SICK | MAIHAK

FLOWSIC600

Modbus Specification

Modbus Specification

Revision number: V3.0

Produced by:

SICK Engineering GmbH
Bergener Ring 27
D-01458 Ottendorf-Okrilla

7/25/2008 11:54:00 AM16

1. History

Version	Changes	Date	Editor
1.6	Initial (hardware V1.x)	2004-04-28	Dietz
2.0	Turbulence registers added, description master burst mode communication (hardware V2.x)	2006-07-28	Dietz
2.1	Description RTU protocol	2007-04-26	Hoffmann
3.0 B E_33118 B	Description Standard Modbus added, description of new registers (Firmware 3.2.00)	2008-01-03	Kirmse
E_33118 C	Addition of registers #7745 und #7746 in table of section 2.1.3	2008-08-25	Tretner
E_33118 D	Addition the description of changes in the History	2008-08-25	Tretner

2. Register Reference

The following tables show the recommended registers for the use with external controlling devices using Modbus connection. Most of the registers are READ-only. The registers with additional WRITE-access can be written without setting the FLOWSIC600 into configuration mode.

2.1.1. Short word registers

Register	Access	Unit	Description
3001	R		Device identification (flow meter type)
3002	R		System control register
3003	R		System status
3004..3007	R		Status register path 1..4
3008..3011	R		Rate of invalid samples of path 1..4 (failure rate)
3012..3019	R	dB	AGC level of receiver 1A, 1B, ..., 4A, 4B
3020	R	mV	Power level
3021	R	1/s	Actual measurement rate
3029	R	Hz	Current frequency
3051	R		Adjust mode
3058..3061	R		Extended status register path 1..4

Register **Name and Description**

3001 **DeviceType**

Register for setting the device identification (flow meter type)

Digit	Description
0	Ex-Class (0 = No Ex, 1 = ExIIA, 2 = ExIIB, 3 = ExIIC)
1	Path numbers
2..3	Meter size (inch)

Example: 241 = 2” Meter, 4 paths, ExIIA

3002 **SystemControl**

Register for controlling several device features

Bit	Description
0	Operation Mode (0 = Measurement, 1 = Configuration Mode)
1	Path 1 deactivation (0 = path active, 1 = path inactive)
2	Path 2 deactivation (0 = path active, 1 = path inactive)
3	Path 3 deactivation (0 = path active, 1 = path inactive)
4	Path 4 deactivation (0 = path active, 1 = path inactive)
5	Path 1 Checkcycle (0 = inactive, 1 = active)
6	Path 2 Checkcycle (0 = inactive, 1 = active)
7	Path 3 Checkcycle (0 = inactive, 1 = active)
8	Path 4 Checkcycle (0 = inactive, 1 = active)
9	Reset error volume counter
10	Device unit system (0 = metric unit system SI [m, m³/h, m/s], 1 = imperial unit system [ft, ft³/h, ft/s])
11	Additional signal filter (0 = inactive, 1 = active)
12	Stops the cyclic watchdog trigger, system restarts after 1,5sec
13	Resets the learned path conditions (path failure compensation)
14	Continuous Mode (0 = deactivated, 1 = activated “1”) If continuous mode is activated the system transmits always an up- and downstream signal. If continuous mode is deactivated the measure cycle will be canceled after receiving a bad upstream signal
15	Air flowtest mode - used for calibration (0 = deactivated, 1= activated) of ambient conditions

Register	Name and Description

3003 **SystemStatus**

Bit	Description
0	Device in measurement mode
1	Measurement valid
2	Check request • at least one path failed • max. frequency of impulse output limit exceeded • HART communication error (pressure or temperature)
3	Limit warning (User defined self diagnosis warning / status)
4	Parameter write protection active
5	Path 1 error
6	Path 2 error
7	Path 3 error
8	Path 4 error
9	Checksum error (CRC error) : • Code CRC error • Parameter CRC error • Real time clock date or time invalid • Volume Counter CRC error • Custody logbook limit exceeded max. entry limit or CRC error
10	Parameter out of range
11	Flow outside of calibration limits
12	Warning impuls output limit exceeded
13	No DSP communication
14	Path compensation valid
15	DSP parameter out of range

