SFC5xxx Mass Flow Controller

SHDLC Communication Interface Reference

Summary

This document describes the UART/RS485 communication with the Sensirion SFC5xxx Mass Flow Controller series, using the SHDLC (Sensirion-HDLC) protocol.

RECENT CHANGES ON THIS DOCUMENT

Date	Version	Author	Why
8. Oct 12	1.0	SWE	Initial Version
12. Sept 13	1.1	RFU	Several minor corrections in the text.
12. Sept 14	1.2	RFU	Added 5.6 Advanced Measurements (0x30)
5. Nov 14	1.3	RFU	Added new sub-commands in 5.7.3 Get Current Calibration Information (0x44)
11. Nov 14	1.4	RFU	Added new commands: 5.3.1 Set/Get Valve Input Source Configuration 5.4.1 Set/Get Medium Unit Configuration 5.5.1 Set/Get Controller Configuration
24. Nov 14	1.5	UBR	Added new commands: 5.2.4 Read Measured Flow (2 Sensors) (0x0A) 5.2.6 Set Setpoint and read Measured Flow (2 Sensors) (0x04) Added new error codes: 65 – 68
5. Dec 14	1.6	UBR	Added missing MISO data for sub-command “Read fullscale flow (0x14)” in 5.7.3 Get Current Calibration Information (0x44)
12. Dec 14	1.7	UBR	5.7.2 Get Calibration Information (0x40): Added new sub-commands 0x16, 0x17 and 0x18 Added minimum required firmware version for each command
22. Jul 15	1.8	UBR	5.6 Advanced Measurements (0x30): added temperature compensated raw thermal conductivity measurement
29. Oct 15	1.9	RFU	Correction: 5.4.1 Set/Get Medium Unit Configuration (0x21): Byte[0] on “get converted fullscale” was 0x02 instead 0x0A

1 Table Of Contents

2 INTRODUCTION

This document contains information for implementing the SHDLC interface, which is used to communicate with the SFC5xxx Mass Flow Controller series. The document is divided into two sections:

1. **SHDLC protocol definition**
   Defines the protocol and frame composition used on UART/RS485 for communication.

2. **SFC5xxx command reference**
   Lists the available commands for the SFC5xxx which are transferred using the SHDLC protocol.

For communication between a PC and a SFC5xxx device, there are drivers (native and .NET) available which implement the SHDLC interface and the command set. Please contact Sensirion for further information.

3 DEFINITIONS AND ABBREVIATIONS

Device	The Massflow Controller to communicate with.
LSb	Least significant bit
LSB	Least significant byte
MFC	Mass Flow Controller
MISO	Master In Slave Out. Frame direction from slave (device) to master.
MOSI	Master Out Slave In. Frame direction from master to slave (device).
MSb	Most significant bit
MSB	Most significant byte
NV memory	Non Volatile memory: a memory which keeps its data content also after soft and power reset.
SFC5xxx	Sensirion Mass Flow Controller. The '5' defines the generation of the controller, the following 3 letters "xxx" define the body, interface and some other parameters.
SHDLC	The name of the protocol, which is used to transfer data between the master and the MFC. SHDLC is the abbreviation for Sensirion High-Level Data Link Control and is related to the ISO HDLC protocol.
UART	Universal Asynchronous Receiver Tansmitter. Defines the byte transfer between the master and device. The SHDLC protocol is based on the UART byte transfer.

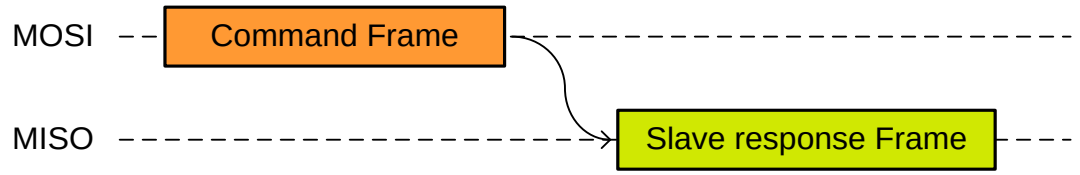
4 SHDLC PROTOCOL DEFINITION

SHDLC is a data link protocol, which is based on the UART byte transfer. It defines the data frames which are transferred from master to slave and vice versa.

The main features of the SHDLC protocol are:

- Master/Slave protocol
- Addressable (1 master and 1…255 slaves on the bus)
- Supports broadcasting
- Up to 255 bytes of data within one data frame (read and write)
- Half-duplex (no transmit and receive at the same time). This allows the usage of 2-wire RS485.
- Based on byte transfers (UART protocol used)
- Selectable baudrate

In SHDLC communication, every transfer is initiated by the master with a MOSI frame. The slave will response every received frame with a MISO frame as shown in the following picture:



4.1 BYTE TRANSMISSION

The SHDLC is a byte-orientated protocol which uses UART for the byte transfer. Use the following UART settings:

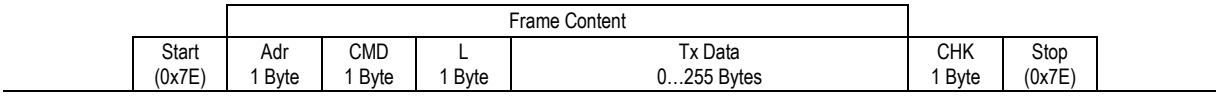
- Baudrate: 115200 baud as factory default, but can be reconfigured by command.
- 8 Data bits (LSb first)
- No parity
- 1 Stop Bit

4.2 FRAME DEFINITION

In the following, the composition of the frame body is shown. This body is used for every transfer between master and slave.

MOSI Frame

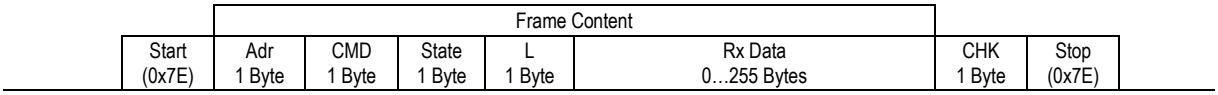
The graphic shows the data flow in relation to time for a MOSI frame (master → slave):



Start/Stop	Unique character (0x7E) which defines the begin and end of a frame
Adr	Address of the slave device
CMD	Command byte which tells the device what to do with the given data
L	Length of the “Tx Data” field
Tx Data	Data to transmit to the device
CHK	Checksum of the “Frame Content”

MISO Frame

The following diagram shows the data flow in relation to time for a MISO frame (slave → master):



Start/Stop	Unique character (0x7E) which defines the begin and end of a frame
Adr	Address of the slave device (the device will send it’s own address)
CMD	Command byte which was received in the MOSI frame
State	Device state information and command execution error state
L	Length of the “Rx Data” field
Rx Data	Data to transmit to the master
CHK	Checksum of the “Frame Content”

4.2.1 Frame Start and Stop (and Byte Stuffing)

Because there is not hardware handshaking, the frame start and stop is signalized by a unique byte:

- Start: 0x7E (01111110b)
- Stop: 0x7E (01111110b)

If this byte (0x7E) occurs anywhere else in the frame, it will be replaced by another two bytes (byte stuffing: first send 0x7D, than the original data byte with bit 5 inverted → 0x5E). This will also be done for Escape (0x7D), XON (0x11) and XOFF (0x13) bytes:

Original data byte	Transferred data bytes
0x7E	0x7D, 0x5E
0x7D	0x7D, 0x5D
0x11	0x7D, 0x31
0x13	0x7D, 0x33

4.2.2 Address Field

The address field in the MOSI frame (1 Byte) defines the receiver of the frame (slave device address). The address range is defined as follows:

- 0…254 slave addresses
- 255 broadcast address

In a MISO frame the address field contains the slave address (sender address).

4.2.3 Command

Typically (in a MOSI frame), this field contains the application command which defines for the specific application what to do with the given data. There are some reserved commands which are used for special frame transfers (see Chapter “Transfer Types” on page 12). In the MISO frame the slave will return the received command in this field.

The following Table shows the command space:

Command ID (Hex)	Size	Usage
0x00 … 0x7F	128	*Individual device command space* Commands which are defined individual for every SHDLC device
0x80 … 0xCF	80	*Device command pattern* Common commands, if available they are implemented similar.
0xD0 … 0xEF	32	*SHDLC common command space* Commands which operate with every SHDLC device (not only MFC)
0xF0 … 0xFF	16	*Special frame identifiers space* (Chapter “Transfer Types” on page 12). With this identifiers, some special transfers can be marked.

The size of the command is 1 byte.

4.2.4 LENGTH

The length byte defines the number of transferred bytes in data field (Rx or Tx). It is the length of the data field before byte stuffing, not the number of bytes which are transferred over the bus.

Example: The sender will transmit data [0xA7, 0xB4, 0x7E, 0x24]. Because of byte stuffing, it needs to transmit the stream [0xA7, 0xB4, 0x7D, 0x5E, 0x24]. The transmitted size information in this case is 0x04.

