AN14041

FS23 quick start guide

Rev. 2.0 — 23 January 2025

Application note

Document information

Information	Content
Keywords	FS2300, FS2320, safety system basis chip, SBC, body and comfort, controller area network (CAN) FD, local interconnect network (LIN)
Abstract	This application note is intended for the engineers involved in software implementation of FS23 fail-safe system basis chips.

1 Introduction

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

This document gives guidance on the implementation of SPI or I²C communication protocol between the MCU and the FS23.

This document explains the initialization procedure of the FS23 device and provides an example of start-up sequence.

1.1 General description

The FS23 SBC offers an expandable family of devices that is pin-to-pin and software compatible. The devices are scalable from the LDO version to the DC-DC version, as well as from QM to ASIL B. The FS23 SBC includes CAN and LIN transceivers, along with a number of system and safety features for the latest generation of automotive electronic control units (ECUs).

The flexibility of the FS23 SBC makes it suitable for S32K processor-based applications, as well as multivendor processors.

Several device versions are available, offering a choice of output voltage settings, operating frequency, power-up sequencing, and input/output configuration to address multiple applications.

1.2 Reference documents

Reference documents and various material are available on the FS23 device webpage. The webpage provides more detailed information about specific topics:

FS23 data sheet: Information, such as features, functional description, parametric description, register mapping.

FS23 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:

FS23 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:

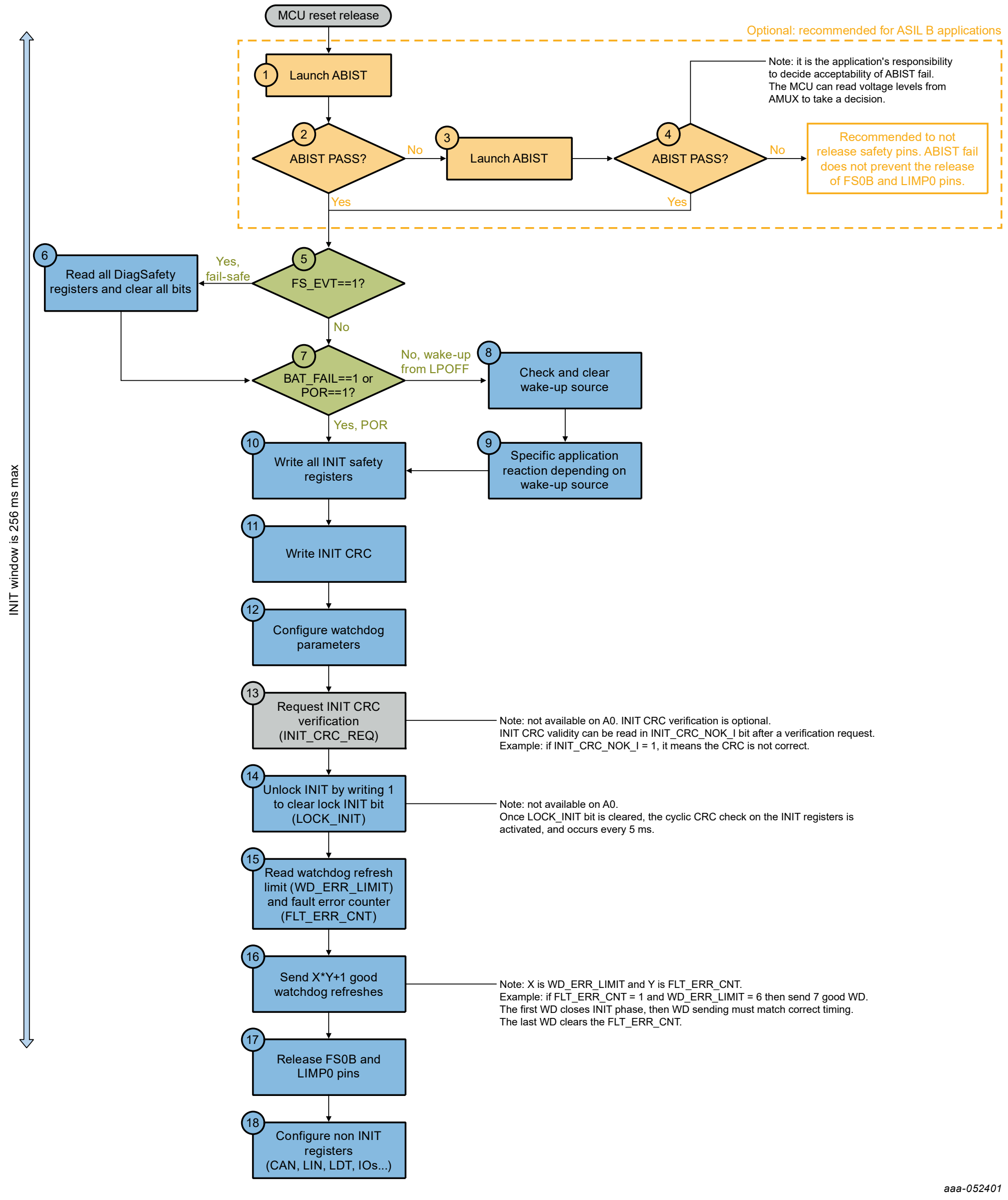
- SPI access register function and events handling (SBC_FS23)
- CAN/LIN function (CANTRCV_FS23 and LINTRCV_FS23)
- Watchdog function (WDG_FS23)

2 FS23 initialization flow chart example

Figure 1 gives an example of FS23 software initialization. After MCU reset is released (RSTB state is high), the MCU can start FS23 initialization. The initialization must be done within the dedicated 256 ms INIT window.

Running the ABIST is optional, though it is recommended for ASIL B applications. ABIST can be run multiple times in a row. In this example, the MCU checks the cause of the MCU reset (POR, LPOFF, fail-safe) and takes action accordingly. Then the MCU writes INIT safety registers, ending with INIT cyclic redundancy check (CRC). The next step is watchdog configuration, and unlocking the INIT CRC cyclic check, followed by watchdog refreshes to clear the fault error counter. The first watchdog refresh closes the INIT phase.

Therefore, the subsequent watchdog refreshes must be sent according to watchdog timing configuration. Once the fault error counter is cleared, safety pins FS0B and LIMP0 can be released.


