S- / D-AGM Plus

Transmitter Electronics for IR, TC and EC Gas Sensors

- Protocol Specification -

1 Contents

2 S-AGM Plus Protocol

The protocol can be used to access all public data points via USB virtual COM-port or over the RS485 interface.

Examples for public data points are:

- gas concentration, additional data like gas name and unit
- temperature and pressure
- serial number
- external analog input and output

2.1 Hardware layer

The S-AGM Plus has two different electrical connections to access the measurement data.

2.1.1 Virtual COM-port

The S-AGM Plus can be accessed as USB communication class device (virtual COM-port). The baud rate must be set to 38400 the other settings must be set to **8N1**.

The interface is not electrically isolated.

2.1.2 RS485

The S-AGM Plus can communicate over a RS485 connection. The baud rate must be set to 38400 all other settings must be set to **8N1**.

The interface is not electrically isolated.

2.2 Data types

The following data types will be used within the protocol:

ID	*Name*	*Size (bit)*	*Size (32bit)*	*Range*
0x00	Boolean	1	1/32	0=false, 1=true
0x10	byte	8	1/4	[0...255]
0x20	word	16	1/2	ca. [0...65535]
0x30	int	32	1	ca. [-2e9...+2e9]
0x40	long	64	2	ca. [-9e18...9e18]
0x50	float	32	1	[-inf...+inf]
0x60	double	64	2	[-inf...+inf]

If not otherwise stated, multi byte data must be transmitted in **MSB first** order!

2.3 Sync

Each data frame will be started by a **DLE STX** pair and will end with a **DLE ETX** pair. Whenever the **DLE** symbol occurs within the message a **DLE ESC** pair will be transmitted.

code	*function*
0x10	DLE
0x02	STX
0x03	ETX
0x1b	ESC

DLE within content will be transmitted as **DLE ESC** sequence
Start of packet will be introduced by **DLE STX** sequence
End of packet will be finished with **DLE ETX** sequence

Any 0x10 within the frame needs to be escaped. A 0x10 within the counter and /or a 0x10 within the CRC need to be escaped also as it have to be done for the 0x10 within the frame.

#1 Example for a single frame with two 0x10 bytes in the data body:

Raw Data: 10 02 ... 10 1b ... 10 1b ... 10 03
Frame: Start ... 10 ... 10 ... End

#2 Example: Send a frame to the bench and the bench response answer shows a DLE ESC 1b for the 10 within their response.

Send:<10 02 9c ff 40 06 00 04 0c 06 00 22 08 48 c7 10 03 >
Response:<10 02 00 9c 41 93 ed e8 3e 00 78 fa 41 12 9c 7d 44 14 6c c1 41 00 00 00 00 10 1b 25 10 03 >
Within the response from the bench a „10“ is listed (red 10 above), and the bench uses the DLE ESC function (green 1b above)

So similar to the example above, if a frame is sent to the bench where a 10 is within the DLE ESC 1b is needed.

S:<10 02 10 ff 40 06 00 04 0c 06 00 22 08 de 55 10 03 > → **missing DLE ESC “1b”, no response coming**
S:<10 02 11 ff 40 06 00 04 0c 06 00 22 08 da a9 10 03 > → **no “10” included, response is coming**
R:<10 02 00 11 41 36 60 64 3f 00 9c f4 41 54 5f 7c 44 ff b0 c1 41 00 00 00 00 7e 66 10 03 >

2.4 Frame

The frame format for a request is:

byte	*byte*	*byte*	*byte[]*	*byte*	*byte*
seq	addr	cmd/res	data	crc16 lo	crc16 hi

The frame format for the answer is:

byte	*byte*	*byte*	*byte[]*	*byte*	*byte*
addr	seq	cmd/res	data	crc16 lo	crc16 hi

Please note: addr and seq is swapped in the answer!

The sender sequence number must be increased with each packet. The sequence number will be echoed in the answer packet.

The special address 0xff can always be used to address a single connected device. The configurable address can be found under: $DEVICE:$SYSTEM:$ADDRESS

Example for a single frame, command is 0x30, seq=4, addr=5 and crc=0x7ac4:

Frame: Start 04 05 30 [n bytes data] c4 7a End

2.5 CRC

The following polynomial must be used for the CRC calculation: x^16 + x^15 + x^2 + 1 (0xa001)

The initial value is 0xffff.

