

Description :	Nicolay flow meter connector with pressure sensor communication protocol		SP#: SP_290322000_cp
Owner (Name) :	PE-BF	Date : 08.12.2025	Nicolay P/N : 290322000
Approver (Name) :	PE-BD	Date : 08.12.2025	Rev: B00

Table of Contents

1. Introduction	2
2. Interfaces Basics	2
3. Addressing	2
4. Pseudocode CRC: CRC-8	3
6. Errors	4
6.1. Transmission Errors	4
6.2. Exception Errors	4
6.2.1.Exception Errors Codes:	4
7. Function codes	4
7.1. Get SW_Version (1)	4
7.2. Get HW_Version (2)	4
7.5. Test command (5)	5
7.6. Get Pressure Sensor Value (6)	5
7.7. Get Pressure (7)	6
7.9. Get Flow and Pressure (9)	6
7.10. Get Sensor Product identifier (10)	6
7.11. Board Hardware Reset (11)	7
7.12. Sensor Hard Reset (12)	7
7.13. Sensor Soft Reset (13)	7
7.15. Get Sensor Serial No (15)	8
7.16. Get Flow Measurement (16)	8
7.17. Get Raw Flow Measurement (17)	8
7.18. Get Flow Sensor Scale (18)	9
7.19. Get Flow Sensor Offset (19)	9
7.20. Get/Set Heater State (20)	9
7.21. Get/Set Heater Power (21)	10
7.22. Get Flow Temperature (22)	10
7.27. Force Temperature Update (27)	10
7.29. BulkRead (29)	11
7.30. STREAM_send (30)	11
7.34. Get/Set UartBaud (34)	11
7.38. Get Sensor Status (38)	12
7.40. Get/Set Flow Signal Filter (40)	12
7.41. Get/Set N averaging (41)	13

1. Introduction

This document describes the communication protocol for the Nicolay flow meter connector with pressure sensor and RS485 or RS232 interface for activation the Sensirion flow meters with the type designation:

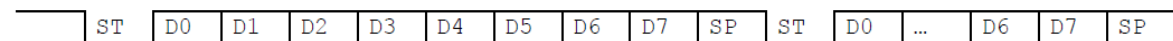
- SFM3200-AW
- SFM3300-AW
- SFM3300-D
- SFM3304-D
- SFM3400-AW
- SFM3400-D

Note: the circuit board and flow meter connector is not marked by Nicolay GmbH as a medical device / medical device software, but only as a demonstration version. The labeling is the responsibility of the distributor of the medical device.

The physical connection is made via a serial RS485 or RS232 interface. This guarantees good immunity to interference and provides a flexible bus configuration, multiple devices (slaves) can be controlled by a master. The interface is operated in half-duplex which minimizes the wiring effort. In order to connect several devices with each other, these are simply connected in parallel (RS485A, RS485B, ...). Each device has its own address and it is possible to create a network with a maximum of 32 participants.

2. Interfaces Basics

Mode: RTU Mode (Remote, Terminal Unit)
Coding System: 8-bit binary, (hexadecimal 0-9, A-F)
Bits per Byte: 1 start bit
8 data bits, least significant bit sent first
0 parity bit: none
1 stop bit
Error Check: 1 Byte Cyclical Redundancy Check (CRC8)
Baudrate: 115.200 (default baudrate, change with command 34)
Bit frame:



ST: start bit, SP: stop bit. A parity bit (if active) is inserted before the SP, D0... D7: 8 data bits

Termination:

Layout: between A and B at the beginning and end of the transmission line.

Value: the same as the line impedance typically 120 ohms.

At least one termination resistor is required to ensure stable communication (in the master RS485/RS232 driver)!

In the case of a fail-safe-master driver and a trouble-free environment, the terminator is not mandatory if the cable is kept short (a few meters).

In order to reduce current peaks, the 2nd resistance value can be selected higher (1kOhm) or even omitted (during the transmission, the current at 2 x 1200 Ohm is approx. 50mA).

Fail safe biasing resistors:

Fail safe Bias resistors ensure defined levels – even in the event of a short circuit or interruption. This is very important for half duplex operation when all devices are in receive mode – here the line is open in case there are no bias resistors on the master.