MCU reset release
Optional: recommended for ASIL B applications
1 Launch ABIST
2 ABIST PASS?
No
3 Launch ABIST
4 ABIST PASS?
No
Recommended to not release safety pins. ABIST fail does not prevent the release of FS0B and LIMP0 pins.
Yes
Yes
5 FS_EVT==1?
Yes, fail-safe
6 Read all DiagSafety registers and clear all bits
No
7 BAT_FAIL==1 or POR==1?
No, wake-up from LPOFF
8 Check and clear wake-up source
9 Specific application reaction depending on wake-up source
Yes, POR
10 Write all INIT safety registers
11 Write INIT CRC
12 Configure watchdog parameters
13 Request INIT CRC verification (INIT_CRC_REQ)
14 Unlock INIT by writing 1 to clear lock INIT bit (LOCK_INIT)
15 Read watchdog refresh limit (WD_ERR_LIMIT) and fault error counter (FLT_ERR_CNT)
16 Send X*Y+1 good watchdog refreshes
17 Release FS0B and LIMP0 pins
18 Configure non INIT registers (CAN, LIN, LDT, IOs...)
INIT window is 256 ms max
aaa-052401

Note: it is the application's responsibility to decide acceptability of ABIST fail. The MCU can read voltage levels from AMUX to take a decision.

Note: not available on A0. INIT CRC verification is optional.
INIT CRC validity can be read in INIT_CRC_NOK_I bit after a verification request.
Example: if INIT_CRC_NOK_I = 1, it means the CRC is not correct.

Note: not available on A0.
Once LOCK_INIT bit is cleared, the cyclic CRC check on the INIT registers is activated, and occurs every 5 ms.

Note: X is WD_ERR_LIMIT and Y is FLT_ERR_CNT.
Example: if FLT_ERR_CNT = 1 and WD_ERR_LIMIT = 6 then send 7 good WD.
The first WD closes INIT phase, then WD sending must match correct timing.
The last WD clears the FLT_ERR_CNT.

Figure 1. FS23 initialization flow chart example

3 Start-up I^{2}C/SPI sequence example (based on flow chart)

Table 1. Start-up I^{2}C/SPI sequence example

		Register	Read	Write	Comment
1	Launch ABIST	FS_ABIST (0x3D)		0x4000	Optional: recommended for ASIL B applications. Full ABIST launch by writing LAUNCH_ABIST bit. Specific ABIST can be launched using appropriate bits.
2	Check ABIST	FS_ABIST (0x3D)	0x07C0		Optional: recommended for ASIL B applications. Check ABIST diagnostic bits.
3	Launch ABIST	FS_ABIST (0x3D)		0x4000	Optional: recommended for ASIL B applications. Full ABIST launch by writing LAUNCH_ABIST bit. Specific ABIST can be launched using appropriate bits.
4	Check ABIST	FS_ABIST (0x3D)	0x07C0		Optional: recommended for ASIL B applications. Check ABIST diagnostic bits.
5	MCU reset from FS?	M_WU1_FLG (0x17)	0x0200		Check FS_EVT bit: 0x0200 if wake-up from fail-safe.
6	Read diagnostic registers and clear all bits	FS_SAFETY_OUTPUTS (0x3F)	0x3804		Default value: RSTB released, FS0B asserted, LIMP0 released
		FS_SAFETY_FLG (0x40)	0x0002		Default value: FCCU1 sensed high
		M_REG_FLG (0x0A)	0x0000		Default value
7	MCU reset from POR/LPOFF?	M_SYS_CONFIG (0x05)	0x5000		Check BAT_FAIL and POR bits: 0x5000 if wake-up from POR
8	Check WU source	M_WU1_FLG (0x17)	0x0000		Check wake-up sources if wake-up from LPOFF: 0x0000 if wake-up from POR
		M_IOWU_FLG (0x15)	0x0000		
10	INIT	FS_I_OVUV_CFG1 (0x32)		0x1F98	Default value
		FS_I_OVUV_CFG2 (0x33)		0x0C18	Default value
		FS_I_FCCU_CFG (0x34)		0x103F	Default value
		FS_I_FSSM_CFG (0x36)		0x0AF1	Default value, FLT_ERR_CNT = 1 and FLT_ERR_LIMIT = 6
		FS_I_WD_CFG (0x37)		0x7080	Default value
11	Send INIT CRC	FS_CRC (0x41)		0x06B4	INIT CRC to be computed to match INIT registers content
12	Configure Watchdog	FS_WDW_CFG (0x38)		0x01AB	Default value
13	Request INIT CRC verification	FS_CRC (0x41)		0x46B4	Optional: INIT CRC to be computed to match INIT registers content
14	Unlock INIT CRC cyclic check	M_SYS_CONFIG (0x05)		0x0400	Clearing LOCK_INIT bit unlocks INIT CRC cyclic check
15	Read watchdog current counter value	FS_I_WD_CFG (0x37)	0x7080		Default value, with watchdog error limit = 6
		FS_I_FSSM_CFG (0x36)	0x0AF1		Default value, with fault error counter = 1
16	Send 7x good WD refresh (if WD_ERR_LIMIT = 6 and FLT_ERR_CNT = 1)	FS_WD_TOKEN (0x39)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x3A)		0x5AB2	Watchdog answer is 0x5A2B (default value)
		FS_WD_TOKEN (0x39)	0xD564		Read watchdog token
		FS_WD_ANSWER (0x3A)		0xD564	Watchdog answer is 0xD564
		FS_WD_TOKEN (0x39)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x3A)		0x5AB2	Watchdog answer is 0x5A2B
		FS_WD_TOKEN (0x39)	0xD564		Read watchdog token
		FS_WD_ANSWER (0x3A)		0xD564	Watchdog answer is 0xD564
		FS_WD_TOKEN (0x39)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x3A)		0x5AB2	Watchdog answer is 0x5A2B
		FS_WD_TOKEN (0x39)	0xD564		Read watchdog token
		FS_WD_ANSWER (0x3A)		0xD564	Watchdog answer is 0xD564
		FS_WD_TOKEN (0x39)	0x5AB2		Read watchdog token
		FS_WD_ANSWER (0x3A)		0x5AB2	Watchdog answer is 0x5A2B
17	Release FS0B	FS_WD_TOKEN (0x39)	0xD564		Read watchdog token
		FS_FS0B_LIMP0_REL		0x7B2A	Compute FS0B and LIMP0 release register value
18	Configure CAN, LIN, LDT, I/Os…	M_CAN (0x2A)		0x02A0	CAN in Normal operation mode
		M_LIN (0x2B)		0x4400	LIN in Normal operation mode

4 I²C/SPI register mapping of main logic

Table 2. Main register mapping

Refer to Table 70 from the FS23 data sheet.

