UM11502

KITFS84AUTEVM evaluation board

Rev. 1 — 28 October 2020 User manual

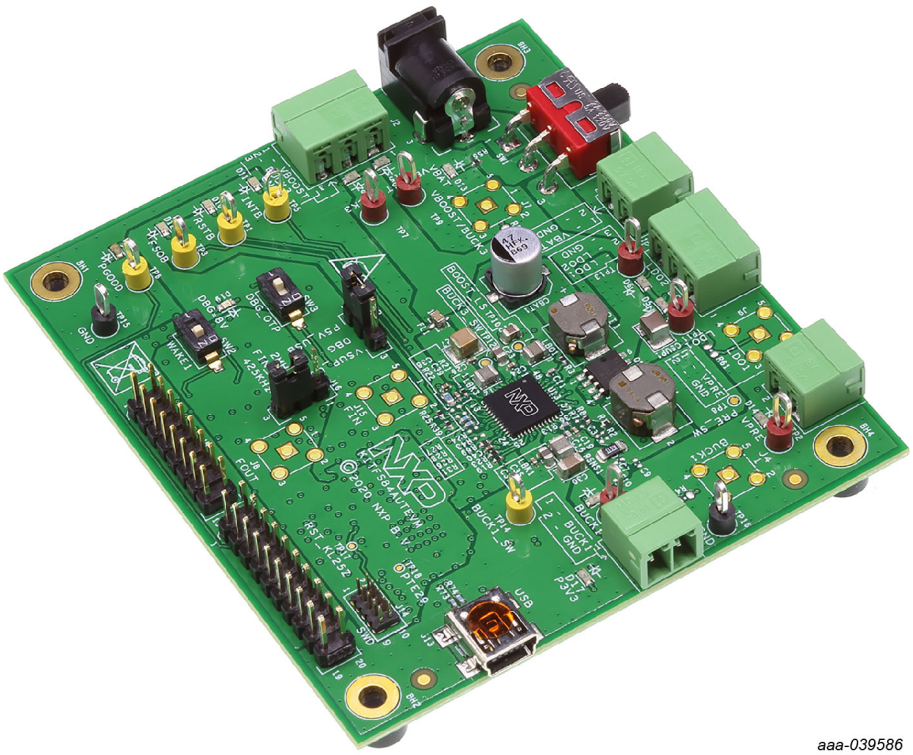


Figure 1. KITFS84AUTEVM

Important Notice

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

This evaluation kit is intended for use of ENGINEERING DEVELOPMENT OR EVALUATION PURPOSES ONLY. It is provided as a sample IC pre-soldered to a printed circuit board to make it easier to access inputs, outputs, and supply terminals. This evaluation board may be used with any development system or other source of I/O signals by simply connecting it to the host MCU or computer board via off-the-shelf cables. This 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 will be heavily dependent on proper printed circuit board layout and heat sinking design as well as attention to supply filtering, transient suppression, and I/O signal quality.

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

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

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

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

Should the Buyer purchase or use NXP products for any such unintended or unauthorized application, the Buyer shall indemnify and hold NXP and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges NXP was negligent regarding the design or manufacture of the part.

1 Introduction

This document is the user guide for the KITFS84AUTEVM evaluation board.
This document is intended for the engineers involved in the evaluation, design, implementation, and validation of FS84 QFN48EP Fail-safe system basis chip with multiple SMPS and LDO.

The scope of this document is to provide the user with information to evaluate the FS8400 Fail-safe system basis chip with multiple SMPS and LDO. This document covers connecting the hardware, installing the software and tools, configuring the environment and using the kit.

The KITFS84AUTEVM enables development on FS84 QFN48EP family of devices. The kit can be connected to the FlexGUI software which allows you to play with registers, try OTP configurations, and burn the part.

It is delivered with empty OTP content in order to leave the opportunity to the user to burn the OTP configuration. The board contains a superset device (MFS8416AMBP0ES), allowing tests on all the FS84 QFN48EP derivatives.

2 Finding kit resources and information on the NXP web site

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

The information page for KITFS84AUTEVM evaluation board is at http://www.nxp.com/KITFS84AUTEVM. The information page provides overview information, documentation, software and tools, parametrics, ordering information and a **Getting Started** tab. The **Getting Started** tab provides quick-reference information applicable to using the KITFS84AUTEVM evaluation board, including the downloadable assets referenced in this document.

2.1 Collaborate in the NXP community

The NXP community is for sharing ideas and tips, ask and answer technical questions, and receive input on just about any embedded design topic.

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

3 Getting ready

Working with the KITFS84AUTEVM requires the kit contents, additional hardware and a Windows PC workstation with installed software.

3.1 Kit contents

- Assembled and tested evaluation board in an anti-static bag
- 3.0 ft USB-STD A to USB-B-mini cable
- Three connectors, terminal block plug, 2 pos., str. 3.81 mm
- Two connectors, terminal block plug, 3 pos., str. 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.

- Power supply with a range of 8.0 V to 40 V and a current limit set initially to 1.0 A (maximum current consumption can be up to 6.5 A)

3.3 Windows PC workstation

This evaluation board requires a Windows PC workstation. Meeting these minimum specifications should produce great results when working with this evaluation board.

- USB-enabled computer with Windows 7 or Windows 10

3.4 Software

Installing software is necessary to work with this evaluation board. All listed software is available on the evaluation board's information page at http://www.nxp.com/KITFS84AUTEVM or from the provided link.

- FlexGUI latest version
- FS84-QFN48EP-OTP.xlsm
- Java installation https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html

4 Getting to know the hardware

The KITFS84AUTEVM provides flexibility to play with all the features of the device and make measurements on the main part of the application. The KL25Z MCU installed on the board, combined with the FlexGUI software allows access to the registers in read and write mode. All regulators are accessible through connectors. Nonuser signal, like DC/DC switcher node is mapped on test points. Digital signals (SPI, RSTB, etc.) are accessible through connectors. Pin WAKE1 has a switch to control (Ignition) them. A VBAT switch is available to power On or Off the device.

This board can be operated in Emulation mode or in OTP mode. In Emulation mode, as long as the power is supplied, the board configuration stays valid. The OTP mode uses the fused configuration. The device can be fused three times. In OTP mode, the device always starts with the fused configuration, except if the user wants to overwrite OTP configuration using Emulation mode. This board is able to fuse the OTP without any extra tools or board.

4.1 Kit overview

The KITFS84AUTEVM is a hardware evaluation tool that allows performance test. The MFS8416AMBP0ES part soldered on the board can be fused three times (see Section 7.3 "Programming the device with an OTP configuration").

An Emulation mode is possible to test as many configurations as needed. The voltage monitoring hardware configuration is done through resistors. Note that this configuration can be changed by selecting the appropriate bridge resistors:

- VMON1: assigned to VPRE, 4.1 V
- VMON2: assigned to EXT_MON2 (VMON bridge for 3.3 V input)

- VMON3: assigned to BUCK3, 2.3 V
- VMON4: assigned to EXT_MON4 (VMON bridge for 5.0 V input)

This configuration can be changed by installing appropriate bridge resistors. This board was designed to sustain up to 6.0 A total on VPRE.

Layout is done using six layer PCB stack up and by following the rules for DC-DC converter layout design. The FS84 QFN48EP family can be evaluated with this board as it is populated with a superset part. The FS8416AMB supports ASIL B design.

VDDIO is assigned by default to P3V3_KL25Z. From USB voltage, an external DC-DC generates the OTP programming voltage (8.0 V) without any need for an external power supply.

4.1.1 KITFS84AUTEVM features

- VBAT power supply connectors (Jack and Phoenix)
- VPRE output capability up to 1.0 A (external MOSFET)
- VBUCK1 up to 3.6 Apeak
- VBUCK3 up to 3.6 Apeak
- VBOOST 5.0 V or 5.74 V, up to 400mA
- LDO1 and LDO2, from 1.1 V to 5.0 V, up to 400mA
- Ignition key switch
- FS0B external safety pin
- Embedded USB connection for easy connection to software GUI (access to SPI bus, IOs, RSTB, FS0B, INTB, Debug, MUX_OUT, regulators)
- LEDs that indicate signal or regulator status
- Support OTP fuse capabilities
- USB connection for register access, OTP emulation and programming

4.1.2 VMON configuration

The VMONx configuration is highly dependent on the use case. This kit is delivered with a default configuration shown in bold in Table 1. However, the user can assign VMONx differently to address the use case. As an example, VMON1 could be reassigned from VPRE to LDO1 with a wire connected between LDO1 and EXT_MON1. In this case, the user has to define the right value for R240 which depends on the nominal voltage. As a consequence, the *Resistor to set column* in Table 1, indicates that R11 and R227 resistors have to be removed.

Table 1 defines how to connect VMONx.

Table 1. VMONx assignment description

VMONx inputs	Assignment	Alternate	Nominal voltage	Resistor to set	Resistor value
VMON1	**VPRE**	**—**	**4.1 V**	**R26**	**90.9 kΩ**
		VMON_08V	0.8 V	R24	0 Ω
		EXT_MON1	User	R57	User
VMON2	**VMON_08V**	**—**	**0.8 V**	**R25**	**0 Ω**
		EXT_MON2	3.3 V	R27	68.1 kΩ
VMON3	**BUCK3**	**—**	**2.3 V**	**R37**	**41.2 kΩ**
		VMON_08V	0.8 V	R34	0 Ω
		EXT_MON3	User	R60	User

VMONx inputs	Assignment	Alternate	Nominal voltage	Resistor to set	Resistor value
VMON4	**VMON_08V**		**0.8 V**	**R36**	**0 Ω**
		EXT_MON4	5.0 V	R38	115 kΩ

VMON_08V is a fixed voltage at 0.8 V which allows to force the right voltage on VMONx. EXT_MONx are available from J10 to feed the desired voltage on VMONx (external). In this case, the resistor value from the upper side of the bridge must be defined.

The resistor location is given in *Resistor to set* column in Table 1. *Nominal voltage* column gives the voltage for which VMONx bridge is defined on the board. The bridge low-side resistor is 22.1 kΩ for each VMON.

4.1.3 VPRE compensation network

This board is delivered with a VPRE compensation network defined for VPRE 4.1 V at 450 kHz. All other VPRE configurations require a new calculation for these components.

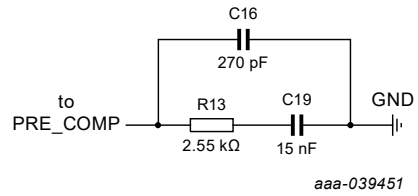


Figure 2. VPRE compensation network

Table 2. Compensation network

Components	VPRE 450 kHz	VPRE 2.2 MHz
C19	15 nF	1.2 nF
C16	270 pF	33 pF
R13	2.55 kΩ	8.4 kΩ
LPRE	**4.7 µH** or 6.8 µH	1.5 µH, 2.2 µH or 4.7 µH

4.1.4 SPI

The SPI bus is connected to KL25Z MCU.

This kit uses a KL25Z MCU to communicate with FlexGUI. It is also possible to connect the SPI to another MCU. In this case, remove R69, R70, R71, and R72 to disconnect the KL25Z MCU (see Figure 4) and connect the external MCU on J5 connector as shown in Figure 5. The external MCU can be connected on J5 connector as shown in Figure 5.

In addition to this change, be sure that VDDIO voltage domain is the same on MCU side and SBC side.

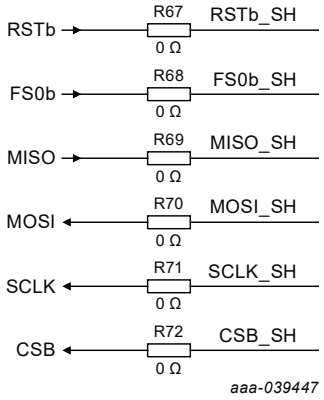


Figure 3. SPI connection to KL25Z

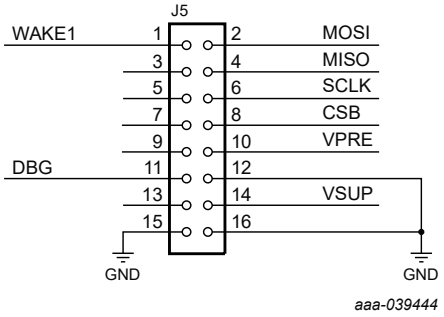


Figure 4. J5 SPI connection

4.1.5 VDDIO

The 3.3 V output voltage of the KL25Z can be used to feed VDDIO, which is the default implementation.

You can still feed VDDIO through other sources using R35, R39, R41, and R43 as shown in the following figure.

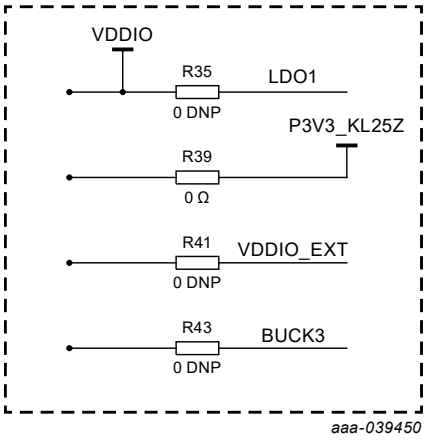


Figure 5. VDDIO selection

4.1.6 FIN external oscillator

In order to ease the FIN evaluation, a standalone oscillator is installed on the board. It supplies either 425 kHz or 2.4 MHz to the FIN input. The configuration is shown in Figure 6.

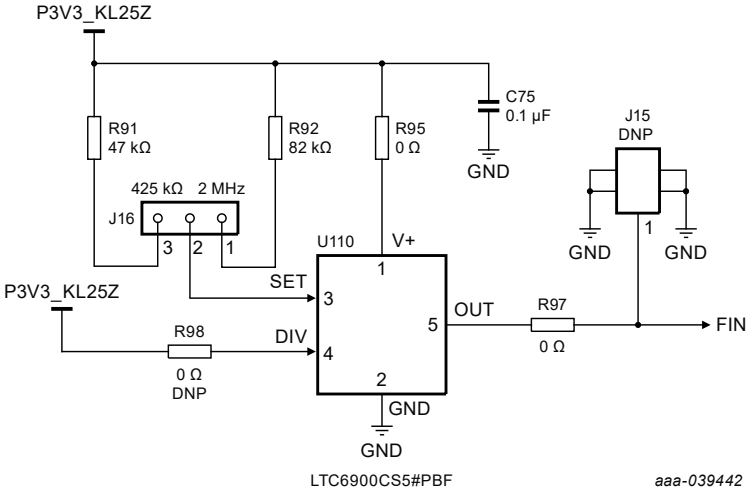


Figure 6. FIN oscillator

4.2 Device OTP user configuration

It is recommended to learn about OTP before operating with the device. The device has a high level of flexibility due to parameter configuration available in the OTP. This impacts the functionality of the device. It is key to understand how OTP parameters can be programmed, the interaction with mirror registers and the FS8416 SoC.

The OTP related operations can be performed either in Emulation mode, where the product uses a given configuration as long as power supply is not switched Off or from OTP fuse content that is valid even after a power down/power up sequence.

4.2.1 OTP and mirrors registers

There are two OTP blocks in the device. One is for the main section, and the other for the fail-safe. During configuration, each of them are using dedicated sectors. The OTP configuration scheme is shown in Figure 7 (same implementation for main and fail-safe).

The device can be fused three times using mirror registers. The user can first load the mirror register content with the desired contents, then decide either to use the device in Emulation mode or to burn the next sector. The first sector to be burned is S1, the second S1bis and the third S1ter. FlexGUI automatically manages the next sector to be burned. It is not possible to revert back to the previous sector. When the user reaches the sector S1ter, there no other possibility for burn, however emulation mode is still available.

Note: *When device is operating in Emulation mode using configuration from mirror registers, few parameters must be overwritten by SPI. This concerns regulator TSD behaviors; VPRE slew rate high-side and low-side VBOOST slew rate. See Section 8.4.10 "TestMode:Mirrors_Main and TestMode:Mirrors_Failsafe" for additional details.*

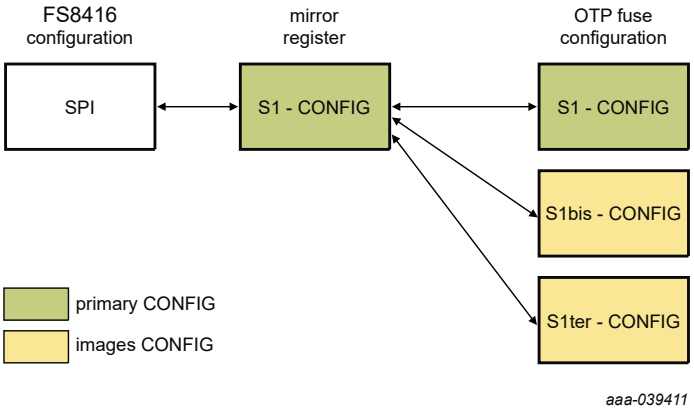


Figure 7. OTP configuration

At boot, the content of the valid sector is loaded into the Mirror Register Sector 1. The mirror register content is accessible from FlexGUI by using specific SPI commands. The mirror configuration is managed by the FlexGUI, which eases the access.

4.2.2 OTP hardware implementation

To work in OTP emulation or OTP programming, it is required to start the device in Debug mode.

Figure 8 shows the sequence to be followed to enter in Debug mode. The voltage sequence on the kit is done using switches installed on the board, while the OTP registers configuration is managed by the FlexGUI GUI. This is described in detail in the following sections.

