AN12386

SPI Communication Procedure Recommendations for the FXLS9xxxx

Rev. 1.2 — 8 January 2021 Application note

1 Introduction

The purpose of this document is to describe the SPI power up, initialization and normal mode procedures for the FXLS9xxxx single and dual channel inertial sensors.

2 Applicable Parts

This document applies to the following NXP sensors:

Table 1. Applicable parts

FXLS9xxxx	uThornapple	Dual Channel SPI Inertial Sensor
FXLS9xxxx	uLaurel	Single Channel SPI Inertial Sensor

3 Definition List

Table 2. Definition list

Term	Definition
Analog Self-Test	A method to test the acceleration signal chain by electrostatically deflecting the transducer proof mass and measuring the device output.
Digital Self-Test	A method to test the digital portion of the acceleration signal chain by forcing a value or a sequence of values at the output of the analog to digital converter and measuring the device output.
DSP	Digital Signal Processing Block
MOSI	Master Output, Slave Input. The MOSI signal is output from the master to all slaves on the bus.
MISO	Master Input, Slave Output. The MISO signal is output from one slave to the master.
POR	Power On Reset
SCLK	Serial Clock. The SCLK signal is output from the master to all the slaves on the bus.
SS_B	Slave Select Bar. The SS_B signal is an active low signal output from the master to each slave independently on the bus.
SPI	Serial Peripheral Interface, a full duplex, synchronous serial interface. The FXLS9xxxx and FXLSAxxxx devices operate using a 4-wire SPI.

4 Further Assistance

For further assistance please contact a local NXP sales representative.

5 References

- FXLS9xxxx data sheet, latest revision: uThornapple data sheet
- FXLS9xxxx data sheet, latest revision: uLaurel data sheet

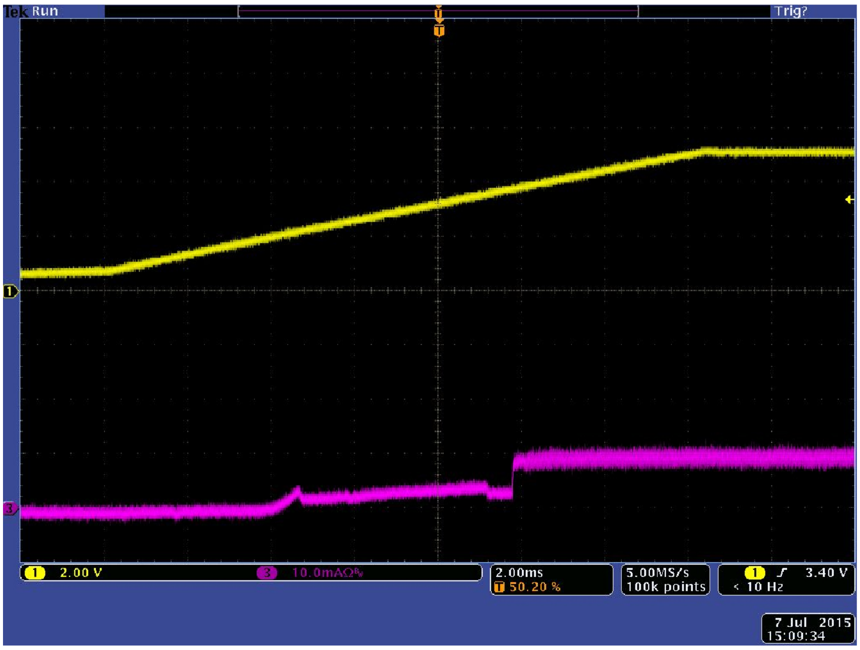
6 Revision History

Table 3. Revision history

Rev. No.	Date	Description
0.1	20141028	• Initial Release
0.2	20150118	• Updated based on customer feedback. • Added Flowcharts for each procedure.
0.3	20150608	• Updated the startup sequence and overall flow. • Updated the soft reset sequence and the status verification sequence. • Updated the self-test flows. • Added expected responses for all section up to the self-test configuration section.
0.4	20150714	• Added expected responses for all sections. • Added the 3-Bit and 4-Bit CRCs for each command. • Clarified all self-test procedures. • Added the self-test accuracy test procedure. • Added the continuous oscillator verification procedure. • Added offset and self-test limit calculations and examples. • Added all delay time references for offset and self-test procedures. • Added reference from Overview Flowcharts to Detailed Flowcharts. • Added an example Self-Test Output.
0.5	20150714	• Updates and corrections after internal review.
1.0	20190313	• Initial revision. Created from AN5358
1.1	20190517	• Analog Self-test sequence modified for uTA 3.1.
1.2	20210108	• Updated Analog Self-Test sequence for uTA 3.2/ uLA 2.1

7 Apply Power to the FXLS9xxxx

Power must be applied to the FXLS9xxxx with the ramp rates specified in the data sheet. The device is verified to properly startup with ramp rates from 10 V/s to 10 V/µs. The device is specified to work with either a 3.3 V supply or a 5 V supply.



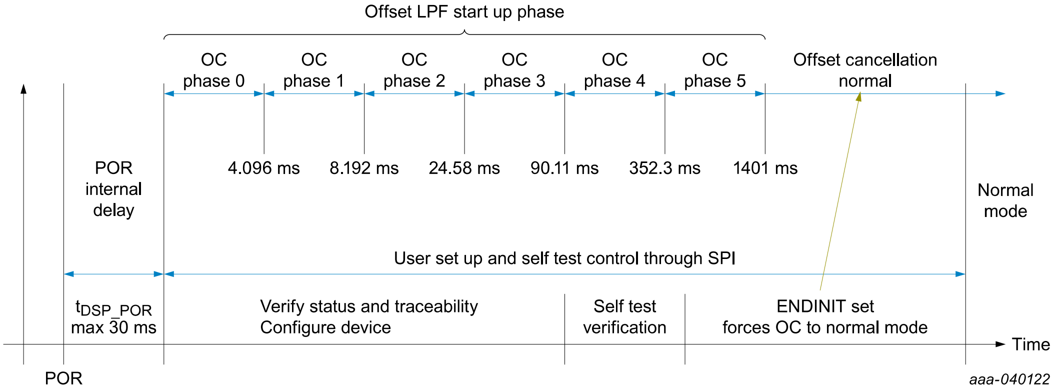
aaa-040121

Figure 1. Example SPI supply ramp

8 Initialize and Configure the Devices

The following sections describe the recommended SPI commands to initialize, configure and test an FXLS9xxxx (uThornapple) device.

Figure 2 shows an example timing diagram for SPI Start Up and initialization and how it compares to the internal offset cancellation start up. Figure 3 shows a high level flow chart of the SPI sequence.



aaa-040122

Figure 2. Example SPI initialization timing diagram

Note: CRC computation provided in this document are using the below configuration:

Table 4. 8-bit CRC

Method	Value
Polynomial	100101111
Non Direct Seeds	11111111

Table 5. 4-bit CRC

Method	Value
Polynomial	10001
Non Direct Seeds	1010

Table 6. 3-bit CRC

Method	Value
Polynomial	1011
Non Direct Seeds	111

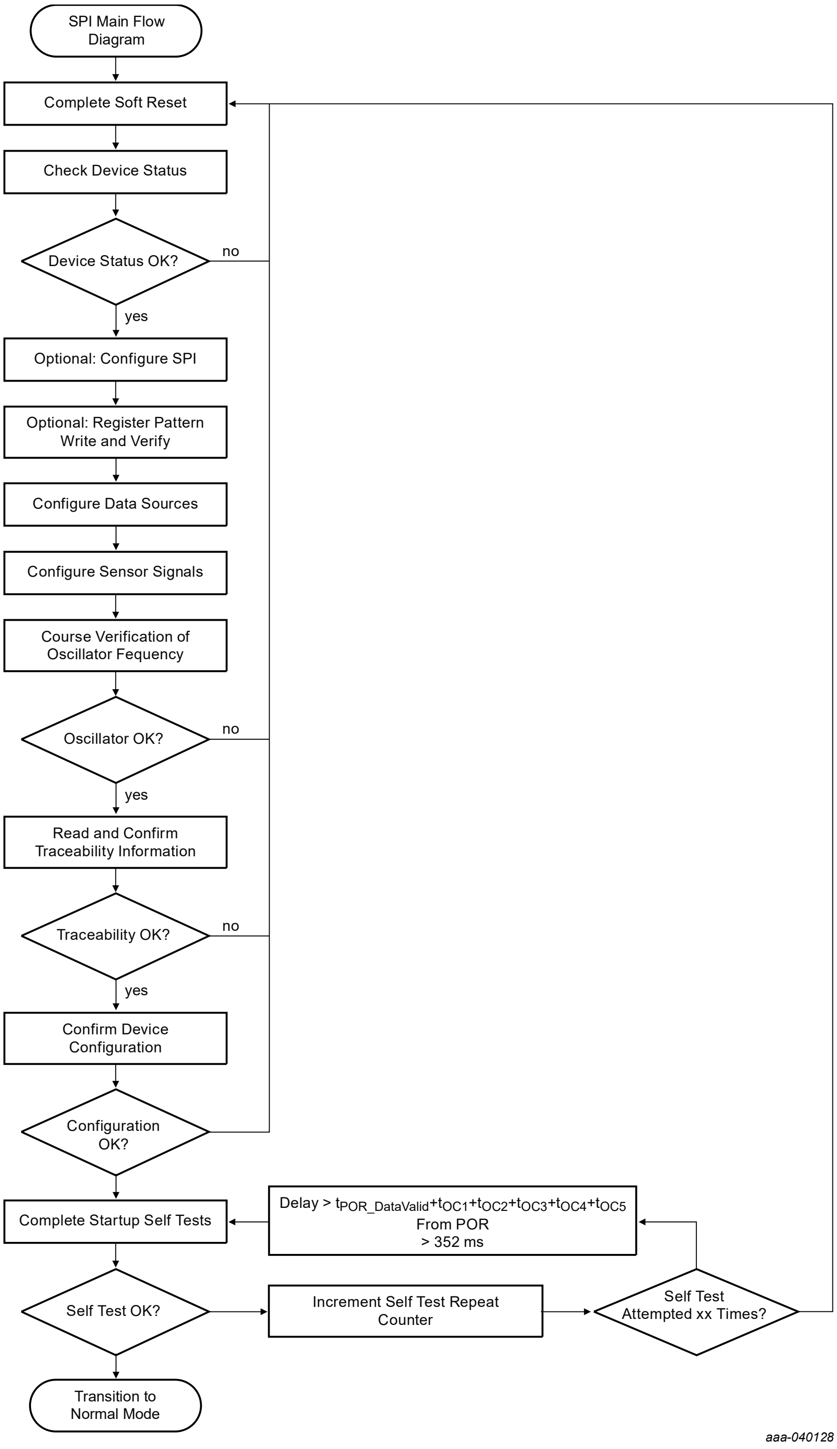
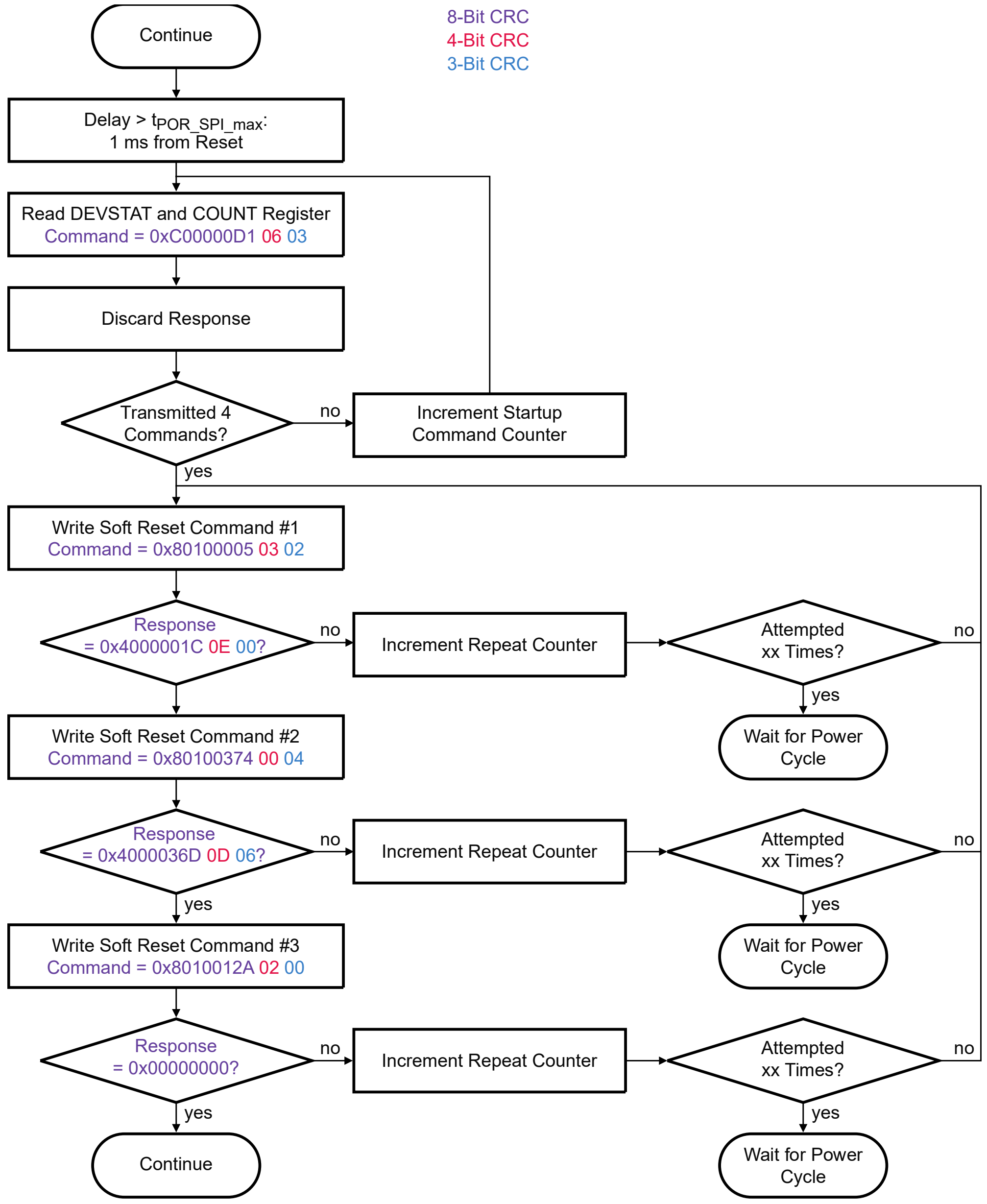

SPI Main Flow Diagram
Complete Soft Reset
Check Device Status
Device Status OK?
no
yes
Optional: Configure SPI
Optional: Register Pattern Write and Verify
Configure Data Sources
Configure Sensor Signals
Course Verification of Oscillator Fequency
Oscillator OK?
no
yes
Read and Confirm Traceability Information
Traceability OK?
no
yes
Confirm Device Configuration
Configuration OK?
Complete Startup Self Tests
Delay > tPOR_DataValid+tOC1+tOC2+tOC3+tOC4+tOC5 From POR > 352 ms
Self Test OK?
Increment Self Test Repeat Counter
Self Test Attempted xx Times?
Transition to Normal Mode
aaa-040128

Figure 3. High level SPI flow chart for the FXLS9xxxx

8.1 Complete a soft reset

The first step is to complete a soft reset of the device to ensure consistent power up operation. This is accomplished by sending three consecutive write commands to the DEVLOCK_WR register of each device. The proper sequence is shown in Section 8.1.



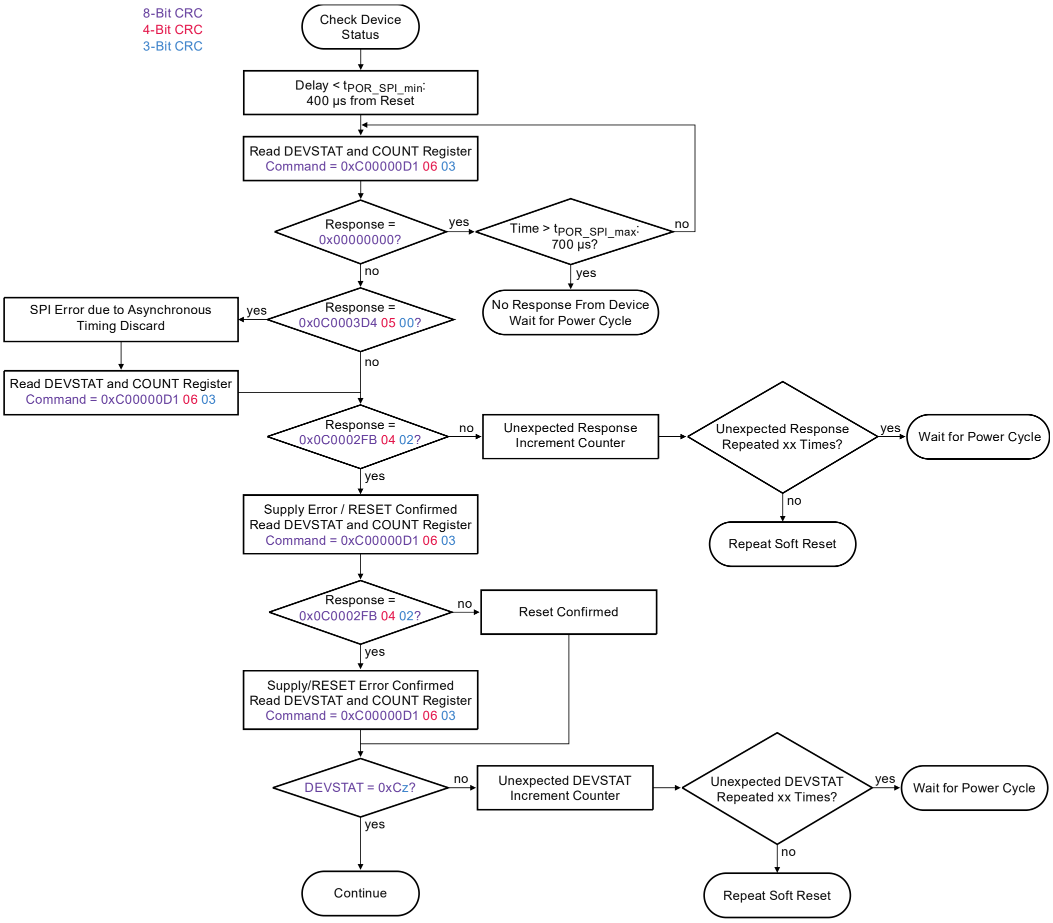
Note:
The expected response to the 3rd soft reset command is “No Response , meaning the device is in reset and not driving te MISO node. The actual received command will vary depending the on the impedances on the MISO node and any pull-ups or pull-downs included.

aaa-040129

Figure 4. Complete a soft reset

8.2 Confirm device status

The next step is to confirm proper SPI communication and the expected status of the device by following the procedure shown in Figure 5. The DEVSTAT register mapping is shown in Figure 6.

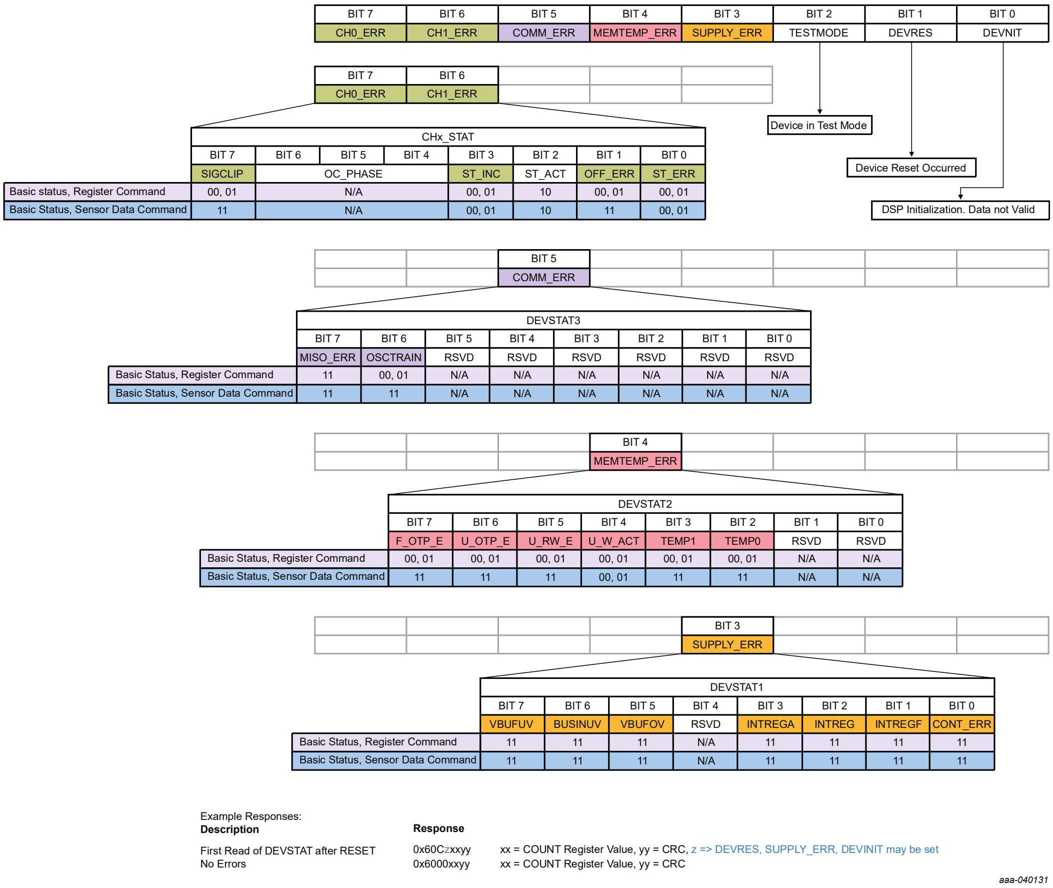

8-Bit CRC
4-Bit CRC
3-Bit CRC
Check Device Status
Delay < tPOR_SPI_min: 400 µs from Reset
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
Response = 0x00000000?
yes
Time > tPOR_SPI_max: 700 µs?
no
yes
No Response From Device
Wait for Power Cycle
no
Response = 0x0C0003D4 05 00?
yes
SPI Error due to Asynchronous Timing Discard
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
no
Response = 0x0C0002FB 04 02?
no
Unexpected Response Increment Counter
Unexpected Response Repeated xx Times?
yes
Wait for Power Cycle
no
Repeat Soft Reset
yes
Supply Error / RESET Confirmed
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
Response = 0x0C0002FB 04 02?
no
Reset Confirmed
yes
Supply/RESET Error Confirmed
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
DEVSTAT = 0xCz?
no
Unexpected DEVSTAT Increment Counter
Unexpected DEVSTAT Repeated xx Times?
yes
Wait for Power Cycle
no
Repeat Soft Reset
yes
Continue

Notes:

- Prior to $t_{POR\_SPI\_max}$, SPI commands may be transmitted to the device and responses ignored
- If commands are sent prior to $t_{POR\_SPI\_max}$, it is recommended to use benign commands like a read of the PDCM_RSPST0_L & PDCM_RSPST0_H registers.
- Responses to all SPI commands transmitted after $t_{POR\_SPI\_max}$, must be evaluated to confirm expected operation

aaa-040130

Figure 5. Confirm device status



BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
CH0_ERR	CH1_ERR	COMM_ERR	MEMTEMP_ERR	SUPPLY_ERR	TESTMODE	DEVRES	DEVNIT

BIT 7	BIT 6
CH0_ERR	CH1_ERR

	CHx_STAT							
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	SIGCLIP	OC_PHASE			ST_INC	ST_ACT	OFF_ERR	ST_ERR
Basic status, Register Command	00, 01	N/A			00, 01	10	00, 01	00, 01
Basic Status, Sensor Data Command	11	N/A			00, 01	10	11	00, 01

BIT 5
COMM_ERR

	DEVSTAT3							
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	MISO_ERR	OSCTRAIN	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD
Basic Status, Register Command	11	00, 01	N/A	N/A	N/A	N/A	N/A	N/A
Basic Status, Sensor Data Command	11	11	N/A	N/A	N/A	N/A	N/A	N/A

BIT 4
MEMTEMP_ERR

	DEVSTAT2							
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	F_OTP_E	U_OTP_E	U_RW_E	U_W_ACT	TEMP1	TEMP0	RSVD	RSVD
Basic Status, Register Command	00, 01	00, 01	00, 01	00, 01	00, 01	00, 01	N/A	N/A
Basic Status, Sensor Data Command	11	11	11	00, 01	11	11	N/A	N/A

BIT 3
SUPPLY_ERR

	DEVSTAT1							
	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	VBUFUV	BUSINUV	VBUFOV	RSVD	INTREGA	INTREG	INTREGF	CONT_ERR
Basic Status, Register Command	11	11	11	N/A	11	11	11	11
Basic Status, Sensor Data Command	11	11	11	N/A	11	11	11	11

Example Responses:

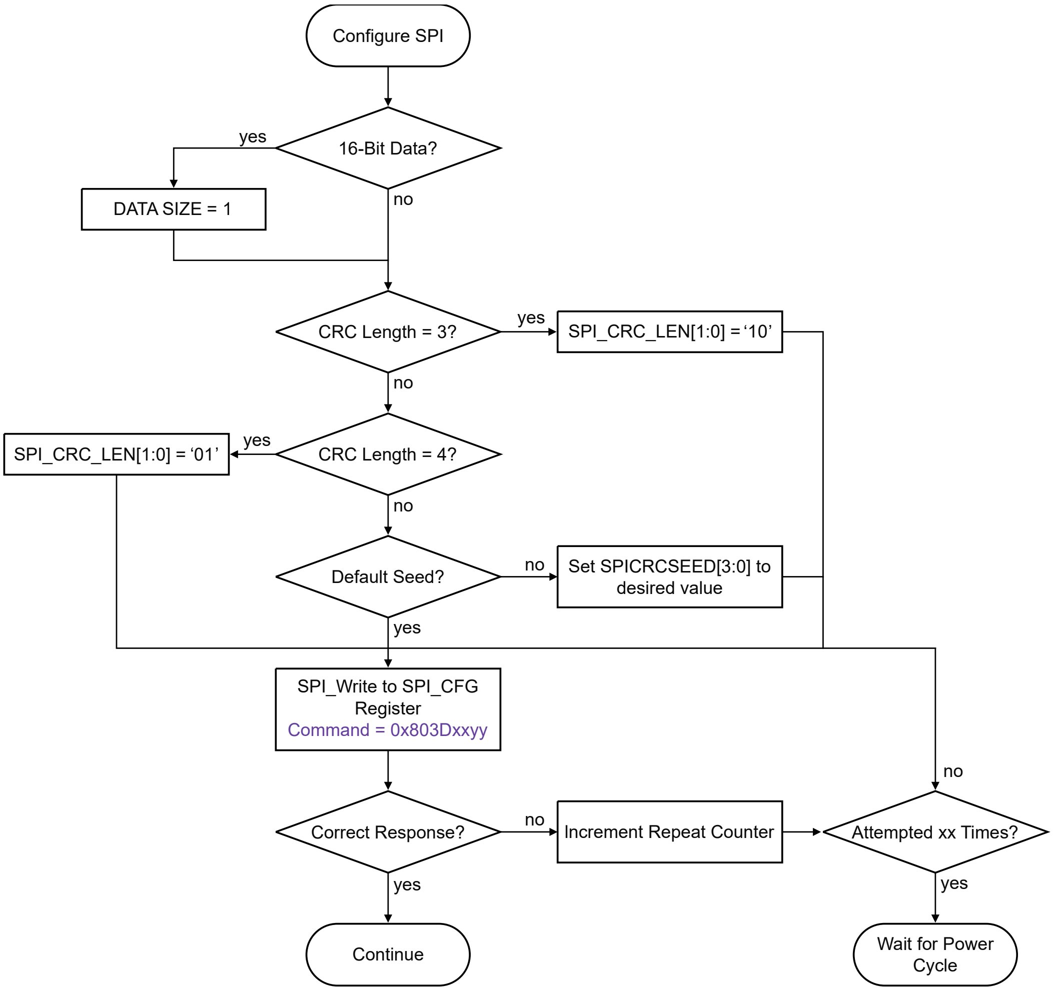
Description	**Response**	
First Read of DEVSTAT after RESET	0x60Czxxyy	xx = COUNT Register Value, yy = CRC, z => DEVRES, SUPPLY_ERR, DEVINIT may be set
No Errors	0x6000xxyy	xx = COUNT Register Value, yy = CRC

aaa-040131

Figure 6. DEVSTAT register mapping

8.3 Configure SPI communication

The next step is to optionally configure the SPI communication for the desired sensor data field size and the desired error detection method including CRC length, CRC seed and Message Counter. This step is not necessary if the default SPI communication settings will be used: 8-bit CRC and 12-bit Sensor Data. The flowchart in Figure 7 shows some of the typical SPI configurations.



Examples:

Configuration	Command	Response	
12-Bit Data, 3-Bit CRC, Default Seed	0x803D2085 0E 03	0x60002080 0E 07	8-Bit CRC
12-Bit Data, 4-Bit CRC, Default Seed	0x803D1058 0D 01	0x6000105D 0D 05	4-Bit CRC
12-Bit Data, 8-Bit CRC, Default Seed	0x803D00F6 0C 06	0x600000F3 0C 02	3-Bit CRC
16-Bit Data, 3-Bit CRC, Default Seed	0x803D6063 0A 02	0x60006066 0A 06	
16-Bit Data, 4-Bit CRC, Default Seed	0x803D50BE 09 00	0x600050BB 0D 04	
16-Bit Data, 8-Bit CRC, Default Seed	0x803D4010 08 07	0x60004015 08 03	

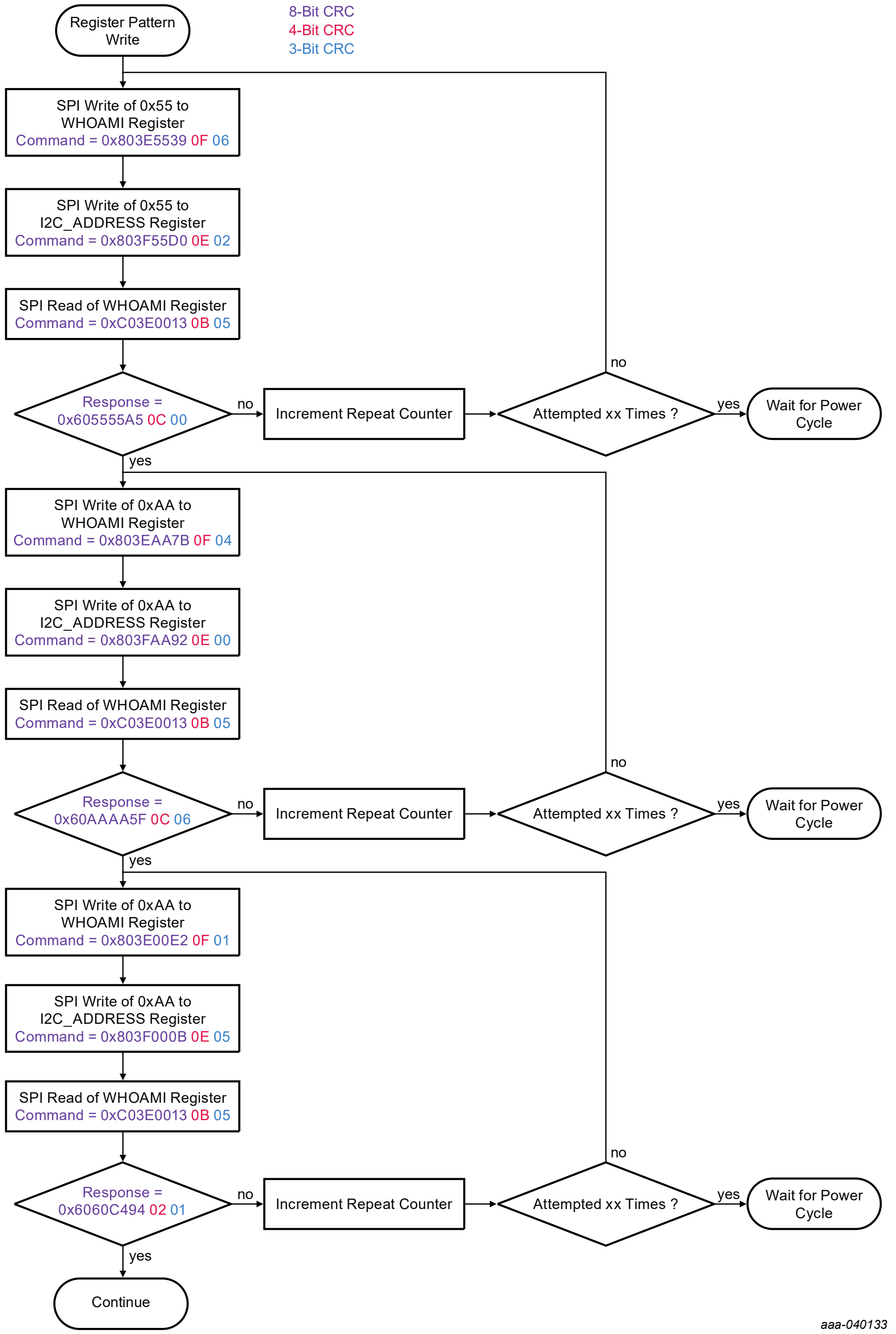
aaa-040132

Figure 7. Configure SPI

8.4 Optional complete register pattern write verification

The next step is to complete a register pattern write verification. This step is optional and not required to meet the diagnostic coverage as documented in the FMEDA.

The recommended procedure for register pattern write verification is shown in Figure 8. In this example, the WHOAMI and I2C_ADDRESS registers are used for pattern writing as they are unused for the SPI application. Other registers can be used as long as the function for the register being written to is considered.

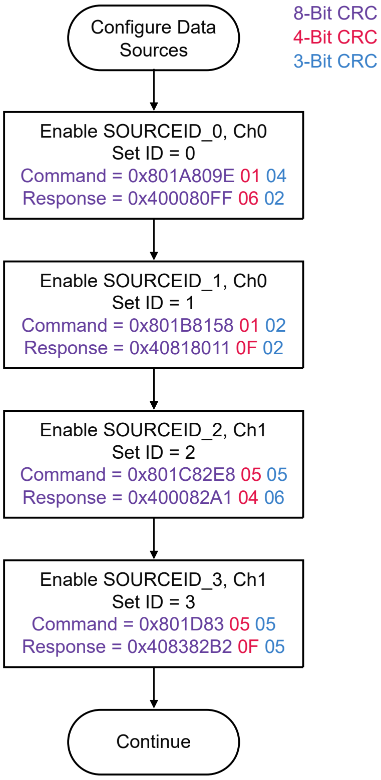

8-Bit CRC
4-Bit CRC
3-Bit CRC
Register Pattern Write
SPI Write of 0x55 to WHOAMI Register
Command = 0x803E5539 0F 06
SPI Write of 0x55 to I2C_ADDRESS Register
Command = 0x803F55D0 0E 02
SPI Read of WHOAMI Register
Command = 0xC03E0013 0B 05
Response = 0x605555A5 0C 00
no
Increment Repeat Counter
Attempted xx Times ?
no
yes
Wait for Power Cycle
yes
SPI Write of 0xAA to WHOAMI Register
Command = 0x803EAA7B 0F 04
SPI Write of 0xAA to I2C_ADDRESS Register
Command = 0x803FAA92 0E 00
SPI Read of WHOAMI Register
Command = 0xC03E0013 0B 05
Response = 0x60AAAA5F 0C 06
no
Increment Repeat Counter
Attempted xx Times ?
no
yes
Wait for Power Cycle
yes
SPI Write of 0xAA to WHOAMI Register
Command = 0x803E00E2 0F 01
SPI Write of 0xAA to I2C_ADDRESS Register
Command = 0x803F000B 0E 05
SPI Read of WHOAMI Register
Command = 0xC03E0013 0B 05
Response = 0x6060C494 02 01
no
Increment Repeat Counter
Attempted xx Times ?
no
yes
Wait for Power Cycle
yes
Continue

aaa-040133

Figure 8. Complete register pattern write verification

8.5 Configure the data sources

The next step is to configure the devices for the desired data sources and source identifiers. Figure 9 shows one example configuration.



Notes:
- For SPI mode, Valid Source IDs are 0x0 to 0x7
- No SPI sensor data requests are implemented for Source IDs 0x8 to 0xF
- Each channel can have 0, 1 or 2 sources:
    - The first source is mapped to data type 0
    - The second source is mapped to data type 1

aaa-040134

Figure 9. Configure the data sources

Each channel of the FXLS9xxxx devices has the capability for two independently configurable data sources. Figure 10 shows a pictorial mapping of the sources to their Source Identifiers and associated data.

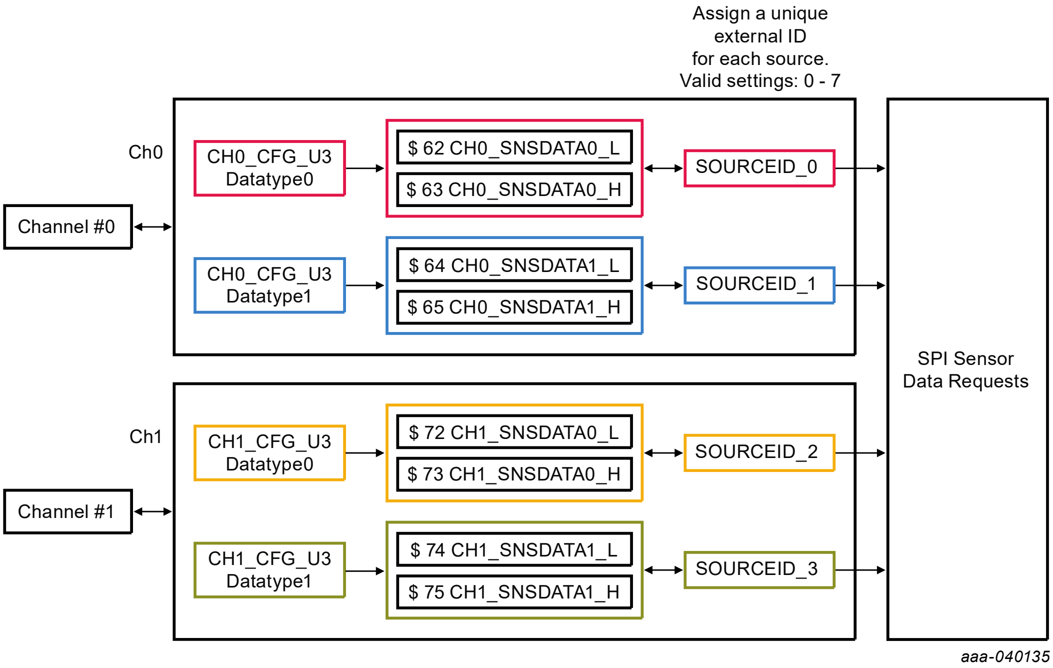


Figure 10. Data source mapping diagram

The sources are enabled and the associated Source Identifiers are set using the registers listed in the Table 7.

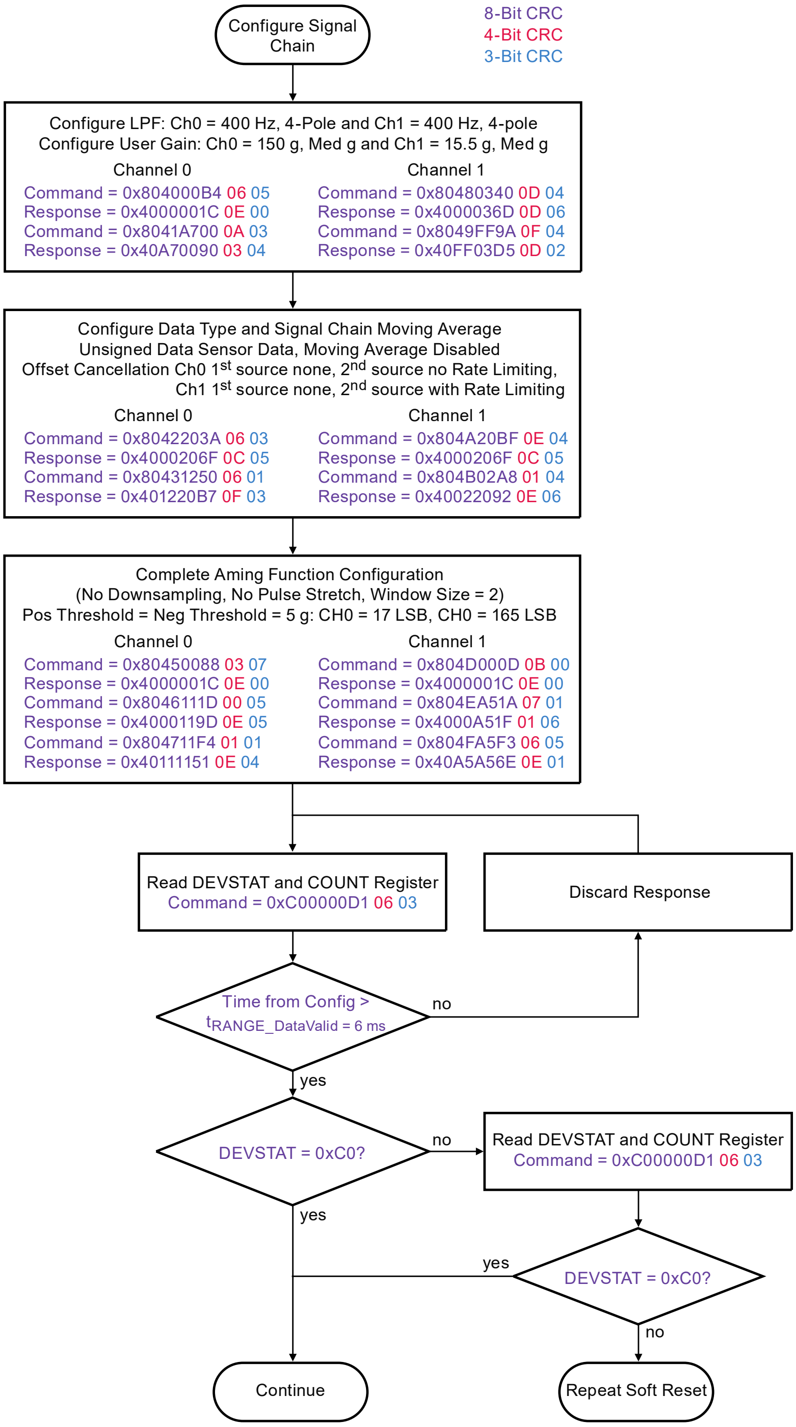
Table 7. Source Identifier registers

Register Address	Register Name	D7	D6	D5	D4	D3	D2	D1	D0
$1A	SOURCEID_0	SID0_EN Enable Channel 0 Source 0, Datatype 0	PDCM_FORMAT[2:0] Not Applicable for SPI mode			SOURCEID_0[3:0] System level source identifier for Channel 0, Datatype 0 Notes: • Each source identifier value for the device must be unique or the device will transmit error messages for the repeated identifier. • For SPI mode, valid source ids are 0x0 to 0x7. No SPI sensor data requests are implemented for 0x8 to 0xF.			
$1B	SOURCEID_1	SID1_EN Enable Channel 0 Source 1, Datatype 1	Reserved	Reserved	Reserved	SOURCEID_1[3:0] System level source identifier for Channel 0, Datatype 1 Notes: • Each source identifier value for the device must be unique or the device will transmit error messages for the repeated identifier. • For SPI mode, valid source ids are 0x0 to 0x7. No SPI sensor data requests are implemented for 0x8 to 0xF.			

Register Address	Register Name	D7	D6	D5	D4	D3	D2	D1	D0
$1C	SOURCEID_2	SID2_EN Enable Channel 1 Source 2, Datatype 0	Reserved	Reserved	Reserved	SOURCEID_2[3:0] System level source identifier for Channel 1, Datatype 0 Notes: • Each source identifier value for the device must be unique or the device will transmit error messages for the repeated identifier. • For SPI mode, valid source ids are 0x0 to 0x7. No SPI sensor data requests are implemented for 0x8 to 0xF.			
$1D	SOURCEID_3	SID3_EN Enable Channel 1 Source 3, Datatype 1	Reserved	Reserved	Reserved	SOURCEID_3[3:0] System level source identifier for Channel 1, Datatype 1 Notes: • Each source identifier value for the device must be unique or the device will transmit error messages for the repeated identifier. • For SPI mode, valid source ids are 0x0 to 0x7. No SPI sensor data requests are implemented for 0x8 to 0xF.			

8.6 Configure the sensor signal chain

The next step is to configure the sensor signal chain. Figure 11 shows an example configuration. The delay times included are specific to the configuration selected and must be adjusted if alternative filters are selected. The self-test verification calculations are also specific to the configuration selected and must be adjusted if alternative gains are used.


Configure Signal Chain
8-Bit CRC
4-Bit CRC
3-Bit CRC
Configure LPF: Ch0 = 400 Hz, 4-Pole and Ch1 = 400 Hz, 4-pole
Configure User Gain: Ch0 = 150 g, Med g and Ch1 = 15.5 g, Med g
Channel 0
Command = 0x804000B4 06 05
Response = 0x4000001C 0E 00
Command = 0x8041A700 0A 03
Response = 0x40A70090 03 04
Channel 1
Command = 0x80480340 0D 04
Response = 0x4000036D 0D 06
Command = 0x8049FF9A 0F 04
Response = 0x40FF03D5 0D 02
Configure Data Type and Signal Chain Moving Average
Unsigned Data Sensor Data, Moving Average Disabled
Offset Cancellation Ch0 1st source none, 2nd source no Rate Limiting,
Ch1 1st source none, 2nd source with Rate Limiting
Channel 0
Command = 0x8042203A 06 03
Response = 0x4000206F 0C 05
Command = 0x80431250 06 01
Response = 0x401220B7 0F 03
Channel 1
Command = 0x804A20BF 0E 04
Response = 0x4000206F 0C 05
Command = 0x804B02A8 01 04
Response = 0x40022092 0E 06
Complete Aming Function Configuration
(No Downsampling, No Pulse Stretch, Window Size = 2)
Pos Threshold = Neg Threshold = 5 g: CH0 = 17 LSB, CH0 = 165 LSB
Channel 0
Command = 0x80450088 03 07
Response = 0x4000001C 0E 00
Command = 0x8046111D 00 05
Response = 0x4000119D 0E 05
Command = 0x804711F4 01 01
Response = 0x40111151 0E 04
Channel 1
Command = 0x804D000D 0B 00
Response = 0x4000001C 0E 00
Command = 0x804EA51A 07 01
Response = 0x4000A51F 01 06
Command = 0x804FA5F3 06 05
Response = 0x40A5A56E 0E 01
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
Discard Response
Time from Config > tRANGE_DataValid = 6 ms
no
yes
DEVSTAT = 0xC0?
no
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
yes
yes
DEVSTAT = 0xC0?
no
Continue
Repeat Soft Reset

aaa-040136

Figure 11. Configure the sensor signal chain

8.6.1 Signal chain low pass filter selection

The signal chain low pass filter is selected by a combination of the LPF bits and the SAMPLERATE bits in the CHx_CFG_U1 register as shown in the datasheet. The LPF selection table is shown below.

Table 8. Signal chain low pass filter selection

LPF[3]	LPF[2]	LPF[1]	LPF[0]	Low Pass Filter Type		
				SAMPLERATE = 00, 01	SAMPLERATE = 10	SAMPLERATE = 11
				16 µs	32 µs	64 µs
0	0	0	0	400 Hz, 4-Pole	200 Hz, 4-Pole	100 Hz, 4-Pole
0	0	0	0	400 Hz, 3-Pole	200 Hz, 3-Pole	100 Hz, 3-Pole
0	0	1	0	400 Hz, 4-Pole	200 Hz, 4-Pole	100 Hz, 4-Pole
0	0	1	1	400 Hz, 3-Pole	200 Hz, 3-Pole	100 Hz, 3-Pole
0	1	0	0	325 Hz, 3-Pole	162.5 Hz, 3-Pole	81.25 Hz, 3-Pole
0	1	0	1	370 Hz, 2-Pole	185 Hz, 2-Pole	92.5 Hz, 2-Pole
0	1	1	0	180 Hz, 2-Pole	90 Hz, 2-Pole	45 Hz, 2-Pole
0	1	1	1	100 Hz, 2-Pole	50 Hz, 2-Pole	25 Hz, 2-Pole
1	0	0	0	1500 Hz, 4-Pole	750 Hz, 4-Pole	375 Hz, 4-Pole
1	0	0	1	500 Hz, 3-Pole	250 Hz, 3-Pole	125 Hz, 3-Pole
1	0	1	0	800 Hz, 4-Pole	400 Hz, 4-Pole	200 Hz, 4-Pole
1	0	1	1	1200 Hz, 4-Pole	600 Hz, 4-Pole	300 Hz, 4-Pole
1	1	0	0	120 Hz, 3-Pole	60 Hz, 3-Pole	30 Hz, 3-Pole
1	1	0	1	20 kHz, 2-Pole	10 kHz, 2-Pole	5 kHz, 2-Pole
1	1	1	0	120 Hz, 2-Pole	60 Hz, 2-Pole	30 Hz, 2-Pole
1	1	1	1	50 Hz, 4-Pole	25 Hz, 4-Pole	12.5 Hz, 4-Pole

8.6.2 Signal chain user gain selection

The signal chain user gain is selected by a combination of the U_SNS_SHIFT bits in the CHx_CFG_U1 register and the U_SNS_MULT bits in the CHx_CFG_U2 register. The equation and some example user range and sensitivities are included in the datasheet. The process and equations for determining the U_SNS_SHIFT and U_SNS_MULT settings from desired range and sensitivity values is also listed below along with a medium g example.

