AN14492

FS26 software quick start guide

Rev. 1.0 — 10 January 2025 Application note

Document information

Information	Content
Keywords	FS2600, Safety, SBC, automotive, low power, ASIL B, ASIL D
Abstract	This application note is meant to be used as a launching point for software engineers, as a complement or a substitute for NXP’s software drivers.

1 Introduction

This application note is meant to be used as a launching point for software engineers, as a complement or a substitute for NXP's software drivers.

This document gives guidance on the implementation of the SPI communication protocol between the MCU and the FS26.

It also explains the initialization procedure for the FS26 device and provides an example startup sequence.

1.1 General description

This family of devices consists of several versions that are pin-to-pin and software compatible. These versions support a wide range of applications with automotive safety integrity levels (ASIL) B or D, offering choices in number of output rails, output voltage settings, operating frequencies, power-up sequencing, and integrated system-level features.

The flexibility of the FS26 system basis chip (SBC) makes it suitable for S32K3 processor-based applications, as well as multivendor processors.

Many OTP configurations are available, offering a choice of output voltage settings, operating frequency, power-up sequencing, and input/output configurations to address multiple applications.

1.2 Reference documents

Reference documents and various materials are available on the FS26 device web page. The webpage provides more detailed information about specific topics:

FS26 data sheet: Information, such as features, functional description, parametric description, register mapping.

FS26 Design Guidelines application note: Information such as application schematics, bill of materials, placement and layout guidelines, application validation data including ISO/non-ISO pulses, Electromagnetic Compatibility (EMC).

The low-level software driver components are provided as part of the basic enablement for the device, and do not incur an additional charge:

FS26 AUTOSAR software drivers: AUTOSAR and ISO 26262-compliant basic start-up drivers for low-level interfaces. Technical documentation is available as part of the software driver package, detailing supported features such as:

- SPI access register function and event handling (SBC_FS26)
- Watchdog function (WDG_FS26)

2 FS26 initialization flow chart example

Figure 1 gives an example of FS26 software initialization. After MCU reset is released (RSTB state goes high), the MCU can start FS26 initialization. The initialization must be done within the dedicated 256 ms INIT_FS window.

Running the LBIST is optional (skipped if LBIST_STDBY_OTP[7:0] = C9h) as it is only available for ASIL D FS26 versions. Then the MCU writes INIT safety registers, ending with the INIT cyclic redundancy check (CRC). The first watchdog refresh closes the INIT phase. Therefore, the subsequent watchdog refreshes must be sent according to the watchdog timing configuration. Once the fault error counter is cleared, safety pins FS0B and FS1B can be released.

An ABIST on demand (ABIST2) can be requested through a SPI command, to allow the system to check the integrity of the safety mechanisms at any point during the Normal mode and to detect potential latent faults when the application is running.