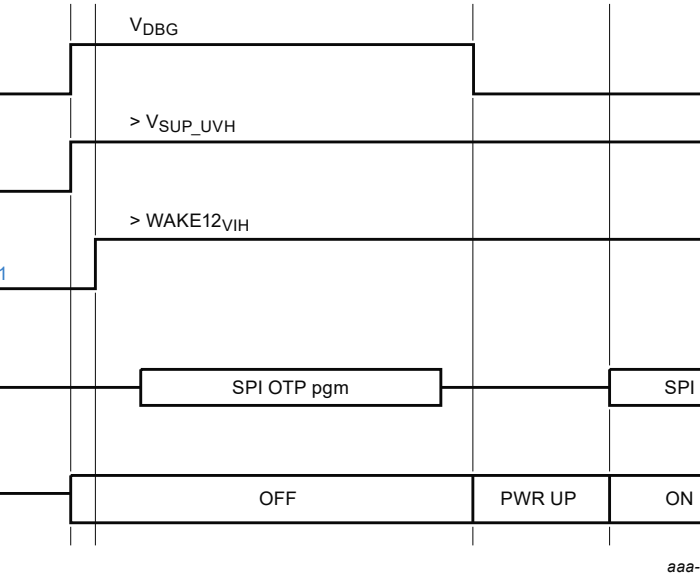


Figure 8. Debug mode entry

Figure 9 shows the hardware kit implementation.

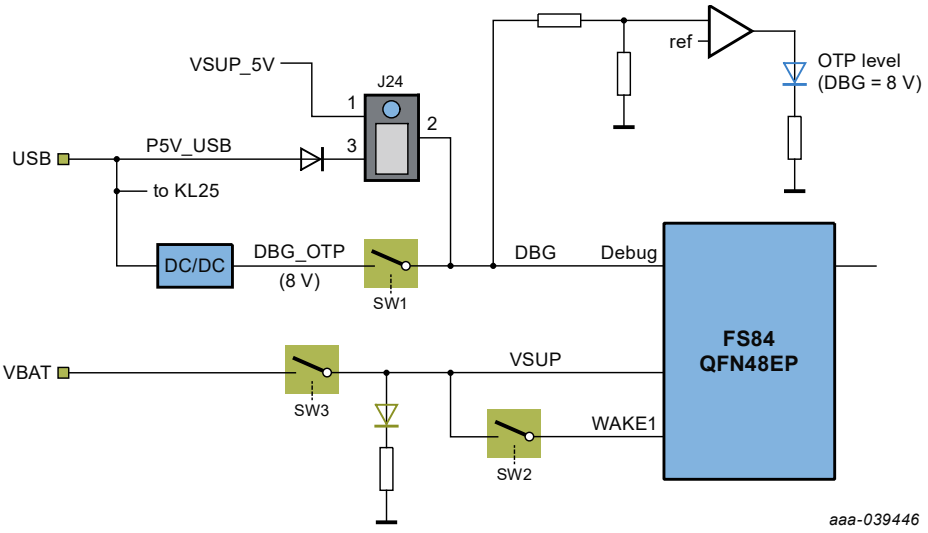
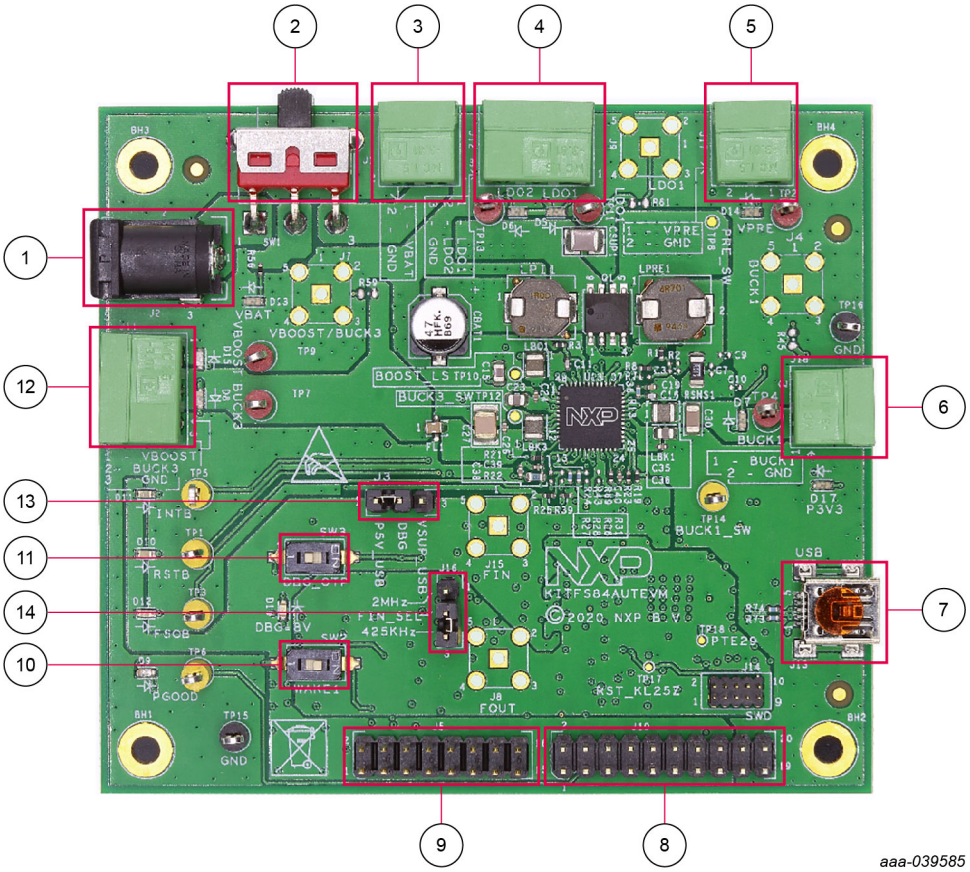


Figure 9. OTP hardware implementation

4.3 Kit featured components

Figure 10 identifies important components on the board and Table 3 provides additional details on these components.



1. VBAT Jack connector
2. VBAT three position switch
3. VBAT Phoenix connector
4. LDO1/LDO2 power supplies
5. VPRE power supply
6. BUCK1 power supply
7. USB connector (for FlexGUI control)
8. Debug connectivity
9. Programming
10. Wake1 switch
11. OTP burning voltage switch
12. VBOOST and BUCK3 power supply
13. DEBUG voltage source
14. FIN frequency selection

Figure 10. Evaluation board featured component locations

Table 3. Evaluation board component descriptions

Number	Description
1	VBAT Jack connector
2	VBAT three position switch • Left position: board supplied by Jack connector • Middle position: board not supplied • Right position: board supplied by Phoenix connector
3	VBAT Phoenix connector

Number	Description
4	LDO1/LDO2 power supply
5	VPRE power supply
6	BUCK1 power supply
7	USB connector (for FlexGUI control)
8	Debug connectivity. Access to: • VSUP, GND • FOUT/FIN • PGOOD/RSTB/FS0B • FCCUx • WAKE2 • PSYNC, AMUX • VMONx
9	Programming • SPI bus • Debug pin • VPRE, VSUP, GND
10	Wake1 switch
11	OTP burning voltage switch
12	VBOOST and BUCK3 power supply
13	DEBUG voltage source either from USB (recommended) or from VSUP
14	FIN frequency selection

4.3.1 FS84 QFN48EP: Fail-safe system basis chip with multiple SMPS and LDO

4.3.1.1 General description

The FS84 QFN48EP family is developed in compliance with ASIL D process, FS84 QFN48EP is QM and ASIL B capable. All device options are pin to pin and software compatible.

The FS84 QFN48EP is an automotive functionally safe multi-output power supply integrated circuit, with focus on Radar, Vision, ADAS domain controller, Radio and Infotainment applications. It includes multiple switch mode and linear voltage regulators. It offers external frequency synchronization input and output, for optimized system EMC performance.

The FS84 QFN48EP includes enhanced safety features, with fail-safe output, becoming a full part of a safety-oriented system partitioning, covering ASIL B safety integrity level. It is developed in compliance with ISO 26262 standard and it is qualified in compliance with AEC-Q100 rev H (Grade1, MSL3).

Several device versions are available, offering choice in number of output rails, output voltage setting, operating frequency and power up sequencing, to address multiple applications.

4.3.1.2 Features

- 40 V DC maximum input voltage for 12 V applications
- VPRE synchronous buck controller with external MOSFETs. Configurable output voltage, switching frequency, and current capability up to 10 A peak, **based on device options**.

- Low voltage integrated synchronous BUCK1 converter, dedicated to MCU core supply with SVS capability. Configurable output voltage and current capability up to 4.5 A peak.
- **Based on device options:** low voltage integrated synchronous BUCK3 converter. Configurable output voltage and current capability up to 4.5 A peak.
- BOOST converter with integrated low-side switch. Configurable output voltage and max input current up to 1.5 A peak.
- EMC optimization techniques including SMPS frequency synchronization, spread spectrum, slew rate control, manual frequency tuning
- **Based on device options:** up to two linear voltage regulators for MCU IOs and ADC supply, external physical layer. Configurable output voltage and current capability up to 400 mA DC.
- OFF mode (power down) with very low quiescent current (10 µA typ)
- 2x input pins for wake-up detection and battery voltage sensing
- Device control via 32 bits SPI
- **Based on device options:** Power synchronization pin to operate 2x FS84 devices or FS84 plus an external PMIC
- Scalable portfolio with independent monitoring circuitry, dedicated interface for MCU monitoring, simple watchdog function, power good, reset and interrupt, built-in self-test, fail-safe output
- Configuration by OTP programming. Prototype enablement to support custom setting during project development in engineering mode.

4.3.2 Indicators

The following LEDs are provided as visual output devices for the evaluation board:

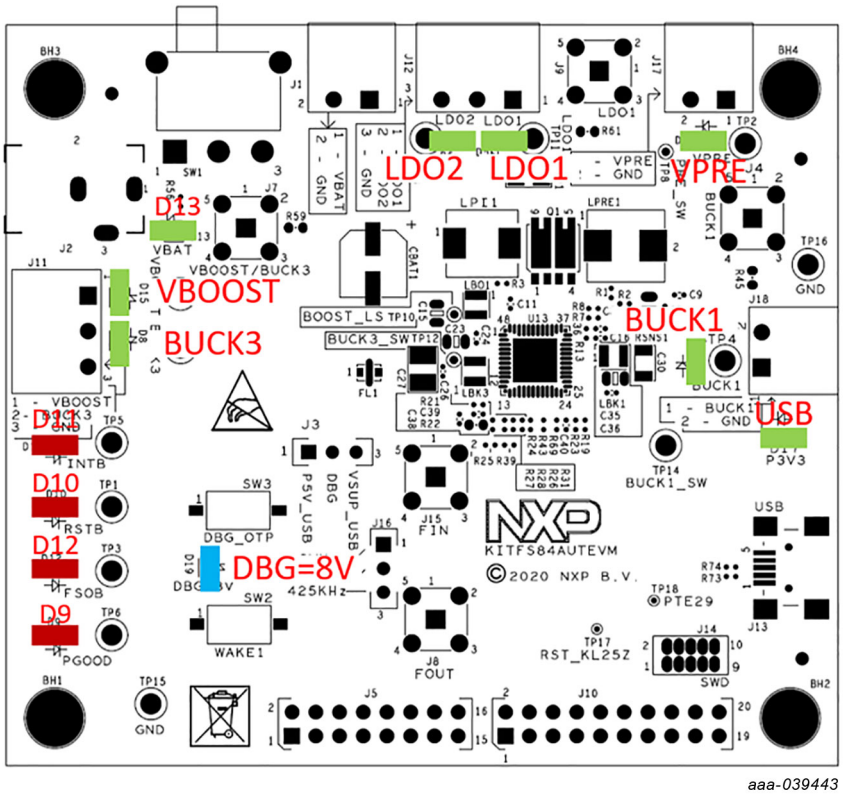


Figure 11. Evaluation board indicator locations

Table 4. Evaluation board indicator descriptions

Label	Name	Color	Description
D5	LDO1	Green	LDO1 On
D6	LDO2	Green	LDO2 On
D7	BUCK1	Green	BUCK1 On
D8	BUCK3	Green	BUCK3 On
D9	PGOOD	Green	PGOOD released
D10	RSTB	Red	RSTB asserted (logic level = 0)
D11	INTB	Red	INTB asserted (logic level = 0)
D12	FS0B	Red	FS0B asserted (logic level = 0)
D13	VBAT	Green	VBAT On
D14	VPRE	Green	VPRE On
D15	VBOOST	Green	VBOOST On
D17	P3V3_KL25	Green	P3V3_KL25 On
D19	DBG > 8.0 V	Blue	DBG pin voltage > 8.0 V (OTP programming)

4.3.3 Connectors

Figure 12 shows the location of connectors on the board.

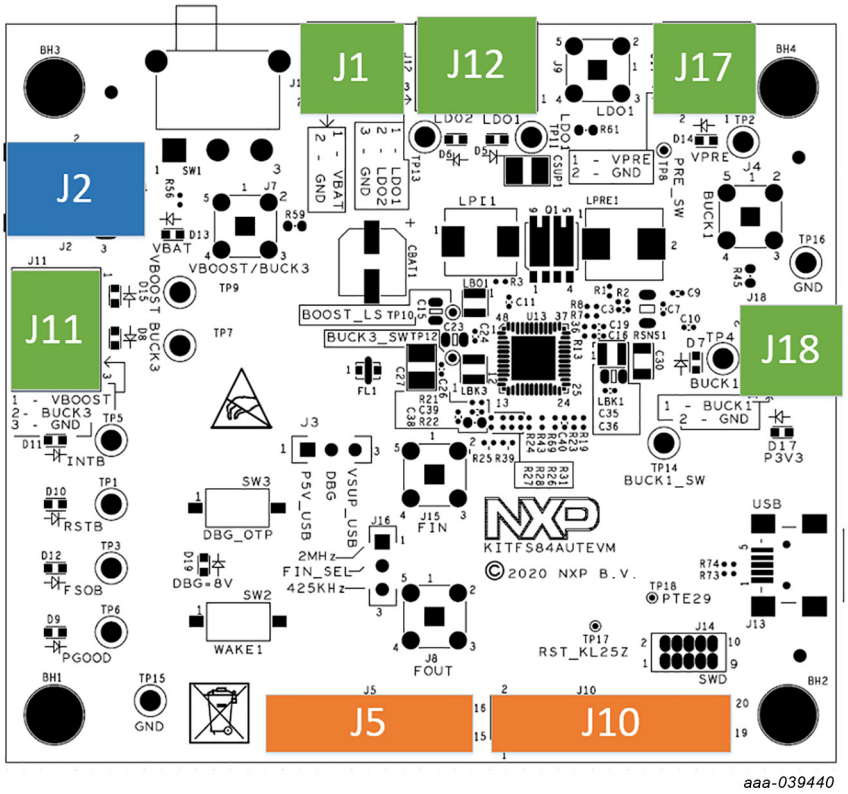


Figure 12. Evaluation board connector locations

4.3.3.1 VBAT connector (J1)

VBAT connects to the board through Phoenix connector (J1).

Table 5. V_{BAT} Phoenix connector (J1)

Schematic label	Signal name	Description
J1-1	VBAT	Battery voltage supply input
J1-2	GND	Ground

4.3.3.2 Output power supply connectors

Table 6. VBOOST/BUCK3 connector (J11)

Schematic label	Signal name	Description
J11-1	VBOOST	VBOOST output
J11-2	BUCK3	BUCK3 power supply output
J11-3	GND	Ground

Table 7. LDO1/LDO2 connector (J12)

Schematic label	Signal name	Description
J12-1	LDO1	LDO1 power supply output
J12-2	LDO2	LDO2 power supply output
J12-3	GND	Ground

Table 8. VPRE connector (J17)

Schematic label	Signal name	Description
J17-1	VPRE	VPRE power supply output
J17-2	GND	Ground

Table 9. BUCK1 connector (J18)

Schematic label	Signal name	Description
J18-1	BUCK1	BUCK1 power supply output
J18-2	GND	Ground

4.3.3.3 Debug connector (J10)

Table 10. Debug connector (J10)

Schematic label	Signal name	Description
J10-1	FOUT	Frequency synchronization output
J10-2	FIN	Frequency synchronization input
J10-3	PGOOD	Power GOOD
J10-4	VMON1_EXT	Voltage monitoring 1, from external reference
J10-5	INTB	Interrupt, active low
J10-6	VMON2_EXT	Voltage monitoring 2, from external reference
J10-7	RSTB	Reset, active low
J10-8	VMON3_EXT	Voltage monitoring 3, from external reference
J10-9	GND34	Ground
J10-10	VMON4_EXT	Voltage monitoring 4, from external reference
J10-11	AMUX	Analog multiplexer
J10-12	FS0B_Out	Fail-safe, active low
J10-13	VDDIO_EXT	VDDIO external reference
J10-14	PSYNC	Power synchronization
J10-15	VDDIO	VDDIO used by FS8416
J10-16	WAKE2_IN	Wake2 input
J10-17	FCCU1	Fault collector control unit 1
J10-18	VSUP	VSUP power supply
J10-19	FCCU2	Fault collector control unit 2
J10-20	GND	Ground

4.3.3.4 Program connector (J5)

Table 11. Program connector (J5)

Schematic label	Signal name	Description
J5-1	WAKE1	WAKE1 input
J5-2	MOSI	SPI master output slave input
J5-3	n.c.	not connected
J5-4	MISO	SPI master input slave output
J5-5	n.c.	not connected

Schematic label	Signal name	Description
J5-6	SCLK	SPI clock
J5-7	n.c.	not connected
J5-8	CSB	SPI chip select
J5-9	n.c.	not connected
J5-10	VPRE	VPRE output
J5-11	DBG	Connected to Debug pin
J5-12	GND	Ground
J5-13	n.c.	not connected
J5-14	VSUP	Connected to VSUP pin
J5-15	GND	Ground
J5-16	GND	Ground

4.3.4 Test points

The following test points provide access to various signals to and from the board.