The size of the length information is 1 byte. This allows to transfer 0…255 bytes data.

4.2.5 STATE

The MISO frame contains a state byte, which allows the master to detect communication and execution errors. An additional error flag signalizes that the device is in an error state.

The following shows the composition of the Status byte:

b7	b6						b0
Device Error Flag	Execution error code						

Execution Error Code

The execution error code signalizes all errors which occur while processing the frame or executing the command. The following table shows the error code mapping:

Error Code	Error Type
0x00	No Error.
0x01 … 0x1F	Common error codes (same codes for all SHDLC devices)
0x20 … 0x7F	Device specific error codes

For a detailed list of all error codes, refer to chapter “Error Codes” on page 48.

Device Error Flag

This flag notifies the master that an error occurred on the device during operation. If this flag is set, the master can read the device error state with the “Get Device Error State (0xD2)” command. For example a supply under voltage condition can cause the setting of the error flag.

4.2.6 DATA

The data has a usable size of [0…255] bytes (original data, before byte stuffing). The meaning of the data content depends on the command.

4.2.7 CHECKSUM

The checksum is built before byte stuffing and checked after removing stuffed bytes from the frame. The checksum defines as follows:

1. Sum all bytes between start and stop (without start and stop bytes)
2. Take the LSB of the result and invert it. This will be the checksum.

For a MOSI frame use Address, Command, Length and Data to calculate the checksum.
For a MISO frame use Address, Command, State, Length and Data to calculate the checksum.

Example (MOSI frame without start/stop and without byte stuffing):

Adr	CMD	L	Tx Data 4 Bytes	CHK
0x02	0x43	0x04	0x64, 0xA0, 0x22, 0xFC	0x94

The checksum calculates as follows:

Adr	0x02
CMD	0x43
L	0x04
Data 0	0x64
Data 1	0xA0
Data 2	0x22
Data 3	0xFC
Sum	0x26B
LSB of Sum	0x6B
Inverted (=Checksum)	**0x94**

4.3 PROTOCOL DEFINITION

This chapter describes the frame communication protocol with SHDLC. There are some basic rules:

1. On every master request (MOSI frame), the addressed slave will respond with a slave response (MISO frame). There are two exclusions where the slave should not send a response:
   - If the checksum of a MOSI frame does not match
   - If the MOSI frame was a broadcast
2. Between receiving a MOSI frame and sending slave response, the slave will not accept any other frame from master. In case of a broadcast, the master has to wait the specified command execution time.

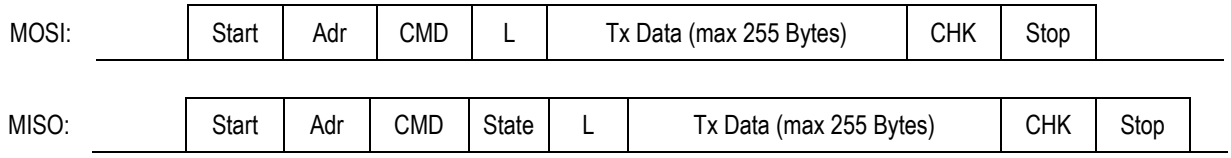
4.3.1 TRANSFER TYPES

By default, the master sends a standard frame which contains up to 255 bytes Tx data. This is called a standard frame transfer. Additionally there are some special frame transfers defined. They are marked with a special frame identifier in the CMD field of the frame. The following chapters describe the different transfer types

Standard Transfer

In this transfer, the Master initiates a transfer with a MOSI frame containing command and up to 255 bytes of data. After executing the command, the slave will respond with a MISO frame containing state and up to 255 bytes of data.

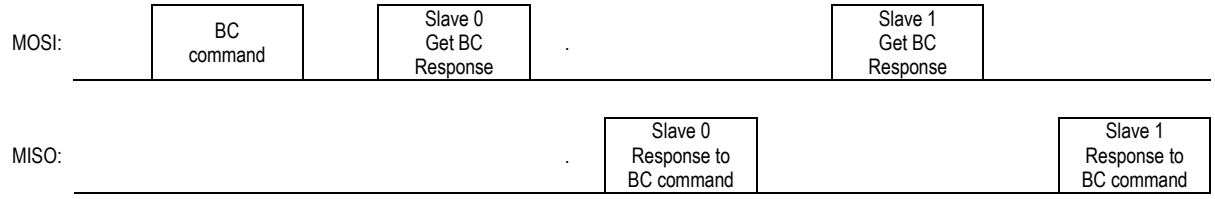
The transfer looks as follows:



Get Broadcast Response Transfer

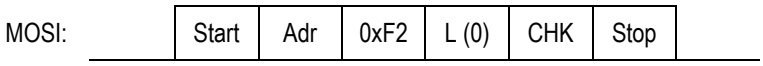
After sending a broadcast command, the slave executes the command but does not send the generated response (the response is stored internally). The “Get Broadcast Response” frame allows you to get the slave response on a previous broadcast command.

The following shows an example with two slaves:



If the next addressed transfer (after a broadcast command) is a “Get Broadcast Response” frame, the slave will send the buffered answer. If any other frame is sent, the buffered response is discarded.

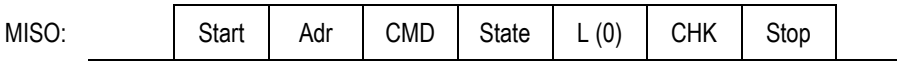
The frame to get the broadcast response (MOSI) looks as follows:



The slave answers with the same response as on an addressed command.

Error Response

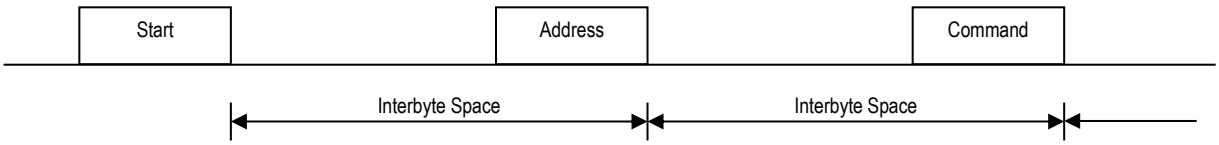
In case of a command execution error, the device will return an error response. This response may be transmitted without data (L=0). That means that a simple error response looks alike for any transfer type:



4.4 PROTOCOL TIMINGS

4.4.1 INTERBYTE TIMEOUT

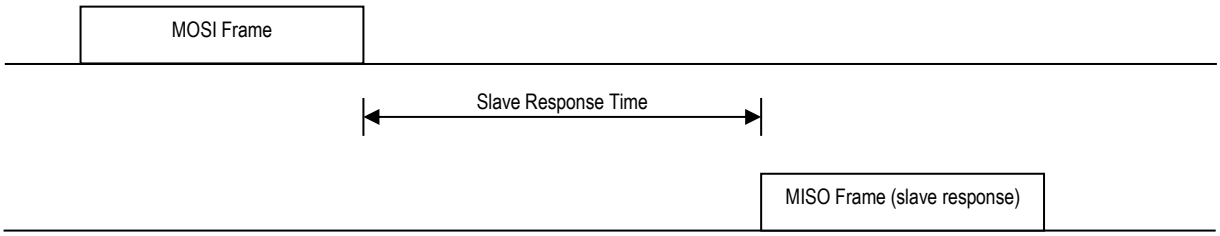
The interbyte time defines the time between two bytes in the same frame. After reception of a frame byte, the receiver waits for the next frame byte. This time is limited by the interbyte timeout. See the following timing diagram which defines the interbyte time:



The interbyte timeout is set to **200ms**. If a timeout occurs, the device will discard the received data (without responding) and wait for the next frame start.

4.4.2 SLAVE RESPONSE TIMEOUT

The slave response time is the time between the MOSI frame has left the master port and the begin of the reception of the MISO frame. This time is defined in the command reference.



Use a timeout which is at least **2 * 'Slave Response Time max'**. The Timeout should not be smaller than **200ms**.

4.5 DATA TYPES AND REPRESENTATION

This chapter shows the transfer of basic data types. The data in the frames is transmitted in **big-endian** order (MSB first).