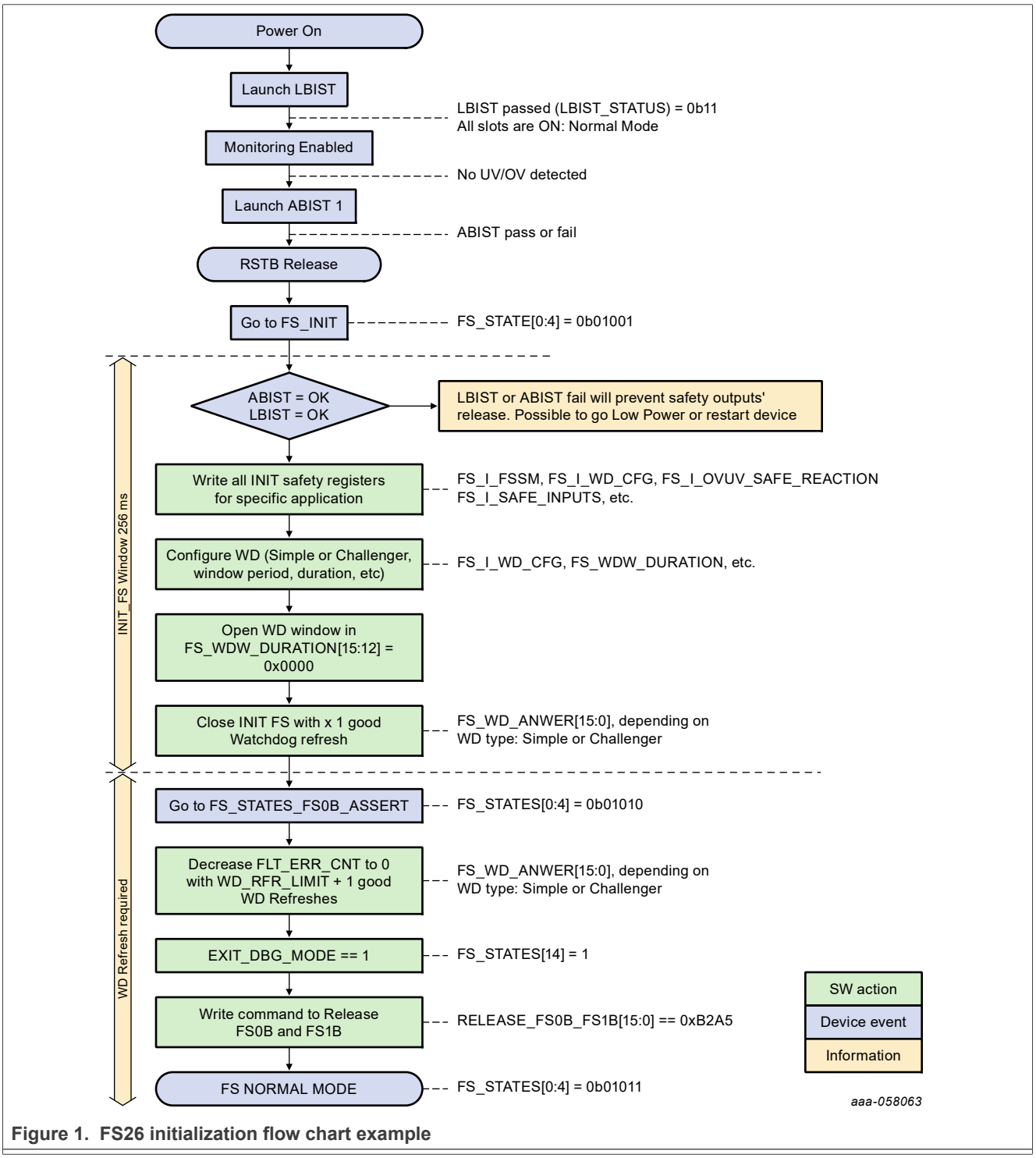

Figure 1. FS26 initialization flow chart example

3 Startup sequence example (based on flow chart)

Table 1. Startup SPI sequence example

		Register	Read	Write	Comment
1	**Launch LBIST**	LBIST_STDBY_ OTP[7:0](0x1F)		0x00 LBIST always performed C9h LBIST Bypass LBIST from standby mode	Applicable for ASIL D applications only:
2	**Check LBIST**	FS_DIAG_ SAFETY1[1:0] (0x54)	0b01 BYPASSED 0b10 FAIL 0b11 OK		LBIST_STATUS: (MSB = LBIST_ CHK_PAT_OK, LSB = LBIST_ CHECKER_OK)
3	**Monitor Enabled**	FS_GRL_ FLAGS[12] (0x40)	0 No Failure 1 Failure		FS_REG_OVUV_ G: Flags Reporting: VPRE_OV, VPRE_ UV, CORE_ OV, CORE_UV, LDO1_OV, LDO1_ UV, LDO2_OV, LDO2_UV, TRK1_ OV, TRK1_UV, TRK2_OV, TRK2_ UV, REF_OV, REF_UV, EXT_OV, EXT_UV
4	**Launch/Check ABIST**	FS_DIAG_ SAFETY1[8] (0x54)	0 ABIST1 Fail or not executed 1 ABIST1 Pass		Report ABIST2 status
5	**RSTB Release**	FS_STATES[4:0] (0x57)	0b01000 RSTB Release		Actual Sate of the Fail Safe State machine
6	**INIT_FS**	FS_STATES[4:0] (0x57)	0b01001 INIT_FS		Actual Sate of the Fail Safe State machine
7	**Configure Init Safety Registers**	FS_I_FSSM(0x49)		0x50c1	Configure error counter limits, error reaction, reset duration, etc
		FS_I_OVUV_ SAFE_ REACTION(0x42)		0x9999	Configure reaction of safety machine in case of under or over voltages on regulators.
		FS_I_SAFE_ INPUTS(0x47)		0x238d	Configure modes, polarity and reactions to FCCU and ERRMON

