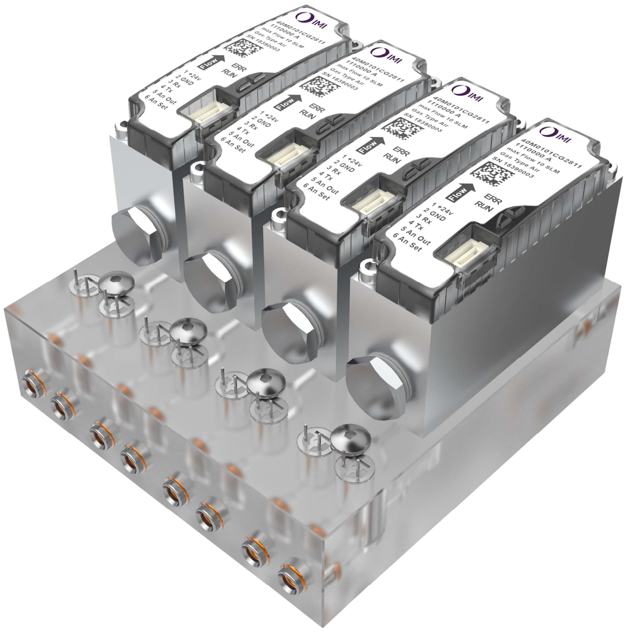
CHIPREG MFC
User Manual &
Protocol Command Set



Manual version: V.4.28

Contents

1 Abbrevations and Acronyms 8
2 Purpose 9
3 Analog I/O 9
3.1 Analog Input 9
3.2 Analog Output 10
4 Digital Communication 11
4.1 RS232/RS485 Peripheral settings 11
4.2 Command Structure 11
4.3 CRC16 Computation 14
4.4 Commands Access 16
4.5 Non Volatile Memory 18
5 Commands Description 20
5.1 Communication mode 20
5.1.1 IMI FAS protocol MODW 20
5.1.1 Modbus RTU protocol to switch from MODBUS to the ASCII mode 20
5.2 Mass Flow Setpoint Read 22
5.2.1 IMI FAS protocol MFSR 22
5.2.2 Modbus RTU protocol to get a setpoint mass flow 23
5.3 Mass Flow Setpoint Write 23
5.3.1 IMI FAS protocol MFSW 23
5.3.2 Modbus RTU protocol to set a setpoint mass flow 25
5.4 Modbus RTU protocol to get the full scaled mass flow 25
5.5 Valve Current Setpoint Read: VCSR 26
5.6 Valve Current Setpoint Write: VCSW 27
5.7 Control Read : CTRR 28
5.8 Control Write 29
5.9 CTRW 29
5.10 Controller Read : CTLR 31
5.11 Controller Write : CTLW 33

5.12 Raw Mass Flow Read: RMFR 35
5.13 Scaled Mass Flow Read 36
5.13.1 IMI FAS protocol SMFR 36
5.13.2 Modbus RTU protocol to get the scaled mass flow 36
5.14 Raw Valve Current Read: RVCR 38
5.15 Scaled Valve Current Read: SVCR 39
5.16 Analog Output Selection Read: AOSR 40
5.17 Analog Output Selection Write: AOSW 41
5.18 Drive Pwm Setpoint Read: DPSR 43
5.19 Drive Pwm Setpoint Write: DPSW 44
5.20 Setpoint Input Selection Read 45
5.20.1 IMI FAS protocol SISR 45
5.20.2 Modbus RTU protocol to get the setpoint input type 46
5.21 Setpoint Input Selection Write 47
5.21.1 IMI FAS protocol SISW 47
5.21.2 Modbus RTU protocol to select the setpoint input type 48
5.22 System Reset : 49
5.22.1 IMI FAS protocol SYRN 49
5.22.1 Modbus RTU protocol to make a HW reset: 49
5.23 Raw Adc Setpoint Read: RASR 51
5.24 Scaled Adc Setpoint Read: SASR 52
5.25 Effective Setpoint Read : EFSR 53
5.26 Raw Dac User Read: RDUR 54
5.27 Raw Dac User Write: RDUW 55
5.28 Scaled Dac User Read: SDUR 56
5.29 Scaled Dac User Write: SDUW 57
5.30 Hardware Status Read 58
5.30.1 IMI FAS protocol HWSR 58
5.30.2 Modbus RTU protocol to get the chipreg's HW status 60
5.31 Raw Drive Pwm Read : RDPR 60
5.32 Raw Analog Output Read: RAOR 62

5.33 Scaled Analog Output Read: SAOR 63
5.34 Raw Drive Voltage Read: RDVR 64
5.35 Scaled Drive Voltage Read: SDVR 65
5.36 Raw Gas Temperature Read : RGTR 66
5.37 Scaled Gas Temperature Read: SGTR 67
5.38 Non-Volatile Memory Status Read: NMSR 68
5.39 Non-Volatile Memory Status Write 69
5.40 Non-Volatile Memory Write Memory: NMWM 71
5.41 Calibration Read : CALR 72
5.42 Calibration Write : CALW 73
5.43 Configuration Read : CONR 74
5.44 Configuration Write : CONW 75
5.45 Identification Read : IDER 76
5.46 Identification Write : IDEW 76
5.47 Factory Password Write : FPWW 78
5.48 Sensor Information Table Read : SITR 79
5.49 Device Address 81
5.49.1 IMI FAS protocol DADR 81
5.49.2 Modbus RTU protocol to get the MFC's device address 82
5.50 Device Address Write 82
5.50.1 IMI FAS protocol DADW 82
5.50.2 Modbus RTU protocol to set the MFC's device address 83
5.51 User Gas Coefficient Read: UGCR 83
5.52 User Gas Coefficient Write: UGCW 85
5.53 Impedance Switch Read: ISWR 86
5.54 Impedance Switch Write: ISWW 87
5.55 Baud Rate Read 88
5.55.1 IMI FAS protocol BDRR 88
5.55.2 Modbus RTU protocol to get the chipreg's baudrate 88
5.56 Baud Rate Write 90
5.56.1 IMI FAS protocol BDRW 90

5.56.2 Modbus RTU protocol to set the chipreg's baudrate 90
5.57 User Pid Parameters Read: UPPR 92
5.58 User Pid Parameters Write: UPPW 94
5.59 User Unit Mode Read: UUMR 96
5.60 Modbus RTU protocol to get the device unit 96
5.61 User Unit Mode Write: UUMW 98
5.61.1 Modbus RTU protocol to set the MFC's device address 99
5.62 Modbus RTU protocol to get the device gas 99
5.63 Multi Gas Factor Read:MGFR 100
5.64 Multi Gas Selection Read:MGSR 101
5.64.1 Modbus RTU protocol to get gas selection 101
5.65 Multi Gas Selection Write:MGSW 103
5.65.1 Modbus RTU protocol to set gas selection 103
5.66 Security Mode Read 105
5.66.1 IMI FAS protocol STYR 105
5.66.2 Modbus RTU protocol to get the security mode 106
5.67 Security Mode Write 107
5.67.1 IMI FAS protocol STYW 107
5.67.2 Modbus RTU protocol to set the security mode 107
5.68 Temperature Compensation Selection Read:TCSR 108
5.69 Temperature Compensation Selection Write: TCSW 110
5.70 Boost Initial Value Read: BIVR 111
5.71 Boost Initial Value Write: BIVW 112
5.72 Mass Flow Average Read: MFAR 113
5.73 Mass Flow Average Write : MFAW 114
5.74 Switch communication mode 114
5.75 Read Fw version : FWVR 116
5.76 Switch communication mode 116
5.77 Regulation time period Write : REGW 118
5.1 Regulation time period Read : REGR 120
5.2 DP raw data average Write : DPAW 120

5.1 DP raw data average Read : DPAR ... 121
5.1 Fw type Read : FWTY ... 123
5.2 Modbus RTU protocol to get parity & stop bit ... 124
5.3 Modbus RTU protocol to set parity & stop bit ... 124
5.4 Modbus RTU protocol to get device unit ... 124
5.5 Modbus RTU protocol to set device unit ... 125
6 Computation of the Digital I/O data ... 125
6.1 Mass Flow ... 125
6.2 Valve current ... 127
6.3 Drive Pwm ... 128
6.4 Adc Setpoint ... 129
6.5 Effective Setpoint ... 130
6.6 Dac User ... 131
6.7 Analog Output ... 132
6.8 Drive Voltage ... 133
6.9 Gas Temperature ... 134
6.10 User Gas Coefficient ... 135
6.11 User Pid Parameters ... 136
6.12 Multi Gas Factor and Selection ... 137
6.13 Boost Initial Value ... 139
7 Troubleshooting ... 140
7.1 LED behaviour ... 140
7.2 ERROR codes ... 140
7.3 Back to default Modbus RTU (since fw 1.07.04) ... 141
8 Scripts ... 143
8.1 Default State ... 143
8.2 Examples ... 143
8.2.1 Scenario1: Change the device address ... 143
8.2.2 Scenario2: Check the main parameters ... 145
8.2.3 Scenario3: Enter in digital mode for mass flow control ... 145
8.2.4 Scenario4: Setpoint and mass flow reading ... 147

8.2.5 Scenario5 : Read and write the user unit mode 147
8.2.6 Scenario 6 : Read and write the user gas coefficient 149
8.3 Modbus examples 150
9 Annex 152
9.1 User data description 152
9.1.1 Data application extended description 152
9.1.2 Data system description 152
9.2 Calibration Data Description 154
9.3 Configuration Data Description 158
9.4 Identification Data Description 160
9.5 Device Address 163
9.6 Security Mode 163

Abbrevations and Acronyms

0d Number	: decimal format	
0x Number	: hexadecimal format	
0b Number	: binary format	
Uint8	: unsigned integer of 8 bit	(0d0..0d255)
Uint16	: unsigned integer of 16 bit	(0d0..0d65535)
Uint32	: unsigned integer of 32 bit	(0d0..0d4294967295)
Int8	: signed integer of 8 bit	(0d-128..0d127)
Int16	: signed integer of 16 bit	(0d-32768..0d32767)
Int32	: signed integer of 32 bit	(0d-2147483648..0d2147483647)
Float32	: single precision floating point	(IEEE754)
NVM	: Non volatile memory	

Purpose

This document serves as a reference guide for users who intend to operate the Chipreg MFC in either Analog or Digital mode. If you are looking for a quick introduction to digital communication, I recommend reading the 'Scripts' chapter.