Algorithm in C:

```
uint16_t crc16_update(uint16_t crc, uint8_t a) {
  int i;
  crc ^= a;
  for (i = 0; i < 8; ++i) {
    if (crc & 1)
      crc = (crc >> 1) ^ 0xA001;
    else
      crc = (crc >> 1);
  }
  return crc;
}
```

In contrast to all other multi-byte values the CRC is transmitted LSB first!

Sequence for CRC:

1. Create the data package
2. calculate the CRC
3. add calculated CRC to the end of data package
4. add start condition in front of command
5. if needed define DLE ESC condition
6. add end condition to the end of command.

2.6 Requests

command	***ID***	***answer***	***ID***
Ping	0x00	Ping reply	0x01
Read configuration	0x10	Read configuration reply	0x11
		Read configuration reply last	0x12
Read strings	0x20	Read strings reply	0x21
		Read strings reply last	0x22
Get ID	0x30	Get ID reply	0x31
		Get ID error	0x32
Read values	0x40	Read values reply	0x41
		Read values error	0x42
Write values	0x50	Write values success	0x51
		Write values error	0x52
Request log data	0x60	Request log data response	0x61
		Request log data stalled response	0x62
		Request log data error	0x63

2.6.1 Ping

The Ping command can be used to test the communication each ping packet will be answered by a ping reply packet.

Request:

0x00

cmd

Response:

0x01

cmd

2.6.2 Read Configuration

The read configuration command can be used to get information about the whole data structure. If there is no more data, the cmd-byte of the reply will be 0x12, otherwise 0x11.

Request:

0x10 *byte*

cmd index

Response:

	n*			
0x11 / **0x12**	*byte*	*byte*	*byte*	*byte*
cmd	type	name	table	size

2.6.3 Read Strings

The *Read Strings* command reports a string array. All strings are introduced with a size byte. If there are no more strings the command byte of the reply will be 0x22, otherwise 0x21.

Request:

0x20	*byte*
cmd	index

Response:

	n*	
0x21 / **0x22**	*byte*	*byte[]*
cmd	size	string

Example for strings: "ABC", "DEF", "GHI", "JKL", "MNO", "PQR":

```
Request: 20 00
Reply:   21 03 41 42 43 03 44 45 46
Request: 20 01
Reply:   21 03 47 48 49 03 4a 4b 4c
Request: 20 02
Reply:   22 03 4d 4e 4f 03 50 51 52
```

2.6.4 Get Id

This command can be used to get the id for a single data entry. The entry is specified by a path which must be supplied in an array. All array entries are introduced with a size-byte.
The root path entry „$DEVICE“ should be left out!

The replied size must be multiplied with the data type-factor to get the size in bytes.

Request:

	n*	
0x30	*byte*	*byte[]*
cmd	size	string

Response:

0x31	*byte*	*byte*	*word*		*byte*
cmd	type	table	offset_H	offset_L	size

Example for path *ABC/DEF/GHI*:

```
Request 0x30, 0x03, 0x41, 0x42, 0x43, 0x03, 0x44, 0x45, 0x46, 0x03, 0x47,
0x48, 0x49
Reply 0x31, 0x__, 0x__, 0x__, 0x__, 0x__
```

2.6.5 Read Values

This command initiates the transmission of some data areas. Each single memory area is described by its table-id, the byte-offset and the byte-size. The caller has to make sure that the answer will fit in a single data frame.

Request:

	n*			
0x40	*byte*	*word*		*byte*
cmd	table-id	offset_H	offset_L	byte-size

Response:

	n*
0x41	*byte[]*
cmd	data

2.6.6 Write Values

Request:

	n*				
0x50	*byte*	*word*		*byte*	*byte[]*
cmd	table	offset_H	offset_L	size	data

Response:

0x51
cmd

2.7 S-AGM Plus Data

The data of the S-AGM Plus is organized in banks. Each bank has its own access rights.