Register	#	Address							R/W SPI	R/W I^2C	Read/Write
		Adr_6	Adr_5	Adr_4	Adr_3	Adr_2	Adr_1	Adr_0			
M_DEV_CFG	0	0	0	0	0	0	0	0	0/1	1/0	Read only
M_DEV_PROG_ID	1	0	0	0	0	0	0	1	0/1	1/0	Read only
M_GEN_FLAG	2	0	0	0	0	0	1	0	0/1	1/0	Read only
M_STATUS	3	0	0	0	0	0	1	1	0/1	1/0	Read only
Reserved	4	0	0	0	0	1	0	0	0/1	1/0	Reserved
M_SYS_CFG	5	0	0	0	0	1	0	1	0/1	1/0	Read/Write
M_SYS1_CFG	6	0	0	0	0	1	1	0	0/1	1/0	Read/Write
M_REG_CTRL	7	0	0	0	0	1	1	1	0/1	1/0	Read/Write
Reserved	8	0	0	0	1	0	0	0	0/1	1/0	Reserved
Reserved	9	0	0	0	1	0	0	1	0/1	1/0	Reserved
M_REG_FLG	10	0	0	0	1	0	1	0	0/1	1/0	Read/Write
M_REG_MSK	11	0	0	0	1	0	1	1	0/1	1/0	Read/Write
M_REG1_FLG	12	0	0	0	1	1	0	0	0/1	1/0	Read/Write
M_REG1_MSK	13	0	0	0	1	1	0	1	0/1	1/0	Read/Write
M_IO_CTRL	14	0	0	0	1	1	1	0	0/1	1/0	Write
M_IO_TIMER_FLG	15	0	0	0	1	1	1	1	0/1	1/0	Read/Write
M_IO_TIMER_MSK	16	0	0	1	0	0	0	0	0/1	1/0	Read/Write
M_VSUP_COM_FLG	17	0	0	1	0	0	0	1	0/1	1/0	Read/Write
M_VSUP_COM_MSK	18	0	0	1	0	0	1	0	0/1	1/0	Read/Write
M_IOWU_CFG	19	0	0	1	0	0	1	1	0/1	1/0	Read/Write
M_IOWU_EN	20	0	0	1	0	1	0	0	0/1	1/0	Read/Write
M_IOWU_FLG	21	0	0	1	0	1	0	1	0/1	1/0	Read/Write
M_WU1_EN	22	0	0	1	0	1	1	0	0/1	1/0	Read/Write
M_WU1_FLG	23	0	0	1	0	1	1	1	0/1	1/0	Read/Write
M_TIMER1_CFG	24	0	0	1	1	0	0	0	0/1	1/0	Read/Write
M_TIMER2_CFG	25	0	0	1	1	0	0	1	0/1	1/0	Read/Write
M_TIMER3_CFG	26	0	0	1	1	0	1	0	0/1	1/0	Read/Write
M_PWM1_CFG	27	0	0	1	1	0	1	1	0/1	1/0	Read/Write
M_PWM2_CFG	28	0	0	1	1	1	0	0	0/1	1/0	Read/Write
M_PWM3_CFG	29	0	0	1	1	1	0	1	0/1	1/0	Read/Write
M_TIMER_PWM_CTRL	30	0	0	1	1	1	1	0	0/1	1/0	Read/Write
M_CS_CFG	31	0	0	1	1	1	1	1	0/1	1/0	Read/Write
M_CS_FLG_MSK	32	0	1	0	0	0	0	0	0/1	1/0	Read/Write
M_HSx_SRC_CFG	33	0	1	0	0	0	0	1	0/1	1/0	Read/Write
M_HSx_CTRL	34	0	1	0	0	0	1	0	0/1	1/0	Read/Write
M_HSx_FLG	35	0	1	0	0	0	1	1	0/1	1/0	Read/Write
M_HSx_MSK	36	0	1	0	0	1	0	0	0/1	1/0	Read/Write
M_AMUX_CTRL	37	0	1	0	0	1	0	1	0/1	1/0	Read/Write
M_LDT_CFG1	38	0	1	0	0	1	1	0	0/1	1/0	Read/Write
M_LDT_CFG2	39	0	1	0	0	1	1	1	0/1	1/0	Read/Write
M_LDT_CFG3	40	0	1	0	1	0	0	0	0/1	1/0	Read/Write
M_LDT_CTRL	41	0	1	0	1	0	0	1	0/1	1/0	Read/Write
M_CAN	42	0	1	0	1	0	1	0	0/1	1/0	Read/Write
M_LIN	43	0	1	0	1	0	1	1	0/1	1/0	Read/Write
M_CAN_LIN_MSK	44	0	1	0	1	1	0	0	0/1	1/0	Read/Write
M_MEMORY0	45	0	1	0	1	1	0	1	0/1	1/0	Read/Write
M_MEMORY1	46	0	1	0	1	1	1	0	0/1	1/0	Read/Write

5 Register mapping of fail-safe logic

Table 3. Safety-related register mapping

Refer to Table 71 from the FS23 data sheet.

Register	#	Address							R/W SPI	R/W I^{2}C	Read/Write
		Adr_6	Adr_5	Adr_4	Adr_3	Adr_2	Adr_1	Adr_0			
FS_I_OVUV_CFG1	50	0	1	1	0	0	1	0	0/1	1/0	Write during INIT, then read only
FS_I_OVUV_CFG2	51	0	1	1	0	0	1	1	0/1	1/0	Write during INIT, then read only
FS_I_FCCU_CFG	52	0	1	1	0	1	0	0	0/1	1/0	Write during INIT, then read only
Reserved	53	0	1	1	0	1	0	1	0/1	1/0	Reserved
FS_I_FSSM_CFG	54	0	1	1	0	1	1	0	0/1	1/0	Write during INIT, thenr ead only
FS_I_WD_CFG	55	0	1	1	0	1	1	1	0/1	1/0	Write during INIT, then read only
FS_WDW_CFG	56	0	1	1	1	0	0	0	0/1	1/0	Read/Write
FS_WD_TOKEN	57	0	1	1	1	0	0	1	0/1	1/0	Read only
FS_WD_ANSWER	58	0	1	1	1	0	1	0	0/1	1/0	Write only
FS_LIMP12_CFG	59	0	1	1	1	0	1	1	0/1	1/0	Read/Write
FS_FS0B_LIMP0_REL	60	0	1	1	1	1	0	0	0/1	1/0	Read/Write
FS_ABIST	61	0	1	1	1	1	0	1	0/1	1/0	Read/Write
Reserved	62	0	1	1	1	1	1	0	0/1	1/0	Reserved
FS_SAFETY_OUTPUTS	63	0	1	1	1	1	1	1	0/1	1/0	Read/Write
FS_SAFETY_FLG	64	1	0	0	0	0	0	0	0/1	1/0	Read/Write
FS_CRC	65	1	0	0	0	0	0	1	0/1	1/0	Read/Write