4.5.1 INTEGER

Integers can be transmitted as signed or unsigned integers. If signed, use the two's complement. The following types of integers are known:

Integer Type	Size	Range
u8t	1 Byte	0 … 2^8-1
u16t	2 Byte	0 … 2^{16}-1
u32t	4 Byte	0 … 2^{32}-1
u64t	8 Byte	0 … 2^{64}-1
i8t	1 Byte	-2^7 … 2^7-1
i16t	2 Byte	-2^{15} … 2^{15}-1
i32t	4 Byte	-2^{31} … 2^{31}-1
i64t	8 Byte	-2^{63} … 2^{63}-1

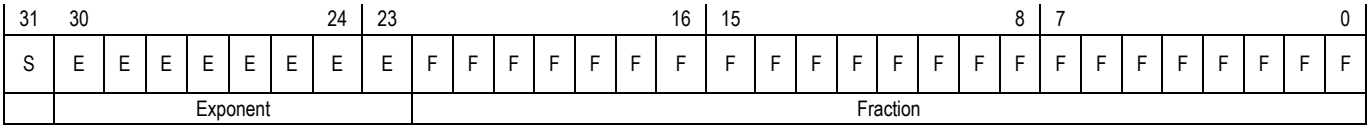
4.5.2 BOOLEAN

A boolean is represented by 1 byte:

- False = 0
- True = 1…255

4.5.3 FLOAT (32-BIT SINGLE PRECISION)

For floating-point representation, the IEEE 754 format is used which has the following structure:



Use the following coding to signal invalid float, positive or negative infinity:

Value	Coding (hex)
invalid float (NaN)	0xFFFFFFFF
+ infinity	0x7F800000
- infinity	0xFF800000

4.5.4 STRING

Strings will be transferred as C-strings. This means in ASCII coding, one byte per character and terminated with a final null-character (0x00). The first letter will be sent first.

4.5.5 ARRAYS

The basic data types (integers, bool or float) can also be defined as arrays (marked with [] after the data type). If an array is transmitted, the value with the lowest index is sent first.

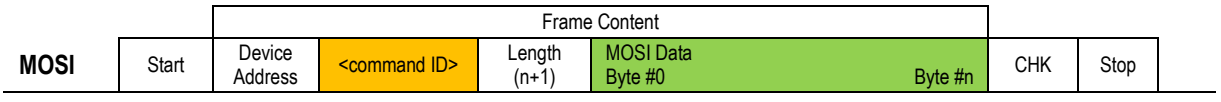
5 SFC5XXX COMMAND REFERENCE

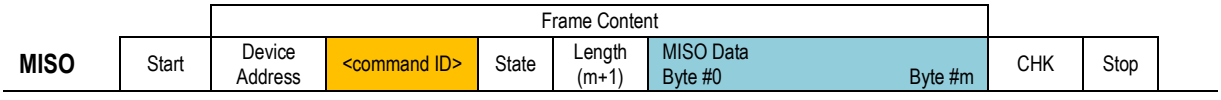
The following section contains the command set used to configure the MFC and exchange process data. To simplify the reading of the commands, they are shown in the following tabular form:

<command name>		
Description	<command description>	
Command ID	<command ID>	
Response Time max	<max response time>	
NV modification	<NV modification>	
MOSI Data	**Byte #**	**Description**
	<location>	*<parameter name> : <parameter type>* <parameter description>
MISO Data	**Byte #**	**Description**
	<location>	*<parameter name> : <parameter type>* <parameter description>

<command name>	Name of the command
<command description>	Description of the command and how to use it
<command ID>	Command byte, which has to be sent with the MOSI frame.
<max response time>	The maximum slave response time in milliseconds
<NV modification>	Defines if the command will modify the non volatile settings of the device (settings which will persist after a system reset).
MOSI Data	A list of all parameters which are sent to the device in the data part of the MOSI frame (lower byte numbers are sent first).
MISO Data	A list of all parameters which are returned from the device in the data part of the MISO frame (lower byte numbers are sent first).
<location>	Location in the data part, where the parameter is located.
<parameter name>	Name of the parameter
<parameter type>	Type of the parameter (see « Data Types and Representation » on page 15)
<parameter description>	Description of the parameter

Filled into the SHDLC frame it will look like this:





5.1 COMMON SHDLC COMMANDS

5.1.1 GET DEVICE INFORMATION (0XD0)

Get Device Information		
Description	This command will return device information as product name, article code or serial number. You will find the same information on the product lable.	
Command ID	0xD0	
Response Time max	10ms	
NV modification	no modification	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0	*Information Type : u8t* This parameter defines which information should be returned: 0x01: Product Name 0x02: Article code 0x03: Serial number
MISO Data	**Byte #**	**Description**
	0 … m	*Requested Information : string* String which contains the requested information

5.1.2 GET VERSION (0XD1)

<table>
<tr><th colspan="3">Get Version</th></tr>
<tr><td>Description</td><td colspan="2">Returns version information of hardware, firmware and SHDLC protocol

Version Format (Firmware, Hardware and SHDLC Protocol):
The version is given in the following format: XX.YY. Where XX represents the major version number and YY the minor number. Note that the minor is always represented by two digits (add leading ‘0’, if minor <10). Example: if you receive 2 for the major and 7 for the minor, this means version 2.07

Firmware Debug State:
Byte #2 in the MISO data is set to 0x00 in all released and delivered firmware versions.</td></tr>
<tr><td>Command ID</td><td colspan="2">0xD1</td></tr>
<tr><td>Response Time max</td><td colspan="2">10ms</td></tr>
<tr><td>NV modification</td><td colspan="2">no modification</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.00</td></tr>
<tr><td>MOSI Data</td><td colspan="2">no data</td></tr>
<tr><td rowspan="8">MISO Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Firmwar Major Version Number : u8t [0…255]</td></tr>
<tr><td>1</td><td>Firmware Minor Version Number : u8t [00…99]</td></tr>
<tr><td>2</td><td>Firmware in Debug State : bool [0x00]</td></tr>
<tr><td>3</td><td>Hardware Major : u8t [0…255]</td></tr>
<tr><td>4</td><td>Hardware Minor: u8t [00…99]</td></tr>
<tr><td>5</td><td>SHDLC protocol version Major : u8t [0…255]</td></tr>
<tr><td>6</td><td>SHDLC protocol version Minor : u8t [00…99]</td></tr>
</table>

5.1.3 GET DEVICE ERROR STATE (0XD2)

Get Device Error State		
Description	When the device is running, some error situations can be detected. This errors will be marked with flags in the state register of the device. If one or more of the error flags are set, this will be signalized to the master by setting the "Device Error Flag" in the state information of the MISO frame (see "4.2.5 State" on page 10). With this function you can readout the state register containing the 32 flags. Find a list of all flags in the appendix (Chapter "6.3 Error Flags in State Register" on page 50).	
Command ID	0xD2	
Response Time max	10ms	
NV modification	no modification	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0	*Clear after read : bool* Defines if the device state and boot error should be cleared after reading. 0x00: Do not clear state register and boot error. 0x01: Clear state register and boot error after reading. Note: if the error situation remains after clearing, the flag will be set again (except the boot error flag #0).
MISO Data	**Byte #**	**Description**
	0…3	*Device state register: u32t [bit encoded]* The device state is a register where 32 independent states can be signalized. The meaning of the bits is defined in the appendix "6.3 Error Flags in State Register" on page 50.
	4	*Boot error : u8t [0…7F]* If an error occurred during system boot, this will be marked by setting flag #0 in the state register. Additional to the flag, this error code defines what exactly went wrong. For a list of the error codes, refer to chapter "6.2 Error Codes" on page 48.

5.1.4 SET/GET DEVICE ADDRESS (0x90)

Note: The command ID to read/write the device address is the same. The length of the MOSI data defines if the device address should be read or written.

Set Device Address		
Description	If the device does not have address switches, the address (RS485/SHDLC) can be modified by software. Use this command to change the device address. The address will be changed in command post processing (after sending the response to the master). Note: If executing this command as broadcast, the device address will be changed immediately after receiving the command. To trigger the broadcast response, use already the new device address.	
Command ID	0x90	
Response Time max	10ms	
NV modification	Modifies address in NV memory	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0	*Slave Address : u8t [0…254]*
MISO Data	no data	

Get Device Address		
Description	Reads the RS485/SHDLC address from the device.	
Command ID	0x90	
Response Time max	10ms	
NV modification	no modification	
Required Firmware	V1.00	
MOSI Data	no data	
MISO Data	**Byte #**	**Description**
	0	*Slave Address: u8t [0…254]*

5.1.5 SET/GET BAUDRATE (0X91)

Set Baudrate		
Description	Changes the RS485/SHDLC baudrate. The slave response will be sent with the old baudrate and afterwards the baudrate will be changed. Note: If executing this command as broadcast, the baudrate will be changed immediately after receiving the command. To trigger the broadcast response, use already the new baudrate.	
Command ID	0x91	
Response Time max	10ms	
NV modification	Modifies baudrate in NV memory	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0…3	*Baudrate: u32t* The new baudrate to set in bits/second. Allowed values are: • 9600 • 19200 • 38400 • 115200 (this is the default baudrate) • 230400 • 460800
MISO Data	no data	

Get Baudrate		
Description	Returns the RS485/SHDLC baudrate	
Command ID	0x91	
Response Time max	10ms	
NV modification	no modification	
Required Firmware	V1.00	
MOSI Data	no data	
MISO Data	**Byte #**	**Description**
	0…3	*Baudrate: u32t* Current baudrate in bits/second.

