SENSIRION SFC6 Mass Flow Controller User Manual

Contents

SENSIRION

SENSIRION SFC6 Mass Flow Controller

Document Summary

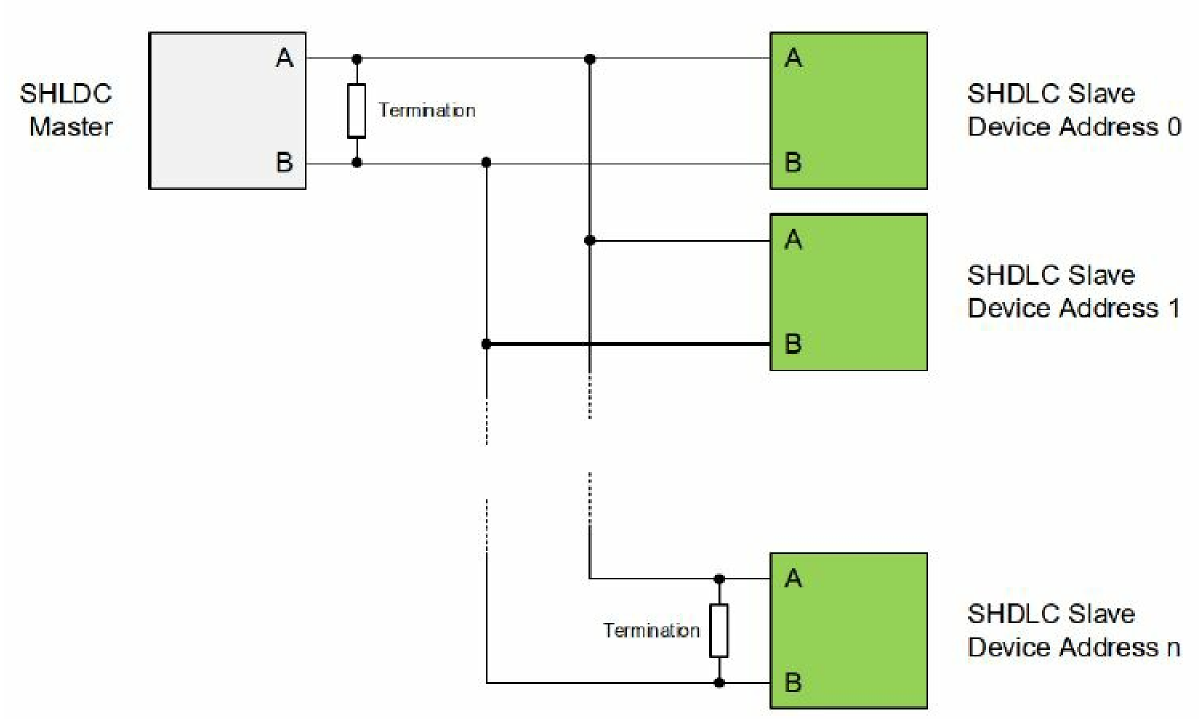
This document describes the SHDLC RS485 communication interface of the Sensirion SFC6xxx mass flow controller and SFM6xxx mass flow meter. The intention of defining the SHDLC protocol was to have a single protocol for different Sensirion devices such as mass flow controllers, meters, and other microcontroller devices. The protocol is based on serial communication and is easy to implement. It allows communication between a single master and several connected slave devices.

In the following, there are some of the SHDLC protocol key features:

- Based on an byte orientated hardware (like UART,. . .)
- Half-duplex system, which allows to use half-duplex hardware
- Single Master/Multiple Slaves protocol without the need for bus arbitration
- Allows up to 255 bytes payload data for read or write transfers
- Broadcasting allows to safe bus load and synchronize devices

Hardware

RS485/RS422



As shown in the picture above, the RS485/422 setup combined with the SHLDC addressing allows to connect several SHDLC devices on the same bus. Due to the differential signals, the communication is robust against disturbances and allows long distance data transmission. For long distance data lines a termination resistor of 120 Ohm is recommended on both ends of the bus. The logic behind the protocol is based on a UART hardware, using the following settings:

- Baudrate as defined in datasheet (default: 115’200 baud)
- 8 Data bits (LSb first)
- No Parity
- 1 Stop Bit

