KP497

User manual

Scope and purpose

This is the user manual of the product KP497. It contains detailed instructions on how to configure the device and how to use it in an application.

For the device's specification, see the corresponding datasheet.

Chapter 1 describes how to configure the serial interface after start up of the device, if it was not locked yet.

In Chapter 2, an overview of the protocol for communication with the device via the serial interface is provided, before the available commands for KP497 are described in details in Chapter 3. The description of the commands in Chapter 3 makes use of the structure and nomenclature introduced in Chapter 2.

Intended audience

This user manual is written for experienced hardware and software engineers involved in the implementation of the device into a system. It is the responsibility of system integrator to ensure that the product is suitable for the chosen application and that the procedures described within this user manual are correctly followed.

Related devices

Product type	Description and features	Package	Marking
KP497	Smart absolute pressure sensor with highly flexible autonomous state and low current consumption	PG-DSOSP-14-84	KP497

Table of contents

Table of contents . . . 2
1 Serial interface selection . . . 4
2 Protocol overview . . . 6
2.1 I^2C protocol . . . 7
2.2 SPI protocol . . . 7
2.3 Serial bus diagnosis checks . . . 8
2.4 Communication flow . . . 9
3 Description of serial interface commands . . . 10
3.1 Commands to perform on-demand measurements . . . 11
3.1.1 Measure Pressure . . . 11
3.1.1.1 Measure Pressure parameters . . . 11
3.1.1.2 Messure Pressure results . . . 11
3.1.2 Measure Acceleration . . . 11
3.1.2.1 Measure Acceleration parameters . . . 11
3.1.2.2 Measure Acceleration results . . . 12
3.1.3 Measure Temperature . . . 12
3.1.3.1 Measure Temperature parameters . . . 12
3.1.3.2 Measure Temperature results . . . 13
3.2 System commands . . . 14
3.2.1 Read ID . . . 14
3.2.1.1 Read ID parameters . . . 14
3.2.1.2 Read ID results . . . 14
3.2.2 Self Test . . . 14
3.2.2.1 Self Test parameters . . . 15
3.2.2.2 Self Test results . . . 15
3.2.3 Trigger Softreset . . . 15
3.2.3.1 Trigger Softreset parameters . . . 15
3.2.4 Read Device Status Register . . . 15
3.2.4.1 Read Device Status Register parameters . . . 16
3.2.4.2 Read Device Status Register results . . . 17
3.2.5 Set Serial Interface . . . 20
3.2.5.1 Set Serial Interface parameters . . . 20
3.2.5.2 Set Serial Interface results . . . 21
3.3 State transitions . . . 22
3.3.1 Enter Autonomous State . . . 22
3.3.1.1 Enter Autonomous State parameters . . . 22
3.3.2 Enter Sleep . . . 22
3.3.2.1 Enter Sleep parameters . . . 22
3.4 Autonomous State . . . 23

Table of contents

3.4.1 Read Autonomous State History . . . 23
3.4.1.1 Read Autonomous State History parameters . . . 23
3.4.1.2 Read Autonomous State History results . . . 23
3.4.2 Read Autonomous State Pressure Buffer . . . 24
3.4.2.1 Read Autonomous State Pressure Buffer parameters . . . 24
3.4.2.2 Read Autonomous State Pressure Buffer results . . . 24
3.4.3 Read Autonomous State Acceleration Buffer . . . 25
3.4.3.1 Read Autonomous State Acceleration Buffer parameters . . . 25
3.4.3.2 Read Autonomous State Acceleration Buffer results . . . 25
3.4.4 Read Acceleration Recording Statistics . . . 26
3.4.4.1 Read Acceleration Recording Statistics parameters . . . 26
3.4.4.2 Read Acceleration Recording Statistics results . . . 26
3.4.4.2.1 Definition of bins . . . 27
3.4.5 Save Autonomous State Configuration . . . 28
3.4.5.1 Save Autonomous State Configuration parameters . . . 28
3.4.5.2 Save Autonomous State Configuration results . . . 33
3.4.6 Read Autonomous State Configuration . . . 34
3.4.6.1 Read Autonomous State Configuration parameters . . . 34
3.4.6.2 Read Autonomous State Configuration results . . . 34
3.5 Data storage . . . 36
3.5.1 Write Data Line . . . 36
3.5.1.1 Write Data Line parameters . . . 36
3.5.1.2 Write Data Line results . . . 36
3.5.2 Read Data Line . . . 37
3.5.2.1 Read Data Line parameters . . . 37
3.5.2.2 Read Data Line results . . . 38
3.6 Processing times for serial interface commands . . . 39
4 Examples for Autonomous State Configurations . . . 40
4.1 Example 1: Single stage pressure monitoring . . . 40
4.2 Example 2: Three stage pressure and acceleration monitoring . . . 43
4.2.1 Autonomous State Configuration for first stage (0x01) . . . 45
4.2.2 Autonomous State Configuration for second stage pressure monitoring (0x02) . . . 47
4.2.3 Autonomous State Configuration for third stage pressure monitoring (0x03) . . . 50
4.2.4 Autonomous State Configuration for second stage acceleration monitoring (0x08) . . . 52
4.3 Example 3: Cyclic pressure monitoring with intermediate alerts . . . 54
4.3.1 Autonomous State Configuration for first stage of cyclic pressure monitoring (0x00) . . . 54
4.3.2 Autonomous State Configuration for second stage of cyclic pressure monitoring (0x05) . . . 56
5 Revision history . . . 59
Disclaimer . . . 60

1 Serial interface selection

Before the first successful execution of the Set Serial Interface command (successful execution means that the status byte returned after executing the command is 0), the serial interface can be selected via pins SCL/CLK, SDA/DATA and NCS/WAKEUP/BUSY after Power On reset or after a Softreset triggered by the external master. For selecting the serial interface, the external master first applies the respective levels on the SCL/CLK and SDA/DATA pins and then pulls NCS/WAKEUP/BUSY low. The KP497 then selects the serial interface according to the levels applied to SCL/CLK as follows:

I^2C	High level at both SCL/CLK and SDA/DATA
SPI	Low level at either SCL/CLK or SDA/DATA or both

The following flow diagram in Figure 1 describes the serial interface selection procedure in more details.

The internal pull-up resistors of pins SCL/CLK, SDA/DATA and NCS/WAKEUP/BUSY are active during this procedure. Therefore, as long as the external master does not pull SCL/CLK and/or SDA/DATA actively low, I^2C is automatically selected when the external master pulls NCS/WAKEUP/BUSY low. In order to select SPI, the external master pulls for example SCL/CLK low before pulling NCS/WAKEUP/BUSY low. This is in contrast to SPI mode 3 used by KP497 where the SCL/CLK must be high in its idle state.

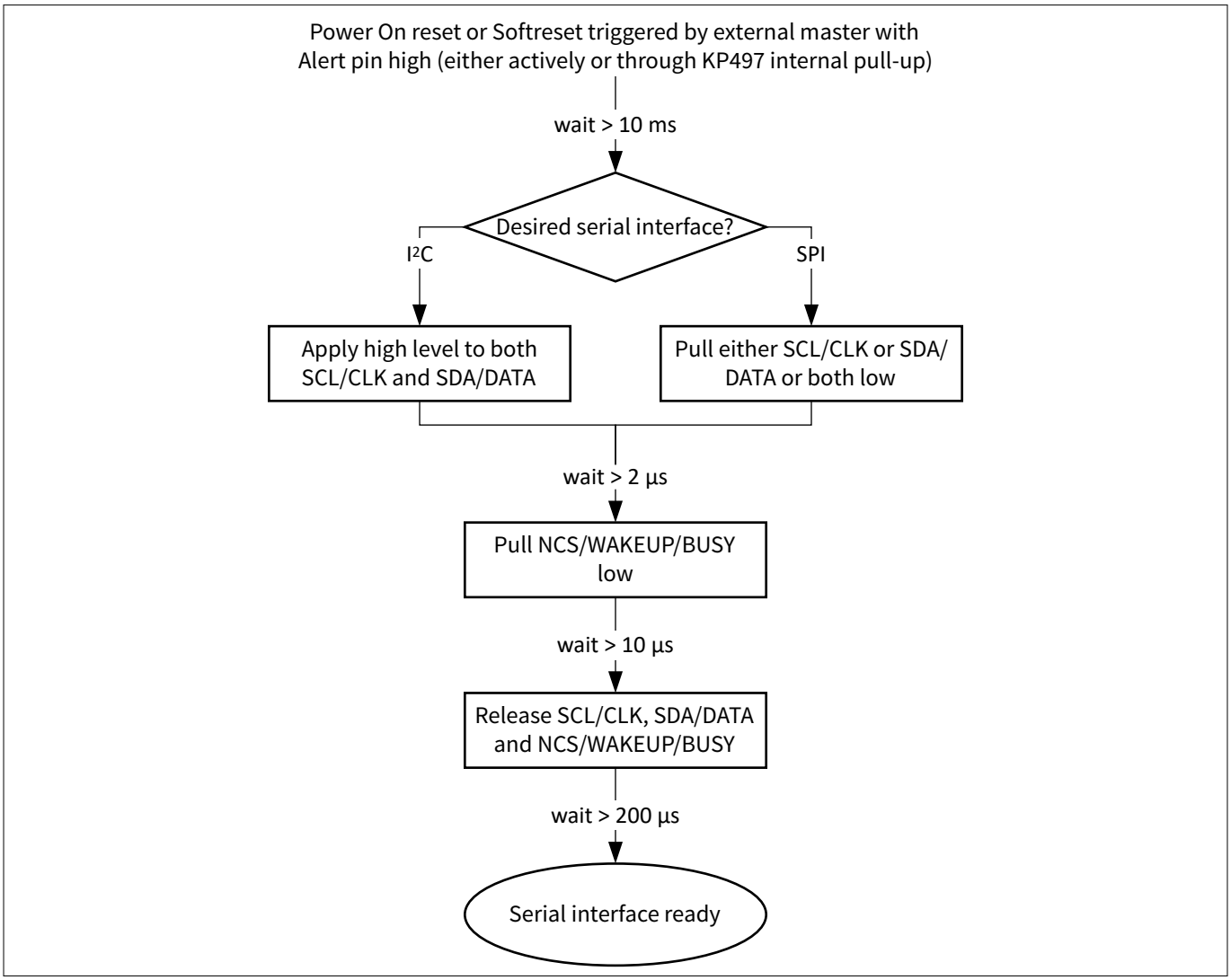


Figure 1 Flow diagram for selection of serial interface

The following figure shows how the flow diagram in Figure 1 translates into the levels applied to the pins SCL/CLK, SDA/DATA and NCS/WAKEUP/BUSY in order to select I^2C after a Power On reset. For selecting I^2C, the external master ensures that the pins SCL/CLK and SDA/DATA remain at a high level while pin NCS/WAKEUP/

BUSY is pulled low for at least 10 µs. The pin NCS/WAKEUP/BUSY must be low 10 ms after the reset release threshold is exceeded on VDD or later. After the NCS/WAKEUP/BUSY pin is at a high level again, it takes up to 200 µs until the I²C interface of KP497 is ready to receive the first serial interface command.

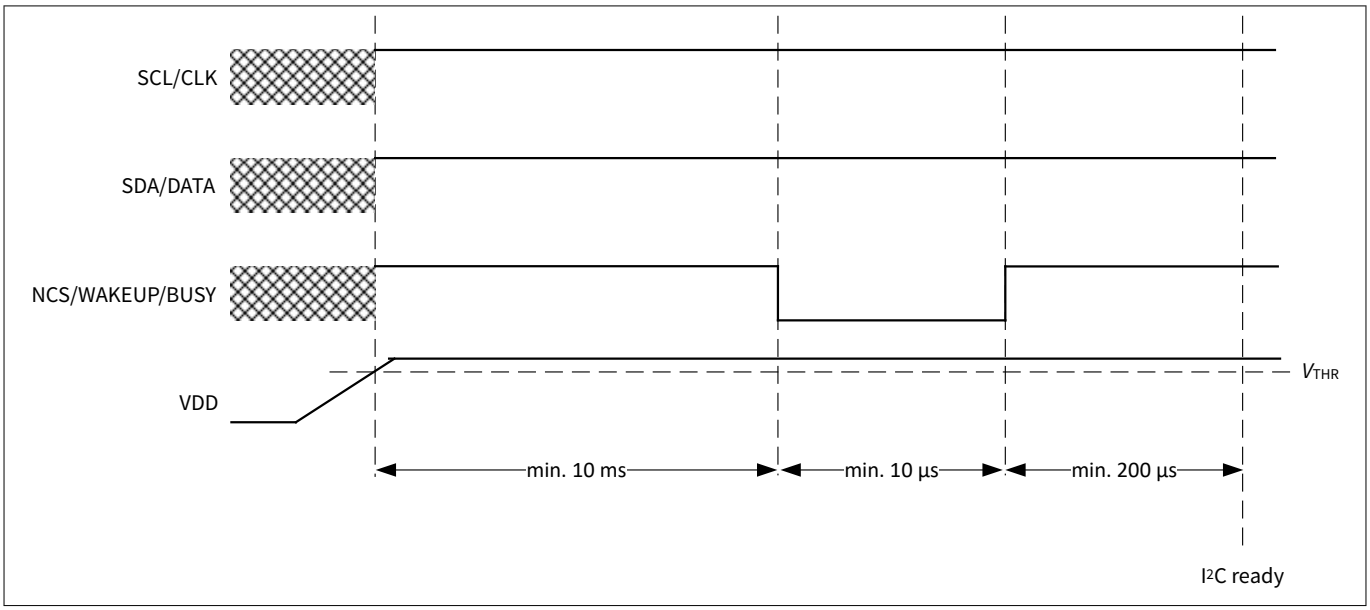


Figure 2 Timing diagram for selection of I²C after reset

The following figure shows how the flow diagram in Figure 1 translates into the levels applied to the pins SCL/CLK, SDA/DATA and NCS/WAKEUP/BUSY in order to select SPI after a Power On reset. For selecting SPI, the external master pulls the pins SCL/CLK or SDA/DATA or both low earliest 10 ms after the reset release threshold is exceeded on the VDD pin. In the figure below, both SCL/CLK and SDA/DATA are pulled low, which is the recommended scenario. After SCL/CLK and SDA/DATA are pulled low, it is recommended to wait at least 2 µs before pin NCS/WAKEUP/BUSY is pulled low as well. The NCS/WAKEUP/BUSY pin must then remain at a low level for at least 10 µs, before a high level is applied to the NCS/WAKEUP/BUSY pin again. After the NCS/WAKEUP/BUSY pin is low for 10 µs, the levels applied to SCL/CLK and SDA/DATA do not matter anymore until a SPI transaction is started by a high-to-low transition on the NCS/WAKEUP/BUSY pin. The SPI interface of KP497 is ready latest 200 µs after a high level is applied again to the NCS/WAKEUP/BUSY pin.

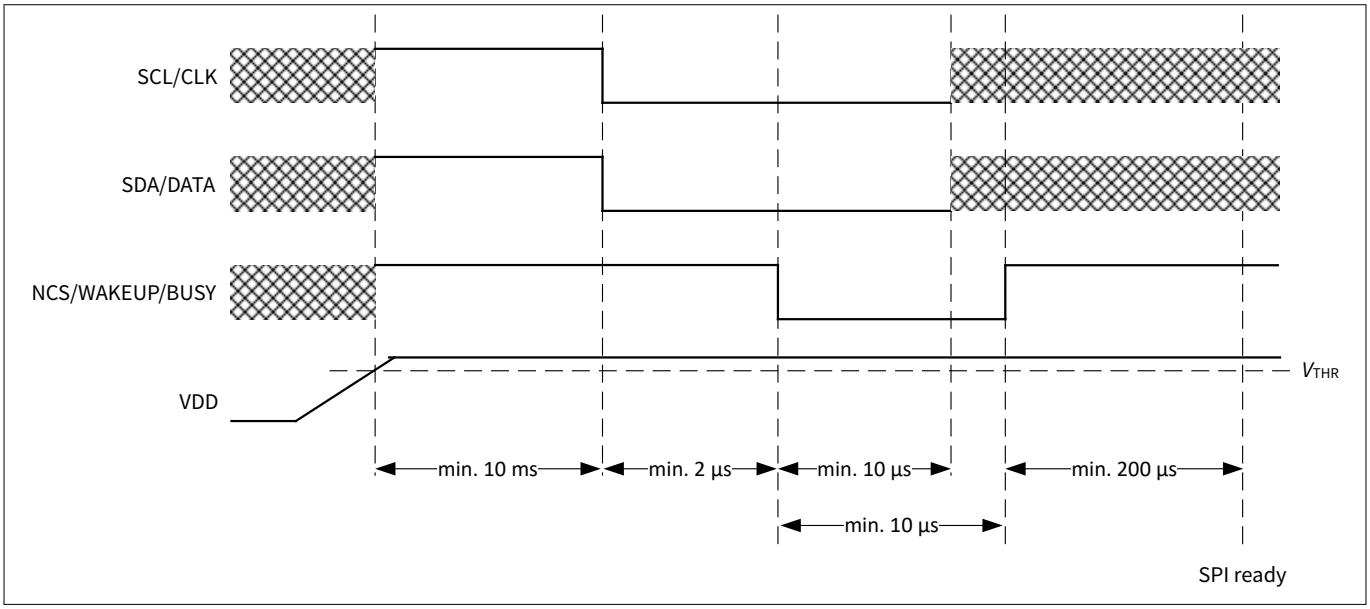


Figure 3 Timing diagram for selection of SPI after reset

It is highly recommended to use the Set Serial Interface command in order to lock the serial interface selection rather than performing this serial interface selection at every Power On reset or Softreset.