Table 1. Startup SPI sequence example...*continued*

		Register	Read	Write	Comment
8	**Configure Watchdog**	FS_I_WD_ CFG(0x45)		0x4200	Configure WD error counter limit, reaction to error, refresh counter, etc.
		FS_WDW_ DURATION(0x4B)		0x008B	Configure WD window duration
10	**Close INIT FS**	FS_WD_ ANWER[15:0](0x4 D)		0x5AB2	x1 good Watchdog refresh
11	**Go to FS_ STATES_FS0B_ ASSERT**	FS_STATES[0:4] (0x57)	0b01010.		You can verify the closing of INIT_FS state by reading
12	**Send 7x good WD refresh (if WD_ ERR_LIMIT = 6 and FLT_ERR_ CNT = 1)**	FS_WD_TOKEN (0x4E)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x4D)		0x5AB2	Watchdog answer is 0x5A2B (default value)
		FS_WD_TOKEN (0x4E)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x4D)		0x5AB2	Watchdog answer is 0x5A2B
		FS_WD_TOKEN (0x4E)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x4D)		0x5AB2	Watchdog answer is 0x5A2B
		FS_WD_TOKEN (0x4E)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x4D)		0x5AB2	Watchdog answer is 0x5A2B
		FS_WD_TOKEN (0x4E)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x4D)		0x5AB2	Watchdog answer is 0x5A2B
		FS_WD_TOKEN (0x4E)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x4D)		0x5AB2	Watchdog answer is 0x5A2B
		FS_WD_TOKEN (0x4E)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x4D)		0x5AB2	Watchdog answer is 0x5A2B
13	**Leave Debug Mode**	FS_STATES[14] (0x57)		1	EXIT_DBG_MODE

Table 1. Startup SPI sequence example*...continued*

		Register	Read	Write	Comment
14	**Release FS0B and FS1B**	FS_RELEASE_ FS0B_FS1B (0x51)		0xA565	
15	**FS NORMAL MODE**	FS_STATES[0:4]	0b01011		You can verify the release of FS0B and FS1B state