Additionally, please note that the Modbus RTU functionality is available starting from firmware version 1.07.04.

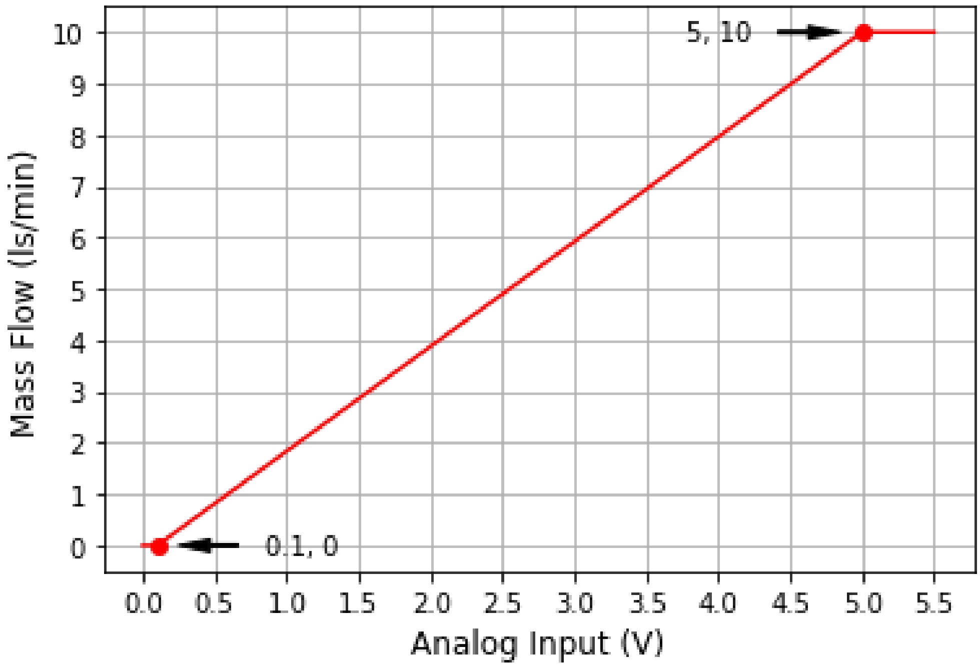
Analog I/O

Analog Input

The Analog Input Voltage applied at the 'Analog Setpoint Flow' pin is the Mass Flow Setpoint represented between 0 and 5 V:

Analog Input Voltage (V)	**Mass Flow Setpoint (ls/min)**
0 →0.1	0
5	Full Scale

The chart below is for the Chipreg MFC 10 ls/min.

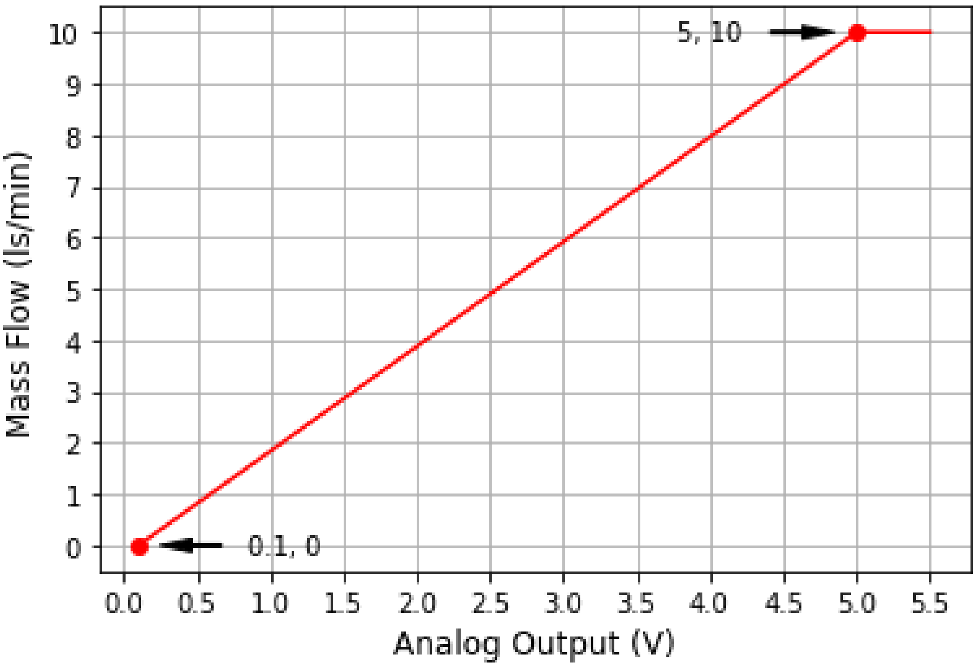


Analog Output

The Analog Output Voltage at the ‘Analog Flow Setpoint’ pin is the regulated Mass Flow represented between 0 and 5 V:

Regulated Mass Flow (ls/min)	**Analog Output Voltage (V)**
0	0.1
Full Scale	5

The chart below is for the Chipreg MFC 10 ls/min.



Digital Communication

RS232/RS485 Peripheral settings

The settings are the following:

- Baud rate 115200 max
- Data 8 bits
- Parity none / (even for Modbus RTU)
- Data bits 8
- Stop bits 1
- Handshaking none
- Level EIA232/EIA485

Command Structure

The serial line uses characters to send / receive 8 bits (1 octet or 1 byte) packets. For numbers all values must be specified in hex format. Thus, each octet needs 2 characters. A command operation integrates two phases: send and receive with always the same structure.

Device Address : 2 char
Command Code : 6 char
Data : n char
CRC16 Code : 4 char

After sending a command (master) the Chipreg MFC (slave) must reply in accordance with the same following format:

Command Send (from Master)

Device		Command						Data				CRC16			
A0	A1	'-'	'>'	C0	C1	C2	C3	D0	D1	D2	Dn	R0	R1	R2	R3

Command Receive (from Slave)

Device		Command						Data				CRC16			
A0	A1	'-'	'>'	C0	C1	C2	C3	D0	D1	D2	Dn	R0	R1	R2	R3

Notice
For numbers (always in hex format) the letters (a, b, c, d, e and f can be written either in uppercase or lowercase). However, for text, the system is case-sensitive.

The default address of all devices is 0XFF

Example:
The user requests a mass flow read with the command ‘SMFR’

Command Send (from Master)

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘M’	‘F’	‘R’	‘a’	‘a’	‘7’	‘e’

- The device address is 0x01 → ‘01’
- The command is composed of 2+4 letters →’->SMFR’
- No Data to send → void
- The CRC16 code of the whole character string ’01->SFMR’ is 0xaa7e → ’aa7e’

Command Receive (from Slave)

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘M’	‘F’	‘R’	‘0’	‘0’	‘0’	‘0’	‘1’	‘3’	‘2’	‘3’

- The device number is 0x01 → ‘01’
- The command is composed of 2+4 letters → ‘>SFMR’
- The returned data from the Chipreg MFC is a 16 bits number of 0x0000 → ’0000’
- The CRC16 code of the whole character string ’01->SFMR0000’ is 0x1323→ ‘1323’

In the case where a number is bigger than 8 bits (16 or 32 bits), we must split that number in several octets. An example where the 16 bits number 0d15893 must be write on the serial line:

0d15893 = 0x3e15 = 0b00111110 00010101
MSByte = 0x3e
LSByte = 0x15

Thus, we need 4 chars:
char0 = ‘3’
char1 = ‘e’
char2 = ‘1’
char3 = ‘5’

For reading operation MSByte and LSByte must be merged together to find the original number. The MSByte must be multiplied by 2^8 (shifted to the left 8 times) and added to LSByte.
We receive through the serial line 4 chars:

char0 = ‘3’
char1 = ‘e’
char2 = ‘1’

char3 = ‘5’

We convert it in 2 bytes:

MSByte = 0x3e
LSByte = 0x15

Number = (MSBytes << 8) + LSByte = 0x3E00 + 0x15 = 0x3E15 = 0d15893

CRC16 Computation

The CRC16 computation (checksum) is performed in accordance with the following algorithm:

```
//      Crc16 Modbus Checksum computation.
// Note                  -
// *charData Array of characters.
// uint8Nbr              Numbers of characters to receive.
// uint16Crc16           Output value.
uint16_t Crc16ModBusComputation (char* charData, uint8_t uint8Nbr)
{
        uint16_t                uint16Crc16 = 0xFFFF;
        uint8_t                 uint8Position;
        uint8_t                 uint8Shift;

        for (uint8Position = 0; uint8Position < uint8Nbr;uint8Position++)
        {
                uint16Crc16 ^= (uint16_t)charData[uint8Position];

                for (uint8Shift = 8; uint8Shift != 0; uint8Shift--)
                {
                        if ((uint16Crc16 & 0x0001) != 0)
                        {
                                uint16Crc16 >>= 1;
                                uint16Crc16 ^= 0xA001;
                        }
                        else uint16Crc16 >>= 1;
                }
        }
        return uint16Crc16;
}
```

The Master can avoid the CRC16 computation replacing it by the character string ‘XXXX’.