2 Protocol overview

The device is controlled through serial interface commands. Unless documented otherwise, every command consist of three steps:

1. **The external master transmits** data consisting of:
   - **a.** One command byte (labeled “(1.a)” in Figure 4 and Figure 5): This byte indicates the request from the master to KP497 (e.g. Enter Autonomous State).
   - **b.** Parameters (labeled “(1.b)” in Figure 4 and Figure 5): These are additional parameters for the request (e.g. selection of configuration for Enter Autonomous State). Not all commands require parameters. The number of bytes used as parameters depends on the command but is otherwise fixed. For each command, the number of bytes required as parameters is documented as part of their description in Chapter 3.
   - **c.** CRC16 over the Command byte and the Parameters (labeled “(1.c)” in Figure 4 and Figure 5).
2. **KP497 processes** the command (Labeled “(2)” in Figure 4 and Figure 5): The device collects all necessary data to be provided to the external master. The time t_{proc} it takes KP497 to process the command depends on the command. During this time, no communication with KP497 is possible and KP497 pulls NCS/WAKEUP/BUSY actively low. For highest robustness and minimum wait time, it is recommended that the external master monitors the NCS/WAKEUP/BUSY pin and waits until it is released by KP497 before continuing with step 3 or before sending a new command.
3. **The external master reads** the result from KP497:
   - In case of a I^2C or SPI bus fault or if the command is not valid, KP497 transmits one byte error code followed by the CRC16 of the error code.
   - Otherwise, KP497 first transmits the command byte (Labeled “(3.a)” in Figure 4 and Figure 5) of the command for which the results are provided and the results of the command itself (Labeled “(3.b)” in Figure 4 and Figure 5) followed by the CRC16 (Labeled “(3.c)" in Figure 4 and Figure 5) calculated over the command byte and the command results. The number of bytes of the command results depends on the command but is otherwise fixed. For each command, the number of bytes returned by KP497 is documented as part of their description in Chapter 3.

   For SPI, the external master read access is a single SPI interaction with the NCS/WAKEUP/BUSY pin pulled low by the external master where the external master transmits the value 0x6C during the first 8 clock cycles. When KP497 recieves the value 0x6C as first byte, the device immedately starts to transmits the data in its output buffer from the 9th clock cycle onwards.

The CRC16 is calculated using the CCITT polynomial 0x1021 with a preload value of 0xFFFF. For the calculation of the CRC, the transmitted bytes are used in the same order in which they are transmitted. The CRC16 value is provided with the high byte first.

***Note**: The following commands do not provide any results on the serial interface when successful and consist of the master write followed by either a reset or a state transition:*

- Trigger Softreset
- Enter Autonomous State
- Enter Sleep

The success of these commands can instead be checked by the absence of a response of KP497 on the serial interface for Enter Autonomous State, and Enter Sleep and by performing the Read Device Status Register command for Trigger Softreset.

2.1 I²C protocol

Figure 4 illustrates the flow of each command when using I²C.

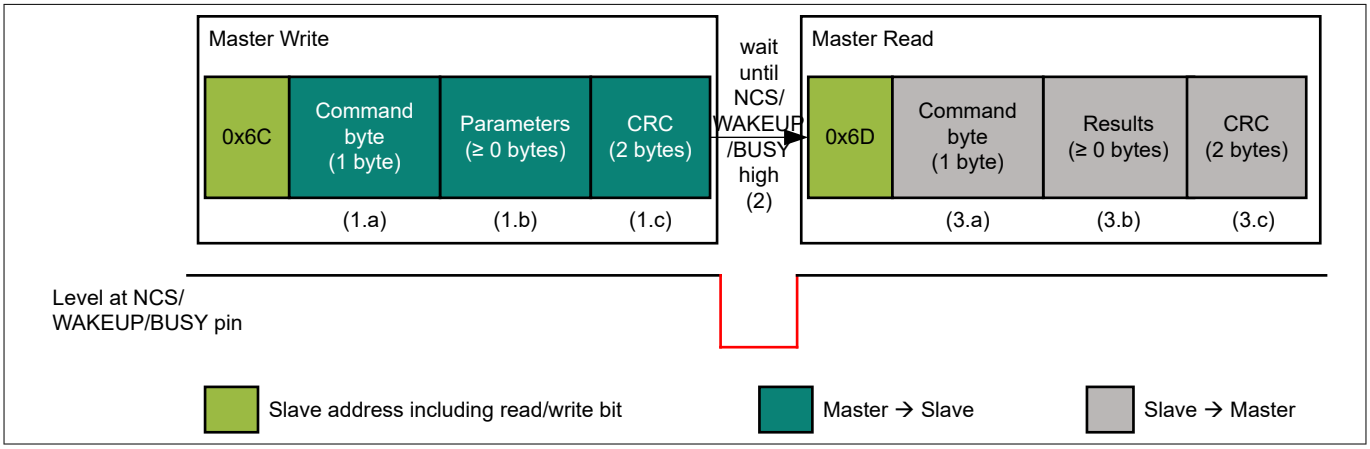


Figure 4 Command flow when I²C is selected

Figure 4 also indicates the level of the NCS/WAKEUP/BUSY pin: Once the Master Write is finished KP497 pulls the NCS/WAKEUP/BUSY pin actively low as indicated by the red line until KP497 finishes processing. After KP497 finishes processing, the NCS/WAKEUP/BUSY pin is released again and is configured as input with internal pull-up resistor. The black line for the NCS/WAKEUP/BUSY pin in Figure 4 indicates that the NCS/WAKEUP/BUSY pin is an input pin with internal pull-up resistor.

2.2 SPI protocol

Figure 5 illustrates the flow of each command when using SPI.

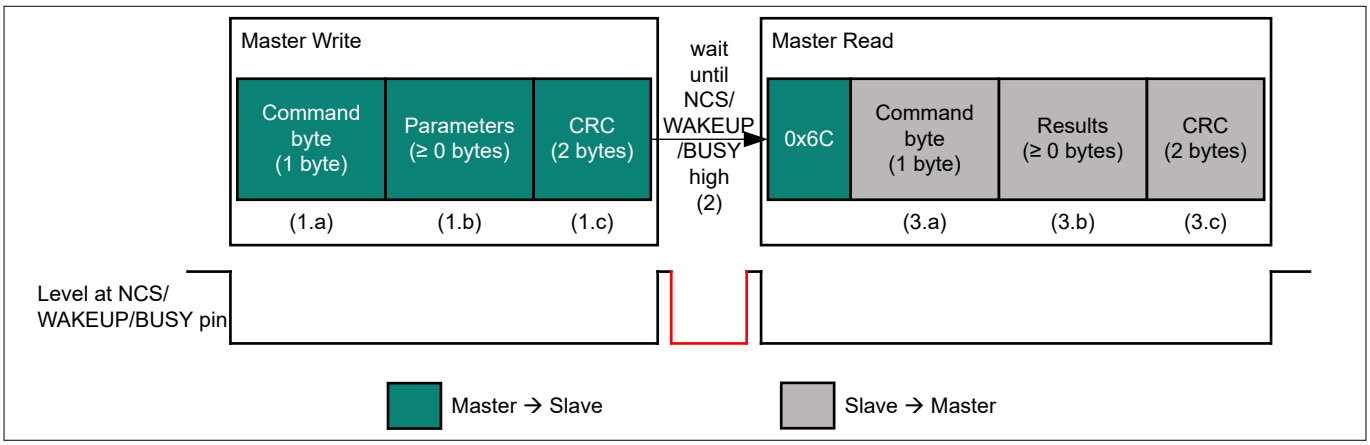


Figure 5 Command flow when SPI is selected

As indicated in Figure 5 sending the command (i.e. the Master Write) and reading out the results (i.e. the Master Read) are two separate SPI interactions. That means that once the master finishes sending all data to KP497, it releases the chip select (NCS/WAKEUP/BUSY pin) or pulls it actively high. Once the Master Write is finished, KP497 pulls the NCS/WAKEUP/BUSY pin actively low as indicated by the red line in Figure 5 until processing of the command is finished. When the SPI interface of KP497 is ready again after processing the command, KP497 releases the NCS/WAKEUP/BUSY again and the pin is an input with internal pull-up resistor as indicated by the black line. The external master can then start the Master Read by pulling the NCS/WAKEUP/BUSY low.

***Note**: Since the chip select and external wake-up share the same pin of KP497, the device can be woken up from Sleep or Autonomous State with a SPI transaction. However, if the external master wakes up the KP497 with an SPI transaction, KP497 will neither receive any data nor provide any data during this transaction.*

2.3 Serial bus diagnosis checks

The command byte provided by the external master in Figure 4 and Figure 5 is echoed by KP497 in the first byte read by the external master. This provides an additional check that the command was received by KP497 correctly.

As part of the processing of the command, KP497 performs a validity check of the received data. If the command is not valid or a bus fault was detected, KP497 does not echo the command byte and provides an error code followed by the 16-bit CRC instead: The first byte of the Master Read is the error code described in Table 1. The second and third byte are the 16-bit CRC of the error code with the high byte of the CRC transmitted in the second byte. If the master reads more than 3 bytes, all remaining bytes are returned by KP497 as 0xE0.

Same as the commands when received correctly, the error codes also require a short processing time and are only provided during the Master Read. The processing time for when the validity check of the received data fails is always shorter than the processing time of the correctly received command.

Table 1 KP497 error codes

Error Code	Description of fault
0xE0	KP497 output buffer empty: No command was received by KP497 since entering to On State. If more bytes are read by the external master than are available, KP497 fills all remaining bytes with this error code as well.
0xE7	KP497 received an invalid command: • Command byte does not match any command or • Number of bytes provided by the external master does not match number of bytes required for the command
0xEC	CRC check of data received by KP497 on serial interface failed.
0xED	Collision on serial interface detected: • For I^2C, a collision is caused by a change of the level on SDA/DATA where it is not permitted by the I^2C specification. • For SPI, a collision is caused if the the external master sends an incomplete byte or when the level at the NCS/WAKEUP/BUSY pin changes while SCL/CLK is low.
0xEF	Overflow of KP497 input buffer. This error typically occurs when the clock speed is too high or when the external master sends too many bytes.

2.4 Communication flow

Figure 6 shows a potential flow for sending a command and evaluating the success of the communication as well as checking the validity of the data received by the external master. In this flow, *CMD* is the command byte sent by the external master, *X* is the number of bytes required for the command and *Y* is the number of bytes of the results, not counting the command byte itself. Moreover *CRCH* and *CRCL* are the high (*CRCH*) and low (*CRCL*) byte of the CRC calculated over the *CMD* and the *X* bytes after *CMD*. Note that the master can also read more than *Y*+3 bytes during the read access, but for checking the CRC and evaluating the results only the first *Y*+3 bytes must be taken into account, while all further bytes are set to 0xE0 by KP497.

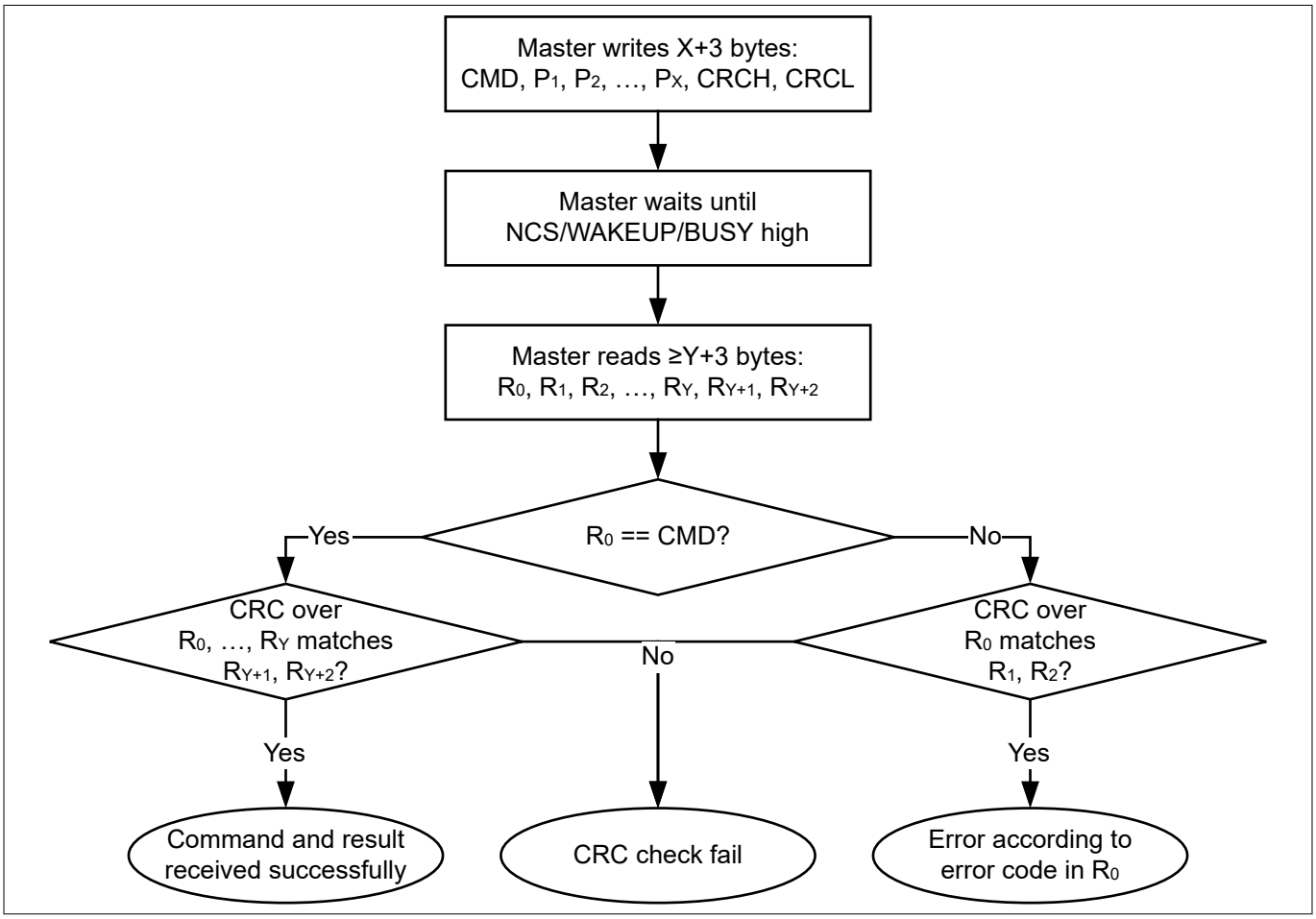


Figure 6 Potential flow for implementation of commands in the external master

3 Description of serial interface commands

The following sections describe the available commands of KP497. These descriptions refer to the protocol as well as the definitions of the command byte, parameters and command results provided in the Protocol overview.

The description of each command starts with the specification of the command byte and a description of the purpose of the command. After that the parameters and results of the command are described in detail, if applicable.