1. Determine the overall sensitivity adjustment factor:
    - Desired Typical User Range = ±100 g with 12-bit data
    - Calculate Desired Sensitivity

$$Sense_{Typical\ Desired} = \frac{2^{11}\text{-}1}{Range_{Typical\ Desired}} = \frac{2047}{100} = 20.47\ \ LSB\Big/g$$

2. Calculate the required sensitivity adjustment for a medium g device

$$SENSE_{Adjust\ Total} = \frac{Sense_{Typical\ Desired}}{Sense_{Typical\ NXP\ Trim}} = \frac{20.47}{33.0161} = 0.6200$$

3. Determine the best U_SNS_SHIFT setting:

Table 9. Signal chain user gain selection

Sense$_{AdjustTotal}$	U_SNS_SHIFT Gain	U_SNS_SHIFT Setting
Sense$_{AdjustTotal}$ < 0.25	Invalid Range	Invalid Range
0.25 ≤ Sense$_{AdjustTotal}$ < 0.50	0.25	00
0.50 ≤ Sense$_{AdjustTotal}$ < 1.00	0.50	01
1.00 ≤ Sense$_{AdjustTotal}$ < 2.00	1.00	10
2.00 ≤ Sense$_{AdjustTotal}$ < 4.00	2.00	11
4.00 ≤ Sense$_{AdjustTotal}$	Invalid Range	Invalid Range

4. Determine the U_SNS_MULT setting:

$$U\ \ SNS\ \ MULT = ROUND\left[\left(\frac{Sense_{Typical\ \ Desired}}{Sense_{Typical\ \ NXP\ \ Trim}{}^{*}U\ \ SNS\ \ SHIFT} - 1\right){}^{*}256\right] = \left(\frac{0.62}{0.50} - 1\right){}^{*}256 = \ \ 61 decimal$$

U_SNS_MULT = 0x3D

8.6.3 Signal chain data type configuration

Each source enabled (as described in Section 8.5) must have its data type configured. Datatype configuration is described in the datasheet. A simplified table is included below:

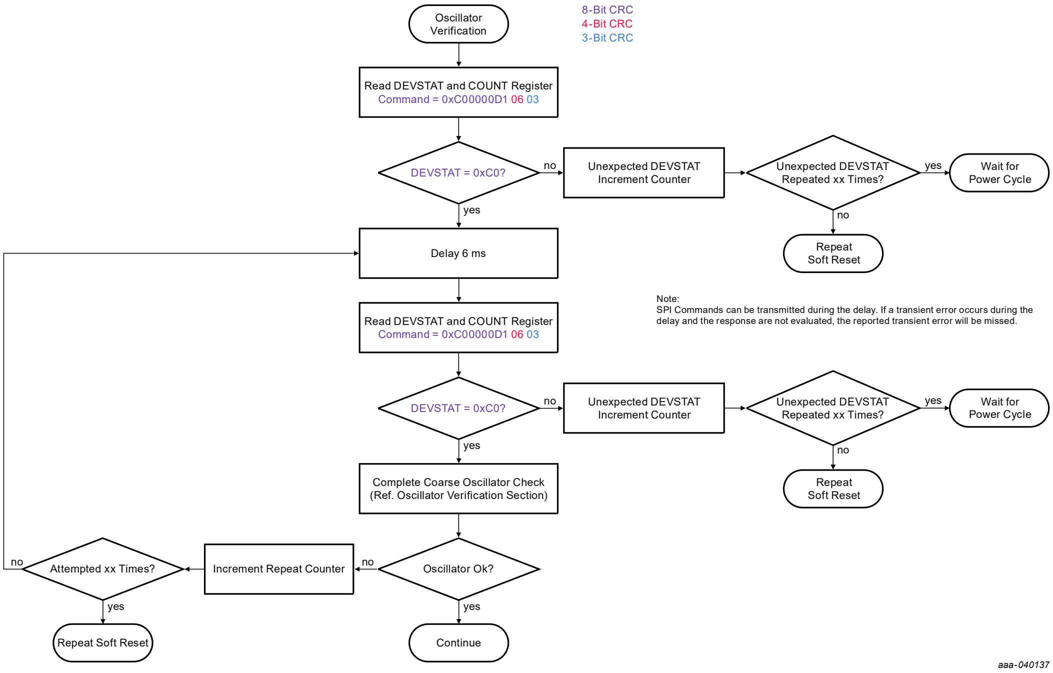
Table 10. Datatype configuration

CHx DATATYPEx[1:0]		Sensor data description
0	0	Offset Cancelled Data as Configured by the OC_FILT bits
0	1	Raw Data (No Offset Cancellation)
1	0	Temperature Sensor Data
1	1	Temperature Sensor Data

8.7 Complete startup coarse oscillator verification

The next step is to complete a coarse verification of the device oscillator. This is accomplished by reading the COUNT register and comparing the values over specific periods of time. At a minimum, the startup coarse oscillator verification is required to meet the diagnostic coverage as documented in the FMEDA.

The recommended procedure for coarse oscillator verification is shown in Figure 12. The procedure described includes a delay of 6 ms. A 6 ms delay can detect an oscillator error of greater ±10% with a master oscillator of less than ±1%. Reference section Section 9 "Optional Continuous Oscillator Verification" for details on the oscillator verification detection capability and the recommended limits for different measurement times and master oscillator accuracy cases.


Oscillator Verification
8-Bit CRC
4-Bit CRC
3-Bit CRC
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
DEVSTAT = 0xC0?
no
Unexpected DEVSTAT Increment Counter
Unexpected DEVSTAT Repeated xx Times?
yes
Wait for Power Cycle
no
Repeat Soft Reset
yes
Delay 6 ms
Read DEVSTAT and COUNT Register
Command = 0xC00000D1 06 03
DEVSTAT = 0xC0?
no
Unexpected DEVSTAT Increment Counter
Unexpected DEVSTAT Repeated xx Times?
yes
Wait for Power Cycle
no
Repeat Soft Reset
yes
Complete Coarse Oscillator Check (Ref. Oscillator Verification Section)
Oscillator Ok?
no
Increment Repeat Counter
Attempted xx Times?
no
yes
Repeat Soft Reset
yes
Continue
aaa-040137

Note:
SPI Commands can be transmitted during the delay. If a transient error occurs during the delay and the response are not evaluated, the reported transient error will be missed.

Figure 12. Confirm device oscillator: coarse verification

8.8 Confirm traceability information

The next step is to confirm the device level traceability information. The IC type, IC manufacturer ID, IC Part Number and IC Serial Number should be read to confirm that the proper device is connected. This can be accomplished on each power up as documented here, or one time only during the system manufacturing. The additional traceability information should also be read and stored in memory, or accessible via other means with a diagnostic tool to assist in failure analysis of quality incidents. The additional traceability information is not required to meet the functional safety requirements of safety systems.

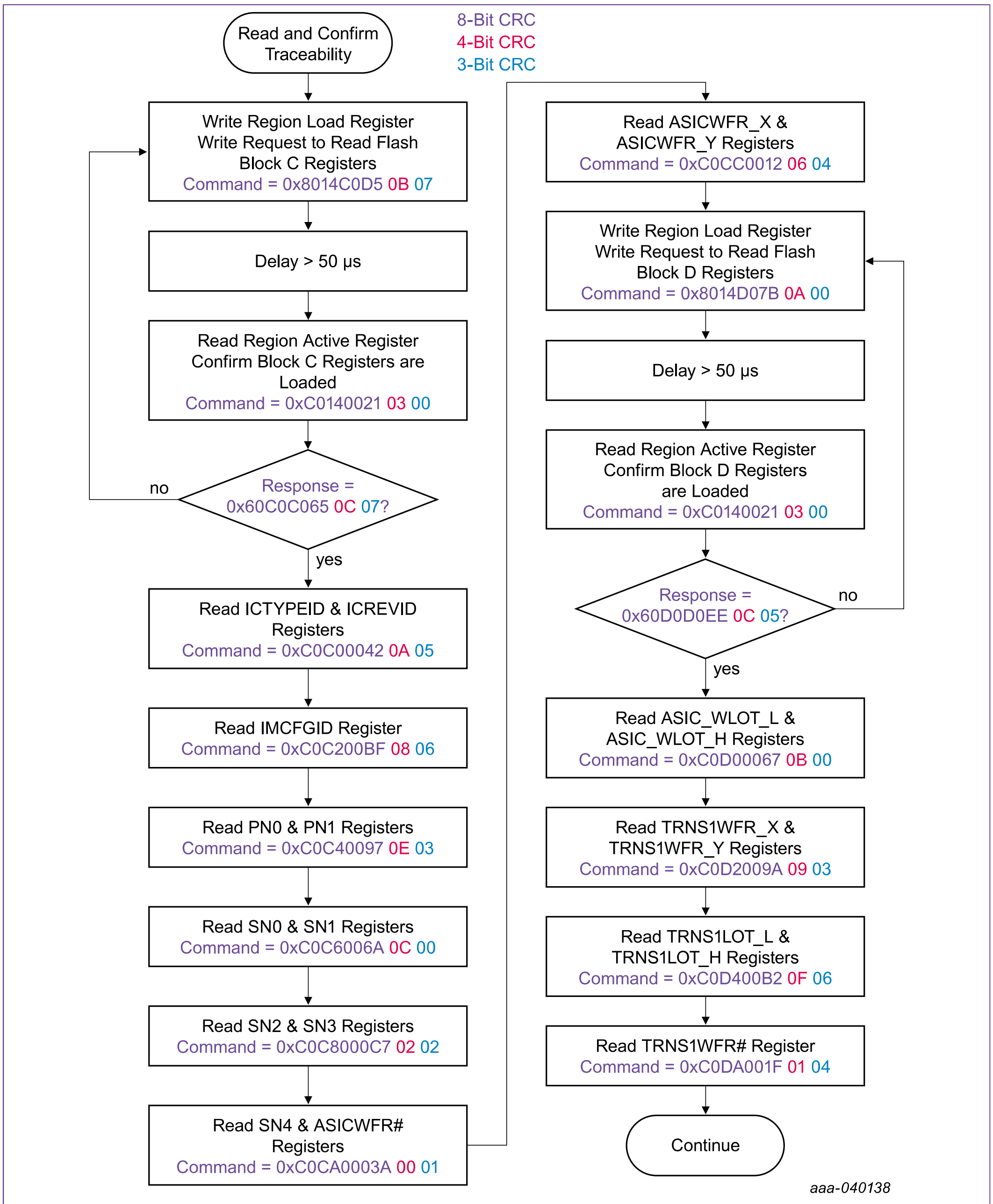

Read and Confirm Traceability
8-Bit CRC
4-Bit CRC
3-Bit CRC
Write Region Load Register Write Request to Read Flash Block C Registers
Command = 0x8014C0D5 0B 07
Delay > 50 µs
Read Region Active Register Confirm Block C Registers are Loaded
Command = 0xC0140021 03 00
Response = 0x60C0C065 0C 07?
no
yes
Read ICTYPEID & ICREVID Registers
Command = 0xC0C00042 0A 05
Read IMCFGID Register
Command = 0xC0C200BF 08 06
Read PN0 & PN1 Registers
Command = 0xC0C40097 0E 03
Read SN0 & SN1 Registers
Command = 0xC0C6006A 0C 00
Read SN2 & SN3 Registers
Command = 0xC0C8000C7 02 02
Read SN4 & ASICWFR# Registers
Command = 0xC0CA0003A 00 01
Read ASICWFR_X & ASICWFR_Y Registers
Command = 0xC0CC0012 06 04
Write Region Load Register Write Request to Read Flash Block D Registers
Command = 0x8014D07B 0A 00
Delay > 50 µs
Read Region Active Register Confirm Block D Registers are Loaded
Command = 0xC0140021 03 00
Response = 0x60D0D0EE 0C 05?
no
yes
Read ASIC_WLOT_L & ASIC_WLOT_H Registers
Command = 0xC0D00067 0B 00
Read TRNS1WFR_X & TRNS1WFR_Y Registers
Command = 0xC0D2009A 09 03
Read TRNS1LOT_L & TRNS1LOT_H Registers
Command = 0xC0D400B2 0F 06
Read TRNS1WFR# Register
Command = 0xC0DA001F 01 04
Continue
aaa-040138

Figure 13. Confirm device traceability information

8.9 Optional read and record stored self-test data

The next step is to read and record the self-test data stored in the device during device manufacturing. Reading this data is only necessary if the optional self-test accuracy verification test is used, as described in Section 8.11.4.1.3 "Optional self-test accuracy verification". The optional self-test accuracy verification test can be run at each power up as documented here, or one time only during system manufacturing. Reference Section 8.10 "Confirm the device configuration" and the associated sub-sections for details on how to use the stored self-test data for additional self-test accuracy.

Figure 14 shows the procedure for reading the stored self-test data.

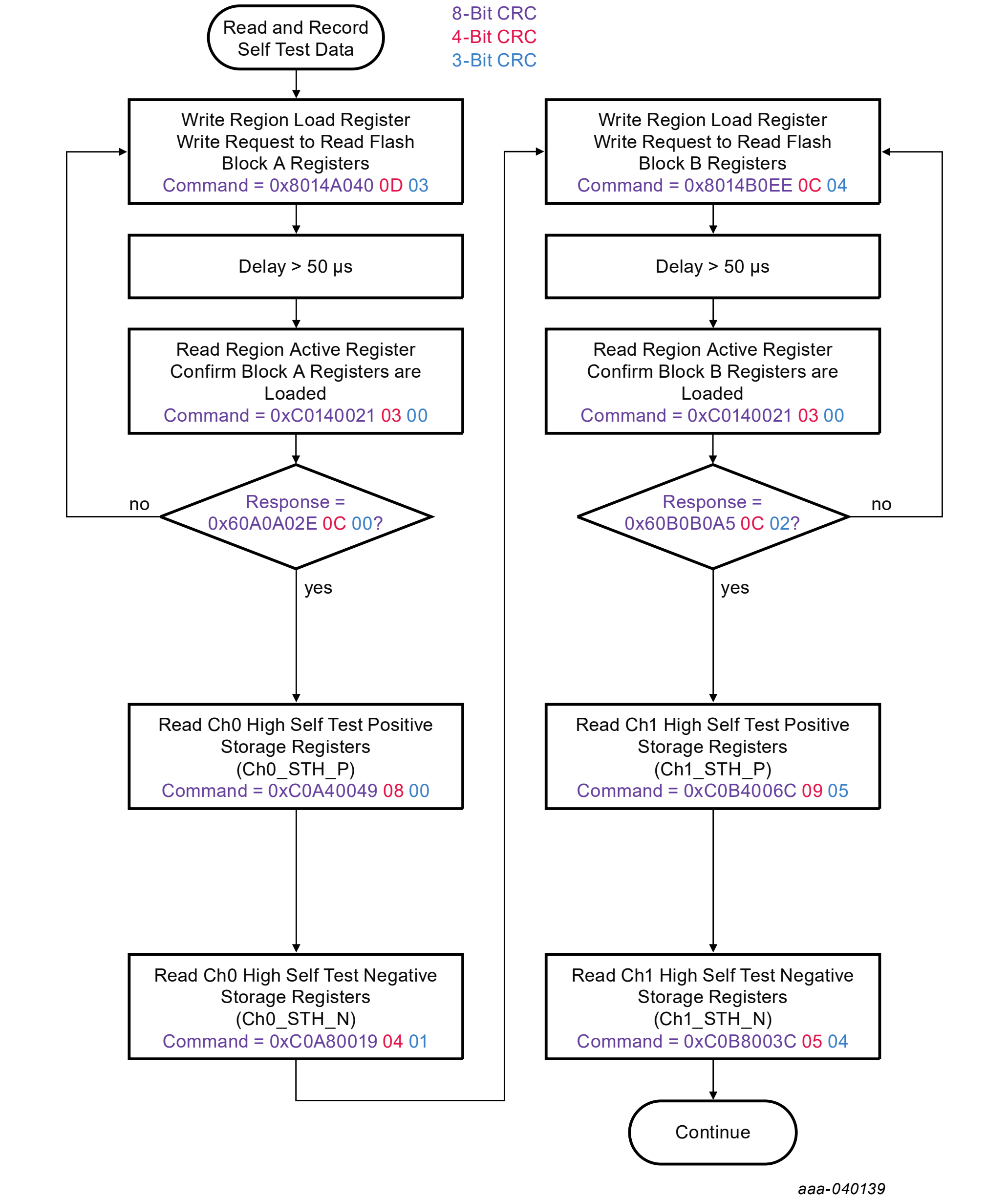
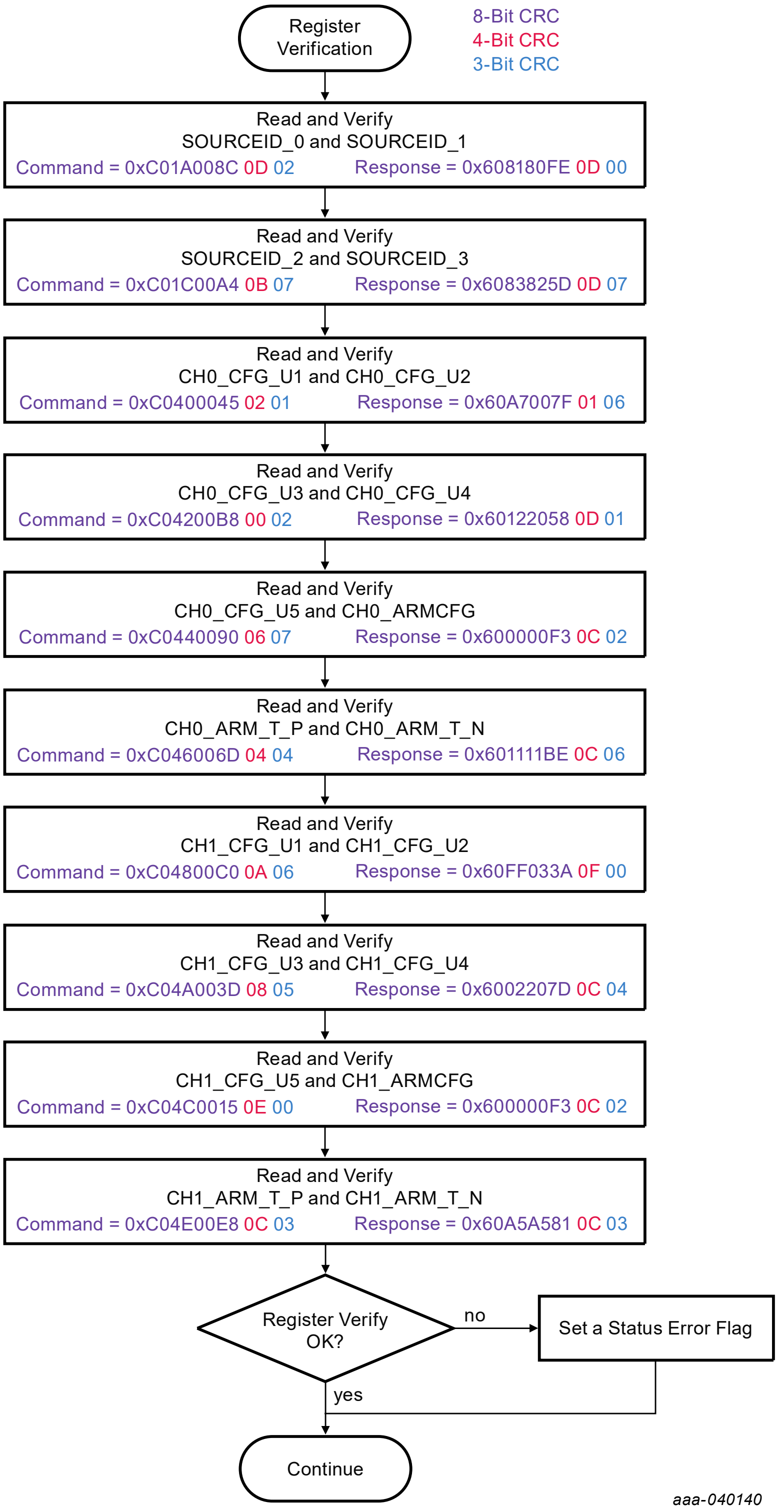

Read and Record Self Test Data
8-Bit CRC
4-Bit CRC
3-Bit CRC
Write Region Load Register Write Request to Read Flash Block A Registers Command = 0x8014A040 0D 03
Delay > 50 µs
Read Region Active Register Confirm Block A Registers are Loaded Command = 0xC0140021 03 00
Response = 0x60A0A02E 0C 00?
no
yes
Read Ch0 High Self Test Positive Storage Registers (Ch0_STH_P) Command = 0xC0A40049 08 00
Read Ch0 High Self Test Negative Storage Registers (Ch0_STH_N) Command = 0xC0A80019 04 01
Write Region Load Register Write Request to Read Flash Block B Registers Command = 0x8014B0EE 0C 04
Delay > 50 µs
Read Region Active Register Confirm Block B Registers are Loaded Command = 0xC0140021 03 00
Response = 0x60B0B0A5 0C 02?
no
yes
Read Ch1 High Self Test Positive Storage Registers (Ch1_STH_P) Command = 0xC0B4006C 09 05
Read Ch1 High Self Test Negative Storage Registers (Ch1_STH_N) Command = 0xC0B8003C 05 04
Continue
aaa-040139

Figure 14. Read and record the stored self-test data

8.10 Confirm the device configuration

The next step is to confirm the device configuration. All registers that impact the communication or signal chain configuration should be read whether they were written by the user or not. This provides an additional verification of the reset state of all registers beyond the internal OTP CRC. Figure 15 shows an example configuration.


Register Verification
8-Bit CRC
4-Bit CRC
3-Bit CRC
Read and Verify SOURCEID_0 and SOURCEID_1
Command = 0xC01A008C 0D 02
Response = 0x608180FE 0D 00
Read and Verify SOURCEID_2 and SOURCEID_3
Command = 0xC01C00A4 0B 07
Response = 0x6083825D 0D 07
Read and Verify CH0_CFG_U1 and CH0_CFG_U2
Command = 0xC0400045 02 01
Response = 0x60A7007F 01 06
Read and Verify CH0_CFG_U3 and CH0_CFG_U4
Command = 0xC04200B8 00 02
Response = 0x60122058 0D 01
Read and Verify CH0_CFG_U5 and CH0_ARMCFG
Command = 0xC0440090 06 07
Response = 0x600000F3 0C 02
Read and Verify CH0_ARM_T_P and CH0_ARM_T_N
Command = 0xC046006D 04 04
Response = 0x601111BE 0C 06
Read and Verify CH1_CFG_U1 and CH1_CFG_U2
Command = 0xC04800C0 0A 06
Response = 0x60FF033A 0F 00
Read and Verify CH1_CFG_U3 and CH1_CFG_U4
Command = 0xC04A003D 08 05
Response = 0x6002207D 0C 04
Read and Verify CH1_CFG_U5 and CH1_ARMCFG
Command = 0xC04C0015 0E 00
Response = 0x600000F3 0C 02
Read and Verify CH1_ARM_T_P and CH1_ARM_T_N
Command = 0xC04E00E8 0C 03
Response = 0x60A5A581 0C 03
Register Verify OK?
no
Set a Status Error Flag
yes
Continue

aaa-040140

Figure 15. Confirm the device configuration

8.11 Complete self-test (self-test overview)

The next step is to complete some or all of the various self-test functions available in the device. Figure 17 shows an overview of a recommended procedure for completing self-test. Test repeats on failure are not shown in the diagrams. The user must determine the number of test repeats for each test type based on the application. Typically test repeats are kept at a minimum for the analog self-test procedures in order to avoid the type of invalid inputs that are common during startup.