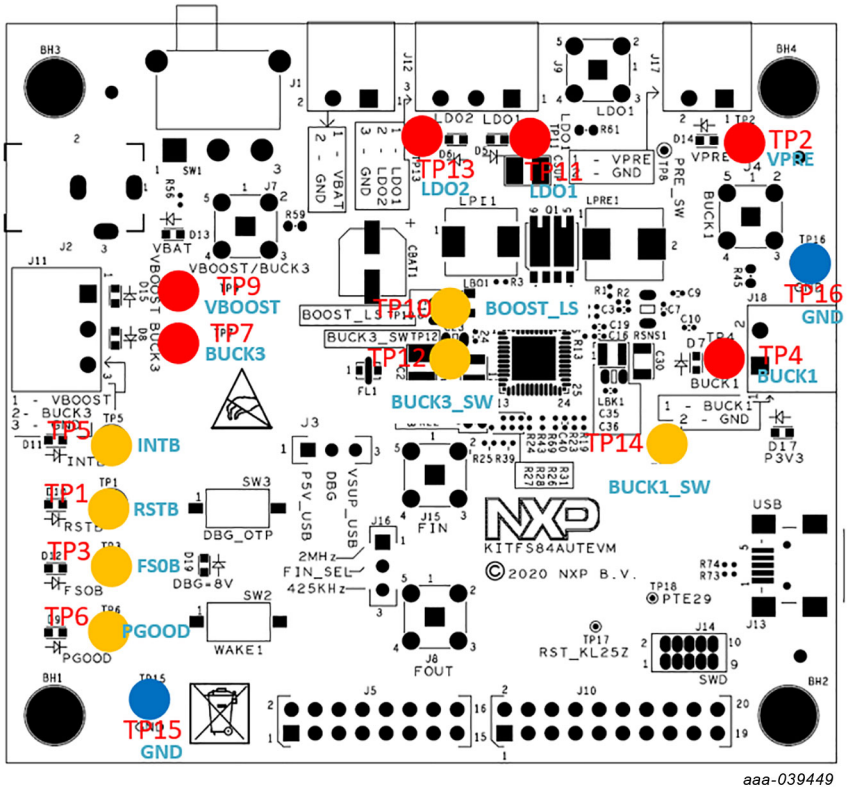


Figure 13. Evaluation board test points

Table 12. Evaluation board test point descriptions

Test point name	Signal name	Description
TP1	RSTB	Reset signal, active low
TP2	VPRE	VPRE DC/DC regulator output
TP3	FS0B	Fail-safe output, active low

Test point name	Signal name	Description
TP4	BUCK1	BUCK1 DC/DC regulator output
TP5	INTB	Interruption signal, active low
TP6	PGOOD	Power GOOD output, active low
TP7	BUCK3	BUCK3 DC/DC regulator output
TP8	PRE_SW	VPRE switcher
TP9	VBOOST	VBOOST DC/DC output
TP10	BOOST_LS	VBOOST low-side switcher
TP11	LDO1	LDO1 regulator output
TP12	BUCK3_SW	BUCK3 switcher
TP13	LDO2	LDO2 regulator output
TP14	BUCK1_SW	BUCK1 switcher
TP15	GND	Ground
TP16	GND	Ground

4.3.5 Jumpers

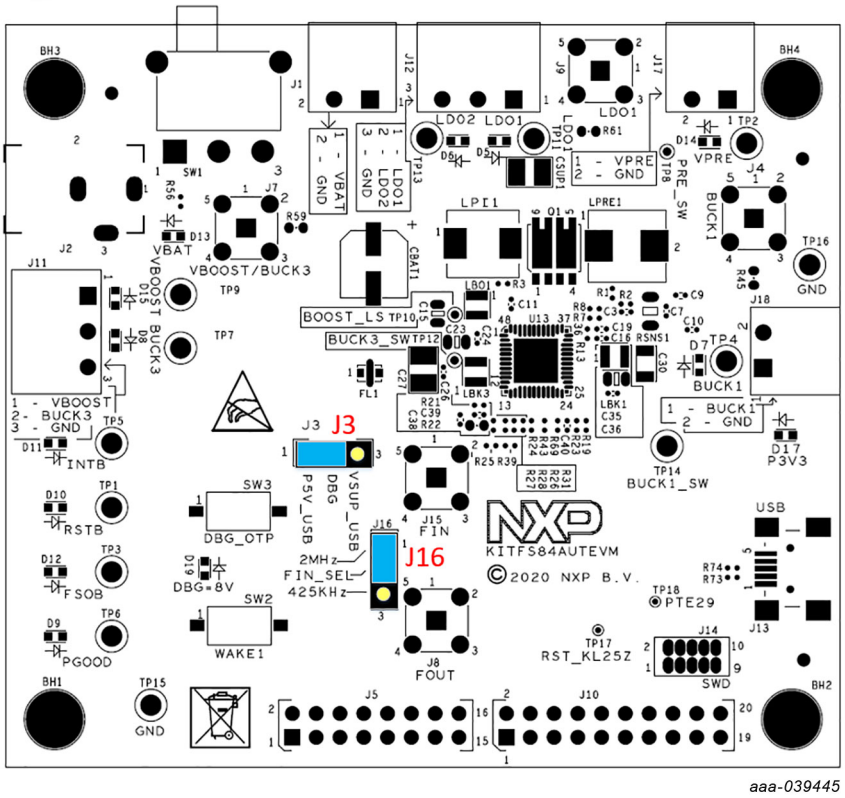


Figure 14. Evaluation board jumper locations

Table 13. Evaluation board jumper descriptions

Name	Function	Pin number	Jumper/pin function
J3	DBG entry	1−2	DBG voltage produced from USB 5.0 V
		2−3	DBG voltage produced from VBAT
J16	FIN clock selection	1−2	FIN set to 2.0 MHz
		2−3	FIN set to 425 kHz

4.3.6 Switches

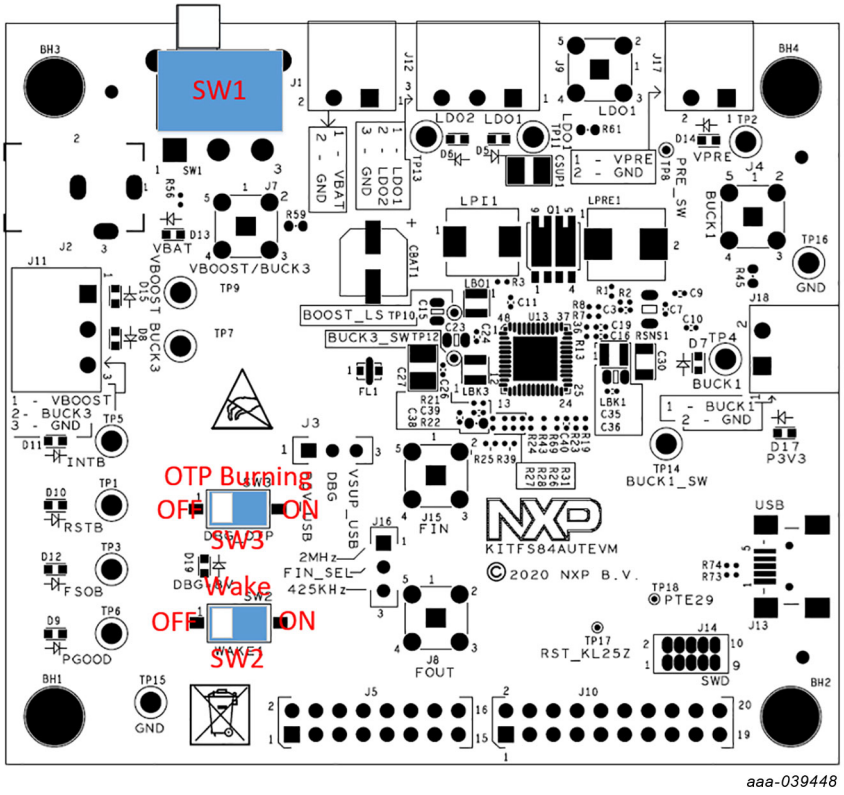


Figure 15. Switch locations

Table 14. SW3

Position	Function	Description
LEFT	OTP programming Off	OTP burning not possible
RIGHT	OTP programming On	8.0 V on DBG pin allows OTP burning (blue LED turns On to indicate this state)

Table 15. SW2

Position	Function	Description
LEFT	WAKE1 open	Wake1 pin not connected to V_{SUP}
RIGHT	WAKE1 closed	Wake1 pin connected to V_{SUP}

Table 16. SW1

Position	Function	Description
LEFT	VBAT On	VBAT from J2
MIDDLE	VBAT Off	Board not supplied
RIGHT	VBAT On	VBAT from J1

4.4 Schematic, board layout and bill of materials

The schematic, board layout and bill of materials for the KITFS84AUTEVM evaluation board are available at http://www.nxp.com/KITFS84AUTEVM.

5 Installing and configuring software and tools

This development kit uses FlexGUI software. FlexGUI software is based on Java JRE.

Preparing the Windows PC workstation consists of three steps.

1. Install the appropriate Java SE Runtime Environment (JRE).
2. Install Windows 7 FlexGUI driver.
3. Install FlexGUI software package.

5.1 Installing the Java JRE

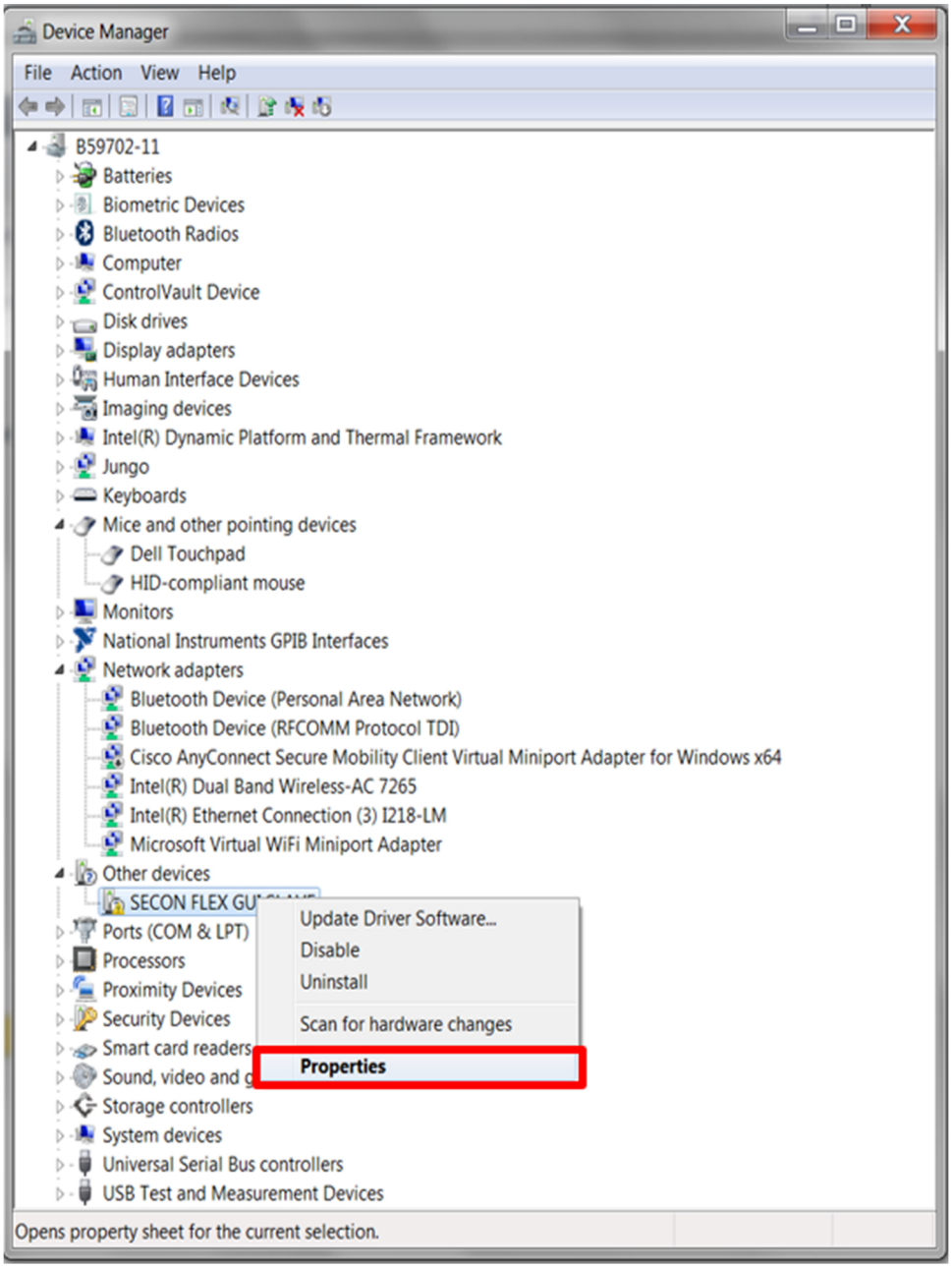
1. Download Java JRE (Java SE Runtime Environment), available at http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html (8u162 or newer).
2. Open the installer and follow the installation instructions.
3. Following the successful installation, restart the computer.

5.2 Installing Windows 7 FlexGUI driver

On Windows 7 PCs, a virtual COM port installation is required. Install the Windows 7 FlexGUI driver using the following procedure.

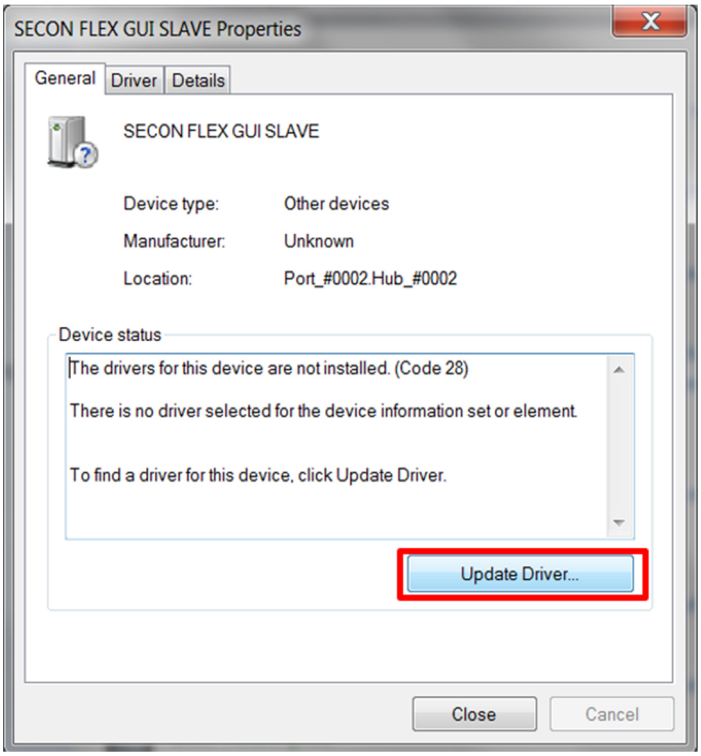
Note: *On Windows 10, it is not necessary to install virtual com port as Windows 10 uses a generic COM port driver.*

1. Connect the kit to the computer as described in Section 6 "Configuring the hardware for startup"
2. On the Windows PC, open the **Device Manager**.
3. In the **Device Manager** window, right-click on **SECON FLEX GUI SLAVE**, and then select **Properties**.