3004 ... 3007 **Path x Status**

Bit	Description
0	Warning SNR (SNR too low)
1	Warning AGC deviation (AGC deviation limit exceeded)
2	Warning AGC limit (max. AGC exceeded
3	Warning SOS deviation (Warning SOS deviation limit exceeded)
4	Read signal from DSP (Path signal is read from DSP)
5	Matrix singular (no fit)
6	MAX too big (Maximum signal amplitude too big, bad signal)
7	MAX too small (Maximum signal amplitude too small, bad signal)
8	MAXPOS too early (Position of maximum signal amplitude too early, bad signal)
9	MAXPOS too late (Position of maximum signal amplitude too late, bad signal)
10	Path error (Error of path exceeds limit)
11	SNR exceeds limit (bad signal)
12	Maximum iterations exceeded
13	Time plausibility
14	Check cycle active
15	Limit MSE (Limit of mean square error exceeded) (no fit)

3051 **Adjust Mode**

Value	Description
0	Adjust inactive
1	Adjust factor
2	Polynom
3	Piece wise linearisation

3058... 3061 **Path x Extended Status**

Bit	Description
0	Warning turbulence limit exceeded
1	Warning performance limit exceeded

2.1.2. Long word registers

Register	Access	Unit	Description
5001	R		Serial number device
5002	R		Software version
5003	R		Serial number analog board
5004	R		Metrology CRC (Custody relevant parameters)
5005	R		Firmware CRC
5006	R		Parameter CRC (user definable parameters)
5007*	R/W	ddmmyyyy	Date (real time clock)
5008*	R/W	hhmmss	Time (real time clock)
5010	R		Volume counter forward a.c. (9 digits)
5011	R		Volume counter forward error a.c. (9 digits)
5012	R		Volume counter reverse a.c. (9 digits)
5013	R		Volume counter reverse error a.c. (9 digits)
5014	R		Counterresolution; Volume in m^3 / $ft^{3:}$: m^3 = CounterResolution * VolumeCount / 1000 VolumeCount: content of the registers 5010..5013, 5016..5019, 5041..5048
5015	R/W	sec	Hysteresis time for Warning signaling
5016	R		Volume count forward total low a.c. (9 digits)
5017	R		Volume count forward total high a.c. (9 digits)
5018	R		Volume count reverse total low a.c. (9 digits)
5019	R		Volume count reverse total high a.c. (9 digits)
5020	R		Modbus ID
5021	R	bps	serial interface 1 (RS485-1(33/34): baudrate
5022	R	msec	serial interface 1 (RS485-1(33/34): responce delay
5023	R		serial interface 1 (RS485-1(33/34): control register
5024	R	bps	serial interface 2 (service - internal): baudrate
5025	R	msec	serial interface 2 (service - internal): responce delay
5026	R		serial interface 2 (service - internal): control register
5027	R	bps	serial interface 3 (RS485-1(81/82): baudrate
5028	R	msec	serial interface 3 (RS485-1(81/82): responce delay
5029	R		serial interface 3 (RS485-1(81/82): control register
5040	R		Extended system state register
5041	R		Volume count forward s.c. (9 digits)
5042	R		Volume count forward error s.c. (9 digits)
5043	R		Volume count reverse s.c. (9 digits)
5044	R		Volume count reverse error s.c. (9 digits)
5045	R		Volume count forward total low s.c. (9 digits)
5046	R		Volume count forward total high s.c. (9 digits)
5047	R		Volume count reverse total low s.c. (9 digits)
5048	R		Volume count reverse total high s.c. (9 digits)
5050	R		serial interface 1 (RS485-1(33/34): mode of masterburst
5051	R	msec	serial interface 1 (RS485-1(33/34): cycletime of masterburst
5052	R		serial interface 2 (service - internal): mode of masterburst
5053	R	msec	serial interface 2 (service - internal): cycletime of masterburst
5054	R		serial interface 3 (RS485-1(81/82): mode of masterburst
5055	R	msec	serial interface 3 (RS485-1(81/82): cycletime of masterburst
5056	R/W		Warning activation register
5057..5058	R		Short device tag (8 printable characters)
5059..5066	R		Long device tag (32 printable characters)

* Time synchronisation details, see Section 7.

Register Name and Description

5040

Extended System State

Register for extended system states

Bit	Description
0	Firmware CRC error
1	Volume counter a.c. CRC error
2	Volume counter s.c. CRC error
3	Parameter CRC error
4	Clock time invalid (battery low)
5	Custody logbook [1] CRC error
6	Custody logbook [1] overflow
7	Warning logbook [2] CRC error
8	Warning logbook [2] overflow
9	Parameter logbook [3] CRC error
10	Parameter logbook [3] overflow
11	Custody logbook [1] or Warning logbook [2] unacknowledged entries
12	Custody logbook [1] or Warning logbook [2] full of unacknowledged entries
13	Data logger 1 CRC error
14	Data logger 1 overflow
15	Data logger 2 CRC error
16	Data logger 2 overflow
17	Data logger 3 CRC error
18	Data logger 3 overflow
19	Default parameter loaded into RAM (due to invalid parameter range or CRC)
20	Path compensation parameters could not be saved
21	DSP measurement invalid
22	DSP boot error
23	HART pressure communication error
24	HART temperature communication error
25	Path failure (at least on path failed)
26	Negative flow direction
27	Warning profile factor limit exceeded
28	Warning symmetry limit exceeded
29	Warning input voltage limit exceeded
30	Warning velocity of gas limit exceeded
31	Factory test mode