Frame Definition

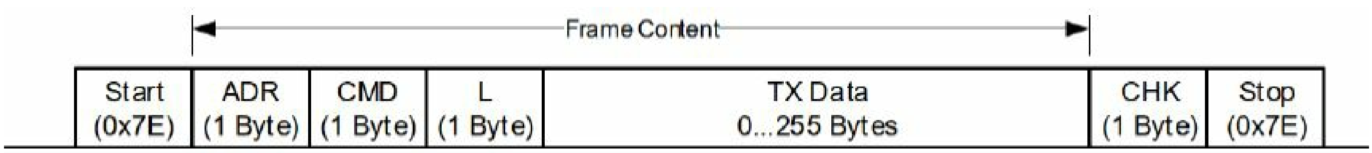
Every transfer in SHDLC is based on MOSI (Master Out Slave In) and MISO (Master In Slave Out) frames. In the following chapter, the composition of the frames and the meaning of the different frame parts is shown.

Note

- In the following, the order of the bytes in frames is defined. The order of the bits inside a byte (MSb/LSb) is defined by the used hardware.

Type of Frames

MOSI Frame

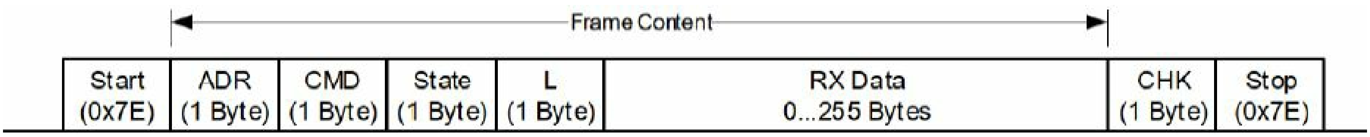


The MOSI frame is the message sent from master to slave as a request. Communication on the SHDLC bus is always initiated by the master. Therefore, this frame is always the begin of communication on the bus:

The begin and end of the frame is marked by start and stop bytes, which are unique bytes (0x7E) in the frame (guaran- teed by byte suffing, see Section 5.2). After the start byte, the address field defines the receiver of the frame, followed by the command which tells the slave what to do with the data. Before the payload data is sent, there is a length information (L) and after the data a checksum to guarantee the validity of the data.

MISO Frame

The MISO frame is the response from a slave device after reception of the MOSI frame. The SHDLC protocol guarantees an answer on any master request (except broadcast commands). This allows the master to ensure his request was correctly transferred and processed. The MISO frame looks as follows:



It is similar to the MOSI frame, except the additional state byte. In this field, the slave reports any problem during command execution and additional general health information. In the address field, the slave fills in his own address and in the command field the command which was received and executed.

Start and Stop

Since SHDLC does not use hardware handshaking, we are using special bytes in the data stream to mark the begin and end of a frame:

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

To make this work properly, we have to avoid this byte (0x7E) inside the frame. For this reason, byte stuffing is used, which means we are replacing 0x7E anywhere inside the frame with a sequence of two other bytes. This will also be done for Escape (0x7D), XON (0x11) and XOFF (0x13) bytes. The following table shows how the original data bytes are replaced in the sent data. All other data bytes are sent 1:1.

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

Address Field

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

- **0. . . 254:** Slave addresses (All device addresses are initially set to 0x0, and can be changed with the get/set slave address command)

- **255:** Broadcast address

In a MISO frame the address field contains the slave address (address of the slave which has sent the MISO frame).

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. In a MISO frame the slave will return the received and processed command in the CMD field. The following table shows the organization of the available command space:

Command ID (Hex)	**Size**	**Usage**
0x00 .. 0x7F	128	Individual, device specific commands. The meaning and payload data of these commands depend on the type of device.
0x80 .. 0xEF	112	Common commands. The meaning and payload data of these commands are identical for all device types. However, not all commands are supported by each device type, i.e. the commands are optional.
0xF0 .. 0xFF	16	Special frame identifiers.

- The size of the command is 1 byte.

State

The MISO frame contains a state byte, which allows the master to detect errors on slave side. There are two kinds of errors signalized in the state byte:

- Execution Errors
- Device Error Flag

The following table shows the composition of the state byte:

b7	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
Device Error Flag	Execution error code						

Length

- The length byte defines the number of transferred bytes in the data field (Rx or Tx). It is the length of the data field before byte stuffing, not the number of bytes which are transferred (because they contain stuffed bytes).