To obtain always defined voltages, a pull-up resp. Pull-down resistor connected to the master: From A to +5V and from B to GND (2x101kOhm). This is always necessary when using such RS485/RS232 drivers that are not fail-safe.

3. Addressing

General:

Each data exchange is initiated by a master, a slave may only send if requested by the master. An address can only be assigned once for a slave on the same bus, if two devices have the same address both would respond. This leads to a conflict on the bus line.

If a transmission interruption occurs between two bytes that is longer than 1.5 bytes, the telegram is judged as incomplete and rejected (typically 200µs @115.200 baud). Between two transmissions to different addresses, there must be added 50µs to the 1.5 bytes in order to switch the RS485 driver. This would be typically 250µs @115.200 Baud.

Addressing master:

The master has no address. For possible multi-master operation (or special applications), the 7th bit is reserved in the address.

Addressing slaves:

Each slave can be addressed via the slave address (DevAddr). Slave addresses up to 250 are possible. Default address for slave is "1". The slave address "0" is reserved for general call of the master. The slave address could be changed for each device.

Note: at address 255 all slaves generally will response with their individual address (identifying a slave). This address is not allowed if there are more than one slave on the bus.

When slave address "0" (general call) no response from the slaves may take place.

Example for general call:

Function code that is regularly sent by the master, so the slaves know that the master works.

A general call follows the standard frame. Function code and data can be sent.

If neither function code nor data is sent along, the function code '00' must be sent along, the check sum is mandatory, a response from the slave does not occur.

Dokument : K20-2-1_Product Specification.docx	Datum : 2016-01-26	Ersteller : R.Gann
Version : 1	Seite : 2 von 13	Prüfer : C.Muz

4. Pseudocode CRC: CRC-8

Polynomial: $x^8 + x^5 + x^4 + 1$ (0x31)
Initial value: 0x0000

```

crc == (crc Xor ByteVal)
for (i= 0 ;i= 7;i=i+1){
    crc = (CRC Shiftleft 1)
    if Bit(crc, 8){
        crc = (crc Xor CRC_Polynom)
        crc = (crc And &h0FF)
    }
}
Return (CRC)

```

TABLE Driven:

0x00 ,0x31 ,0x62 ,0x53 ,0xC4 ,0xF5 ,0xA6 ,0x97 ,0xB9 ,0x88 ,0xDB ,0xEA ,0x7D ,0x4C ,0x1F ,0x2E
 0x43 ,0x72 ,0x21 ,0x10 ,0x87 ,0xB6 ,0xE5 ,0xD4 ,0xFA ,0xCB ,0x98 ,0xA9 ,0x3E ,0x0F ,0x5C ,0x6D
 0x86 ,0xB7 ,0xE4 ,0xD5 ,0x42 ,0x73 ,0x20 ,0x11 ,0x3F ,0x0E ,0x5D ,0x6C ,0xFB ,0xCA ,0x99 ,0xA8
 0xC5 ,0xF4 ,0xA7 ,0x96 ,0x01 ,0x30 ,0x63 ,0x52 ,0x7C ,0x4D ,0x1E ,0x2F ,0xB8 ,0x89 ,0xDA ,0xEB
 0x3D ,0x0C ,0x5F ,0x6E ,0xF9 ,0xC8 ,0x9B ,0xAA ,0x84 ,0xB5 ,0xE6 ,0xD7 ,0x40 ,0x71 ,0x22 ,0x13
 0x7E ,0x4F ,0x1C ,0x2D ,0xBA ,0x8B ,0xD8 ,0xE9 ,0xC7 ,0xF6 ,0xA5 ,0x94 ,0x03 ,0x32 ,0x61 ,0x50
 0xBB ,0x8A ,0xD9 ,0xE8 ,0x7F ,0x4E ,0x1D ,0x2C ,0x02 ,0x33 ,0x60 ,0x51 ,0xC6 ,0xF7 ,0xA4 ,0x95
 0xF8 ,0xC9 ,0x9A ,0xAB ,0x3C ,0x0D ,0x5E ,0x6F ,0x41 ,0x70 ,0x23 ,0x12 ,0x85 ,0xB4 ,0xE7 ,0xD6
 0x7A ,0x4B ,0x18 ,0x29 ,0xBE ,0x8F ,0xDC ,0xED ,0xC3 ,0xF2 ,0xA1 ,0x90 ,0x07 ,0x36 ,0x65 ,0x54
 0x39 ,0x08 ,0x5B ,0x6A ,0xFD ,0xCC ,0x9F ,0xAE ,0x80 ,0xB1 ,0xE2 ,0xD3 ,0x44 ,0x75 ,0x26 ,0x17
 0xFC ,0xCD ,0x9E ,0xAF ,0x38 ,0x09 ,0x5A ,0x6B ,0x45 ,0x74 ,0x27 ,0x16 ,0x81 ,0xB0 ,0xE3 ,0xD2
 0xBF ,0x8E ,0xDD ,0xEC ,0x7B ,0x4A ,0x19 ,0x28 ,0x06 ,0x37 ,0x64 ,0x55 ,0xC2 ,0xF3 ,0xA0 ,0x91
 0x47 ,0x76 ,0x25 ,0x14 ,0x83 ,0xB2 ,0xE1 ,0xD0 ,0xFE ,0xCF ,0x9C ,0xAD ,0x3A ,0x0B ,0x58 ,0x69
 0x04 ,0x35 ,0x66 ,0x57 ,0xC0 ,0xF1 ,0xA2 ,0x93 ,0xBD ,0x8C ,0xDF ,0xEE ,0x79 ,0x48 ,0x1B ,0x2A
 0xC1 ,0xF0 ,0xA3 ,0x92 ,0x05 ,0x34 ,0x67 ,0x56 ,0x78 ,0x49 ,0x1A ,0x2B ,0xBC ,0x8D ,0xDE ,0xEF
 0x82 ,0xB3 ,0xE0 ,0xD1 ,0x46 ,0x77 ,0x24 ,0x15 ,0x3B ,0x0A ,0x59 ,0x68 ,0xFF ,0xCE ,0x9D ,0xAC