5.1.6 DEVICE RESET (0XD3)

Device Reset	
Description	Resets the device. This operation has the same effect as a power-reset. Note: The device will response immediately, but needs about 500ms to be ready for communication again after sending the response. Note: If executing this command as broadcast, the device will reset immediately after reception of the command. Reading the broadcast response is not possible.
Command ID	0xD3
Response Time max	10ms
NV modification	no modification
Required Firmware	V1.00
MOSI Data	no data
MISO Data	no data

5.1.7 FACTORY RESET (0x92)

Factory Reset	
Description	This command will rollback all configurations to it's factory defaults. Note: After execution of the command, the device will perform a system reset and needs about 500ms to be ready for communication again. Note: If executing this command as broadcast, the reset will be done after receiving the command. Reading the broadcast response is not possible.
Command ID	0x92
Response Time max	100ms
NV modification	Resets all NV settings to its delivery state.
Required Firmware	V1.00
MOSI Data	no data
MISO Data	no data

5.2 SETPOINT AND MEASURED FLOW

5.2.1 SET/GET SETPOINT (0X00)

Set Setpoint		
Description	Sets the flow setpoint which is used by the flow controller as reference input.	
Command ID	0x00	
Response Time max	5ms	
NV modification	Depends on the setpoint persist setting (see "5.2.7 Set/Get Setpoint Persist (0x02)" on page 30). If setpoint persist is active, the setpoint will be written to NV memory. Else this setting will not modify the NV memory.	
Required Firmware	V1.00, V1.40	
MOSI Data	**Byte #**	**Description**
	0	*Scaling of Setpoint : u8t* Defines if the setpoint is a physical value or normalized (refer to chapter "6.4 Data Scaling" on page 51): 0x00: Normalized setpoint in range [0.0 … 1.0] 0x01: Setpoint represents a physical value. The range depends on the flow unit and calibration range. 0x02: V1.40 Setpoint represents a value in the user defined medium unit.
	1…4	*Setpoing : float* Setpoint as float value. The scaling is defined by the « Scaling of Setpoint » parameter.
MISO Data	no data	

Get Setpoint		
Description	Returns the current setpoint.	
Command ID	0x00	
Response Time max	5ms	
NV modification	no modification	
Required Firmware	V1.00, V1.40	
MOSI Data	**Byte #**	**Description**
	0	*Scaling of Setpoint : u8t* Defines if the setpoint should be returned as physical value or normalized (refer to chapter "6.4 Data Scaling" on page 51): 0x00: Normalized setpoint in range [0.0 … 1.0] 0x01: Setpoint represents a physical value. The range depends on the flow unit and calibration range. 0x02: V1.40 Setpoint represents a value in the user defined medium unit.
MISO Data	**Byte #**	**Description**
	0…3	*Setpoing : float* Setpoint as float value. The scaling is defined by the « Scaling of Setpoint » parameter.

5.2.2 READ MEASURED FLOW (0X08)

Read Measured Flow		
Description	The command returns the latest measured flow value. The value can be read as physical or normalized value.	
Command ID	0x08	
Response Time max	5ms	
NV modification	no modification	
Required Firmware	V1.00, V1.40	
MOSI Data	**Byte #**	**Description**
	0	*Scaling of measured flow : u8t* Defines if the measured flow should be returned as physical value or normalized (refer to chapter “6.4 Data Scaling” on page 51): 0x00: Normalized flow in range [0.0 … 1.0] 0x01: Flow represents a physical value. The range depends on the flow unit and calibration range. 0x02: V1.40 Flow represents a value in the user defined medium unit.
MISO Data	**Byte #**	**Description**
	0 … 3	*Measured flow value : float* Measured flow as float value. The scaling is defined by the « Scaling of measured flow » parameter.

5.2.3 READ MEASURED FLOW BUFFERED (0X09)

Read Measured Flow Buffered		
Description	The MFC has an internal ring buffer in which the measured flow values are automatically stored in a regular interval. The size of the buffer is between 85 and 256 (depends on the specific device configuration). With this command you can readout the buffered values (maximum 60 values per transfer, due to the limited space in the data part of the SHDLC frame). If you do not readout the buffer or you are too slow with reading, the oldest values will be lost. Note that the values which have been read will automatically be cleared from the ring buffer.	
Command ID	0x09	
Response Time max	5ms	
NV modification	no modification	
Required Firmware	V1.00, V1.40	
MOSI Data	**Byte #**	**Description**
	0	*Scaling of measured flow values : u8t* Defines if the measured flow should be returned as physical value or normalized values (refer to chapter “6.4 Data Scaling” on page 51): 0x00: Normalized flow in range [0.0 … 1.0] 0x01: Flow represents a physical value. The range depends on the flow unit and calibration range. 0x02: V1.40 Flow represents a value in the user defined medium unit.
MISO Data	**Byte #**	**Description**
	0 … 3	*Number of measured values lost : u32t* If the time between the ‘Read Measured Flow Buffered’ command calls is to large, the internal ring buffer will overrun. In this case, the oldest value in the buffer is cleared when a new value enters. This number is a counter which counts the missing values between the function calls (number of values which were not readout by the bus master).
	4 … 7	*Number of values remaining in buffer : u32t* The number of values which remains in the buffer after this function call (the number of returned values is limited to 60 values because the maximum allowed data part in the SHDLC frame is 255 bytes).
	8 … 11	*Sampling Time : ft* Time in second between the measured values
	12 … n	*Measured values : ft[]* The measured values read from the ring buffer (0 … 60 values)

5.2.4 READ MEASURED FLOW (2 SENSORS) (0X0A)

Read Measured Flow (2 Sensors)		
Description	The command returns the latest measured flow value from both flow sensors. The values can be read as physical or normalized values. Note: This command is available only on devices with multiple flow sensors.	
Command ID	0x0A	
Response Time max	5ms	
NV modification	no modification	
Required Firmware	V1.48	
MOSI Data	**Byte #**	**Description**
	0	*Scaling of measured flow : u8t* Defines if the measured flow should be returned as physical value or normalized (refer to chapter "6.4 Data Scaling" on page 51): 0x00: Normalized flow in range [0.0 … 1.0] 0x01: Flow represents a physical value. The range depends on the flow unit and calibration range. 0x02: Flow represents a value in the user defined medium unit.
MISO Data	**Byte #**	**Description**
	0 … 3	*Measured flow value main sensor : float* Measured flow from the main flow sensor (used for the flow controller) as float value. The scaling is defined by the « Scaling of measured flow » parameter.
	4 … 7	*Measured flow value secondary sensor : float* Measured flow from the secondary flow sensor as float value. The scaling is defined by the « Scaling of measured flow » parameter.

5.2.5 SET SETPOINT AND READ MEASURED FLOW (0X03)

Set Setpoint and read Measured Flow		
Description	This command is a combination of the two commands "Set/Get Setpoint (0x00)" and "Read Measured Flow (0x08)". This command is intended for process data exchange (setpoint and flow) and safes a lot of protocol overhead compared to separate command usage.	
Command ID	0x03	
Response Time max	5ms	
NV modification	Depends on the setpoint persist setting (see "5.2.7 Set/Get Setpoint Persist (0x02)" on page 30). If setpoint persist is active, the setpoint will be written to NV memory. Else this setting will not modify the NV memory.	
Required Firmware	V1.00, V1.40	
MOSI Data	**Byte #**	**Description**
	0	*Scaling of Setpoint and Flow : u8t* Defines if the setpoint and returned flow are physical values or normalized (refer to chapter "6.4 Data Scaling" on page 51): 0x00: Normalized setpoint/flow in range [0.0 … 1.0] 0x01: Setpoint/flow represents physical values. The range depends on the flow unit and calibration range. 0x02: V1.40 Setpoint/flow represents values in the user defined medium unit.
	1…4	*Setpoint : float* Setpoint as float value. The scaling is defined by the « *Scaling of Setpoint and Flow*» parameter.
MISO Data	**Byte #**	**Description**
	0 … 3	*Measured flow value : float* Measured flow as float value. The scaling is defined by the « *Scaling of Setpoint and Flow*» parameter.