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

Execution Error Code

The execution error code signalizes all errors which occured while processing the frame or executing the command. The errors are binary encoded in 7 bits with the following meaning:

Error Code	Meaning
0x00	No Error. Processing and execution of the command was successful.
0x01 . . . 0x1F	Common error codes, which have the same meaning on every SHDLC device.
0x20 . . . 0x7F	Error codes which are specific for every device. See appendix for definition.

Device Error Code

- Device Error Flag is not used in SFx6xxx, this flag is always zero.

Data

- The data has a usable size of [0. . . 255] bytes (original data, before byte stuffing). The meaning of the data content is defined in the command section.

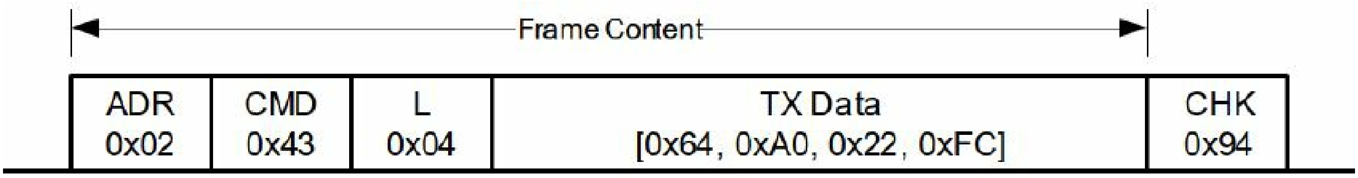
Checksum

The checksum is built before byte stuffing and checked after removing stuffed bytes from the frame. It is calculated 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.

Length

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



The checksum calculates as follows:

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

Protocol Definition

This chapter describes the frame communication protocol in a SHDLC system. There are some basic rules:

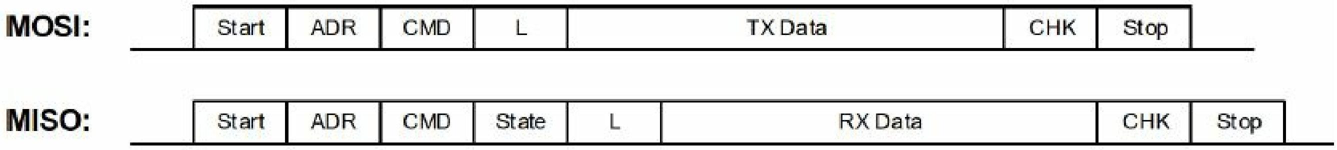
1. On every master request (MOSI frame), the slave will respond with a slave response (MISO frame). There are two exclusions where the slave will not send a response:
    - If the checksum of a MOSI frame did not match
    - If the MOSI frame was a broadcast
2. Between receiving a MOSI frame and sending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

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.

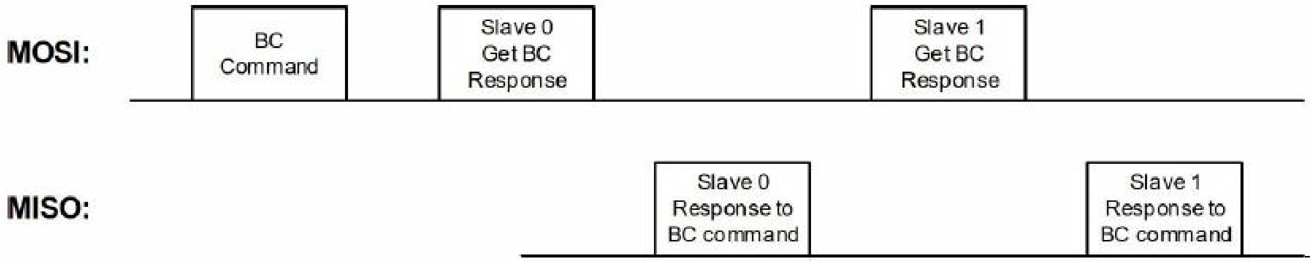
Standard Transfer

In this transfer, the master initiates a transfer with a MOSI frame containing command ID and up to 255 bytes of Tx data. After executing the command, the slave will respond with a MISO frame containing state and up to 255 bytes of Rx data. The command field of the frame is used for an application command. The transfer looks as follows:



Broadcast Transfer

After sending a broadcast command (ADR = 0xFF), the slave executes the command but does not send the generated response. The “Get Broadcast Response” frame allows the master to get the slave response on a previously sent broadcast command. The following shows an example with two slaves:

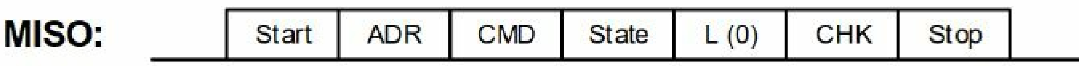


If the broadcast transfer is followed by a “Get Broadcast Response” frame, the slave will send the buffered answer. If any other frame is sent, the buffered response is discarded. The frame to get the broadcast response (MOSI) looks as follows and has to be addressed to a specific slave device:

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

Error Response

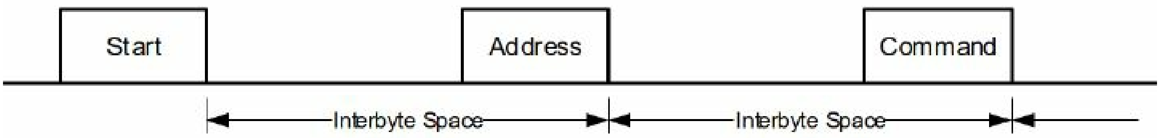
In case of a command execution error, the device will return an error response. This response is transmitted without data (L=0). That means that a simple error response looks alike for any transfer type:



Protocol Timings

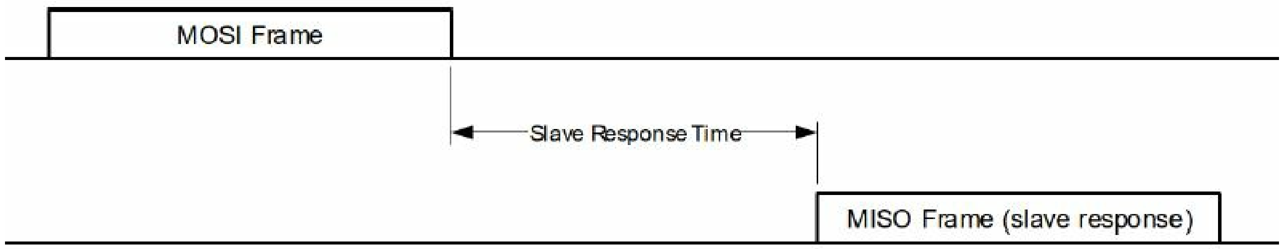
Interbyte Timeout

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 on slave side, the received data is discarded, and no response frame will be sent to the master.

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.
- The timeout should be at least twice the specified maximum response time of the corresponding command and in non real time systems it should not be smaller than 200ms due to possible system side delays.

Data Types and Representation

The data in the frames is transmitted in big-endian order (MSB first).

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 . . . 28-1
u16t	2 Byte	0 . . . 216-1
u32t	4 Byte	0 . . . 232-1
u64t	8 Byte	0 . . . 264-1
i8t	1 Byte	-27 . . . 27-1
i16t	2 Byte	-215 . . . 215-1
i32t	4 Byte	-231 . . . 231-1
i64t	8 Byte	-263 . . . 263-1

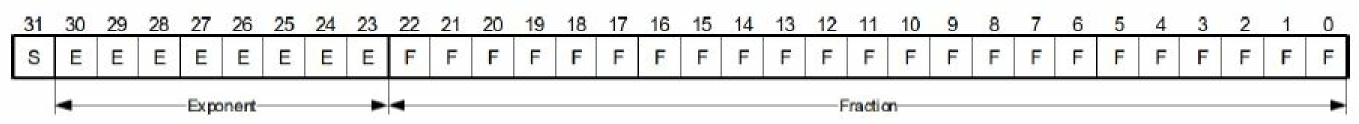
Boolean

A boolean is represented by 1 byte:

- False = 0
- True = 1. . . 255

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

String

Strings are 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. To make string handling as safe and robust as possible, it’s highly recommended to follow these rules when implementing an SHDLC master:

- When sending strings to the slave (MOSI data), always terminate the string with exactly one null-character (0x00). In case a fixed data length is specified for a particular string (e.g. if more MOSI parameters are following that string), fill all unused data bytes after the null-termination with 0x00 as well. But usually, string parameters are specified with a variable data length, i.e. no such padding is needed.
- When receiving strings from the slave (MISO data), parse the string up to the first null-character (0x00) in the received data and discard all the following data bytes (if any). In case no null-character is contained in the whole frame data, take the received data length into account to determine the string length, i.e. abort string parsing after the last data byte to accept strings which are not null-terminated.