For details regarding cyclic redundancy checking please refer to the Sensirion Application Note SFM3000 CRC Checksum.

5. Frame structure

Frame program of a master

Device address	Function code	Number of following data	n-Data	CRC- 8
8 Bit	0 7Bit	8 Bit	nByte(s)	8 Bit
8 Bit	8 Bit	8 Bit	n x 8 Bit	8 Bit

- Device address: address of the slave
address 0 is reserved for broadcasting
address 250-254 is reserved for development
address 255 generally responses with the individual slave address
- Function code: command number
A function is selected and executed on the device. The function code is decoded with 7 bit (bit 7 is always 0).
- Number of data: determines the number of user data without CRC and header (follower).
- n-Data: Attached user data (parameters), the number was previously in the field defined by follower. For values > 8 bits, they follow the format Low / High.
- CRC8: 8-bit checksum. In case of error, the transmission is discarded.

Note: the minimum length of a transmission from the master is at least 4 bytes.

Frame of the answer from the slave

Device address	Function code	Number of following dates	n-Data	CRC- 8
8 Bit	X 7Bit	8 Bit	nByte(s)	8 Bit
8 Bit	8 Bit	8 Bit	n x 8 Bit	8 Bit

- Device address: Address of the slave, this belongs to the answering device.
address 255 generally responses with the individual slave address
- Function code: The command number is identical to that sent by the master.
If the bit 7 (most significant bit) X=0 the function has been executed, bit X = 1 is an exception error occurred.
- Number of data: determines the number of data without CRC and headers that are returned. The maximum number is limited to the buffer of the device.
- n-Data: Data requested by the master will follow here. The number was previously defined in the follower field. For values > 8 bits, they follow the format Low / High.
- CRC8: 16-bit checksum. In case of error, the transmission is discarded.

Dokument : K20-2-1_Product Specification.docx	Datum : 2016-01-26	Ersteller : R.Gann
Version : 1	Seite : 3 von 13	Prüfer : C.Muz

6. Errors

6.1. Transmission Errors

- A received transmission is too short, too long time has passed between the individual bytes, the transmission has already been discarded by the receiver.
- A transmission is too long and exceeds the buffer size of the receiver, the checksum does not arrive.
- The CRC8 checksum is wrong.

In these cases the transmission will not be answered by the slave. This creates a timeout on the master, which then repeats the request.

All data contained in these programs are discarded, the slave remains in receive mode.