Public accessible banks are:

- **0** public constant (r)
- **2** public static (rw)
- **3** protected static (r)
- **5** public dynamic (rw)
- **6** protected dynamic (r)

Private banks are:

- **1** private constant (r)
- **4** private static (rw)
- **7** private dynamic (rw)

2.8 Units

For a better interpretation of the data points the type is further divided. Special sub-types exist for several units and for strings and hex-arrays.

Data type	***Unit***	***ID***	***Comment***
byte		0x10	unspecified byte data
byte	string	0x11	UTF8 encoded string data, null-padded
byte	hex data	0x12	hexadecimal data
float/double		0x50/0x60	unspecified floating point data
float/double	volt	0x51/0x61	voltage
float/double	ampere	0x52/0x62	current
float/double	watt	0x53/0x63	power
float/double	ohm	0x54/0x64	resistance
float/double	bar	0x55/0x65	pressure
float/double	Kelvin	0x56/0x66	temperature in Kelvin
float/double	second	0x57/0x67	time

3 Examples

The following example shows a communication between host software and the S-AGM Plus. For channel1 the current value and the temperature will be polled. Afterwards a zero bench calibration is performed for channel 1.

3.1 Lookup Data Ids

In a first step the Ids of the value, temperature and the calibration-command data point must be requested. The examples are shown at the framing level, without start and end symbol (1002 1003) and without escape sequences (101e).

Request:
Get Id „Channel 1:Data:$VALUE“
a0b0 30 094368616e6e656c2031 0444617461 062456414c5545 00 [CRCL CRCH]

Response:
b1a0 31 5006000401 [CRCL CRCH]
Type is float (0x50), table is protected dynamic(6), byte offset is 4 and the size is 1.
The calculated size in bytes is 4 (float size is 32 bit).

Request:
Get Id „Channel 1:Data:temperature“
a1b1 30 094368616e6e656c2031 0444617461 0b74656d7065726174757265 00
[CRCL CRCH]

Response:
b2a1 31 5606001401 [CRCL CRCH]
Type is float degree (0x56), table is protected dynamic(6), byte offset is 20 and the size is 1.
The calculated size in bytes is 4 (float size is 32 bit).

Request:
Get Id „Channel 1:Calibration:command“
a2b2 30 094368616e6e656c2031 0b43616c6962726174696f6e 07636f6d6d616e64
00 [CRCL CRCH]

Response:
b3a2 31 1005000901 [CRCL CRCH]
Type is byte (0x10), table is protected dynamic(5), byte offset is 9 and the size is 1.
The calculated size in bytes is 1 (byte size is 8 bit).

3.2 Poll Values

The following command shows how to poll the value and the temperature of channel 1 within a single request.

Request:
Read values 6/4/4 and 6/20/4 (bank/offset/size in byte)
a3b3 40 06000404 06001404 [CRCL CRCH]

Response:
b4a3 41 xxxxxxxx yyyyyyyy [CRCL CRCH]
The result contains the IEEE754 encoded floating point values for the actual value and the temperature

3.3 Write Value

To perform a zero bench for channel 1 the write command (0x10) has to be written to the Calibration data point.

Request:
Write value 5/9/1/[0x10] (bank/offset/size in byte/[data])
a4b4 50 05000901 10 [CRCL CRCH]

Response:
b5a4 41 10 [CRCL CRCH]
The result contains the written value as if the read value command was used.