4 Register mapping of main logic

LEGEND		READ/ WRITE	READ	WRITE													
Address (Hex)	Register Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00	M_DEVICEID	FULL_LAYER_REV [15:13]			METAL_LAYER_REV [12:10]			FAM_ID [9:6]				DEV_ID [5:0]					
01	M_PROGID	PROG_IDH [15:8]								PROG_IDL [7:0]							
02	M_STATUS	TWARN_S	VDBG_ VOLT_S	VBST_ ACTIVE_S	VBSTFB_ UV_S	WK2_S	WK1_S	GPIO2_S	GPIO1_S	VREF_S	VBST_S	VPRE_S	TRK2_S	TRK1_S	CORE_S	LDO2_S	LDO1_S
03	M_TSD_FLG	TWARN_I	0	0	0	0	0	0	0	0	GPIO1 TSD_I	VPRETSD_I	TRK2TSD_I	TRK1TSD_I	CORETSD_I	LDO2TSD_I	LDO1TSD_I
04	M_TSD_MSK	TWARN_M	0	0	0	0	0	0	0	0	GPIO1 TSD_M	VPRETSD_ M	TRK2 TSD_M	TRK1 TSD_M	CORETSD_ M	LDO2 TSD_M	LDO1 TSD_M
05	M_REG_FLG	0	0	0	0	0	0	0	VBSTOV_I	VPREUVH_I	VBSTOC_I	VPREOC_I	TRK2OC_I	TRK1OC_I	COREOC_I	LDO2OC_I	LDO1OC_I
06	M_REG_MSK	0	0	0	0	0	0	0	VBSTOV_ M	VPREUVH_ M	VBSTOC_M	VPREOC_ M	TRK2OC_M	TRK1OC_M	COREOC_M	LDO2OC_M	LDO1OC_M
07	M_VSUP_ FLG	0	0	0	0	0	0	0	0	0	0	0	0	VBOSUVH_ I	VSUPOV_I	VSUPUV6_I	VSUPUVH_I
08	M_VSUP_ MSK	0	0	0	0	0	0	0	0	0	0	0	0	VBOSUVH_ M	VSUPOV_M	VSUPUV6_ M	VSUPUVH_ M
09	M_WIO_FLG	WU_CLR	0	0	0	WUEVENT [11:8]							LDT_I	IO2_I	IO1_I	WK2_I	WK1_I
0A	M_WIO_MSK	0	0	0	0	0	0	0	0	0	0	0	LDT_M	IO2_M	IO1_M	WK2_M	WK1_M
0B	M_COM_FLG	0	0	0	0	0	0	0	0	0	0	0	0	0	MSPI_ CRC_I	MSPI_ CLK_I	MSPI_ REQ_I
0C	M_COM_MSK	0	0	0	0	0	0	0	0	0	0	0	0	0	MSPI_ CRC_M	MSPI_ CLK_M	MSPI_ REQ_M
0D	M_SYS_CFG	RETRY_CNT [15:8]								RETRY_ CLR	0	0	INTB_TEST	INT_ PWIDTH	FSS_FMOD	0	FSS_EN
0E	M_TSD_CFG	0	0	0	0	0	0	0	0	0	0	VPRETDFS	TRK2TDFS	TRK1TDFS	CORETDFS	LDO2TDFS	LDO1TDFS
0F	M_REG_CFG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	VREF_PD	VBST_TMD
10	M_WIO_CFG	0	0	0	0	GPIO2 WUPOL	GPIO1 WUPOL	WAKE2 POL	WAKE1 POL	0	0	CSBWUEN	LDTWUEN	GPIO2 WUEN	GPIO1 WUEN	WK2WUEN	WK1WUEN
11	M_REG_ CTRL1	0	0	GPIO2LP_ON	GPIO1 LP_ON	0	0	GPIO2HI	GPIO1HI	VREFEN	VBSTEN	0	TRK2EN	TRK1EN	COREEN	LDO2EN	LDO1EN
12	M_REG_ CTRL2	0						IO2LO	IO1LO	VREFDIS	VBSTDIS	0	TRK2DIS	TRK1DIS	COREDIS	LDO2DIS	LDO1DIS
13	M_AMUX_ CTRL	0	0	0	0	0	0	0	0	0	AMUX_EN	AMUX_DIV	AMUX				
14	M_LDT_CFG1	LDT_AFTER_RUN [15:0]															
15	M_LDT_CFG2	LDT_WUP_L [15:0]															
16	M_LDT_CFG3	0	0	0	0	0	0	0	0	LDT_WUP_H [7:0]							
17	M_LDT_CTRL	0	0	0	0	0	0	0	0	0	LDT_FNCT [6:4]			LDT_SEL	LDT_MODE	LDT_EN	LDT_RUN
18	M_MEMORY0	MEM0 [15:0]															

LEGEND		READ/ WRITE	READ	WRITE	
19	M_MEMORY1	MEM1 [15:0]			