5056

Warning Activation Register

Register for activation of several user warnings

Bit	Description
1	Limit warning Velocity of gas (0 = inactive, 1 = active)
2	Limit warning input voltage (0 = inactive, 1 = active)
3	Limit warning performance (0 = inactive, 1 = active)
4	Limit warning turbulence (0 = inactive, 1 = active)
5	Limit warning SOS deviation (0 = inactive, 1 = active)
6	Limit warning AGC level (0 = inactive, 1 = active)
7	Limit warning AGC deviation (0 = inactive, 1 = active)
8	Limit warning SNR level (0 = inactive, 1 = active)
9	Limit warning profile factor (0 = inactive, 1 = active)
10	Limit warning symmetry (0 = inactive, 1 = active)
11	Warning Custody logbook [1] or Warning logbook [2] unacknowledged entries (0 = inactive, 1 = active)
12	Warning Custody logbook [1] or Warning logbook [2] full of unacknowledged entries (0 = inactive, 1 = active)

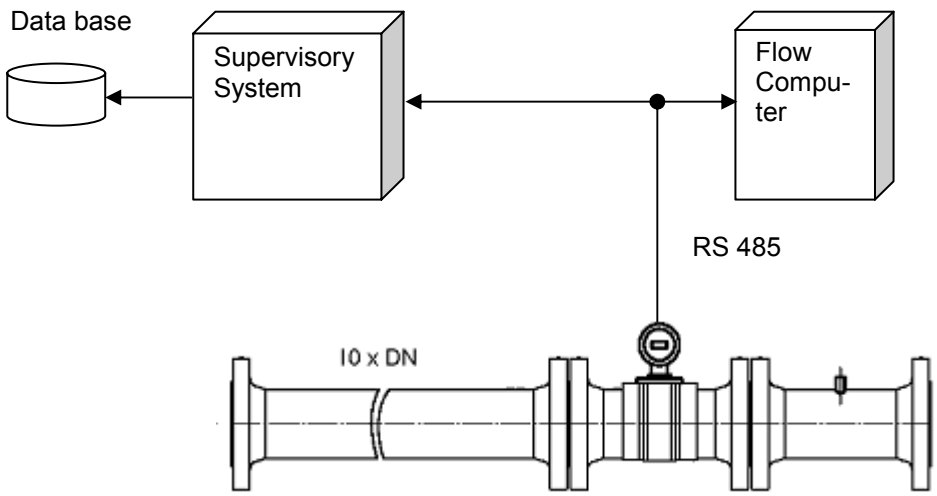
2.1.3. Floating point registers

Register	Access	Unit	Description
7001	R	m³/h	Volume flow rate at actual conditions (a.c.)
7002	R	m³/h	Volume flow rate at standard conditions (s.c.)
7003	R	m/s	Speed of sound (N-path average value)
7004	R	m/s	Velocity of gas (N-path average value)
7005...7008	R	m/s	Speed of sound per path 1..4 (averaged values)
7009...7012	R	m/s	Velocity of gas per path 1..4 (averaged values)
7013...7020	R	dB	SNR receiver 1A, 1B, ..., 4A, 4B (averaged values)
7021	R	°C	Gas temperature
7022	R	bar	Pressure(abs)
7023	R		Compressibility
7024	R	°C	Temperature base
7025	R	bar	Pressure base
7026	R		Compressibility base
7027	R		Meter factor (number of impulses per m³/ft³)
7028	R	m³/h	AO low limit
7029	R	m³/h	AO high limit
7030	R	s	AO time constant
7031	R	mA	AO test value
7032	R	mA	AO error current
7033	R		AO gain
7034	R	mA	AO offset
7035	R	mA	AO actual value
7036	R	m³/h	LowFlowCutoff
7037	R		Adjust factor forward
7038	R		Adjust factor reverse
7039	R	m³/h	Zero flow offset
7040	R	°C	Temperature fix
7041	R	bar	Pressure(abs) fix
7042	R	°C	Compressebility fix
7629 ... 7632	R	%	Relative turbulance of paths 1..4
7745	R		Profile factor
7746	R		Symmetry factor

3. Master Mode Communication

The FLOWSIC600 has the ability to communicate to more than one device (usually a flowcomputer and a supervisory system). This facilitates customer specific diagnostic data visualisation and data storage in a - mostly PC based - supervisory system for the metering station.