If a command requires parameters to be sent by the master, the parameters are described by a table of the following format:

***N* parameters (*X* bytes)**			
Parameter 1 (x_1 bytes)	*Parameter 2* (x_2 bytes)	…	*Parameter N* (x_N bytes)

In this table, *N* is the number of parameters required by the command and *X* the sum of the number of bytes for all parameters. Both *N* and *X* vary from command to command. The total number of bytes required to be transmitted by the external master for a command not counting the slave address is *X*+3.

Parameter 1 through *Parameter N* define a name for each of the *N* parameters that are used in the description of the command to provide more details. The number of bytes for each parameter *i* is provided by the value of $x_i \geq 1$. The sum of all x_i is always $x_1 + \ldots + x_N = X$.

Parameters that have more than 1 byte must always be transmitted from highest to lowest byte, i.e. the highest byte is transmitted first and the lowest byte last.

Similarly, the command results provided by KP497 is described in the following sections by a table of the following format:

***M* command results (*Y* bytes)**			
Result 1 (y_1 bytes)	*Result 2* (y_2 bytes)	…	*Result M* (y_M bytes)

Again, the number of results of the command, *M*, and the sum of the number of bytes for all results, *Y* vary from command to command. The total number of bytes transmitted by KP497 is *Y*+3.

Result 1 through *Result M* define a name for each of the *M* results that are used in the description of the command to provide more details. The number of bytes for each result *i* is provided by the value of $y_i \geq 1$. The sum of all y_i is always $y_1 + \ldots + y_M = Y$.

Results that have more than 1 byte are always transmitted from highest to lowest byte, i.e. the highest byte is transmitted first and the lowest byte last.

For simplicity, the 16-bit CRC is omitted in the description of each individual commands.

3.1 Commands to perform on-demand measurements

3.1.1 Measure Pressure

Command byte
0xA8

The command causes KP497 to perform a new pressure measurement. After successful execution of the command, the external master can read out the result of this pressure measurement on the serial interface.

3.1.1.1 Measure Pressure parameters

The command has no parameters ($N = X = 0$).

3.1.1.2 Messure Pressure results

During the master read for the Measure Pressure command, the KP497 provides the following results:

2 Command results (3 bytes)	
Status (1 byte)	*Pressure* (2 bytes)

The command was executed successful, if the value of *Status* is 0. Otherwise, *Status* indicates the fault that occurred. The result of the pressure measurement is provided by *Pressure*. The following table describes the results of the Measure Pressure command in detail:

Table 2 Results of the Measure Pressure command

Name	Bit(s)	Description
Status	0	1 if an ADC underflow occurred, 0 else
	1	1 if an ADC overflow occurred, 0 else
	2	1 if the Wire Bond Check failed, 0 else
	3	(not used)
	4	1 if a numerical underflow occurred, 0 else
	5	1 if a numerical overflow occurred, 0 else
	6	1 if the CRC check of calibration data failed, 0 else
	7	1 if a timeout occurred during ADC conversion, 0 else
Pressure	15:0	Signed (using 2's complement) 16-bit pressure value. The pressure value is transmitted with its high byte first and with a resolution of 1/16 kPa/LSB, i.e. 16 LSB correspond to 1 kPa.

3.1.2 Measure Acceleration

Command byte
0xA9

The command causes KP497 to perform a new acceleration measurement. After successful execution of the command, the external master can read out the result of this acceleration measurement on the serial interface.

3.1.2.1 Measure Acceleration parameters

The command has no parameters ($N = X = 0$).

3.1.2.2 Measure Acceleration results

During the master read for the Measure Acceleration command, the KP497 provides the listed in the table below followed by the CRC:

2 Command results (3 bytes)	
Status (1 byte)	*Acceleration* (2 bytes)

The command was executed successful, if the value of *Status* is 0. Otherwise, *Status* indicates the fault that occurred. The result of the acceleration measurement is provided by *Acceleration*. The following table describes the results of the Measure Acceleration command in detail:

Table 3 Results of the Measure Acceleration command

Name	Bit(s)	Description
Status	0	1 if an ADC underflow occurred, 0 else
	1	1 if an ADC overflow occurred, 0 else
	2	1 if the Wire Bond Check failed, 0 else
	3	1 if the Diagnosis Resistor Check failed, 0 else
	4	1 if a numerical underflow occurred, 0 else
	5	1 if a numerical overflow occurred, 0 else
	6	1 if the CRC check of calibration data failed, 0 else
	7	1 if a timeout occurred during ADC conversion, 0 else
Acceleration	15:0	Signed (using 2's complement) 16-bit acceleration value. The acceleration value is transmitted with its high byte first and with a resolution of 1/16 g/LSB, i.e. 16 LSB correspond to 1 g.

3.1.3 Measure Temperature

Command byte
0xAA

The command causes KP497 to perform a new temperature measurement. After successful execution of the command, the external master can read out the result of this temperature measurement on the serial interface.

3.1.3.1 Measure Temperature parameters

The command has no parameters ($N = X = 0$).

3.1.3.2 Measure Temperature results

During the master read for the Measure Temperature command, the KP497 provides the listed in the table below followed by the CRC:

2 Command results (3 bytes)	
Status (1 byte)	*Temperature* (2 bytes)

The command was executed successful, if the value of *Status* is 0. Otherwise, *Status* indicates the fault that occurred. The result of the temperature measurement is provided by *Temperature*. The following table describes the results of the Measure Temperature command in detail:

Table 4 Results of the Measure Temperature command

Name	Bit(s)	Description
Status	0	1 if an ADC underflow occurred, 0 else
	1	1 if an ADC overflow occurred, 0 else
	3:2	(not used)
	4	1 if a numerical underflow occurred, 0 else
	5	1 if a numerical overflow occurred, 0 else
	6	1 if the CRC check of calibration data failed, 0 else
	7	1 if a timeout occurred during ADC conversion, 0 else
Temperature	15:0	Signed (using 2's complement) 16-bit temperature value. The temperature value is transmitted with its high byte first and with a resolution of 1/128°C/LSB, i.e. 128 LSB correspond to 1°C.

3.2 System commands

3.2.1 Read ID

Command byte
0xC3

The command causes KP497 to provide its sensor ID and its product code on the serial interface.

3.2.1.1 Read ID parameters

The command has no parameters ($N = X = 0$).

3.2.1.2 Read ID results

During the master read for the Read ID command, the KP497 provides the listed in the table below followed by the CRC:

2 Command results (6 bytes)	
ID (4 bytes)	*Product code* (2 bytes)

The sensor ID of the device is provided by *ID* and its product code by *Product code*. The following table describes the results of the Read ID command in detail:

Table 5 Results of the Read ID command

Name	Bit(s)	Description
ID	31:0	Sensor ID of the device. The bytes of the sensor ID are transmitted in descending order, i.e. with its highest byte first and its lowest byte last.
Product code	15:0	Product code of the device. The product code is transmitted with its high byte first.

3.2.2 Self Test

Command byte
0xC5

The command causes KP497 to perform a self-test of the device's signal path, of the internal clock generators, and optionally of the non-volatile memories and provides the results on the serial interface.

The clock generator for the ADC and for the internal MCU is not checked directly by the command itself. However, whenever the command is executed with *Config* = 0, the external can check the clock generator for the ADC and MCU by measuring the processing time of the command, i.e. by measuring the duration for which KP497 pulls its NCS/WAKEUP/BUSY pin actively low. For *Config* = 0 and with the clock generator running correctly, the processing time t_{proc} is in the range 2.037 ms ≤ t_{proc} ≤ 2.391 ms.

***Note**: With Config ≠ 0, the processing time of the Self test command increases up to around 250 ms.*

3.2.2.1 Self Test parameters

The Self Test command has one parameter for enabling or disabling the memory check:

1 Parameter (1 byte)
Config (1 byte)

The following table describes the parameter required for the Self Test command in detail:

Table 6 Parameters for Self Test command

Name	Bit(s)	Description
Config	7:0	0 to disable the memory check, ≠0 to perform the memory check

3.2.2.2 Self Test results

During the master read for the Self Test command, the KP497 provides the listed in the table below followed by the CRC:

1 Command result (1 byte)
Status (1 byte)

The Self Test was successful, if *Status* is 0. Otherwise, the *Status* indicates the fault detected by the Self Test according to the following table:

Table 7 Results of the Self Test command

Name	Bit(s)	Description
Status	0	1 if the CRC check of the internal memories failed, 0 else.
	1	1 if the signal path check failed, 0 else.
	2	1 if the check of the internal RC oscillators failed, 0 else.
	7:3	(not used)

3.2.3 Trigger Softreset

Command byte
0xC7

The command triggers a Softreset of KP497. Therefore, KP497 does not provide and data on the serial interface after successful reception of this command.

3.2.3.1 Trigger Softreset parameters

The command has no parameters ($N = X = 0$).

3.2.4 Read Device Status Register

Command byte
0xD8

The command causes KP497 to provide the data in the Device Status Register on the serial interface.

3.2.4.1 Read Device Status Register parameters

The Read Device Status Register command has one parameter that specifies whether to clear the Device Status Register and the Alert.

1 Parameter (1 byte)
Clear (1 byte)

The following table describes the parameter required for the Read Device Status Register command in detail:

Table 8 Parameters for Read Device Status Register command

Name	Bit(s)	Description
Clear	7:0	Setting the parameter to 0xAC clears the Device Status Register and the Alert pin. Otherwise, the Device Status Register keeps its content and the Alert pin remains high. It is recommended to use the value 0x53 when the Device Status Register and the Alert pin shall not be cleared.

3.2.4.2 Read Device Status Register results

During the master read for the Device Status Register command, the KP497 provides the listed in the table below followed by the CRC:

4 Command results (7 bytes)			
Device Status (3 bytes)	*Autonomous Status* (2 bytes)	*Measurement Status* (1 byte)	*Last ID* (1 byte)

The command results of the Read Device Status Register command are described in detail in the following table:

Table 9 Results of the Read Device Status Register command

Name	Bit(s)	Description
Device Status	7:0	For functional device always 0. Otherwise provides additional failure information for Infineon internal use.
	8	(not used)
	9	1 if On State was entered through one of the state transitions documented in Figure “State Transitions” in section “Device State” of the KP497 datasheet, 0 else
	10	1 if a wake-up from Sleep was triggered by the external master, 0 else
	11	1 if a wake-up from Autonomous State was triggered by the external master, 0 else
	12	1 if On State was entered due to a corruption of the flow of device states, 0 else.
	13	1 if the CRC check of the Device Status register failed, 0 else
	15:14	(not used)
	16	1 if a Brown-out reset occurred, 0 else
	17	1 if an ECC2 reset occurred, 0 else
	18	1 if a Watchdog reset occurred, 0 else
	19	1 if a Power On reset occurred, 0 else
	20	1 if a Softreset occurred, 0 else
	21	(not used)
	22	1 if an Undervoltage reset occurred, 0 else
	23	(not used)

(table continues...)

Table 9 (continued) Results of the Read Device Status Register command

Name	Bit(s)	Description
Autonomous Status	0	1 if the pressure dropped below the lower absolute or relative threshold, 0 else
	1	1 if the pressure exceeded the upper absolute or relative threshold for pressure, 0 else
	2	1 if the pressure rate of change dropped below the lower rate of change threshold for pressure, 0 else
	3	1 if the pressure rate of change exceeded the upper rate of change threshold for pressure, 0 else
	4	1 if the acceleration dropped below the lower absolute or relative threshold, 0 else
	5	1 if the acceleration exceeded the upper absolute or relative threshold for pressure, 0 else
	6	1 if the acceleration rate of change dropped below the lower rate of change threshold for pressure, 0 else
	7	1 if the acceleration rate of change exceeded the upper rate of change threshold for pressure, 0 else
	8	1 if during Autonomous State the temperature was below the under-temperature threshold T_{UT} or above the over-temperature threshold T_{OT}, 0 else. *Note*: T_{UT} and T_{OT} are specified in section “Autonomous State Characteristics” of the KP497 datasheet.
	9	1 if another fault occurred during Autonomous State, 0 else
	10	1 if the CRC check during Autonomous State failed, 0 else
	11	1 if a time-out from Autonomous State occurred, 0 else
	12	1 if a transition to an undefined or invalid Autonomous State configuration occurred, 0 else
	13	(not used)
	14	1 if a fault occurred during Acceleration Recording, 0 else
	15	1 if Acceleration Recording finished successfully, 0 else

(table continues...)

Table 9 (continued) Results of the Read Device Status Register command

Name	Bit(s)	Description
Measurement Status	0	1 if an ADC underflow occurred during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
	1	1 if an ADC overflow occurred during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
	2	1 if a wirebond check failed during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
	3	1 if the diagnosis resistor check for acceleration measurement failed during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
	4	1 if a numerical underflow occurred during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
	5	1 if a numerical overflow occurred during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
	6	1 if the CRC check of calibration data failed during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
	7	1 if a time-out occurred during ADC conversion during a pressure or acceleration measurement in Autonomous State or during Acceleration Recording, 0 else
Last ID	4:0	The ID of the Autonomous State configuration last used
	5	1 if the ID in bits 4:0 is invalid, 0 else. The bit is set after reset and after the Device Status register was cleared until Autonomous State is entered. When Autonomous State is entered, the bit is cleared.
	7:6	(not used)

Attention**:* ***KP497 sets its ALERT pin high, whenever it enters On State, no matter whether On State was entered due to a reset, from Sleep or from Autonomous State. The ALERT pin remains high until the Device Status Register is cleared, i.e. until the Read Device Status Register command is called with Config = 0xAC. As long as the Device status register is not cleared the ALERT pin remains high even when the device enters Sleep or Autonomous State. Therefore, it is recommended to clear the Device Status Register before entering Autonomous State.

***Note**:* *The bits in the Device Status Register are cumulative: If a bit of the Device Status Register is set due to an event, all bits that are already set when the event occurs remain set as well. For example, if bit 9 (alert generated by entering On State) of the Device Status is set when an invalid state transition occurs, afterwards both bits 9 and 12 are set in the Device Status.*

***Note**:* *Bit 12 of the Device Status indicates a fault that prevents KP497 to identify which device state it is in. In this case, KP497 transitions to On State and sets bit 12 of the Device Status to 1 while bit 9 remains unchanged. In all other scenarios, including resets, external wake-ups or faults during Autonomous State, bit 9 of the Device Status is set and bit 12 is left unchanged.*

***Note**:* *If bit 13 of Device Status is 1, the data in the Device Status Register must be considered as corrupted and not reliable.*

3.2.5 Set Serial Interface

Command byte
0xD0

The command causes KP497 to lock its serial interface to the one provided by the external master. Once the command is executed successfully, the serial interface cannot be changed anymore and the Serial Interface Selection is no longer needed after reset.

3.2.5.1 Set Serial Interface parameters

The Set Serial Interface command has one parameter for selecting the serial interface:

1 Parameter (1 byte)
Type (1 byte)

The following table describes the parameter required for the Set Serial Interface command in detail:

Table 10 Parameters for Set Serial Interface command

Name	Bit(s)	Description
Type	7:0	Selects the type of serial interface: • 0 = I^2C, spike filter disabled • 1 = SPI • 2 = I^2C, spike filter enabled • All other values: Serial interface remains unset

***Note**: The I^2C spike filter can help to improve robustness of the I^2C communication in case there are potential disturbances on the PCB traces of the I^2C bus. Typically, it is not needed.*
In order to achieve I^2C data rates of 1 Mbit/s, the spike filter needs to be disabled, i.e. Type = 0.
If the serial interface is not set and I^2C is selected via the Serial Interface Selection after reset, the spike filter is disabled.

3.2.5.2 Set Serial Interface results

During the master read for the Set Serial Interface command, the KP497 provides the listed in the table below followed by the CRC:

1 Command result (1 byte)
Status (1 byte)

The serial interface was set successfully, if *Status* is 0. Otherwise, the *Status* indicates the fault detected by the Set Serial Interface according to the following table:

Table 11 Results of the Set Serial Interface command

Name	Bit(s)	Description
Status	0	(not used)
	1	1 if the value of *Type* is invalid, 0 else. If the bit is set, the serial interface remains unset.
	2	(not used)
	3	1 if the Serial Interface was already set previously, 0 else. If the bit is set, the serial interface remains unchanged.
	6:4	(not used)
	7	1 if writing the serial interface to flash failed, 0 else.

3.3 State transitions

3.3.1 Enter Autonomous State

Command byte
0xC1

The command sets KP497 into Autonomous State. Therefore, KP497 does not provide any data on the serial interface after successful reception of this command.