Commands

[0x00] Setpoint

Get/Set Setpoint

<table>
<tr><td colspan="3">Get Setpoint</td></tr>
<tr><td>Description</td><td colspan="2">Get the current flow setpoint as a physical value.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x00</td></tr>
<tr><td>Max. Response Time</td><td colspan="2">10 ms</td></tr>
<tr><td>Post Processing Time</td><td colspan="2">0 ms</td></tr>
<tr><td rowspan="2">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Subcommand: uint8

Must be set to 0x01.</td></tr>
<tr><td rowspan="2">MISO Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0 . . .

3</td><td>Setpoint: float

The current setpoint.</td></tr>
</table>

<table>
<tr><td colspan="3">Set Setpoint</td></tr>
<tr><td>Description</td><td colspan="2">Set the flow setpoint as a physical value which is used by the flow controller as re ference input.
Note: the setpoint is set to 0 if calibration is changed.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x00</td></tr>
<tr><td>Max. Response Time</td><td colspan="2">10 ms</td></tr>
<tr><td>Post Processing Time</td><td colspan="2">0 ms</td></tr>
<tr><td rowspan="3">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Subcommand: uint8
Must be set to 0x01.</td></tr>
<tr><td>1 . . .
4</td><td>Setpoint: float
The new setpoint</td></tr>
<tr><td>MISO Data</td><td colspan="2">None</td></tr>
</table>

[0x08] Read Measured Value

Read Measured Value

The command returns the latest measured flow as physical value.

Read Measured Value		
Command ID	0x08	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x01.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Measured Value: float The latest measured flow.

Read Averaged Measured Value

- The command returns the average of given numbers of flow measurement as a physical value.
- A single measurement has a duration of 1ms, so the command response time depends on the given number of measurements to average.

Read Averaged Measured Value		
Command ID	0x08	
Max. Response Time	200 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x11.
	1	Measurements: uint8 number of measurements to average(1..100)
MISO Data	**Byte #**	**Description**
	0 . . . 3	Averaged Measured Value: float The averaged flow measurement.

[0x03] Set Setpoint and Read Measured Value

This command is a combination of the two commands “Set Setpoint (0x00)” and “Read Measured Value (0x08)”. It is intended for process data exchange (setpoint and flow) and saves a lot of protocol overhead compared to separate command usage.

Set Setpoint And Read Measured Value		
Command ID	0x03	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x01.
	1 . . . 4	Setpoint: float The new setpoint as physical value.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Measured Value: float The latest measured flow as physical value.

[0x22] Controller Configuration

- Get and set user settings of the flow controller.

Get/Set User Controller Gain

Get User Controller Gain		
Description	Get the user controller gain.	
Command ID	0x22	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x00.

Get User Controller Gain		
MISO Data	Byte #	Description
	0 . . . 3	Gain: float The current user controller gain.

Set User Controller Gain		
Description	Set the user controller gain. **Note** This configuration is not stored in non-volatile memory of the device and this not persists after a device reset.	
Command ID	0x22	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	Byte #	Description
	0	Subcommand: uint8 Must be set to 0x00.
	1 . . . 4	Gain: float The user controller gain to set.
MISO Data	None	

Get/Set User Init Ste

Get User Init Step		
Description	Get the user init step of flow controller	
Command ID	0x22	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x03.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Init Step: float The current user init step.

Set User Init Step		
Description	Set the init step of flow controller **Note** This configuration is not stored in non-volatile memory of the device, and this doe s not persist after a device reset.	
Command ID	0x22	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x03.
	1 . . . 4	Init Step: float the user init step to set.
MISO Data	None	

[0x30] Advanced Measurements

Advanced measurements for get more information about gas characteristics

Measure Raw Flow

- Return the measured raw flow ticks from the sensor.