To avoid bus conflicts, it is necessary to let the FLOWSIC act as the communication master.



3.1. Implementation

If the FLOWSIC is the communication master, a data block is periodically transmitted over the serial bus. This data block implementation is conform to the Modbus specification. This makes it possible, to read the same data structure by a request, if the periodical transmission is not activated. All values in this data block are simple value copies of the existing value registers like flowrate or velocity of gas at this moment.

The data transfer is possible with all protocol frames which are implemented (SICK/GENERIC Modbus, ASCII/RTU Mode).

3.2. Register reference

UINT 2 byte unsigned integer
ULONG 4 byte long unsigned integer
FLOAT 4 byte float

3.2.1. Control Register

Register	Range	Unit	Description
5050	0..3		serial interface 1 (RS485-1(33/34): source of the structure to be transmitted
5051	500 .. 10000	msec	serial interface 1 (RS485-1(33/34): cycle time of master burst
5052	0..3		serial interface 2 (service – internal): source of the structure to be transmitted
5053	500 .. 10000	msec	serial interface 2 (service – internal): cycle time of master burst
5054	0..3		serial interface 3 (RS485-1(81/82): source of the structure to be transmitted
5055	500 .. 10000	msec	serial interface 3 (RS485-1(81/82): cycle time of master burst

Register	Name and Description
5050	**Master burst Source**
5052	Register for the type of structure which has to be transmitted
5054	Value Description 0 Master burst mode disabled 1 Only flowrate, speed of sound, the four volume counters and system status 2 Additionally the status, speed of sound and velocity of gas per path 3 Additionally the SNR ratio and the AGC level per path

3.2.2. Structure Type 1 (total size: 26 bytes)

Register	Size	Unit	Type	Description
(7001)	4 byte	m³/h	FLOAT	Actual Flowrate
(7003)	4 byte	m/s	FLOAT	Speed of sound
(5010)	4 byte		ULONG	Volume counter forward a.c. Note: The counter resolution value has to be considered!
(5012)	4 byte		ULONG	Volume counter reverse a.c. Note: The counter resolution value has to be considered!
(5011)	4 byte		ULONG	Volume counter error forward a.c. Note: The counter resolution value has to be considered!
(5013)	4 byte		ULONG	Volume counter error reverse a.c. Note: The counter resolution value has to be considered!
(3003)	2 byte		UINT	System status

3.2.3. Structure Type 2 (total size: 68 bytes)

Register	Size	Unit	Type	Description
(7001)	4 byte	m³/h	FLOAT	Actual Flowrate
(7003)	4 byte	m/s	FLOAT	Speed of Sound
(5010)	4 byte		ULONG	Volume counter forward a.c. Note: The counter resolution value has to be considered!
(5012)	4 byte		ULONG	Volume counter reverse a.c. Note: The counter resolution value has to be considered!
(5011)	4 byte		ULONG	Volume counter error forward a.c. Note: The counter resolution value has to be considered!
(5013)	4 byte		ULONG	Volume counter error reverse a.c. Note: The counter resolution value has to be considered!
(3003)	2 byte		UINT	System status
(3004)	2 byte		UINT	Path status 1
(3005)	2 byte		UINT	Path status 2
(3006)	2 byte		UINT	Path status 3
(3007)	2 byte		UINT	Path status 4
(7009)	4 byte	m/s	FLOAT	Velocity of gas path 1
(7010)	4 byte	m/s	FLOAT	Velocity of gas path 2
(7011)	4 byte	m/s	FLOAT	Velocity of gas path 3
(7012)	4 byte	m/s	FLOAT	Velocity of gas path 4
(7005)	4 byte	m/s	FLOAT	Speed of sound path 1
(7006)	4 byte	m/s	FLOAT	Speed of sound path 2
(7007)	4 byte	m/s	FLOAT	Speed of sound path 3
(7008)	4 byte	m/s	FLOAT	Speed of sound path 4