The initial Autonomous State Configuration is selected by the parameter Configuration ID of the command.

3.3.1.1 Enter Autonomous State parameters

The master write of the Enter Autonomous State command has one parameter that specifies the initial Autonomous State Configuration:

1 Parameter (1 byte)
Configuration ID (1 byte)

The following table describes the parameter required for the Enter Autonomous State command in detail:

Table 12 Parameters for Enter Autonomous State command

Name	Bit(s)	Description
Configuration ID	4:0	ID of the initial Autonomous State Configuration for the Autonomous State
	7:5	(not used)

***Note**: If the Configuration ID provided in the Enter Autonomous State command is not configured, the device re-enters On State instead of Autonomous State and the Device Status Register indicates a transition to an undefined or invalid Autonomous State configuration (see also section Read Device Status Register Results).*

3.3.2 Enter Sleep

Command byte
0xC2

The command sets KP497 into Sleep state. Therefore, KP497 does not provide data on the serial interface after successful reception of this command.

3.3.2.1 Enter Sleep parameters

The command has no parameters ($N = X = 0$).

3.4 Autonomous State

3.4.1 Read Autonomous State History

Command byte
0xA0

The command causes KP497 to provide the content of the Autonomous State History for read-out on the serial interface.

3.4.1.1 Read Autonomous State History parameters

The command has no parameters ($N = X = 0$).

3.4.1.2 Read Autonomous State History results

During the master read for the Read Autonomous State History command, the KP497 provides the listed in the table below followed by the CRC:

17 Command result (17 bytes)				
Length (1 byte)	ID_0 (1 byte)	ID_1 (1 byte)	…	ID_{15} (1 byte)

The results provided by the Read Autonomous State Hisotry command are described in detail in the following table:

Table 13 Results of the Read Autonomous State History command

Name	Bit(s)	Description
Length	7:0	Number of valid entries in the Autonomous State History.
ID_0	7:0	If *Length* ≥ 1 the ID of the least recently used Autonomous State Configuration still in the Autonomous State History, otherwise not used
ID_1	7:0	If *Length* ≥ 2 the ID of the Autonomous State Configuration that was used after ID_0. Otherwise not used
…		
ID_{14}	7:0	If *Length* ≥ 15 the ID of the Autonomous State Configuration used after ID_{13}. Otherwise not used
ID_{15}	7:0	If *Length* = 16 the ID of the Autonomous State Configuration used after ID_{14}. Otherwise not used

***Note**: One and the same Autonomous State Configuration ID may occur multiple times in the results of the Read Autonomous State History command. This can be the case for example, if after time-out the device transitions back to a previous Autonomous State configuration.*

Tip: Whenever Length > 0, $ID_{Length-1}$ is the ID of the Autonomous State Configuration last used during Autonomous State.

Tip: For a description of the Autonomous State History and its content see section "Autonomous State History" in the KP497 datasheet.

3.4.2 Read Autonomous State Pressure Buffer

Command byte
0xA1

The command causes KP497 to provide the content of the Autonomous State Measurement Buffer for pressure measurement to be read out on the serial interface.

3.4.2.1 Read Autonomous State Pressure Buffer parameters

The command has no parameters ($N = X = 0$).

3.4.2.2 Read Autonomous State Pressure Buffer results

During the master read for the Read Autonomous State Pressure Buffer command, the KP497 provides the listed in the table below followed by the CRC:

17 Command result (33 bytes)				
Length (1 byte)	$Pressure_0$ (2 bytes)	$Pressure_1$ (2 bytes)	…	$Pressure_{15}$ (2 bytes)

The results provided by the Read Autonomous State Pressure Buffer command are described in detail in the following table:

Table 14 Results of the Read Autonomous State Pressure Buffer command

Name	Bit(s)	Description
Length	7:0	Number of valid pressure measurement value in the Autonomous State Measurement Buffer for pressure measurements
$Pressure_0$	15:0	If *Length* ≥ 1 the least recent pressure value in the Measurement Buffer acquired during Autonomous State. Otherwise not used.
$Pressure_1$	15:0	If *Length* ≥ 2 the pressure value acquired one pressure measurement interval after $Pressure_0$ in Autonomous State. Otherwise not used.
…		
$Pressure_{14}$	15:0	If *Length* ≥ 14 the pressure value acquired one pressure measurement interval after $Pressure_{13}$ in Autonomous State. Otherwise not used.
$Pressure_{15}$	15:0	If *Length* = 15 the pressure value acquired one pressure measurement interval after $Pressure_{14}$ in Autonomous State. Otherwise not used.

Each pressure value $Pressure_0$, …, $Pressure_{15}$ is a 16-bit value transmitted with its high byte first. The resolution of each pressure value is 1/16 kPa/LSB, i.e. 16 LSB correspond to 1 kPa.

Tip: Whenever Length > 0, $Pressure_{Length-1}$ is the value acquired by the last pressure measurement during Autonomous State.

Tip: For a description of the Autonomous State Measurement Buffers and their content see section "Autonomous State Measurement Buffers" in the KP497 datasheet.

3.4.3 Read Autonomous State Acceleration Buffer

Command byte
0xA2

The command causes KP497 to provide the content of the Autonomous State Measurement Buffer for acceleration measurement to be read out on the serial interface.

3.4.3.1 Read Autonomous State Acceleration Buffer parameters

The command has no parameters ($N = X = 0$).

3.4.3.2 Read Autonomous State Acceleration Buffer results

During the master read for the Read Autonomous State Acceleration Buffer command, the KP497 provides the listed in the table below followed by the CRC:

17 Command result (33 bytes)				
Length (1 byte)	$Acceleration_0$ (2 bytes)	$Acceleration_1$ (2 bytes)	…	$Acceleration_{15}$ (2 bytes)

The results provided by the Read Autonomous State Acceleration Buffer command are described in detail in the following table:

Table 15 Results of the Read Autonomous State Acceleration Buffer command

Name	Bit(s)	Description
Length	7:0	Number of valid acceleration measurement value in the Autonomous State Measurement Buffer for acceleration measurements
$Acceleration_0$	15:0	If *Length* ≥ 1 the least recent acceleration value in the Measurement Buffer acquired during Autonomous State. Otherwise not used.
$Acceleration_1$	15:0	If *Length* ≥ 2 the acceleration value acquired one acceleration measurement interval after $Acceleration_0$ in Autonomous State. Otherwise not used.
…		
$Acceleration_{14}$	15:0	If *Length* ≥ 14 the acceleration value acquired one acceleration measurement interval after $Acceleration_{13}$ in Autonomous State. Otherwise not used.
$Acceleration_{15}$	15:0	If *Length* = 15 the acceleration value acquired one acceleration measurement interval after $Acceleration_{14}$ in Autonomous State. Otherwise not used.

Each acceleration value $Acceleration_0$, …, $Acceleration_{15}$ is a 16-bit value transmitted with its high byte first. The resolution of each acceleration value is 1/16 g/LSB, i.e. 16 LSB correspond to 1 g.

Tip: Whenever Length > 0, $Acceleration_{Length-1}$ is the value acquired by the last acceleration measurement during Autonomous State.

Tip: For a description of the Autonomous State Measurement Buffers and their content see section "Autonomous State Measurement Buffers" in the KP497 datasheet.

3.4.4 Read Acceleration Recording Statistics

Command byte
0xD7

The command causes KP497 to provide the results of the Acceleration recording for read-out on the serial interface.

3.4.4.1 Read Acceleration Recording Statistics parameters

The master write of the Read Acceleration Recording Statistics provides the number of bins and their size as parameters:

2 Parameters (2 bytes)	
Num_Bins (1 byte)	*Size_Bins* (1 byte)

The two parameters of the Read Acceleration Recording Statistics command are described in detail in the following table:

Table 16 Parameters of the Read Acceleration Recording Statistics command

Name	Bit(s)	Description
Num_Bins	7:0	Number of bins for the histogram to be provided by KP497. Valid range: 1, ..., 30
Size_Bins	7:0	Size of the bins for the histogram to be provided by KP497 in steps of a_{BIN_REC}. The nominal size of each bin, except the bin around 0 g, is a_{BIN_REC} × *Size_Bins*. The bin around 0 has the nominal range (-a_{BIN_REC}, a_{BIN_REC}). Valid range: 1, ..., 30

It is recommended to set *Num_Bins* and *Size_Bins* such that *Num_Bins* × *Size_Bins* = 30.

Tip: a_{BIN_REC} *is specified in section "Acceleration recording characteristics" of the KP497 datasheet.*

3.4.4.2 Read Acceleration Recording Statistics results

During the master read for the Read Acceleration Recording Statistics command, the KP497 provides the listed in the table below followed by the CRC:

2×*Num_Bins*+3 Command result (4×*Num_Bins*+6 bytes)					
Min (2 bytes)	*Max* (2 bytes)	BIN_{-Num_Bins} (2 bytes)	BIN_{-Num_Bins+1} (2 bytes)	…	BIN_{Num_Bins} (2 byte)

The results provided by the Read Acceleration Recording Statistics command are described in detail in the following table:

Table 17 Results of the Read Acceleration Recording Statistics command

Name	Bit(s)	Description
Min	15:0	The minimum of all acceleration values acquired during Acceleration Recording.
Max	15:0	The maximum of all acceleration values acquired during Acceleration Recording.
BIN_{-Num_Bins}	15:0	Number of acceleration measurements during Acceleration recording in bin -*Num_Bins* See also detailed description of bins below.

(table continues...)

Table 17 (continued) Results of the Read Acceleration Recording Statistics command

Name	Bit(s)	Description
BIN_{-Num_Bins+1}	15:0	Number of acceleration measurements during Acceleration recording in bin *-Num_Bins+1* See also detailed description of bins below.
…		
BIN_{Num_Bins}	15:0	Number of acceleration measurements during Acceleration recording in bin *Num_Bins* See also detailed description of bins below.

3.4.4.2.1 Definition of bins

Depending on the values of *Num_Bins* and *Size_Bins* not all bins in Table 17 are used: Whenever *Num_Bins* × *Size_Bins* ≥ 30, only bins -⌈30/*Size_Bins*⌉, …, ⌈30/*Size_Bins*⌉ are used. In this case, bin -⌈30/*Size_Bins*⌉ contains the number of acceleration samples taken during acceleration recording with a value $a_{Sample} \le$ -⌈30/*Size_Bins*⌉ × *Size_Bins* × a_{BIN_REC}, and bin -⌈30/*Size_Bins*⌉ contains the number of acceleration samples taken during acceleration recording with a value $a_{Sample} \ge$ ⌈30/*Size_Bins*⌉ × *Size_Bins* × a_{BIN_REC}. Additionally, the command returns the value 0 for bins *Num_Bins*, ..., -⌈30/*Size_Bins*⌉ - 1, ..., and for bins ⌈30/*Size_Bins*⌉ + 1, ..., *Num_Bins*. In the opposite case *Num_Bins* × *Size_Bins* < 30, all bins returned by the command are used with bin *-Num_Bins* providing the number of acceleration samples $a_{Sample} \le$ *-Num_Bins* × *Size_Bins* × a_{BIN_REC} and bin *Num_Bins* providing the number of acceleration samples $a_{Sample} \ge$ *Num_Bins* × *Size_Bins* × a_{BIN_REC}

Bin 0 always provides the number of acceleration samples taken during acceleration recording with an acceleration value a_{Sample} in the range $-a_{BIN_REC} < a_{Sample} < a_{BIN_REC}$.

All other bins have a size of *Size_Bins* × a_{BIN_REC}. That means that for -min(*Num_Bins* , ⌈30/*Size_Bins*⌉) < *n* < 0 bin *n* provides the number of acceleration samples with an acceleration value a_{Sample} in the range (*n* × *Size_Bins* - 1) × $a_{BIN_REC} < a_{Sample} \le$ ((*n* + 1) × *Size_Bins* - 1) × a_{BIN_REC}. Similarly, for 0 < n < min(*Num_Bins* , ⌈30/*Size_Bins*⌉), bin n provides the number of acceleration samples with an acceleration value a_{Sample} in the range ((*n* - 1) × *Size_Bins* + 1) × $a_{BIN_REC} < a_{Sample} \le$ (*n* × *Size_Bins* + 1) × a_{BIN_REC}.

This definition of the bins as they are returned by the Read Acceleration Recording Statistics command is also summarized in Figure 7 below. In Figure 7, *M* is defined as *M* = min(*Num_Bins* , ⌈30/*Size_Bins*⌉).

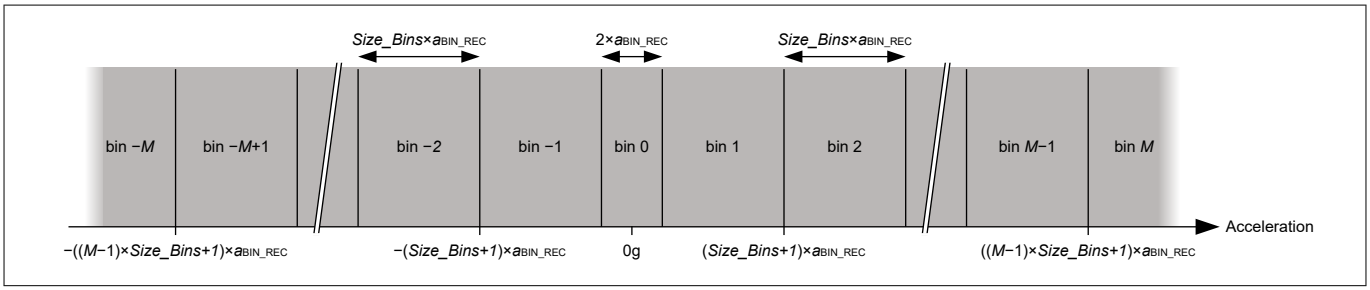


Figure 7 Definition of bins for acceleration recording

Examples for definition of bins

Example 1: *Size_Bins* = 2, *Num_Bins* = 15

The nominal acceleration ranges for the bins are

BIN_{-15}	BIN_{-14}	BIN_{-13}	…	BIN_{-1}	BIN_0	BIN_1	BIN_2	…	BIN_{14}	BIN_{14}
$a \le$ -58 g	-58 g < a ≤ -54 g	-54 g < a ≤ -50 g		-6 g < a ≤ -2 g	-2 g < a < -2 g	2 g ≤ a < 6 g	6 g ≤ a < 10 g		54 g ≤ a < 58 g	58 g ≤ a

Example 2: *Size_Bins* = 4, *Num_Bins* = 3

The nominal acceleration ranges for the bins are

BIN_{-3}	BIN_{-2}	BIN_{-1}	BIN_0	BIN_1	BIN_2	BIN_3
-26 g < a ≤ -18 g	-18 g < a ≤ -10 g	-10 g < a ≤ -2 g	-2 g < a < -2 g	2 g ≤ a < 10 g	10 g ≤ a < 18 g	18 g ≤ a < 26 g

Example 3: *Size_Bins* = 7, *Num_Bins* ≥ 5

The nominal acceleration ranges for the bins are

BIN_{-5}	BIN_{-4}	BIN_{-3}	BIN_{-2}	BIN_{-1}	BIN_0	BIN_1	BIN_2	BIN_3	BIN_3	BIN_5
a ≤ -58 g	-58 g < a ≤ -44 g	-44 g < a ≤ -30 g	-30 g < a ≤ -16 g	-16 g < a ≤ -2 g	-2 g < a < -2 g	2 g ≤ a < 16 g	16 g ≤ a < 30 g	30 g ≤ a < 44 g	44 g ≤ a < 58 g	58 g ≤ a

Additionally, when *Num_Bins* > 5, the bins BIN_{-Num_Bins}, …, BIN_{-6} and BIN_6, …, BIN_{Num_Bins} are provided on the serial interface as well but will not contain any valid values.

Example 4: *Size_Bins* = 3, *Num_Bins* ≥ 11

The nominal acceleration ranges for the bins are

BIN_{-10}	BIN_{-9}	BIN_{-8}	…	BIN_{-1}	BIN_0	BIN_1	BIN_2	…	BIN_9	BIN_{10}
a ≤ -56 g	-56 g < a ≤ -50 g	-50 g < a ≤ -44 g		-8 g < a ≤ -2 g	-2 g < a < -2 g	2 g ≤ a < 8 g	8 g ≤ a < 14 g		50 g ≤ a < 56 g	56 g ≤ a

Additionally, the bins BIN_{-Num_Bins}, …, BIN_{-11} and BIN_{11}, …, BIN_{Num_Bins} are provided on the serial interface as well but will not contain any valid values.