5 Register mapping of fail-safe logic

Table 2. Register mapping of fail-safe logic

Addr Hex	Register Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
40	FS_GRL_ FLAGS	FS_COM_G	FS_WD_G	FS_IO_G	FS_REG_ OVUV_G	FS_BIST_G	0	0	0	0	0	0	0	0	0	0	0
41	FS_I_ OVUV_ SAFE_ REACTION1	VMON_PRE_OV_FS_ REACTION		VMON_PRE_UV_FS_ REACTION		VMON_CORE_OV_FS_ REACTION		VMON_CORE_UV_FS_ REACTION		VMON_LDO1_OV_FS_ REACTION		VMON_LDO1_UV_FS_ REACTION		VMON_LDO2_OV_FS_ REACTION		VMON_LDO2_UV_FS_ REACTION	
42	FS_I_NOT_ OVUV_ SAFE_ REACTION1	NOT_VMON_PRE_OV_ FS_REACTION		NOT_VMON_PRE_UV_ FS_REACTION		NOT_VMON_CORE_OV_ FS_REACTION		NOT_VMON_CORE_UV_ FS_REACTION		NOT_VMON_LDO1_OV_ FS_REACTION		NOT_VMON_LDO1_UV_ FS_REACTION		NOT_VMON_LDO2_OV_ FS_REACTION		NOT_VMON_LDO2_UV_ FS_REACTION	
43	FS_I_ OVUV_ SAFE_ REACTION2	VMON_EXT_OV_FS_ REACTION		VMON_EXT_UV_FS_ REACTION		VMON_REF_OV_FS_ REACTION		VMON_REF_UV_FS_ REACTION		VMON_TRK2_OV_FS_ REACTION		VMON_TRK2_UV_FS_ REACTION		VMON_TRK1_OV_FS_ REACTION		VMON_TRK1_UV_FS_ REACTION	
44	FS_I_NOT_ OVUV_ SAFE_ REACTION2	NOT_VMON_EXT_OV_ FS_REACTION		NOT_VMON_EXT_UV_ FS_REACTION		NOT_VMON_REF_OV_ FS_REACTION		NOT_VMON_REF_UV_ FS_REACTION		NOT_VMON_TRK2_OV_ FS_REACTION		NOT_VMON_TRK2_UV_ FS_REACTION		NOT_VMON_TRK1_OV_ FS_REACTION		NOT_VMON_TRK1_UV_ FS_REACTION	
45	FS_I_WD_ CFG	WD_ERR_LIMIT		0	WD_RFR_LIMIT		0	WD_FS_REACTION		0	WD_RFR_CNT [6:4]			WD_ERR_CNT [3:0]			
46	FS_I_NOT_ WD_CFG	NOT_WD_ERR_LIMIT		0	NOT_WD_RFR_LIMIT		0	NOT_WD_FS_REACTION		0	NOT_WD_RFR_CNT [6:4]			NOT_WD_ERR_CNT [3:0]			
47	FS_I_ SAFE_ INPUTS	FCCU_CFG [15:13]			FCCU12_ FLT_POL	FCCU1_ FLT_POL	FCCU2_ FLT_POL	FCCU12_ FS_ REACTION	FCCU1_FS_ REACTION	FCCU2_FS_ REACTION	0	ERRMON_ FLT_ POLARITY	ERRMON_ACK_TIME [4:3]		ERRMON_ FS_ REACTION	FCCU12_FILT [1:0]	
48	FS_I_NOT_ SAFE_ INPUTS	NO_FCCU_CFG [15:13]			NO_ FCCU12_ FLT_POL	NO_ FCCU1_ FLT_POL	NO_ FCCU2_ FLT_POL	NO_ FCCU12_ FS_ REACTION	NO_ FCCU1_FS_ REACTION	NO_ FCCU2_FS_ REACTION	1	NO_ ERRMON_ FLT_ POLARITY	NO_ERRMON_ACK_TIME [4:3]		NO_ ERRMON_ FS_ REACTION	NO_FCCU12_FILT [1:0]	
49	FS_I_FSSM	FLT_ERR_CNT_LIMIT [15:14]		0	FLT_ERR_REACTION [12:11]		0	RSTB_DUR	0	BACKUP_ SAFETY_ PATH_FS0B	0	CLK_MON_ DIS	DIS8S	FLT_ERR_CNT [3:0]			
4A	FS_I_NOT_ FSSM	NO_FLT_ERR_CNT_LIMIT [15:14]		0	NO_FLT_ERR_REACTION [12:11]		0	NO_RSTB_ DUR	0	NO_ BACKUP_ SAFETY_ PATH_FS0B	1	NO_CLK_ MON_DIS	NO_DIS8S	NO_FLT_ERR_CNT [3:0]			
4B	FS_WDW_ DURATION	WDW_PERIOD				0	0	0	WDW_DC		0	0	0	WDW_RECOVERY			
4C	FS_NOT_ WDW_ DURATION	NO_WDW_PERIOD				0	0	0	NO_WDW_DC		0	0	0	NO_WDW_RECOVERY			