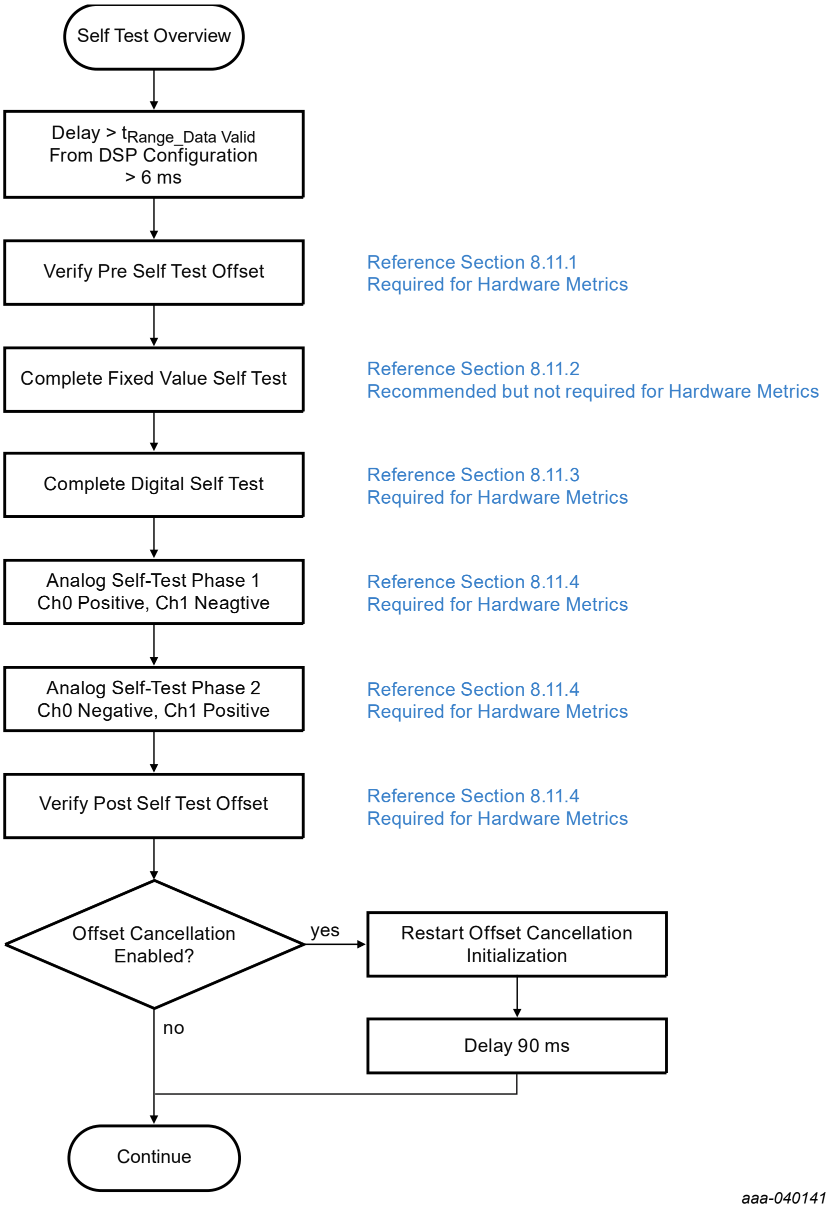

Self Test Overview
Delay > t_Range_Data Valid From DSP Configuration > 6 ms
Verify Pre Self Test Offset
Reference Section 8.11.1 Required for Hardware Metrics
Complete Fixed Value Self Test
Reference Section 8.11.2 Recommended but not required for Hardware Metrics
Complete Digital Self Test
Reference Section 8.11.3 Required for Hardware Metrics
Analog Self-Test Phase 1 Ch0 Positive, Ch1 Neagtive
Reference Section 8.11.4 Required for Hardware Metrics
Analog Self-Test Phase 2 Ch0 Negative, Ch1 Positive
Reference Section 8.11.4 Required for Hardware Metrics
Verify Post Self Test Offset
Reference Section 8.11.4 Required for Hardware Metrics
Offset Cancellation Enabled?
yes
Restart Offset Cancellation Initialization
Delay 90 ms
no
Continue
aaa-040141

Figure 16. Self-test overview

8.11.1 Complete pre-self-test offset

The next step is to complete an offset verification. The purpose of the offset verification is to:

1. Verify the offset of the device and any change in offset before and after the self-test motion.
2. Capture the pre-self-test offset that will be subtracted from the measured self-test values during analog self-test.

The flow charts below show an example procedure for capturing the sensor offset.

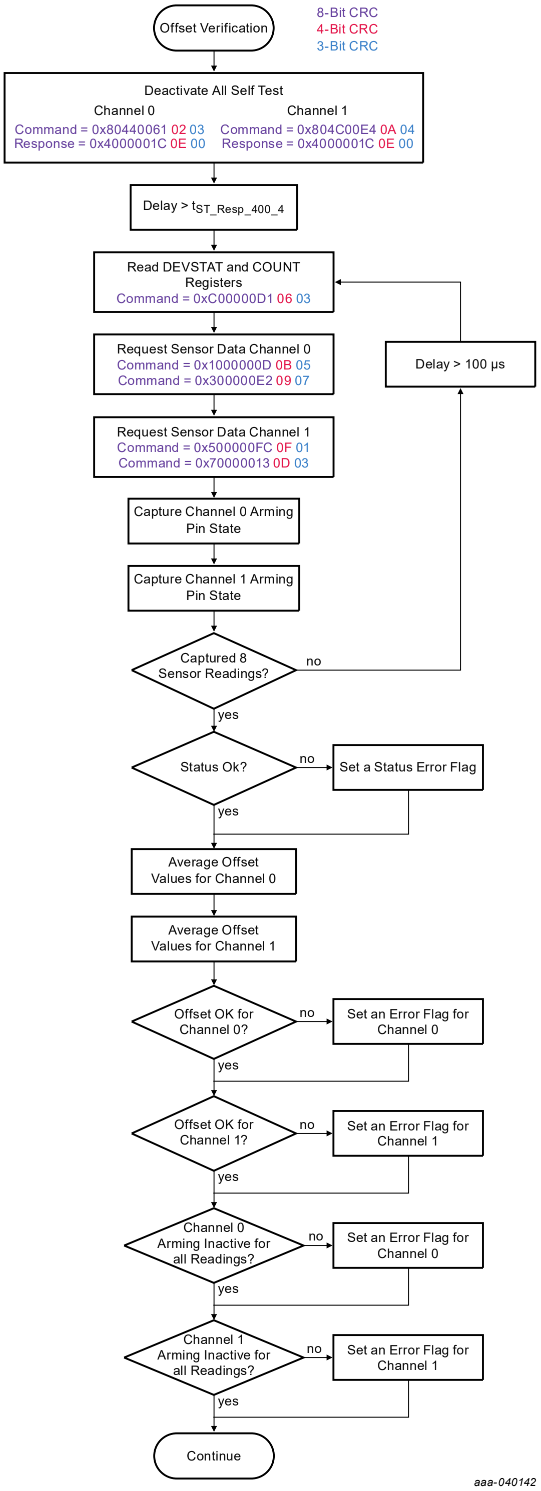

Offset Verification
8-Bit CRC
4-Bit CRC
3-Bit CRC
Deactivate All Self Test
Channel 0
Command = 0x80440061 02 03
Response = 0x4000001C 0E 00
Channel 1
Command = 0x804C00E4 0A 04
Response = 0x4000001C 0E 00
Delay > $t_{ST\_Resp\_400\_4}$
Read DEVSTAT and COUNT Registers
Command = 0xC00000D1 06 03
Request Sensor Data Channel 0
Command = 0x1000000D 0B 05
Command = 0x300000E2 09 07
Request Sensor Data Channel 1
Command = 0x500000FC 0F 01
Command = 0x70000013 0D 03
Capture Channel 0 Arming Pin State
Capture Channel 1 Arming Pin State
Delay > 100 µs
Captured 8 Sensor Readings?
no
yes
Status Ok?
no
Set a Status Error Flag
yes
Average Offset Values for Channel 0
Average Offset Values for Channel 1
Offset OK for Channel 0?
no
Set an Error Flag for Channel 0
yes
Offset OK for Channel 1?
no
Set an Error Flag for Channel 1
yes
Channel 0 Arming Inactive for all Readings?
no
Set an Error Flag for Channel 0
yes
Channel 1 Arming Inactive for all Readings?
no
Set an Error Flag for Channel 1
yes
Continue
aaa-040142

Figure 17. Pre-self-test offset

8.11.2 Complete fixed value self-test

The next step is to complete a fixed value self-test verification for each device. The purpose of the fixed value self-test is to confirm that the output data register and communication block have no stuck bit conditions. Figure 18 shows an example procedure for completing self-test with two fixed values. The example alternates 0x5555 and 0xAAAA by channel to confirm both states of each bit in the data field and to maximize verification of channel independent data. Expected responses are included for each self-test request.

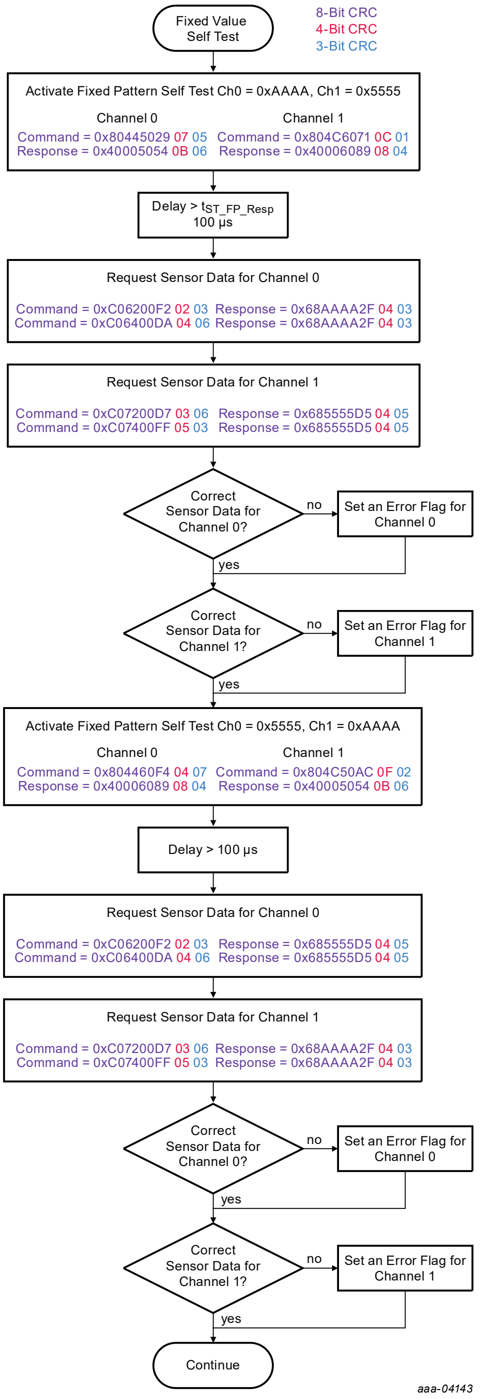

Fixed Value Self Test
8-Bit CRC
4-Bit CRC
3-Bit CRC
Activate Fixed Pattern Self Test Ch0 = 0xAAAA, Ch1 = 0x5555
Channel 0
Channel 1
Command = 0x80445029 07 05
Response = 0x40005054 0B 06
Command = 0x804C6071 0C 01
Response = 0x40006089 08 04
Delay > tST_FP_Resp 100 µs
Request Sensor Data for Channel 0
Command = 0xC06200F2 02 03 Response = 0x68AAAA2F 04 03
Command = 0xC06400DA 04 06 Response = 0x68AAAA2F 04 03
Request Sensor Data for Channel 1
Command = 0xC07200D7 03 06 Response = 0x685555D5 04 05
Command = 0xC07400FF 05 03 Response = 0x685555D5 04 05
Correct Sensor Data for Channel 0?
no
Set an Error Flag for Channel 0
yes
Correct Sensor Data for Channel 1?
no
Set an Error Flag for Channel 1
yes
Activate Fixed Pattern Self Test Ch0 = 0x5555, Ch1 = 0xAAAA
Channel 0
Channel 1
Command = 0x804460F4 04 07
Response = 0x40006089 08 04
Command = 0x804C50AC 0F 02
Response = 0x40005054 0B 06
Delay > 100 µs
Request Sensor Data for Channel 0
Command = 0xC06200F2 02 03 Response = 0x685555D5 04 05
Command = 0xC06400DA 04 06 Response = 0x685555D5 04 05
Request Sensor Data for Channel 1
Command = 0xC07200D7 03 06 Response = 0x68AAAA2F 04 03
Command = 0xC07400FF 05 03 Response = 0x68AAAA2F 04 03
Correct Sensor Data for Channel 0?
no
Set an Error Flag for Channel 0
yes
Correct Sensor Data for Channel 1?
no
Set an Error Flag for Channel 1
yes
Continue
aaa-04143

Figure 18. Complete fixed value self-test

8.11.3 Complete digital self-test

The next step is to complete a digital self-test verification for each device. The purpose of the digital self-test is to achieve a more accurate verification of the digital signal chain. The digital self-test forces a known value into the input of the digital signal chain. After a defined interval of time, dependent on the low pass filter selected, the signal chain output can be verified against an expected value plus or minus a small tolerance. Figure 19 shows an example procedure for completing a self-test of one digital value (Digital Self-Test 0xC) and confirming the expected output value.

If offset cancellation is being used, bypass the offset cancellation filter for digital self-test to eliminate the effects of the filter on the digital self-test result. The procedure below includes offset cancellation bypass during digital self-test.

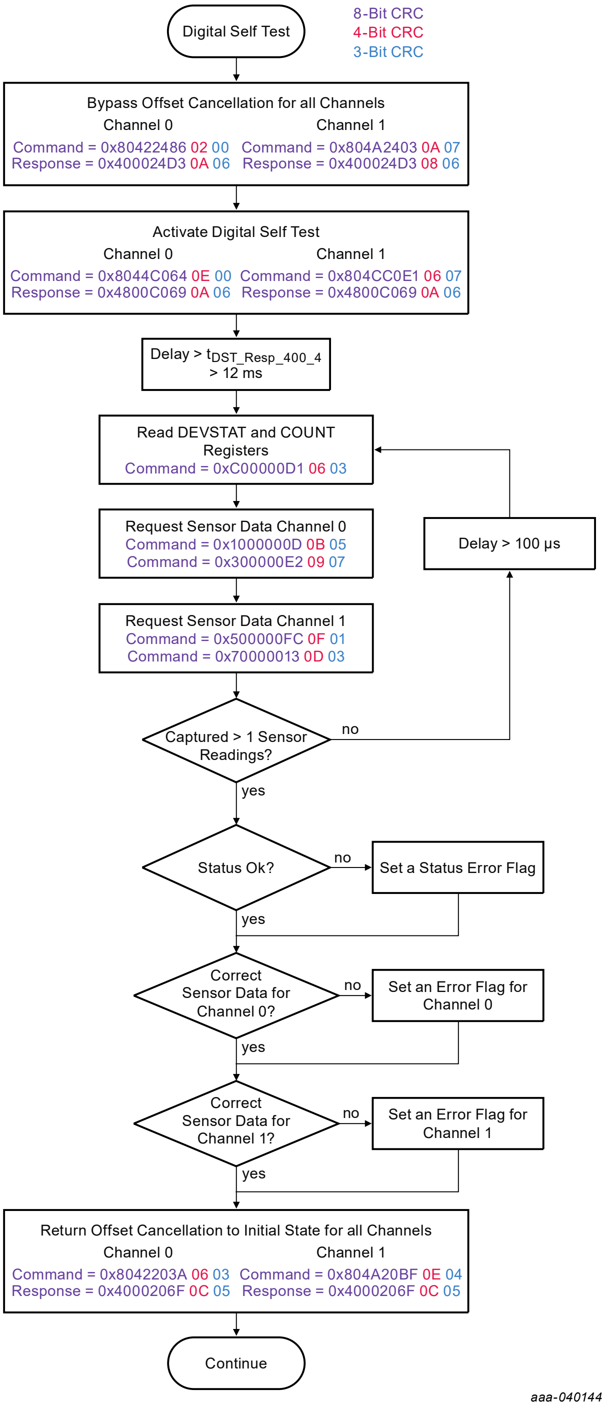

Digital Self Test
8-Bit CRC
4-Bit CRC
3-Bit CRC
Bypass Offset Cancellation for all Channels
Channel 0
Channel 1
Command = 0x80422486 02 00
Response = 0x400024D3 0A 06
Command = 0x804A2403 0A 07
Response = 0x400024D3 08 06
Activate Digital Self Test
Channel 0
Channel 1
Command = 0x8044C064 0E 00
Response = 0x4800C069 0A 06
Command = 0x804CC0E1 06 07
Response = 0x4800C069 0A 06
Delay > tDST_Resp_400_4 > 12 ms
Read DEVSTAT and COUNT Registers
Command = 0xC00000D1 06 03
Request Sensor Data Channel 0
Command = 0x1000000D 0B 05
Command = 0x300000E2 09 07
Request Sensor Data Channel 1
Command = 0x500000FC 0F 01
Command = 0x70000013 0D 03
Delay > 100 µs
Captured > 1 Sensor Readings?
no
yes
Status Ok?
no
Set a Status Error Flag
yes
Correct Sensor Data for Channel 0?
no
Set an Error Flag for Channel 0
yes
Correct Sensor Data for Channel 1?
no
Set an Error Flag for Channel 1
yes
Return Offset Cancellation to Initial State for all Channels
Channel 0
Channel 1
Command = 0x8042203A 06 03
Response = 0x4000206F 0C 05
Command = 0x804A20BF 0E 04
Response = 0x4000206F 0C 05
Continue
aaa-040144

Figure 19. Complete digital self-test

8.11.3.1 Digital self-test limit calculation

The digital self-test provides a constant value to the sensor data output regardless of the user gain settings. The limits for the minimum gain setting are listed in the datasheet and included in Table 11 below.

Table 11. Data sheet digital self-test values with minimum gain

Self-test ST_CTRL[3:0]	Expected value CHx_SNSDATAx register read (signed HEX)			Expected value CHx_SNSDATAx register read (unsigned HEX)			Expected value CHx_SNSDATAx register read (decimal)		
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
0xC	E77F	E780	E781	677F	6780	6781	-6273	-6272	-6271
0xD	0FA3	0FA4	0FA5	8FA3	8FA4	8FA5	4003	4004	4005
0xE	EFA2	EFA3	EFA4	6FA2	6FA3	6FA4	-4190	-4189	-4188
0xF	07B7	07B8	07B9	87B7	87B8	87B9	1975	1976	1977

8.11.4 Complete analog self-test

The next step is to complete an offset and analog self-test verification. The purpose of the analog self-test is to:

1. Confirm unimpeded motion of the proof mass in both the positive and negative acceleration directions.
2. Verify the sensitivity accuracy of the device. The FXLS9xxxx devices contain multiple self-test capabilities and procedures that have different sensitivity accuracy verification capabilities.
3. Verify the offset of the device and any change in offset before and after the self-test motion.

The flow charts below show an example analog self-test procedure for measuring both positive and negative self-test. Section 8.11.4.1 "Analog self-test pass fail limits" explains the pass/fail criteria for analog self-test.

When sensor data is read for any of the analog self-test functions, the sensor data can be accessed either by using the Sensor Data Request commands or by reading the SNSDATA registers directly. For some user gain settings, the analog self-test results in a potential railed sensor data output via the Sensor Data Request commands. For these test cases, the data must be read via the SNSDATA registers. Note that if the SNSDATA registers are read directly, the arming function is not updated and arming pin verification is not feasible. If the arming function is to be verified, either a separate self-test arming verification test can be run or both the Sensor Data Request command and the SNSDATA registers reads can be completed simultaneously.

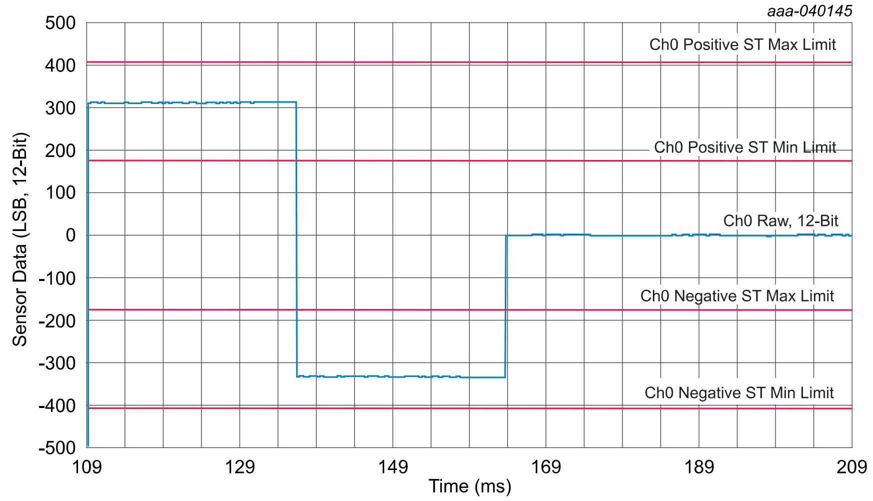


Figure 20. Example analog self-test response, Channel 0

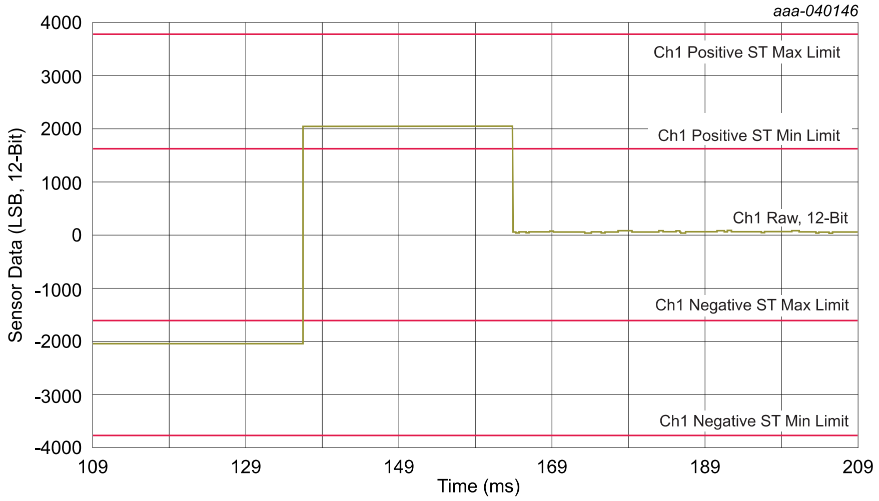


Figure 21. Example analog self-test response, Channel 1

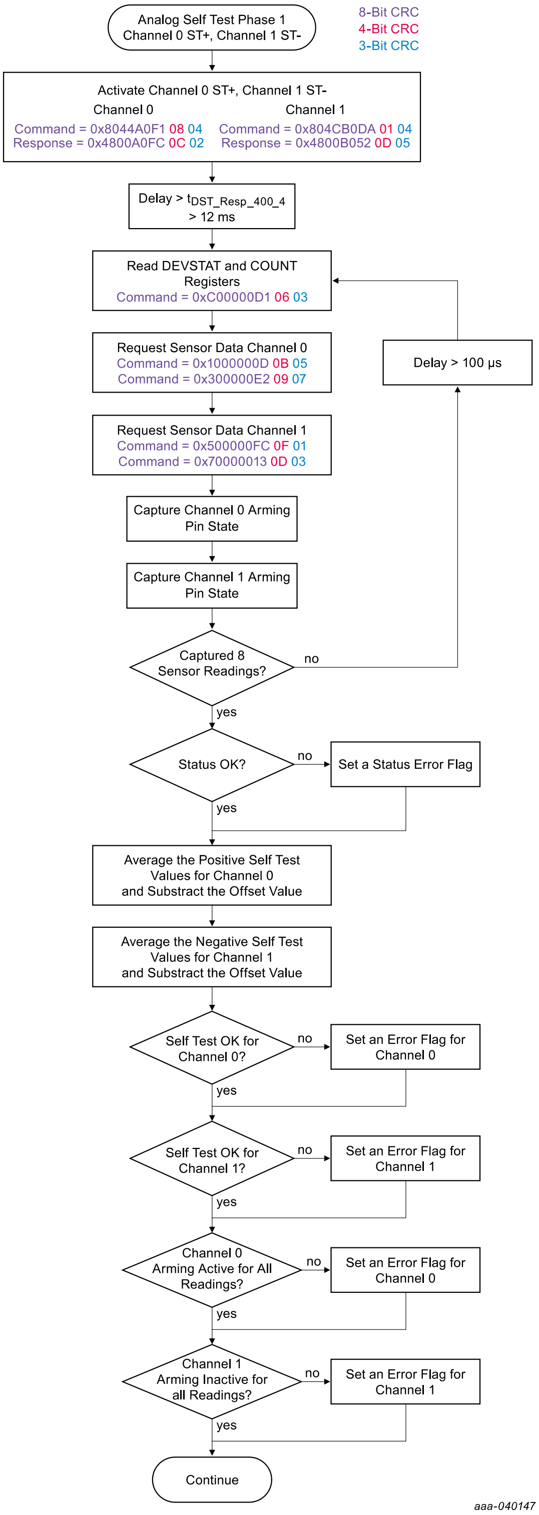

Analog Self Test Phase 1
Channel 0 ST+, Channel 1 ST-
8-Bit CRC
4-Bit CRC
3-Bit CRC
Activate Channel 0 ST+, Channel 1 ST-
Channel 0
Channel 1
Command = 0x8044A0F1 08 04
Response = 0x4800A0FC 0C 02
Command = 0x804CB0DA 01 04
Response = 0x4800B052 0D 05
Delay > $t_{DST\_Resp\_400\_4}$ > 12 ms
Read DEVSTAT and COUNT Registers
Command = 0xC00000D1 06 03
Request Sensor Data Channel 0
Command = 0x1000000D 0B 05
Command = 0x300000E2 09 07
Request Sensor Data Channel 1
Command = 0x500000FC 0F 01
Command = 0x70000013 0D 03
Delay > 100 µs
Capture Channel 0 Arming Pin State
Capture Channel 1 Arming Pin State
Captured 8 Sensor Readings?
no
yes
Status OK?
no
Set a Status Error Flag
yes
Average the Positive Self Test Values for Channel 0 and Substract the Offset Value
Average the Negative Self Test Values for Channel 1 and Substract the Offset Value
Self Test OK for Channel 0?
no
Set an Error Flag for Channel 0
yes
Self Test OK for Channel 1?
no
Set an Error Flag for Channel 1
yes
Channel 0 Arming Active for All Readings?
no
Set an Error Flag for Channel 0
yes
Channel 1 Arming Inactive for all Readings?
no
Set an Error Flag for Channel 1
yes
Continue
aaa-040147