Example:
The user requests a mass flow read with the command ‘SMFR’ avoiding the CRC16 computation.

Command Send (from Master)

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘M’	‘F’	‘R’	‘X’	‘X’	‘X’	‘X’

- The device number is 0x01 → ‘01’
- The command is composed of 2+ 4 letters → ‘->SFMR’
- No Data to send → void
- Instead of CRC16 code, the user can use ‘XXXX’ (capital letters) → ’XXXX’

Commands Access

The list here after concerns the command availability.

Command	Type	Command	Type
MFSR	U	CALR	F
MFSW	U	CALW	FPW/M
VCSR	U	CONR	F
VCSW	U	CONW	FPW/M
CTRR	U	IDER	U
CTRW	U	IDEW	FPW/M
CTLR	U	FPWW	F
CTLW	U/M	SITR	F
RMFR	F	DADR	U
CRSN	U	DADW	U/M
SMFR	U	UGCR	U
RVCR	F	UGCW	U/M
SVCR	U	ISWR	U
AOSR	U	ISWW	U/M
AOSW	U	BDRR	U
DPSR	U	BDRW	U/M
DPSW	U	UPPR	U
SISR	U	UPPW	U/M
SISW	U/M	UUMR	U
SYRN	U	UUMW	U/M
RASR	F	MGFR	U
SASR	U	MGSR	U
RDUR	F	MGSW	U
RDUW	F	SYTR	F
SDUR	U	SYTW	F/M
SDUW	U	TCSR	F
EFSR	U	TCSW	F
HWSR	U	BIVR	F
RDPR	U	BIVW/M	F
RAOR	F	MFAR	U
SAOR	U	MFAW	U
RDVR	F		
SDVR	U		
RGTR	F		
SGTR	U		

NMSR	F		
NMSW	FPW/M		
NMWM	U		

U : User (customer) oriented command
F : Factory oriented command (but available for user)
FPW : Need factory password (no access for user)
M : Written data storable in the non-volatile memory

Non Volatile Memory

Some user-oriented commands allow to store values in NVM :

- CTLW
- SISW
- DADW
- UGCW
- ISWW
- BDRW
- UPPW
- UUMW
- BIVW
- MFAW
- STYW

Important

- To store the written values in NVM, the user must perform the command: NMWM (write data in NVM and automatic system reset).

- The written values will be immediately activated before NMWM command. This is true for the following commands:

  CTLW
  SISW
  UGCW
  UPPW
  UUMW
  BIVW
  MFAW
  STYW

- The written values will be activated after NMWM command. This is true for the following commands:

  DADW
  ISWW
  BDRW

Commands Description

Communication mode

This function could be used to change the communication type

IMI FAS protocol MODW

Change ASCII to the MODBUS mode.
Name
MODW

Purpose.

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Value	Notice
Mode type	Uint8	2	The system should be previously initialised as `NVM_COMPLETE = 1`

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'f'	'f'	'-'	'>'	'M'	'O'	'D'	'W'	'0'	'2'	'X'	'X'	'X'	'X'

Command Receive
After receiving the command, the system is restarted therefore there is no response.

Modbus RTU protocol to switch from MODBUS to the ASCII mode

Fct code	Address	Size	Range	Note

Write Single Holding Register (6)	0x2000	16bits	1	The system should be previously initialised as `NVM_COMPLETE_MODBUS = 2`

Example
Address = 0xEB, value = 1
Request : EB 06 20 00 00 00 94 C0
Response: -- *After receiving the command, the system is restarted therefore there is no response.*

Mass Flow Setpoint Read

This function sends back the latest mass setpoint that has been written by the function "mass flow setpoint".

IMI FAS protocol MFSR

Name
MFSR

Purpose
Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Mass Flow Setpoint	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'M'	'F'	'S'	'R'	'd'	'0'	'0'	'7'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'M'	'F'	'S'	'R'	'0'	'0'	'c'	'8'	'a'	'0'	'2'	'6'

- The mass flow setpoint read is 0x00c8 (0d200).

Modbus RTU protocol to get a setpoint mass flow

Fct code	Address	Size	Range
Read Holding registers (3)	8	16bits	0 – 4095

Example:
Address = 0xEA, value = 2000(0x07D0)
Request : EA 03 00 08 00 01 12 D3
Response : EA 03 02 07 D0 9F FF

Mass Flow Setpoint Write

This function sends a mass flow setpoint to the system.

IMI FAS protocol MFSW

Name
MFSW

Purpose
Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Mass Flow Setpoint	Uint16	0x0000 (0d0000)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Data Receive

void

Example

Command Send

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘M’	‘F’	‘S’	‘W’	‘0’	‘0’	‘c’	‘8’	‘a’	‘0’	‘e’	‘a’

Command Receive

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘M’	‘F’	‘S’	‘W’	‘d’	‘3’	‘c’	‘7’

- The mass flow setpoint written is 0x00c8 (0d200)

Modbus RTU protocol to set a setpoint mass flow

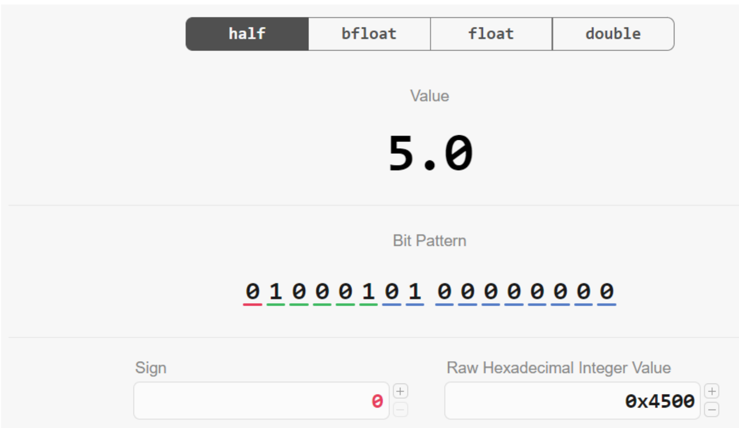
Fct code	Address	Size	Range
Write Single Holding Register (6)	8	16bits	0 – 4095

Example:
Address = 0xEA, value = 2000(0x07D0)
Request : EA 06 00 08 07 D0 1C BF
Response : EA 06 00 08 07 D0 1C BF

Modbus RTU protocol to get the full scaled mass flow

Fct code	Address	Size	Range
Read Holding register (3)	0x2F	16bits	Half float

Example:
Address = 0xEB, value = 5.0(0x4500)
Request : EB 03 00 2F 00 01 A3 09
Response : EB 03 02 45 00 93 03



*https://float.exposed/

Valve Current Setpoint Read: VCSR

Name
VCSR

Purpose
Read the last valve current setpoint written

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Valve Current Setpoint	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘V’	‘C’	‘S’	‘R’	‘3’	‘5’	‘1’	‘1’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-’	‘>’	‘V’	‘C’	‘S’	‘R’	‘0’	‘b’	‘b’	‘8’	‘5’	‘e’	‘9’	‘3’

- The valve current read is 0x0bb8 (0d3000).

Valve Current Setpoint Write: VCSW

Name
VCSW

Purpose
Write the valve current setpoint.

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Valve Current Setpoint	Uint16	0x0000 (0d0000)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Data Receive

void

Example

Command Send

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'V'	'C'	'S'	'W'	'0'	'b'	'b'	'8'	'5'	'e'	'5'	'f'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'V'	'C'	'S'	'W'	'3'	'6'	'd'	'1'

- The valve current setpoint written is 0x0bb8 (0d3000)

Control Read : CTRR

Name
CTRR

Purpose
Read the control configuration.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Control	Uint8	0x00 (0d0)	0x03 (0d3)	0..1	1)

1) 0x00 : No Control
   0x01 : Valve Current
   0x02 : Mass Flow
   0x03 : Drive Pwm

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘C’	‘T’	‘R’	‘R’	‘a’	‘d’	‘a’	‘4’

Command Receive

Device		Command						Data		CRC16			
‘0’	‘1’	‘-’	‘>’	‘C’	‘T’	‘R’	‘R’	‘0’	‘2’	‘a’	‘b’	‘2’	‘e’

- The control configuration read is 0x02 (0d2).

Control Write

This function could be used to configure the nature of the setpoint.

CTRW

Name
CTRW

Purpose
Write the control configuration. After a ‘CTRW’, the CTLW must rewrite.