Table 2. Register mapping of fail-safe logic*...continued*

Addr Hex	Register Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
4D	FS_WD_ ANSWER	WD_ANSWER [15:0]															
4E	FS_WD_ TOKEN	WD_TOKEN [15:0]															
4F	FS_ABIST_ ON_ DEMAND	LAUNCH_ ABIST2	0	0	0	0	0	0	0	ABIST2_ EXT	ABIST2_ REF	ABIST2_ TRK2	ABIST2_ TRK1	ABIST2_ LDO2	ABIST2_ LDO1	ABIST2_ CORE	ABIST2_ VPRE
50	FS_OVUV_ REG_ STATUS	VPRE_OV	VPRE_UV	CORE_OV	CORE_UV	LDO1_OV	LDO1_UV	LDO2_OV	LDO2_UV	TRK1_OV	TRK1_UV	TRK2_OV	TRK2_UV	REF_OV	REF_UV	EXT_OV	EXT_UV
51	FS_ RELEASE_ FS0B_FS1B	RELEASE_FS0B_FS1B															
52	FS_SAFE_ IOS_1	EXT_RSTB	RSTB_DRV	RSTB_SNS	RSTB_ EVENT	RSTB_DIAG	RSTB_REQ	FS0B_DRV	FS0B_SNS	FS0B_DIAG	FS0B_REQ	FS1B_DRV	FS1B_SNS	FS1N_DIAG	FS1B_REQ	GOTO_INIT	0
53	FS_SAFE_ IOS_2	0	0	0	0	0	0	FS1B_TDELAY [9:5]					FS1B_TDUR [4:0]				
54	FS_DIAG_ SAFETY1	0	0	0	0	0	BAD_WD_ DATA	BAD_WD_ TIMING	ABIST1_ PASS	ABIST2_ PASS	ABIST2_ DONE	SPI_FS_ CLK	SPI_FS_ REQ	SPI_FS_ CRC	FS_OSC_ DRIFT	LBIST_STATUS [1:0]	
55	FS_DIAG_ SAFETY2	0	0	0	0	0	0	0	0	FCCU12	FCCU1	FCCU2	FCCU1_RT	FCCU2_RT	ERRMON_ ACK	ERRMON	ERRMON_ PIN_ STATUS
56	FS_INTB_ MASK	VPRE_M	CORE_M	LDO1_M	LDO2_M	TRK1_M	TRK2_M	REF_M	EXT_M	FCCU1_M	FCCU2_M	BAD_WD_M	ERRMON_ M	0	0	0	0
57	FS_STATES	0	EXIT_DBG_ MODE	DBG_MODE	OTP_ CORRUPT	REG_ CORRUPT	0	0	0	0	0	0	FS_STATES				
58	FS_LP_REQ	0	0	0	0	0	0	0	STBY_ WAKE_UP	FS_LP_REQ							
59	FS_LDT_ LPSEL	0	0	0	0	0	0	0	0	LDT_SEL							

6 Watchdog answer procedure

A watchdog is implemented through the SPI bus to continuously check the microcontroller software activity and its ability to perform basic computing. The FS26 performs this check by waiting for a specific answer from the microcontroller during a predefined period called the watchdog window. The first half of the watchdog window is said to be CLOSED and the second half is said to be OPEN.

A good watchdog refresh is a good watchdog answer during the OPEN window. A bad watchdog refresh is a bad watchdog answer during the OPEN window, no watchdog refresh during the OPEN window, or a good watchdog answer during the CLOSED window. After a good or a bad watchdog refresh, a new window period starts immediately for the microcontroller to keep the synchronization with the windowed watchdog.

Table 3. Watchdog answer procedure

SPI	Window watchdog		Timeout watchdog
	CLOSED	OPEN	(always open)
BAD key	WD_NOK	WD_NOK	WD_NOK
GOOD key	WD_NOK	**WD_OK**	**WD_OK**
None (timeout)	NA	WD_NOK	WD_NOK

6.1 Simple watchdog

The simple watchdog monitoring feature is enabled for ASIL B devices. The microcontroller can send its own seed in FS_WD_TOKEN register or can use the default value 0x5AB2. This seed must be written in the FS_WD_ANSWER register during the OPEN watchdog window. When the result is right, the watchdog window is restarted. When the result is wrong, the WD error counter is incremented, and the watchdog window is restarted. In simple watchdog configuration, it is impossible to write 0xFFFF and 0x0000 in the FS_WD_TOKEN register. A communication error is reported in case of 0x0000 and 0xFFFF write tentative and the configuration is ignored.

6.2 Challenger watchdog

The challenger watchdog monitoring feature is enabled for ASIL D devices. The challenger watchdog is based on a question/answer process with the microcontroller. A 16-bit pseudo-random word is generated by implementing a linear feedback shift register (LFSR) in the FS26. During the initialization phase, the microcontroller can send its own seed for the LFSR, or it can use the default LFSR value generated by the FS26 (0x5AB2), available in the FS_WD_TOKEN register. With the LFSR value, the microcontroller performs a simple calculation based on the formula below and sends the results in the FS_WD_ANSWER register. The result is sent through the SPI bus during the OPEN watchdog window and is verified by the FS26. When the result is right, the watchdog window is restarted and a new LFSR is generated. When the result is wrong, the WD error counter is incremented, the watchdog window is restarted, and the LFSR value is not changed.