6.2. Exception Errors

The transmission was received correctly (no transmission error occurred), but the function number and / or transferred parameters are invalid:

Device address	Function code	Number of following dates (follower)	Exception Code	CRC 8
8 Bit	1 7Bit	0x01	8-bit code	8Bit

6.2.1. Exception Errors Codes:

- 1 Function code is unknown, is not supported
- 2 Device cannot start firmware, stays in bootloader mode
- 3 Device is in the initialization phase and cannot process request
- 4 Device cannot process request at present, is busy
- 5 Followers, number of dates is wrong, too much / too less
- 6 The number of requested data is too much / too little (buffer overflow/underflow)
- 7 Subcode too big / too small / not supported
- 8 Date too big / too small (plausibility of the date to be set)
- 9 No ACK in read / write from Sensor_EEPROM
- 10 TimeOut in read / write from / to Sensor_EEPROM
- 11 CheckSum from generic I2C_CMD not valid

- 15 Sensor is in Shutdown mode HW-Reset is required
- 16 An attempt is made to import an UpDate, but the bootloader is not started
- 17 The checksum in HexLine is wrong, probably HexFile corrupt.
- 18 syntax error: the new line in the UpDate process does not start with ":"

7. Function codes

Note: description in the format high-low, must be sent low-high.

Note: command 38,40 and 41 are only for the sensor SFM 3304-D.

7.1. Get SW_Version (1)

Request:

DevAddr	0x01	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x01	0x03	Index as ASCII	Firmware version minor	Firmware version major	CRC 8
---------	------	------	----------------	------------------------	------------------------	-------

Get Software Version (Firmware of Cable)

Returns 3 Bytes: Index, Minor Version, Major Version

Example: 61,63,00 -> Firmware version 0.99a

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device is busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.2. Get HW_Version (2)

Request:

DevAddr	0x02	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x02	0x02	Hardware version minor	Hardware version major	CRC 8
---------	------	------	------------------------	------------------------	-------

2x UnsignedByte

MajorHw. MinorHW ->Hardware version 12.34

Dokument : K20-2-1_Product Specification.docx	Datum : 2016-01-26	Ersteller : R.Gann
Version : 1	Seite : 4 von 13	Prüfer : C.Muz

Example: 00,02 -> Hardware version 2.00

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.5. Test command (5)

Request:

DevAddr	0x05	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x05	0x02	0x55	0xAA	0x7D
---------	------	------	------	------	------

Defined test command

On host request [0x01], 0x05, 0x00, 0x31

follows the answer with [0x01], 0x05, 0x02, 0x55, 0xAA and ChkSum 0x7D

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.6. Get Pressure Sensor Value (6)

Request:

DevAddr	0x06	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x06	0x09	Type	Min Pressure Low	Min Pressure High	Max Pressure Low	Max Pressure High	digOut-Pmin Low	digOut-Pmin High	digOut-Pmax Low	digOut-Pmax High	CRC 8

Type UByte

- (0) = "NONE"
- (1) = "AMS5915_0005_D"
- (2) = "AMS5915_0005_D_B"
- (3) = "AMS5915_0010_D"
- (4) = "AMS5915_0010_D_B"
- (5) = "AMS5915_0020_D"
- (6) = "AMS5915_0020_D_B"
- (7) = "AMS5915_0050_D"
- (8) = "AMS5915_0050_D_B"
- (9) = "AMS5915_0100_D"
- (10) = "AMS5915_0100_D_B"
- (11) = "AMS5915_0200_D"
- (12) = "AMS5915_0200_D_B"
- (13) = "AMS5915_0350_D"
- (14) = "AMS5915_0350_D_B"
- (15) = "AMS5915_1000_D"
- (16) = "AMS5915_1000_D_B"
- (17) = "AMS5915_2000_D"
- (18) = "AMS5915_4000_D"
- (19) = "AMS5915_7000_D"
- (20) = "AMS5915_10000_D"
- (21) = "AMS5915_1000_A"
- (22) = "AMS5915_1200_B"

Min Pressure = 16bit Signed Integer (xFF38 = -200dez)