3.4.5 Save Autonomous State Configuration

Command byte
0xD4

The command stores a new Autonomous State Configuration or overwrites an existing one in the device.

3.4.5.1 Save Autonomous State Configuration parameters

During the Save Autonomous State Configuration command the master provides the ID of the Autonomous State Configuration as well as the content of the configuration. If a configuration with the same ID already exists, the existing one is replaced by the one provided in the command. Otherwise, a new configuration is created.

The following table lists the command byte and the parameters of the Save Autonomous State Configuration command:

6 Parameter (41 byte)					
ID (1 byte)	*Precounter* (1 byte)	*Postcounter* (2 byte)	*Timeout* (1 byte)	*Meas_1* (18 byte)	*Meas_2* (18 byte)

The following table describes the parameters required for the Save Autonomous State Configuration command in detail:

Table 18 Parameters for Save Autonomous State Configuration command

Name	Bit(s)	Description
ID	4:0	The Configuration ID of the Autonomous State Configuration
	7:5	(not used)
Precounter	7:0	Precounter of the Autonomous State Timer in milliseconds. If the value provided is 0, KP497 replaces it by 1. Similarly, if the value provided is larger than 179, it is replaced by 179. See also the note below the table. Valid range: 1, ..., 179.
Postcounter	15:0	Postcounter of the Autonomous State Timer. The value must be transmitted with its high byte first. If the value provided is larger than 2^{12}, KP497 replaces it by 2^{12}. See also the note below the table. Valid range: 0, ..., 2^{12}.
Timeout	7:0	Sets the timeout for the Autonomous State Configuration. When set to 0, the timeout is disabled. Otherwise, the timeout occurs after $2^{Timeout}$ Autonomous State Timer periods. If the value provided is larger than 31, KP497 replaces it by 31. Valid range: 0, ..., 31.
Meas_1	143:0	Sets the measurement configuration for the high priority measurement. See Table 19 for details.
Meas_2	143:0	Sets the measurement configuration for the low priority measurement. See Table 19 for details.

***Note**: The overall period of the Autonomous State Timer in milliseconds is given by Precounter × Postcounter. Whenever this product is smaller than the time KP497 needs to perform the measurement(s) and evaluate the thresholds, the device performs measurements and evaluates the thresholds as fast as possible. That means, that once all measurements are performed and all thresholds are evaluated, KP497 immediately starts performing the next measurement(s) for threshold evaluation.*

The following table describes the content of *Meas_1* and *Meas_2* in detail:

Table 19 Measurement Configurations

Name	Bit(s)	Description
Sensor	7:0	The sensor for the measurement. The valid values are: • 0 = Pressure • 1 = Acceleration • 0xFF = Measurement is disabled For any other value, KP497 selects the measurement based on bit 0.
Postcounter	7:0	The measurement postcounter. The value is the number of times that the Autonomous State Timer needs to elapse between each acquisition and threshold evaluation of the measurement. If the value provided is 0, KP497 replaces it by 1. Valid range: 1, ..., 255.
Rec_Samples	15:0	The number of samples to be acquired by KP497 during Acceleration Recording, in case it transitions to Acceleration Recording from this Autonomous State Configuration. The number is a 16-bit value and must be transmitted with high byte first.
Threshold_1	55:0	Sets the threshold configuration for the high priority threshold. See Table 20 for details.
Threshold_2	55:0	Sets the threshold configuration for the low priority threshold. See Table 20 for details.

Attention**: **If any of the measurement specific postcounters Meas_1.Postounter or Meas_2.Postcounter is set to 0, the bit 0 will be set in the Status of the Save Autonomous State Configuration command, independent of the values provided for Meas_1.Sensor and Meas_2.Sensor

Tip: The measurement specific postcounters Meas_1.Postcounter and Meas_1.Postcounter allow to set different measurement intervals for pressure and acceleration measurements in Autonomous state when both measurements are enabled.

The following table describes the content of *Threshold_1* and *Threshold_2* in detail:

Table 20 Threshold Configurations

Name	Bit(s)	Description
Config	0	If 1, the lower threshold is enabled. If 0, the lower threshold is disabled and the values in *Trans_Lower* and *Thrld_Lower* are ignored.
	1	If 1, the upper threshold is enabled. If 0, the upper threshold is disabled and the values in *Trans_Upper* and *Thrld_Upper* are ignored.
	3:2	The type of the threshold: • 0 = Absolute threshold • 1 = Rate of change threshold • 2 = Threshold relative to On State • 3 = Threshold relative to configuration start
	7:4	Number of measurements used to determine rate of change. For evaluation the rate of change threshold, the device performs a linear fit through the last *Config*[7:4] + 1 measurements. That is, the device uses a sliding window to calculate the rate of change and the number of measurements in the sliding window is one larger than the value provided here. If the value is 0, the device uses 2 measurements. Only used, if value in bits 3:2 is 1
Trans_Lower	4:0	ID of the configuration that KP497 transitions to, if the lower threshold of this threshold configuration is violated. Only used, if the value in bits 7:6 is 1.
	5	If 1, the ALERT pin is pulled high when the lower threshold of this threshold configuration is violated until the next measurement in Autonomous State is performed. If 0, the ALERT pin remains low. Only used, if the value in bits 7:6 is 1. For any other value in bits 7:6, the ALERT pin goes high when On State is entered.
	7:6	Selection of the action taken by KP497 when the lower threshold of this threshold configuration is violated: • 0 = Transition to On State • 1 = Change to Autonomous State Configuration indicated in bits 4:0 • 2 = Transition to Acceleration Recording • 3 = Transition to On State
Thrld_Lower	15:0	Value for the lower threshold. The value is a 16-bit value and is provided with the high byte first with a resolution of 1/16 kPa/LSB (i.e. 1 kPa corresponds to 16 LSB) for pressure and a resolution of 1/16 g/LSB (i.e. 1 g corresponds to 16 LSB) for acceleration.

(table continues...)

Table 20 (continued) Threshold Configurations

Name	Bit(s)	Description
Trans_Upper	4:0	ID of the configuration that KP497 transitions to, if the upper threshold of this threshold configuration is violated. Only used, if the value in bits 7:6 is 1.
	5	If 1, the ALERT pin is pulled high when the upper threshold of this threshold configuration is violated until the next measurement in Autonomous State is performed. If 0, the ALERT pin remains low. Only used, if the value in bits 7:6 is 1. For any other value in bits 7:6, the ALERT pin goes high when On State is entered.
	7:6	Selection of the action taken by KP497 when the upper threshold of this threshold configuration is violated: • 0 = Transition to On State • 1 = Change to Autonomous State Configuration indicated in bits 4:0 • 2 = Transition to Acceleration Recording • 3 = Transition to On State
Thrld_Upper	15:0	Value for the upper threshold. The value is a 16-bit value and is provided with the high byte first with a resolution of 1/16 kPa/LSB (i.e. 1 kPa corresponds to 16 LSB) for pressure and a resolution of 1/16 g/LSB (i.e. 1 g corresponds to 16 LSB) for acceleration.

Tip: The threshold types that can be selected through the Config parameter of the Threshold Configuration are described in section "Autonomous State Thresholds" of the KP497 datasheet.

3.4.5.2 Save Autonomous State Configuration results

During the master read for the Save Autonomous State Configuration command, the KP497 provides the listed in the table below followed by the CRC:

1 Command result (1 byte)
Status (1 byte)

The Save Autonomous State Configuration was successful, if *Status* is 0. Otherwise, the *Status* indicates the fault detected by the Save Autonomous State Configuration according to the following table:

Table 21 Results of the Save Autonomous State Configuration command

Name	Bit(s)	Description
Status	0	1 if one of the parameters *ID*, *Precounter*, *Postcounter*, *Timeout*, *Meas_x.Sensor*, or *Meas_x.Postcounter* was outside its valid range. The Autonomous State Configuration is still saved, but with adjusted values as described in section Save Autonomous State Configuration Parameters. Otherwise 0.
	1	1 if at least one of the following conditions is fulfilled: • *Meas_1.Sensor* = *Meas_2.Sensor* • *Meas_1*.Sensor = 0xFF and *Meas_2.Threshold_1.Config*[1:0] = 0 and *Meas_2.Threshold_2.Config*[1:0] = 0 • *Meas_2.Sensor* = 0xFF and *Meas_1.Threshold_1.Config*[1:0] = 0 and *Meas_1.Threshold_2.Config*[1:0] = 0 • *Meas_1.Threshold_1.Config*[1:0] = 0 and *Meas_1.Threshold_2.Config*[1:0] = 0 and *Meas_2.Threshold_1.Config*[1:0] = 0 and *Meas_2.Threshold_2.Config*[1:0] = 0. Otherwise 0. If this bit is set, the Autonomous State Configuration is *not* saved. *Note: Config[1:0] denotes bits 1 and 0 of Config*.
	2	1, if an Autonomous State Configuration with the specified *ID* already existed but was corrupt. The new Autonomous State Configuration is still saved but may not be reliable anymore. KP497 considers the Autonomous State configuration with the specified *ID* to be written more than 255 times and bit 6 is set as well. 0 if the Autonomous State Configuration with the specified *ID* is written for the first time or if the CRC check of the configuration that is overwritten was successful.
	5:3	(not used)
	6	1, if the Autonomous State Configuration with the specified *ID* has been written more than 255 times. The new Autonomous State Configuration is still saved but may not be reliable. Otherwise 0.
	7	1 if writing the Autonomous State Configuration to flash failed, 0 else.

3.4.6 Read Autonomous State Configuration

Command byte
0xD5

The command causes KP497 to provide the content of an Autonomous State Configuration on the serial interface.

3.4.6.1 Read Autonomous State Configuration parameters

The external master needs to provide the Configuration ID of the Autonomous State Configuration to be read out:

1 Parameter (1 byte)
ID (1 byte)

The following table describes the parameter required for the Read Autonomous State Configuration command in detail:

Table 22 Parameters for Save Autonomous State Configuration command

Name	Bit(s)	Description
ID	4:0	The Configuration ID of the Autonomous State Configuration
	7:5	(not used)

3.4.6.2 Read Autonomous State Configuration results

During the master read for the Read Autonomous State Configuration command, the KP497 provides the listed in the table below followed by the CRC:

7 Command results (42 byte)						
Status (1 byte)	*ID* (1 byte)	*Precounter* (1 byte)	*Postcounter* (2 byte)	*Timeout* (1 byte)	*Meas_1* (18 byte)	*Meas_2* (18 byte)

The results provided by the Read Autonomous State Configuration command are described in detail in the following table:

Table 23 Results for Read Autonomous State Configuration command

Name	Bit(s)	Description
Status	7:0	1 if the CRC check of the Autonomous State Configuration failed, 0 else
ID	7:0	The Configuration ID of the Autonomous State Configuration
Precounter	7:0	Precounter of the Autonomous State Timer in milliseconds. See also the note below the table.
Postcounter	15:0	Postcounter of the Autonomous State Timer. The value is transmitted with its high byte first. See also the note below the table.
Timeout	7:0	Sets the timeout for the Autonomous State Configuration. 0, if the timeout is disabled. Otherwise, the timeout occurs after $2^{Timeout}$ Autonomous State Timer periods.

(table continues...)

Table 23 (continued) Results for Read Autonomous State Configuration command

Name	Bit(s)	Description
Meas_1	143:0	The measurement configuration for the high priority measurement. See Table 19 for details.
Meas_2	143:0	The measurement configuration for the low priority measurement. See Table 19 for details.

***Note**: The overall period of the Autonomous State Timer in milliseconds is given by Precounter × Postcounter. Whenever this product is smaller than the time KP497 needs to perform the measurement(s) and evaluate the thresholds, the device performs measurements and evaluates the thresholds as fast as possible. That means, that once all measurements are performed and all thresholds are evaluated, KP497 immediately starts performing the next measurement(s) for threshold evaluation.*

3.5 Data storage

3.5.1 Write Data Line

Command byte
0xB1

The command stores 32 byte of custom data in KP497's flash memory. The data can be read out again later with the Read Data Line command.

KP497 does not use the data provided in this command.

3.5.1.1 Write Data Line parameters

The Write Data Line command has the following parameters:

3 Parameter (35 bytes)		
Keyword (1 byte)	*Address* (2 bytes)	*Data* (32 bytes)

The following table describes the parameter required for the Write Data Line command in detail:

Table 24 Parameters for Self Test command

Name	Bit(s)	Description
Keyword	7:0	If 0x9D, the provided *Data* is stored in the device's flash memory. Otherwise, no action is performed.
Address	15:0	Address of the 32-byte block for storing the *Data*. The address must be a multiple of 32 and provided with its high byte first. Valid range: 0x0000, …, 0x0BE0
Data	255:0	The data to be stored in the device's flash memory.

Attention**: **The last two bytes of each 1 kB block of the flash memory are used as lockbytes (see section "Data storage" of the KP497 datasheet).

3.5.1.2 Write Data Line results

During the master read for the Write Data Line command, the KP497 provides the listed in the table below followed by the CRC:

1 Command result (1 byte)
Status (1 byte)

The Write Data Line was successful, if *Status* is 0. Otherwise, the *Status* indicates the fault detected by the Write Data Line according to the following table:

Table 25 Results of the Write Data Line command

Name	Bit(s)	Description
Status	0	always 1 if any of the bits 7:1 is set, 0 else.
	1	1 if the provided address is invalid, 0 else.
	3	1 if the keyword is not correct, 0 else.
	5	1 if the sector that the *Address* belongs to is locked, 0 else.
	7	1 if the verification of the data written to flash failed, 0 else.

3.5.2 Read Data Line

Command byte
0xB2

The command causes KP497 to provide custom data previously stored in the device's flash memory on the serial interface.

3.5.2.1 Read Data Line parameters

The Read Data Line command has one parameter for selecting the 32-byte block of data to be read:

1 Parameter (2 byte)
Address (2 byte)

The following table describes the parameter required for the Read Data Line command in detail:

Table 26 Parameters for Read Data Line command

Name	Bit(s)	Description
Address	15:0	The address of the 32-byte block of data to be read. The address must be provided with its high byte first. It does not need to be 32-byte aligned. Valid range: 0x0000, …, 0x0BE0

3.5.2.2 Read Data Line results

During the master read for the Read Data Line command, the KP497 provides the listed in the table below followed by the CRC:

2 Command result (33 byte)	
Status (1 byte)	*Data* (32 byte)

The Read Data Line was successful, if *Status* is 0. Otherwise, the *Status* indicates the fault detected by the Read Data Line according to the following table:

Table 27 Results of the Read Data Line command

Name	Bit(s)	Description
Status	0	always 1 if any of the bits 7:1 is set, 0 else.
	1	1 if the provided address is invalid, 0 else.
	7	1 if the verification of the data read from flash failed, 0 else.
Data	255:0	The custom data stored in flash. The data is provided in the same order as it was provided in the Write Data Line command.

3.6 Processing times for serial interface commands

The following table provides the maximum processing time for each command. Unless specifically mentioned otherwise, the values can be understood as the maximum processing time over all possible combinations of input parameters.

Table 28 Maximum processing times of serial interface commands

Command	Condition	Processing time in milliseconds
Measure Pressure	without ADC time-out (Bit 7 of *Status* is 0)	1.3
	with ADC time-out (Bit 7 of *Status* is 1)	4.9
Measure Acceleration	without ADC time-out (Bit 7 of *Status* is 0)	0.52
	with ADC time-out (Bit 7 of *Status* is 1)	4.9
Measure Temperature	without ADC time-out (Bit 7 of *Status* is 0)	0.15
	with ADC time-out (Bit 7 of *Status* is 1)	2.5
Read ID	–	0.076
Self Test	without memory check (*Config* = 0)	2.391
	with memory check (*Config* ≠ 0)	280
Read Device Status Register	–	0.16
Set Serial Interface	–	8.7
Read Autonomous State History	*Length* = 3 in command results	0.24
	Length = 16 in command results	0.82
Read Autonomous State Pressure Buffer	–	0.15
Read Autonomous State Acceleration Buffer	–	0.15
Read Acceleration Recording Statistics	–	0.44
Save Autonomous State Configuration	–	16
Read Autonomous State Configuration	–	0.24
Write Data Line	–	8.8
Read Data Line	–	0.27

For the commands Trigger Softreset, Enter Autonomous State, and Enter Sleep the device does not remain in On State and therefore, these commands are left out of Table 28. After the Trigger Softreset command the time it takes for the serial interface of KP497 to be ready for the next serial interface command is given by the Power On time t_{INI} specified in section “Electrical characteristics” of the KP497 datasheet.