Figure 22. Channel 0 positive self-test, Channel 1 negative self-test

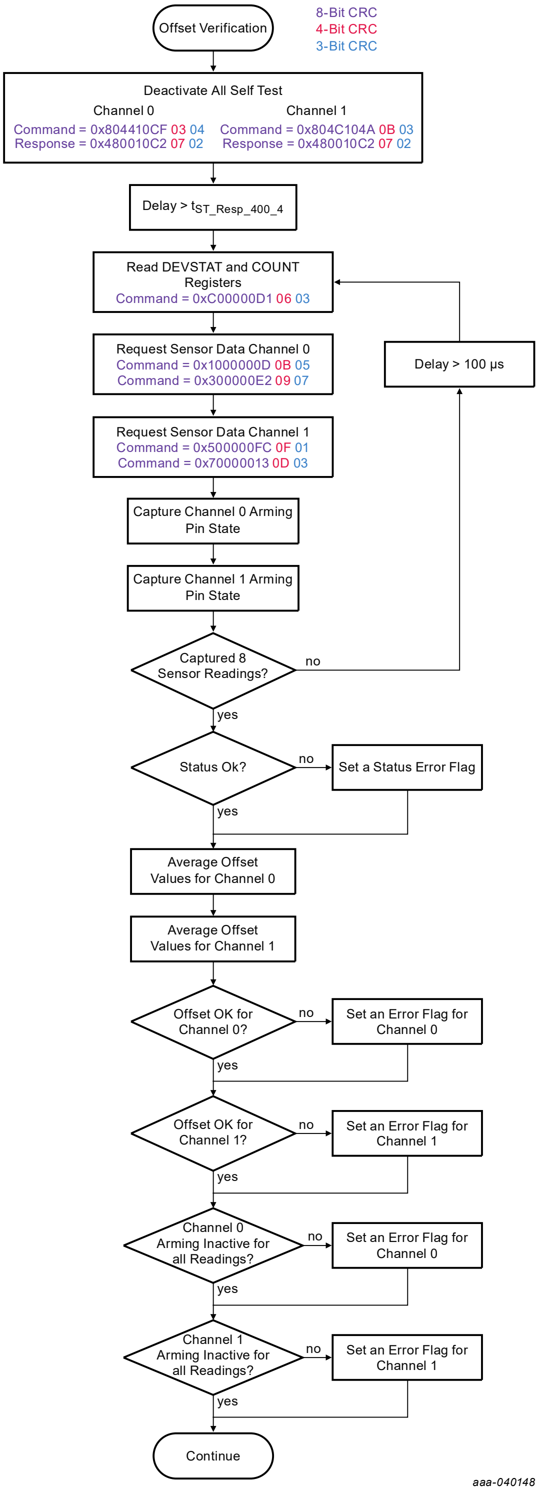

Offset Verification
8-Bit CRC
4-Bit CRC
3-Bit CRC
Deactivate All Self Test
Channel 0
Channel 1
Command = 0x804410CF 03 04
Response = 0x480010C2 07 02
Command = 0x804C104A 0B 03
Response = 0x480010C2 07 02
Delay > $t_{ST\_Resp\_400\_4}$
Read DEVSTAT and COUNT Registers
Command = 0xC00000D1 06 03
Delay > 100 μs
Request Sensor Data Channel 0
Command = 0x1000000D 0B 05
Command = 0x300000E2 09 07
Request Sensor Data Channel 1
Command = 0x500000FC 0F 01
Command = 0x70000013 0D 03
Capture Channel 0 Arming Pin State
Capture Channel 1 Arming Pin State
Captured 8 Sensor Readings?
no
yes
Status Ok?
no
Set a Status Error Flag
yes
Average Offset Values for Channel 0
Average Offset Values for Channel 1
Offset OK for Channel 0?
no
Set an Error Flag for Channel 0
yes
Offset OK for Channel 1?
no
Set an Error Flag for Channel 1
yes
Channel 0 Arming Inactive for all Readings?
no
Set an Error Flag for Channel 0
yes
Channel 1 Arming Inactive for all Readings?
no
Set an Error Flag for Channel 1
yes
Continue
aaa-040148

Figure 23. Mid self-test offset

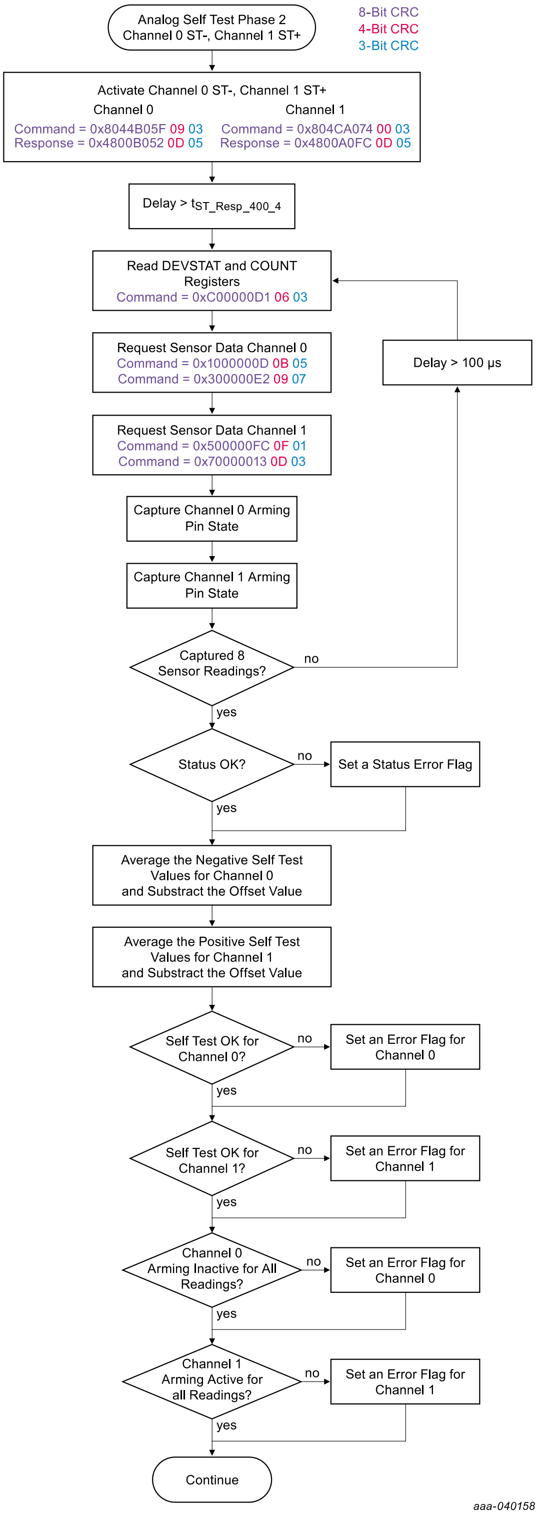

Analog Self Test Phase 2
Channel 0 ST-, Channel 1 ST+
8-Bit CRC
4-Bit CRC
3-Bit CRC
Activate Channel 0 ST-, Channel 1 ST+
Channel 0
Command = 0x8044B05F 09 03
Response = 0x4800B052 0D 05
Channel 1
Command = 0x804CA074 00 03
Response = 0x4800A0FC 0D 05
Delay > tST_Resp_400_4
Read DEVSTAT and COUNT Registers
Command = 0xC00000D1 06 03
Request Sensor Data Channel 0
Command = 0x1000000D 0B 05
Command = 0x300000E2 09 07
Request Sensor Data Channel 1
Command = 0x500000FC 0F 01
Command = 0x70000013 0D 03
Delay > 100 µs
Capture Channel 0 Arming Pin State
Capture Channel 1 Arming Pin State
Captured 8 Sensor Readings?
no
yes
Status OK?
no
Set a Status Error Flag
yes
Average the Negative Self Test Values for Channel 0 and Substract the Offset Value
Average the Positive Self Test Values for Channel 1 and Substract the Offset Value
Self Test OK for Channel 0?
no
Set an Error Flag for Channel 0
yes
Self Test OK for Channel 1?
no
Set an Error Flag for Channel 1
yes
Channel 0 Arming Inactive for All Readings?
no
Set an Error Flag for Channel 0
yes
Channel 1 Arming Active for all Readings?
no
Set an Error Flag for Channel 1
yes
Continue
aaa-040158

Figure 24. Channel 0 negative self-test, Channel 1 positive self-test

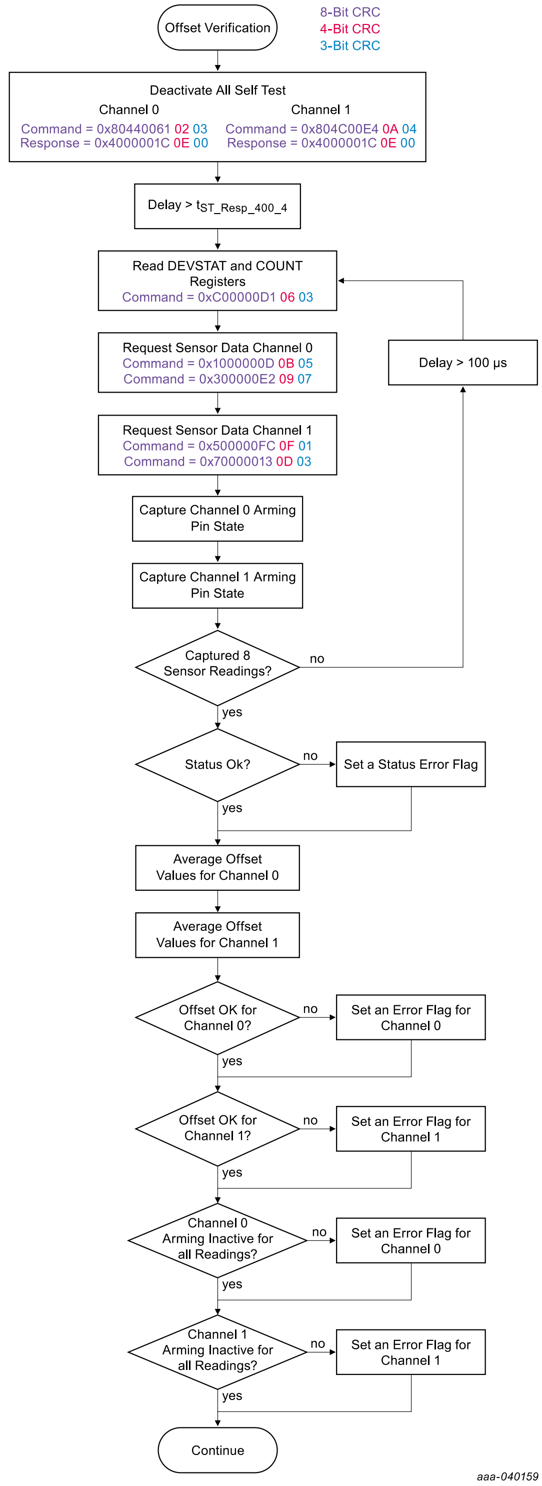

Offset Verification
8-Bit CRC
4-Bit CRC
3-Bit CRC
Deactivate All Self Test
Channel 0
Channel 1
Command = 0x80440061 02 03
Response = 0x4000001C 0E 00
Command = 0x804C00E4 0A 04
Response = 0x4000001C 0E 00
Delay > $t_{ST\_Resp\_400\_4}$
Read DEVSTAT and COUNT Registers
Command = 0xC00000D1 06 03
Request Sensor Data Channel 0
Command = 0x1000000D 0B 05
Command = 0x300000E2 09 07
Delay > 100 µs
Request Sensor Data Channel 1
Command = 0x500000FC 0F 01
Command = 0x70000013 0D 03
Capture Channel 0 Arming Pin State
Capture Channel 1 Arming Pin State
Captured 8 Sensor Readings?
no
yes
Status Ok?
no
Set a Status Error Flag
yes
Average Offset Values for Channel 0
Average Offset Values for Channel 1
Offset OK for Channel 0?
no
Set an Error Flag for Channel 0
yes
Offset OK for Channel 1?
no
Set an Error Flag for Channel 1
yes
Channel 0 Arming Inactive for all Readings?
no
Set an Error Flag for Channel 0
yes
Channel 1 Arming Inactive for all Readings?
no
Set an Error Flag for Channel 1
yes
Continue
aaa-040159

Figure 25. Post self-test offset

8.11.4.1 Analog self-test pass fail limits

The analog self-test results can be evaluated via several methods. This section covers the various methods for evaluating the offset and self-test results to improve diagnostic coverage. The tests described in Section 8.11.4.1.1 "Standard offset verification" and

Section 8.11.4.1.2 "Standard self-test verification" are required to meet the diagnostic coverage as documented in the FMEDA. The additional tests improve diagnostic coverage.

8.11.4.1.1 Standard offset verification

The average offset calculated from each data source captured from the test in Figure 17 is compared against the offset limits in the datasheet. If offset cancellation (high pass filter) is not used and a lower range is programmed than the range condition specified for offset (25 g for medium g, 100 g for high g), the offset limits must be scaled with the user gain. Table 12 shows some example offset test limits for different ranges and settings.

Offset verification in this procedure is included as part of the initialization procedure. If it is more convenient for the user, offset verification can be completed after entering Normal Mode.

Table 12. Standard offset verification test limits, medium g

Offset cancellation enabled?	User programmed range medium g (g)	Minimum limit (LSB, 12-bit)	Maximum limit (LSB, 12-bit)	Pass / fail criteria
Yes	All	-1	+1	Min Limit ≤ Result ≤ Max Limit
No	15.5	-162	+162	
No	16.0	-157	+157	
No	20.0	-126	+126	
No	25.0	-100	+100	
No	35.0	-72	+72	
No	50.0	-50	+50	
No	60.0	-42	+42	
No	62.0	-41	+41	
No	62.5	-41	+41	
No	75.0	-34	+34	
No	85.3	-30	+30	
No	100.0	-25	+25	
No	105.0	-24	+24	
No	112.5	-23	+23	
No	125.0	-21	+21	
No	128.0	-20	+20	
No	150.0	-17	+17	

Table 13. Standard offset verification test limits, high g

Offset cancellation enabled?	User programmed range high g (g)	Minimum limit (LSB, 12-bit)	Maximum limit (LSB, 12-bit)	Pass / fail criteria
Yes	All	-1	+1	Min Limit ≤ Result ≤ Max Limit
No	50.0	-200	+200	
No	60.0	-167	+167	
No	62.0	-162	+162	
No	62.5	-160	+160	
No	100.0	-100	+100	
No	105.0	-96	+96	
No	112.5	-89	+89	
No	125.0	-80	+80	
No	128.0	-79	+79	
No	150.0	-67	+67	
No	187.0	-54	+54	
No	250.0	-40	+40	
No	312.5	-32	+32	
No	375.0	-27	+27	
No	500.0	-20	+20	
No	748.0	-14	+14	

The table below summarizes the post-self-test offset values measured in the example run for this application note.

Table 14. Offset measured values

Channel	User programmed range medium g (g)	Offset cancellation?	Minimum limit (LSB, 12-bit)	20ms average post-self-test offset (LSB, 12-bit)	Maximum limit (LSB, 12-bit)	Pass / fail
0	150g	No	-100	0.25	+100	Pass
0	150g	Yes	-1	0.51	+1	Pass
1	15.5g	No	-167	7.20	+167	Pass
1	15.5g	Yes	-1	0.57	+1	Pass

8.11.4.1.2 Standard self-test verification

The resulting positive and negative self-test values from the tests in Figure 22 through Figure 25 are compared against the self-test limits in the datasheet. The self-test limits must be scaled with the user gain. The equation used to convert the specified self-test in g to minimum and maximum self-test limits is shown below. Table 15 through Table 18 show some example self-test limits for different ranges and settings.

$$AnalogSelfTest_{Min@UserGain} = Truncate\left[ST_{MH\_62X\_13min} * (UserGain)\right]$$

$$AnalogSelfTest_{Max@UserGain} = RoundUp\left[ST_{MH\_62X\_13max} * (UserGain)\right]$$

Where:

- $AnalogSelfTest_{Min@UserGain}$ = The SNSDATAx register analog self-test minimum limit for the configured user gain
- $AnalogSelfTest_{Max@UserGain}$ = The SNSDATAx register analog self-test maximum limit for the configured user gain
- $ST_{MH\_62X\_13min}$ = The min med g SNSDATAx register self-test delta from offset for a user gain of 1
- $ST_{MH\_62X\_13max}$ = The max med g SNSDATAx register self-test delta from offset for a user gain of 1
- *UserGain* = The configured gain, calculated as shown in Section 8.6.2 "Signal chain user gain selection"

Table 15. Standard self-test verification test limits, medium g X-axis

Device type	NXP trim value (LSB/g)	User programmed range (g)	User programmed gain	Self-test limits, SNSDATA register value [1]) (LSB)		Pass / fail criteria
				Minimum	Maximum	
Medium g X	33.0161	15.5	3.9922	7038	16428	Min Limit ≤ Result ≤ Max Limit
		16.0	3.8750	6831	15946	
		20.0	3.1016	5468	12763	
		25.0	2.4766	4366	10192	
		35.0	1.7695	3119	7282	
		50.0	1.2383	2183	5096	
		60.0	1.0352	1824	4260	
		62.0	1.0000	1763	4115	
		62.5	0.9922	1749	4083	
		64.0	0.9688	1707	3987	
		75.0	0.8262	1456	3400	
		85.3	0.7266	1280	2990	
		100.0	0.6191	1091	2548	
		105.0	0.5898	1039	2428	
		112.5	0.5508	971	2267	
		125.0	0.4961	874	2042	
		128.0	0.4844	853	1994	
		150.0	0.4131	728	1700	
		248.0	0.2500	440	1029	

[1] If the self-test limits exceed 2^{11}-1 (2047) LSB, self-test value must be read using reads of the SNSDATA register to avoid railed value.

Table 16. Standard self-test verification test limits, medium g Z-axis

Device type	NXP trim value (LSB/g)	User programmed range (g)	User programmed gain	Self-test limits, SNSDATA register value[1]) (LSB)		Pass / fail criteria
				Minimum	Maximum	
Medium g Z	33.0161	15.5	3.9922	3421	7993	Min Limit ≤ Result ≤ Max Limit
		16.0	3.8750	3320	7758	
		20.0	3.1016	2658	6210	
		25.0	2.4766	2122	4959	
		35.0	1.7695	1516	3543	
		50.0	1.2383	1061	2480	
		60.0	1.0352	887	2073	
		62.0	1.0000	857	2002	
		62.5	0.9922	850	1987	
		64.0	0.9688	830	1940	
		75.0	0.8262	708	1654	
		85.3	0.7266	622	1455	
		100.0	0.6191	530	1240	
		105.0	0.5898	505	1181	
		112.5	0.5508	472	1103	
		125.0	0.4961	425	994	
		128.0	0.4844	415	970	
		150.0	0.4131	354	827	
		248.0	0.2500	214	501	

[1] If the self-test limits exceed 2^{11}-1 (2047) LSB, self-test value must be read using reads of the SNSDATA register to avoid railed value.

Table 17. Standard self-test verification test limits, high g X-axis

Device type	NXP trim value (LSB/g)	User programmed range (g)	User programmed gain	Self-test limits, SNSDATA register value [1]) (LSB)		Pass / fail criteria
				Minimum	Maximum	
High g X	10.9465	46.875	3.9922	2886	6743	Min Limit ≤ Result ≤ Max Limit
		50	3.7422	2705	6321	
		60	3.1172	2253	5265	
		62	3.0156	2180	5094	
		62.5	2.9922	2163	5054	
		100	1.8711	1352	3161	
		105	1.7813	1287	3009	
		112.5	1.6641	1203	2811	
		125	1.4961	1081	2527	
		128	1.4609	1056	2468	
		150	1.2461	900	2105	
		187	1.0000	723	1689	
		250	0.7480	540	1264	
		312.5	0.5977	432	1010	
		375	0.4990	360	843	
		500	0.3740	270	632	
		748	0.2500	180	423	

[1] If the self-test limits exceed 2^{11}-1 (2047) LSB, self-test value must be read using reads of the SNSDATA register to avoid railed value.

Table 18. Standard self-test verification test limits, high g Z-axis

Device type	NXP trim value (LSB/g)	User programmed range (g)	User programmed gain	Self-test limits, SNSDATA register value[1]) (LSB)		Pass / fail criteria
				Minimum	Maximum	
High g Z	10.9465	46.875	3.9922	3329	7773	Min Limit ≤ Result ≤ Max Limit
		50	3.7422	3120	7287	
		60	3.1172	2599	6070	
		62	3.0156	2515	5872	
		62.5	2.9922	2495	5826	
		100	1.8711	1560	3644	
		105	1.7813	1485	3469	
		112.5	1.6641	1387	3240	
		125	1.4961	1247	2913	
		128	1.4609	1218	2845	
		150	1.2461	1039	2427	
		187	1.0000	834	1947	
		250	0.7480	623	1457	
		312.5	0.5977	498	1164	
		375	0.4990	416	972	
		500	0.3740	311	729	
		748	0.2500	208	487	

[1] If the self-test limits exceed 2^{11}-1 (2047) LSB, self-test value must be read using reads of the SNSDATA register to avoid railed value.

8.11.4.1.3 Optional self-test accuracy verification

The resulting positive and negative self-test values from the tests in Figure 22 through Figure 25 can also be compared against the self-test values stored in the device OTP (see Section 8.9 "Optional read and record stored self-test data"). This method provides a much tighter accuracy of the self-test limits. Independent positive and negative self-test values are stored for each channel. The stored values are equal to the absolute value of the difference between the self-test and offset values measured during the NXP factory trim/test at nominal temperature. Self-test stored values are captured with the user gain set to 1. Thus, the self-test limits must be calculated and then scaled with the user gain. The equation used to convert the stored self-test value to minimum and maximum self-test accuracy limits for the SNSDATA register values is shown below. Table 19 shows some example self-test accuracy limits for different ranges and settings.

$$AnalogSelfTest_{AccMin@UserGain} = Truncate[ST_{Stored}*2*(1-\Delta STACC\_T)*(UserGain)]$$

$$AnalogSelfTest_{AccMax@UserGain} = RoundUp[ST_{Stored}*2*(1+\Delta STACC\_T)*(UserGain)]$$

Where:

- $\mathit{AnalogSelfTest}_{AccMin@UserGain}$ = The SNSDATA register self-test accuracy min limit for the configured user gain
- $\mathit{AnalogSelfTest}_{AccMax@UserGain}$ = The SNSDATA register self-test accuracy min limit for the configured user gain
- ST_{Stored} = The stored self-test value for the associated channel and self-test
- *ΔSTACC_T* = The self-test accuracy specified tolerance
- *UserGain* = The configured gain, calculated as shown in Section Section 8.6.2 "Signal chain user gain selection"

Table 19. Self-test accuracy verification example test limits

Device type	User programmed range (g)	User programmed gain	Example self-test stored value (LSB)	Self-test accuracy limits SNSDATA register value [1]) (LSB)		Pass / fail criteria
				Minimum	Minimum	
Medium g X	62	1	1000.00	1800	2200	Min Limit ≤ Result ≤ Max Limit
	62	1	1470.00	2646	3234	
	62	1	1800.00	3240	3960	
	125	0.4961	1000.00	892	1092	
	125	0.4961	1470.00	1312	1605	
	125	0.4961	1800.00	1607	1965	
	25	2.477	1000.00	4458	5450	
	25	2.477	1470.00	6554	8011	
	25	2.477	1800.00	8025	9809	
	15	3.875	1000.00	6975	8525	
	15	3.875	1470.00	10253	12532	
	15	3.875	1800.00	12555	15345	

[1] If the self-test limits exceed 2^{11}-1 (2047) LSB, self-test value must be read using reads of the SNSDATA register to avoid railed value.

8.11.4.1.4 Optional self-test cross-coupling verification

The self-test cross-coupling values can also be compared against the self-test cross-coupling output limits specified in the datasheet. The limits are independent of range so the cross-coupling values are directly compared against the specified limits: ΔSTCh0_1 and ΔSTCh1_0.

8.11.4.1.5 Optional post self-test delta offset verification

The post self-test offset values can also be compared against the self-test delta offset limits specified in the datasheet. The limits are independent of range so the post self-test offset delta values are directly compared against the specified limits: ΔSTOFF_T.

8.11.5 Restart offset cancellation fast startup

Once Self-test is complete, if offset cancellation is enabled, the offset cancellation fast startup is reset to re-zero the offset cancellation. Figure 26 shows the procedure for restarting the fast offset cancellation.