6 Readable registers

Table 4. Readable registers

Logic	Register name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Main	M_DEV_CFG	0	0	CAN_EN	LIN_EN	LDTIM_EN	HSD13_EN	HSD24_EN	V2_EN	V1_PNP_EN	ABIST_EN	FCCU_EN	FS0B_EN	LIMP0_EN	V0MON_EN	0	0
	M_DEV_PROG_ID	FULL_LAYER_REV				METAL_LAYER_REV				PROG_IDH				PROG_IDL			
	M_GEN_FLAG	0	0	0	0	0	0	0	0	HSxG	SAFETYG	PHYG	WUG	IOTIMG	COMG	VSUPG	VxG
	M_STATUS	V1TWARN_S	LPON_S	NORMAL_S	INIT_S	0	WK2_S	WK1_S	HVIO2_S	HVIO1_S	LVI5_S	LVIO4_S	LVIO3_S	V1_MODE	V1_S	V2_S	V3_S
	M_SYS_CFG	0	BAT_FAIL	0	POR	0	LOCK_INIT	0	0	0	0	INT_TO_WUEN	0	INTB_DUR	0	MOD_CONF	MOD_EN
	M_SYS1_CFG	0	0	0	VBOS2 V1_SW_ALWAYS_EN	0	LOAD_OTP_BYP	SLOT_BYP	TSLOT_DOWN_CFG	0	0	0	0	DBG_MODE	0	0	OTP_MODE
	M_REG_CTRL	0	0	0	BUCK_SRHSOFF		BUCK_SRHSON			0	0	V2ON_LPON	0	0	V3ON_LPON	0	0
	M_REG_FLG	V0UV_I	V0OV_I	V1TWARN_I	V1TSD_I	V2TSD_I	V3TSD_I	V2OL_I	V1UV_I	V2UV_I	V3UV_I	V1OV_I	V2OV_I	V3OV_I	V1OC_I	V2OC_I	V3OC_I
	M_REG_MSK	V0UV_M	V0OV_M	V1TWARN_M	V1TSD_M	V2TSD_M	V3TSD_M	V2OL_M	V1UV_M	V2UV_M	V3UV_M	V1OV_M	V2OV_M	V3OV_M	V1OC_M	V2OC_M	V3OC_M
	M_REG1_FLG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V1_OCLS_I
	M_REG1_MSK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	V1_OCLS_M
	M_IO_TIMER_FLG	0	0	0	0	0	0	0	LDT_I	LVI5_I	LVIO4_I	LVIO3_I	HVIO2_I	HVIO1_I	0	WK2_I	WK1_I
	M_IO_TIMER_MSK	0	0	0	0	0	0	0	LDT_M	LVI5_M	LVIO4_M	LVIO3_M	HVIO2_M	HVIO1_M	0	WK2_M	WK1_M
	M_VSUP_COM_FLG	0	0	0	VBOS2 V1SW_S	VBOS_UV	0	I2C_CRC_I	I2C_REQ_I	SPI_CRC_I	SPI_CLK_I	SPI_REQ_I	0	VSHS_OV_I	VSHS_UV_I	VSUPOV_I	VSUPUV_I
	M_VSUP_COM_MSK	0	0	0	0	0	0	I2C_CRC_M	I2C_REQ_M	SPI_CRC_M	SPI_CLK_M	SPI_REQ_M	0	VSHS_OV_M	VSHS_UV_M	VSUPOV_M	VSUPUV_M
	M_IOWU_CFG	LVI5_WUCFG	LVIO4_WUCFG	LVIO3_WUCFG	0	HVIO2_DGLT	HVIO1_DGLT	WK2_DGLT	WK1_DGLT	HVIO2_WUCFG		HVIO1_WUCFG		WK2_WUCFG		WK1_WUCFG	
	M_IOWU_EN	0	0	LVI5_WUEN		LVIO4_WUEN		LVIO3_WUEN		HVIO2_WUEN		HVIO1_WUEN		WK2_WUEN		WK1_WUEN	
	M_IOWU_FLG	LVI5_WU_I	LVIO4_WU_I	LVIO3_WU_I	0	HVIO2_CYS_RDY	HVIO1_CYS_RDY	HVIO2_CYC_S	HVIO1_CYC_S	HVIO2_WU_I	HVIO1_WU_I	WK2_CYS_RDY	WK1_CYS_RDY	WK2_CYC_S	WK1_CYC_S	WK2_WU_I	WK1_WU_I
	M_WU1_EN	0	0	0	0	0	0	0	0	0	0	LDT_WUEN		LIN_WUEN		CAN_WUEN	
	M_WU1_FLG	0	0	0	0	0	0	FS_EVT	EXT_RSTB_WU	WD_OFL_WU	V1_UVLP_WU	INT_TO_WU	GO2 NORMAL_WU	0	LDT_WU_I	LIN_WU_I	CAN_WU_I
	M_TIMER1_CFG	0	0	0	0	0	0	0	TIMER1_DLY		TIMER1_ON				TIMER1_PER		
	M_TIMER2_CFG	0	0	0	0	0	0	0	TIMER2_DLY		TIMER2_ON				TIMER2_PER		
	M_TIMER3_CFG	0	0	0	0	0	0	0	TIMER3_DLY		TIMER3_ON				TIMER3_PER		
	M_PWM1_CFG	0	0	0	PWM1_DLY		PWM1_F	PWM1_DC									
	M_PWM2_CFG	0	0	0	PWM2_DLY		PWM2_F	PWM2_DC									
	M_PWM3_CFG	0	0	0	PWM3_DLY		PWM3_F	PWM3_DC									
	M_TIMER_PWM_CTRL	0	0	0	0	0	0	0	0	0	TIM1_EN	TIM2_EN	TIM3_EN	0	PWM1_EN	PWM2_EN	PWM3_EN
	M_CS_CFG	0	0	0	0	0	0	HS_FLT_WU_FORCE	0	HVIO2_HS_SEL		HVIO1_HS_SEL		WK2_HS_SEL		WK1_HS_SEL	
	M_CS_FLG_MSK	0	0	0	0	0	0	0	HVIO2_OL_M	HVIO1_OL_M	WAKE2_OL_M	WAKE1_OL_M	0	HVIO2_OL_I	HVIO1_OL_I	WAKE2_OL_I	WAKE1_OL_I
	M_HSx_SRC_CFG	HS4_SRC_SEL				HS3_SRC_SEL				HS2_SRC_SEL				HS1_SRC_SEL			
	M_HSx_CTRL	0	HS_VSHSUVOV_REC	HS_VSHSUV_DIS	HS_VSHSOV_DIS	0	0	0	0	0	HS4_EN	0	HS3_EN	0	HS2_EN	0	HS1_EN
	M_HSx_FLG	0	0	0	HS4_OL_I	HS4_OC_I	0	HS3_OL_I	HS3_OC_I	HS34_TSD_I	0	HS2_OL_I	HS2_OC_I	0	HS1_OL_I	HS1_OC_I	HS12_TSD_I