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Control	Uint8	0x00 (0d0)	0x03 (0d3)	0..1	1)

1) 0x00 : No Control
   0x01 : Valve Current
   0x02 : Mass Flow
   0x03 : Drive Pwm

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘C’	‘T’	‘R’	‘W’	‘0’	‘2’	‘a’	‘9’	‘3’	‘e’

Command Receive

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘C’	‘T’	‘R’	‘W’	‘a’	‘e’	‘6’	‘4’

- The control configuration written is 0x02 (0d2)

Controller Read : CTLR

Name
CTLR

Purpose
Read the controller configuration.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Controller	Uint8	0x00 (0d0)	0x05 (0d5)	0..1	1)

1) 0x00 : No Controller
   0x01 : Basic
   0x02 : Slow PID
   0x03 : Medium PID
   0x04 : Fast PID
   0x05 : User PID
   0x06 : Drive Pwm

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'C'	'T'	'L'	'R'	'0'	'd'	'a'	'd'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'C'	'T'	'L'	'R'	'0'	'2'	'8'	'0'	'2'	'8'

- The controller configuration read is 0x02 (0d2).

Controller Write : CTLW

Name
CTLW

Purpose
Write the controller configuration.

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Controller	Uint8	0x00 (0d0)	0x05 (0d5)	0..1	1)

1) 0x00 : No Controller
0x01 : Basic
0x02 : Slow PID
0x03 : Medium PID
0x04 : Fast PID
0x05 : User PID
0x06 : Drive Pwm

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘C’	‘T’	‘L’	‘W’	‘0’	‘2’	‘8’	‘1’	‘3’	‘8’

Command Receive

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘C’	‘T’	‘L’	‘W’	‘0’	‘e’	‘6’	‘d’

- The controller configuration written is 0x02 (0d2)

Raw Mass Flow Read: RMFR

Name
RMFR

Purpose
Read a raw data from the mass flow sensor

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw Mass Flow	Int16	0x0000 (0d)	0xFFFF (0d±32767)	0..3	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘M’	‘F’	‘R’	‘5’	‘6’	‘7’	‘f’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘M’	‘F’	‘R’	‘0’	‘0’	‘0’	‘1’	‘1’	‘f’	‘2’	‘3’

- The raw data read is 0x0001 (0d1).

Scaled Mass Flow Read

This function provides scaled raw data.

IMI FAS protocol SMFR

Name
SMFR

Purpose
Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Scaled Mass Flow	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘M’	‘F’	‘R’	‘a’	‘a’	‘7’	‘e’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘M’	‘F’	‘R’	‘0’	‘0’	‘0’	‘0’	‘1’	‘3’	‘2’	‘3’

- The mass flow read is 0x0000 (0d0).

Modbus RTU protocol to get the scaled mass flow

Fct code	Address	Size	Range

Read Holding registers (3)	0x1110	16bits*	0 – 4095

Example:
Address = 0xEA, value = 2000(0x07D0)
Request : EA 03 11 10 00 01 97 E8
Response : EA 03 02 07 D0 9F FF

Raw Valve Current Read: RVCR

Name
RVCR

Purpose
Read a raw data from the valve current ADC

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw Valve Current	Uint16	0x0000 (0d)	0x0FFF (0d4095)	0..3	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘V’	‘C’	‘R’	‘0’	‘1’	‘0’	‘c’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-	‘>’	‘R’	‘V’	‘C’	‘R’	‘0’	‘0’	‘0’	‘0’	‘8’	‘b’	‘4’	‘9’

- The raw data read is 0x0000 (0d0).

Scaled Valve Current Read: SVCR

Name
SVCR

Purpose
Read the valve current

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Scaled Valve Current	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'V'	'C'	'R'	'f'	'd'	'0'	'd'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'S'	'V'	'C'	'R'	'0'	'0'	'0'	'0'	'4'	'7'	'8'	'8'

- The valve current read is 0x0000 (0d0).

Analog Output Selection Read: AOSR

Name
AOSR

Purpose
Read the analog output selection.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Analog Output Selection	Uint8	0x00 (0d0)	0x03 (0d3)	0..1	1)

1) 0x00 : No Analog Out
   0x01 : Valve Current
   0x02 : Mass Flow
   0x03 : Scaled User
   0x04 : Raw User

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'A'	'O'	'S'	'R'	'8'	'2'	'd'	'4'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'A'	'O'	'S'	'R'	'0'	'2'	'b'	'4'	'4'	'a'

- The analog output selection read is 0x02 (0d2).

Analog Output Selection Write: AOSW

Name
AOSW

Purpose
Write the analog output selection.

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Analog Output Selection	Uint8	0x00 (0d0)	0x04 (0d4)	0..1	1) 2)

1) 0x00 : No Analog Out
   0x01 : Valve Current
   0x02 : Mass Flow
   0x03 : Scaled User
   0x04 : Raw User

2) For 'User' mode see command 'SDUW' and 'RDUW'

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'A'	'O'	'S'	'W'	'0'	'2'	'b'	'5'	'5'	'a'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'A'	'O'	'S'	'W'	'8'	'1'	'1'	'4'

- The analog output selection written is 0x02 (0d2)

Drive Pwm Setpoint Read: DPSR

Name
DPSR

Purpose
Read the last drive pwm setpoint written

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Drive Pwm Setpoint	Uint16	0x0000 (0d0)	0x0F9F (0d3999)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'D'	'P'	'S'	'R'	'8'	'8'	'e'	'5'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'D'	'P'	'S'	'R'	'0'	'5'	'd'	'c'	'c'	'1'	'c'	'2'

- The drive pwm setpoint read is 0x05dc (0d1500).

Drive Pwm Setpoint Write: DPSW

Name
DPSW

Purpose
Write the drive pwm setpoint.

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Drive Pwm Setpoint	Uint16	0x0000 (0d0000)	0x0F9F (0d3999)	0..3	1)

1) See section 'Computation of the digital I/O data'

Data Receive

void

Example

Command Send

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'D'	'P'	'S'	'W'	'0'	'5'	'd'	'c'	'c'	'1'	'0'	'e'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'D'	'P'	'S'	'W'	'8'	'b'	'2'	'5'

- The drive pwm setpoint written is 0x05dc (0d1500).

Setpoint Input Selection Read

This function could be used to get the setpoint input type.

IMI FAS protocol SISR

Name
SISR

Purpose
Read the setpoint input selection.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Setpoint Input Selection	Uint8	0x00 (0d0)	0x02 (0d3)	0..1	1) 2)

1) 0x00 : No Setpoint Input
   0x01 : Adc (analog)
   0x02 : RS232 (digital)

2) Analog Input option only for the mass flow control.
   (valve current and drive pwm setpoint through RS232 only)

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'I'	'S'	'R'	'f'	'b'	'3'	'1'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'S'	'I'	'S'	'R'	'0'	'1'	'c'	'7'	'8'	'1'

- The setpoint input selection read is 0x01 (0d1).

Modbus RTU protocol to get the setpoint input type

Fct code	Address	Size	Range
Read Holding register (3)	0x1F00	16bits	0 – 2

Example:
Address = 0xFF
Request : FF 03 1F 00 00 01 96 00
Response : FF 03 02 00 02 10 51

Setpoint Input Selection Write

This function could be used to select the setpoint input type.

IMI FAS protocol SISW

Name
SISW

Purpose
Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Setpoint Input Selection	Uint8	0x00 (0d0)	0x02 (0d2)	0..1	1) 2)

1) 0x00 : No Setpoint Input
   0x01 : Adc (analog)
   0x02 : RS232 (digital)

2) Analog Input option only for the mass flow control.
   (valve current and drive pwm must be set through RS232 only)

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘I’	‘S’	‘W’	‘0’	‘1’	‘c’	‘6’	‘9’	‘1’

Command Receive

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘I’	‘S’	‘W’	‘f’	‘8’	‘f’	‘1’

- The setpoint input selection written is 0x01 (0d1)

Modbus RTU protocol to select the setpoint input type

Fct code	Address	Size	Range
Write Single Holding Register (6)	0x1F00	16bits	0 – 2

Example:

Address = 0xEA, value = 2
Request : EA 06 E0 01 00 02 79 10
Response : EA 06 E0 01 00 02 79 10

System Reset :

This function could be used to made a HW reset.

IMI FAS protocol SYRN

Name
SYRN

Purpose
Perform a soft reset of device.

Data Send (char) : 0
Data Receive (char) : 0

Data Send

void

Data Receive

void

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'Y'	'R'	'N'	'6'	'7'	'3'	'0'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'Y'	'R'	'N'	'6'	'7'	'3'	'0'

Modbus RTU protocol to make a HW reset:

Fct code	Address	Size	Range
Write Single Coil (5)	0x2500	16bits	x

Example:
Address = 0xEA, value = 1
Request : EB 05 25 00 00 00 D0 0C
Response : -- *System is restarting then no response*

Raw Adc Setpoint Read: RASR

Name
RASR

Purpose
Read a raw data from the setpoint ADC

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw Adc Setpoint	Uint16	0x0000 (0d)	0x0FFF (0d4095)	0..3	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘A’	‘S’	‘R’	‘c’	‘5’	‘b’	‘1’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘A’	‘S’	‘R’	‘0’	‘0’	‘0’	‘0’	‘1’	‘a’	‘2’	‘c’

- The raw data read is 0x0000 (0d0).

Scaled Adc Setpoint Read: SASR

Name
SASR

Purpose
Read the analog Input setpoint

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Scaled Adc Setpoint	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'A'	'S'	'R'	'3'	'9'	'b'	'0'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'S'	'A'	'S'	'R'	'0'	'0'	'0'	'0'	'd'	'6'	'e'	'd'

- The analog input setpoint read is 0x0000 (0d0).

Effective Setpoint Read : EFSR

Name
EFSR

Purpose
Read the last effective setpoint written

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Effective Setpoint	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1) 2)

1) See section 'Computation of the digital I/O data'

2) Depend on command 'CTRW' and 'SISW'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'E'	'F'	'S'	'R'	'b'	'0'	'0'	'5'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'E'	'F'	'S'	'R'	'0'	'0'	'0'	'0'	'3'	'0'	'1'	'a'

- The effective setpoint read is 0x0000 (0d0).