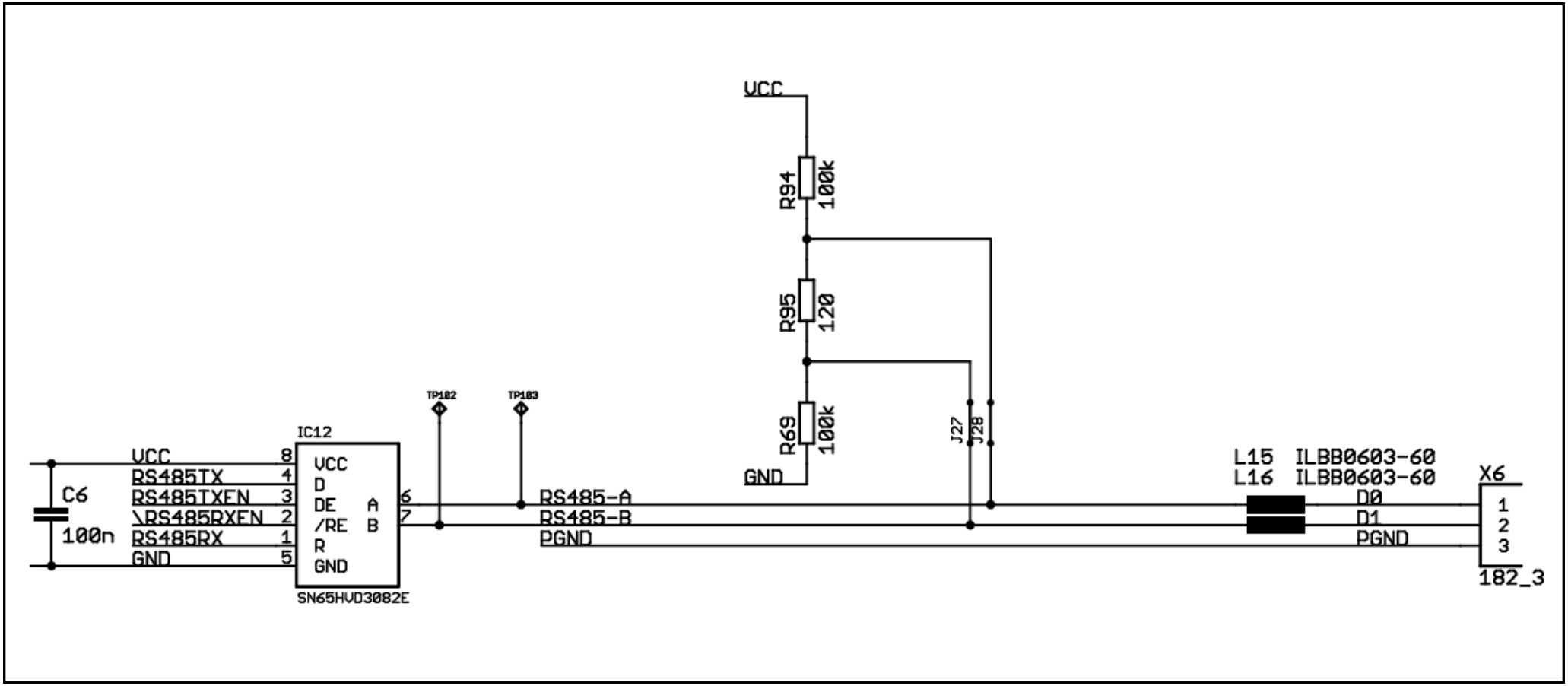
3.4 Overview

10 02 9b ff 40 06 00 04 0c 06 00 22 08 52 b3 10 03

10 02	Start
9b	counter
ff	Address (Broadcast Address, if no RS-485 Address is set, this one is a standard)
40	Command: read values
06 00 04	Measurement value address (CH1-Data-Value)
0c	[FF (form feed)] (following 12 Byte requested, so also temp and pressure are requested in this example)
06 00 22	Measurement value Global- Supply
08	FF (form feed)] (following 8 Byte requested, so also the analog input value is requested
52 b3	CRC
10 03	End

4 RS-485 schematic

Based on schematic listed below following information can be supplied. **D0 = D0 = A | D1 = D1 = B | PGND = PGND = PGND**



5 Overview table / calibration command

Information based on firmware 1877 and host software 1.4.4. Regarding earlier firmware versions please take a look into the structure.xml on the S-AGM Plus which is available or contact Sensors Europe GmbH for any further information.

S- / D-AGM Plus Version: 3.4.1 Revi: 1877 Basis: http://127.0.0.1:8080/device1/structure.xml

The addresses listed in the table below are based on the Firmware 1550 and on Host software 1.3.1. On further firmware versions the addresses may change. Therefore it is advised to use the "Get ID" function, which always responses the correct addresses.