5.2.6 SET SETPOINT AND READ MEASURED FLOW (2 SENSORS) (0X04)

Set Setpoint and read Measured Flow (2 Sensors)		
Description	This command is a combination of the two commands "Set/Get Setpoint (0x00)" and "Read Measured Flow (2 Sensors) (0x0A)". This command is intended for process data exchange (setpoint and flow) and safes a lot of protocol overhead compared to separate command usage. Note: This command is available only on devices with multiple flow sensors.	
Command ID	0x04	
Response Time max	5ms	
NV modification	Depends on the setpoint persist setting (see "5.2.7 Set/Get Setpoint Persist (0x02)" on page 30). If setpoint persist is active, the setpoint will be written to NV memory. Else this setting will not modify the NV memory.	
Required Firmware	V1.46	
MOSI Data	**Byte #**	**Description**
	0	*Scaling of Setpoint and Flow : u8t* Defines if the setpoint and returned flow are physical values or normalized (refer to chapter "6.4 Data Scaling" on page 51): 0x00: Normalized setpoint/flow in range [0.0 … 1.0] 0x01: Setpoint/flow represents physical values. The range depends on the flow unit and calibration range. 0x02: Setpoint/flow represents values in the user defined medium unit.
	1…4	*Setpoint : float* Setpoint as float value. The scaling is defined by the « *Scaling of Setpoint and Flow*» parameter.
MISO Data	**Byte #**	**Description**
	0 … 3	*Measured flow value main sensor : float* Measured flow from the main sensor (used for the flow controller) as float value. The scaling is defined by the « *Scaling of Setpoint and Flow*» parameter.
	4 … 7	*Measured flow value secondary sensor : float* Measured flow from the secondary sensor as float value. The scaling is defined by the « *Scaling of Setpoint and Flow*» parameter.

5.2.7 SET/GET SETPOINT PERSIST (0X02)

Set Setpoint Persist		
Description	Allows to define, if a setpoint should persist after a reset (soft or hardreset) or if it should be set to 0.	
Command ID	0x02	
Response Time max	10ms	
NV modification	Setting is stored in NV memory	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0	*Type of configuration : u8t* Set to 0x00
	1	*Setpoint persist : bool* 0x00 : Set setpoint to 0 after a soft or hardreset 0x01 : Set old setpoint after a soft or hardreset
MISO Data	no data	

Get Setpoint Persist		
Description	Returns the setpoint persist configuration	
Command ID	0x02	
Response Time max	10ms	
NV modification	no modification	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0	*Type of configuration : u8t* Set to 0x80
MISO Data	**Byte #**	**Description**
	0	*Setpoint persist : bool* 0x00 : Sets setpoint to 0 after a soft or hardreset 0x01 : Sets old setpoint after a soft or hardreset

5.3 VALVE INPUT SOURCE CONFIGURATION

5.3.1 SET/GET VALVE INPUT SOURCE CONFIGURATION (0X20)

<table>
<tr><th colspan="3">Set Valve Input Source Configuration</th></tr>
<tr><td>Description</td><td colspan="2">By default, the valve is controlled by the controller. To change this behavior, the valve input source can be changed.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x20</td></tr>
<tr><td>Response Time max</td><td colspan="2">5ms</td></tr>
<tr><td>NV modification</td><td colspan="2">no modification</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.40</td></tr>
<tr><td rowspan="3">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Defines the configuration which should set : u8t
0x00: set valve input source
0x01: set user defined valve value</td></tr>
<tr><td>1 … n</td><td>Set valve input source:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1</td><td>Input Source : u8t
0x00 = Controller (default), driven by the flow controller
0x01 = Force Closed, valve remains fully closed
0x02 = Force Open, valve remains fully open
0x03 = Hold, hold the voltage on the valve
0x10 = User Defined, user defined value 0 … 1</td></tr>
</table>
Set user defined valve value:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1 … 5</td><td>User Defined Value (0 … 1): float
0 → fully closed
1 → fully open</td></tr>
</table></td></tr>
<tr><td>MISO Data</td><td colspan="2">no data</td></tr>
</table>

<table>
<tr><th colspan="3">Get Valve Input Source Configuration</th></tr>
<tr><td>Description</td><td colspan="2">Gets the current valve input source.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x20</td></tr>
<tr><td>Response Time max</td><td colspan="2">5ms</td></tr>
<tr><td>NV modification</td><td colspan="2">no modification</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.40</td></tr>
<tr><td rowspan="2">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Defines the configuration which should get : u8t
0x00: get valve input source
0x01: get user defined valve value</td></tr>
<tr><td rowspan="2">MISO Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0 … n</td><td>Get valve input source:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0</td><td>Input Source : u8t
0x00 = Controller (default), driven by the flow controller
0x01 = Force Closed, valve remains fully closed
0x02 = Force Open, valve remains fully open
0x03 = Hold, hold the voltage on the valve
0x10 = User Defined, user defined value 0 … 1</td></tr>
</table>
Get user defined valve value:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0 … 4</td><td>User Defined Value (0 … 1): float
0 → fully closed
1 → fully open</td></tr>
</table></td></tr>
</table>

5.4 MEDIUM UNIT CONFIGURATION

5.4.1 SET/GET MEDIUM UNIT CONFIGURATION (0X21)

<table>
<tr><th colspan="3">Set Medium Unit Configuration</th></tr>
<tr><td>Description</td><td colspan="2">You can define your own unit for the flow values. For example, if the calibration is done in norm liter per minute but sccm is needed, you can set the medium unit to sccm. If you select user defined medium unit for flow scaling, the MFC recalculates all flow values into the specified unit.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x21</td></tr>
<tr><td>Response Time max</td><td colspan="2">5ms</td></tr>
<tr><td>NV modification</td><td colspan="2">Setting is stored in NV memory</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.40</td></tr>
<tr><td rowspan="3">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Defines the configuration which should set : u8t
0x00: set user defined medium unit</td></tr>
<tr><td>1 … n</td><td>Set user defined medium unit:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1</td><td>Medium Unit Prefix: i8t
See appendix for encoding
0x7F = Use medium unit prefix from calibration</td></tr>
<tr><td>2</td><td>Medium Unit: u8t
See appendix for encoding
0xFF = Use medium unit from calibration</td></tr>
<tr><td>3</td><td>Timebase: u8t
See appendix for encoding
0xFF = Use timebase from calibration</td></tr>
</table></td></tr>
<tr><td>MISO Data</td><td colspan="2">no data</td></tr>
</table>

Get Medium Unit Configuration

Description	Gets the user defined medium unit configuration.	
Command ID	0x21	
Response Time max	5ms	
NV modification	no modification	
Required Firmware	V1.40	
MOSI Data	**Byte #**	**Description**
	0	*Defines the configuration which should get : u8t* 0x00: get user defined medium unit 0x01: get user defined medium unit without wildcards 0x0A: get converted fullscale
MISO Data	**Byte #**	**Description**
	0 … n	*Get user defined medium unit:*

Byte #	Description
0	*Medium Unit Prefix: i8t* See appendix for encoding 0x7F = Use medium unit prefix from MCB
1	*Medium Unit: u8t* See appendix for encoding 0xFF = Use medium unit from MCB
2	*Timebase: u8t* See appendix for encoding 0xFF = Use timebase from MCB

Get user defined medium without wildcards:

Byte #	Description
0	*Medium Unit Prefix: i8t* See appendix for encoding
1	*Medium Unit: u8t* See appendix for encoding
2	*Timebase: u8t* See appendix for encoding

Get converted fullscale:

Byte #	Description
0 … 3	*Fullscale for user defined medium : float*

5.5 CONTROLLER CONFIGURATION

5.5.1 SET/GET CONTROLLER CONFIGURATION (0X22)

<table>
<tr><th colspan="3">Set Controller Configuration</th></tr>
<tr><td>Description</td><td colspan="2">Sets user settings for the flow controller.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x22</td></tr>
<tr><td>Response Time max</td><td colspan="2">5ms</td></tr>
<tr><td>NV modification</td><td colspan="2">Setting is stored in NV memory</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.42, V1.45</td></tr>
<tr><td rowspan="2">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Defines the configuration which should set : u8t
0x00: set user controller gain
0x10: set pressure dependent gain on/off
0x11: set inlet pressure for gain correction
0x20: V1.45 set gas temperature compensation on/off
0x21: V1.45 set inlet gas temperature for compensation</td></tr>
<tr><td></td><td>1 … n</td><td>Set user controller gain:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1 … 4</td><td>User Controller Gain: float</td></tr>
</table>
Set pressure dependent gain on/off:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1</td><td>Enable: u8t
0x00 = off
0x01 … 0xFF = on</td></tr>
</table>
Set inlet pressure for gain correction:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1 … 4</td><td>Inlet pressure [Bar]: float</td></tr>
</table>
Set gas temperature compensation on/off:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1</td><td>Enable: u8t
0x00 = off
0x01 … 0xFF = on</td></tr>
</table>
Set inlet gas temperature for compensation:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1 … 4</td><td>Inlet gas temperature [°C]: float</td></tr>
</table></td></tr>
<tr><td>MISO Data</td><td colspan="2">no data</td></tr>
</table>

Get Controller Configuration

Description	Gets the user settings from the flow controller.	
Command ID	0x22	
Response Time max	5ms	
NV modification	no modification	
Required Firmware	V1.42, V1.45	
MOSI Data	**Byte #**	**Description**
	0	*Defines the configuration which should get : u8t* 0x00: get user controller gain 0x10: get pressure dependent gain on/off 0x11: get inlet pressure for gain correction 0x20: V1.45 get gas temperature compensation on/off 0x21: V1.45 get inlet gas temperature for compensation
MISO Data	**Byte #**	**Description**
	0 … n	*Get user controller gain:* **Byte #** / **Description** 0 … 3 / *User Controller Gain: ftoat* *Get pressure dependent gain on/off:* **Byte #** / **Description** 0 / *Enable: u8t* 0x00 = off 0x01 = on *Get inlet pressure for gain correction:* **Byte #** / **Description** 0 … 3 / *Inlet pressure: float* *Get gas temperature compensation on/off:* **Byte #** / **Description** 1 / *Enable: u8t* 0x00 = off 0x01 … 0xFF = on *Get inlet gas temperature for compensation:* **Byte #** / **Description** 1 … 4 / *Inlet gas temperature [°C]: float*

5.6 ADVANCED MEASUREMENTS (0X30)

<table>
<tr><th colspan="3">Advanced Measurements</th></tr>
<tr><td>Description</td><td colspan="2">For the advanced measurements, the sensor is reconfigured to read different raw data. Meanwhile, the current flow can't be read. To keep the flow as stable as possible, the valve voltage is held during this short time.