Raw Dac User Read: RDUR

Name
RDUR

Purpose
Read a raw data from the DAC user

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw Dac User	Uint16	0x0000 (0d)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'R'	'D'	'U'	'R'	'6'	'4'	'a'	'2'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'R'	'D'	'U'	'R'	'0'	'0'	'6'	'4'	'1'	'f'	'7'	'b'

- The raw data read is 0x0064 (0d100).

Raw Dac User Write: RDUW

Name
RDUW

Purpose
Write a raw data to the DAC user

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Raw Dac User	Uint16	0x0000 (0d0000)	0x0FFF (0d4095)	0..3	1) 2)

1) See section 'Computation of the digital I/O data'
2) To use this command, the user must select the 'User' option in command 'AOSW'

Data Receive

void

Example

Command Send

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'R'	'D'	'U'	'W'	'0'	'0'	'6'	'4'	'1'	'f'	'b'	'7'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'R'	'D'	'U'	'W'	'6'	'7'	'6'	'2'

- The dac user written is 0x0064 (0d100)

Scaled Dac User Read: SDUR

Name
SDUR

Purpose
Read a scaled data from the DAC user

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Scaled Dac User	Uint16	0x0000 (0d)	0x0FFF (0d4095)	0..3	1)

2) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'D'	'U'	'R'	'9'	'8'	'a'	'3'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'S'	'D'	'U'	'R'	'0'	'7'	'd'	'0'	'b'	'1'	'3'	'7'

- The scaled data read is 0x07d0 (0d2000).

Scaled Dac User Write: SDUW

Name
SDUW

Purpose
Write a scaled data to the DAC user

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Scaled Dac User	Uint16	0x0000 (0d0000)	0x0FFF (0d4095)	0..3	1) 2)

3) See section 'Computation of the digital I/O data'

4) To use this command, the user must select the 'User' option in command 'AOSW'

Data Receive

void

Example

Command Send

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'S'	'D'	'U'	'W'	'0'	'7'	'd'	'0'	'b'	'1'	'f'	'b'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'D'	'U'	'W'	'9'	'b'	'6'	'3'

- The scaled data written is 0x07d0 (0d2000)

Hardware Status Read

This function could be used to get the chipreg's HW status.

IMI FAS protocol HWSR

Name
HWSR

Purpose
Read the status of the critical parts of the device.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Hardware Status	Uint8	0x00 (0d0)	0xFF (0d255)	0..1	1)

1) bx00000000 : No trouble
bx00000001 : Control Saturation
bx00000010 : Control Overload
bx00000100 : Drive Voltage High
bx00001000 : Drive Voltage Low
bx00010000 : Reserved1
bx00100000 : Reserved2
bx01000000 : Reserved3
bx10000000 : Sensor Lost

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'H'	'W'	'S'	'R'	'1'	'9'	'5'	'7'

Command Receive

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘H’	‘W’	‘S’	‘R’	‘0’	‘0’	‘e’	‘e’	‘e’	‘b’

- The hardware status read is 0x00 (0d0) → 0b00000000
- That means: No trouble

Modbus RTU protocol to get the chipreg's HW status

Fct code	Address	Size	Range
Read Holding register (3)	0x1112	16bits	0 ..255

Example
Address = 0xEB, value = 0
Request : EB 03 11 12 00 01 37 F9
Response : EB 03 02 00 00 A1 93

Raw Drive Pwm Read : RDPR

Name
RDPR

Purpose
Read the raw drive pwm.

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw Drive Pwm	Uint16	0x0000 (0d)	0x0F9F (0d3999)	0..3	1)

3) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'R'	'D'	'P'	'R'	'3'	'4'	'a'	'1'

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-’	‘>’	‘R’	‘D’	‘P’	‘R’	‘0’	‘0’	‘0’	‘0’	‘2’	‘9’	‘7’	‘9’

- The raw data read is 0x0000 (0d0).

Raw Analog Output Read: RAOR

Name
RAOR

Purpose
Read a raw data from the analog output ADC

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw Analog Output	Uint16	0x0000 (0d)	0x0FFF (0d4095)	0..3	1)

2) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'R'	'A'	'O'	'R'	'0'	'5'	'b'	'9'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'R'	'A'	'O'	'R'	'0'	'0'	'3'	'4'	'7'	'5'	'2'	'f'

- The raw data read is 0x0034 (0d52).

Scaled Analog Output Read: SAOR

Name
SAOR

Purpose
Read the analog output

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Scaled Analog Output	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

2) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'A'	'O'	'R'	'f'	'9'	'b'	'8'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'S'	'A'	'O'	'R'	'0'	'0'	'3'	'6'	'7'	'8'	'6'	'f'

- The analog output read is 0x0036 (0d54).

Raw Drive Voltage Read: RDVR

Name
RDVR

Purpose
Read a raw data from the drive voltage ADC

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw Drive Voltage	Uint16	0x0000 (0d)	0x0FFF (0d4095)	0..3	1)

3) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘D’	‘V’	‘R’	‘9’	‘4’	‘a’	‘2’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘D’	‘V’	‘R’	‘0’	‘7’	‘5’	‘2’	‘1’	‘f’	‘4’	‘a’

- The raw data read is 0x0752 (0d1874).

Scaled Drive Voltage Read: SDVR

Name
SDVR

Purpose
Read the drive voltage

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Scaled Drive Voltage	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'D'	'V'	'R'	'6'	'8'	'a'	'3'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'S'	'D'	'V'	'R'	'0'	'7'	'5'	'2'	'd'	'3'	'8'	'b'

- The analog output read is 0x0752 (0d1874).

Raw Gas Temperature Read : RGTR

Name
RGTR

Purpose
Read a raw data from the gas temperature sensor

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Raw GasTemperature	Int16	0x0000 (0d)	0xFFFF (0d±32767)	0..3	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘G’	‘T’	‘R’	‘f’	‘4’	‘5’	‘3’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘R’	‘G’	‘T’	‘R’	‘0’	‘0’	‘0’	‘0’	‘a’	‘d’	‘4’	‘b’

- The raw gas temperature read is 0x0000 (0d0).

Scaled Gas Temperature Read: SGTR

Name
SGTR

Purpose
Read a scaled data from the gas temperature sensor

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Scaled Gas Temperature	Uint16	0x0000 (0d0)	0x0FFF (0d4095)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'G'	'T'	'R'	'0'	'8'	'5'	'2'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'S'	'G'	'T'	'R'	'0'	'0'	'0'	'0'	'6'	'1'	'8'	'a'

- The scaled gas temperature read is 0x0000 (0d0).

Non-Volatile Memory Status Read: NMSR

Name
NMSR

Purpose
Read the status of the non-volatile memory.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Non-Volatile Memory Status	Uint8	0x00 (0d0)	0x01 (0d1)	0..1	1)

1) 0x00 : Nvm Incomplete
   0x01 : Nvm Complete

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'N'	'M'	'S'	'R'	'5'	'6'	'7'	'6'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'N'	'M'	'S'	'R'	'0'	'1'	'8'	'a'	'7'	'3'

- The non-volatile memory status read is 0x01 (0d1).

Non-Volatile Memory Status Write

This function could be used to configure the Non-Valatile Memory Status as Incomplete(0), Complete for IMI FAS protocol(1) or Complete for Modbus RTU protocol(2).
However, the command "NMWM" should be executed right after when the parameter to write is "Modbus RTU protocol(2)", so that MFC can be used in Modbus RTU.

Name
NMSW

Purpose
Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Non-Volatile Memory Status	Uint8	0x00 (0d0)	0x02 (0d2)	0..1	1)

1) 0x00 : Nvm Incomplete
0x01 : Nvm Complete
0x02 : Nvm Complete Modbus* (Until now there is no official firmware version supports this parameter.)

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'N'	'M'	'S'	'W'	'0'	'1'	'8'	'b'	'6'	'3'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'N'	'M'	'S'	'W'	'5'	'5'	'b'	'6'

- The non-volatile memory status written is 0x01 (0d1).

Non-Volatile Memory Write Memory: NMWM

Name
NMWM

Purpose
Perform a non-volatile memory write (the control CTR must be disabled)

Data Send (char) : 0
Data Receive (char) : 0

Data Send

void

Data Receive

void

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘N’	‘M’	‘W’	‘M’	‘5’	‘e’	‘3’	‘5’

Command Receive

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘N’	‘M’	‘W’	‘M’	‘5’	‘e’	‘3’	‘5’

Calibration Read : CALR

Name
CALR

Purpose
Read calibration data

Data Send (char) : 0
Data Receive (char) : 208

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Calibration Data				0..207	1)

1) See Annex for details.

Calibration Write : CALW

Name
CALW

Purpose
Write calibration data

Data Send (char) : 208
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Calibration Data				0..207	1)

1) See Annex for details.

Data Receive

void

Configuration Read : CONR

Name
CONR

Purpose
Read configuration data

Data Send (char) : 0
Data Receive (char) : 310

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Configuration Data				0..309	1)

1) See Annex for details.

Configuration Write : CONW

Name
CONW

Purpose
Write configuration data

Data Send (char) : 310
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Configuration Data				0..309	1)

1) See Annex for details.

Data Receive

void

Identification Read : IDER

Name
IDER

Purpose
Read identification data

Data Send (char) : 0
Data Receive (char) : 153

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Part Number	Char			0..152	1)

1) See Annex for details.