Measure Raw Flow		
Command ID	0x30	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x00.
MISO Data	**Byte #**	**Description**
	0 . . . 1	Flow: uint16 Measured raw flow in ticks.

Measure Raw Thermal Conductivity With Closed Valve

- Perform a thermal conductivity measurement and return the measured raw tick value.
- The valve is automatically closed during the measurement.

Measure Raw Thermal Conductivity With Closed Valve		
Command ID	0x30	
Max. Response Time	600 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x02.
MISO Data	**Byte #**	**Description**
	0 . . . 1	Thermal Conductivity: uint16 Measured raw thermal conductivity in ticks.

Measure Temperature

- Return the temperature of flow sensor.

Measure Temperature		
Command ID	0x30	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x10.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Temperature: float Measured temperature [°C].

[0x40] Get Calibration Information

- This command allows to read information about all available gas calibrations on the device.

Get Number of Calibrations

- Get the number of calibrations, i.e. how many calibration the device memory is able to hold.

Note: Not all calibrations actually contain a valid gas calibration. Use the command Get Calibration Validity to check which calibrations are valid and thus can be activated.

Get Number Of Calibrations		
Command ID	0x40	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x00.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Number Of Calibrations: uint32 Number of calibrations.

Get Calibration Validity

- Check whether there exists a valid calibration at a specific index or not.

Get Calibration Validity		
Command ID	0x40	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x10.
	1 . . . 4	Index: uint32 The index to check whether there is a valid calibration or not.
MISO Data	**Byte #**	**Description**
	0	Validity: bool Whether there exists a valid calibration at the specified index or not.

Get Calibration Gas Id

- Get the gas ID of a specific calibration index.

Get Calibration Gas Id		
Command ID	0x40	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x12.
	1 . . . 4	Index: uint32 The calibration index to read the requested information from.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Gas Id: uint32 The read gas ID.

Get Calibration Gas Unit

- Get the gas unit of a specific calibration index.

<table>
<tr><td colspan="3">Get Calibration Gas Unit</td></tr>
<tr><td>Command ID</td><td colspan="2">0x40</td></tr>
<tr><td>Max. Response Time</td><td colspan="2">10 ms</td></tr>
<tr><td>Post Processing Time</td><td colspan="2">0 ms</td></tr>
<tr><td rowspan="3">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Subcommand: uint8
Must be set to 0x13.</td></tr>
<tr><td>1 . . .
4</td><td>Index: uint32
The calibration index to read the requested information from.</td></tr>
<tr><td rowspan="4">MISO Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Prefix: int8
Medium unit prefix, see appendix for encoding.</td></tr>
<tr><td>1</td><td>Unit: uint8
Medium unit, see appendix for encoding.</td></tr>
<tr><td>2</td><td>Timebase: uint8
Timebase, see appendix for encoding.</td></tr>
</table>

Get Calibration Full scale

- Get the full scale flow of a specific calibration index.

Get Calibration Fullscale		
Command ID	0x40	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x14.
	1 . . . 4	Index: uint32 The calibration index to read the requested information from.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Fullscale: float The read fullscale flow in the unit of the corresponding calibration.

[0x44] Get Current Calibration Information

- Returns calibration information of the current (active) calibration.

Get Current Gas Id

- Get the gas ID of the currently active calibration.

Get Current Gas Id		
Command ID	0x44	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x12.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Gas Id: uint32 The read gas ID.

Get Current Gas Unit

- Get the gas unit of the currently active calibration.

Get Current Gas Unit		
Command ID	0x44	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x13.
MISO Data	**Byte #**	**Description**
	0	Prefix: int8 Medium unit prefix, see appendix for encoding.
	1	Unit: uint8 Medium unit, see appendix for encoding.
	2	Timebase: uint8 Timebase, see appendix for encoding.

Get Current Full scale

- Get the full scale flow of the currently active calibration.

Get Current Full scale		
Command ID	0x44	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x14.
MISO Data	**Byte #**	**Description**
	0 . . . 3	Full scale: float The read full scale flow in the unit of the corresponding calibration.