3.2.4. Structure Type 3 (total size: 114 bytes)

Register	Size	Unit	Type	Description
(7001)	4 byte	m³/h	FLOAT	Actual Flowrate
(7003)	4 byte	m/s	FLOAT	Speed of Sound
(5010)	4 byte		ULONG	Volume counter forward a.c. Note: The counter resolution value has to be considered!
(5012)	4 byte		ULONG	Volume counter reverse a.c. Note: The counter resolution value has to be considered!
(5011)	4 byte		ULONG	Volume counter error forward a.c. Note: The counter resolution value has to be considered!
(5013)	4 byte		ULONG	Volume counter error reverse a.c. Note: The counter resolution value has to be considered!
(3003)	2 byte		UINT	System status
(3004)	2 byte		UINT	Path status 1
(3005)	2 byte		UINT	Path status 2
(3006)	2 byte		UINT	Path status 3
(3007)	2 byte		UINT	Path status 4
(7009)	4 byte	m/s	FLOAT	Velocity of gas path 1
(7010)	4 byte	m/s	FLOAT	Velocity of gas path 2
(7011)	4 byte	m/s	FLOAT	Velocity of gas path 3
(7012)	4 byte	m/s	FLOAT	Velocity of gas path 4
(7005)	4 byte	m/s	FLOAT	Speed of sound path 1
(7006)	4 byte	m/s	FLOAT	Speed of sound path 2
(7007)	4 byte	m/s	FLOAT	Speed of sound path 3
(7008)	4 byte	m/s	FLOAT	Speed of sound path 4
(7013)	4 byte	dB	FLOAT	SNR path 1 AB
(7014)	4 byte	dB	FLOAT	SNR path 1 BA
(7015)	4 byte	dB	FLOAT	SNR path 2 AB
(7016)	4 byte	dB	FLOAT	SNR path 2 BA
(7017)	4 byte	dB	FLOAT	SNR path 3 AB
(7018)	4 byte	dB	FLOAT	SNR path 3 BA
(7019)	4 byte	dB	FLOAT	SNR path 4 AB
(7020)	4 byte	dB	FLOAT	SNR path 4 BA
(3012)	2 byte	dB	UINT	AGC path 1 AB
(3013)	2 byte	dB	UINT	AGC path 1 BA
(3014)	2 byte	dB	UINT	AGC path 2 AB
(3015)	2 byte	dB	UINT	AGC path 2 BA
(3016)	2 byte	dB	UINT	AGC path 3 AB
(3017)	2 byte	dB	UINT	AGC path 3 BA
(3018)	2 byte	dB	UINT	AGC path 4 AB
(3019)	2 byte	dB	UINT	AGC path 4 BA

4. Modbus Reference Specification

4.1. Data Protocol

Data Transfer (standard values boldface):

Property	Value
Data transfer	Serial, asynchronous, half duplex
Baudrate	1200bps 2400bps 4800bps **9600bps** 19200bps 38400bps 57600bps 115200bps
Start bits	**1 Bit**
Data bits	7 Bit **8 Bit** 9 Bit
Stop bits	**1 Bit** 2 Bit
Handshake	None
Parity	**none** even odd
Protocols	**SICK Modbus ASCII** SICK Modbus RTU Standard Modbus ASCII Standard Modbus RTU

4.2. Device Address

The FLOWSIC600 can use the communication slave addresses in the range of 1 through 127 (Register 5020). A parameter reset sets the device address always back to “1“ (factory setting). A query on the broadcast-address “0“ is answered by the system, giving the device address in the answer.

4.3. Communication Protocol

The communication protocol is implemented based on “Modicon Modbus III reference `J´.“ (http://www.modicon.com (part: PI-MBUS-300)).

4.3.1. SICK Modbus ASCII

a) Protocol Frame

In ASCII protocol mode two ASCII characters (0-9, A-F) are used to transfer one byte of data.
The frame starts with a “**:**” as preamble. The frame is closed with the character group “Carriage Return (**CR**) Line Feed (**LF**)” as the postamble.

Modbus ASCII telegram:

Start	Address	Function	Data	LRC Check	End
1 Char	2 Chars	2 Chars	n Chars	2 Chars	2 Chars
0x3A	1 - 127	1 - 255			0x0D 0x0A

b) Timeout

The maximum allowed response timeout is 2 seconds. The maximum timeout between two received characters is 1 second. In general a request will be answered immediately, at the latest in the next measuring cycle (response time typically less than 100msec). If necessary, the response time can be delayed by setting a delay time in the register „ModbusDelay“ (registers 5022, 5025, 5028, value 0...1000msec).

c) Error Detection

The data packet is combined with a longitudinal redundancy checksum (**LRC**) to increase the reliability of the transmitted data.

Algorithm:
All hexadecimally coded characters will be converted to 8-bit binary characters. All these characters are added. The overflow flag will be ignored. Finally the two-complement of the sum is formed. Pre- and postamble will not be used during the checksum calculation.