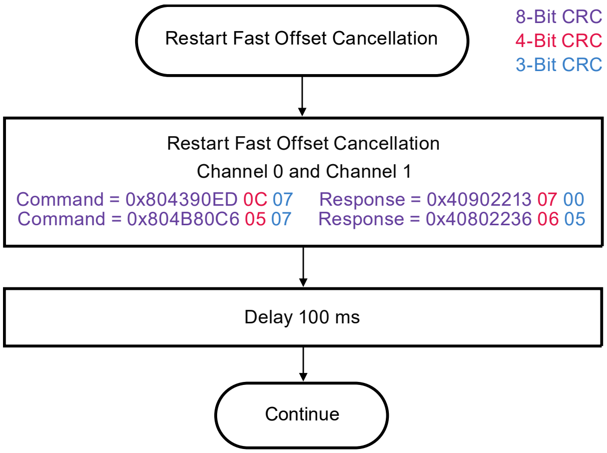

Restart Fast Offset Cancellation
8-Bit CRC
4-Bit CRC
3-Bit CRC
Restart Fast Offset Cancellation
Channel 0 and Channel 1
Command = 0x804390ED 0C 07
Response = 0x40902213 07 00
Command = 0x804B80C6 05 07
Response = 0x40802236 06 05
Delay 100 ms
Continue
aaa-040171

Figure 26. Restart offset cancellation fast startup

8.12 Transition to normal mode

Once all self-test procedures are completed and verified, the system can transition the device from initialization to normal mode. This is done by setting the ENDINIT bit.

Figure 27 shows the command to set the ENDINIT bit and an example Normal Mode command sequence.

- The command order for normal mode is chosen to provide status, Channel 0 data, and Channel 1 data. The device status is latched on the falling edge of the SS_B for the response. This allows for the request to be transmitted during the previous loop.
- Each sensor data response also includes status. Thus, the status can change from the beginning of the current loop to the end of the current loop.
  - Channel specific errors (CHx_ERR bit set, offset errors, etc.) should always use the most recent status to determine the state of the data for that loop.
  - Device specific error status (memory errors, supply errors) are asynchronous and may change between channels. If a device specific error occurs at any time in the current loop, all sensor data for that loop should be considered to have the error status.

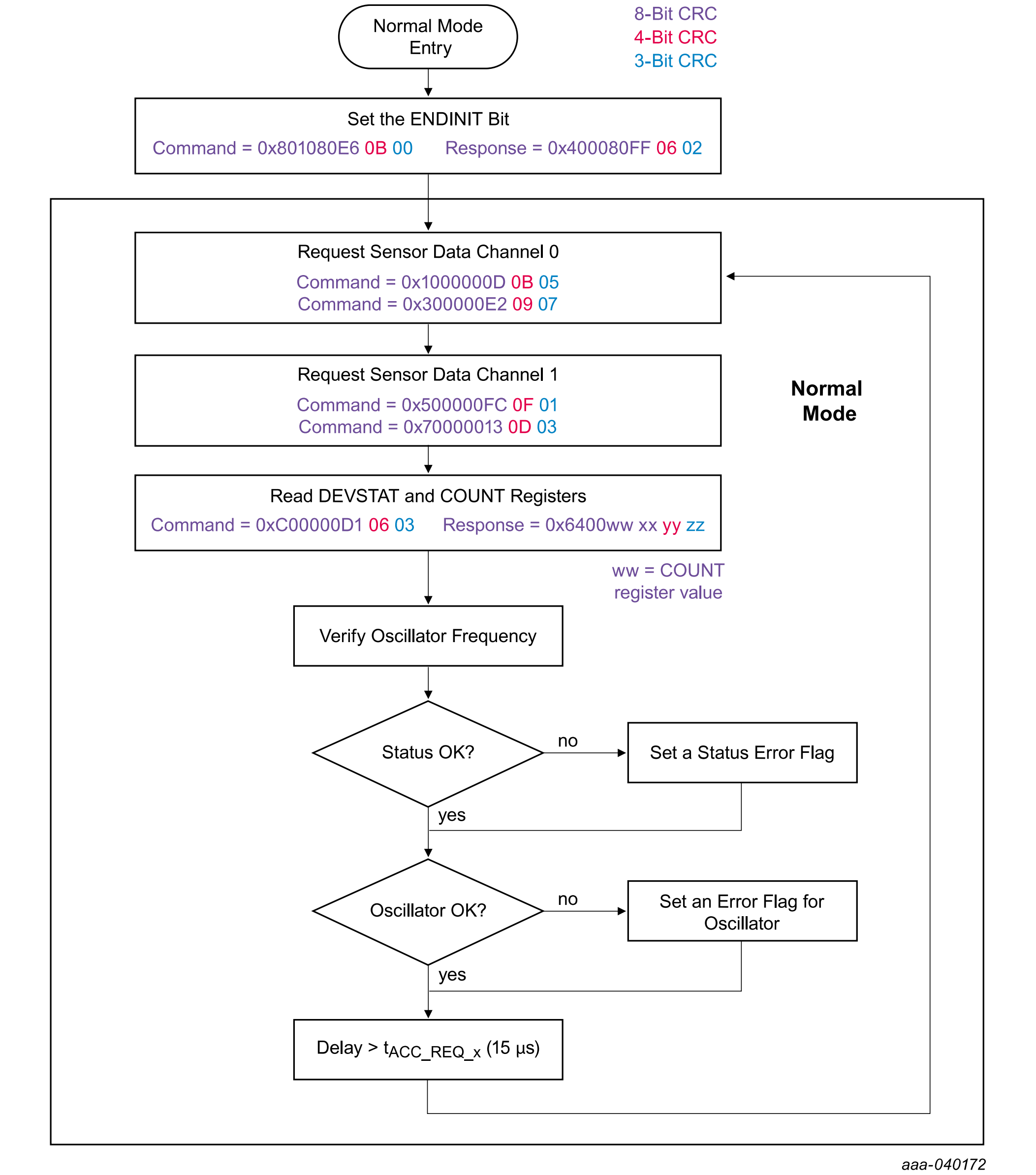

Normal Mode Entry
8-Bit CRC
4-Bit CRC
3-Bit CRC
Set the ENDINIT Bit
Command = 0x801080E6 0B 00
Response = 0x400080FF 06 02
Request Sensor Data Channel 0
Command = 0x1000000D 0B 05
Command = 0x300000E2 09 07
Request Sensor Data Channel 1
Command = 0x500000FC 0F 01
Command = 0x70000013 0D 03
Normal Mode
Read DEVSTAT and COUNT Registers
Command = 0xC00000D1 06 03
Response = 0x6400ww xx yy zz
ww = COUNT register value
Verify Oscillator Frequency
Status OK?
no
Set a Status Error Flag
yes
Oscillator OK?
no
Set an Error Flag for Oscillator
yes
Delay > $t_{ACC\_REQ\_x}$ (15 µs)
aaa-040172

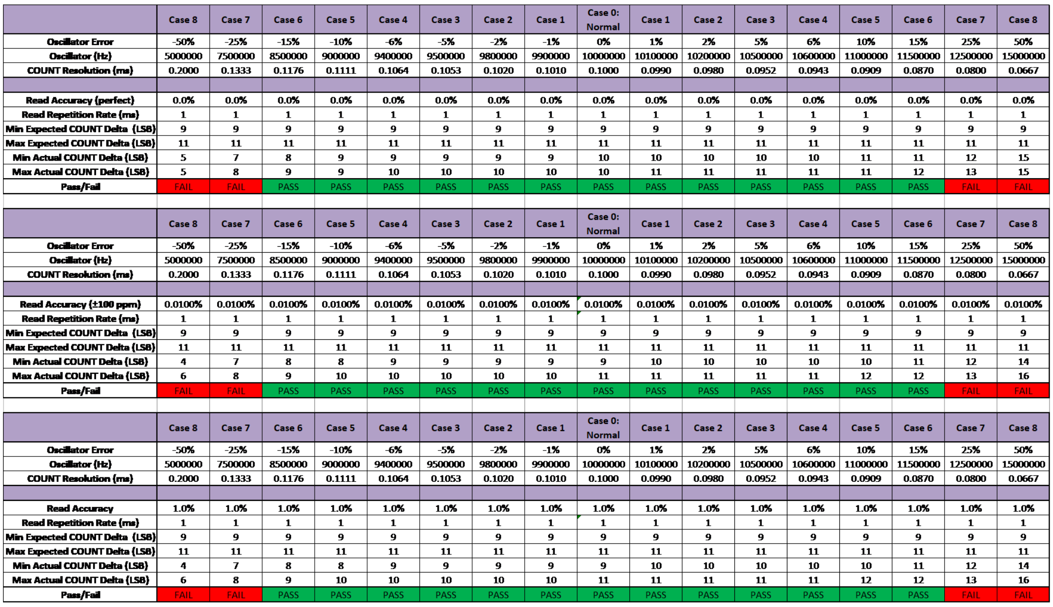
Figure 27. Transition to Normal Mode

9 Optional Continuous Oscillator Verification

In addition to the coarse oscillator verification during initialization, the device oscillator can be continuously verified for accuracy via a continuous monitor of the COUNT register during normal mode. Figure 27 shows a recommended SPI command and response sequence for normal mode. The sequence includes a read of the DEVSTAT and COUNT register at a periodic rate. Figure 28, Figure 29 and Figure 30 below show the accuracy of the COUNT register oscillator verification method with different read repetition rates. Note that the COUNT register will roll over every 25.6 ms typically. Each of the examples below must account for the possibility of one counter rollover. A summary of the accuracy is shown in Table 20

Table 20. Oscillator verification accuracy summary

Count register read repetition rate (ms)	Master oscillator accuracy	Oscillator accuracy detection (%)
1	0 ppm	25
1	± 100 ppm	25
1	± 1%	25
6	0 ppm	10
6	± 100 ppm	10
6	± 1%	10
24	0 ppm	6
24	± 100 ppm	6
24	± 1%	10

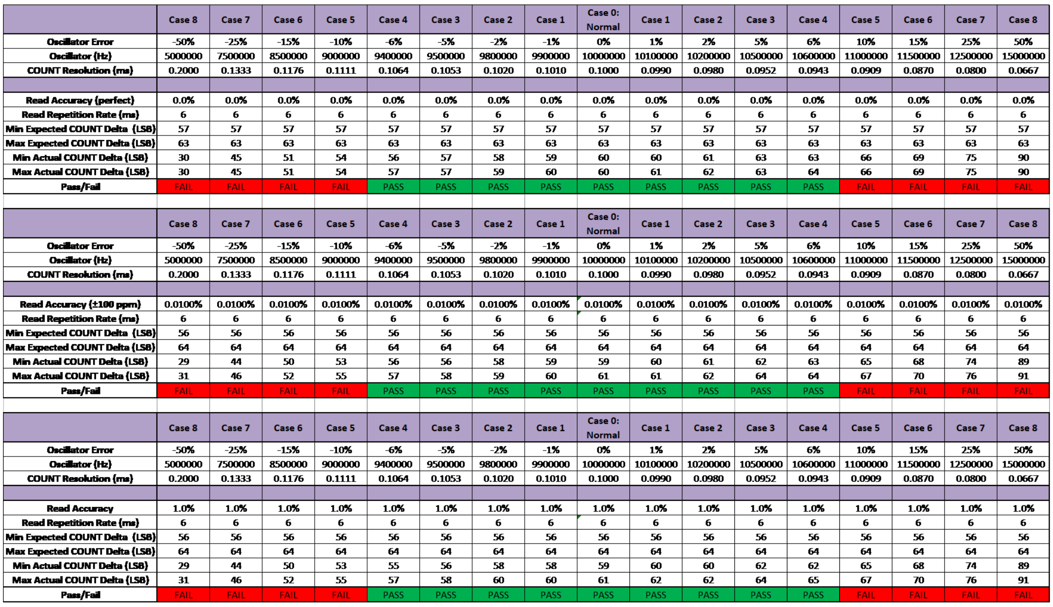


	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy (perfect)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Read Repetition Rate (ms)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Min Expected COUNT Delta (LSB)	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Max Expected COUNT Delta (LSB)	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
Min Actual COUNT Delta (LSB)	5	7	8	9	9	9	9	9	10	10	10	10	10	11	11	12	15
Max Actual COUNT Delta (LSB)	5	8	9	9	10	10	10	10	10	11	11	11	11	11	12	13	15
Pass/Fail	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL

	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy (±100 ppm)	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%
Read Repetition Rate (ms)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Min Expected COUNT Delta (LSB)	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Max Expected COUNT Delta (LSB)	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
Min Actual COUNT Delta (LSB)	4	7	8	8	9	9	9	9	9	10	10	10	10	10	11	12	14
Max Actual COUNT Delta (LSB)	6	8	9	10	10	10	10	10	11	11	11	11	11	12	12	13	16
Pass/Fail	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL

	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%
Read Repetition Rate (ms)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Min Expected COUNT Delta (LSB)	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Max Expected COUNT Delta (LSB)	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
Min Actual COUNT Delta (LSB)	4	7	8	8	9	9	9	9	9	10	10	10	10	10	11	12	14
Max Actual COUNT Delta (LSB)	6	8	9	10	10	10	10	10	11	11	11	11	11	12	12	13	16
Pass/Fail	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL

Figure 28. Oscillator verification accuracy: 1 ms read repetition rate

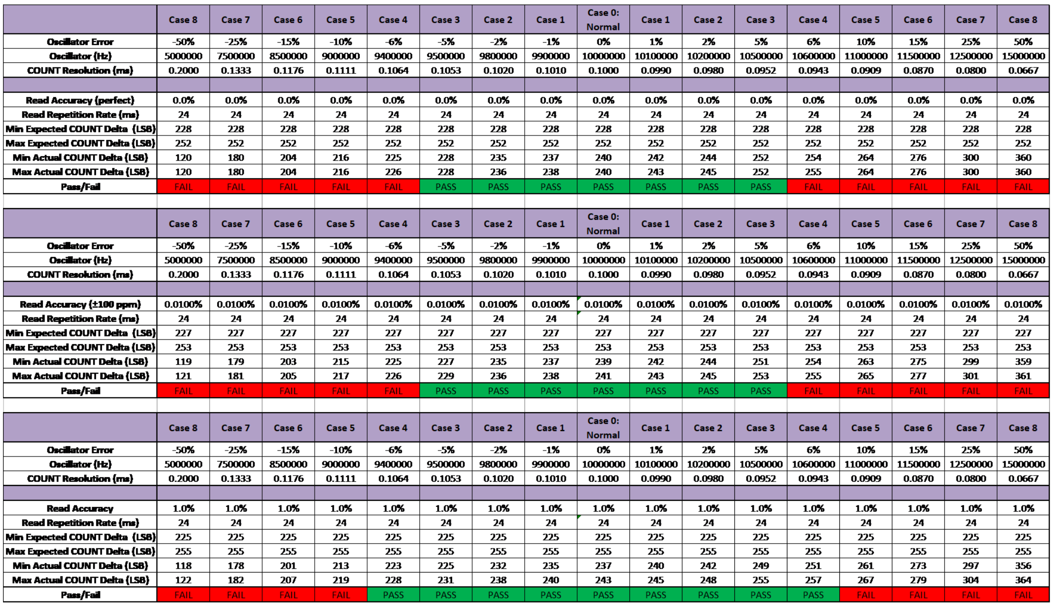


	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy (perfect)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Read Repetition Rate (ms)	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Min Expected COUNT Delta (LSB)	57	57	57	57	57	57	57	57	57	57	57	57	57	57	57	57	57
Max Expected COUNT Delta (LSB)	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63	63
Min Actual COUNT Delta (LSB)	30	45	51	54	56	57	58	59	60	60	61	63	63	66	69	75	90
Max Actual COUNT Delta (LSB)	30	45	51	54	57	57	59	60	60	61	62	63	64	66	69	75	90
Pass/Fail	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL

	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy (±100 ppm)	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%
Read Repetition Rate (ms)	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Min Expected COUNT Delta (LSB)	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56
Max Expected COUNT Delta (LSB)	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
Min Actual COUNT Delta (LSB)	29	44	50	53	56	56	58	59	59	60	61	62	63	65	68	74	89
Max Actual COUNT Delta (LSB)	31	46	52	55	57	58	59	60	61	61	62	64	64	67	70	76	91
Pass/Fail	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL

	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%
Read Repetition Rate (ms)	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
Min Expected COUNT Delta (LSB)	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56	56
Max Expected COUNT Delta (LSB)	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
Min Actual COUNT Delta (LSB)	29	44	50	53	55	56	58	58	59	60	60	62	62	65	68	74	89
Max Actual COUNT Delta (LSB)	31	46	52	55	57	58	60	60	61	62	62	64	65	67	70	76	91
Pass/Fail	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL

Figure 29. Oscillator verification accuracy: 6 ms read repetition rate



	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy (perfect)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Read Repetition Rate (ms)	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
Min Expected COUNT Delta (LSB)	228	228	228	228	228	228	228	228	228	228	228	228	228	228	228	228	228
Max Expected COUNT Delta (LSB)	252	252	252	252	252	252	252	252	252	252	252	252	252	252	252	252	252
Min Actual COUNT Delta (LSB)	120	180	204	216	225	228	235	237	240	242	244	252	254	264	276	300	360
Max Actual COUNT Delta (LSB)	120	180	204	216	226	228	236	238	240	243	245	252	255	264	276	300	360
Pass/Fail	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL	FAIL

	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy (±100 ppm)	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%	0.0100%
Read Repetition Rate (ms)	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
Min Expected COUNT Delta (LSB)	227	227	227	227	227	227	227	227	227	227	227	227	227	227	227	227	227
Max Expected COUNT Delta (LSB)	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253
Min Actual COUNT Delta (LSB)	119	179	203	215	225	227	235	237	239	242	244	251	254	263	275	299	359
Max Actual COUNT Delta (LSB)	121	181	205	217	226	229	236	238	241	243	245	253	255	265	277	301	361
Pass/Fail	FAIL	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL	FAIL

	Case 8	Case 7	Case 6	Case 5	Case 4	Case 3	Case 2	Case 1	Case 0: Normal	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Oscillator Error	-50%	-25%	-15%	-10%	-6%	-5%	-2%	-1%	0%	1%	2%	5%	6%	10%	15%	25%	50%
Oscillator (Hz)	5000000	7500000	8500000	9000000	9400000	9500000	9800000	9900000	10000000	10100000	10200000	10500000	10600000	11000000	11500000	12500000	15000000
COUNT Resolution (ms)	0.2000	0.1333	0.1176	0.1111	0.1064	0.1053	0.1020	0.1010	0.1000	0.0990	0.0980	0.0952	0.0943	0.0909	0.0870	0.0800	0.0667
Read Accuracy	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%
Read Repetition Rate (ms)	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
Min Expected COUNT Delta (LSB)	225	225	225	225	225	225	225	225	225	225	225	225	225	225	225	225	225
Max Expected COUNT Delta (LSB)	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
Min Actual COUNT Delta (LSB)	118	178	201	213	223	225	232	235	237	240	242	249	251	261	273	297	356
Max Actual COUNT Delta (LSB)	122	182	207	219	228	231	238	240	243	245	248	255	257	267	279	304	364
Pass/Fail	FAIL	FAIL	FAIL	FAIL	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	PASS	FAIL	FAIL	FAIL	FAIL

Figure 30. Oscillator verification accuracy: 24 ms read repetition rate

10 Optional User Diagnostics

This section describes some additional system level diagnostics that can be used to improve the safety performance of the device in its intended application. These diagnostics are not inherent in the device and, if used, must be conducted externally.

10.1 Continuous railed output detection

Using the sensor data from each source collected during normal mode, the following procedure can be used to detect a railed output. The example procedure uses the configurations described in this application note and is documented for only one data source, although it should be completed on each data source.

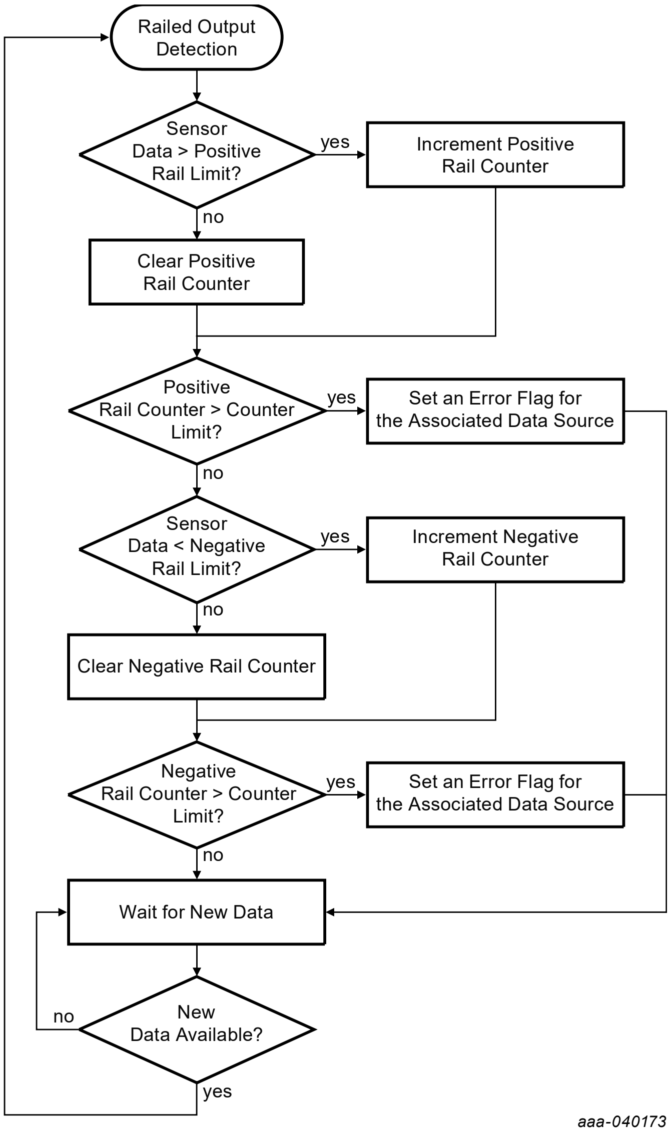

Railed Output Detection
Sensor Data > Positive Rail Limit?
yes
Increment Positive Rail Counter
no
Clear Positive Rail Counter
Positive Rail Counter > Counter Limit?
yes
Set an Error Flag for the Associated Data Source
no
Sensor Data < Negative Rail Limit?
yes
Increment Negative Rail Counter
no
Clear Negative Rail Counter
Negative Rail Counter > Counter Limit?
yes
Set an Error Flag for the Associated Data Source
no
Wait for New Data
no
New Data Available?
yes
aaa-040173

Figure 31. Continuous railed output detection

10.2 Continuous sustained step detection

Using the sensor data from each source collected during normal mode, the following procedure can be used to detect a sustained step on the output. The example procedure uses the configurations described in this application note. Step detection limits are gain, range, and application dependent. Therefore, the user must determine the limits for the step detection based on the application.

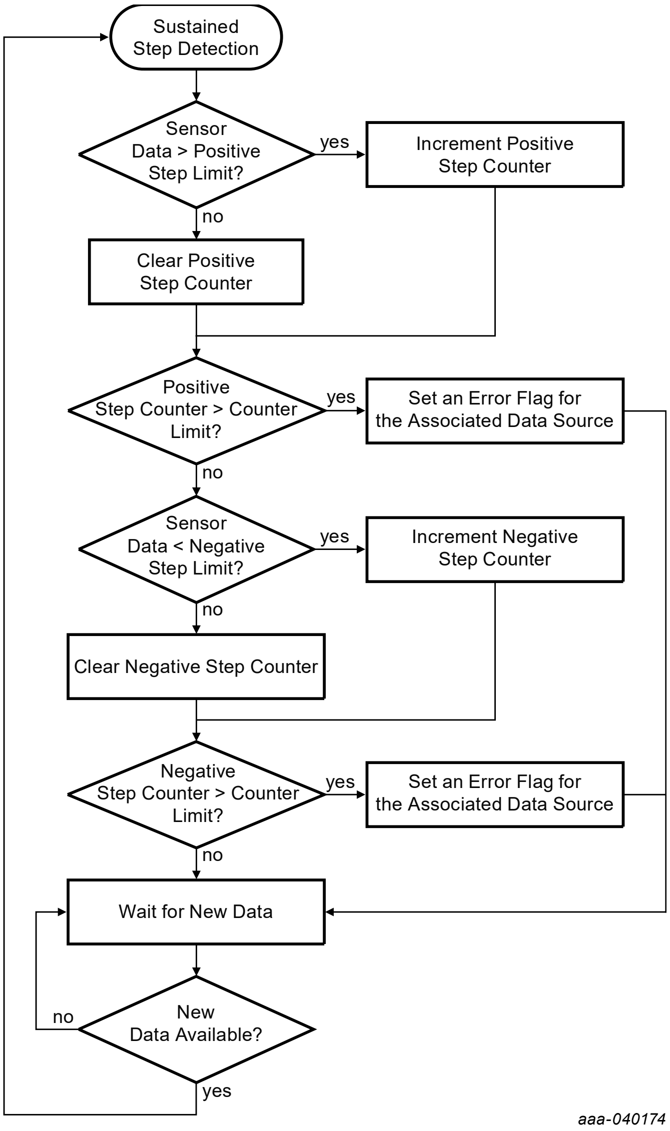

Sustained Step Detection
Sensor Data > Positive Step Limit?
yes
Increment Positive Step Counter
no
Clear Positive Step Counter
Positive Step Counter > Counter Limit?
yes
Set an Error Flag for the Associated Data Source
no
Sensor Data < Negative Step Limit?
yes
Increment Negative Step Counter
no
Clear Negative Step Counter
Negative Step Counter > Counter Limit?
yes
Set an Error Flag for the Associated Data Source
no
Wait for New Data
no
New Data Available?
yes
aaa-040174

Figure 32. Continuous sustained step detection

10.3 Normal Mode memory protection verification

When ENDINIT is set, the UF2 user memory block is locked and verified via CRC as described in the datasheet. The following procedure tests the memory lock to confirm that writes to the UF2 array are prevented. In this example, the PDCM_RSPST3 registers are used for memory protection verification as they are unused for the SPI application. Other registers can be used as long as the function for the register being written to is considered.