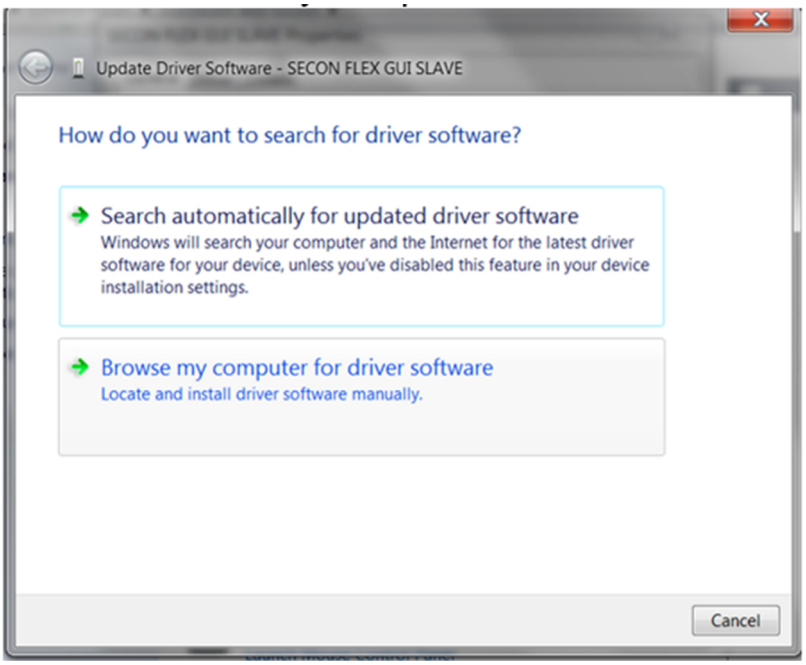
aaa-031982

4. In the **SECON FLEX GUI SLAVE Properties** window, click **Update Driver**.



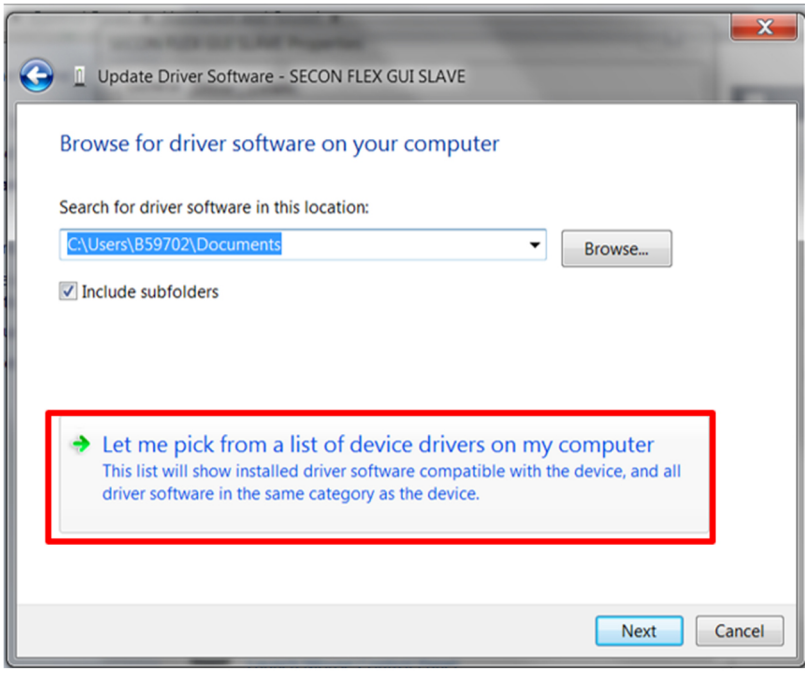
aaa-031983

5. in the **Update Software Driver window**, select **Browse my computer for driver software**.



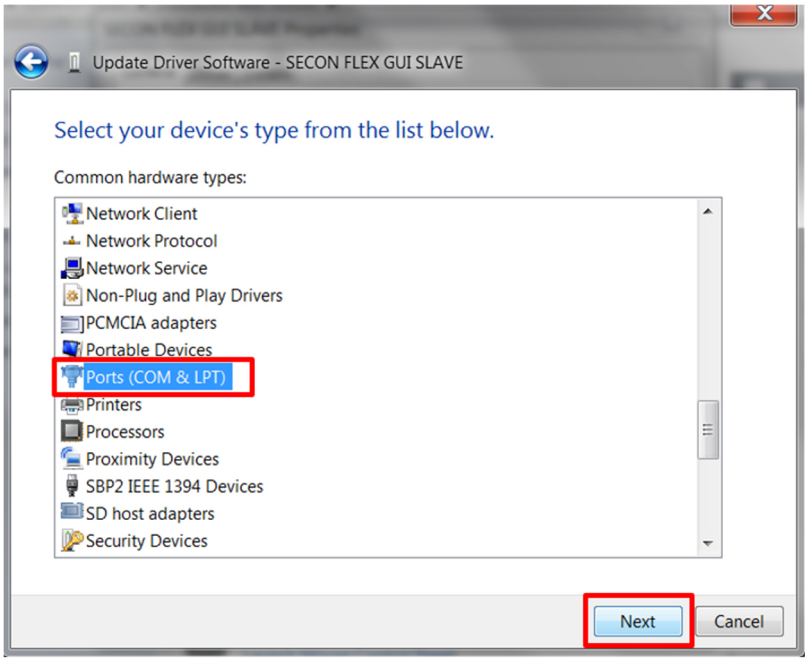
aaa-031984

6. Select **Let me pick from a list of device drivers on my computer**, and then click **Next**.



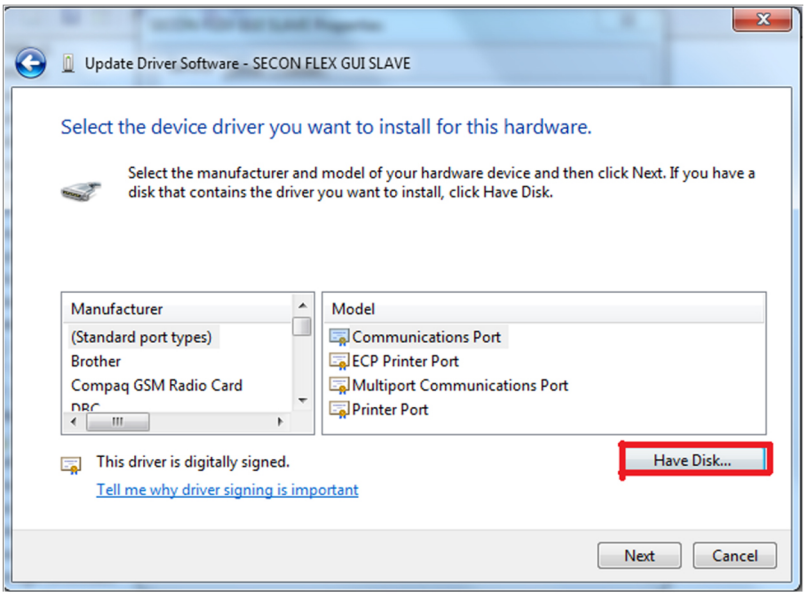
aaa-031985

7. Select **Ports (COM & LPT)** from the list, and then click **Next**.



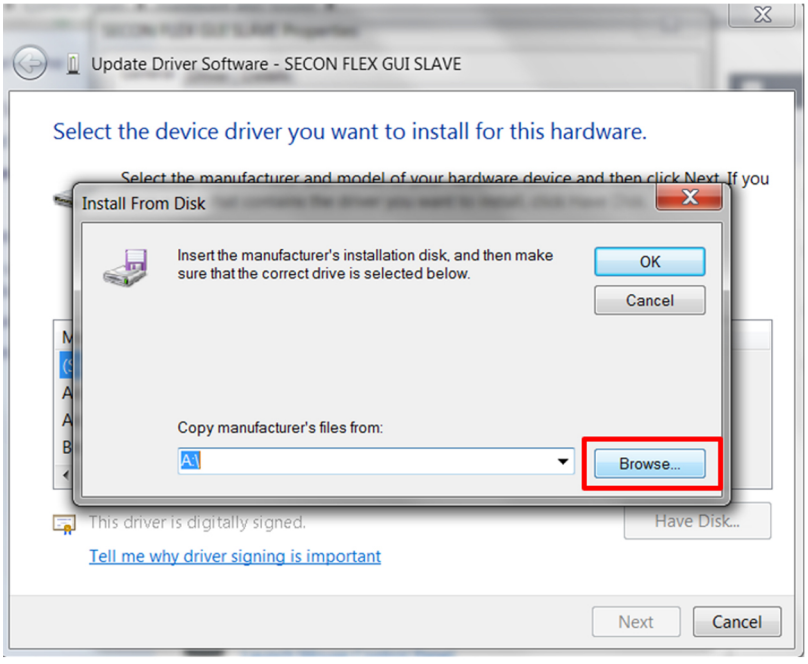
aaa-031986

8. Click **Have Disk**.



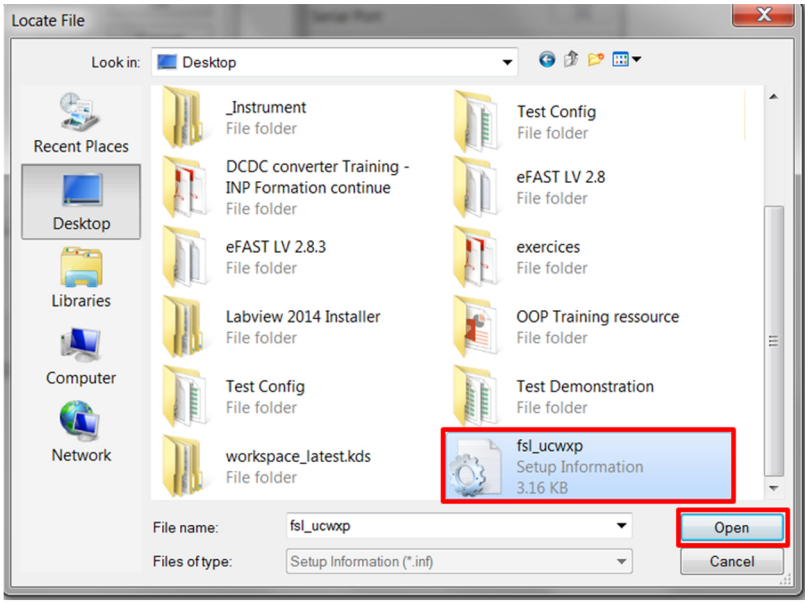
aaa-031987

9. Click **Browse**.



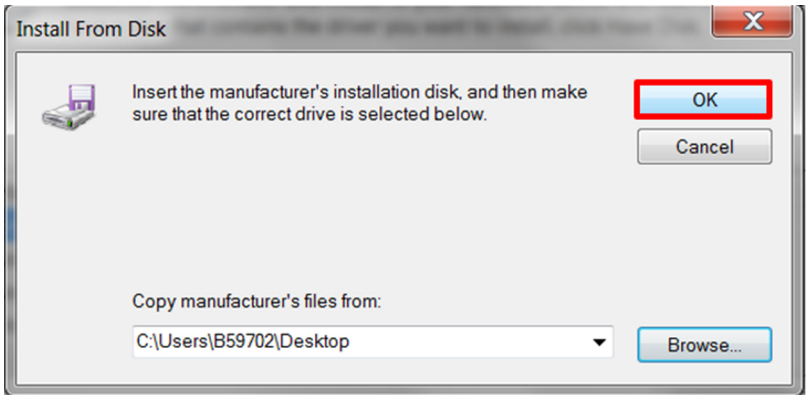
aaa-031988

10. In the **Locate File** window, locate and select **fsl_ucwxp**, and then click **Open**.



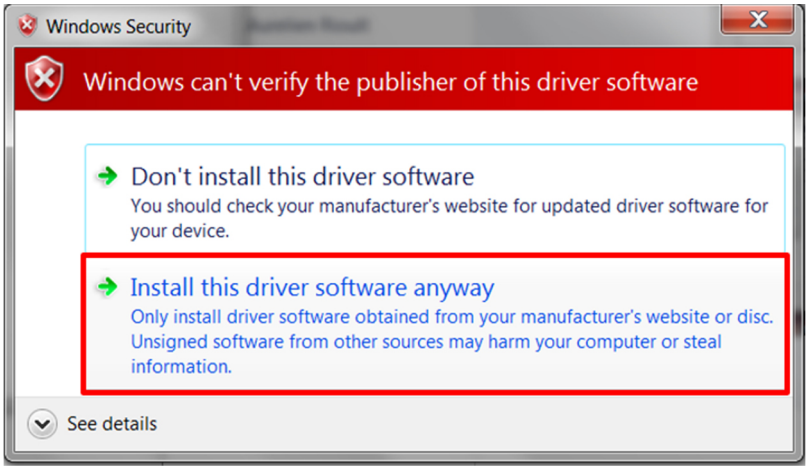
aaa-031989

11. In the **Install from Disk** window, click **OK**.



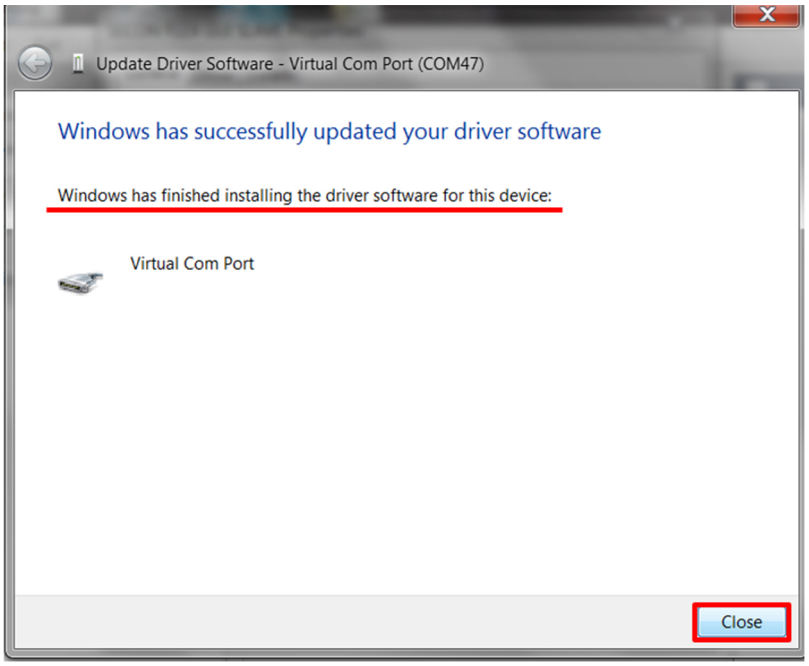
aaa-031990

12. If prompted, in the **Windows Security** window, click **Select this driver software anyway**.



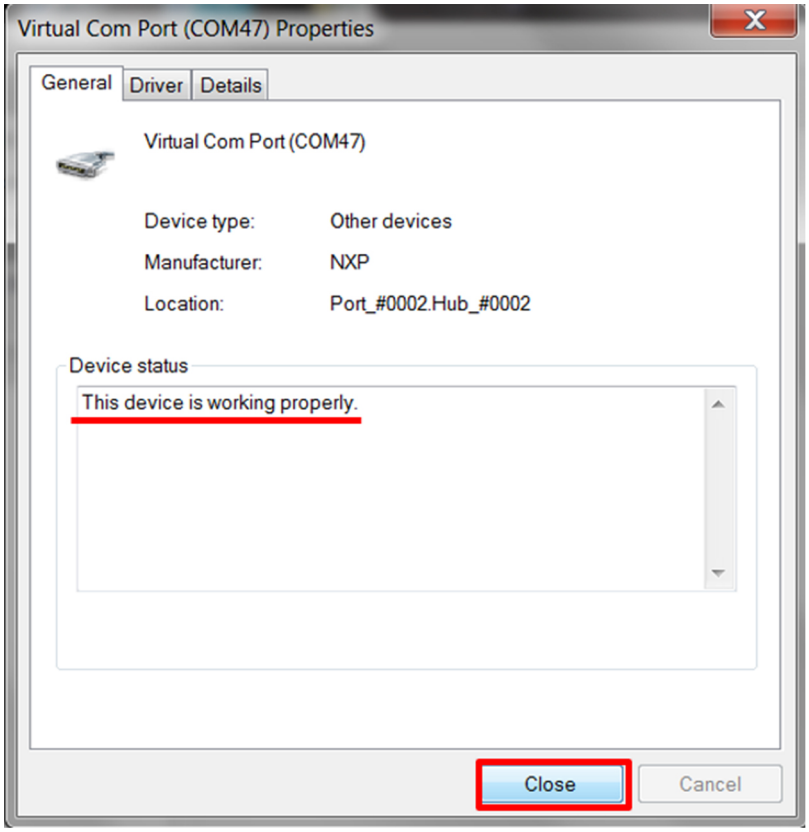
aaa-031991

13. Close the window when the installation is complete.



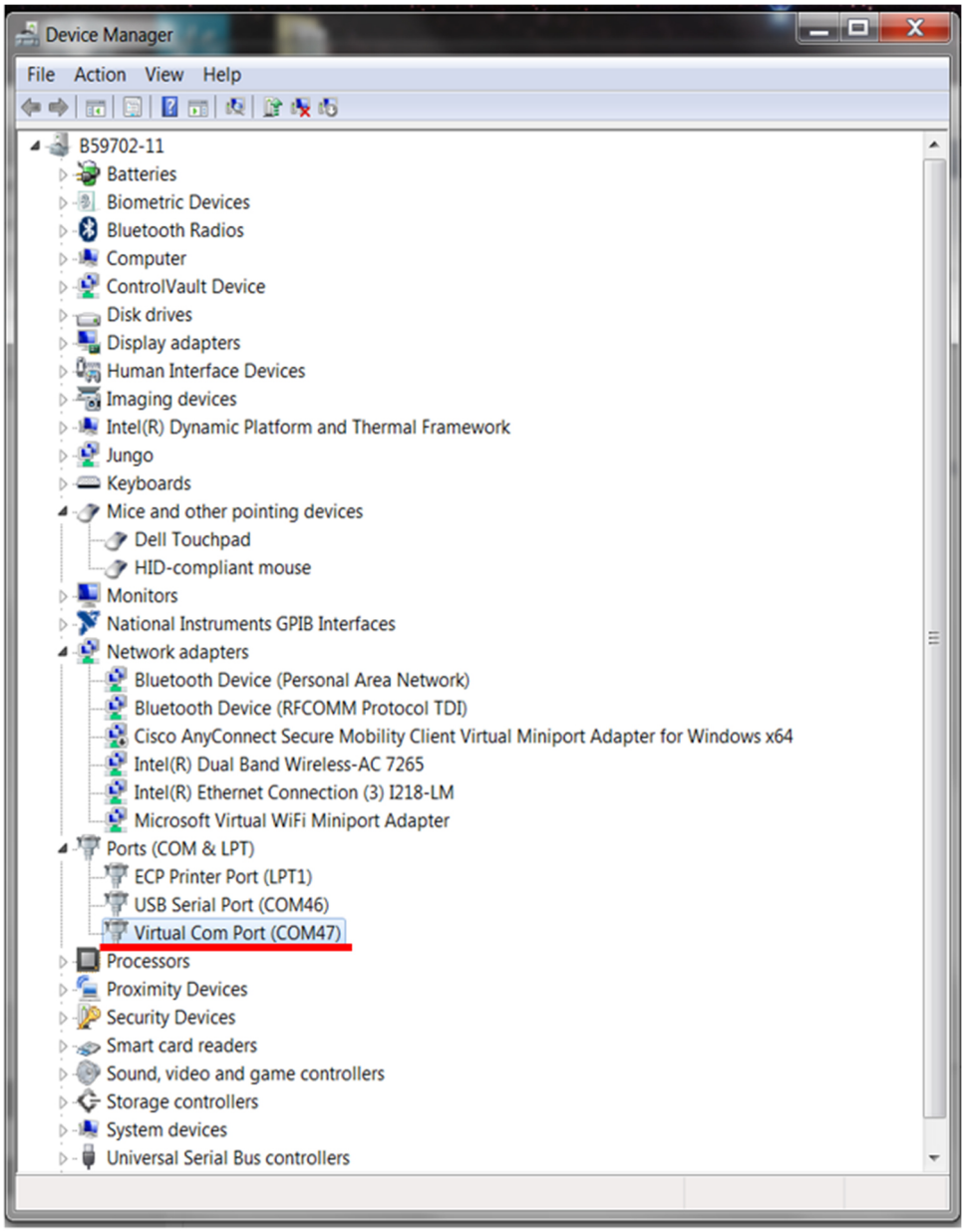
aaa-031992

14. In the **Virtual Com Port Properties** window, verify that the device is working properly, and then click **Close**.



aaa-031993

The Virtual Com Port appears in the Device Manager window.



aaa-031994

5.3 Installing FlexGUI software package

The FlexGUI software installation requires only extracting the zip file in a desired location.