Note: The temperature compensated raw thermal conductivity measurement requires a special temperature calibration, otherwise an error will be returned (applies only to FW >= 1.56; older versions return always the uncompensated thermal conductivity value).</td></tr>
<tr><td>Command ID</td><td colspan="2">0x30</td></tr>
<tr><td>Response Time max</td><td colspan="2">600ms</td></tr>
<tr><td>NV modification</td><td colspan="2">no modification</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.43, V1.56</td></tr>
<tr><td rowspan="3">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Defines the measurement which should be done : u8t
0x00: Raw Flow (5ms)
0x01: Raw Thermal Conductivity (5ms)
0x02: Raw Thermal Conductivity with closed valve (500ms)
0x10: Temperature in °C (1ms)</td></tr>
<tr><td>1 … n</td><td>For Raw Flow measurement, no additional parameters are required.

For Raw Thermal Conductivity measurement:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1</td><td>Use temperature compensation : u8t
This is an optional parameter since V1.56 to choose between uncompensated and temperature compensated measurement. If this parameter is not used, the compensated value will be returned if possible, otherwise the uncompensated value will be returned.
0x00: Read uncompensated thermal conductivity
0x01: Read temp. compensated thermal conductivity</td></tr>
</table>
For Raw Thermal Conductivity with closed valve measurement:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>1</td><td>Use temperature compensation : u8t
This is an optional parameter since V1.56 to choose between uncompensated and temperature compensated STP measurement. If this parameter is not used, the compensated value will be returned if possible, otherwise the uncompensated value will be returned.
0x00: Read uncompensated thermal conductivity
0x01: Read temp. compensated thermal conductivity</td></tr>
</table>
For Raw Temperature measurement, no additional parameters are required.

For Temperature in °C measurement, no additional parameters are required.</td></tr>
<tr><td>MISO Data</td><td>Byte #</td><td>Description</td></tr>
</table>

<table>
<tr>
<td></td>
<td>0 … n</td>
<td>
For Raw Flow measurement:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0 … 1</td><td>Raw Flow measurement result : u16t</td></tr>
</table>
For Raw Thermal Conductivity measurement:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0 … 1</td><td>Raw Thermal Conductivity measurement result : u16t</td></tr>
</table>
For Raw Thermal Conductivity with closed valve measurement:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0 … 1</td><td>Raw Thermal Conductivity measurement result : u16t</td></tr>
</table>
For Raw Temperature measurement:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0 … 1</td><td>Sensor raw temperature : u16t</td></tr>
</table>
For Temperature in °C measurement:
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0 … 3</td><td>Sensor temperature in °C : float</td></tr>
</table>
</td>
</tr>
</table>

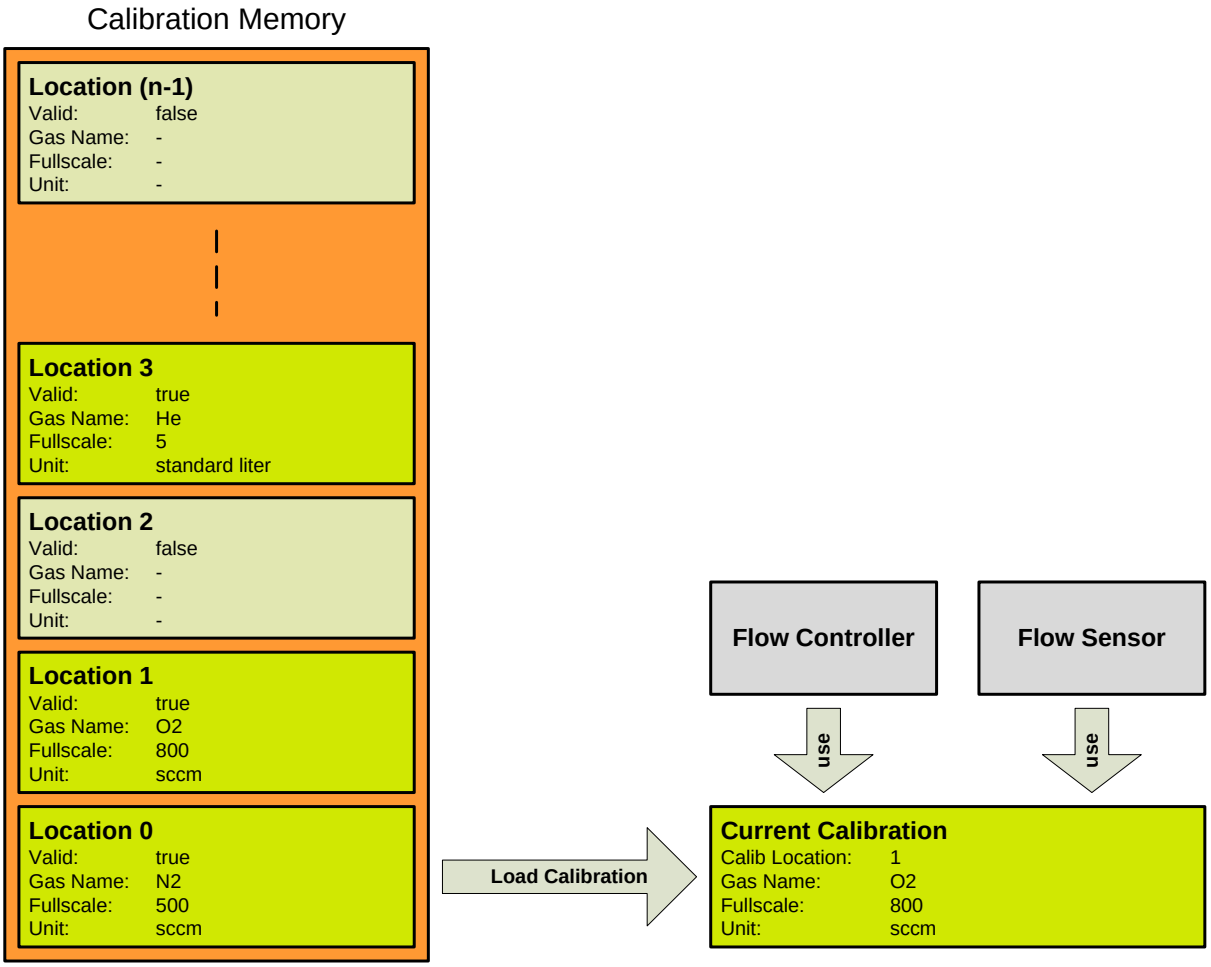
5.7 CALIBRATION HANDLING

The SFC5xxx MFC system has the possibility to store several calibrations. This chapter describes how to get information about the available calibrations and how to switch between calibrations.

The picture shows the calibration handling on the MFC. There is a calibration memory which holds all available calibrations. To work with a calibration, it must be loaded, so the flow sensor and flow controller can work with.

The calibration memory can hold up to n calibrations (n depends on the specific product). Every calibration is stored in a numbered location from 0 to (n-1). Note that not every location contains a valid calibration and the invalid locations need not necessarily be located at the end.

Use the command “Get Calibration Information (0x40)” to get information about a specific calibration or to generate a list of all available calibrations. The “Load Calibration and Run (0x45)” command loads a calibration from memory and runs the flow controller. With the “Get Current Calibration Information (0x44)” command, you can get information about the current calibration.



Note: The picture above is an example. The calibrated gases and their locations depend on the specific product.