Table 4. Readable registers*...continued*

Logic	Register name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	M_HSx_MSK	0	0	0	HS4_OL_M	HS4_OC_M	0	HS3_OL_M	HS3_OC_M	HS34_TSD_M	0	HS2_OL_M	HS2_OC_M	0	HS1_OL_M	HS1_OC_M	HS12_TSD_M
	M_AMUX_CTRL	0	0	0	0	0	0	AMUX_EN	AMUX_DIV	0	0	0	AMUX				
	M_LDT_CFG1	LDT_AFTER_RUN															
	M_LDT_CFG2	LDT_WUP_L															
	M_LDT_CFG3	0	0	0	0	0	0	0	0	LDT_WUP_H							
	M_LDT_CTRL	0	0	0	0	0	0	0	0	LDT2LP	LDT_FNCT			LDT_SEL	LDT_MODE	LDT_EN	LDT_RUN
	M_CAN	0	0	0	0	0	0	CAN_MODE		CAN_ACTIVE_MODE_S	0	CAN_FS_DIS	0	0	0	CAN_TXD_TO_I	CAN_TSD_I
	M_LIN	0	LIN_MODE		LIN_SLOPE		LIN_FS_DIS	LIN_VSHSUV_DIS	LIN_SC	LIN_TXD_TO	0	0	0	0	LIN_SC_I	LIN_TXD_TO_I	LIN_TSD_I
	M_CAN_LIN_MSK	0	0	LIN_FSM_STATE_S					LIN_SC_M	LIN_TXD_TO_M	LIN_TSD_M	0	CAN_FSM_STATE_S			CAN_TXD_TO_M	CAN_TSD_M
	M_MEMORY0	MEMORY0[15]	MEMORY0[14]	MEMORY0[13]	MEMORY0[12]	MEMORY0[11]	MEMORY0[10]	MEMORY0[9]	MEMORY0[8]	MEMORY0[7]	MEMORY0[6]	MEMORY0[5]	MEMORY0[4]	MEMORY0[3]	MEMORY0[2]	MEMORY0[1]	MEMORY0[0]
	M_MEMORY1	MEMORY1[15]	MEMORY1[14]	MEMORY1[13]	MEMORY1[12]	MEMORY1[11]	MEMORY1[10]	MEMORY1[9]	MEMORY1[8]	MEMORY1[7]	MEMORY1[6]	MEMORY1[5]	MEMORY1[4]	MEMORY1[3]	MEMORY1[2]	MEMORY1[1]	MEMORY1[0]
Fail-safe	FS_I_OVUV_CFG1	0	0	0	V1MON_OV_RSTB_IMPACT	V1MON_OV_FS0B_ IMPACT	V1MON_OV_LIMP0_IMPACT	V1MON_UV_RSTB_IMPACT	V1MON_UV_FS0B_ IMPACT	V1MON_UV_LIMP0_IMPACT	0	V2MON_OV_RSTB_IMPACT	V2MON_OV_FS0B_IMPACT	V2MON_OV_LIMP0_IMPACT	V2MON_UV_RSTB_IMPACT	V2MON_UV_FS0B_IMPACT	V2MON_UV_LIMP0_IMPACT
	FS_I_OVUV_CFG2	0	0	0	V3MON_OV_RSTB_IMPACT	V3MON_OV_FS0B_IMPACT	V3MON_OV_LIMP0_IMPACT	V3MON_UV_RSTB_IMPACT	V3MON_UV_FS0B_ IMPACT	V3MON_UV_LIMP0_IMPACT	0	V0MON_OV_RSTB_IMPACT	V0MON_OV_FS0B_IMPACT	V0MON_OV_LIMP0_IMPACT	V0MON_UV_RSTB_IMPACT	V0MON_UV_FS0B_IMPACT	V0MON_UV_LIMP0_IMPACT
	FS_I_FCCU_CFG	0	FCCU_CFG			FCCU2_ASSIGN			FCCU12_FLT_POL	FCCU2_FLT_POL	FCCU1_FLT_POL	FCCU2_RSTB_IMPACT	FCCU2_FS0B_ IMPACT	FCCU2_LIMP0_IMPACT	FCCU1_RSTB_IMPACT	FCCU1_FS0B_ IMPACT	FCCU1_LIMP0_IMPACT
	FS_I_FSSM_CFG	0	EXT_RSTB_DIS	RSTB8S_DIS	RSTB_DUR	LIMP0_SC_RSTB_IMPACT	EXTRSTB_FS0B_ IMPACT	FS0B_SC_RSTB_IMPACT	FLT_ERR_LIMIT		FLT_MID_RSTB_IMPACT	FLT_MID_FS0B_ IMPACT	FLT_MID_LIMP0_IMPACT	FLT_ERR_CNT			
	FS_I_WD_CFG	0	WD_RSTB_IMPACT	WD_FS0B_IMPACT	WD_LIMP0_IMPACT	WD_DIS_LPON	WD_RFR_LIMIT		WD_ERR_LIMIT		WD_RFR_CNT			WD_ERR_CNT			
	FS_WDW_CFG	0	0	0	0	WDW_REC_EN	WDW_EN	0	WDW_PERIOD				0	WDW_RECOVERY			
	FS_WD_TOKEN	WD_TOKEN															
	FS_LIMP12_CFG	0	0	0	0	0	0	0	LIMP2_DC_CFG		LIMP2_CFG		0	0	LIMP1_CFG		0
	FS_FS0B_LIMP0_REL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	FS_ABIST	ABIST_READY	0	0	ABIST_DONE	ABIST_ONGOING	ABIST_V0MON_DIAG	ABIST_V1UVLP_DIAG	ABIST_V1MON_DIAG	ABIST_V2MON_DIAG	ABIST_V3MON_DIAG	0	ABIST_V0MON	ABIST_V1UVLP	ABIST_V1MON	ABIST_V2MON	ABIST_V3MON
	FS_SAFETY_OUTPUTS	0	RSTB_EXT	RSTB_EVT	RSTB_DRV	RSTB_SNS	RSTB_DIAG	0	FS0B_DRV	FS0B_SNS	FS0B_DIAG	0	0	LIMP0_DRV	LIMP0_SNS	LIMP0_DIAG	0
	FS_SAFETY_FLG	FCCU12_ERR_S	FCCU1_ERR_S	FCCU2_ERR_S	INIT_CRC_NOK_M	INIT_CRC_NOK_I	WD_NOK_M	WD_NOK_I	0	FCCU12_M	FCCU1_M	FCCU2_M	FCCU12_I	FCCU1_I	FCCU2_I	FCCU1_S	FCCU2_S
	FS_CRC	0	0	0	0	0	INIT_CRC_FS0B_ IMPACT	INIT_CRC_LIMP0_IMPACT	0	CRC_VALUE							