4 Examples for Autonomous State Configurations

This section provides examples for Autonomous State Configurations and how to store them in the device in order to illustrate the use of Autonomous State Configurations.

The examples refer to the definitions of the parameters of the Save Autonomous State Configuration command and of the results of the Read Autonomous State Configuration command.

4.1 Example 1: Single stage pressure monitoring

This example describes how to store and use a Autonomous State Configuration that consists of only one stage with:

- Monitoring of pressure only
- Pressure measurement interval 10 ms
- Upper absolute threshold at 150 kPa, all other thresholds disabled
- Transition to On State
- No timeout

In this example, the Autonomous State Configuration is assigned the ID 0. To store the Autonomous State Configuration in the device, the Save Autonomous State Configuration command is executed with the following parameters:

Table 29 Example Autonomous State Configuration for single stage pressure monitoring

Parameter	Value		
	Decimal	**Hexadecimal**	**Binary**
ID	0	0x00	0000 0000
Precounter	10	0x0A	0000 1010
Postcounter	1	0x0001	0000 0000 0000 0001
Timeout	0	0x00	0000 0000
Meas_1.Sensor	0	0x00	0000 0000
Meas_1.Postcounter	1	0x01	0000 0001
Meas_1.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Config	2	0x02	0000 0010
Meas_1.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Upper	2400	0x0960	0000 1001 0110 0000
Meas_1.Threshold_2.Config	0	0x00	0000 0000
Meas_1.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000
Meas_2.Sensor	255	0xFF	1111 1111
Meas_2.Postcounter	1	0x01	0000 0001

(table continues...)

Table 29 (continued) Example Autonomous State Configuration for single stage pressure monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
Meas_2.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Config	0	0x00	0000 0000
Meas_2.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Upper	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Config	0	0x00	0000 0000
Meas_2.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000

The *Precounter* and *Postcounter* as well as the *Meas_1.Postcounter* are determined in such a way that the product *Precounter* × *Postcounter* × *Meas_1.Postcounter* is the measurement interval of 10 ms with the *Precounter* as large as possible and the *Meas_1.Postcounter* as small as possible.

Since the configuration monitors only pressure, *Meas_1.Sensor* is set to select the pressure measurement (*Meas_1.Sensor* = 0) while *Meas_2.Sensor* is set to disable the second measurement. With this setting, the device only takes into account the parameters *Meas_1.Postcounter*, *Meas_1.Rec_Samples*, *Meas_1.Threshold_1*, and *Meas_1.Threshold_2*. The parameters *Meas_2.Postcounter*, *Meas_2.Rec_Samples*, *Meas_2.Threshold_1*, and *Meas_2.Threshold_2*, on the other hand, are ignored but must still be provided to the Save Autonomous State Configuration command. *Meas_2.Postcounter* is set to 1 in order to avoid bit 0 of the *Status* to be set.

With *Meas_1.Threshold_1.Config* set to 0x02, the upper threshold is enabled (bit 1 set) and the lower threshold is disabled (bit 0 unset). Moreover, since bits 3 and 2 are both 0, the absolute threshold is selected. Bits 7:4 are ignored due to the selection of the absolute threshold. Since only the upper threshold is enabled, the parameters *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Thrld_Lower* are ignored by the device but must still be provided.

In order to cause the device to transition to On State when the upper absolute threshold is exceeded, bits 7 and 6 of *Meas_1.Threshold_1.Trans_Upper* are set to 0. Due to selection of the transition to On State, all other bits of *Meas_1.Threshold_1.Trans_Upper* are ignored by the device.

The upper threshold of 150 kPa is set by the parameter *Meas_1.Threshold_1.Thrld_Upper*: The value of 2400 is derived by the formula *Meas_1.Threshold_1.Thrld_Upper* = 150 kPa × 16 LSB/kPa.

Finally, bits 1 and 2 of *Meas_1.Threshold_2.Config* are both set to 0 such that the second threshold for the pressure measurement is disabled. All other bits of *Meas_1.Threshold_2.Config* as well as the values for *Meas_1.Threshold_2.Trans_Lower*, *Meas_1.Threshold_2.Thrld_Lower*, *Meas_1.Threshold_2.Trans_Upper*, and *Meas_1.Threshold_2.Thrld_Upper* are ignored by the device but must still be provided in the Save Autonomous State Configuration command.

The resulting command on the serial interface to store this Autonomous State Configuration including command byte and CRC in hexadecimal is:

```
0xD4 0x00 0x0A 0x00 0x01 0x00 0x00 0x01 0x00 0x00 0x02 0x00 0x00 0x00 0x00 0x09 0x60 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0xFF 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x90 0x82
```

Once the command shown above is performed successfully, the single stage pressure monitoring described in this section can be started using the Enter Autonomous State command. The corresponding command including its command byte and CRC in hexadecimal values is:

```
0xC1 0x00 0x38 0x6A
```

4.2 Example 2: Three stage pressure and acceleration monitoring

This example focusses on demonstrating the implementation of multi-stage monitoring of pressure and acceleration in parallel including the acceleration recording. The example uses three stages implemented by four Autonomous State Configurations

1. Autonomous State Configuration with ID 0x01:
   - Pressure measurement every 10 s with
     - Upper rate of change threshold at a rate of 1 kPa/s, calculated over 4 measurements and with transition to Autonomous State Configuration with ID 0x02
     - Upper absolute threshold of 200 kPa with transition to On State
     - No further pressure threshold activated
   - Acceleration measurement every 40 ms with
     - Upper threshold of 5 g relative to the acceleration at the transition form On State to Autonomous State
     - Lower threshold of -5 g relative to the acceleration at the transition form On State to Autonomous State
     - Both thresholds transition to Autonomous State Configuration with ID 0x08
     - No further acceleration threshold activated
   - No timeout
2. Autonomous State Configuration with ID 0x02:
   - Pressure measurement every 25 ms with
     - Upper rate of change threshold at a rate of 10 kPa/s, calculated over 16 measurements and with transition to Autonomous State Configuration with ID 0x03 and intermediate alert
     - Upper absolute threshold of 200 kPa with transition to On State
     - No further pressure threshold activated
   - Acceleration measurement every 10 ms with
     - Upper threshold of 5 g relative to the acceleration at the transition form On State to Autonomous State
     - Lower threshold of -5 g relative to the acceleration at the transition form On State to Autonomous State
     - Both thresholds transition to Autonomous State Configuration with ID 0x08
     - No further acceleration threshold activated
   - Time-out after 1280 ms
3. Autonomous State Configuration with ID 0x03:
   - Continuous Pressure measurements (i.e. roughly every 2 ms)
   - Upper absolute threshold of 200 kPa with transition to On State
   - No further pressure threshold activated
   - No acceleration monitoring
   - Time-out after 128 measurements
4. Autonomous State Configuration with ID 0x08:
   - Acceleration monitoring only with measurement interval 1 ms
   - Upper threshold of 5 g relative to the acceleration at the start of Autonomous State Configuration ID 0x08
   - Lower threshold of -5 g relative to the acceleration at the start of Autonomous State Configuration ID 0x08
   - Both thresholds transition to Acceleration Recording with 10,000 samples
   - Time-out after 64 ms

The following flow diagram illustrates how the device transitions in this example from one Configuration to another, to Acceleration recording or to On State:

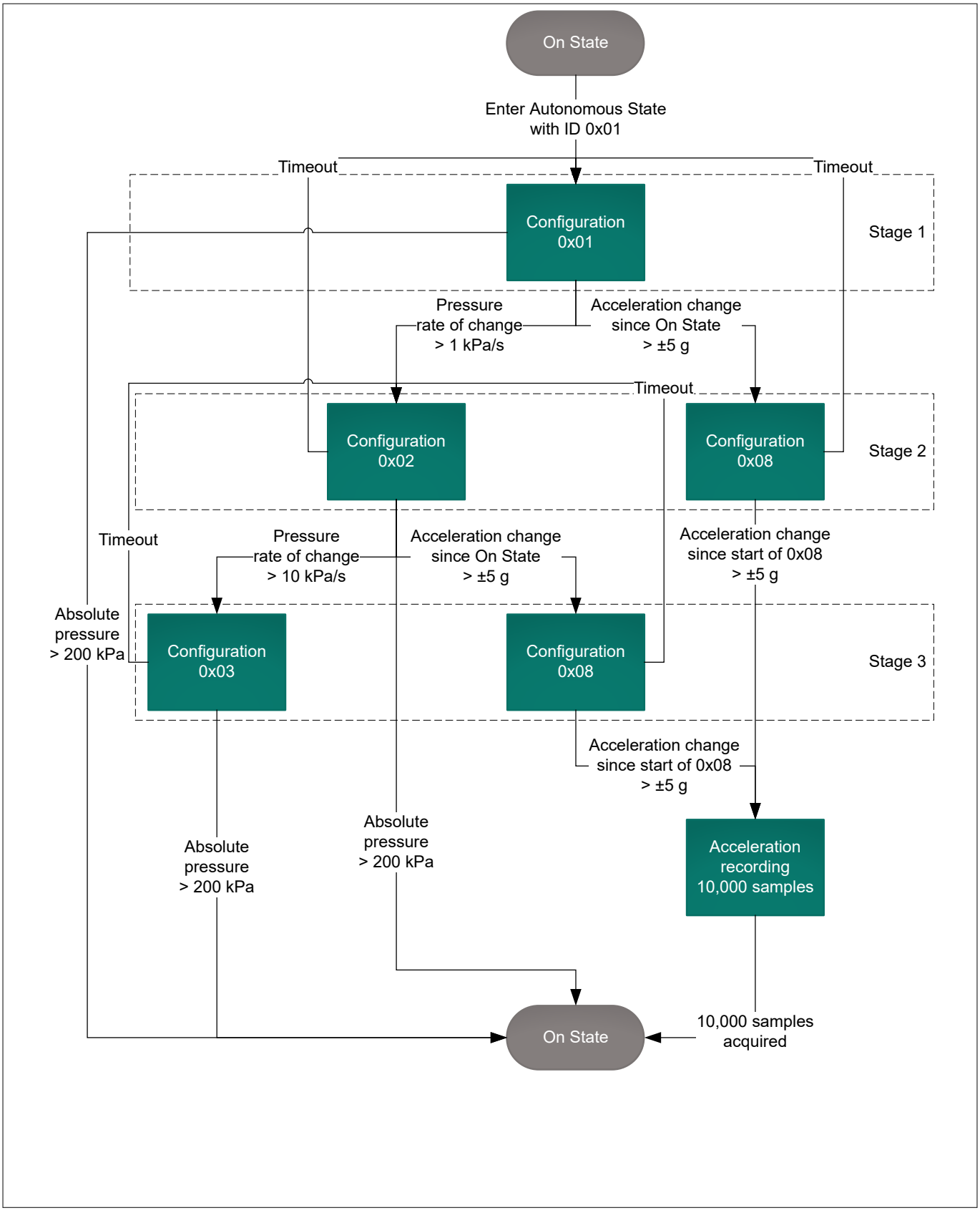


Figure 8 Flow of Autonomous State Configurations for Example 2: Three stage pressure and acceleration monitoring

In this example, the Autonomous State Configuration for the first stage is assigned the ID 1, the one for the second stage, if a pressure threshold is exceeded the ID 2 and the third stage of the pressure monitoring the ID

3. If an acceleration threshold in the first stage or during the second pressure monitoring stage is exceeded, the Autonomous State Configuration with ID 8 is entered. To store the Autonomous State Configurations in the device, the Save Autonomous State Configuration command is executed once for each of the Autonomous State Configurations. The sections Chapter 4.2.1, Chapter 4.2.2, Chapter 4.2.3, and Chapter 4.2.4 describe the parameters and the resulting command on the serial interface to store the Autonomous State Configuration for each configuration separately.

Once the Autonomous State Configurations are stored successfully, the three stage pressure and acceleration monitoring described in this section can be started using the Enter Autonomous State command. The corresponding command including its command byte and CRC in hexadecimal values is:

```
0xC1 0x01 0x28 0x4B
```

4.2.1 Autonomous State Configuration for first stage (0x01)

The Autonomous State Configuration used as initial configuration in the three stage monitoring in this example has the ID 0x01. The parameters used in the Save Autonomous State Configuration command to store the Autonomous State Configuration with ID 0x01 are summarized in the following table:

Table 30 Example Autonomous State Configuration 0x01 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	**Hexadecimal**	**Binary**
ID	1	0x01	0000 0001
Precounter	40	0x28	0010 1000
Postcounter	1	0x0001	0000 0000 0000 0001
Timeout	0	0x00	0000 0000
Meas_1.Sensor	0	0x00	0000 0000
Meas_1.Postcounter	250	0xFA	1111 1010
Meas_1.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Config	54	0x36	0011 0110
Meas_1.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Trans_Upper	66	0x42	0100 0010
Meas_1.Threshold_1.Thrld_Upper	160	0x00A0	0000 0000 1010 0000
Meas_1.Threshold_2.Config	2	0x02	0000 0010
Meas_1.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Upper	3200	0x0C80	0000 1100 1000 0000
Meas_2.Sensor	1	0x01	0000 0001

(table continues...)

Table 30 (continued) Example Autonomous State Configuration 0x01 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
Meas_2.Postcounter	1	0x01	0000 0001
Meas_2.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Config	11	0x0B	0000 1011
Meas_2.Threshold_1.Trans_Lower	72	0x48	0100 1000
Meas_2.Threshold_1.Thrld_Lower	-80	0xFFB0	1111 1111 1011 0000
Meas_2.Threshold_1.Trans_Upper	72	0x48	0100 1000
Meas_2.Threshold_1.Thrld_Upper	80	0x0050	0000 0000 0011 0010
Meas_2.Threshold_2.Config	0	0x00	0000 0000
Meas_2.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000

The *Precounter* and *Postcounter* as well as the *Meas_1.Postcounter* are determined in such a way that the product *Precounter* × *Postcounter* is the greatest common divisor of the measurement intervals for pressure and acceleration measurement: For an acceleration measurement interval of 40 ms and a pressure measurement interval of 10 s, the greatest common divisor is 40ms. Setting *Precounter* and *Postcounter* such that the product *Precounter* × *Postcounter* is the greatest common divisor of both measurement intervals minimizes the values for *Meas_1.Postcounter* and *Meas_2.Postcounter* leading to the smallest average charge consumption. The value for *Meas_1.Postcounter* is then determined by dividing the pressure measurement interval of 10s by the product *Precounter* × *Postcounter*, i.e. *Meas_1.Postcounter* = 10 s / 40 ms = 250. Similarly, *Meas_2.Postcounter* is calculated by dividing the acceleration measurement interval of 40ms by the product *Precounter* × *Postcounter*, i.e. *Meas_2.Postcounter* = 40 ms / 40 ms = 1.

With *Meas_1.Sensor* set to select the pressure measurement (*Meas_1.Sensor* = 0) and *Meas_2.Sensor* set to select the acceleration measurement (*Meas_2.Sensor* = 1), pressure and acceleration are monitored in parallel. Since the pressure measurement is selected as first measurement via *Meas_1*, the pressure measurement has higher priority. That means that whenever the postcounters for pressure and acceleration measurement elapse at the same time (in this case every 10 s), the pressure measurement is taken first and if one of the pressure thresholds is violated, the acceleration measurement is not performed any more.

With *Meas_1.Threshold_1.Config* set to 0x36, the upper threshold is enabled (bit 1 set) and the lower threshold is disabled (bit 0 unset). Moreover, since bit field 3:2 is set to 1, the rate of change threshold is selected. With the value 3 in bits 7:4 of *Meas_1.Threshold_1.Config* the number of measurements over which the rate of change is determined is set to 4. Since only the upper threshold is enabled, the parameters *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Thrld_Lower* are ignored by the device but must still be provided.

In order to cause the device to remain in Autonomous State but change to the Autonomous State Configuration with ID 0x02 when the upper rate of change threshold is exceeded, bit field 7:6 of *Meas_1.Threshold_1.Trans_Upper* is set to 1 while bits 4:0 are set to the ID of the Autonomous State Configuration 0x02. Since bit 5 of *Meas_1.Threshold_1.Trans_Upper* is 0, no alert is generated when the upper rate of change threshold is violated.