[0x45] Get/Set Calibration

Get Calibration		
Description	This command returns the actual set calibration number of Flow controller.	
Command ID	0x45	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	None	
MISO Data	**Byte #**	**Description**
	0 . . . 3	Calibration Number: uint32 The current activated calibration number.

Set Calibration		
Description	This command stops the controller, changes the used calibration of the flow contr oller and starts the controller. The selected calibration is stored and also used aft er a power-on or reset. **Note:** Because of the limited write cycles of the flash, this command should not b e called periodically with new calibration (max. 50’000 times). If calibration is alre ady selected the function will not do a write cycle to flash.	
Command ID	0x45	
Max. Response Time	50 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0 . . . 3	Calibration Number: uint32 The calibration number to activate.
MISO Data	None	

[0x46] Set Calibration Volatile

- This command stops the controller, changes the used calibration of the flow controller and starts the controller.
- The selected calibration is not stored to non volatile memory.

Set Calibration Volatile		
Command ID	0x46	
Max. Response Time	20 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0 . . . 3	Calibration Number: uint32 The calibration number to activate.
MISO Data	None	

[0x90] Get/Set Slave Address

- This command changes the device address.

Get Slave Address		
Description	Gets the SHDLC slave address of the device.	
Command ID	0x90	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	None	
MISO Data	**Byte #**	**Description**
	0	Slave Address: uint8 The current slave address of the device.

<table>
<tr><th colspan="3">Set Slave Address</th></tr>
<tr><td>Description</td><td colspan="2">Sets the SHDLC slave address of the device.

Note: The slave address is stored in non-volatile memory of the device and thus persists after a device reset. So the next time connecting to the device, you have to use the new address. When changing the address of a slave, make sure there isn’t already a slave with that address on the same bus! In that case you would g et communication issues which can only be fixed by disconnecting one of the slaves.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x90</td></tr>
<tr><td>Max. Response Time</td><td colspan="2">50 ms</td></tr>
<tr><td>Post Processing Time</td><td colspan="2">0 ms</td></tr>
<tr><td rowspan="2">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Slave Address: uint8

The new slave address to set.

Allowed range: 0 . . . 254</td></tr>
<tr><td>MISO Data</td><td colspan="2">None</td></tr>
</table>

[0x91] Get/Set Baudrate

<table>
<tr><th colspan="3">Get Baudrate</th></tr>
<tr><td>Description</td><td colspan="2">Gets the SHDLC baudrate of the device.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x91</td></tr>
<tr><td>Max. Response Time</td><td colspan="2">10 ms</td></tr>
<tr><td>Post Processing Time</td><td colspan="2">0 ms</td></tr>
<tr><td>MOSI Data</td><td colspan="2">None</td></tr>
<tr><td rowspan="2">MISO Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0 . . . 3</td><td>Baudrate: uint32

Current baudrate in bit/s.</td></tr>
</table>

<table>
<tr><th colspan="3">Set Baudrate</th></tr>
<tr><td>Description</td><td colspan="2">Sets the SHDLC baudrate of the device.

Note: The baudrate is stored in non-volatile memory of the device and thus persists after a device reset. The next time you connect to the device, you have to use the new baudrate.</td></tr>
<tr><td>Command ID</td><td colspan="2">0x91</td></tr>
<tr><td>Max. Response Time</td><td colspan="2">50 ms</td></tr>
<tr><td>Post Processing Time</td><td colspan="2">0 ms</td></tr>
<tr><td rowspan="2">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0 . . . 3</td><td>Baudrate: uint32

The new baudrate in bit/s. Allowed values are 9600, 19200,

38400, 57600 and 115200 (default)</td></tr>
<tr><td>MISO Data</td><td colspan="2">None</td></tr>
</table>

[0xD0] Device Information

Get Product Type

Gets the product type from the device.

<table>
<tr><th colspan="3">Get Product Type</th></tr>
<tr><td>Command ID</td><td colspan="2">0xD0</td></tr>
<tr><td>Max. Response Time</td><td colspan="2">10 ms</td></tr>
<tr><td>Post Processing Time</td><td colspan="2">0 ms</td></tr>
<tr><td rowspan="2">MOSI Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0</td><td>Subcommand: uint8

Must be set to 0x00.</td></tr>
<tr><td rowspan="2">MISO Data</td><td>Byte #</td><td>Description</td></tr>
<tr><td>0 . . .

n</td><td>Product Type: string

String containing the product type.</td></tr>
</table>

Get Product Name

Get Product Name		
Description	Gets the product name from the device.	
Command ID	0xD0	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x01.
MISO Data	**Byte #**	**Description**
	0 . . . n	Product Name: string String containing the product name.