7 Writable registers

Table 5. Writable registers

Logic	Register name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default value
Main	M_SYS_CFG	-	-	-	-	-	LOCK_INIT	GO2INIT	GO2NORMAL	GO2LPON	GO2LPOFF	INT_TO_WUEN	INTB_REQ	INTB_DUR	-	MOD_CONF	MOD_EN	OTP fuse
	M_SYS1_CFG	-	-	-	VBOS2V1_SW_ALWAYS_EN	-	LOAD_OTP_BYP	SLOT_BYP	TSLOT_DOWN_CFG	-	SOFTPOR_REQ	-	DBG_EXIT	-	-	OTP_EXIT	-	OTP fuse
	M_REG_CTRL	-	-	-	BUCK_SRHSOFF		BUCK_SRHSON			-	-	V2ON_LPON	V2EN	V2DIS	V3ON_LPON	V3EN	V3DIS	OTP fuse
	M_REG_MSK	V0UV_M	V0OV_M	V1TWARN_M	V1TSD_M	V2TSD_M	V3TSD_M	V2OL_M	V1UV_M	V2UV_M	V3UV_M	V1OV_M	V2OV_M	V3OV_M	V1OC_M	V2OC_M	V3OC_M	0x0000
	M_REG1_CTRL	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	V1_OCLS_I	0x0000
	M_REG1_MSK	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	V1_OCLS_M	0x0000
	M_IO_CTRL	-	-	-	-	-	-	HVIO1HI	HVIO1LO	HVIO2HI	HVIO2LO	LVIO3HI	LVIO3LO	LVIO4HI	LVIO4LO	LVO6HI	LVO6LO	0x0000
	M_IO_TIMER_MSK	-	-	-	-	-	-	-	LDT_M	LVI5_M	LVIO4_M	LVIO3_M	HVIO2_M	HVIO1_M	-	WK2_M	WK1_M	0x0000
	M_VSUP_COM_MSK	-	-	-	-	-	-	I2C_CRC_M	I2C_REQ_M	SPI_CRC_M	SPI_CLK_M	SPI_REQ_M	-	VSHS_OV_M	VSHS_UV_M	VSUPOV_M	VSUPUV_M	0x0000
	M_IOWU_CFG	LVI5_WUCFG	LVIO4_WUCFG	LVIO3_WUCFG	Reserved	HVIO2_DGLT	HVIO1_DGLT	WK2_DGLT	WK1_DGLT	HVIO2_WUCFG		HVIO1_WUCFG		WK2_WUCFG		WK1_WUCFG		0x0005
	M_IOWU_EN	-	-	LVI5_WUEN		LVIO4_WUEN		LVIO3_WUEN		HVIO2_WUEN		HVIO1_WUEN		WK2_WUEN		WK1_WUEN		0x00FF
	M_WU1_EN	-	-	-	-	-	-	-	-	-	-	LDT_WUEN		LIN_WUEN		CAN_WUEN		0x000F
	M_TIMER1_CFG	-	-	-	-	-	-	-	TIMER1_DLY		TIMER1_ON				TIMER1_PER			0x0000
	M_TIMER2_CFG	-	-	-	-	-	-	-	TIMER2_DLY		TIMER2_ON				TIMER2_PER			0x0000
	M_TIMER3_CFG	-	-	-	-	-	-	-	TIMER3_DLY		TIMER3_ON				TIMER3_PER			0x0000
	M_PWM1_CFG	-	-	-	PWM1_DLY		PWM1_F	PWM1_DC										0x0000
	M_PWM2_CFG	-	-	-	PWM2_DLY		PWM2_F	PWM2_DC										0x0000
	M_PWM3_CFG	-	-	-	PWM3_DLY		PWM3_F	PWM3_DC										0x0000
	M_TIMER_PWM_CTRL	-	-	-	-	-	-	-	-	-	TIM1_EN	TIM2_EN	TIM3_EN	-	PWM1_EN	PWM2_EN	PWM3_EN	0x0000
	M_CS_CFG	-	-	-	-	-	-	HS_FLT_WU_FORCE	-	HVIO2_HS_SEL		HVIO1_HS_SEL		WK2_HS_SEL		WK1_HS_SEL		0x0000
	M_CS_FLG_MSK	-	-	-	-	-	-	-	HVIO2_OL_M	HVIO1_OL_M	WAKE2_OL_M	WAKE1_OL_M	-	-	-	-	-	0x0000
	M_HSx_SRC_CFG	HS4_SRC_SEL				HS3_SRC_SEL				HS2_SRC_SEL				HS1_SRC_SEL				0x0000
	M_HSx_CTRL	-	HS_VSHSUVOV_REC	HS_VSHSUV_DIS	HS_VSHSOV_DIS	-	-	-	-	-	HS4_EN	-	HS3_EN	-	HS2_EN	-	HS1_EN	0x0000
	M_HSx_MSK	-	-	-	HS4_OL_M	HS4_OC_M	-	HS3_OL_M	HS3_OC_M	HS34_TSD_M	-	HS2_OL_M	HS2_OC_M	-	HS1_OL_M	HS1_OC_M	HS12_TSD_M	0x0000
	M_AMUX_CTRL	-	-	-	-	-	-	AMUX_EN	AMUX_DIV	-	-	-	AMUX					0x0000
	M_LDT_CFG1	LDT_AFTER_RUN																0x0000
	M_LDT_CFG2	LDT_WUP_L																0x0000
	M_LDT_CFG3	-	-	-	-	-	-	-	-	LDT_WUP_H								0x0000
	M_LDT_CTRL	-	-	-	-	-	-	-	-	LDT2LP	LDT_FNCT			LDT_SEL	LDT_MODE	LDT_EN	-	0x0000
	M_CAN	-	-	-	-	-	-	CAN_MODE		-	-	CAN_FS_DIS	-	-	-	CAN_TXD_TO_I	CAN_TSD_I	0x0000
	M_LIN	-	LIN_MODE		LIN_SLOPE		LIN_FS_DIS	LIN_VSHSUV_DIS	LIN_SC	LIN_TXD_TO	-	-	-	-	LIN_SC_I	LIN_TXD_TO_I	LIN_TSD_I	0x0000
	M_CAN_LIN_MSK	-	-	LIN_FSM_STATE_S					LIN_SC_M	LIN_TXD_TO_M	LIN_TSD_M	-	CAN_FSM_STATE_S			CAN_TXD_TO_M	CAN_TSD_M	0x0000