1. If necessary, install the Java JRE and Windows 7 FlexGUI driver.
2. Download the latest FlexGUI (32-bit or 64-bit) version, available at http://www.nxp.com/KITFS84AUTEVM.
3. Extract all the files to a desired location on your PC.
   FlexGUI is started by running the batch file, *\bin\flexgui-app-spm.bat*.

6 Configuring the hardware for startup

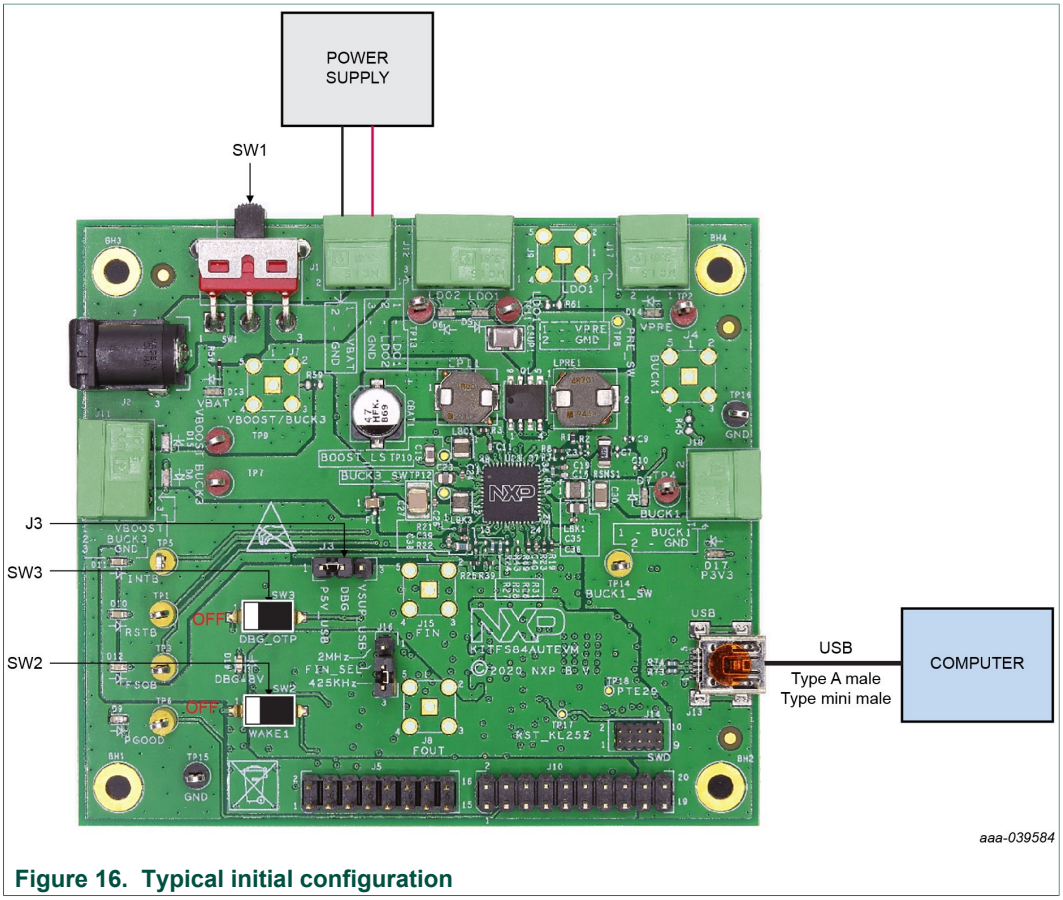


Figure 16. Typical initial configuration

Figure 16 presents a typical hardware configuration incorporating the development board, power supply and Windows PC workstation.

To configure the hardware and workstation as illustrated in Figure 16, complete the following procedure:

1. Install jumpers for the configuration.

Table 17. Jumper configuration

Jumper	Configuration
J3	connect 1-2 (connect 5.0 V on DBG pin from the USB)

2. Configure switches for the configuration

Table 18. Switch configuration

Switch	Configuration
SW1	middle position (VBAT off)
SW3	open (OTP programming Off)
SW2	open (WAKE1)

3. Connect the Windows PC USB port to the KITFS84AUTEVM development board using the provided USB 2.0 cable.

Set the DC power supply to 12 V and current limit to 1.0 A. With power turned Off, attach the DC power supply positive and negative output to KITFS84AUTEVM V_{BAT} Phoenix connector (J1).

4. Turn on the power supply.
5. Close SW2.

Note: *At this step, the product is in debug mode and all regulators are turned Off. The user can then power up with OTP configuration or configure the mirror registers before power up. Power up is effective as soon as J3 jumper is removed.*

7 Using the KITFS84AUTEVM evaluation board

This section summarizes the overall setup. Detailed description is provided in the following sections.

Before starting the process, choose the mode you want to run the device.

- In Normal mode, the configuration comes from OTP fuses.
- In Debug mode, you can either use the current configuration from OTP fuse, if any, or use the OTP emulation mode to write in the mirror register.

The Normal mode or Debug mode is defined at startup depending on the DBG pin level.

- Normal mode is set by tying DBG to ground.
- Debug mode is set by setting DBG voltage to 5.0 V.

In OTP emulation, you can overwrite the mirror registers from a given OTP fuse configuration. See Section 4.2.1 "OTP and mirrors registers" and Section 8.3 "Working with the Script editor" to define your configuration.

In OTP fuse configuration, use the configuration fused in the OTP. So, if a valid OTP fuse configuration exists, then it is copied to the mirror registers at startup.

7.1 Generating the OTP configuration file

Define and generate your OTP configuration using the excel file *FS84-QFN48EP-OTP.xlsm*. This file allows configuring the device for parameters controlled by the main state machine and the fail-safe state machine.

To generate the script:

1. Fill data in the **OTP_conf_main_reg** sheet.

MAIN OTP_REGISTERS											
Register Name	ADDRESS	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Data_Bin	Data_Hex
OTP_CFG_VPRE_1	14	-	-	VPREV[5:0]							
		0	0	001111 - 3.3V						00001111	0x0F
OTP_CFG_VPRE_2	15	-	-	VPRES[5:0]							
		0	0	000101 - 50mV/µs						00000101	0x05
OTP_CFG_VPRE_3	16	VPREILIM[1:0]		VPRETOFF[1:0]		VPRESRLS[1:0]		VPRESRHS[1:0]			
		00 - 50mV		10 - 40ns		11 - PU/PD/900mA		10 - PU/PD/520mA		00101110	0x2E
OTP_CFG_BOOST_1	17	-	-	VPRE_MODE	Reserved	VBSTV[3:0]					
		0	0	0 - Force PWM	0	0110 - 5.00V				00000110	0x06
OTP_CFG_BOOST_2	18	BOOSTEN	VBSTTONTIME[1:0]			VBSTSC[4:0]					
		1 - Enabled	00 - 60ns			01100 - 125mV/µs				10001100	0x8C
OTP_CFG_BOOST_3	19	VBSTRCOMP[1:0]		VBSTCCOMP[1:0]		VBSTILIM[1:0]		VBSTSR[1:0]			
		00 - 750kohms		00 - 125pF		01 - 2A		11 - 500V/µs		00000111	0x07
OTP_CFG_BUCK1_1	1A	VB1V[7:0]									
		01100000 - 1.0V								01100000	0x60
OTP_CFG_BUCK1_2	1B	otp_SPARE0[2:0]		VB1INDOPT[1:0]		VB1SWILIM[1:0]		-			
		000		00 - 1µH		01 - 2.6A		0		00000010	0x02
OTP_CFG_BUCK3_1	1E	BUCK3EN	VB3INDOPT[1:0]			VB3V[4:0]					
		1 - Enabled	00 - 1µH			01000 - 1.8V				10001000	0x88
OTP_CFG_BUCK3_2	1F	-				VB1GMCOMP[2:0]		VB3SWILIM[1:0]			
		000				010 - 32.5 GM		01 - 2.6A		00001001	0x09
OTP_CFG_LDO	20	LDO2ILIM	LDO2V[2:0]			LDO1LIM	LDO1V[2:0]				
		0 - 400mA	110 - 3.3V			0 - 400mA	110 - 3.3V			01100110	0x66
OTP_CFG_SEQ_1	21	-	-	-		VB1S[2:0]					
		0	0	000		010 - Regulator Start and Stop in Slot 2				00000010	0x02
OTP_CFG_SEQ_2	22	-	-	LDO2S[2:0]		LDO1S[2:0]					
		0	0	111 - Regulator does not Start (Enabled by SPI)		111 - Regulator does not Start (Enabled by SPI)				00111111	0x3F
OTP_CFG_SEQ_3	23	DVS_BUCK1[1:0]		DVS_BUCK3[1:0]		Tslot	VB3S[2:0]				
		00 - 7.81mV/µs		00 - 10.41mV/µs		0 - 250µs	000 - Regulator Start and Stop in Slot 0			00000000	0x00
OTP_CFG_CLOCK_1	24	-	-	VPRE_ph[2:0]			CLK_DIV2[2:0]				
		0	0	000 - delay 0			100 - divide by 44 - CLK2=455KHz			00000100	0x04
OTP_CFG_CLOCK_2	25	-	-	BUCK1_ph[2:0]			VBST_ph[2:0]				
		0	0	110 - delay 6			000 - delay 0			00110000	0x30
OTP_CFG_CLOCK_3	26	-	-	BUCK3_ph[2:0]			-				
		0	0	000 - delay 0			000			00000000	0x00
OTP_CFG_CLOCK_4	27	BUCK3_clk_sel	-	BUCK1_clk_sel	VBST_clk_sel	VPRE_clk_sel	PLL_sel	CLK_DIV1[1:0]			
		0 - CLK1	0	0 - CLK1	0 - CLK1	1 - CLK2	0 - Disabled	10 - divide by 9 - CLK1=2.22MHz		00001010	0x0A
OTP_CFG_SM_1	28	-	-	conf_txd[5:0]							
		0	0	0 - BOOST Shutdown	0 - BUCK1 Shutdown	0	0 - BUCK3 Shutdown	0 - LDO1 Shutdown	0 - LDO2 Shutdown	00000000	0x00
OTP_CFG_SM_2	29	otp_SPARE1[2:0			VPRE_off_dly	Autoretry_Infinite	Autoretry_en	PSYNC_CFG	PSYNC_EN		
		000			0 - 250µs	0 - Disabled	1 - Enabled	FS84_QFN48EP and 1x ext. f	1 - Enabled	00000111	0x07
OTP_CFG_VSUP_UV	2A	otp_SPARE2[6:0]							VSUPCFG		
		0000000							0 - 4.9V for Vpre < 4.5V	00000000	0x00
OTP_CFG_DV	2C	-	-	-	-	-	VDDIO_REG_ASSIGN[2:0]				
		0	0	0	0	0	001 - VPRE			00000001	0x01
OTP_CFG_DEVID	2D	DeviceID[7:0]									
		00000001								00000001	0x01
OTP_M_S1_CRC_LSB	2E	OTP_M_S1_CRC_LSB[7:0]									
		Automatically filled in by Sidence IP								00000000	0x00
OTP_M_S1_CRC_MSB	2F	OTP_M_S1_CRC_MSB[7:0]									
		Automatically filled in by Sidence IP								00000000	0x00

Figure 17. OTP_conf_main_reg spreadsheet example

2. Fill data in the **OTP_conf_failsafe_reg** sheet.

Register Name	ADDRESS	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	Data_Bin	Data_Hex
OTP_CFG_UVOV_1	0A	VCORE_V[7:0]									
		01100000 - 1.0V								01100000	0x60
OTP_CFG_UVOV_2	0B	VDDIOOVTH[3:0]				VCOREOVTH[3:0]					
		1011 - 110%				1011 - 110%				10111011	0xBB
OTP_CFG_UVOV_3	0C	-	-	VDDIO_V	VCORE_SVS_CLAMP[4:0]						
		0	0	0 - 3.3V	00000 - No SVS					00000000	0x00
OTP_CFG_UVOV_4	0D	VMON2OVTH[3:0]				VMON1OVTH[3:0]					
		1011 - 110%				1011 - 110%				10111011	0xBB
OTP_CFG_UVOV_5	0E	VMON4OVTH[3:0]				VMON3OVTH[3:0]					
		1011 - 110%				1011 - 110%				10111011	0xBB
OTP_CFG_UVOV_6	0F	VDDIOUVTH[3:0]				VCOREUVTH[3:0]					
		1011 - 90%				1011 - 90%				10111011	0xBB
OTP_CFG_UVOV_7	10	VMON2UVTH[3:0]				VMON1UVTH[3:0]					
		1011 - 90%				1011 - 90%				10111011	0xBB
OTP_CFG_UVOV_8	11	VMON4UVTH[3:0]				VMON3UVTH[3:0]					
		1011 - 90%				1011 - 90%				10111011	0xBB
OTP_CFG_PGOOD	12	-	PGOOD_RSTB	PGOOD_VMON4	PGOOD_VMON3	PGOOD_VMON2	PGOOD_VMON1	PGOOD_VDDIO	PGOOD_VCORE		
		0	0 - Not assigned	1 - Assigned	1 - Assigned	1 - Assigned	1 - Assigned	1 - Assigned	1 - Assigned	00111111	0x3F
OTP_CFG_ABIST1	13	otp_SPARE0[1:0]		ABIST1_VMON4	ABIST1_VMON3	ABIST1_VMON2	ABIST1_VMON1	ABIST1_VDDIO	ABIST1_VCORE		
		00		1 - Assigned	1 - Assigned	1 - Assigned	1 - Assigned	1 - Assigned	1 - Assigned	00111111	0x3F
OTP_CFG_ASIL	14	WD_DIS	WD_Selection	-	FCCU_EN	VMON4_EN	VMON3_EN	VMON2_EN	VMON1_EN		
		0 - Enabled	0 - Simple WD	0	1 - Enabled	1 - Enabled	1 - Enabled	1 - Enabled	1 - Enabled	00011111	0x1F
OTP_CFG_I2C	15	otp_SPARE1[2:0]			FLT_RECOVERY_EN	-					
		000			0 - Disabled	0000				00000000	0x00
OTP_CFG_DGLT_DUR_1	16	otp_SPARE2[1:0]		VCORE_UV_DGLT[1:0]		VCORE_OV_DGLT	VDDIO_UV_DGLT[1:0]		VDDIO_OV_DGLT		
		00		10 - 25µs		1 - 45µs	10 - 25µs		1 - 45µs	00101101	0x2D
OTP_CFG_DGLT_DUR_2	17	otp_SPARE3[4:0]					VMONx_UV_DGLT[1:0]		VMONx_OV_DGLT		
		00000					10 - 25µs		1 - 45µs	00000101	0x05
OTP_FS_S1_CRC_LSB	18	OTP_FS_S1_CRC_LSB[7:0]									
		Automatically filled in by Sidence IP								00000000	0x00
OTP_FS_S1_CRC_MSB	19	OTP_FS_S1_CRC_MSB[7:0]									
		Automatically filled in by Sidence IP								00000000	0x00

Figure 18. OTP_conf_failsafe_reg spreadsheet example

3. See the **OTP_conf_summary** sheet to review the complete configuration (main and fail-safe).

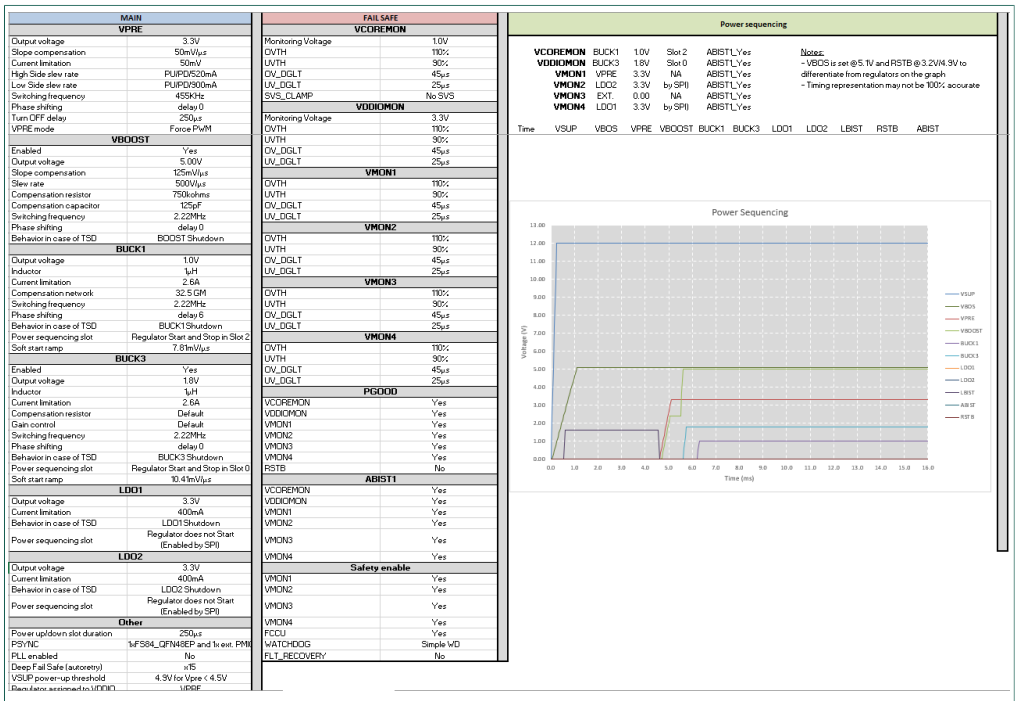


Figure 19. OTP_configuration_summary_example

4. Generate script in the **OTP_conf_file_generation** sheet.
   Once the configuration is ready, the user can generate the script file. Go to **OTP_conf_file_generation**, enter the path in the **File repository**, and then click **Write_OTP_File_GUI**.