Name	Address	Size	Function	Description	Unit	Information
System-Device-Type	H000000	8 Byte	Read	Type of Device	ASCII Sign	
System-Device-Version	H0000008	8 Byte	Read	Version of card software	ASCII Sign	
System-Device-Revision	H0000010	12 Byte	Read	Revision of card software	ASCII Sign	
System-Serial-Number	H0030000	12 Byte	Read	Serial number	ASCII Sign	
System-RS-485-Adress	H0020014	1 Byte	Read/Write	Address of RS-485	int. Byte	
Ch1-Name	H0030020	8 Byte	Read	Kind of gas measured	ASCII Sign	
Ch1-Info	H0030029	20 Byte	Read	Info, measurement range	ASCII Sign	
Ch1-Data-Value	H0060004	4 Byte	Read	Measurement value	IEEE754-Number	
Ch1-Data-Unit	H003004D	4 Byte	Read	Measurement unit	ASCII Sign	
Ch1-Data-Error	H0060008	1 Byte	Read	Error-status flag	Int. Byte	
Ch1-Data-Temperature	H0060009	4 Byte	Read	Temperature	IEEE754-Number	
Ch1-Data-Pressure	H006000D	4 Byte	Read	Gas pressure	IEEE754-Number	
Ch1-Calibration-Command	H0050009	4 Byte	Read/Write	Calibration status register	H00	No calibration [0]
					H10	Start zero calibration [16]
					H11 - H1F	Zero calibration running [16…31]
					H20	Start OPC calibration [32]
					H21 - H2F	OPC calibration running [32…47]
Ch1-OPC-reference	H0020015	4 Byte	Read/Write	Calibration reference value-OPC	IEEE754-Number	One-Point-Calibration ref.

Name	Address	Size	Function	Description	Unit	Information
Ch2-Name	H0030053	8 Byte	Read	Kind of gas measured	ASCII Sign	
Ch2-Info	H003005C	20 Byte	Read	Info, measurement range	ASCII Sign	
Ch2-Data-Value	H0060016	4 Byte	Read	Measurement value	IEEE754-Number	
Ch2-Data-Unit	H0030080	4 Byte	Read	Measurement unit	ASCII Sign	
Ch2-Data-Error	H006001A	1 Byte	Read	Error-status flag	Int. Byte	
Ch2-Data-Temperature	H006001B	4 Byte	Read	Temperature	IEEE754-Number	
Ch2-Data-Pressure	H006001F	4 Byte	Read	Gas pressure	IEEE754-Number	
Ch2-Calibration-Command	H005000E	4 Byte	Read/Write	Calibration status register	H00	No calibration [0]
					H10	Start zero calibration [16]
					H11 - H1F	Zero calibration running [16…31]
					H20	Start OPC calibration [32]
					H21 - H2F	OPC calibration running [32…47]
Ch2-OPC-reference	H002002E	4 Byte	Read/Write	Calibration reference value-OPC	IEEE754-Number	One-Point-Calibration ref.

Name	Address	Size	Function	Description	Unit	Information
Global-Supply	H0060030	4 Byte	Read	Supply voltage	IEEE754-Number	
Global-Input-Name	H002004F	8 Byte	Read/Write	Name of analog input	ASCII Sign	
Global-Input-Value	H0060034	4 Byte	Read	Connected analog signal	IEEE754-Number	
Global-Input-Unit	H0020057	4 Byte	Read/Write	Measurement range of analog input	ASCII Sign	
Global-Input-low	H002005D	4 Byte	Read/Write	Lower boarder	IEEE754-Number	
Global-Input-high	H0020061	4 Byte	Read/Write	Higher boarder	IEEE754-Number	
Global-Input-offset	H0020066	4 Byte	Read/Write	Offset	IEEE754-Number	
Global-Input-gain	H002006A	4 Byte	Read/Write	Gain	IEEE754-Number	
Global-Output1-low	H002006E	4 Byte	Read/Write	Lower boarder	IEEE754-Number	
Global-Output1-high	H0020072	4 Byte	Read/Write	Higher boarder	IEEE754-Number	
Global-Output1-offset	H0020077	4 Byte	Read/Write	Offset	IEEE754-Number	
Global-Output1-gain	H002007B	4 Byte	Read/Write	Gain	IEEE754-Number	
Global-Output1-fault	H002007F	4Byte	Read	Fault	IEEE754-Number	