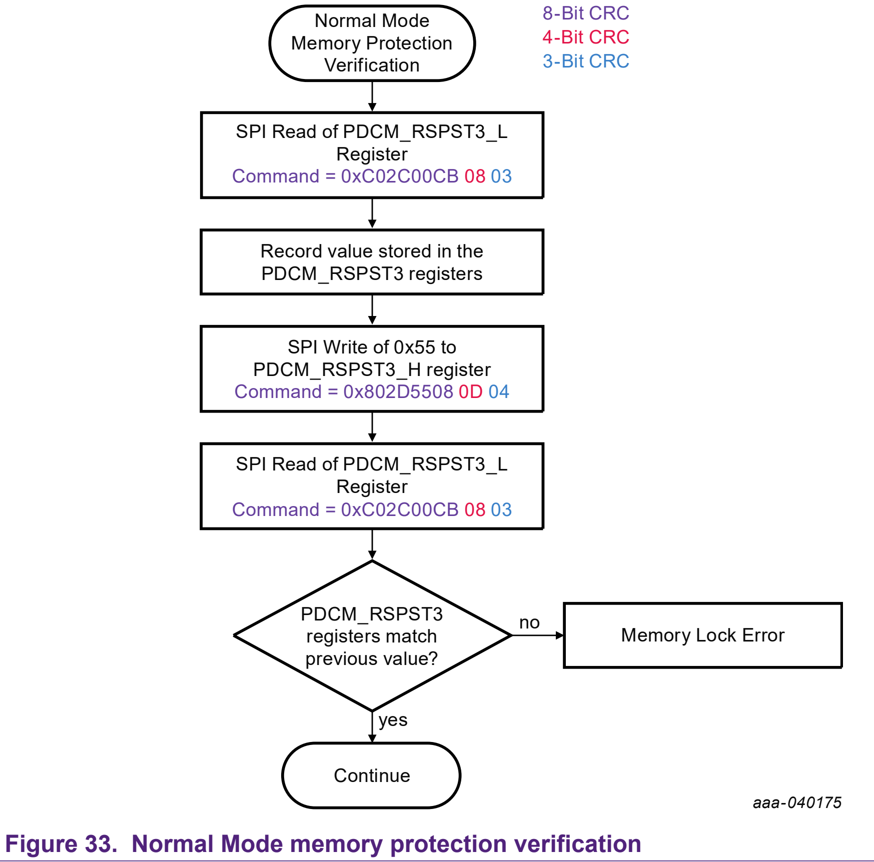

Normal Mode Memory Protection Verification
8-Bit CRC
4-Bit CRC
3-Bit CRC
SPI Read of PDCM_RSPST3_L Register
Command = 0xC02C00CB 08 03
Record value stored in the PDCM_RSPST3 registers
SPI Write of 0x55 to PDCM_RSPST3_H register
Command = 0x802D5508 0D 04
SPI Read of PDCM_RSPST3_L Register
Command = 0xC02C00CB 08 03
PDCM_RSPST3 registers match previous value?
no
Memory Lock Error
yes
Continue
aaa-040175

Figure 33. Normal Mode memory protection verification

11 Bus Contention Considerations

Certain SPI failure modes on the FXLS9xxxx can prevent or corrupt communication from other devices on the SPI bus. Likewise, some SPI failure modes of other devices on the SPI bus can prevent or corrupt communication from the FXLS9xxxx device(s).

The following internal safety mechanisms enable the detection of corrupted communication to and from the FXLS9xxxx device(s) due to a failure mode from another device on the SPI bus. Reference the datasheet for more details on each safety mechanism.

1. SPI MOSI CRC verification
2. Verification of SCLK low when SS_B is asserted
3. Verification of SCLK low when SS_B is deasserted
4. Verification that the number of SCLKs while SS_B is asserted is exactly 32
5. SPI MOSI fixed bit verification for all commands
   - The device verifies the state of all non-command and non-CRC bits in a command. Thus, all 32 SPI bits are verified.
6. SPI Sensor Data Request SOURCEID verification
   - The source of the data is verified against the requested SOURCEID prior to transmission. If a mismatch occurs, an error message is transmitted instead of sensor data.
7. SPI MISO Error Detection

The following external safety mechanisms are required to enable the detection of corrupted communication from the FXLS9xxxx device(s) due to a failure mode from another device on the SPI bus.

1. SPI MISO CRC verification
2. SPI sensor data SOURCEID verification

The following external safety mechanisms are optional to improve the detection of corrupted communication from the FXLS9xxxx device(s) due to a failure mode from another device on the SPI bus.

1. SPI MISO fixed bit verification for all commands
2. SPI sensor data message counter(s) verification for 4-bit CRC implementation

12 Summary and Conclusion

This application note describes the recommended procedures for initializing and configuring FXLS9xxx devices on a SPI bus, completing self-test on the devices, and transitioning the devices to Normal Mode. These recommended procedures are essential to meeting the functional safety requirements of the intended system.

13 Appendix

13.1 SPI state transition diagram

Figure 34 shows a state transition diagram for the internal SPI controller.

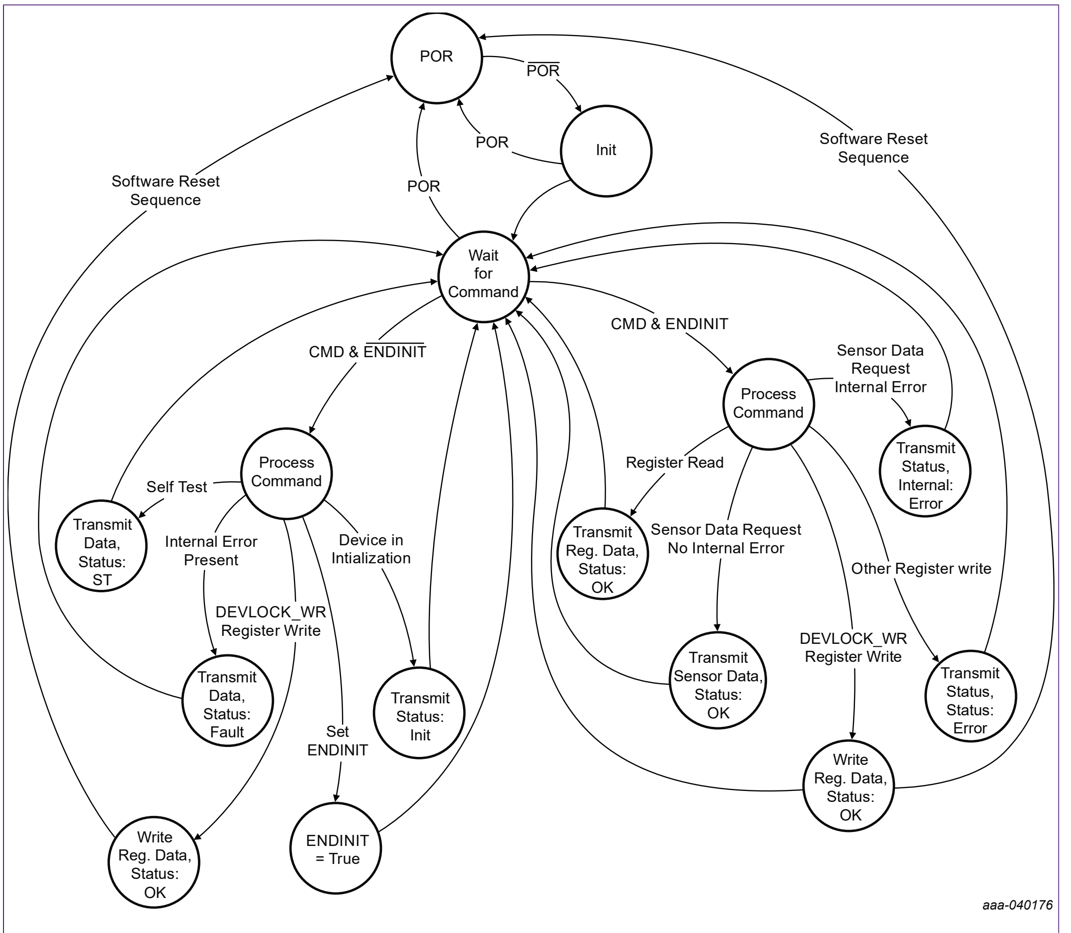

POR
$\overline{\text{POR}}$
Init
POR
POR
Software Reset Sequence
Software Reset Sequence
Wait for Command
CMD & $\overline{\text{ENDINIT}}$
CMD & ENDINIT
Process Command
Process Command
Sensor Data Request Internal Error
Transmit Status, Internal: Error
Self Test
Transmit Data, Status: ST
Internal Error Present
Device in Intialization
DEVLOCK_WR Register Write
Transmit Data, Status: Fault
Set ENDINIT
Transmit Status: Init
ENDINIT = True
Write Reg. Data, Status: OK
Register Read
Transmit Reg. Data, Status: OK
Sensor Data Request No Internal Error
Transmit Sensor Data, Status: OK
DEVLOCK_WR Register Write
Write Reg. Data, Status: OK
Other Register write
Transmit Status, Status: Error
aaa-040176

Figure 34. SPI state transition diagram

13.2 SPI startup error response timing

Figure 35 shows a representative timing diagram for SPI Start Up error conditions.

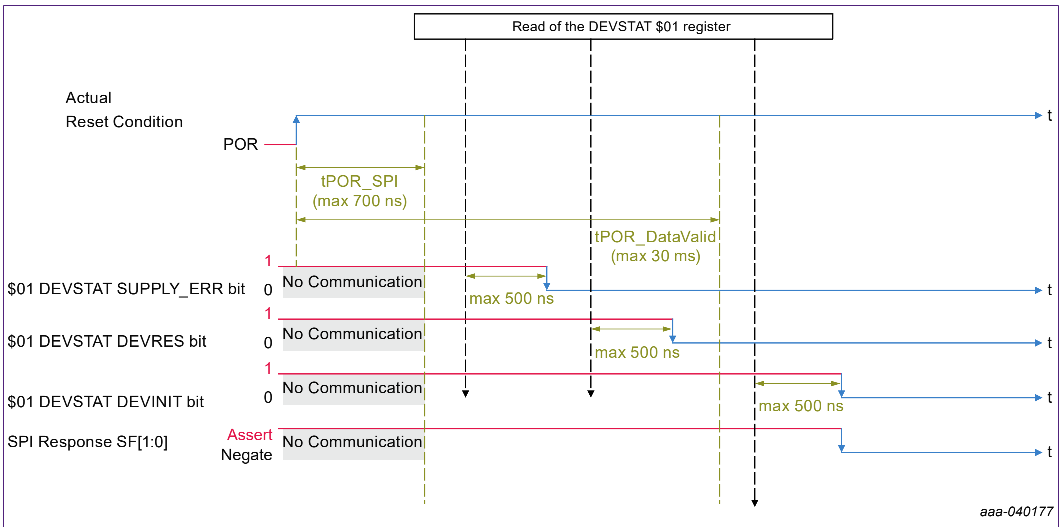


Figure 35. SPI start up error conditions

13.3 Internal error response timing

13.3.1 Standard internal error conditions: transient

Figure 36 shows a representative timing diagram for transient internal error conditions.

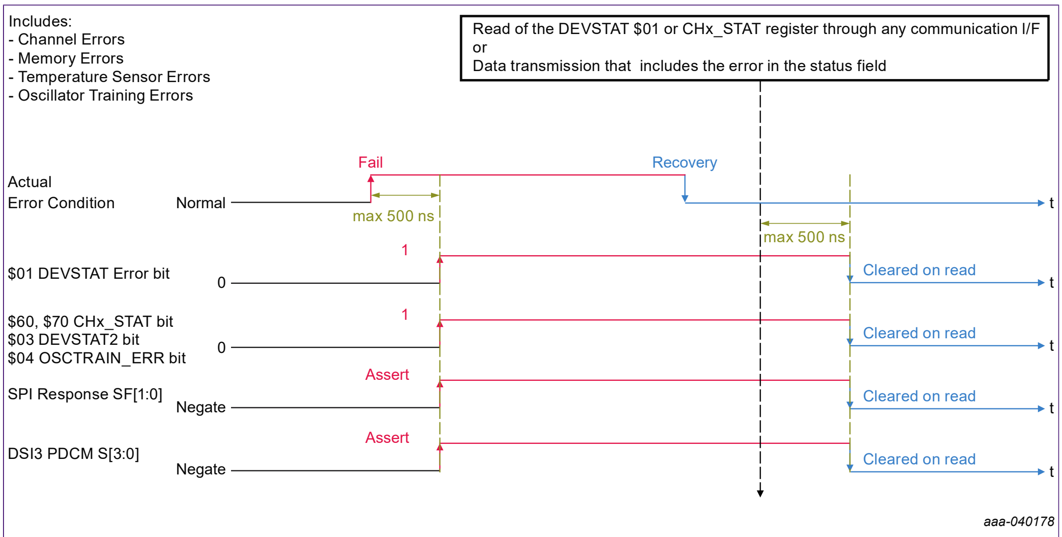


Figure 36. Standard internal error conditions: transient

13.3.2 Standard internal error conditions: persistent

Figure 37 shows a representative timing diagram for persistent internal error conditions.

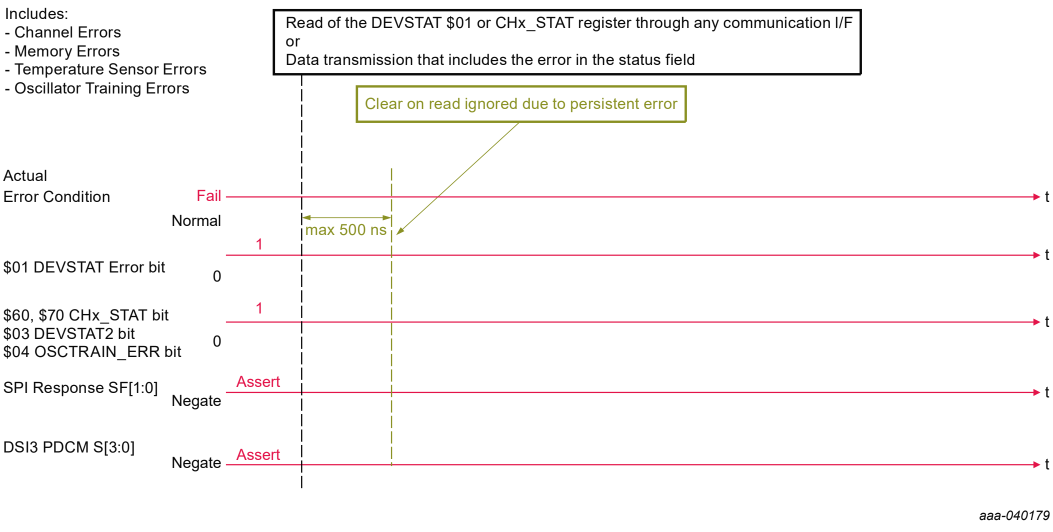

Includes:
- Channel Errors
- Memory Errors
- Temperature Sensor Errors
- Oscillator Training Errors
Read of the DEVSTAT $01 or CHx_STAT register through any communication I/F
or
Data transmission that includes the error in the status field
Clear on read ignored due to persistent error
Actual
Error Condition
Fail
Normal
max 500 ns
t
$01 DEVSTAT Error bit
1
0
$60, $70 CHx_STAT bit
$03 DEVSTAT2 bit
$04 OSCTRAIN_ERR bit
SPI Response SF[1:0]
Assert
Negate
DSI3 PDCM S[3:0]
aaa-040179

Figure 37. Standard internal error conditions: persistent

13.3.3 Apparent external error conditions: transient

Figure 38 shows a representative timing diagram for apparent transient external error conditions.

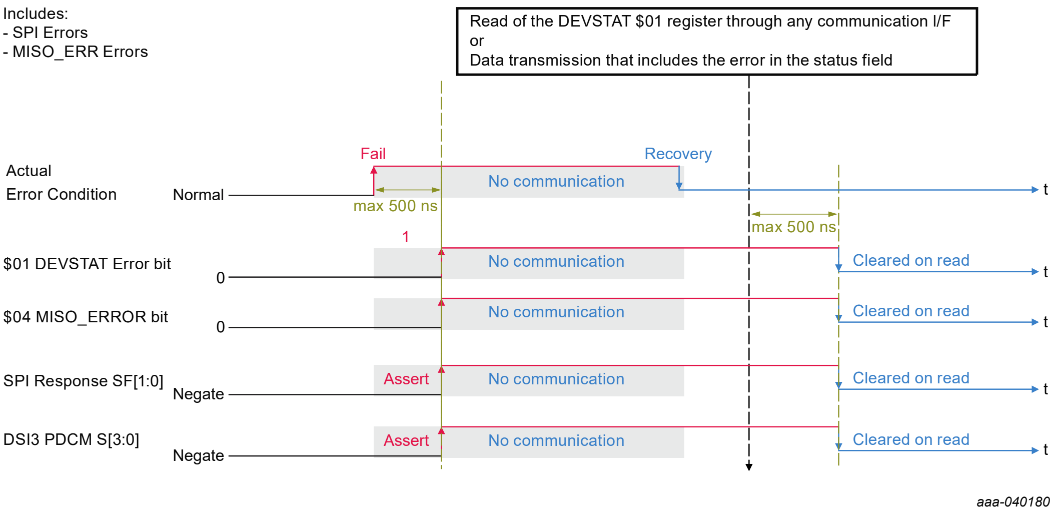


Figure 38. External error conditions: transient

13.3.4 Apparent external error conditions: persistent

Figure 39 shows a representative timing diagram for apparent Persistent external error conditions.

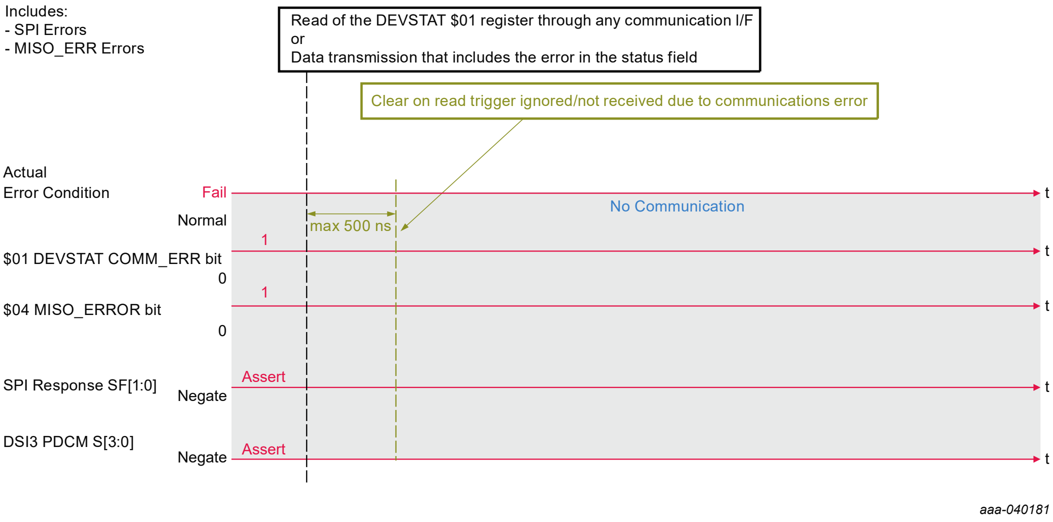


Figure 39. External error conditions: persistent

13.3.5 Supply errors: transient

Figure 40 shows a representative timing diagram for transient supply error conditions.

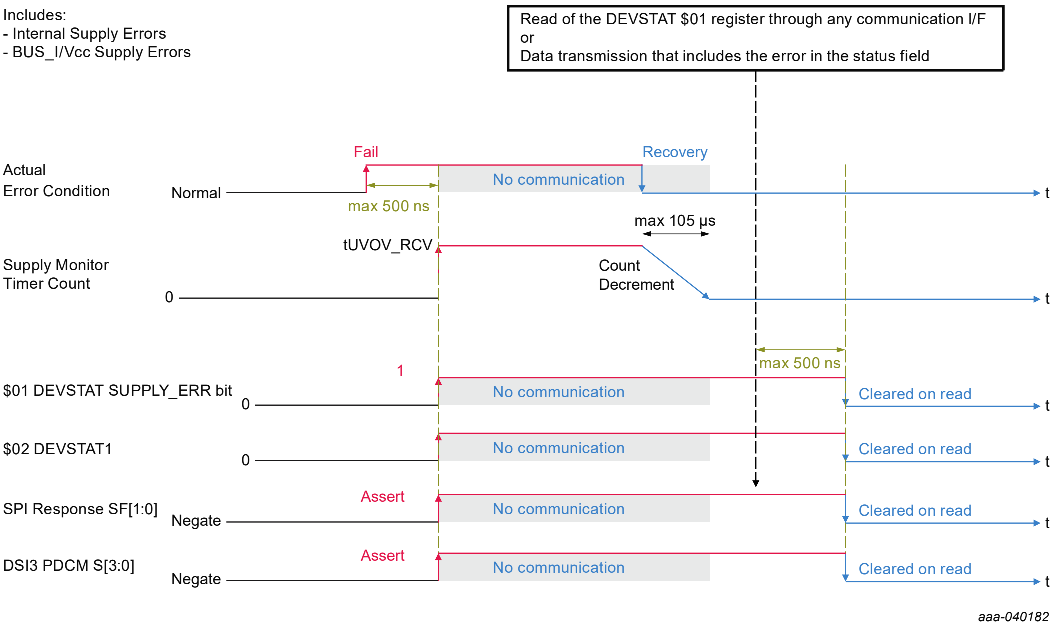
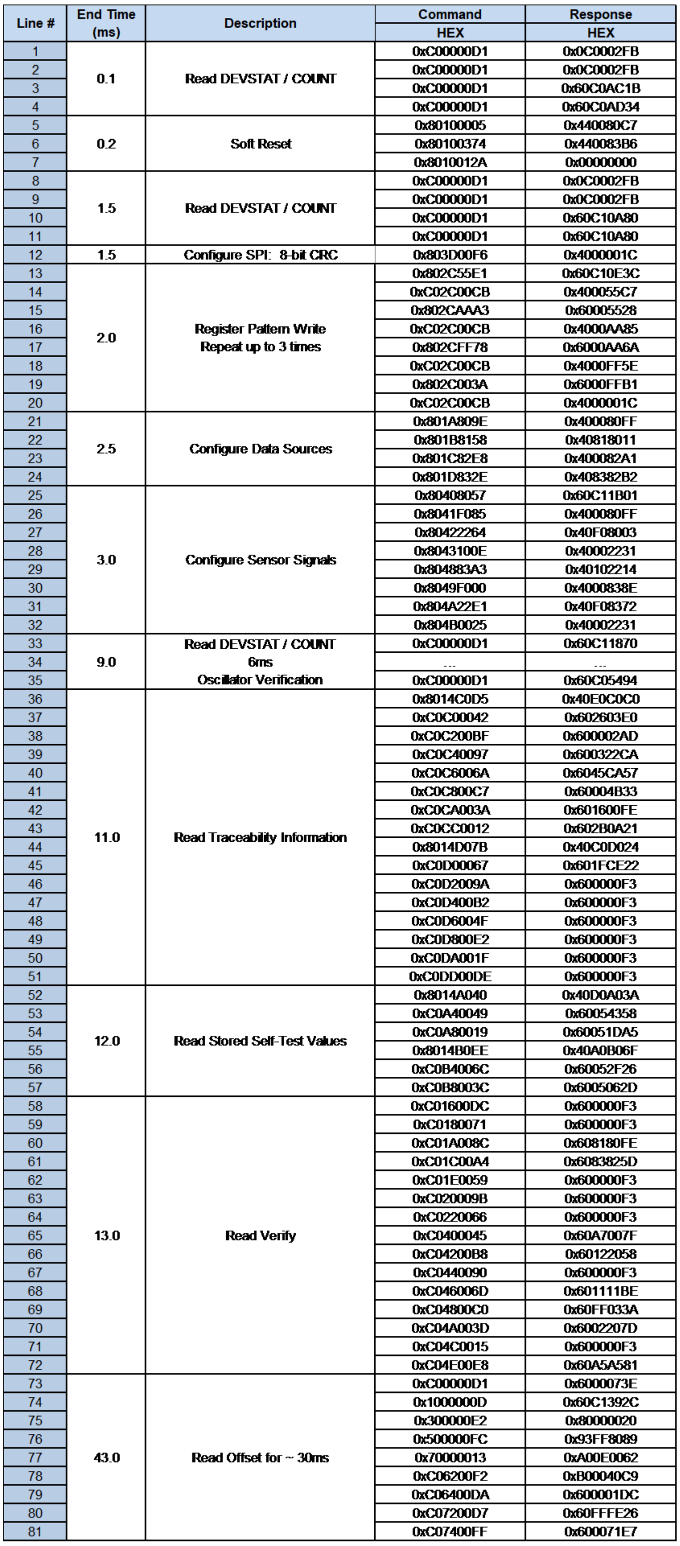