Max Pressure = 16bit Signed Integer (x00C8 = +200dez)

e.g.: RawPressure = 0x12FFD → (((tRAW_PRESSURE – digOut PMin))/(((digOut PMax – digOut PMin)))/((pMax – pMin))+pMin) = -0,08 mbar (AMS5915_0200_D_B)

digOut PMin = 16bit Signed specified minimum pressure Counts

digOut PMax = 16bit Signed specified maximum pressure Counts

note: Refer to ams5915-datasheet

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

Dokument : K20-2-1_Product Specification.docx	Datum : 2016-01-26	Ersteller : R.Gann
Version : 1	Seite : 5 von 13	Prüfer : C.Muz

7.7. Get Pressure (7)

Request:

DevAddr	0x07	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x07	0x02	Pressure low	Pressure high	CRC 8
---------	------	------	--------------	---------------	-------

16-Bit (Integer)_pressure counts

Note: the digital output values for pressure is only 14 bit wide, the 2 higher bits are masked out and always zero
Note: refer to ams5915-datasheet

The current pressure in bar (or PSI) is calculated from the digital pressure value using the following formulas:

$$p = \frac{Digout(p) - Digout_{min}}{Sensp} + p_{min} \quad \text{with} \quad Sensp = \frac{Digout_{max} - Digout_{min}}{p_{max} - p_{min}} \quad (1)$$

Therein p is the current pressure in bar (or PSI), p_{min} is the specified minimum pressure and p_{max} is the specified maximum pressure in bar (or PSI); depending on the specified pressure range, $Digout(p)$ is the current digital 14-bit pressure value in counts, $Digout_{min}$ and $Digout_{max}$ are the digital pressure values at minimum and maximum specified pressure in counts and $Sensp$ is the sensitivity of the pressure sensor in counts/bar (or counts/PSI).

with:

$Digout_{max} = 14745$ counts

$Digout_{min} = 1638$ counts

$p_{max} = \text{max. pressure of sensor type}$

$p_{min} = \text{min. pressure of sensor type}$

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.9. Get Flow and Pressure (9)

Request:

DevAddr	0x09	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x09	0x06	Flow8	Flow16	Flow24	Flow32	Raw Pressure Low	Raw Pressure High	CRC 8
---------	------	------	-------	--------	--------	--------	------------------	-------------------	-------

32-Bit (Signed Integer) + Pressure: 16-Bit (Signed Integer)
Combined Command to read flow and pressure at once

Flow: 32-Bit (Signed Integer)

Note: Refer to Command16 Get_FlowMeasurement for further explanation

RawPressure: 16-Bit (Signed Integer)

Note: Refer to Command 7 Get_Pressure for further explanation

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.10. Get Sensor Product identifier (10)

Request:

DevAddr	0x0A	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x0A	0x04 SFM3304-D: 0x0C	Nr. low	Nr. mid	Nr. mid	Nr. High	CRC 8
---------	------	-------------------------	---------	---------	---------	----------	-------

32 bit answer corresponds to Sensirion product identifier.

Bits <27:8> provide the product identifier.

- SFM3200-AW: 0x18ABA
- SFM3300-AW: 0x18ABC
- SFM3400-AW: 0x18CA9
- SFM3300-D: 0x18ABD
- SFM3400-D: 0x18CEB
- SFM3304-D: 0x40501

Note: the cable outputs the number that was read upon start up and does not perform a new read to the sensor. A new read to the sensor can be performed if:

- the internal plausibility check identifies an invalid value (0x0 or 0xFF)
- an argument to the "get" function is added.

When a new read is requested, the measurement is stopped and restarts automatically. This takes approx. 4 ms during which no other command to the sensor can be sent. You must not use this command in a loop.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.11. Board Hardware Reset (11)

Request:

DevAddr	0x0B	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x0B	0x00	CRC 8
---------	------	------	-------

Performs a complete reset / restart of the processor and sensor.
The (flow) measurement is restarted automatically.

Exception Errors:

- 2 If device has no Firmware

7.12. Sensor Hard Reset (12)

Request:

DevAddr	0x0C	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x0C	0x00	CRC 8
---------	------	------	-------

Performs a complete hard reset of the sensor
The supply of the sensor is switched off and automatically switched on again.
The (flow) measurement is restarted automatically.