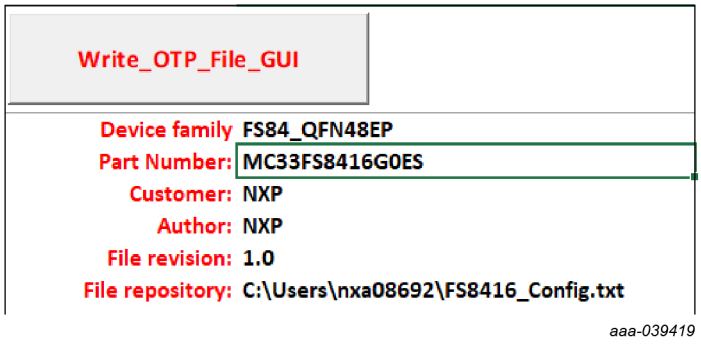


Figure 20. OTP script generation

7.2 Working in OTP emulation mode

At startup, the device always uses the content from the mirror register. This content can come from OTP fuse or from configuration written directly in the mirror register. OTP emulation means that the user can emulate the OTP writing in the mirror register. This allows trials before burning the OTP.

1. Configure the hardware. See Section 6 "Configuring the hardware for startup".
2. Launch the FlexGUI software.
3. Switch to Debug mode:
   a. Place SW1 in the right direction (VBAT switched On).

b. Close SW2 (WAKE1).

While in Debug mode, all regulators are turned Off.

4. Load the mirror registers to work in OTP emulation mode. See Section 8.3 "Working with the Script editor".
5. Unplug jumper J3 1-2 to start the device with the mirror configuration setting.
   a. If the mirror registers are filled (with a configuration using the Script editor), that configuration is used in the emulation session.
   b. If the mirror registers are not filled (with a configuration using the Script editor), the currently-programmed OTP fuse configuration is used, if it exists.
   c. Otherwise, the mirror registers are not filled and the OTP fuse is not burned, and the device does not start up.

   As long as initialization phase is not closed by a first good WD_Answer, the WD does not start and regulators do not stay alive. Also, as long as Debug mode is not exited by writing FS_STATES:[DBG_EXIT] bit to 1, the FS0B pin cannot be released.
6. Use the FlexGUI software to evaluate the device configured. See Section 8 "Using FlexGUI".

7.2.1 Example script: closing initialization phase, disabling FCCU monitoring and releasing FS0B

The following script can be used to:

- Disable the WD.
- Disable the FCCU monitoring.
  On the hardware kit, the FCCU1 is pulled to GND and FCCU2 is pulled to VDDIO, which is detected as error phase by default. Disabling the FCCU by SPI avoids safety issue at startup.
- Close the initialization phase.
- Exit the Debug mode.
- Release FS0B pin. This is valid only if WD is activated in OTP.
  Seven good consecutive WD answers are required to have the FLT_ERR_CNTR back to 0. This is one of the conditions to allow FS0B release.

Table 19. FS8416 starting sequence example

Step	Register name	Value	Description
1	FS_WD_WINDOW	0x0200	WDW_WINDOWS[3:0] = 0x0 => Watchdog disabled
2	FS_NOT_WD_WINDOW	0xF50F	NOT of FS_WD_WINDOW
3	FS_I_SAFE_INPUTS	0x51C6	FCCU_CFG[1:0] = 0x0 => 0x1 => Monitoring by pair FCCU12_FLT_POL[0] = 1 => FCCU1 or 2 = 0 is a fault
4	FS_I_NOT_SAFE_INPUTS	0xAC18	NOT of FS_I_SAFE_INPUTS
5	FS_WD_ANSWER	0x5AB2	1st good WD answer (for simple WD selection in OTP) Close the initialization phase
6	FS_STATES	0x4000	DBG_EXIT[0]=1 => Exit Debug mode
7	FS_WD_ANSWER	0x5AB2	2nd good WD answer
8	FS_WD_ANSWER	0x5AB2	3rd good WD answer
9	FS_WD_ANSWER	0x5AB2	4th good WD answer
10	FS_WD_ANSWER	0x5AB2	5th good WD answer
11	FS_WD_ANSWER	0x5AB2	6th good WD answer
12	FS_WD_ANSWER	0x5AB2	7th good WD answer
13	FS_RELEASE_FS0B	0xB2A5	FS0B pin released (pulled to high level)
14	MFLAG2	0x40F1	Clear flags VSUPUV7; VPREUVL, VSUPUVL, WAKE1FLG
15	FS_OVUVREG_STATUS	0x4550	Clear UV status flags

This sequence can be sent using a script built with FlexGUI. See Section 8.3.2 "Script sequence files".

7.3 Programming the device with an OTP configuration

The device configuration can be changed three times (see Section 4.2.1 "OTP and mirrors registers"). The programming steps are exactly the same as the OTP emulation mode up to step 6.

Then, the user has to burn the part with FlexGUI. See Section 8.4.8 "OTP programming". Follow the instructions on the screen to proceed.

8 Using FlexGUI

To follow the steps in this section, make sure that the board is connected using the appropriate hardware configuration (see Section 7.2 "Working in OTP emulation mode").

Note: *It is recommended to use the latest version of FlexGUI.*

8.1 Starting the FlexGUI application

After FlexGUI is launched with the *flexgui-app-spm-fs8416.exe* file, the FlexGUI launcher displays available kits.

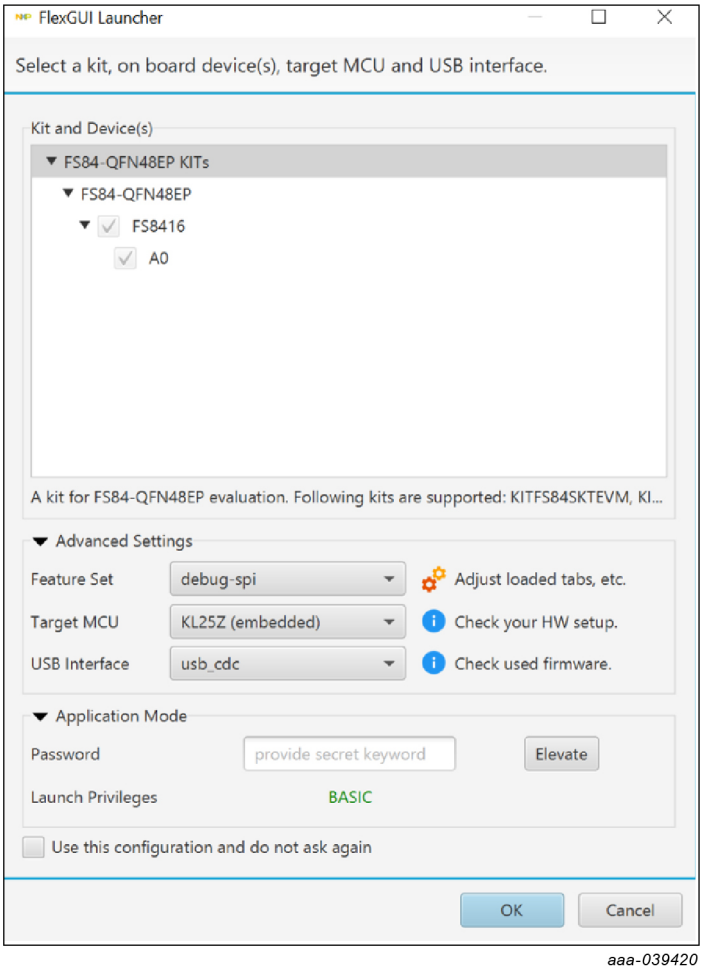


Figure 21. Launcher panel - bus selection

When the configuration is selected, click **OK**.

8.2 Establishing the connection between FlexGUI and the hardware

The board must be connected to the USB before establishing a connection.

- Click **Search** to detect the COM port of the board.
- Click **Start** to enable the connection.

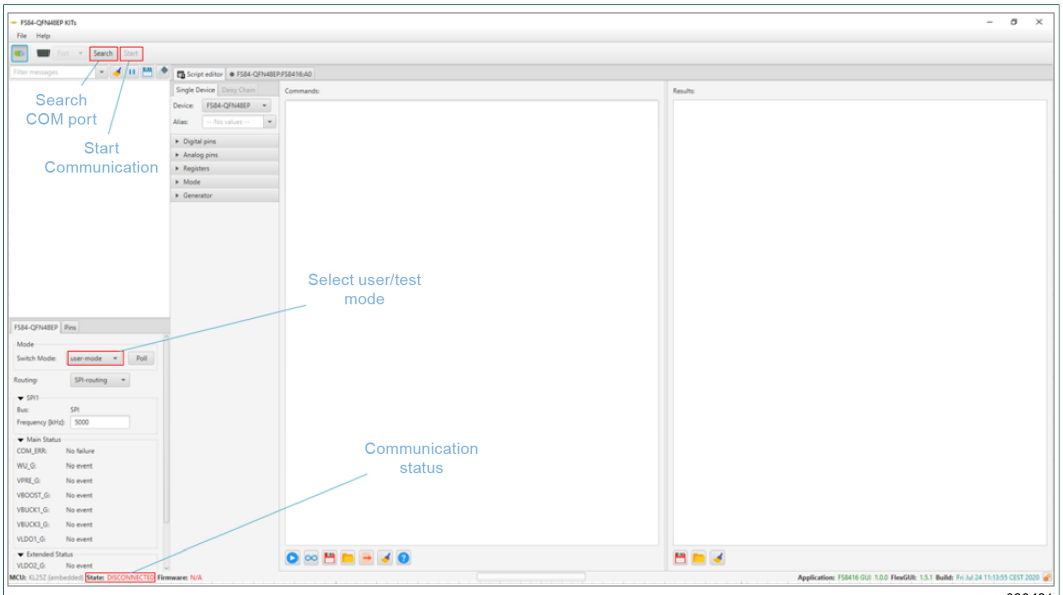


Figure 22. Main panel

Figure 22 shows the mode selection. At first launch, the FlexGUI starts in User mode. The user can then decide to switch to Test mode using the Switch mode drop-down list followed by clicking **Apply**.

The **GUI-Device Status** field checks the connection from MCU to the device. The **ONLINE** status indicates a good connection, while **ERROR** status indicates an issue (e.g. V_{SUP} is not provided to the device).

It is also possible to change the clock frequency using this panel.

The user can read in which mode the device is operating. It is also possible to switch from user mode to test mode (and vice-versa).

The current mode is refreshed only when Poll button is activated. If required, this has to be done at start up (Poll button is disabled by default). See Figure 23.

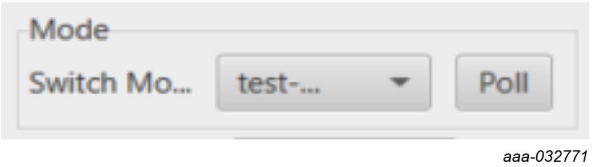


Figure 23. Disabling device mode polling

To move from one mode to the other, select the mode with switch mode drop-down button. If the requested mode is not confirmed by the device (if debug pin is not set, for instance), the drop-down menu switches back to the previous mode.

8.3 Working with the Script editor

The register and OTP emulation can be configured with the script editor. This is particularly useful to try various OTP configurations in Emulation mode.

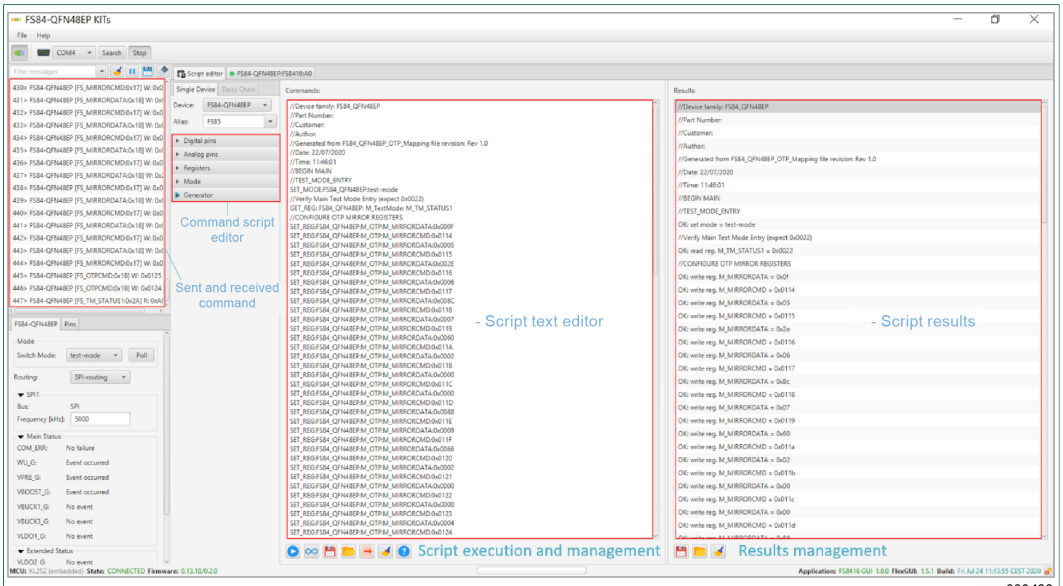


Figure 24. Script Editor

The main subareas of this panel are:

- **Send and receive command**: displays a summary of commands sent and received from the device
- **Command script editor**: builds commands to be sent to the device
- **Script text editor**: sends a sequence of register configurations from a text file or from command edited directly in this area
- **Script results**: displays result status of each command sent to the device

8.3.1 Script text editor

Using Script editor, you can execute any command either directly or from a file. It is also possible to save and modify a script. Using the brush symbol, it is possible to clean windows if needed.

All commands have to follow a specific syntax. The Help menu describes commands available in the script editor and their syntax.

This help page describes commands available in the script editor and their format.

List of commands

- **SET_REG**: sets value of a selected register.
- **READ_REG**: reads value of a selected register.
- **SET_DPIN**: sets value of a selected digital pin.
- **GET_DPIN**: gets value of a selected digital pin.
- **GET_APIN**: gets value of a selected analog pin. Returned value is in mV.
- **PAUSE**: shows a dialog with user defined message. The script is paused until the user cofirms the dialog.
- **EXIT**: stops execution of the script.
- **SET_MODE**: sets device mode. List of modes depends on a device.

Command format

The following table describes command parameters. All paramaters are mandatory.

	1st parameter	2nd parameter	3rd parameter	4th parameter	5th parameter
SET_REG	Device	Reg. set	Reg. name / Reg. address	Reg. value	-
GET_REG	Device	Reg. set	Reg. name / Reg. address	-	-
SET_DPIN	Device	Pin name	Dig. pin value	-	-
GET_DPIN	Device	Pin name	-	-	-
GET_APIN	Device	Pin name	-	-	-
PAUSE	Message	-	-	-	-
EXIT	-	-	-	-	-

Description of command parameters mentioned in the table above:

- **Device**: device name (alias used in application).
- **Reg. set**: register set name. Register sets allows to associate registers which have similar function.
- **Reg. name**: register name as defined in datasheet.
- **Reg. address**: register address in decimal or hexadecimal (with 0x prefix) format.
- **Reg. value**: register value in decimal or hexadecimal (with 0x prefix) format.
- **Pin name**: name of digital or analog pin as defined in device datasheet.
- **Dig. pin value**: value of digital pin. Allowed strings are 'low' and 'high'.
- **Message**: a message to be displayed in a dialog. It cannot contain ':' character, which is used as delimiter of parameters.
- **Mode**: name of a device mode.