5.7.1 LOAD CALIBRATION AND RUN (0X45)

Load Calibration and Run		
Description	This command loads a defined calibration from calibration memory and runs the flow controller. Note: If the selected calibration is already loaded, the function will not do anything. **Caution:** This command will cause a write operation in an EEPROM. Because of the limited write cycles of the EEPROM, this command should not be called periodical (max 50'000 times with a new calibration). Loading the same calibration again is not a problem and will not cause a write operation.	
Command ID	0x45	
Response Time max	1600ms (to load calibration which is not already loaded)	
NV modification	The loaded calibration is saved in NV memory.	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0 … 3	*Location of calibration in memory : u32t* The location of the calibration in the calibration memory [0…n-1].
MISO Data	no data	

5.7.2 GET CALIBRATION INFORMATION (0X40)

<table>
<tr><th colspan="3">Get Calibration Information</th></tr>
<tr><td>Description</td><td colspan="2">Returns information about the calibration memory and the available calibrations.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x40</td></tr>
<tr><td>Response Time max</td><td colspan="2">10ms</td></tr>
<tr><td>NV modification</td><td colspan="2">no modification</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.00, V1.10, V1.41</td></tr>
<tr><td rowspan="3">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Type of information to read : u8t
This parameter defines what kind of information to read (and also defines how the data part in the MISO frame looks like):

0x00: Read the size of the calibration memory (number of possible calibrations n).

0x10: Read validity of calibration
0x11: Read gas description string
0x12: Read gas ID (every gas has a unique ID)
0x13: Read gas unit
0x14: Read fullscale flow
0x15: V1.10 Initial calibration condition
0x16: V1.10 Recalibration condition
0x17: V1.41 Thermal conductivity reference value</td></tr>
<tr><td>[1 … 4]</td><td>Calibration memory location : u32t
Defines the location in memory to read information about a specific calibration (required for information type 0x10…0x14). To read the size of calibration memory (0x00), omit this parameter.</td></tr>
<tr><td rowspan="2">MISO Data
(Note: the MISO data depends on the selected type of information in the MOSI frame)</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0 … n</td><td>MISO data for “Size of calibration memory (0x00)”
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0…4</td><td>Size of the calibration memory : u32t
Returns the size of the calibration memory (number of possible calibrations n).</td></tr>
</table>

MISO data for “Validity of calibration (0x10)”
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0</td><td>Validity of calibration : bool
0x00 : No valid calibration at this location
0x01 : Valid calibration (can be loaded)</td></tr>
</table>

MISO data for “Gas description string (0x11)”
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0…n</td><td>Gas description : string
A string which describes the calibrated gas at the given calibration memory location</td></tr>
</table>
</td></tr>
</table>

MISO data for "Gas ID (0x12)"

Byte #	Description
0…3	*Gas ID : u32t* Returns the unique gas ID.

MISO data for "Gas Unit (0x13)"

See chapter "6.5 Gas Unit Encoding" on page 52 for a description of the unit encoding.

Byte #	Description
0	*Gas Unit Prefix: i8t*
1	*Gas Unit: u8t*
2	*Timebase: u8t*

MISO data for "Fullscale Flow (0x14)"

Byte #	Description
0…3	*Fullscale Flow: float* The calibrated flow range (with the used flow unit).

MISO data for "Initial calibration condition (0x15)"

Byte #	Description
0…49	*Company : string* Defines which company has calibrated
50…99	*Operator : string* Operator who has calibrated
100…101	*Year of calibration : u16t*
102	*Month of calibration : u8t*
103	*Day of calibration : u8t*
104	*Hour of calibration : u8t*
105	*Minute of calibration : u8t*
106…109	*Calibration Temperature : float* System/Gas temperature [°C] of calibration
110…113	*Calibration inlet pressure : float* Absolute pressure of gas inlet [bar]
114 … 117	*Calibration differential pressure : float* Pressure difference between inlet and outlet [bar]
118	*Real gas calibration : bool* True : calibrated with process gas, False : calculated
119…122	*Calibration accuracy (setpoint) : float* Calibration accuracy in percent of the current setpoint. This accuracy is valid, if larger than the accuracy of fullscale.
123…126	*Calibration accuracy (fullscale) : float* Calibration accuracy in percent of fullscale. This value is valid, if larger than the accuracy of the current setpoint.

MISO data for "Recalibration condition (0x16)"

Byte #	Description
0…49	*Company : string* Defines which company has recalibrated
50…99	*Operator : string* Operator who has recalibrated
100…101	*Year of recalibration : u16t*
102	*Month of recalibration : u8t*
103	*Day of recalibration : u8t*
104	*Hour of recalibration : u8t*
105	*Minute of recalibration : u8t*
106…109	*Calibration Temperature : float* System/Gas temperature [°C] of recalibration
110…113	*Calibration inlet pressure : float* Absolute pressure of gas inlet [bar]
114 … 117	*Calibration differential pressure : float* Pressure difference between inlet and outlet [bar]
118	*Real gas calibration : bool* True : recalibrated with process gas, False : calculated
119…122	*Calibration accuracy (setpoint) : float* Calibration accuracy in percent of the current setpoint. This accuracy is valid, if larger than the accuracy of fullscale.
123…126	*Calibration accuracy (fullscale) : float* Calibration accuracy in percent of fullscale. This value is valid, if larger than the accuracy of the current setpoint.

MISO data for "Thermal conductivity reference value (0x17)"

Byte #	Description
0…1	*Thermal conductivity reference value : u16t* Thermal conductivity reference value for the gas.

5.7.3 GET CURRENT CALIBRATION INFORMATION (0X44)

<table>
<tr><th colspan="3">Get Current Calibration Information</th></tr>
<tr><td>Description</td><td colspan="2">Returns calibration information of the current (active) calibration.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x44</td></tr>
<tr><td>Response Time max</td><td colspan="2">10ms</td></tr>
<tr><td>NV modification</td><td colspan="2">no modification</td></tr>
<tr><td>Required Firmware</td><td colspan="2">V1.00, V1.10, V1.41</td></tr>
<tr><td rowspan="2">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Type of information to read : u8t
This parameter defines what kind of information to read (and also defines how the data part in the MISO frame looks like):

0x11: Read gas description string
0x12: Read gas ID (every gas has a unique ID)
0x13: Read gas unit
0x14: Read fullscale flow
0x15: V1.10 Initial calibration condition
0x16: V1.10 Recalibration condition
0x17: V1.41 Thermal conductivity reference value</td></tr>
<tr><td rowspan="2">MISO Data
(Note: the MISO data depends on the selected type of information in the MOSI frame)</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0…n</td><td>MISO data for “Gas description string (0x11)”
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0…n</td><td>Gas description : string
A string which describes the current gas.</td></tr>
</table>
MISO data for “Gas ID (0x12)”
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0…3</td><td>Gas ID : u32t
Returns the unique gas ID.</td></tr>
</table>
MISO data for “Gas Unit (0x13)”
See chapter “6.5 Gas Unit Encoding” on page 52 for a description of the unit encoding.
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0</td><td>Gas Unit Prefix: i8t</td></tr>
<tr><td>1</td><td>Gas Unit: u8t</td></tr>
<tr><td>2</td><td>Timebase: u8t</td></tr>
</table>
MISO data for “Fullscale Flow (0x14)”
<table>
<tr><th>Byte #</th><th>Description</th></tr>
<tr><td>0…3</td><td>Fullscale Flow: float
The calibrated flow range (with the used flow unit).</td></tr>
</table></td></tr>
</table>

MISO data for "Initial calibration condition (0x15)"

Byte #	Description
0…49	*Company : string* Defines which company has calibrated
50…99	*Operator : string* Operator who has calibrated
100…101	*Year of calibration : u16t*
102	*Month of calibration : u8t*
103	*Day of calibration : u8t*
104	*Hour of calibration : u8t*
105	*Minute of calibration : u8t*
106…109	*Calibration Temperature : float* System/Gas temperature [°C] of calibration
110…113	*Calibration inlet pressure : float* Absolute pressure of gas inlet [bar]
114 … 117	*Calibration differential pressure : float* Pressure difference between inlet and outlet [bar]
118	*Real gas calibration : bool* True : calibrated with process gas, False : calculated
119…122	*Calibration accuracy (setpoint) : float* Calibration accuracy in percent of the current setpoint. This accuracy is valid, if larger than the accuracy of fullscale.
123…126	*Calibration accuracy (fullscale) : float* Calibration accuracy in percent of fullscale. This value is valid, if larger than the accuracy of the current setpoint.

MISO data for "Recalibration condition (0x16)"

Byte #	Description
0…49	*Company : string* Defines which company has recalibrated
50…99	*Operator : string* Operator who has recalibrated
100…101	*Year of recalibration : u16t*
102	*Month of recalibration : u8t*
103	*Day of recalibration : u8t*
104	*Hour of recalibration : u8t*
105	*Minute of recalibration : u8t*
106…109	*Calibration Temperature : float* System/Gas temperature [°C] of recalibration
110…113	*Calibration inlet pressure : float* Absolute pressure of gas inlet [bar]
114 … 117	*Calibration differential pressure : float* Pressure difference between inlet and outlet [bar]
118	*Real gas calibration : bool* True : recalibrated with process gas, False : calculated
119…122	*Calibration accuracy (setpoint) : float* Calibration accuracy in percent of the current setpoint. This accuracy is valid, if larger than the accuracy of fullscale.
123…126	*Calibration accuracy (fullscale) : float* Calibration accuracy in percent of fullscale. This value is valid, if larger than the accuracy of the current setpoint.