Exception Errors:

- 2 If device has no Firmware

7.13. Sensor Soft Reset (13)

Request:

DevAddr	0x0D	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x0D	0x00	CRC 8
---------	------	------	-------

Performs a softreset of the Flowsensor.
The flow-measurement is restarted automatically.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

Note: This takes approx. 4ms. Due to the behaviour of the sensor, two consecutive writes or reads with this command within 4 ms may trigger a device busy exception

Dokument : K20-2-1_Product Specification.docx	Datum : 2016-01-26	Ersteller : R.Gann
Version : 1	Seite : 7 von 13	Prüfer : C.Muz

7.15. Get Sensor Serial No (15)

Request:

DevAddr	0x0F	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x0F	0x04 / 0x08	Nr. low	Nr. mid	Nr. mid	Nr. High	CRC 8
---------	------	-------------	---------	---------	---------	----------	-------

32-Bit (unsigned Integer)

64-Bit (unsigned Integer) at SFM3304-D

The SFM3304-xxx-D has a 64bit serial number so if received 8bytes then it is this one.

Returns the serial number of the sensor

Note: if this is 0xFFFFFFFF, the sensor is unreadable.

Note: the cable outputs the number that was read upon start up and does not perform a new read to the sensor. A new read to the sensor can be performed if:

- the internal plausibility check identifies an invalid value (0x0 or 0xFF)
- an argument to the "get" function is added.

When a new read is requested, the measurement is stopped and restarts automatically. This takes approx. 4 ms during which no other command to the sensor can be sent. You must not use this command in a loop.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.16. Get Flow Measurement (16)

Request:

DevAddr	0x10	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x10	0x04	Nr. low	Nr. mid	Nr. mid	Nr. High	CRC 8
---------	------	------	---------	---------	---------	----------	-------

32-Bit (Signed Integer)

From the board calculated value in mSlm² (milli-standard-liters per minute). A division by 1000 gives slpm.

Always the last valid sensor reading is returned.

Note: If value permanently 0x7FFFFFFF (2147483647), the Sensor is not readable.

A Hardreset of the sensor should be performed. If Autoreset (enabled in Config by default) is enabled this would be done automatically but should be monitored if it takes effect.

Note: for performance reason, the onboard calculation could be shut of (in Config).

Response time: 259.5 µs @ 576000

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.17. Get Raw Flow Measurement (17)

Request:

DevAddr	0x11	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x11	0x02	Raw Low	Raw High	CRC 8
---------	------	------	---------	----------	-------

16-Bit (Integer) raw measured value of the sensor

The last valid sensor value is returned.

Note:

If value permanently 0xFFFF (65535), the Sensor is not readable.

A Hardreset of the Sensor should be performed. If Autoreset (enabled in Config by default) is enabled this would be done automatically.

Note: The sensor SFM3304-D output has been scaled to match the output of the other sensors SFM3x00-D/AW:

zero flow is 32768

-1 is 32767

+1 is 32769

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.18. Get Flow Sensor Scale (18)

Request:

DevAddr	0x12	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x12	0x02	Scale low	Scale High	CRC 8
---------	------	------	-----------	------------	-------

Note: This reflect the internal readed number at StartUp, it does not perform a new read to the sensor.
16-Bit (U-Integer) scaling factor of the sensor
returns the scale value as a basis for calculating the flow

Note: if reading is 0xFFFF, a Hardreset of the sensor should be performed.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.19. Get Flow Sensor Offset (19)

Request:

DevAddr	0x13	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x13	0x02	Offset low	Offset High	CRC 8
---------	------	------	------------	-------------	-------

Note: This reflect the internal readed number at StartUp, it does not perform a new read to the sensor.
16-Bit (unsigned) sensor Offset
returns the offset value as a basis for calculating the flow

Note: if reading is 0xFFFF, a Hardreset of the sensor should be performed.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.20. Get/Set Heater State (20)

Request: GET

DevAddr	0x14	0x00	CRC 8
---------	------	------	-------

Request: SET

DevAddr	0x14	0x01	[Bit 0 heating 0=OFF 1=ON]	CRC 8
---------	------	------	----------------------------	-------

Switches heating on / off to the value defined with command Set_HeaterPower (default 100%).

Note: Refer to command 21 Set_Heater_Power

Note: the heater has a soft start function, so it may takes up to 100ms to reach full power.