Identification Write : IDEW

Name
IDEW

Purpose
Write identification data

Data Send (char) : 153
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Part Number	Char			0..152	1)

1) See Annex for details.

Data Receive

void

Factory Password Write : FPWW

Name
FPWW

Purpose
Enter in factory mode

Data Send (char) : 8
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Factory Password	Uint32	0x00000000 (0d0)	0xFFFFFFFF (0d 4294967296)	0..7	1)

1) Only for factory calibration

Data Receive

void

Example

Command Send

Device		Command						Data	CRC16			
'0'	'1'	'-'	'>'	'F'	'P'	'W'	'W'	See below	'6'	'6'	'd'	'3'

- The data written is: 0x000000f5 (0d245)

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'E'	'R'	'R'	'N'	'0'	'7'	'0'	'b'	'a'	'7'

- This is the error message if the password is wrong

Sensor Information Table Read : SITR

Name
SITR

Purpose
Read information about main sensor

Data Send (char) : 0
Data Receive (char) : 21

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Sensor Type	Char			0..10	
Sensor ID	Char			11..12	
Sensor Week	Uint8	0x00 (0d0)	0xFF (0d0)	13..14	
Sensor Year	Uint8	0x00 (0d0)	0xFF (0d0)	15..16	
Sensor Sequence	Uint16	0x0000 (0d0)	0xFFFF (0d65535)	17..20	

1) The serial number includes Sensor ID, Week, Year and Sequence

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘I’	‘T’	‘R’	‘c’	‘b’	‘3’	‘3’

Command Receive

Device		Command						Data	CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘I’	‘T’	‘R’	See below	‘7’	‘c’	‘4’	‘f’

- Data: 'LMIS500BB3SAD12120064'

Device Address

This function provides the MFC's device address.

IMI FAS protocol DADR

Name
DADR

Purpose
Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Device Address	Uint8	0x00 (0d0)	0xFF* (0d255)	0..1	1)

1) See Annex for details.

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'D'	'A'	'D'	'R'	'7'	'd'	'b'	'a'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'D'	'A'	'D'	'R'	'0'	'1'	'9'	'5'	'6'	'6'

- The device address read is 0x01 (0d1).

*__Please Note:__
The address 255 (equivalent to 0xFF) is exclusively reserved for broadcasting purposes. It is strongly advised not to assign the slave device this address. In cases where you are uncertain about the device's address, you have the option to employ it for sending your request.

Modbus RTU protocol to get the MFC's device address

Fct code	Address	Size	Range
Read Holding registers (3)	1	16bits	0 – 255

By default, the Chipreg's address is set to 0xFF.
However it's no way to find out Chipreg's address by sending a read address command even using the 0xFF address! You have loop "the read address command" to check the response in order to get Chipreg's address.

Example:
Address = 0xFF, value = 0x00FF
Request : FF 03 00 01 00 01 C0 14
Response : FF 03 02 00 FF D1 D0

Device Address Write

This function could be used to change the MFC's device address.

IMI FAS protocol DADW

Name
DADW

Purpose
Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Device Address	Uint8	0x00 (0d0)	0xFF* (0d255)	0..1	1)

1) See Annex for details.

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'D'	'A'	'D'	'W'	'0'	'2'	'9'	'5'	'3'	'6'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'D'	'A'	'D'	'W'	'7'	'e'	'7'	'a'

- The address device written is 0x02 (0d2)

***Please Note:**
The address 255 (equivalent to 0xFF) is exclusively reserved for broadcasting purposes. It is strongly advised not to assign the slave device this address. In cases where you are uncertain about the device's address, you have the option to employ it for sending your request.

Modbus RTU protocol to set the MFC's device address

Fct code	Address	Size	Range
Write Single Holding Register (6)	1	16bits	0 – 255

Example
Address = 0xEA, value = 0x00EB
Request : EA 06 00 01 00 EB 8F 5E
Response : EA 06 00 01 00 EB 8F 5E

User Gas Coefficient Read: UGCR

Name
UGCR

Purpose
Read the user gas coefficient

Data Send (char) : 0

Data Receive (char) : 8

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
User Gas Coefficient	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	0..7	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'U'	'G'	'C'	'R'	'7'	'0'	'5'	'd'

Command Receive

Device		Command						Data	CRC16			
'0'	'1'	'-'	'>'	'U'	'G'	'C'	'R'	See below	'c'	'2'	'a'	'f'

- Data is: 0x3f800000, in accordance with IEEE754 → the coefficient read is 1.0

User Gas Coefficient Write: UGCW

Name
UGCW

Purpose
Write the user gas coefficient

Data Send (char) : 8
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
User Gas Coefficient	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	0..7	1)

1) See section 'Computation of the digital I/O data'

Data Receive

void

Example

Command Send

Device		Command						Data	CRC16			
'0'	'1'	'-'	'>'	'U'	'G'	'C'	'W'	See below	'b'	'9'	'7'	'b'

- Data is: 0x3f866666, in accordance with IEEE754 → the coefficient written is 1.05

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'U'	'G'	'C'	'W'	'7'	'3'	'9'	'd'

Impedance Switch Read: ISWR

Name
ISWR

Purpose
Read the connection of terminating resistor for RS485 line

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Impedance Switch	Uint8	0x00 (0d0)	0x01 (0d1)	0..1	1)

1) 0x00 : Not connected
   0x01 : Connected

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘I’	‘S’	‘W’	‘R’	‘e’	‘4’	‘1’	‘5’

Command Receive

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘I’	‘S’	‘W’	‘R’	‘0’	‘0’	‘c’	‘f’	‘1’	‘a’

- The impedance switch read is 0x00 (0d0).

Impedance Switch Write: ISWW

Name
ISWW

Purpose
Write the connection of terminating resistor for RS485 line

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Impedance Switch	Uint8	0x00 (0d0)	0x01 (0d1)	0..1	1)

1) 0x00 : Not connected
   0x01 : Connected

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'I'	'S'	'W'	'W'	'0'	'1'	'0'	'e'	'c'	'b'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'I'	'S'	'W'	'W'	'e'	'7'	'd'	'5'

- The impedance switch written is 0x01 (0d1)

Baud Rate Read

This function could be used to read the chipreg's baudrate.

IMI FAS protocol BDRR

Name
BDRR

Purpose
Read the baud rate value

Data Send (char) : 0
Data Receive (char) : 8

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Baud Rate	Uint32	0x00000000 (0d)	0x0001C200 (0d115200)	0..7	1)

1) Available values are: 9600, 14400, 19200, 28800, 38400, 56000, 57600 and 115200

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'B'	'D'	'R'	'R'	'9'	'4'	'a'	'4'

Command Receive

Device		Command						Data	CRC16			
'0'	'1'	'-'	'>'	'B'	'D'	'R'	'R'	See below	'f'	'e'	'3'	'e'

- Data is: 0x0001c200 (0d115200)

Modbus RTU protocol to get the chipreg's baudrate

Fct code	Address	Size	Range
Read Holding register (3)	0x15	16bits	1 ..8

Available values are: 1 for 9600, 2 for 14400, 3 for 19200, 4 for 28800,5 for 38400, 6 for 56000, 7 for 57600 and 8 for 115200

Example

Address = 0xEB, value = 8 (for 115200 bauds)
Request : EB 03 00 15 00 01 83 04
Response : EB 03 02 00 08 A0 55

Baud Rate Write

This function could be used to set the chipreg's baudrate.

IMI FAS protocol BDRW

Name
BDRW

Purpose
Write the baud rate value.

Data Send (char) : 8
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Baud Rate	Uint32	0x00000000 (0d0)	0 x0001C200 (0d115200)	0..7	1)

1) Available values are: 9600, 14400, 19200, 28800, 38400, 56000, 57600 and 115200

Data Receive

void

Example

Command Send

Device		Command						Data	CRC16			
'0'	'1'	'-'	'>'	'B'	'D'	'R'	'W'	See below	'a'	'e'	'0'	'1'

- Data is: 0x0001c200 (0d115200)

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'B'	'D'	'R'	'W'	'9'	'7'	'6'	'4'

Modbus RTU protocol to set the chipreg's baudrate

Fct code	Address	Size	Range
Write Single Holding Register (6)	0x15	16bits	1 ..8

Available values are: 1 for 9600, 2 for 14400, 3 for 19200, 4 for 28800,5 for 38400, 6 for 56000, 7 for 57600 and 8 for 115200

Example
Address = 0xEB, value = 8 (for 115200 bauds)
Request : EB 06 00 15 00 08 8F 02
Response : --

User Pid Parameters Read: UPPR

Name
UPPR

Purpose
Read the user Pid parameters for mass flow control.

Data Send (char) : 0
Data Receive (char) : 24

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Mass Flow User P	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	0..7	1)
Mass Flow User I	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	8..15	1)
Mass Flow User D	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	16..23	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘U’	‘P’	‘P’	‘R’	‘4’	‘4’	‘e’	‘0’

Command Receive

Device		Command						Data	CRC16			
‘0’	‘1’	‘-‘	‘>’	‘U’	‘P’	‘P’	‘R’	See below	‘0’	‘9’	‘6’	‘e’

- Data is: 0x3dcccccd3d75c28f00000000 (that means 0x3dcccccd, 0x3d75c28f and 0x00000000). In accordance with IEEE754 → P = 0.1, I = 0.06, D = 0

User Pid Parameters Write: UPPW

Name
UPPW

Purpose
Write the user Pid parameters for mass flow control.