Name	Address	Size	Function	Description	Unit	Information
Global-Output2-low	H0020083	4 Byte	Read/Write	Lower boarder	IEEE754-Number	
Global-Output2-high	H0020087	4 Byte	Read/Write	Higher boarder	IEEE754-Number	
Global-Output2-offset	H002008C	4 Byte	Read/Write	Offset	IEEE754-Number	
Global-Output2-gain	H0020090	4 Byte	Read/Write	Gain	IEEE754-Number	
Global-Output2-fault	H0020094	4 Byte	Read	Fault	IEEE754-Number	

Examples

Type in Structure.xml	Hex	Datatype
16	0h10	byte
17	0h11	byte
18	0h12	byte
48	0h30	int
80	0h50	float
81	0h51	float
85	0h55	float
86	0h56	float
0		boolean

Example: Data of "Value" from CH1

<data name="$VALUE" type="80" bank="6" size="1" id="14" linkid="0x00060004">0.06</data>

LinkID=0x00060004 393220 regarding firmware 1395

Type =80 0h50 float

0h50=float = 32bit = 4Byte

Size=1 0h50 (float) = 4 Byte * Size = total size

0h50 (float) = 4 Byte * 1 = **4 Byte**

Data types	Size(bit)
float	32
int	32
byte	8
boolean	8

Please make sure the names and addresses could be changed in further firmware version. It is advised to use the "Get ID" function to get the correct addresses on every firmware version!

Calibration command:

After the calibration command is set, for example zero bench command (H10 = 16), the value within the calibration command keeps increasing till the end value is reached (H10 to H1F = 16 to 31). If end value is reached (H1F = 31), the calibration (zero bench in this example) is done. The end value will be stored in the calibration command. A 0 within this calibration command is only set in if the bench is started up for first time. After any calibration command is set, the specific end value will be listed within. For one-point-calibration start command is H20 (H20=32). It will be increased till end value of H2F (H2F=47) is reached. Then the one-point-calibration is done.

Please note: Within the customer manual a description about “**error**” and “**fault**” definition is given on chapter 8.3 “Correct troubleshoot of your S-AGM Plus”.

Customizable IIR filter

For both channels a separate customizable IIR filter is available. In addition to the averaging done within the original manufacturer settings, this customizable IIR filter may be added if required by the customer. A further detailed description about this IIR filter is available within the customer manual.

For example (keep in mind that total summary should be 1):

- 90% from old value should be used and 10% from actual new value (this setting leads to a strong averaging)
    - f_factor: 0.1 (similar to 10%)
    - r_factor: -0.9 (similar to 90%)

Recommended IIR filter settings (shown will be only the r_factor value):

- Strong IIR filter: -0.9
- Less strong IIR filter: -0.85
- Ideal IIR filter: -0.8
- Low IIR filter: -0.5

Only the first factor is used for each f_factor and r_factor. Unused signs just stay 0. (3 signs for f_factor and 2 signs for r_factor)

Name	Address	Size	Function	Description	Unit	Information
CH1_ClientFlt-Enable	H0020019	1 Byte	Read/Write	CH1 IIR filter enable		
CH1_ClientFlt-f_factor	H002001A	12 Byte	Read/Write	CH1 Forward factors		Must be set in with (+)
CH1_ClientFlt-r_factor	H0020026	8 Byte	Read/Write	CH1 Reward factors		Must be set in with (-)
CH2_ClientFlt-Enable	H0020032	1 Byte	Read/Write	CH2 IIR filter enable		
CH2_ClientFlt-f_factor	H0020033	12 Byte	Read/Write	CH2 Forward factors		Must be set in with (+)
CH2_ClientFlt-r_factor	H002003F	8 Byte	Read/Write	CH2 Reward factors		Must be set in with (-)

Also it is possible to capture the measurement signal before the pressure compensation will be applied on it. So both values are available.

CH1_pressure_Mod_in	H0060012	4 Byte	Read	CH1 pre-pressure compensation		Signal without pressure comp. for CH1
CH2_pressure_Mod_in	H0060024	4 Byte	Read	CH2 pre-pressure compensation		Signal without pressure comp. for CH2

6 Addendum

An overview regarding the protocol structure is available if a web browser or the delivered host software is used and following address is set in: http://127.0.0.1:8080/device1/structure.xml. Depending on which device is connected and selected.

A copy of the complete structure.xml can be found on the CD delivered with the S-AGM Plus.