Figure 25 shows an example to build a command from the panel.

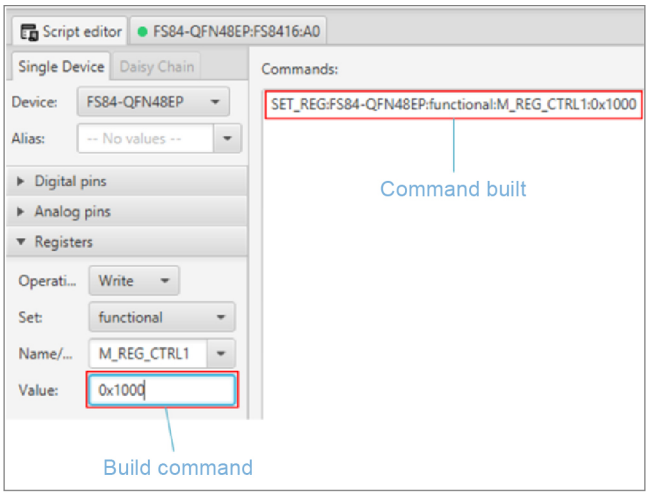


Figure 25. Build a command

The value 0x1000 is sent to the register M_REG_CTRL1 (BUCK1DIS). The user can then send it to the device by clicking the arrow (see Figure 26).

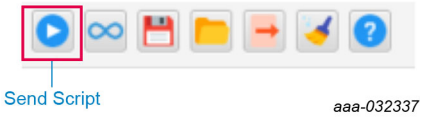


Figure 26. Send script

```
Commands:
//This command will disable
//Regulator Buck1
SET_REG:FS84-QFN48EP:functional:M_REG_CTRL1:0x1000
```

Figure 27. Correct format

```
Commands:
//This command will disable
Regulator Buck1
SET_REG:FS84-QFN48EP:functional:M_REG_CTRL1:0x1000
```

Figure 28. Wrong format (“//” missing in second line)

8.3.2 Script sequence files

The Script editor allows the user to save script sequence files. A script sequence file is text file that contains a set of commands sent to the device in the order they are written, as shown in the following example.

```
// FS84_Release_FS0b
SET_REG:FS84-QFN48EP:safety:FS_WD_WINDOW:0x0200
SET_REG:FS84-QFN48EP:safety:FS_NOT_WD_WINDOW:0xF50F
SET_REG:FS84-QFN48EP:Write_INIT_Safety:FS_I_SAFE_INPUTS:0x51C6
SET_REG:FS84-
QFN48EP:Write_INIT_Safety:FS_I_NOT_SAFE_INPUTS:AC18
SET_REG:FS84-QFN48EP:safety:FS_WD_ANSWER:0x5AB2
SET_REG:FS84-QFN48EP:safety:FS_STATES:0x4000
SET_REG:FS84-QFN48EP:safety:FS_WD_ANSWER:0x5AB2
SET_REG:FS84-QFN48EP:safety:FS_WD_ANSWER:0x5AB2
SET_REG:FS84-QFN48EP:safety:FS_WD_ANSWER:0x5AB2
SET_REG:FS84-QFN48EP:safety:FS_WD_ANSWER:0x5AB2
SET_REG:FS84-QFN48EP:safety:FS_WD_ANSWER:0x5AB2
SET_REG:FS84-QFN48EP:safety:FS_WD_ANSWER:0x5AB2
SET_REG:FS84-QFN48EP:safety:FS_RELEASE_FS0B:0xB2A5
```

Note: *Comments can be added with a // prefix.*

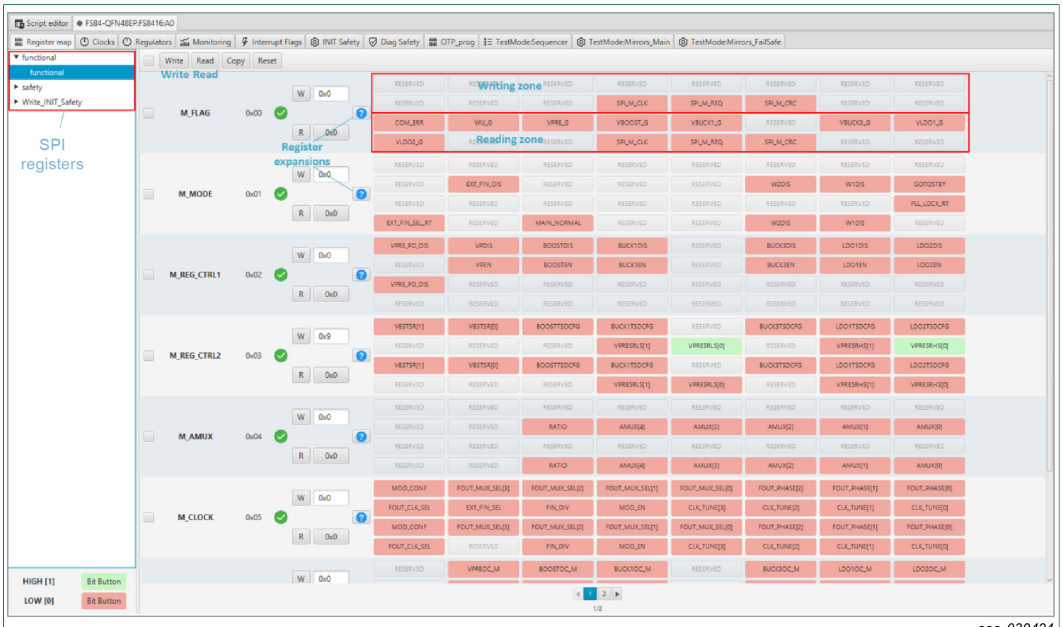
8.4 Understanding the FS84 QFN48EP workspace

The FS84 QFN48EP workspace consists of several tabs, each dedicated to a specific aspect of device functionality or configuration.

- Register map
- Clocks
- Regulators
- Measurements
- Interrupt flags
- INIT safety
- Diag safety
- OTP programming
- TestMode:Sequencer
- TestMode:Mirrors_Main and TestMode:Mirrors_Failsafe

8.4.1 Register map

All SPI registers can be accessed in write and read mode using this tab.

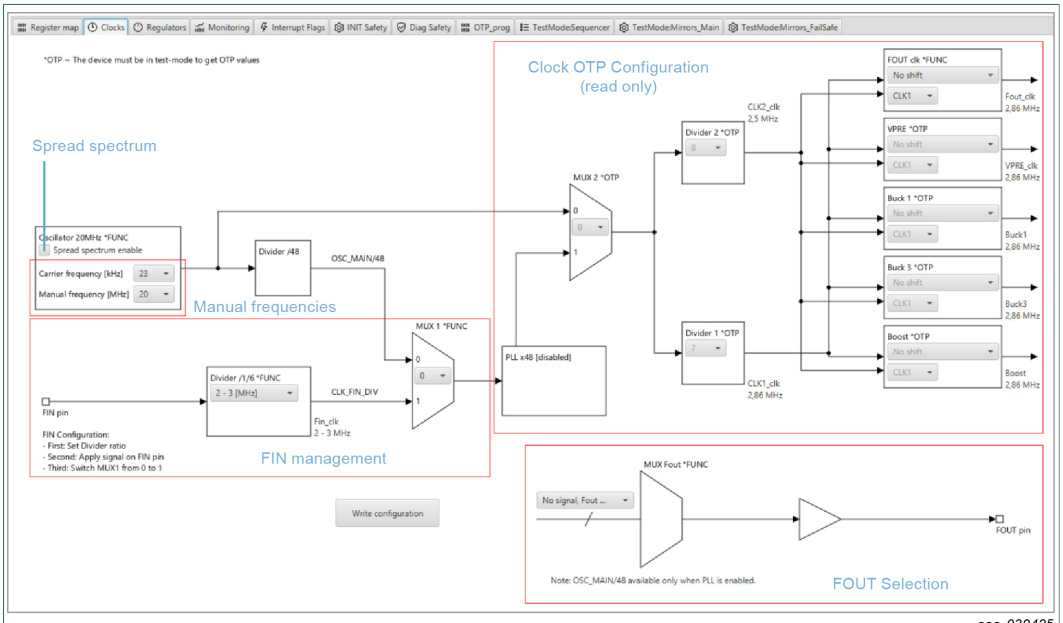


aaa-039424

Figure 29. Register map

- **Register map**: allows access to functional register, safety register and write init register which are accessible only during initialization phase
- **Read**: allows you to read any register either individually or by bank
- **Write**: allows you to write any register either individually or by bank
- **Register expansion**: displays the value of each device parameter

8.4.2 Clocks



aaa-039425

Figure 30. Clocks

This tab allows:

OTP:

- Read current OTP configuration (write operation is not possible). To display the accurate data, the device needs to operate in Test mode.

SPI:

- Configure the device to work with FIN input
- Select the signal to apply on FOUT pin
- Play with manual frequencies and spread spectrum

8.4.3 Regulators

The regulator has two main areas:

- Low voltage (LV) regulators configuration
- VPRE compensation network calculation

Each regulator can either be enabled or disabled by SPI. The thermal shutdown behavior can be configured to either shutdown the regulator, or shutdown the regulator and transition to deep fail-safe. The write button applies to the entire table. The VPRE compensation network calculator helps to define the value for VPRE external compensation network.

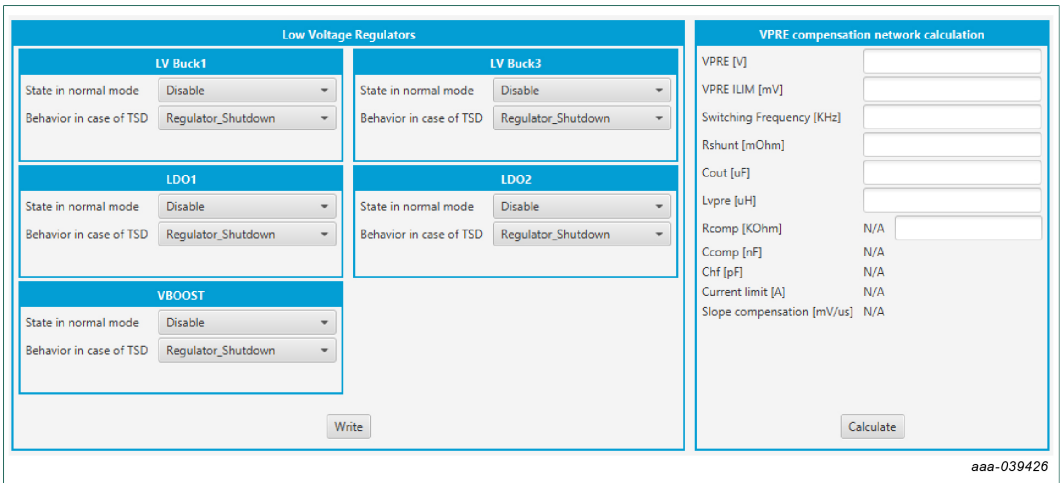
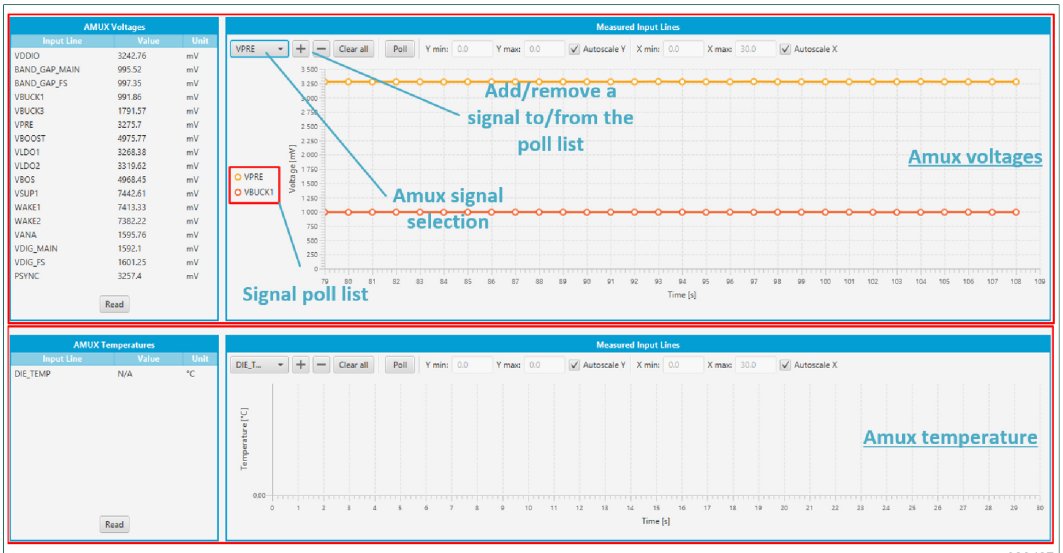


Figure 31. Regulators

8.4.4 Measurements

This tab enables two features:

- Read any of the AMUX signals over time
- Display regulator voltage summary

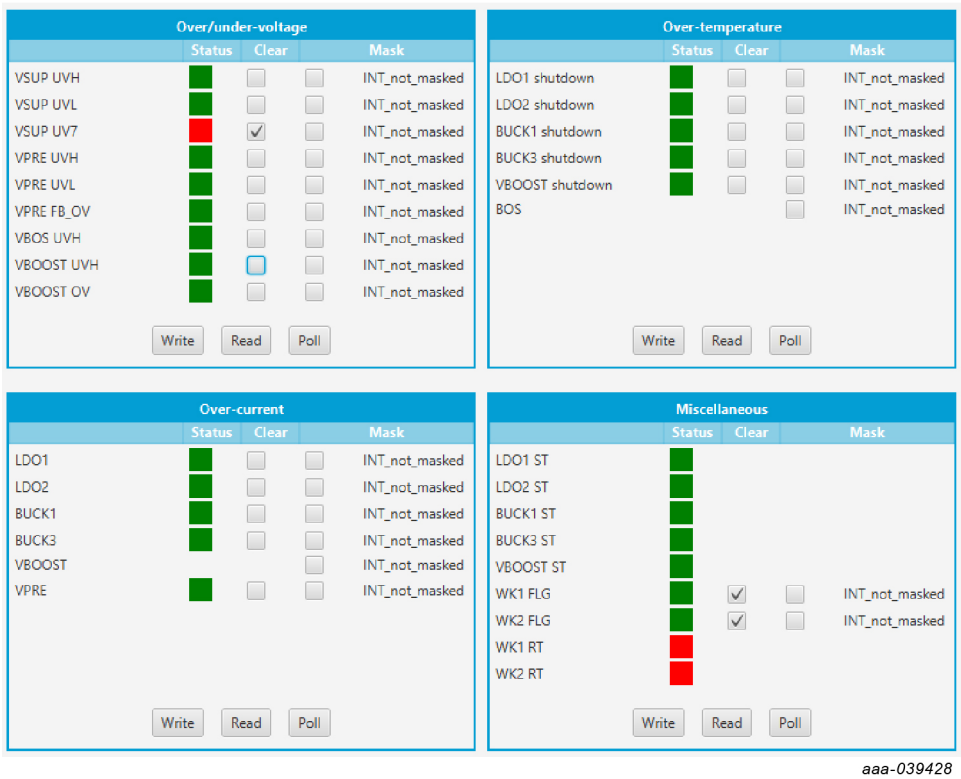


aaa-039427

Figure 32. Measurements

8.4.5 Interrupt flags

This tab allows you to set or clear flags. It is also possible to mask the interruption.



aaa-039428

Figure 33. Interrupt flags

8.4.6 INIT safety

This tab allows you to manage all registers that can be configured to close the initialization phase. The initialization phase is closed by the first good watchdog refresh before 256 ms timeout.

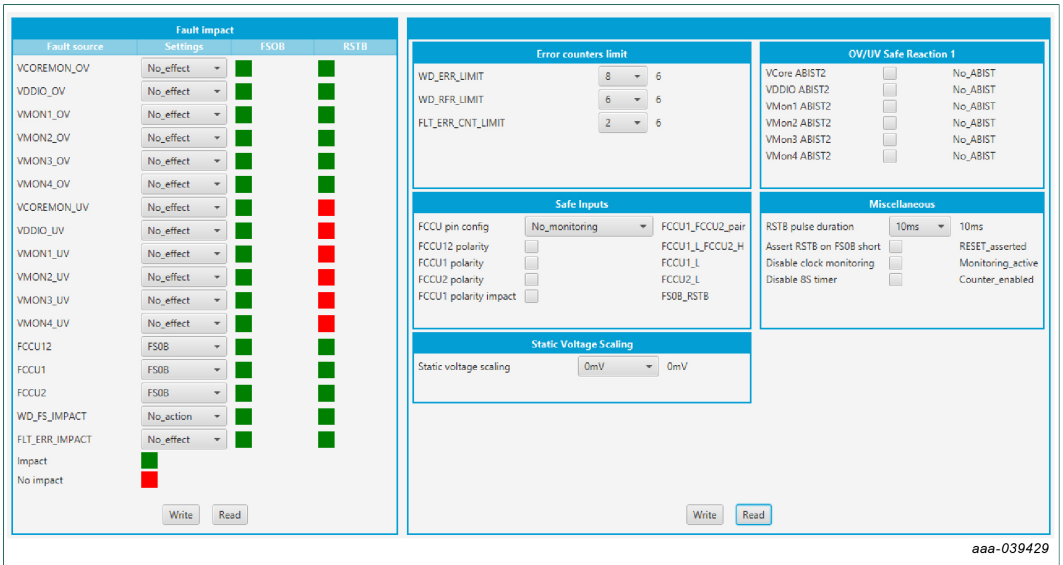


Figure 34. INIT safety

8.4.7 Diag safety

The watchdog type configured in the OTP has to be manually selected in the drop-down list to play with the watchdog features. If the user is not aware about the type of watchdog configured in the OTP, it can be found in TestMode:Mirrors_Failsafe and Miscellaneous tabs.