4.3.2. SICK Modbus RTU

a) Protocol Frame

In RTU mode all data is transferred as binary values. Possible characters are 0 – 9 and A – F hexadecimal.

Modbus RTU telegram:

Start	Address	Function	Data	CRC Check	End
	8 Bit	8 Bit	n x 8 Bit	16 Bit	
3,5 t_{Byte}	1 – 127	1 - 255		CRC low, CRC High	3,5 t_{Byte}

t_{Byte} = length of one character

Example for 1t_{Byte}:
Baud rate: 57600 bps
t_{Bit} = 1/57600 = 17,36 µs
t_{Byte} = 8 Bit * 17,36 µs = 138,88 µs + (2*17,36 µs) = **173,6 µs**

b) Timeout

Before start of any transmission a break of 3,5 t_{Byte} is required. After this break the data telegram must be transmitted in a continuous stream of characters. The telegram will be ignored in case of more than 1,5 t_{Byte} interruption.

c) Error Detection

Similar to the ASCII protocol, check sums are calculated using the cyclic redundancy check (CRC) method. The CRC is calculated for the complete data of each telegram and is represented by a 16 bit integer. The CRC is transferred by 2 bytes starting with the least significant bit (LSB) followed by the most significant bit (MSB).

Algorithm to calculate the CRC:
1. init of CRC-register with 0xFFFF
2. XOR conjunction of the first data byte with the LSB of the CRC-register
3. write result into CRC-register
4. shift CRC-register right by 1 bit
5. fill MSB of CRC-register with 0
6. If LSB was 0: proceed with step 4
   If LSB was 1: XOR conjunction of CRC-register with fixed value (polynomial)
7. repeat step 3 and 4 for 8 shifts

4.3.3. STANDARD Modbus ASCII

The implementation of STANDARD Modbus ASCII protocol is mostly the same as SICK Modbus ASCII. Due to the fact that only 16-bit data types are supported, a few differences exist:

- The register number which has to be named is always the desired `REG_NR-1`
- A write action to 32-bit registers has to be done by the command 0x10 “Write Multiple Registers”
- The command 0x06 “Write Single Register” is not available

4.3.4. STANDARD Modbus RTU

The implementation of STANDARD Modbus RTU protocol is mostly the same as SICK Modbus RTU. Due to the fact that only 16-bit data types are supported, a few differences exist:

- The register number which has to be named is always `REG_NR-1`
- A write action to 32-bit registers has to be done by the command 0x10 “Write Multiple Registers”
- The command 0x06 “Write Single Register” is not available

4.4. Implemented Data Types

All implemented data types are grouped. This allows sorting distinguishing of data types by the register number. Note that the 32bit registers are counted as one register number unlike some Modbus implementations which use two register numbers to represent one 32bit number!

Integer 16bit

Register Group 3xxx				
16-bit unsigned integer	MSB		LSB	
Bits	NNNN	NNNN	NNNN	NNNN
Order	B0	B1	B2	B3

Integer 32bit

Register Group 5xxx								
32-bit unsigned integer	MSB						LSB	
Bytes	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN	NNNN
Order	B0	B1	B2	B3	B4	B5	B6	B7

Float 32bit (IEEE-754)

Register Group 7xxx								
IEEE float	Sign (1bit), Exponent (8bits), Mantissa (23bits)							
Bytes	SEEEEEEE		EMMMMMMM		MMMMMMMM		MMMMMMMM	
Order	B0	B1	B2	B3	B4	B5	B6	B7

4.5. Implemented Commands

The following command descriptions are in the form of SICK Modbus ASCII protocol. They have to be adapted if different protocol type is used.

4.5.1. Code 0x03 "Read Multiple Registers"

Used to read out the value of one or more (successive) registers. The command must declare:

- the register number of the first register to read (REG_NR)
- and the amount of registers to read (REG_CNT).

The maximum amount of registers wich can be read by a request is 50. Note that the read request of a block with a "gap" between defined register numbers results in an error message (Code 0x02, unsupported register number).

Command:

```
:, ADDR, 0x03, REG_NR MSB, REG_NR LSB, REG_CNT MSB, REG_CNT LSB, LRC, CR, LF
```

Answer:

```
:, ADDR, 0x03, BYTE_CNT, ... DATA ..., LRC, CR, LF
```

4.5.2. Code 0x06 "Write Single Register"

Sets a new value into the defined register. The command must declare the number of the register to be modified (REG_NR) and the new value (VALUE). This command is only available in SICK Modbus ASCII/RTU mode.