Data Send (char) : 24
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Mass Flow User P	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	0..7	1)
Mass Flow User I	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	8..15	1)
Mass Flow User D	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	16..23	1)

1) See section ‘Computation of the digital I/O data’

Data Receive

void

Example

Command Send

Device		Command						Data	CRC16			
‘0’	‘1’	‘-‘	‘>’	‘U’	‘P’	‘P’	‘W’	See below	‘1’	‘b’	‘f’	‘b’

- Data is: 0x3de147ae3d4ccccd00000000 (that means 0x3de147ae, 0x3d4ccccd and 0x00000000). In accordance with IEEE754 → P = 0.11, I = 0.05, D = 0

Command Receive

Device	Command	CRC16

‘0’	‘1’	‘-‘	‘>’	‘U’	‘P’	‘P’	‘W’	‘4’	‘7’	‘2’	‘0’

User Unit Mode Read: UUMR

Name
UUMR

Purpose
Read the user unit mode.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
User Unit Mode	Uint8	0x00 (0d0)	0x02 (0d2)	0..1	1)

1) 0x00 : No mode (same as device unit)
   0x01 : Unit in standard mode.
   0x02 : Unit in normal mode.

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'U'	'U'	'M'	'R'	'1'	'5'	'f'	'9'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'U'	'U'	'M'	'R'	'0'	'0'	'8'	'b'	'9'	'7'

- The user unit mode read is 0x00 (0d0).

Modbus RTU protocol to get the device unit

Fct code	Address	Size	Range

Read Holding register (3)	0x0031	16bits	0 (no mode), 1 (liter), 2 (ml)

Example:
Address = 0xEB, value = 1
Request : EB 03 00 31 00 01 C3 0F
Response : EB 03 02 00 01 60 53

User Unit Mode Write: UUMW

Name
UUMW

Purpose
Write the user unit mode.

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
User Unit Mode	Uint8	0x00 (0d0)	0x02 (0d2)	0..1	1)

1) 0x00 : No mode (same as device unit).
   0x01 : Unit in standard mode.
   0x02 : Unit in normal mode.

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'U'	'U'	'M'	'W'	'0'	'2'	'4'	'b'	'0'	'6'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'U'	'U'	'M'	'W'	'1'	'6'	'3'	'9'

- The user unit mode written is 0x02 (0d2)

Modbus RTU protocol to set the MFC's device address

Fct code	Address	Size	Range
Write Single Holding Register (6)	0x0031	16bits	0 (no mode), 1 (liter), 2 (ml)

Example
Address = 0xFF, value = 0x0002
Request : FF 06 00 31 00 02 4C 1A
Response : ---

Modbus RTU protocol to get the device gas

Fct code	Address	Size	Range
Read Holding register (3)	0x0032	16bits	*Range_value

Range_value

Helium (He) : 1
Argon (Ar) : 4
Air : 8
Nitrogen (N2) : 13
Oxygen(O2) : 15
Carbon Dioxide (CO2) : 25

Example:
Address = 0Xff
Request : FF 03 00 32 00 01 30 1B
Response : FF 03 02 00 08 90 56

Multi Gas Factor Read:MGFR

Name
MGFR

Purpose
Read the multi gas factor

Data Send (char) : 0
Data Receive (char) : 8

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Multi Gas Factor	Float32	0x00000000 (0d1.17..E-38)	0xFFFFFFFF (0d3.40..E38)	0..7	1)

1) See section ‘Computation of the digital I/O data’

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘M’	‘G’	‘F’	‘R’	‘8’	‘0’	‘5’	‘8’

Command Receive

Device		Command						Data	CRC16			
‘0’	‘1’	‘-‘	‘>’	‘M’	‘G’	‘F’	‘R’	See below	‘6’	‘8’	‘c’	‘7’

- Data is: 0x3f4ccccd, in accordance with IEEE754 → the coefficient written is 0.8

Multi Gas Selection Read:MGSR

Name
MGSR

Purpose
Read the multi gas selection.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Multi Gas Selection	Uint8	0x00 (0d0)	0xFF (0d255)	0..1	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'M'	'G'	'S'	'R'	'1'	'0'	'5'	'6'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'M'	'G'	'S'	'R'	'0'	'8'	'b'	'e'	'2'	'b'

- The multi gas selection read is 0x08 (0d8).

Modbus RTU protocol to get gas selection

Fct code	Address	Size	Range
Read Holding register (3)	0x33	16bits	*range values

Helium (He)	: 1
Argon (Ar)	: 4
Air	: 8
Nitrogen (N2)	: 13
Oxygen(O2)	: 15
Carbon Dioxide (CO2)	: 25

Example

Request : FF 03 00 33 00 01 61 DB

Multi Gas Selection Write:MGSW

Name
MGSW

Purpose
Write the multi gas selection.

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Multi Gas Selection	Uint8	0x00 (0d0)	0xFF (0d255)	0..1	1)

1) See section 'Computation of the digital I/O data'

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'M'	'G'	'S'	'W'	'0'	'8'	'b'	'f'	'3'	'b'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'M'	'G'	'S'	'W'	'1'	'3'	'9'	'6'

- The multi gas selection written is 0x08 (0d8)

Modbus RTU protocol to set gas selection

Fct code	Address	Size	Range
Read Holding register (3)	0x33	16bits	*range values

Helium (He)	: 1
Argon (Ar)	: 4
Air	: 8
Nitrogen (N2)	: 13
Oxygen(O2)	: 15
Carbon Dioxide (CO2)	: 25

Example

Request : FF 06 00 33 00 08 6D DD

Security Mode Read

This function provides the system's security mode.

IMI FAS protocol STYR

Name
STYR

Purpose
Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Security Mode	Uint8	0x00 (0d0)	0x01 (0d1)	0..1	1) 2)

1) 0x00 : Security deactivated.
   0x01 : Security activated.

2) See Annex for details.

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'S'	'T'	'Y'	'R'	'5'	'd'	'a'	'7'

Command Receive

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'S'	'T'	'Y'	'R'	'0'	'1'	'1'	'd'	'6'	'e'

- The security mode read is 0x01 (0d1).

Modbus RTU protocol to get the security mode

Fct code	Address	Size	Range
Read Holding register (3)	0x1111	16bits	0 ..1

Example
Address = 0xEB, value = 0
Request : EB 03 11 11 00 01 C7 F9
Response : EB 03 02 00 00 A1 93

Security Mode Write

This function could be used to activate(1) or deactivate(0) the security mode.
IMI FAS protocol STYW

Name
STYW

Purpose
Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Security Mode	Uint8	0x00 (0d0)	0x01 (0d1)	0..1	1) 2)

1) 0x00 : Security deactivated.
   0x01 : Security activated.

2) See Annex for details.

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘T’	‘Y’	‘W’	‘0’	‘0’	‘d’	‘c’	‘b’	‘f’

Command Receive

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘S’	‘T’	‘Y’	‘W’	‘5’	‘e’	‘6’	‘7’

- The security mode written is 0x00 (0d0)

Modbus RTU protocol to set the security mode

Fct code	Address	Size	Range
Write single register (6)	0x1111	16bits	0 .. 1

Example
Address = 0xEB, value = 1
Request : EB 06 11 11 00 01 0B F9
Response : EB 03 02 00 00 A1 93

Temperature Compensation Selection Read:TCSR

Name
TCSR

Purpose
Read the temperature compensation selection.

Data Send (char) : 0
Data Receive (char) : 2

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Temper. Comp. Selection	Uint8	0x00 (0d0)	0x01 (0d1)	0..1	1)

1) 0x00 : Temperature Compensation deactivated.
   0x01 : Temperature Compensation activated.

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'T'	'C'	'S'	'R'	'8'	'd'	'1'	'0'

Command Receive

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘T’	‘C’	‘S’	‘R’	‘0’	‘1’	‘7’	‘1’	‘1’	‘8’

- The temperature compensation selection read is 0x01 (0d1).

Temperature Compensation Selection Write: TCSW

Name
TCSW

Purpose
Write the temperature compensation selection.

Data Send (char) : 2
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Temper. Comp. Selection	Uint8	0x00 (0d0)	0x01 (0d1)	0..1	1)

1) 0x00 : Temperature Compensation deactivated.
   0x01 : Temperature Compensation activated.

Data Receive

void

Example

Command Send

Device		Command						Data		CRC16			
'0'	'1'	'-'	'>'	'T'	'C'	'S'	'W'	'0'	'0'	'b'	'0'	'c'	'9'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'T'	'C'	'S'	'W'	'8'	'e'	'd'	'0'

- The temperature compensation selection written is 0x00 (0d0)

Boost Initial Value Read: BIVR

Name
BIVR

Purpose
Read the initial boost value.

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Boost Initial Value	Uint16	0x0000 (0d0)	0x0F9F (0d3999)	0..3	1)

1) See section 'Computation of the digital I/O data'

Example

Command Send

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'B'	'I'	'V'	'R'	'9'	'7'	'3'	'7'

Command Receive

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'B'	'I'	'V'	'R'	'0'	'1'	'f'	'4'	'2'	'0'	'c'	'b'

- The boost initial value read is 0x01f4 (0d500).

Boost Initial Value Write: BIVW

Name
BIVW

Purpose
Write the boost initial value.