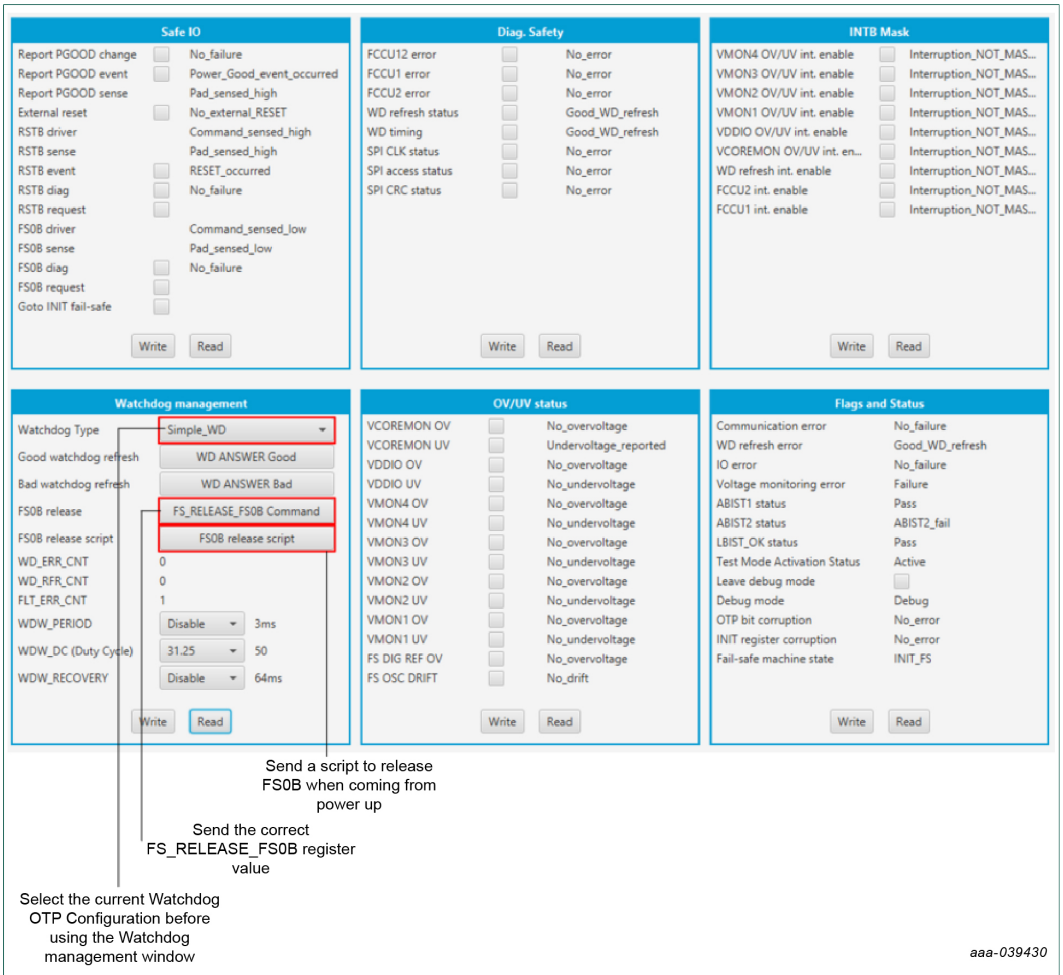


Figure 35. Diag safety

The FS_Release_FS0B command calculates and sends the right secure16-bit word to release FS0B.

A simplified way to release FS0B after power up is to hit FS0B Release script button. This sends the right sequence to close the initialization sequence, sets the error counter back to 0, then releases FS0B.

8.4.8 OTP programming

This tab allows you to burn the OTP using a script generated by the excel file OTP configuration (see Section 7.1 "Generating the OTP configuration file ").

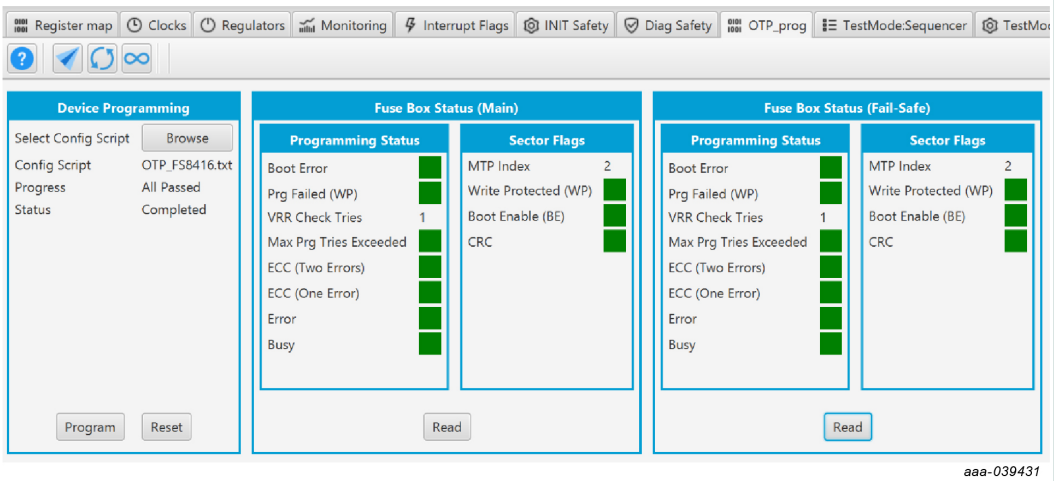


Figure 36. OTP burning

To set up the hardware before OTP burning, see Section 7.3 "Programming the device with an OTP configuration".

See Figure 36 and follow the steps:

- Browse and load the script file you want to burn. The program button is then available.
- Click **Program**.

FlexGUI pops up to turn the 8.0 V On, and then turns Off. Note that the blue LED on the board indicates that an 8.0 V voltage is available on the Debug pin. This voltage is used only during the burning process, and should not be applied in any other configuration. At the end of the first OTP programming, the MTP index = 1, WP, BE and CRC flags are green.

The Sector Flags area provides status and Table 20 provides the state of main flags after a read. This helps to determine how many times the part was burned.

Table 20. OTP burning flag status

OTP burning step	BE	WP	CRC	MTP Index
OTP not burn Mirrors empty	Red	Red	Red	1
OTP not burn Mirrors filled	Red	Red	Green	1
1	Green	Green	Green	1
2	Green	Green	Green	2
3	Green	Green	Green	3

Example shown in Figure 36 corresponds to the OTP burning step 2 from Table 20.

To check if a valid OTP configuration is already burned, switch V_{BAT} Off, then On, and start the device. The device starts with the OTP configuration.

8.4.9 TestMode:Sequencer

The sequencer allows you to display the slot configuration for the device. To be able to access this tab, the device has to be in Test mode. The configuration is read from mirror register. It is possible to modify it and update the mirror register.

As an example, the slot sequence is filled at start up with the content of OTP fuses. Then the user can decide to modify any of the configurations coming from the OTP fuse. Note that all these actions are done with Debug pin at 5.0 V and in test mode.

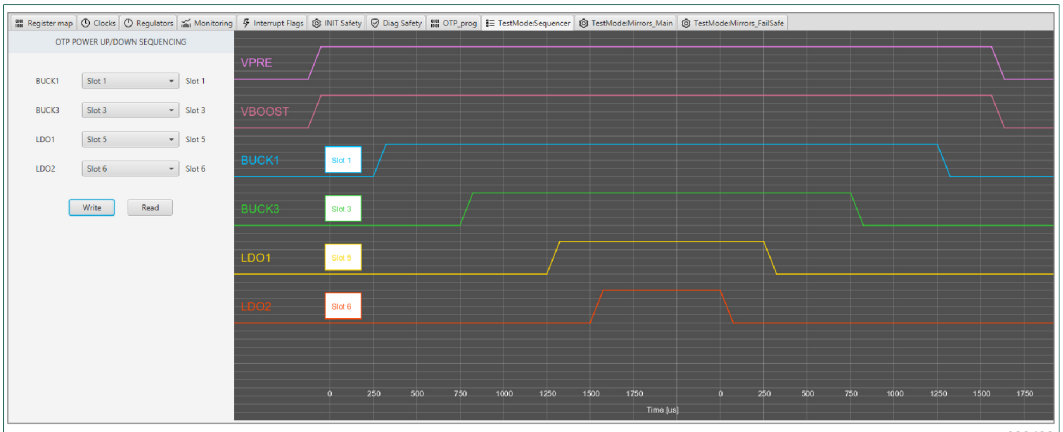


Figure 37. TestMode:Sequencer

Use the drop-down button (see Figure 38) to select the appropriate slot. The selection configuration can be sent to the device by clicking Write button. The current status can be read by using Read button.

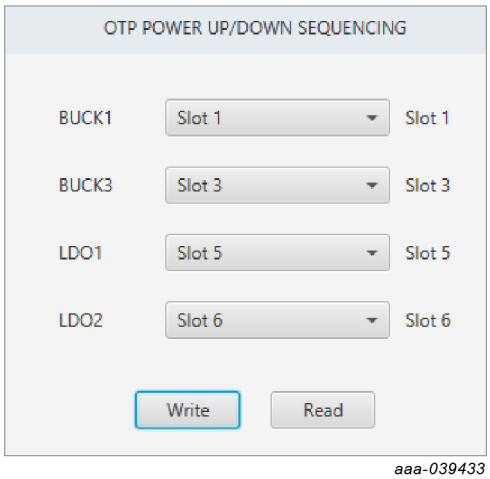


Figure 38. Slot management

8.4.10 TestMode:Mirrors_Main and TestMode:Mirrors_Failsafe

The TestModeMirrors_Main and TestModeMirrors_FailSafe tabs allow access to the OTP main mirrors and fail-safe registers. These tabs are available in Test mode.

In this panel, some parameters are highlighted in red. The red indicates that these parameters are not considered in case of emulation mode (accessible only in debug mode). **The user must rewrite by SPI after startup.**

This concerns only:

- VPRE and VBOOST slew rate
- All regulator behavior in case of TSD

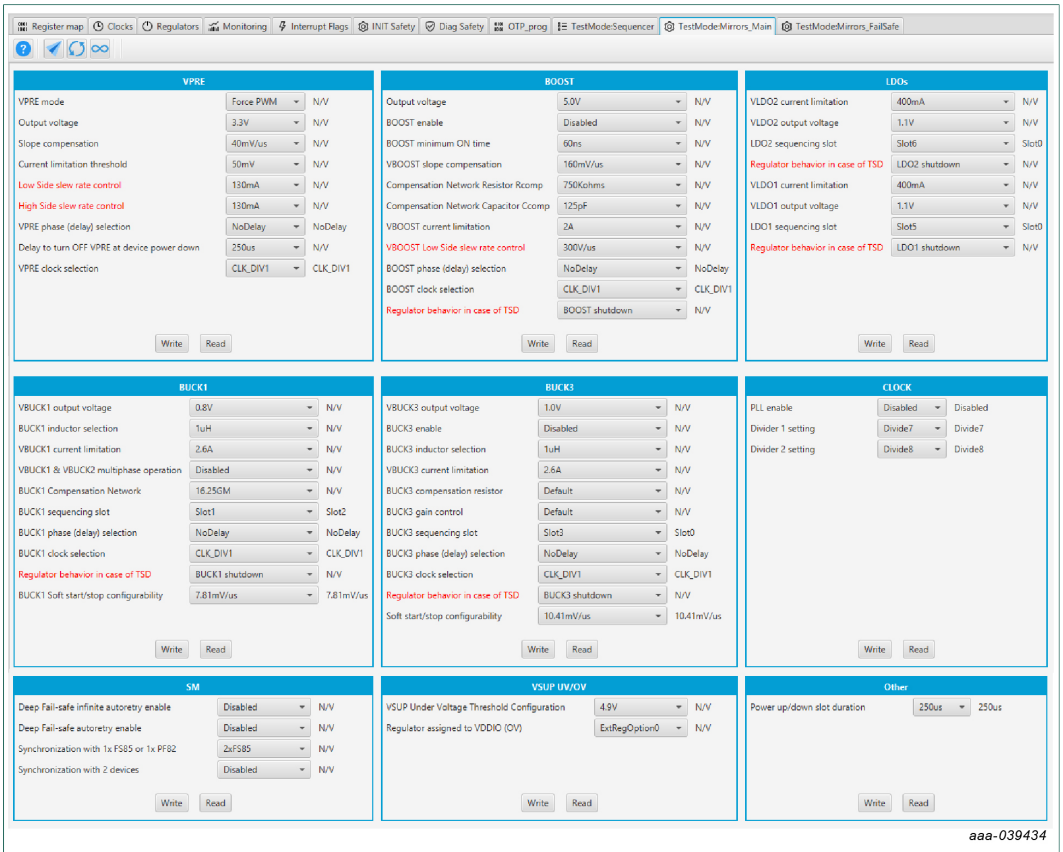


Figure 39. TestMode: Mirrors_Main

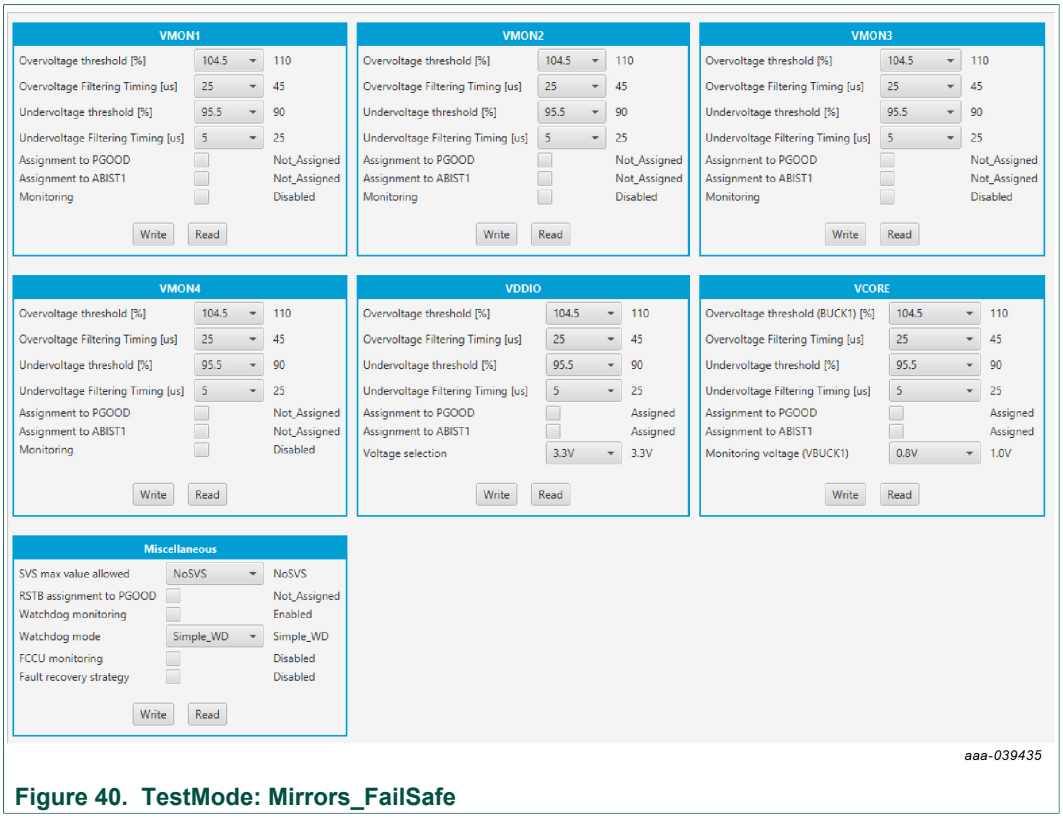


Figure 40. TestMode: Mirrors_FailSafe

The Read button provides the current status. The Write button changes the configuration in mirror register. This can be useful, for example, to modify few parameters from OTP fuse to start up the board.

9 References

[1] **KITFS84AUTEVM** — detailed information on this board, including documentation, downloads, and software and tools http://www.nxp.com/KITFS84AUTEVM

[2] **FS8400** — product information on FS8400, Safety system basis chip for S32 microcontroller, fit for ASIL B http://www.nxp.com/FS8400

[3] **FS84-QFN48EP-OTP.xlsm** — OTP configuration file

10 Revision history

Revision history

Rev	Date	Description
v.1	20201028	• Initial version

11 Legal information

11.1 Definitions

Draft — The document is a draft version only. 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 herein and shall have no liability for the consequences of use of such information.

11.2 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.

Suitability for use in automotive applications — This NXP Semiconductors product has been qualified for use in automotive applications. Unless otherwise agreed in writing, the product is not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

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

11.3 Trademarks

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

NXP — is a trademark of NXP B.V.

SMARTMOS — is a trademark of NXP B.V.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 28 October 2020
Document identifier: UM11502