For the rate of change threshold, the threshold is provided with a resolution of 1/16 kPa/LSB per measurement interval. Therefore, with the measurement interval of 10 s, setting the upper threshold through parameter *Meas_1.Threshold_1.Thrld_Upper* to 160, the resulting upper rate of change threshold is 160 LSB × 1/16 kPa/LSB / 10 s = 1 kPa/s.

The second threshold configuration *Meas_1.Threshold_2* enables the absolute upper threshold through the value *Meas_1.Threshold_2.Config* = 0x02. As the second threshold, the absolute upper threshold is for each measurement only evaluated, if the rate of change threshold is not violated. Since bit 0 of *Meas_1.Threshold_2.Config* is 0, the lower absolute threshold is disabled and the parameters *Meas_1.Threshold_2.Trans_Lower* and *Meas_1.Threshold_2.Thrld_Lower* are ignored by the device but must still be provided. Bits 7:4 of *Meas_1.Threshold_2.Config* are ignored for the absolute threshold as well.

In order to cause the device to transition to On State when the upper absolute threshold is exceeded, bits 7:6 of *Meas_1.Threshold_2.Trans_Upper* are set to 0. The remaining bits of *Meas_1.Threshold_2.Trans_Upper* are ignored.

The upper absolute threshold is set through parameter *Meas_1.Threshold_1.Thrld_Upper* to 3200 LSB × 1/16 kPa/LSB = 200 kPa.

With *Meas_2.Threshold_1.Config* set to 0x0B, both the lower and upper thresholds are enabled (bits 0 and 1 both set). Moreover, since bit field 3:2 is set to 2, the acceleration threshold is relative to On State, i.e. relative to the first acceleration measurement taken at the transition from On State to Autonomous State. Bits 7:4 of *Meas_2.Threshold_1.Config* is ignored due to the selection of the relative threshold.

In order to cause the device to remain in Autonomous State but change to the Autonomous State Configuration with ID 0x08 when either the upper or lower threshold is violated both *Meas_2.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Trans_Upper* are set to 0x48, which provides the ID 0x08 in bits 4:0 and specifies the transition to another Autonomous State configuration through the value 2 in bits 7:6. The intermediate alert is disabled for both transitions, because bit 5 is set to 0.

The relative thresholds *Meas_2.Threshold_1.Lower* and *Meas_2.Threshold_1.Upper* have a resolution of 1/16 g/LSB. That means the values -80 for *Meas_2.Threshold_1.Lower* and 80 for *Meas_2.Threshold_1.Upper* sets the lower threshold to -80 LSB × 16 g/LSB = -5 g and the upper threshold to +5 g.

The resulting command on the serial interface to store this Autonomous State Configuration including command byte and CRC in hexadecimal is:

```
0xD4 0x01 0x28 0x00 0x01 0x00 0x00 0xFA 0x00 0x00 0x36 0x00 0x00 0x00 0x42 0x00 0xA0 0x02 0x00
0x00 0x00 0x00 0x0C 0x80 0x01 0x01 0x00 0x00 0x0B 0x48 0xFF 0xB0 0x48 0x00 0x50 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x0C 0x9B
```

4.2.2 Autonomous State Configuration for second stage pressure monitoring (0x02)

When the pressure rate of change threshold in configuration 0x01 is exceeded, the device transitions to the Autonomous State Configuration with ID 0x02. The parameters used in the Save Autonomous State Configuration command to store the Autonomous State Configuration with ID 0x02 are summarized in the following table:

Table 31 Example Autonomous State Configuration 0x02 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
ID	2	0x02	0000 0010
Precounter	5	0x05	0000 0101
Postcounter	1	0x0001	0000 0000 0000 0001
Timeout	8	0x08	0001 0000
Meas_1.Sensor	0	0x00	0000 0000

(table continues...)

Table 31 (continued) Example Autonomous State Configuration 0x02 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
Meas_1.Postcounter	5	0x05	0000 0101
Meas_1.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Config	246	0xF6	1111 0110
Meas_1.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Trans_Upper	99	0x63	0110 0011
Meas_1.Threshold_1.Thrld_Upper	4	0x0004	0000 0000 0000 0100
Meas_1.Threshold_2.Config	2	0x02	0000 0010
Meas_1.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Upper	3200	0x0C80	0000 1100 1000 0000
Meas_2.Sensor	1	0x01	0000 0001
Meas_2.Postcounter	2	0x02	0000 0002
Meas_2.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Config	11	0x0B	0000 1011
Meas_2.Threshold_1.Trans_Lower	72	0x48	0100 1000
Meas_2.Threshold_1.Thrld_Lower	-80	0xFFB0	1111 1111 1011 0000
Meas_2.Threshold_1.Trans_Upper	72	0x48	0100 1000
Meas_2.Threshold_1.Thrld_Upper	80	0x0050	0000 0000 0011 0010
Meas_2.Threshold_2.Config	0	0x00	0000 0000
Meas_2.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000

The *Precounter* and *Postcounter* as well as the *Meas_1.Postcounter* are determined in such a way that the product *Precounter* × *Postcounter* is the greatest common divisor of the measurement intervals for pressure and acceleration measurement: For an acceleration measurement interval of 10 ms and a pressure measurement interval of 25 ms, the greatest common divisor is 5 ms. Setting *Precounter* and *Postcounter* such that the product *Precounter* × *Postcounter* is the greatest common divisor of both measurement intervals minimizes the values for *Meas_1.Postcounter* and *Meas_2.Postcounter* leading to the smallest average charge consumption.
The value for *Meas_1.Postcounter* is then determined by dividing the pressure measurement interval of 25 ms by the product *Precounter* × *Postcounter*, i.e. *Meas_1.Postcounter* = 25 ms / 5 ms = 5.
Similarly, *Meas_2.Postcounter* is calculated by dividing the acceleration measurement interval of 40 ms by the product *Precounter* × *Postcounter*, i.e. *Meas_2.Postcounter* = 10 ms / 5 ms = 2.

The configuration has its time-out enabled through parameter *Timeout*. The time-out is provided in powers of 2 and proportional to the Autonomous State Timer period of 5 ms. Therefore, *Timeout* = 8 results in a time-out of $2^8 \times 5$ ms = 1.28 s. Since configuration 0x02 is entered from configuration 0x01 in this example, the time-out causes KP497 to return to configuration 0x01, if no threshold is violated after 1.28 s

With *Meas_1.Sensor* set to select the pressure measurement (*Meas_1.Sensor* = 0) and *Meas_2.Sensor* set to select the acceleration measurement (*Meas_2.Sensor* = 1), pressure and acceleration are monitored in parallel. Since the pressure measurement is selected as first measurement via *Meas_1*, the pressure measurement has higher priority. That means that whenever the postcounters for pressure and acceleration measurement elapse at the same time (in this case every 50 ms), the pressure measurement is taken first and if one of the pressure thresholds is violated, the acceleration measurement is not performed any more.

With *Meas_1.Threshold_1.Config* set to 0xF6, the upper threshold is enabled (bit 1 set) and the lower threshold is disabled (bit 0 unset). Moreover, since bit field 3:2 is set to 1, the rate of change threshold is selected. With the value 15 in bit field 7:4 of *Meas_1.Threshold_1.Config* the number of measurements over which the rate of change is determined is set to 16. Since only the upper threshold is enabled, the parameters *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Thrld_Lower* are ignored by the device but must still be provided.

In order to cause the device to remain in Autonomous State but change to the Autonomous State Configuration with ID 0x03 when the upper rate of change threshold is exceeded, bit field 7:6 of *Meas_1.Threshold_1.Trans_Upper* is set to 1 while bits 4:0 are set to the ID of the Autonomous State Configuration 0x03. Since bit 5 of *Meas_1.Threshold_1.Trans_Upper* is 1, the device pulls the ALERT pin high when the upper rate of change threshold is exceeded and pulls the ALERT pin low again when the first measurement in configuration 0x03 is performed.

For the rate of change threshold, the threshold is provided with a resolution of 1/16 kPa/LSB per measurement interval. Therefore, with the measurement interval of 25 ms, setting the upper threshold through parameter *Meas_1.Threshold_1.Thrld_Upper* to 4, the resulting upper rate of change threshold is 4 LSB × 1/16 kPa/LSB / 25 ms = 10 kPa/s.

The second threshold configuration *Meas_1.Threshold_2* enables the absolute upper threshold through the value *Meas_1.Threshold_2.Config* = 0x02. As the second threshold, the absolute upper threshold is for each measurement only evaluated, if the rate of change threshold is not violated. Since bit 0 of *Meas_1.Threshold_2.Config* is 0, the lower absolute threshold is disabled and the parameters *Meas_1.Threshold_2.Trans_Lower* and *Meas_1.Threshold_2.Thrld_Lower* are ignored by the device but must still be provided. Bits 7:4 of *Meas_1.Threshold_2.Config* are ignored for the absolute threshold as well.

In order to cause the device to transition to On State when the upper absolute threshold is exceeded, bits 7:6 of *Meas_1.Threshold_2.Trans_Upper* are set to 0. The remaining bits of *Meas_1.Threshold_2.Trans_Upper* are ignored.

The upper absolute threshold is set through parameter *Meas_1.Threshold_1.Thrld_Upper* to 3200 LSB × 1/16 kPa/LSB = 200 kPa.

With *Meas_2.Threshold_1.Config* set to 0x0B, both the lower and upper thresholds are enabled (bits 0 and 1 both set). Moreover, since bits 3:2 are set to 2, the acceleration threshold is relative to On State, i.e. relative to the first acceleration measurement taken at the transition from On State to Autonomous State. Bits 7:4 of *Meas_2.Threshold_1.Config* are ignored due to the selection of the relative threshold.

In order to cause the device to remain in Autonomous State but change to the Autonomous State Configuration with ID 0x08 when either the upper or lower threshold is violated both *Meas_2.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Trans_Upper* are set to 0x48, which provides the ID 0x08 in bits 4:0 and specifies the transition to another Autonomous State configuration through the value 2 in bits 7:6. The intermediate alert is disabled for both transitions, because bit 5 is set to 0.

The relative thresholds *Meas_2.Threshold_1.Lower* and *Meas_2.Threshold_1.Upper* have a resolution of 1/16 g/LSB. That means the values -80 for *Meas_2.Threshold_1.Lower* and 80 for *Meas_2.Threshold_1.Upper* sets the lower threshold to -80 LSB × 16 g/LSB = -5 g and the upper threshold to +5 g.

The resulting command on the serial interface to store this Autonomous State Configuration including command byte and CRC in hexadecimal is:

```
0xD4 0x02 0x05 0x00 0x01 0x08 0x00 0x05 0x00 0x00 0xF6 0x00 0x00 0x00 0x63 0x00 0x04 0x02 0x00
0x00 0x00 0x00 0x0C 0x80 0x01 0x02 0x00 0x00 0x0B 0x48 0xFF 0xB0 0x48 0x00 0x50 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x0F 0x53
```

4.2.3 Autonomous State Configuration for third stage pressure monitoring (0x03)

The third stage of the pressure monitoring of this example is implemented by the Autonomous State Configuration with ID 0x03. The parameters used in the Save Autonomous State Configuration command to store the Autonomous State Configuration with ID 0x03 are summarized in the following table:

Table 32 Example Autonomous State Configuration 0x03 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	**Hexadecimal**	**Binary**
ID	3	0x03	0000 0011
Precounter	1	0x01	0000 0001
Postcounter	0	0x0000	0000 0000 0000 0000
Timeout	7	0x07	0000 0111
Meas_1.Sensor	0	0x00	0000 0000
Meas_1.Postcounter	5	0x05	0000 0101
Meas_1.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Config	2	0x02	0000 0010
Meas_1.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Upper	3200	0x0C80	0000 1100 1000 0000
Meas_1.Threshold_2.Config	0	0x00	0000 0000
Meas_1.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000
Meas_2.Sensor	255	0xFF	1111 1111
Meas_2.Postcounter	1	0x01	0000 0001
Meas_2.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Config	0	0x00	0000 0000

(table continues...)

Table 32 (continued) Example Autonomous State Configuration 0x03 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
Meas_2.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Upper	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Config	0	0x00	0000 0000
Meas_2.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000

Since the *Postcounter* is set to 0, in Autonomous State Configuration 0x03 a new measurement is performed and evaluated for its thresholds as soon as all measurements and threshold evalutions from the preceeding measurement interval are finished. In this case, the *Precounter* is ignored. Moreover, the measurement postcounter *Meas_1.Postcounter* is set to 1 in order to achieve the fastest possible measurement rate.

The time-out in this case is also given in number of measurements instead of a time interval: With *Timeout* set to 7, the device returns to configuration 0x02 after of 2^7 = 128 measurements, if no threshold is violated.

Meas_1.Sensor = 0 enables the pressure measurement, while *Meas_2.Sensor* = 0xFF disables the second measurement. Therefore, only pressure is monitored in this configuration.

The first threshold configuration *Meas_1.Threshold_1* enables the absolute upper threshold through the value *Meas_1.Threshold_1.Config* = 0x02. Since bit 0 of *Meas_1.Threshold_1.Config* is 0, the lower absolute threshold is disabled and the parameters *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Thrld_Lower* are ignored by the device but must still be provided. Bits 7:4 of *Meas_1.Threshold_1.Config* are ignored for the absolute threshold as well.

In order to cause the device to transition to On State when the upper absolute threshold is exceeded, bits 7:6 of *Meas_1.Threshold_1.Trans_Upper* are set to 0. The remaining bits of *Meas_1.Threshold_2.Trans_Upper* are ignored.

The upper absolute threshold is set through parameter *Meas_1.Threshold_1.Thrld_Upper* to 3200 LSB × 1/16 kPa/LSB = 200 kPa.

***Note**: Even though Meas_2.Sensor is set to 0xFF and therefore the second measurement is disabled, Meas_2.Postcounter is set to 1. This is in order to avoid that bit 0 is set in the Status of the Save Autonomous State command.*

The resulting command on the serial interface to store this Autonomous State Configuration including command byte and CRC in hexadecimal is:

```
0xD4 0x03 0x01 0x00 0x00 0x07 0x00 0x01 0x00 0x00 0x02 0x00 0x00 0x00 0x00 0x0C 0x80 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0xFF 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x6F 0x11
```

4.2.4 Autonomous State Configuration for second stage acceleration monitoring (0x08)

A violation of one of the acceleration thresholds in the Autonomous State Configurations 0x01 or 0x02 causes the device to transition to the Autonomous State Configuration with ID 0x08. The parameters used in the Save Autonomous State Configuration command to store the Autonomous State Configuration with ID 0x08 are summarized in the following table:

Table 33 Example Autonomous State Configuration 0x08 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
ID	8	0x08	0000 0011
Precounter	1	0x01	0000 0001
Postcounter	1	0x0001	0000 0000 0000 0001
Timeout	6	0x06	0000 0110
Meas_1.Sensor	1	0x01	0000 0001
Meas_1.Postcounter	0	0x00	0000 0001
Meas_1.Rec_Samples	10,000	0x03E8	0000 0011 1110 1000
Meas_1.Threshold_1.Config	15	0x0F	0000 1111
Meas_1.Threshold_1.Trans_Lower	128	0x80	1000 0000
Meas_1.Threshold_1.Thrld_Lower	-80	FFB0	1111 1111 1011 0000
Meas_1.Threshold_1.Trans_Upper	128	0x80	1000 0000
Meas_1.Threshold_1.Thrld_Upper	80	0x0050	0101 0000
Meas_1.Threshold_2.Config	0	0x00	0000 0000
Meas_1.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000
Meas_2.Sensor	255	0xFF	1111 1111
Meas_2.Postcounter	1	0x01	0000 0001
Meas_2.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Config	0	0x00	0000 0000
Meas_2.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Upper	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Config	0	0x00	0000 0000
Meas_2.Threshold_2.Trans_Lower	0	0x00	0000 0000

(table continues...)

Table 33 (continued) Example Autonomous State Configuration 0x08 for example 2: three stage pressure and acceleration monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
Meas_2.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000

In order to set the measurement interval to 1 ms, the *Precounter*, *Postcounter* and *Meas_1.Postcounter* are all set to 1.