Response:

DevAddr	0x14	0x01	State [Bit 0 heating 0=OFF 1=ON]	CRC 8
---------	------	------	----------------------------------	-------

8-Bit (unsigned)

Returns the status of the heating [Bit 0 heating 0=OFF (default); 1=ON]

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.21. Get/Set Heater Power (21)

Get Request:

DevAddr	0x15	0x00	CRC 8
---------	------	------	-------

Set Request:

DevAddr	0x15	0x01	HeatPower 0 –100%	CRC 8
---------	------	------	-------------------	-------

Sets heat output in bytes value in percent.

Does not turn heating on or off, refer to Command 20

Note: Please consider it is recommended to only apply sufficient power for the intended scenario, e.g. to avoid condensation, and not to continuously set the heater to 100%.

Note: the heater is regulated (if enabled) and will reduce or increase power by itself, but never increase more than the heater power set.

Bit 1 = Regulated Heater (default 1) switches regulation on/off

Note: values greater than 100 are not allowed, this could result in unexpected heater behaviour

Response:

DevAddr	0x15	0x01	HeatPower 0 –100%	CRC 8
---------	------	------	-------------------	-------

8-Bit (unsigned)

Returns the used value in 0-100%

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.22. Get Flow Temperature (22)

Request:

DevAddr	0x16	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x16	0x02	low scale	high scale	CRC 8
---------	------	------	-----------	------------	-------

Returns last calculated (chip-) temperature in °C, resolution in 0,01°C (divided by 100 provides °C)

Note: this command delivers only valid value if autoreadtemp in Config is enabled (default is disabled) depending the UpDateRate.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.27. Force Temperature Update (27)

Request:

DevAddr	0x1B	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x1B	0x02	Temperature low	Temperature high	CRC 8
---------	------	------	-----------------	------------------	-------

(16-Bit Signed fixedpoint integer)

Triggers a temperature measurement and provides calculated (chip) temperature in 0,01°C regardless of the update rate and allowed calculation.

Resolution in 0,01 °C (divided by 100 provides °C).

Note: Also updates the raw value of the temperature.

Note: a read to the sensor stalls time to reenale the Flow sensor. The next read of flow measurement could be significantly longer than the 0.5ms after read of the temperature from sensor.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.29. BulkRead (29)

Request:

DevAddr	0x1D	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x1D	0xC8/0xFF	Dat0	...	Dat200/400	CRC 8
---------	------	-----------	------	-----	------------	-------

Returns buffered RawFlow and Pressure (if exist) value.
This is a read back of the last 100 values stored in memory in FiFo order.

If Pressure sensor not exist
Follower = 200 (0xC8) FlowRawL; FlowRawH in row.

If Pressure sensor exist
Follower = 255 (0xFF) FlowRawL; FlowRawH; PressureL; PressureH in row.
Note: in case of Pressure sensor exist the following Data are 400Bytes + ChkSum

Note: each readed Byte will mark with an 0x00, so if you see those Zeros your Reading speed ist o fast
If Sensor request gives an Error for some reason the Value is marked with a 0x0FF

Exception Errors:
2 If device has no Firmware
3 If device has not yet been initialised
4 If device busy
15 Sensor is in Shutdown mode, HW-reset is required

7.30. STREAM_send (30)

Request:

DevAddr	0x1E	0x00	CRC 8
---------	------	------	-------

Response:

Flow8	Flow16	Flow24	Flow32	(PressureL)	(PressureH)	0xFF	0x03 (ETX)
-------	--------	--------	--------	-------------	-------------	------	------------

Starts sending immediately with no HeaderFrame beforehand
Delivers continuous Datastream without new request, where Flow8-32 represents the calculated Flow and Pressure the RawPressure.
To synchronize, each Packet will end with 0xFF,0x03 (ETX)
IF PressureSensor not Exist Packet length is shortened by those two Bytes

Note: Each Byte received on TX will stop sending immediately, regardless of the value or matching Address. Is this a valid command, it may answered or not.
This is recommended for use with Rs232 or onboard USB only.