Get Article Code

Get Article Code		
Description	Get the article code of the device. This information is also contained on the produ ct label.	
Command ID	0xD0	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x02.
MISO Data	**Byte #**	**Description**
	0 . . . n	Article Code: string The article code as an ASCII string.

Get Serial Number

Get Serial Number		
Description	Gets the serial number of the SFC6xxx sensor.	
Command ID	0xD0	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	**Byte #**	**Description**
	0	Subcommand: uint8 Must be set to 0x03.
MISO Data	**Byte #**	**Description**
	0 . . .n	Serial Number: string String containing the serial number of the SFC6xxx sensor.

[0xD1] Get Version

Gets the version information for the hardware, firmware and SHDLC protocol.

Get Version		
Command ID	0xD1	
Max. Response Time	10 ms	
Post Processing Time	0 ms	
MOSI Data	None	
MISO Data	**Byte #**	**Description**
	0	Firmware Major: uint8 Firmware major version number.
	1	Firmware Minor: uint8 Firmware minor version number.
	2	Firmware Debug: bool Firmware debug state. If the debug state is set, the firmware is in development.
	3	Hardware Major: uint8 Hardware major version number.
	4	Hardware Minor: uint8 Hardware minor version number.
	5	Protocol Major: uint8 Protocol major version number.
	6	Protocol Minor: uint8 Protocol minor version number.

[0xD3] Device Reset

- Executes a reset on the device. This has the same effect as a power cycle.

Note: The device will reply before executing the reset.

Device Reset	
Command ID	0xD3
Max. Response Time	100 ms

- **Post Processing Time:** 300 ms
- **MOSI Data:** None
- **MISO Data:** None

Appendix

Gas unit decoding

The following tables show the encoding of the gas unit:

Prefix

Prefix Code (i8t)	**Prefix**	**Symbol**	**10ˆn**
-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			100
1	deca	da	101
2	hecto	h	102
3	kilo	k	103
6	mega	M	106
9	giga	G	109
12	tera	T	1012
15	peta	P	1015
18	exa	E	1018
21	zetta	Z	1021
24	yotta	Y	1024
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	–

State Response Error Codes

The state response error codes describe the meaning of the lower 7 bits of the 'state' byte which is returned in each MISO frame. See Section 5.6.1 for details.

Code		Name	Description
0	0x00	Success	Command successfully executed (no error occurred).
1	0x01	Data Size Error	Illegal data size of the MOSI frame. Either an invalid frame was sent or the firmware on the device does not support the requested feature.
2	0x02	Unknown Command Error	The device does not know this command. Either an invalid command ID was sent or the firmware on the device doesn't support it.
4	0x04	Parameter Error	A sent parameter is out of range. Check if you sent the correct command parameters and if the firmware on the device supports them.
41	0x29	I2c Nack Error	NACK received from I2C device.
42	0x2A	I2c Master Hold Error	Master hold not released in I2C.
43	0x2B	I2c Crc Error	I2C CRC mismatch.
44	0x2C	Sensor Data Write Error	Sensor data read back differs from written value.
45	0x2D	Sensor Measure Loop Not Running Error	Sensor measure loop is not running or runs on wrong gas number.
51	0x33	Invalid Calibration Index Error	No valid gas calibration at given index.
66	0x42	Sensor Busy Error	The sensor is busy at the moment. Please note that the power-up time after a reset is 300ms.
67	0x43	Command Not Allowed In Current State	Command is not allowed in the current state.
127	0x7F	Fatal Error	An error without specific error code occurred.

Revision History

Date	Version	Pages	Changes
Mar 2023	1.0	all	Initial version
Sept 2023	1.1	6	Addendum regarding default address of device added

Documents / Resources

	SENSIRION SFC6 Mass Flow Controller [pdf] User Manual SFC6 Mass Flow Controller, SFC6, Mass Flow Controller, Flow Controller, Controller

References

- User Manual