Figure 40. Supply Error Conditions: Transient

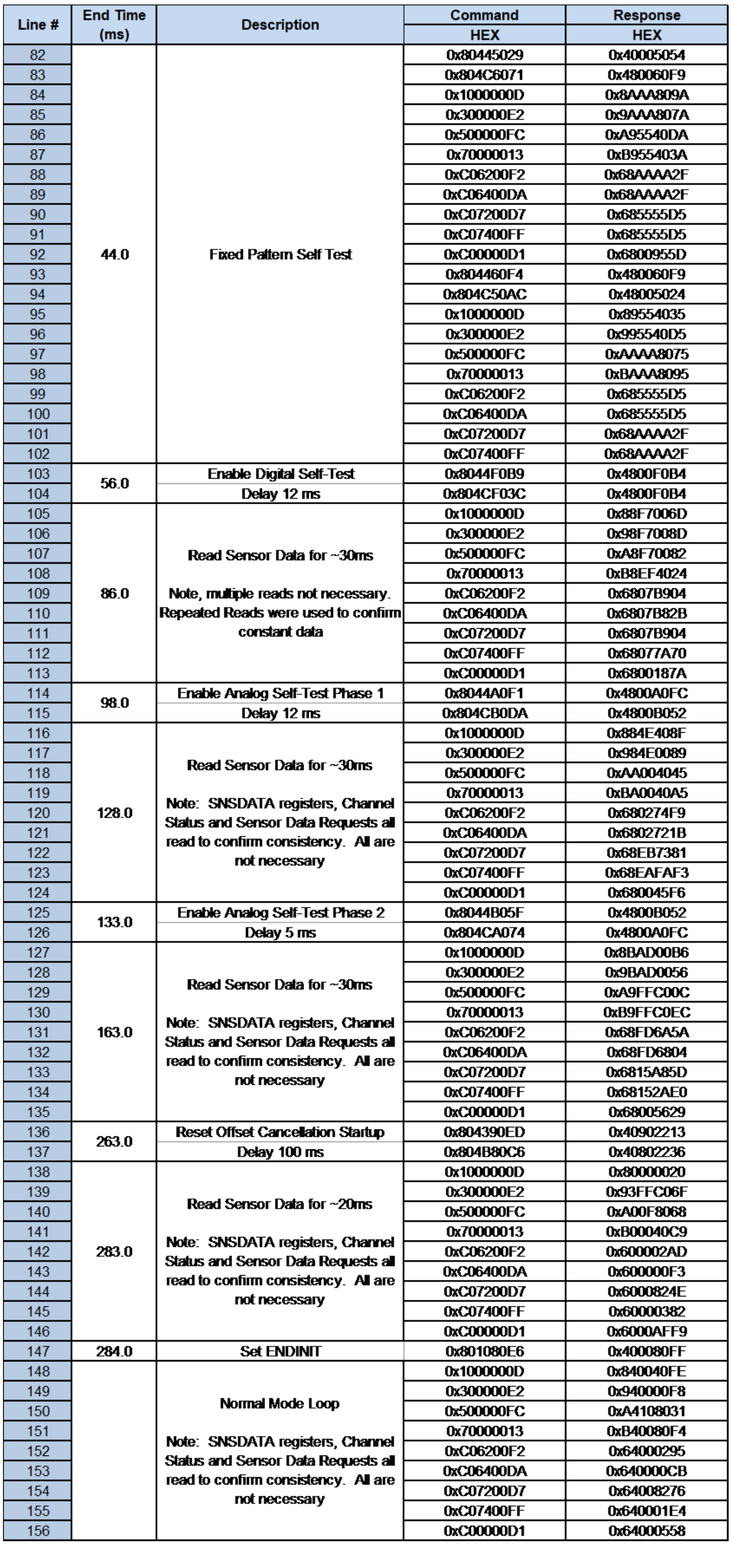
13.4 Example SPI command list

The commands and responses for the sequence used in this application note are summarized below, all using the 8-bit CRC.



Line #	End Time (ms)	Description	Command	Response
			HEX	HEX
1	0.1	Read DEVSTAT / COUNT	0xC00000D1	0x0C0002FB
2			0xC00000D1	0x0C0002FB
3			0xC00000D1	0x60C0AC1B
4			0xC00000D1	0x60C0AD34
5	0.2	Soft Reset	0x80100005	0x440080C7
6			0x80100374	0x440083B6
7			0x8010012A	0x00000000
8	1.5	Read DEVSTAT / COUNT	0xC00000D1	0x0C0002FB
9			0xC00000D1	0x0C0002FB
10			0xC00000D1	0x60C10A80
11			0xC00000D1	0x60C10A80
12	1.5	Configure SPI: 8-bit CRC	0x803D00F6	0x4000001C
13	2.0	Register Pattern Write Repeat up to 3 times	0x802C55E1	0x60C10E3C
14			0xC02C00CB	0x400055C7
15			0x802CAAA3	0x60005528
16			0xC02C00CB	0x4000AA85
17			0x802CFF78	0x6000AA6A
18			0xC02C00CB	0x4000FF5E
19			0x802C003A	0x6000FFB1
20			0xC02C00CB	0x4000001C
21	2.5	Configure Data Sources	0x801A809E	0x400080FF
22			0x801B8158	0x40818011
23			0x801C82E8	0x400082A1
24			0x801D832E	0x408382B2
25	3.0	Configure Sensor Signals	0x80408057	0x60C11B01
26			0x8041F085	0x400080FF
27			0x80422264	0x40F08003
28			0x8043100E	0x40002231
29			0x804883A3	0x40102214
30			0x8049F000	0x4000838E
31			0x804A22E1	0x40F08372
32			0x804B0025	0x40002231
33	9.0	Read DEVSTAT / COUNT 6ms Oscillator Verification	0xC00000D1	0x60C11870
34			...	...
35			0xC00000D1	0x60C05494
36	11.0	Read Traceability Information	0x8014C0D5	0x40E0C0C0
37			0xC0C00042	0x602603E0
38			0xC0C200BF	0x600002AD
39			0xC0C40097	0x600322CA
40			0xC0C6006A	0x6045CA57
41			0xC0C800C7	0x60004B33
42			0xC0CA003A	0x601600FE
43			0xC0CC0012	0x602B0A21
44			0x8014D07B	0x40C0D024
45			0xC0D00067	0x601FCE22
46			0xC0D2009A	0x600000F3
47			0xC0D400B2	0x600000F3
48			0xC0D6004F	0x600000F3
49			0xC0D800E2	0x600000F3
50			0xC0DA001F	0x600000F3
51			0xC0DD00DE	0x600000F3
52	12.0	Read Stored Self-Test Values	0x8014A040	0x40D0A03A
53			0xC0A40049	0x60054358
54			0xC0A80019	0x60051DA5
55			0x8014B0EE	0x40A0B06F
56			0xC0B4006C	0x60052F26
57			0xC0B8003C	0x6005062D
58	13.0	Read Verify	0xC01600DC	0x600000F3
59			0xC0180071	0x600000F3
60			0xC01A008C	0x608180FE
61			0xC01C00A4	0x6083825D
62			0xC01E0059	0x600000F3
63			0xC020009B	0x600000F3
64			0xC0220066	0x600000F3
65			0xC0400045	0x60A7007F
66			0xC04200B8	0x60122058
67			0xC0440090	0x600000F3
68			0xC046006D	0x601111BE
69			0xC04800C0	0x60FF033A
70			0xC04A003D	0x6002207D
71			0xC04C0015	0x600000F3
72			0xC04E00E8	0x60A5A581
73	43.0	Read Offset for ~ 30ms	0xC00000D1	0x6000073E
74			0x1000000D	0x60C1392C
75			0x300000E2	0x80000020
76			0x500000FC	0x93FF8089
77			0x70000013	0xA00E0062
78			0xC06200F2	0xB00040C9
79			0xC06400DA	0x600001DC
80			0xC07200D7	0x60FFFE26
81			0xC07400FF	0x600071E7

]



Line #	End Time (ms)	Description	Command	Response
			HEX	HEX
82	44.0	Fixed Pattern Self Test	0x80445029	0x40005054
83			0x804C6071	0x480060F9
84			0x1000000D	0x8AAA809A
85			0x300000E2	0x9AAA807A
86			0x500000FC	0xA95540DA
87			0x70000013	0xB955403A
88			0xC06200F2	0x68AAAA2F
89			0xC06400DA	0x68AAAA2F
90			0xC07200D7	0x685555D5
91			0xC07400FF	0x685555D5
92			0xC00000D1	0x6800955D
93			0x804460F4	0x480060F9
94			0x804C50AC	0x48005024
95			0x1000000D	0x89554035
96			0x300000E2	0x995540D5
97			0x500000FC	0xAAAA8075
98			0x70000013	0xBAAA8095
99			0xC06200F2	0x685555D5
100			0xC06400DA	0x685555D5
101			0xC07200D7	0x68AAAA2F
102			0xC07400FF	0x68AAAA2F
103	56.0	Enable Digital Self-Test	0x8044F0B9	0x4800F0B4
104		Delay 12 ms	0x804CF03C	0x4800F0B4
105	86.0	Read Sensor Data for ~30ms Note, multiple reads not necessary. Repeated Reads were used to confirm constant data	0x1000000D	0x88F7006D
106			0x300000E2	0x98F7008D
107			0x500000FC	0xA8F70082
108			0x70000013	0xB8EF4024
109			0xC06200F2	0x6807B904
110			0xC06400DA	0x6807B82B
111			0xC07200D7	0x6807B904
112			0xC07400FF	0x68077A70
113			0xC00000D1	0x6800187A
114	98.0	Enable Analog Self-Test Phase 1	0x8044A0F1	0x4800A0FC
115		Delay 12 ms	0x804CB0DA	0x4800B052
116	128.0	Read Sensor Data for ~30ms Note: SNSDATA registers, Channel Status and Sensor Data Requests all read to confirm consistency. All are not necessary	0x1000000D	0x884E408F
117			0x300000E2	0x984E0089
118			0x500000FC	0xAA004045
119			0x70000013	0xBA0040A5
120			0xC06200F2	0x680274F9
121			0xC06400DA	0x6802721B
122			0xC07200D7	0x68EB7381
123			0xC07400FF	0x68EAFAF3
124			0xC00000D1	0x680045F6
125	133.0	Enable Analog Self-Test Phase 2	0x8044B05F	0x4800B052
126		Delay 5 ms	0x804CA074	0x4800A0FC
127	163.0	Read Sensor Data for ~30ms Note: SNSDATA registers, Channel Status and Sensor Data Requests all read to confirm consistency. All are not necessary	0x1000000D	0x8BAD00B6
128			0x300000E2	0x9BAD0056
129			0x500000FC	0xA9FFC00C
130			0x70000013	0xB9FFC0EC
131			0xC06200F2	0x68FD6A5A
132			0xC06400DA	0x68FD6804
133			0xC07200D7	0x6815A85D
134			0xC07400FF	0x68152AE0
135			0xC00000D1	0x68005629
136	263.0	Reset Offset Cancellation Startup	0x804390ED	0x40902213
137		Delay 100 ms	0x804B80C6	0x40802236
138	283.0	Read Sensor Data for ~20ms Note: SNSDATA registers, Channel Status and Sensor Data Requests all read to confirm consistency. All are not necessary	0x1000000D	0x80000020
139			0x300000E2	0x93FFC06F
140			0x500000FC	0xA00F8068
141			0x70000013	0xB00040C9
142			0xC06200F2	0x600002AD
143			0xC06400DA	0x600000F3
144			0xC07200D7	0x6000824E
145			0xC07400FF	0x60000382
146			0xC00000D1	0x6000AFF9
147	284.0	Set ENDINIT	0x801080E6	0x400080FF
148		Normal Mode Loop Note: SNSDATA registers, Channel Status and Sensor Data Requests all read to confirm consistency. All are not necessary	0x1000000D	0x840040FE
149			0x300000E2	0x940000F8
150			0x500000FC	0xA4108031
151			0x70000013	0xB40080F4
152			0xC06200F2	0x64000295
153			0xC06400DA	0x640000CB
154			0xC07200D7	0x64008276
155			0xC07400FF	0x640001E4
156			0xC00000D1	0x64000558

13.5 CRC calculation examples

13.5.1 8-bit CRC

Figure 41 shows some example visual basic to calculate the SPI 8-bit CRC.

- Function SPICRC8(Data32 As String, Poly As String, SEED As String) As String
  - Data32 is the 24-bit message in binary to be verified with 8 zeroes appended in place of the CRC

Example: Command = 0x804013xx: Data32 = 1000 0000 0100 0000 0001 0011 0000 0000

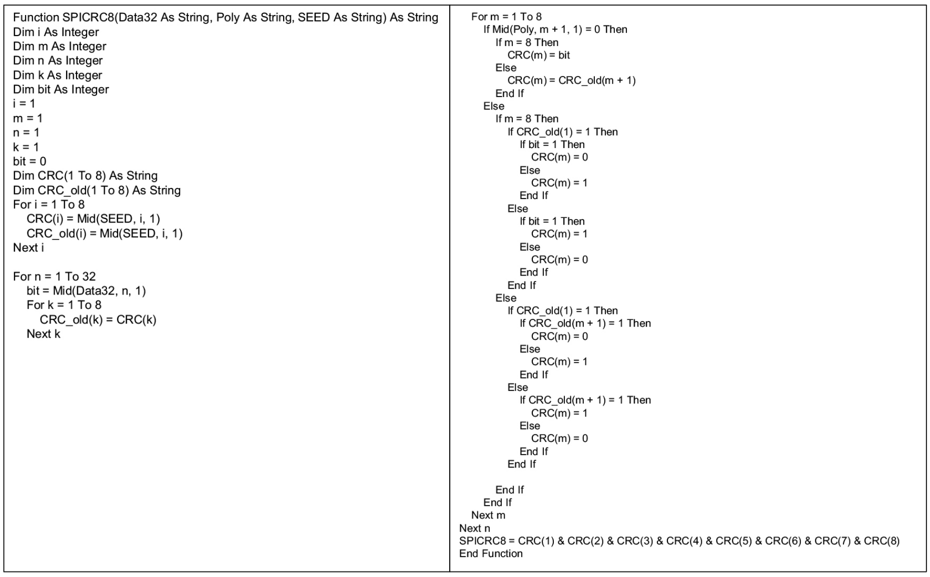
- Poly is the 9-bit CRC polynomial in binary

Example: Polynomial = $X^8+X^5+X^3+X^2+X+1$ Poly = 1 0010 1111

- SEED is the 8-bit CRC Initial value in binary

Example: Seed = 0xFF SEED = 1111 1111

In this example, the CRC = 0x6B



```
Function SPICRC8(Data32 As String, Poly As String, SEED As String) As String
Dim i As Integer
Dim m As Integer
Dim n As Integer
Dim k As Integer
Dim bit As Integer
i = 1
m = 1
n = 1
k = 1
bit = 0
Dim CRC(1 To 8) As String
Dim CRC_old(1 To 8) As String
For i = 1 To 8
    CRC(i) = Mid(SEED, i, 1)
    CRC_old(i) = Mid(SEED, i, 1)
Next i

For n = 1 To 32
    bit = Mid(Data32, n, 1)
    For k = 1 To 8
        CRC_old(k) = CRC(k)
    Next k
    For m = 1 To 8
        If Mid(Poly, m + 1, 1) = 0 Then
            If m = 8 Then
                CRC(m) = bit
            Else
                CRC(m) = CRC_old(m + 1)
            End If
        Else
            If m = 8 Then
                If CRC_old(1) = 1 Then
                    If bit = 1 Then
                        CRC(m) = 0
                    Else
                        CRC(m) = 1
                    End If
                Else
                    If bit = 1 Then
                        CRC(m) = 1
                    Else
                        CRC(m) = 0
                    End If
                End If
            Else
                If CRC_old(1) = 1 Then
                    If CRC_old(m + 1) = 1 Then
                        CRC(m) = 0
                    Else
                        CRC(m) = 1
                    End If
                Else
                    If CRC_old(m + 1) = 1 Then
                        CRC(m) = 1
                    Else
                        CRC(m) = 0
                    End If
                End If

            End If
        End If
    Next m
Next n
SPICRC8 = CRC(1) & CRC(2) & CRC(3) & CRC(4) & CRC(5) & CRC(6) & CRC(7) & CRC(8)
End Function
```

aaa-040183

Figure 41. SPI, 8-bit CRC Visual Basic

13.5.2 4-bit CRC

Figure 42 shows some example visual basic to calculate the SPI 4-bit CRC.

- Function SPICRC4(Data32 As String, Poly As String, SEED As String) As String
  – Data32 is the 24-bit message in binary to be verified with 8 zeroes appended in place of the CRC

Example: Command = 0x8040130x: Data32 = 1000 0000 0100 0000 0001 0011 0000 0000

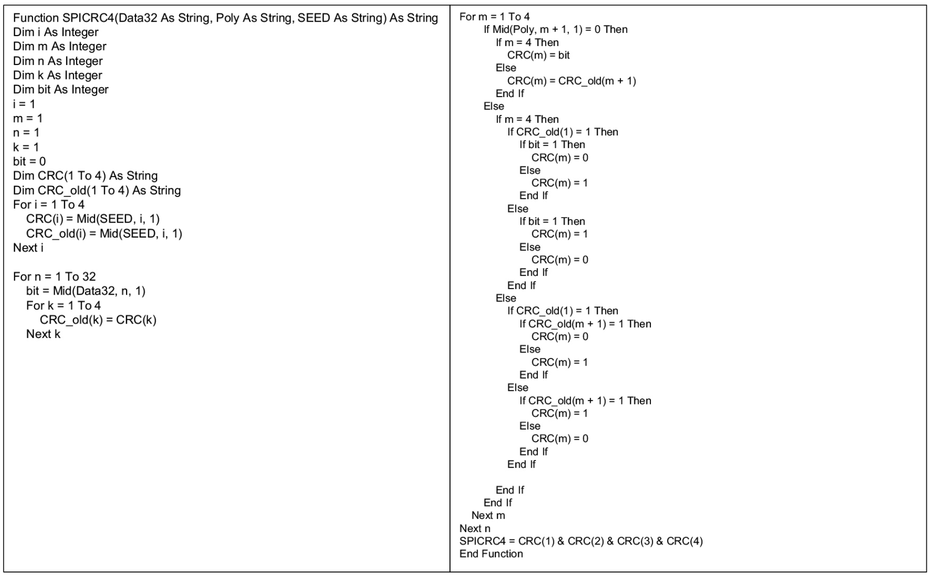
- Poly is the 9-bit CRC polynomial in binary

Example: Polynomial = X^4+1 Poly = 1 0001

- SEED is the 8-bit CRC Initial value in binary

Example: Seed = 0xA SEED = 1010

In this example, the CRC = 0x4



```
Function SPICRC4(Data32 As String, Poly As String, SEED As String) As String
Dim i As Integer
Dim m As Integer
Dim n As Integer
Dim k As Integer
Dim bit As Integer
i = 1
m = 1
n = 1
k = 1
bit = 0
Dim CRC(1 To 4) As String
Dim CRC_old(1 To 4) As String
For i = 1 To 4
   CRC(i) = Mid(SEED, i, 1)
   CRC_old(i) = Mid(SEED, i, 1)
Next i

For n = 1 To 32
   bit = Mid(Data32, n, 1)
   For k = 1 To 4
      CRC_old(k) = CRC(k)
   Next k
```

```
For m = 1 To 4
      If Mid(Poly, m + 1, 1) = 0 Then
         If m = 4 Then
            CRC(m) = bit
         Else
            CRC(m) = CRC_old(m + 1)
         End If
      Else
         If m = 4 Then
            If CRC_old(1) = 1 Then
               If bit = 1 Then
                  CRC(m) = 0
               Else
                  CRC(m) = 1
               End If
            Else
               If bit = 1 Then
                  CRC(m) = 1
               Else
                  CRC(m) = 0
               End If
            End If
         Else
            If CRC_old(1) = 1 Then
               If CRC_old(m + 1) = 1 Then
                  CRC(m) = 0
               Else
                  CRC(m) = 1
               End If
            Else
               If CRC_old(m + 1) = 1 Then
                  CRC(m) = 1
               Else
                  CRC(m) = 0
               End If
            End If

         End If
      End If
   Next m
Next n
SPICRC4 = CRC(1) & CRC(2) & CRC(3) & CRC(4)
End Function
```

aaa-040184

Figure 42. SPI, 4-bit CRC Visual Basic

13.5.3 3-bit CRC

Figure 43 shows some example visual basic to calculate the SPI 3-bit CRC.

- Function SPICRC3(Data32 As String, Poly As String, SEED As String) As String
  - Data32 is the 29-bit message in binary to be verified with 3 zeroes appended in place of the CRC

Example: Command = 0x8040270x: Data32 = 1000 0000 0100 0000 0010 0111 0000 0000

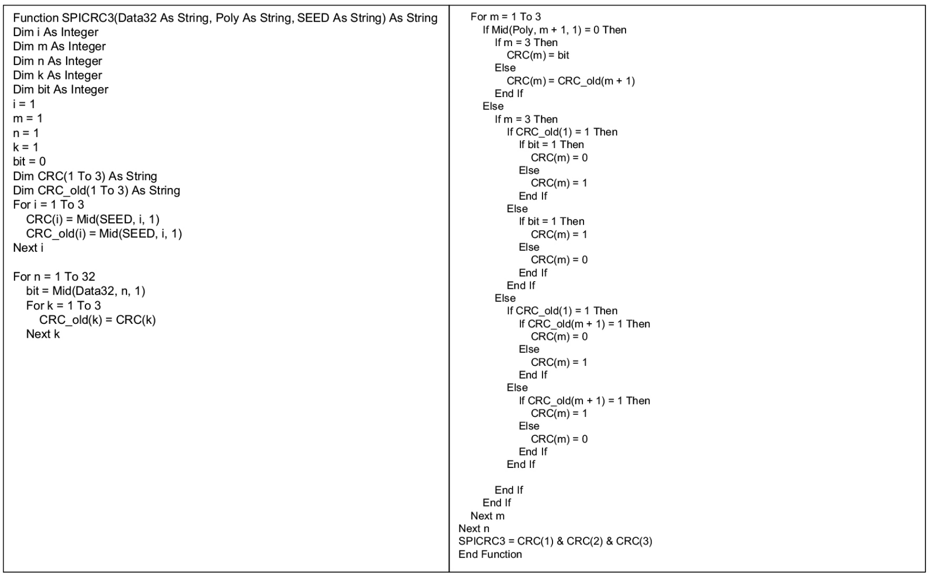
- Poly is the 9-bit CRC polynomial in binary

Example: Polynomial = X^3+X+1 Poly = 1011

- SEED is the 8-bit CRC Initial value in binary

Example: Seed = 0x7 SEED = 111

In this example, the CRC = 0x5



```
Function SPICRC3(Data32 As String, Poly As String, SEED As String) As String
Dim i As Integer
Dim m As Integer
Dim n As Integer
Dim k As Integer
Dim bit As Integer
i = 1
m = 1
n = 1
k = 1
bit = 0
Dim CRC(1 To 3) As String
Dim CRC_old(1 To 3) As String
For i = 1 To 3
   CRC(i) = Mid(SEED, i, 1)
   CRC_old(i) = Mid(SEED, i, 1)
Next i

For n = 1 To 32
   bit = Mid(Data32, n, 1)
   For k = 1 To 3
      CRC_old(k) = CRC(k)
   Next k
   For m = 1 To 3
      If Mid(Poly, m + 1, 1) = 0 Then
         If m = 3 Then
            CRC(m) = bit
         Else
            CRC(m) = CRC_old(m + 1)
         End If
      Else
         If m = 3 Then
            If CRC_old(1) = 1 Then
               If bit = 1 Then
                  CRC(m) = 0
               Else
                  CRC(m) = 1
               End If
            Else
               If bit = 1 Then
                  CRC(m) = 1
               Else
                  CRC(m) = 0
               End If
            End If
         Else
            If CRC_old(1) = 1 Then
               If CRC_old(m + 1) = 1 Then
                  CRC(m) = 0
               Else
                  CRC(m) = 1
               End If
            Else
               If CRC_old(m + 1) = 1 Then
                  CRC(m) = 1
               Else
                  CRC(m) = 0
               End If
            End If

         End If
      End If
   Next m
Next n
SPICRC3 = CRC(1) & CRC(2) & CRC(3)
End Function
```

aaa-040185

Figure 43. SPI, 3-Bit CRC Visual Basic

14 Legal information

14.1 Definitions

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

14.2 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.

Suitability for use — NXP Semiconductors products are 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.

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.

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

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

Security — While NXP Semiconductors has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP Semiconductors accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

14.3 Trademarks

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

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

Date of release: 8 January 2021
Document identifier: AN12386