The configuration also has its timeout set through the *Timeout* parameter to $2^6 \times 1$ ms = 64 ms. When the time-out occurs without one of the thresholds of this configuration being violated first, the device transitions back to configuration 0x01 or 0x02, depending on which of these two configurations was executed last before entering configuration 0x08.

Meas_1.Sensor = 1 enables the acceleration measurement, while *Meas_2.Sensor* = 0xFF disables the second measurement. Therefore, only acceleration is monitored in this configuration.

With *Meas_1.Threshold_1.Config* set to 0x0F, both the lower and upper threshold are enabled (both bit 0 and bit 1 set). Moreover, since bit field 3:2 is set to 3, the threshold relative to configuration start is selected, i.e. the thresholds are the change of acceleration since the transition from configuration 0x01 or 0x02 to configuration 0x08. Since the relative threshold is selected, the value in bits 7:4 of *Meas_1.Threshold_1.Config* is ignored.

Bit field 7:5 of both *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Trans_Upper* is set to 2 causing the device to transition to acceleration recording. Due to the selection for the transition to acceleration recording, all other bits of *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Trans_Upper* are ignored. If acceleration recording is entered from this configuration due to a violation of the first threshold, the measurement interval during acceleration recording is 1 ms. Moreover, the device stays in acceleration recording until 10,000 acceleration measurements are performed as specified by parameter *Meas_1.Rec_Samples*.

The relative thresholds *Meas_1.Threshold_1.Lower* and *Meas_1.Threshold_1.Upper* have a resolution of 1/16 g/LSB. That means the values -80 for *Meas_1.Threshold_1.Lower* and 80 for *Meas_1.Threshold_1.Upper* sets the lower threshold to -80 LSB × 16 g/LSB = -5 g and the upper threshold to +5 g.

***Note**: Even though Meas_2.Sensor is set to 0xFF and therefore the second measurement is disabled, Meas_2.Postcounter is set to 1. This is in order to avoid that bit 0 is set in the Status of the Save Autonomous State command.*

The resulting command on the serial interface to store this Autonomous State Configuration including command byte and CRC in hexadecimal is:

```
0xD4 0x08 0x01 0x00 0x01 0x06 0x01 0x01 0x03 0xE8 0x0F 0x80 0xFF 0xB0 0x80 0x00 0x50 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0xFF 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0xF3 0x75
```

4.3 Example 3: Cyclic pressure monitoring with intermediate alerts

This example considers a two stage pressure and acceleration monitoring where acceleration is monitored only in the first stage and pressure in both stages. Additionally, the second stage has a relative and an absolute threshold. The relative threshold in the second stage causes the device to either transition back to the first stage without intermediate alert or to transition to the second stage with an intermediate alert. When the absolute threshold in the second stage is exceeded, the device transitions to On State.

In this example, the Autonomous State Configuration for the first stage is assigned the ID 0 and the one for the second stage is assigned the ID 5. To store the Autonomous State Configurations in the device, the Save Autonomous State Configuration command is executed once for each of the Autonomous State Configurations. The sections Chapter 4.3.1 and Chapter 4.3.2 describe the parameters and the resulting command on the serial interface to store the Autonomous State Configuration for each configuration separately.

Once the Autonomous State Configurations are stored successfully, the cyclic monitoring described in this section can be started using the Enter Autonomous State command. The corresponding command including its command byte and CRC in hexadecimal values is:

```
0xC1 0x00 0x38 0x6A
```

4.3.1 Autonomous State Configuration for first stage of cyclic pressure monitoring (0x00)

The Autonomous State Configuration used as initial configuration in the three cyclic monitoring in this example has the ID 0x00. The parameters used in the Save Autonomous State Configuration command to store the Autonomous State Configuration with ID 0x00 are summarized in the following table:

Table 34 Example Autonomous State Configuration 0x00 for example 3: cyclic pressure monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
ID	0	0x00	0000 0000
Precounter	10	0x0A	0000 1010
Postcounter	1	0x0001	0000 0000 0000 0001
Timeout	0	0x00	0000 0000
Meas_1.Sensor	0	0x00	0000 0000
Meas_1.Postcounter	2	0x02	0000 0010
Meas_1.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Config	14	0x0E	0000 1110
Meas_1.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Trans_Upper	97	0x61	0110 0001
Meas_1.Threshold_1.Thrld_Upper	160	0x00A0	0000 0000 1010 0000
Meas_1.Threshold_2.Config	2	0x02	0000 0010
Meas_1.Threshold_2.Trans_Lower	0	0x00	0000 0000

(table continues...)

Table 34 (continued) Example Autonomous State Configuration 0x00 for example 3: cyclic pressure monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
Meas_1.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_1.Threshold_2.Thrld_Upper	2400	0x0960	0000 1001 0110 0000
Meas_2.Sensor	1	0x01	0000 0001
Meas_2.Postcounter	1	0x01	0000 0001
Meas_2.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Config	11	0x0B	0000 1011
Meas_2.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Lower	-100	0xFF9C	1111 1111 1001 1100
Meas_2.Threshold_1.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Upper	100	0x0064	0000 0000 0110 0100
Meas_2.Threshold_2.Config	0	0x00	0000 0000
Meas_2.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000

In this example, the pressure measurement interval is set to *Precounter* × *Postcounter* × *Meas_1.Postcounter* = 10 ms × 1 × 2 = 20 ms. Similarly, the acceleration measurement interval is set to *Precounter* × *Postcounter* × *Meas_1.Postcounter* = 10 ms. The *Precounter* is set to 10 ms, such that it is the greatest common divisor of the pressure and acceleration measurement intervals.

The time-out is in this example disabled by setting *Timeout* to 0.

Since the configuration monitors both pressure and acceleration and the pressure priority is higher than the acceleration priority, *Meas_1.Sensor* is set to select the pressure measurement (*Meas_1.Sensor* = 0) and *Meas_2.Sensor* is set to select the acceleration measurement (*Meas_2.Sensor* = 1). This way, the acceleration measurement in each interval is performed, if either the pressure measurement postcounter did not elapse or if none of the pressure thresholds were violated.

With *Meas_1.Threshold_1.Config* set to 0x0E, the upper threshold is enabled (bit 1 set) and the lower threshold is disabled (bit 0 unset). Moreover, since the bit field 3:2 is set to 3, the threshold relative to configuration start is selected. Therefore, every time configuration 0x00 of this example is entered, no matter whether from On State or from configuration 0x05, a new reference pressure measurement is taken and the thresholds specify the allowed change compared to the most recent reference pressure measurement. Bits 7:4 are ignored due to the selection of the relative threshold. Since only the upper threshold is enabled, the parameters *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Thrld_Lower* are ignored by the device but must still be provided.

In order to cause the device to remain in Autonomous State but change to the Autonomous State Configuration with ID 0x05 when the upper relative threshold is exceeded, bit field 7:6 of *Meas_1.Threshold_1.Trans_Upper* is set to 1. The ID 0x05 of the Autonomous State Configuration to change to when the threshold is violated is set in bit field 4:0 of *Meas_1.Threshold_1.Trans_Upper*. Since bit 5 of *Meas_1.Threshold_1.Trans_Upper* is 1, an intermediate alert is generated when the upper pressure threshold is violated. For this intermediate alert, the ALERT pin is pulled high when the upper relative pressure threshold is exceeded and is pulled low again when the first measurement in configuration 0x05 takes place.

The upper pressure threshold is set with a resolution of 1/16 kPa/LSB in *Meas_1.Threshold_1.Thrld_Upper*: The value of 160 sets an upper relative pressure threshold of 160 LSB × 1/16 LSB/kPa = 10 kPa.

The second threshold configuration *Meas_1.Threshold_2* enables the absolute upper threshold through the value *Meas_1.Threshold_2.Config* = 0x02. As the second threshold, the absolute upper threshold is for each measurement only evaluated, if the rate of change threshold is not violated. Since bit 0 of *Meas_1.Threshold_2.Config* is 0, the lower absolute threshold is disabled and the parameters *Meas_1.Threshold_2.Trans_Lower* and *Meas_1.Threshold_2.Thrld_Lower* are ignored by the device but must still be provided. Bits 7:4 of *Meas_1.Threshold_2.Config* are ignored for the absolute threshold as well.

In order to cause the device to transition to On State when the upper absolute threshold is exceeded, bits 7:6 of *Meas_1.Threshold_2.Trans_Upper* are set to 0. The remaining bits of are *Meas_1.Threshold_2.Trans_Upper* ignored.

The upper absolute threshold is set through parameter *Meas_1.Threshold_2.Thrld_Upper* to 2400 LSB × 1/16 kPa/LSB = 150 kPa.

The upper absolute threshold is set through parameter *Meas_1.Threshold_1.Thrld_Upper* to 3200 LSB × 1/16 kPa/LSB = 200 kPa.

With *Meas_2.Threshold_1.Config* set to 0x0B, both the lower and upper thresholds are enabled (bits 0 and 1 both set). Moreover, since bits 3:2 are set to 2, the acceleration threshold is relative to On State, i.e. relative to the first acceleration measurement taken at the transition from On State to Autonomous State. Bits 7:4 of *Meas_2.Threshold_1.Config* is ignored due to the selection of the relative threshold.

If either the lower or upper acceleration threshold is violated, the device transitions to On State as set by *Meas_2.Threshold_1.Trans_Lower* = 0 and *Meas_2.Threshold_1.Trans_Upper* = 0.

The relative thresholds *Meas_2.Threshold_1.Lower* and *Meas_2.Threshold_1.Upper* have a resolution of 1/16 g/LSB. That means the values -100 for *Meas_2.Threshold_1.Lower* and 100 for *Meas_2.Threshold_1.Upper* sets the lower threshold to -100 LSB × 16 g/LSB = -6.25 g and the upper threshold to +6.25 g.

The resulting command on the serial interface to store this Autonomous State Configuration including command byte and CRC in hexadecimal is:

```
0xD4 0x00 0x0A 0x00 0x01 0x00 0x00 0x02 0x00 0x00 0x0E 0x00 0x00 0x00 0x65 0x00 0xA0 0x02 0x00
0x00 0x00 0x00 0x09 0x60 0x01 0x01 0x00 0x00 0x0B 0x00 0xFF 0x9C 0x00 0x00 0x64 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0xF1 0xE5
```

4.3.2 Autonomous State Configuration for second stage of cyclic pressure monitoring (0x05)

The Autonomous State Configuration used as second configuration in the cyclic monitoring in this example has the ID 0x05. The parameters used in the Save Autonomous State Configuration command to store the Autonomous State Configuration with ID 0x05 are summarized in the following table:

Table 35 Example Autonomous State Configuration 0x05 for example 3: cyclic pressure monitoring

Parameter	Value		
	Decimal	**Hexadecimal**	**Binary**
ID	5	0x05	0000 0101
Precounter	20	0x14	0001 0100
Postcounter	1	0x0001	0000 0000 0000 0001
Timeout	0	0x00	0000 0000
Meas_1.Sensor	0	0x00	0000 0000

(table continues...)

Table 35 (continued) Example Autonomous State Configuration 0x05 for example 3: cyclic pressure monitoring

Parameter	Value		
	Decimal	Hexadecimal	Binary
Meas_1.Postcounter	1	0x02	0000 0010
Meas_1.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Config	2	0x02	0000 00100
Meas_1.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_1.Threshold_1.Trans_Upper	00	0x00	0000 0000
Meas_1.Threshold_1.Thrld_Upper	2400	0x0960	0000 1001 0110 0000
Meas_1.Threshold_2.Config	15	0x0F	0000 1111
Meas_1.Threshold_2.Trans_Lower	64	0x40	0100 0000
Meas_1.Threshold_2.Thrld_Lower	-160	0xFF60	1111 1111 0110 0000
Meas_1.Threshold_2.Trans_Upper	101	0x65	0110 0101
Meas_1.Threshold_2.Thrld_Upper	160	0x00A0	0000 0000 1010 0000
Meas_2.Sensor	255	0xFF	1111 1111
Meas_2.Postcounter	1	0x01	0000 0001
Meas_2.Rec_Samples	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Config	0	0x00	0000 0000
Meas_2.Threshold_1.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_1.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_1.Thrld_Upper	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Config	0	0x00	0000 0000
Meas_2.Threshold_2.Trans_Lower	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Lower	0	0x0000	0000 0000 0000 0000
Meas_2.Threshold_2.Trans_Upper	0	0x00	0000 0000
Meas_2.Threshold_2.Thrld_Upper	0	0x0000	0000 0000 0000 0000

The pressure measurement interval in this configuration is set to *Precounter* × *Postcounter* × *Meas_1.Postcounter* = 20 ms.

The configuration enables the pressure measurement through *Meas_1.Sensor* = 0 and disables the second measurement via *Meas_2.Sensor* = 0xFF. Since the second measurement is enabled, all further parameters for *Meas_2* are ignored, but must still be provided in the command to save the Autonomous State Configuration.

The first threshold configuration *Meas_1.Threshold_1* enables the absolute upper threshold through the value *Meas_1.Threshold_1.Config* = 0x02. Since bit 0 of *Meas_1.Threshold_1.Config* is 0, the lower absolute threshold is disabled and the parameters *Meas_1.Threshold_1.Trans_Lower* and *Meas_1.Threshold_1.Thrld_Lower* are ignored by the device but must still be provided. Bits 7:4 of *Meas_1.Threshold_1.Config* are ignored for the absolute threshold as well.

In order to cause the device to transition to On State when the upper absolute threshold is exceeded, bits 7:6 of *Meas_1.Threshold_1.Trans_Upper* are set to 0. The remaining bits of are *Meas_1.Threshold_1.Trans_Upper* ignored.

The upper absolute threshold is set through parameter *Meas_1.Threshold_1.Thrld_Upper* to 2400 LSB × 1/16 kPa/LSB = 150 kPa.

With *Meas_1.Threshold_2.Config* set to 0x0F, both the lower and upper secondary thresholds are enabled (bits 0 and 1 both set). The secondary threshold is evaluated in each measurement interval only, if the first threshold was not violated. Moreover, since bits 3:2 are set to 3, the relative pressure threshold is relative to the start of Autonomous State Configuration 0x05, i.e. relative to the pressure measurement taken at the transition from Autonomous State Configuration 0x00 to Autonomus State Configuration 0x05 or the pressure taken when the last time the upper relative threshold of configuraiton 0x05 is exceeded. Bits 7:4 of *Meas_2.Threshold_1.Config* is ignored due to the selection of the relative threshold.

Both the lower and upper thresholds are configured to remain in Autonomous State when the threshold is violated by setting bit field 7:6 of both *Meas_1.Threshold_2.Trans_Lower* and *Meas_1.Threshold_2.Trans_Upper* to 1. However, while the upper threhsold is configured to stay in Autonomous State configuration 0x05 with an update of the reference pressure value for the relative threshold by setting bit field 4:0 of *Meas_1.Threshold_2_Trans_Upper* to 0, the lower threshold changes back to Autonomous State Configuration 0x00, because bits 4:0 of *Meas_1.Threshold_2.Trans_Lower* are set to 0. The intermediate alert is enabled for the upper threshold (bit 5 of *Meas_1.Threshold_2.Trans_Upper* set to 1) and disabled for the lower threshold (bit 5 of *Meas_1.Threshold_2.Trans_Upper* set to 0).

The relative thresholds *Meas_2.Threshold_1.Lower* and *Meas_2.Threshold_1.Upper* have a resolution of 1/16 g/LSB. That means the values -160 for *Meas_2.Threshold_1.Lower* and 160 for *Meas_2.Threshold_1.Upper* sets the lower threshold to -160 LSB × 16 g/LSB = -10 g and the upper threshold to +10 g.

***Note**: Even though Meas_2.Sensor is set to 0xFF and therefore the second measurement is disabled, Meas_2.Postcounter is set to 1. This is in order to avoid that bit 0 is set in the Status of the Save Autonomous State command.*

The resulting command on the serial interface to store this Autonomous State Configuration including command byte and CRC in hexadecimal is:

```
0xD4 0x05 0x14 0x00 0x01 0x00 0x00 0x01 0x00 0x00 0x02 0x00 0x00 0x00 0x00 0x09 0x60 0x0F 0x40
0xFF 0x60 0x65 0x00 0xA0 0xFF 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0xAE 0x2D
```

5 Revision history

Table 36 Revision History

Document version	Date of release	Description of changes
1.00	2025-12-22	Initial release

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-12-22

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG

All Rights Reserved.

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-rdt1756121379345

Important notice

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.