Command:

```
:, ADDR, 0x06, REG_NR MSB, REG_NR LSB, VALUE, LRC, CR, LF
```

Answer:

```
:, ADDR, 0x06, REG_NR MSB, REG_NR LSB, VALUE, LRC, CR, LF
```

4.5.3. Code 0x10 "Write Multiple Registers"

Used to write the values of one or more (successive) registers The command must declare:

- the register number of the first register to write (REG_NR)
- and the amount of registers to write (REG_CNT).

The maximum amount of registers which can be written by a request is 2. Note that the write request of a block with a "gap" between defined register numbers results in an error message (Code 0x02, unsupported register number). This command is only available in STANDARD Modbus ASCII/RTU mode.

Command:

```
:, ADDR, 0x10, REG_NR MSB, REG_NR LSB, REG_CNT MSB, REG_CNT LSB, VALUE LRC, CR, LF
```

Answer:

```
:, ADDR, 0x10, BYTE_CNT, ... DATA ..., LRC, CR, LF
```

Command:

```
:, ADDR, 0x06, REG_NR MSB, REG_NR LSB, REG_CNT MSB, REG_CNT LSB, VALUE (1) MSB, VALUE (1) LSB ....
VALUE (REG_CNT) MSB, VALUE (REG_CNT) LSB, LRC, CR, LF
```

Answer:

```
:, ADDR, 0x06, REG_NR MSB, REG_NR LSB, REG_CNT MSB, REG_CNT LSB, VALUE (1) MSB, VALUE (1) LSB ....
VALUE (REG_CNT) MSB, VALUE (REG_CNT) LSB, LRC, CR, LF
```

4.6. Exception Responses

The following transmission status descriptions are in the form of SICK Modbus ASCII protocol. They have to be adapted if another protocol type is used.

4.6.1. Error Code

A corrupted request will be answered with an error status code. The function code of the answer is formed by adding 0x80hex.

Status Code	Name	Description
Code 0x01	Unknown function code	The received function code is not supported by the device
Code 0x02	Unsupported register Number	The requested register number is not used by the device
Code 0x03	Invalid data value	The received data value exceeds the defined valid range

Example: Unknown register number

Command:

ASCII Bytes	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13
Hex	0x3A	0x31	0x31	0x30	0x33	0x39	0x39	0x39	0x39	0x30	0x31	0xB9	0x0D	0x0A
ASCII	:	**1**	**1**	**0**	**3**	**9**	**9**	**9**	**9**	**0**	**1**	**-**	**-**	**-**
	Start	Address		Function code		Register				Number		LRC	CRLF	

Answer:

ASCII Bytes	B0	B1	B2	B3	B4	B5	B6	B11	B12	B13
Hex	0x3A	0x31	0x31	0x38	0x33	0x30	0x32	0x6A	0x0D	0x0A
ASCII	:	**1**	**1**	**8**	**3**	**0**	**2**	**-**	**-**	**-**
	Start	Address		Function code		Error Status Code		LRC	CRLF	

4.6.2. Error Sources

CODE 0x01 Unknown function code

a.) Command code is not supported (only command codes 0x03 or 0x06 are supported)
b.) Write access to a parameter with configuration mode inactive (measurement mode active)
c.) Write access to a protected parameter on active parameter protection switch.
d.) Write access to a “read only” defined register.

CODE 0x02 Unsupported register number

a.) The declared register number is not used and supported by the device.
b.) Read command: The register number is valid, but the number of registers to read exceeds the register group border, or the amount of registers to read is more than 50.

CODE 0x03 Invalid data value

a.) Write command: The declared data value exceeds the defined value range of the register.

4.7. Examples

The following transmission status descriptions are in the form of SICK Modbus ASCII protocol. They have to be adapted if another protocol type is used.

4.7.1. Read One 3xxx Register (Command "Read Multiple Registers")

Assumed register value 0x1234h
Register 3001 (0x0BB9h)
Slave address 0x11h
LRC **0xFF -** (0x11+ 0x03 + 0x0B + 0xB9 + 0x01) **+ 0x01**

Query:
Transmitted ASCII string:

Hex	3A	31	31	30	33	30	42	42	39	30	30	30	31
ASCII	**:**	**1**	**1**	**0**	**3**	**0**	**B**	**B**	**9**	**0**	**0**	**0**	**1**
Desc.	Start	Address		Function Code		Register				amount of registers			

Hex	32	37	0D	0A
ASCII	**2**	**7**	**#10**	**#13**
Desc.	**LRC**		**CRLF**	

Response:
Received ASCII string:

Hex	3A	31	31	30	33	30	32	31	32	33	34	41	34	0D	0A
ASCII	**:**	**1**	**1**	**0**	**3**	**0**	**2**	**1**	**2**	**3**	**4**	**A**	**4**	**#10**	**#13**
Desc.	Start	Address		Function code		Byte count		Register value				LRC		CRLF	

4.7.2. Read One 5xxx Register (Command "Read Multiple Registers")

Assumed register value 0x12345678h
Register 5006 (0x138Eh)
Slave address 0x11h
LRC **0xFF -** (0x11+ 0x03 + 0x13 + 0x8E + 0x00 + 0x01) **+ 0x01**

Query:
Transmitted ASCII string:

Hex	3A	31	31	30	33	31	33	38	45	30	30	30	31	34	41	0D	0A
ASCII	**:**	**1**	**1**	**0**	**3**	**1**	**3**	**8**	**E**	**0**	**0**	**0**	**1**	**4**	**A**	**#10**	**#13**
Desc.	Start	Address		Function Code		Register				Number of points				LRC		CRLF	

Response:
Received ASCII string:

Hex	3A	31	31	30	33	30	34	31	32	33	34	35	36	37	38	44	34	0D	0A
ASCII	**:**	**1**	**1**	**0**	**3**	**0**	**4**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**D**	**4**	**#10**	**#13**
Desc.	Start	Slave-Address		Function code		Byte count		Register value								LRC		CRLF	

4.7.3. Write One 5xxx Register (Command "Write Single Register")

Assumed register value 0x00002580h
Register 5002 (0x138Ah)
Slave Address 0x11h
LRC **0xFF -** (0x11+ 0x06 + 0x13 + 0x8A + 0x00 + 0x00 + 0x25 + 0x80) **+ 0x01**

Query:
Transmitted ASCII String:

Hex	3A	31	31	30	36	31	33	38	41	30	30	30	30	32	35	38	30
ASCII	:	**1**	**1**	**0**	**6**	**1**	**3**	**8**	**A**	**0**	**0**	**0**	**0**	**2**	**5**	**8**	**0**
Desc.	Start	Address		Function code		Register				Register value							

Hex	41	37	0D	0A
ASCII	**A**	**7**	**#10**	**#13**
Desc.	LRC		CRLF	

Response:
Received ASCII String (echo!):

Hex	3A	31	31	30	36	31	33	38	41	30	30	30	30	32	35	38	30	41	37	0D	0A
ASCII	:	**1**	**1**	**0**	**6**	**1**	**3**	**8**	**A**	**0**	**0**	**0**	**0**	**2**	**5**	**8**	**0**	**A**	**7**	**#10**	**#13**
Desc.	Start	Address		Function code		Register				Register value								LRC		CRLF	

4.7.4. Write One 5xxx Register (Command "Write Multiple Register")

Assumed register value 0x00002580h
Register 5002 (0x138Ah)
Slave Address 0x11h
LRC **0xFF-**(0x11+ 0x10 + 0x13 + 0x89 + 0x00 + 0x02 + 0x00+ 0x00+ 0x25 + 0x80)**+0x01**

Query:
Transmitted ASCII String:

Hex	3A	31	31	30	36	31	33	38	41	0	0	0	2	30	30	30	30	32	35	38	30
ASCII	:	**1**	**1**	**1**	**0**	**1**	**3**	**8**	**9**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**2**	**5**	**8**	**0**
Desc.	Start	Address		Function code		Register-1				Number of points				Value							

Hex	41	37	0D	0A
ASCII	**A**	**7**	**#10**	**#13**
Desc.	LRC		CRLF	

Response:
Received ASCII String (echo!):

Hex	3A	31	31	30	36	31	33	38	41	30	30	30	30	32	35	38	30	41	37	0D	0A
ASCII	:	**1**	**1**	**0**	**6**	**1**	**3**	**8**	**A**	**0**	**0**	**0**	**0**	**2**	**5**	**8**	**0**	**A**	**7**	**#10**	**#13**
Desc.	Start	Address		Function code		Register				Register value								LRC		CRLF	

5. Time Synchronisation

The date and the time of the FLOWSIC600 can be set seperately by an external write. Each operation for date and time causes a seperate entry in the custody logbook [1].

Alternatively the synchronization function can be used. To use this method, the date register (#5007) and the time register (#5008) have to be written sequentially within 2 seconds. The date register (#5007) has to be written first. The write operation can be accomplished without setting the FLOWSIC600 into configuration mode. This synchronizaton causes a logbook entry only if the time change is greater than 3% of the time elapsed since the last synchronisation.

MEPAFLOW600 CBM offers the use of the synchronization function via a button in the “Meter Information” screen.