012E
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0000
SET_REG:FS2300:OTP:M_MIRRORCMD:0x012F
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0000
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0130
SET_REG:FS2300:OTP:M_MIRRORDATA:0x00ff
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0131
SET_REG:FS2300:OTP:M_MIRRORDATA:0x00cf
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0132
SET_REG:FS2300:OTP:M_MIRRORDATA:0x00cf
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0133
SET_REG:FS2300:OTP:M_MIRRORDATA:0x00cf
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0134
SET_REG:FS2300:OTP:M_MIRRORDATA:0x00ea
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0135
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0008
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0136
SET_REG:FS2300:OTP:M_MIRRORDATA:0x0030
SET_REG:FS2300:OTP:M_MIRRORCMD:0x0137
//OTP Command CRC_Fill + GO SET_REG:FS2300:M_OTP:M_
OTPCMD:0x0125
//OTP Command CRC_Check + GO
SET_REG:FS2300:M_OTP:M_OTPCMD:0x0124
//Verify test mode entry
GET_REG:FS2300:M_TestMode:M_TM_STATUS1
// END MAIN
```

8 Legal information

8.1 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.

8.2 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 http://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.

8.3 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.

UM11879

User manual

All information provided in this document is subject to legal disclaimers.

Rev. 2 — 25 September 2023

© 2023 NXP B.V. All rights reserved.

69 / 73

Tables

UM11879

User manual

All information provided in this document is subject to legal disclaimers.

Rev. 2 — 25 September 2023

© 2023 NXP B.V. All rights reserved.

70 / 73

Figures

UM11879
All information provided in this document is subject to legal disclaimers.
© 2023 NXP B.V. All rights reserved.

User manual
Rev. 2 — 25 September 2023

71 / 73

Contents