Exception Errors:
2 If device has no Firmware
3 If device has not yet been initialised
4 If device busy
15 Sensor is in Shutdown mode, HW-reset is required

7.34. Get/Set UartBaud (34)

Request:

DevAddr	0x22	0x01	Baudcode	CRC 8
---------	------	------	----------	-------

8-bit Uint

- 0= 4.800 (4799) (-0.20)
- 1= 9.600(9603.84) (0.040%)
- 2= 14400(14388) (-0.08%)
- 3= 19.200 (19184.65) (-0.080%)
- 4= 28800 (28776) (-0,8%)
- 5 = 31250 (31250) (0,00%)
- 6 = 38400 (38461) (0.16%)
- 7= 57.600 (57553.96) (-0,080%)
- 8= 115.200 (115942.03) (0,644%) [Factory Default]
- 9= 128000(126984) (-0.79%)
- 10= 230.400 (228571.43) (-0,794%)
- 11= 250000 (250000) (0.0%)
- 12= 256000 (258064) (0.81%)

Dokument : K20-2-1_Product Specification.docx	Datum : 2016-01-26	Ersteller : R.Gann
Version : 1	Seite : 11 von 13	Prüfer : C.Muz

13= 384000 (380952) (-0.79%)
 14= 500000 (500000) (0.0%)
 15= 576.000(571428.57) (-0,794%)

Response:

DevAddr	0x22	0x01	Baudcode	CRC 8
---------	------	------	----------	-------

8-bit UByte

Note: the response after changing Baudrate will be done at previous Rate
 After sending the response the Baudrate changes immediately
 Standard after PowerOn is 115.200

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.38. Get Sensor Status (38)

Request:

DevAddr	0x26	0x00	CRC 8
---------	------	------	-------

Response:

DevAddr	0x26	0x2	StatLo	StatHi	CRC 8
---------	------	-----	--------	--------	-------

Returns the status of the Sensor. Only for the Sensor SFM3304-D.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.40. Get/Set Flow Signal Filter (40)

Request:

DevAddr	0x28	0x1 / 0x0	FilterVal	CRC8
---------	------	-----------	-----------	------

Response:

DevAddr	0x28	0x1	FilterVal	CRC8
---------	------	-----	-----------	------

Set time of filter.

Default: 3ms response time filter

Case	Response Time
0 (default)	3 ms
1	1 ms
2	5 ms
3	10 ms

For more details see datasheet SFM3304-xxx-D.

Note: This will return the value written to the Board, it does not reflect the real Sensor setting. After sensor is changed this value is setted to default (3ms response time filter).

Note for "get" commands: the cable outputs the number that was read upon start up and does not perform a new read to the sensor. A new read to the sensor can be performed if:

- an argument to the "get" function is added.

When a new read is requested, the measurement is stopped and restarts automatically. This takes approx. 4 ms during which no other command to the sensor can be sent. You must not use this command in a loop.

Note for "set" commands: To perform a "set" command, the measurement is stopped and restarts automatically. This takes approx. 4 ms during which no other command to the sensor can be sent. You must not use this command with an argument in a Loop.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required

7.41. Get/Set N averaging (41)

Request:

DevAddr	0x29	0x1 / 0x0	average	CRC8
---------	------	-----------	---------	------

Response:

DevAddr	0x29	0x1	average	CRC8
---------	------	-----	---------	------

Returns the Sensor Average.

Note: The value will hold internal, after Sensor is changed, this value is settled to default (N=2). A readout with Get Sensor status will deliver real Sensor values. SET_averaging 0- 127 needs low and high byte internal, high byte is ignored by FW and always zero Setting > 127 will set to 127 by the Sensor.

An Average by 100 will decrease reading speed of flow values to the Sensor also by hundred. An Average of 100 will result in an update rate of 25mS. With an Average by one it is possible to increase UpDate rate to 0,25mS, to get this speed you should consider to switch off the internal calculation in config.

If a pressure sensor exists, due to limited Bandwith on I²C a rate ~350µS is realistic but will run out of Spec of pressure sensor.

Note for "get" commands: the cable outputs the number that was read upon start up and does not perform a new read to the sensor. A new read to the sensor can be performed if:

- an argument to the "get" function is added.

When a new read is requested, the measurement is stopped and restarts automatically. This takes approx. 4 ms during which no other command to the sensor can be sent. You must not use this command in a loop.

Note for "set" commands: To perform a "set" command, the measurement is stopped and restarts automatically. This takes approx. 4 ms during which no other command to the sensor can be sent. You must not use this command with an argument in a Loop.

Exception Errors:

- 2 If device has no Firmware
- 3 If device has not yet been initialised
- 4 If device busy
- 15 Sensor is in Shutdown mode, HW-reset is required