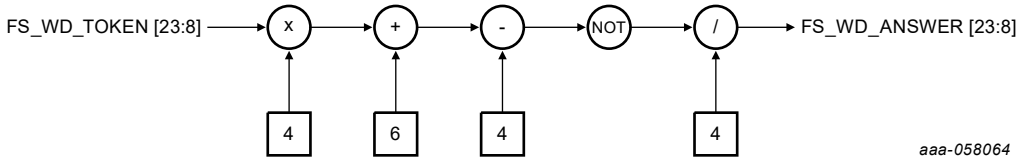

Figure 2. Challenger watchdog

7 Cyclic redundancy check

An 8-bit cyclic redundancy check (CRC) is required for each write and read SPI command. Computation of a cyclic redundancy check is derived from the mathematics of polynomial division, modulo two. The CRC polynomial used is x^8+x^4+x^3+x^2+1 (identified by 0x1D) with a SEED value of hexadecimal '0xFF'.

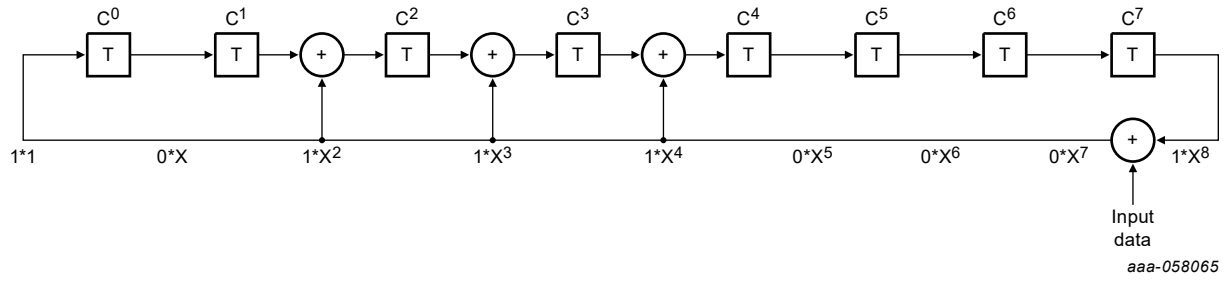

Figure 3. Cyclic redundancy check

8 FS0B/FS1B release procedure

The procedure to compute the RELEASE_FS0B_FS1B [15:0] value to release the safety outputs is described in Table 4, illustrating all these steps with an example:

1. Get the FS_WD_TOKEN value.
2. Swap MSB/LSB of the value get in step No. 1.
3. Invert all computed bits at step No. 2.
4. Write bits 12 to 0 computed in step No. 3 into RELEASE_FS0B_FS1B [12:0] register. Bits 15 to 13 are used to select the safety output(s) to release as shown in Table 5.

Table 4. Procedure to compute the RELEASE_FS0B_FS1B [15:0] value

Step #1	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Read FS_WD_TOKEN register	1	1	1	0	0	1	0	0	1	1	1	1	0	0	0	0
Step #1	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Reverse LSB/MSB	0	0	0	0	1	1	1	1	0	0	1	0	0	1	1	1
Step #1	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Complement bits	1	1	1	1	0	0	0	0	1	1	0	1	1	0	0	0
Step #1	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Write RELEASE_FS0B_FS1B[12:0]	1	0	1	1	0	0	0	0	1	1	0	1	1	0	0	0

Table 5. RELEASE_FS0B_FS1B bits

Bit	Symbol	Description
15 to 13	RELEASE_FS0B_FS1B[15:13]	Bits to select the desired safety output to release
		011 Release FS0B only
		110 Release FS1B only
		101 Release FS0B and FS1B

9 Summary of references

Documentation

[1] FS26 device webpage

[2] FS26 design guidelines application note

Software resources

[3] FS26 AUTOSAR software drivers

Evaluation resources

[4] FS26 graphical user interface (GUI)

[5] Socketed evaluation board

[6] Soldered evaluation board

10 Revision history

Table 6. Revision history

Document ID	Release date	Description
AN14492 v.1.0	10 Jan. 2025	Initial version

Legal information

Definitions

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

Disclaimers

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

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

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

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

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

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

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

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

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

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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

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

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

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

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

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

Trademarks

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

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

Tables

Figures

Contents