Table 5. Writable registers*...continued*

Logic	Register name	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default value
	M_MEMORY0	MEMORY0																0x0000
	M_MEMORY1	MEMORY1																0x0000
Fail-safe	FS_I_OVUV_CFG1	-	-	-	V1MON_OV_RSTB_IMPACT	V1MON_OV_FS0B_IMPACT	V1MON_OV_LIMP0_IMPACT	V1MON_UV_RSTB_IMPACT	V1MON_UV_FS0B_IMPACT	V1MON_UV_LIMP0_IMPACT	-	V2MON_OV_RSTB_IMPACT	V2MON_OV_FS0B_IMPACT	V2MON_OV_LIMP0_IMPACT	V2MON_UV_RSTB_IMPACT	V2MON_UV_FS0B_IMPACT	V2MON_UV_LIMP0_IMPACT	OTP fuse
	FS_I_OVUV_CFG2	-	-	-	V3MON_OV_RSTB_IMPACT	V3MON_OV_FS0B_IMPACT	V3MON_OV_LIMP0_IMPACT	V3MON_UV_RSTB_IMPACT	V3MON_UV_FS0B_IMPACT	V3MON_UV_LIMP0_IMPACT	-	V0MON_OV_RSTB_IMPACT	V0MON_OV_FS0B_IMPACT	V0MON_OV_LIMP0_IMPACT	V0MON_UV_RSTB_IMPACT	V0MON_UV_FS0B_IMPACT	V0MON_UV_LIMP0_IMPACT	OTP fuse
	FS_I_FCCU_CFG	-	FCCU_CFG			FCCU2_ASSIGN			FCCU12_FLT_POL	FCCU2_FLT_POL	FCCU1_FLT_POL	FCCU2_RSTB_IMPACT	FCCU2_FS0B_ IMPACT	FCCU2_LIMP0_IMPACT	FCCU1_RSTB_IMPACT	FCCU1_FS0B_ IMPACT	FCCU1_LIMP0_IMPACT	0X103F
	FS_I_FSSM_CFG	-	EXT_RSTB_DIS	RSTB8S_DIS	RSTB_DUR	LIMP0_SC_RSTB_IMPACT	EXTRSTB_FS0B_IMPACT	FS0B_SC_RSTB_IMPACT	FLT_ERR_LIMIT		FLT_MID_RSTB_IMPACT	FLT_MID_FS0B_IMPACT	FLT_MID_LIMP0_IMPACT	FLT_ERR_CNT				OTP fuse
	FS_I_WD_CFG	-	WD_RSTB_IMPACT	WD_FS0B_IMPACT	WD_LIMP0_IMPACT	WD_DIS_LPON	WD_RFR_LIMIT		WD_ERR_LIMIT		-	-	-	-	-	-	-	0x7080
	FS_WDW_CFG	-	-	-	-	WDW_REC_EN	WDW_EN	-	WDW_PERIOD				-	WDW_RECOVERY				0x01AB
	FS_WD_ANSWER	WD_ANSWER																0x0000
	FS_LIMP12_CFG	-	-	-	-	-	-	-	LIMP2_DC_CFG		LIMP2_CFG		LIMP2_REQ	-	LIMP1_CFG		LIMP1_REQ	OTP fuse
	FS_FS0B_LIMP0_REL	RELEASE_FS0B_LIMP0																0x0000
	FS_ABIST	-	LAUNCH_ABIST	CLEAR_ABIST	-	-	-	-	-	-	-	-	ABIST_V0MON	ABIST_V1UVLP	ABIST_V1MON	ABIST_V2MON	ABIST_V3MON	0x0000
	FS_SAFETY_OUTPUTS	-	-	-	-	-	-	RSTB_REQ	-	-	-	FS0B_REQ	-	-	-	-	LIMP0_REQ	0x0000
	FS_SAFETY_FLG	-	-	-	INIT_CRC_NOK_M	-	WD_NOK_M	-	-	FCCU12_M	FCCU1_M	FCCU2_M	-	-	-	-	-	0x0000
	FS_CRC	-	INIT_CRC_REQ	-	-	-	INIT_CRC_FS0B_IMPACT	INIT_CRC_LIMP0_IMPACT	-	CRC_VALUE								0x0000

8 FS0B and/or LIMP0 release calculation procedure

When the fail-safe output FS0B is asserted low by the device because of a fault, or after a power up, some conditions must be validated before allowing the FS0B pin to be released by the device.

These conditions are:

✓ No fault affecting FS0B reported

✓ Fault error counter equal to zero

✓ Device in Normal mode

✓ Device not in Debug mode and not in INIT mode

✓ FS_FS0B_LIMP0_REL register filled with the correct value, depending on current WD_TOKEN[15:0] value as per Table 6. Refer to Table 56 from the FS23 data sheet:

Table 6. FS0B and/or LIMP0 release commands

FS_FS0B_LIMP0_REL[15:0]	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Release FS0B	**0**	**1**	**1**	**NOT(WD_TOKEN[0:12])**												
Release LIMP0	**1**	**1**	**0**	**NOT(WD_TOKEN[3:15])**												
Release both FS0B and LIMP0	**1**	**0**	**1**	**NOT(WD_TOKEN[0:6])**							**NOT(WD_TOKEN[10:15])**					

9 Watchdog answer procedure[1]

[1] Refer to Section 20.2 from the FS23 data sheet.

The watchdog uses two keys, 0x5AB2 (default value after POR) and 0xD564 to validate the answer. The key is stored in the WD_TOKEN register, and is changed alternatively after each good WD refresh.