MISO data for "Thermal conductivity reference value (0x17)"

Byte #	Description
0…1	*Thermal conductivity reference value : u16t* Thermal conductivity reference value for the current gas.

5.8 System Settings

5.8.1 User Memory Access (0x6E)

The user memory space is a region of 100 bytes in non-volatile memory. For some OEM products, this memory is used to supply the customer with additional (customer-specific) data. Standard products do not use this memory space. In any case, the MFC system does not read this data or rely on it.

The memory may be used by the customer to save additional information (e.g. location of the MFC, who hast installed it, last service,...).

Note that a factory reset will also reset the content of the user memory space to its delivery state.

User Memory Access		
Description	Command to read or write the user memory space (or part of it).	
Command ID	0x6E	
Response Time max	10ms	
NV modification	Modifies user memory space in NV memory	
Required Firmware	V1.00	
MOSI Data	**Byte #**	**Description**
	0	*Startadress : u8t* Address from where to start reading/writing [0…99]
	1	*Number of bytes to read/write : u8t* Defines how many bytes should be read/written [1…100]. The size of the MOSI data defines if the data should be read (data size = 2) or written (data size > 2).
	[2…n]	*Write data : ut8[]* If data should be written, transmit the write data here.
MISO Data	**Byte #**	**Description**
	[0…n]	*Read data : u8t[]* If the MOSI data contains only 2 bytes (size = 2), the function will read the defined number of bytes, beginning at the startaddress. The read data is returned here. In case of writing data, the MISO frame does not contain any data.

6 APPENDIX

6.1 COMMAND LIST

6.2 ERROR CODES

The following table shows the error codes which can be reported from the device. Note that most of these errors are system internal errors which require additional knowledge to understand. In case of a problem, they will help Sensirion to localize and solve the issue.

Error Code		
dec	hex	Meaning
0	0x00	No Error
1	0x01	Wrong data length for this command (too much or less data)
2	0x02	Unknown command
3	0x03	Insufficient access rights to execute this command
4	0x04	Illegal command parameter or parameter out of allowed range
5…31		undefined
32	0x20	Functionality not implemented
33	0x21	Address of non-volatile memory out of range
34	0x22	Frame checksum error (no MISO frame, when this error detected)
35	0x23	Invalid address in frame (no MISO frame, when this error detected)
36	0x24	Illegal special frame identifier
37	0x25	Wrong data size for given subcommand
38	0x26	Frame length information does not match with received number of bytes
39	0x27	Trigger broadcast response, but no valid response available
40	0x28	Internal function argument out of range
41	0x29	NACK received from I2C device
42	0x2A	Master hold not released in I2C
43	0x2B	I2C CRC mismatch
44	0x2C	Sensor data read back differs from written value
45	0x2D	Sensor measure loop is not running
46	0x2E	Timeout while starting signal processor
47	0x2F	Timeout while stopping signal processor
48	0x30	Error while trying to recover SF04 sensor
49	0x31	Not possible to modify signal processor while startup or shut down
50	0x32	Hardware communication failed
51	0x33	No valid calibration block at given flash location
52	0x34	No valid calibration at given sensor location
53	0x35	No appropriate gain setting found with valve adaption
54	0x36	I2C lines (SCL or SDA) low before sending start condition
55	0x37	Supply voltage out of range
56	0x38	Unknown HW type
57	0x39	Unknown HW version
58	0x3A	Flash memory is not cleared
59	0x3B	FRAM write error (read back mismatch)
60	0x3C	Flash write error (read back mismatch)
61	0x3D	Sensor EEPROM write error (read back mismatch)
62	0x3E	Sensor NACK
63	0x3F	Missing gas pressure, could not reach setpoint
64	0x40	Could not startup external oscillator
65	0x41	Communication adapter not available
66	0x42	Sensor busy

67	0x43	Command not allowed in the current state of the device
68	0x44	Functionality not supported by the device
69…126		undefined
127	0x7F	Fatal system error

6.3 ERROR FLAGS IN STATE REGISTER

The following table shows the meaning of the error flags, which are located in the state register of the device.

Flag #	Meaning
0	Boot Error Error occurred while system boot. The “boot error” code which can be read with the “Get Device Error State” command will provide more information.
1	Command Post Processing Error There are some commands which will be processed after sending the command response to the master (for example the change baud rate command, which will reconfigure the baud rate after sending the command response). This flag is set, if something is going wrong during post processing.
2	Input Supply out of Range The power supply of the MFC is out of range.
3	Valve Supply out of Range The valve supply voltage is out of range.
4	Signal Processor Initialization Could not proper startup the flow controller.
5	Sensor Communication Error Communication problem between microcontroller and flow sensor.
6	Setpoint Input Error Problem when reading the setpoint (only analog MFC)
7	Actuator Output Error Problem when writing the control value to the valve.
8	Signal Output Error Problem with a signal output (only analog MFC)
9	Signal Buffer Error Problem with the flow data buffer.
10	Missing Gas Pressure Gas pressure is too low. The flow controller cannot reach the given setpoint also when valve is fully open.
11…31	unused (set to 0)

6.4 DATA SCALING

The setpoint and measured flow can be read or written as normalized or physical (scaled) value. This data representation can be selected with every command which operates with these values.

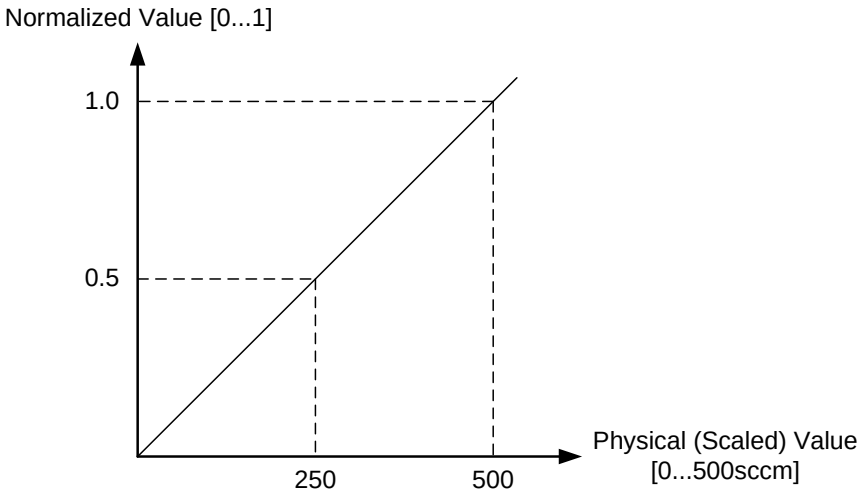
Normalized Values

The values are scaled in the range 0 (no flow) to 1 (full-scale flow). With this data representation, you don't have to care about the full-scale value or the flow unit. Interpret these values as relative values in relation to the calibrated full-scale.

Physical (Scaled) Values

A value which represents a physical flow. The values are in the range 0 (no flow) up to the full-scale value, which depends on the calibrated range and flow unit.

The following graphic shows the relation between normalized and scaled values for an MFC which is calibrated for 500 sccm:



6.5 GAS UNIT ENCODING

The following tables show the encoding of the gas unit:

Prefix

Prefix Code (i8t)	Prefix	Symbol	10^n
-24	yocto	y	10^{-24}
-21	zepto	z	10^{-21}
-18	atto	a	10^{-18}
-15	femto	f	10^{-15}
-12	pico	p	10^{-12}
-9	nano	n	10^{-9}
-6	micro	u	10^{-6}
-3	milli	m	10^{-3}
-2	centi	c	10^{-2}
-1	deci	d	10^{-1}
0			10^{0}
1	deca	da	10^{1}
2	hecto	h	10^{2}
3	kilo	k	10^{3}
6	mega	M	10^{6}
9	giga	G	10^{9}
12	tera	T	10^{12}
15	peta	P	10^{15}
18	exa	E	10^{18}
21	zetta	Z	10^{21}
24	yotta	Y	10^{24}
127	undefined	-	-

Unit

Unit Code (u8t)	Description	Symbol
0	norm liter (0°C, 1013 hPa)	l
1	standard Liter (20°C, 1013 hPa)	l
8	liter (liqui)	l
9	gram	g
16	pascal	Pa
17	bar	bar
18	meter H2O	mH2O
19	inch H2O	iH2O
255	undefined	-

Time Base

Time Base Code (u8t)	Description	Symbol
0	no time base	
1	per microsecond	/us
2	per millisecond	/ms
3	per second	/s
4	per minute	/min
5	per hour	/h
6	per day	/day
255	undefined	-