The MCU reads the WD_TOKEN register and writes the correct answer (WD_TOKEN register value) through the SPI/I2C in WD_ANSWER register, in the right timing. The WD error counter is incremented when the answer is wrong or not given at the right moment, or not given at all at the end of the watchdog period. Refer to Table 35 from the FS23 data sheet.

The first good watchdog refresh closes the INIT phase. This first good watchdog refresh is sent by the MCU in less than 256 ms (default period duration). Then the watchdog window is running and the MCU must refresh the watchdog every period.

Table 7. Watchdog answer and refresh validation

SPI / I²C	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

10 SPI/I2C CRC calculation procedure

An 8-bit CRC is required for each write and read SPI command. Computation of a CRC is derived from the mathematics of polynomial division, modulo two.

The CRC parameters are:

- Polynomial: x^8+x^4+x^3+x^2+1 (identified by **0x1D**)
- Seed: **0xFF**.

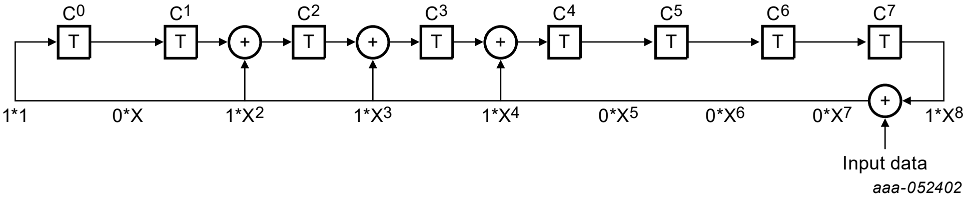


Figure 2. CRC encoder example

10.1 For SPI communication

SPI message construction includes the register address, the read/write bit, data, and CRC. Refer to Tables 65 and 66 from the FS23 data sheet. The bit B32 must be set to 1 to execute a write command, and to 0 to execute a read command.

Table 8. SPI write command message construction

	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
MOSI	Register address [6:0]							R/W	Write data [15:8]							
MISO	General status flag								Register content before Write [15:8]							
	B15	**B14**	**B13**	**B12**	**B11**	**B10**	**B9**	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
MOSI	Write data [7:0]								CRC[7:0]							
MISO	Register content before Write [7:0]								CRC[7:0] - response							

Table 9. SPI read command message construction

	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
MOSI	Register address [6:0]							R/W	0x00							
MISO	General status flag								Read data [15:8]							
	B15	**B14**	**B13**	**B12**	**B11**	**B10**	**B9**	**B8**	**B7**	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**
MOSI	0x00								CRC[7:0]							
MISO	Read data [7:0]								CRC[7:0] - response							

When using SPI communication, the input for CRC calculation is a 24-bit word composed of the register address, the read/write bit, and data.

10.2 For I^2C communication

I²C message construction includes the device address, read/write bit, register address, data, and CRC. Refer to Table 61 from the FS23 data sheet. The bit B32 must be set to 0 to execute a write command, and to 1 to execute a read command.

Table 10. I²C message construction

								B39	B38	B37	B36	B35	B34	B33	B32
								ID[6:0]							**R/W**
								Device address							R/W
B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
0	**ADR[6:0]**							**DATA[15:8]**							
0	Register address							Data MSB							
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
DATA[7:0]								**CRC[7:0]**							
Data LSB								CRC							

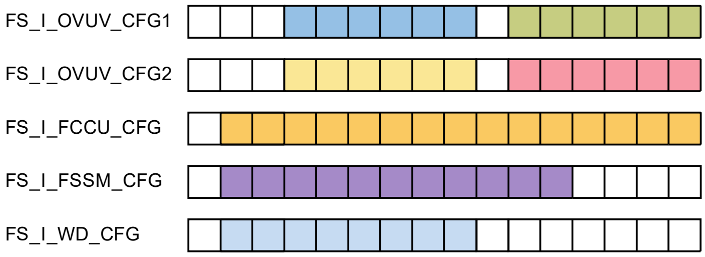
When using I²C communication, the input for CRC calculation is a 32-bit word composed of device address, read/write bit, register address and data.

11 INIT CRC calculation procedure

INIT fail-safe registers are protected by a CRC. The same polynomial and seed used for SPI/I²C CRC are necessary to compute this INIT CRC: The polynomial is x^8+x^4+x^3+x^2+1 (identified by 0x1D) with the seed value of 0xFF.

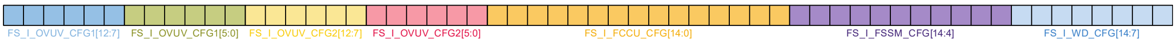
Three steps are required to compute INIT CRC:

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



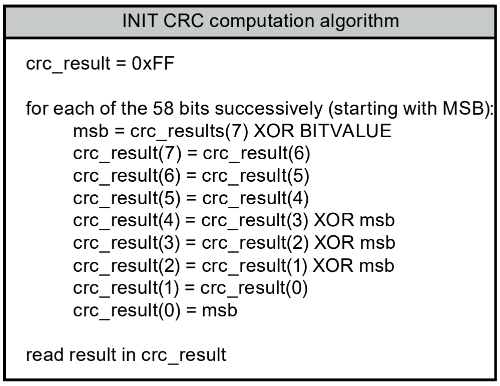
aaa-052403

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



aaa-052404

3. Compute INIT CRC bitwise using 0x1D polynomial. The figure below gives an example for **bitwise** CRC computation algorithm.



INIT CRC computation algorithm

```
crc_result = 0xFF

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

read result in crc_result
```

aaa-052405

12 Revision history

Revision history

Document ID	Release date	Description
AN14041 v.2.0	23 January 2025	• Global editing for grammar and style. • Moved document from secure files to public access on nxp.com. • Section 12 was relocated from the front of this document to the end to conform with NXP's document content hierarchy. • Updated Legal information
AN14041 v.1.0	13 September 2023	Initial version

13 References

Documentation

[1] FS23 application note – product guidelines, nxp.com

[2] FS23 data sheet, nxp.com

Software resources

[3] FS23 AUTOSAR software drivers, nxp.com

[4] Real-Time Drivers (RTD) general information, nxp.com

Evaluation resources

[5] FS23 graphical user interface (GUI), revision 3.1.382, nxp.com

[6] FS23 user manual for socketed / soldered Buck / soldered LDO EVB, nxp.com

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

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: https://www.nxp.com

Date of release: 23 January 2025
Document identifier: AN14041