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Boost Initial Value	Uint16	0x0000 (0d0000)	0x0F9F (0d3999)	0..3	1)

1) See section ‘Computation of the digital I/O data’

Data Receive

void

Example

Command Send

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘B’	‘I’	‘V’	‘W’	‘0’	‘2’	‘5’	‘8’	‘d’	‘5’	‘c’	‘b’

Command Receive

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘B’	‘I’	‘V’	‘W’	‘9’	‘4’	‘f’	‘7’

- The boost initial value written is 0x0258 (0d600).

Mass Flow Average Read: MFAR

Name
MFAR

Purpose
Read the number of measurements in a moving average window to calculate the mass flow.

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type	Min	Max	Char	Notice
Mass Flow Average	Uint16	0x0000 (0d0)	0x0020 (0d32)	0..3	

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘M’	‘F’	‘A’	‘R’	‘7’	‘0’	‘0’	‘b’

Command Receive

Device		Command						Data				CRC16			
‘0’	‘1’	‘-‘	‘>’	‘M’	‘F’	‘A’	‘R’	‘0’	‘0’	‘2’	‘0’	‘8’	‘4’	‘1’	‘9’

- The mass flow average read is 0x0020 (0d32).

Mass Flow Average Write : MFAW

Name
MFAW

Purpose
Write the number of measurements in a moving average window to calculate the mass flow.

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Mass Flow Average	Uint16	0x0000 (0d0000)	0x0020 (0d32)	0..3	

Data Receive

void

Example

Command Send

Device		Command						Data				CRC16			
'0'	'1'	'-'	'>'	'M'	'F'	'A'	'W'	'0'	'0'	'2'	'0'	'8'	'4'	'd'	'5'

Command Receive

Device		Command						CRC16			
'0'	'1'	'-'	'>'	'M'	'F'	'A'	'W'	'7'	'3'	'c'	'b'

- The mass flow average write is 0x0020 (0d32).

Switch communication mode

The MFC could be switched from Modbus to the IMI FAS communication mode.

Fct code	Address	Size	Range
Write Single Holding Register (6)	0xE110	16bits	0 .. 2

Range:
0 : Nvm Incomplete
1 : Nvm Complete
2 : Nvm Complete Modbus

Read Fw version : FWVR

Name
FWVR

Purpose
Read the fw version from the non volatile memory.

Data Send (char) : 0
Data Receive (char) : 9

Data Send

void

Data Receive

Parameter	Type
Fw version read	char

Example

Command Send

Device		Command						CRC16			
‘0’	‘1’	‘-‘	‘>’	‘F’	‘W’	‘V’	‘R’	‘X’	‘X’	‘X’	‘X’

Command Receive

Device		Command						Data	CRC16			
‘0’	‘1’	‘-‘	‘>’	‘F’	‘W’	‘V’	‘R’	Fw version string	‘f’	‘1’	‘f’	‘4’

Fw version string:”01.06.02A”

Switch communication mode

The MFC could be switched from Modbus to the IMI FAS communication mode.

Fct code	Address	Size	Range
Write Single Holding Register (6)	0xE110	16bits	0 .. 2

Range:
0 : Nvm Incomplete

1 : Nvm Complete
2 : Nvm Complete Modbus

Regulation time period Write : REGW

Since Fw version > 1.06.02

Name
REGW

Purpose
Write the number of ms for the period regulation time. The number of ms should be [5;255]

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Regulation time period in ms	Uint16	0x0005 (0d0005)	0x00FF (0d255)	0..3	

Data Receive
void
Example

Command Send

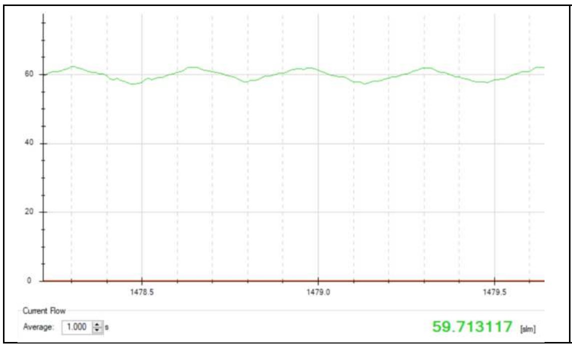
Device		Command						Data				CRC16			
'f'	'f'	'-'	'>'	'R'	'E'	'G'	'W'	'0'	'0'	'0'	'9'	'X'	'X'	'X'	'X'

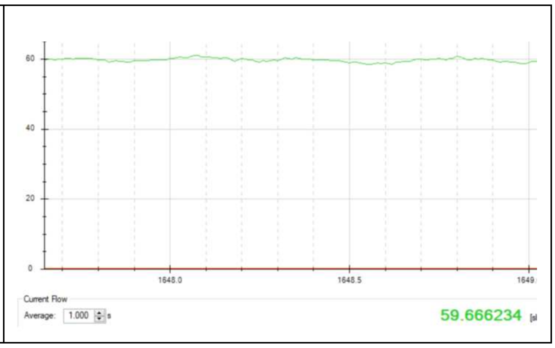
Command Receive

Device		Command						CRC16			
'f'	'f'	'-'	'>'	'R'	'E'	'G'	'W'	'd'	'4'	'9'	'c'

Note: The CMD REGW is recommended for use only in high-flow conditions, specifically at 40 liters per minute or higher. This parameter aids in tuning to prevent flow oscillation during operation. It should be combined with the DADW cmd.

Regulation period = 5ms	Regulation period = 15ms


60
40
20
0
1478.5
1479.0
1479.5
Current Flow
Average: 1.000 s
59.713117 [slm]


60
40
20
0
1648.0
1648.5
1649.
Current Flow
Average: 1.000 s
59.666234

Regulation time period Read : REGR

Since Fw version > 1.06.02

Name
REGR

Purpose
Read the regulation time period from the non volatile memory.

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type
Regulation time period read	char

Example

Command Send

Device		Command						CRC16			
'f'	'f'	'-'	'>'	'R'	'E'	'G'	'R'	'X'	'X'	'X'	'X'

Command Receive

Device		Command						Data				CRC16			
'f'	'f'	'-'	'>'	'R'	'E'	'G'	'R'	'0'	'0'	'0'	'9'	'7'	'5'	'7'	'9'

DP raw data average Write : DPAW

Since Fw version > 1.06.02

Name
DPAW

Purpose
Write the number of items for the sliding average of DP raw. The number should be [1;32]

Data Send (char) : 4
Data Receive (char) : 0

Data Send

Parameter	Type	Min	Max	Char	Notice
Number for the average	Uint16	0x0001 (0d0005)	0x0020 (0d32)	0..3	

Data Receive
void

Example

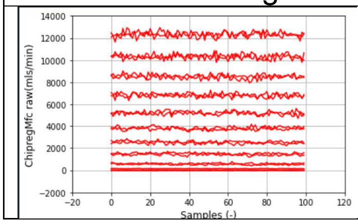
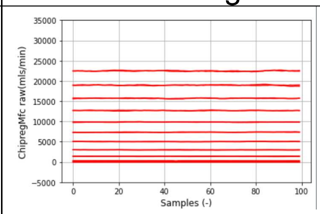
Command Send

Device		Command						Data				CRC16			
‘f’	‘f’	‘-‘	‘>’	‘D’	‘P’	‘A’	‘W’	‘0’	‘0’	‘0’	‘9’	‘X’	‘X’	‘X’	‘X’

Command Receive

Device		Command						CRC16			
‘f’	‘f’	‘-‘	‘>’	‘D’	‘P’	‘A’	‘W’	‘f’	‘8’	‘8’	‘a’

Note: The CMD DPAW is recommended for use only in high-flow conditions, specifically at 40 liters per minute or higher. This parameter aids in tuning to prevent flow oscillation of dp raw data. It should be combined with the REGW cmd.

Without DP average	With DP average
	

DP raw data average Read : DPAR

Since Fw version > 1.06.02

Name
REGR

Purpose
Read the number of items for the sliding average of DP raw.

Data Send (char) : 0
Data Receive (char) : 4

Data Send

void

Data Receive

Parameter	Type
Number for the average read	char

Example

Command Send

Device		Command						CRC16			
'f'	'f'	'-'	'>'	'D'	'P'	'A'	'R'	'X'	'X'	'X'	'X'

Command Receive

Device		Command						Data				CRC16			
'f'	'f'	'-'	'>'	'D'	'P'	'A'	'R'	'0'	'0'	'0'	'9'	'f'	'4'	'b'	'c'

Fw type Read : FWTY

Since Fw version > 1.07.00

Name
FWTY

Purpose
Read FW type for MFC: FAS_MFC.

Data Send (char) : 0
Data Receive (char) : 7

Data Send

void

Data Receive

Parameter	Type
FW type read	char

Example

Command Send

Device		Command						CRC16			
‘f’	‘f’	‘-‘	‘>’	‘F’	‘W’	‘T’	‘Y’	‘X’	‘X’	‘X’	‘X’

Command Receive

Device		Command						Data	CRC16			
‘f’	‘f’	‘-‘	‘>’	‘F’	‘W’	‘T’	‘Y’	‘F’ ’A’ ’S’ ’ ’_ ’ ’M’ ’F’ ’C’	‘4’	‘5’	‘A’	‘A’

Modbus RTU protocol to get parity & stop bit

Fct code	Address	Size	Range
Read Holding register (3)	0x16	16bits	*range values

*range values:

Parity	Stop bit number
PARITY_NONE = 0, PARITY_EVEN = 1, PARITY_ODD = 2,	NBR_STOP_BIT_ONE = 1, NBR_STOP_BIT_DEUX = 2,

Example
Request : FF 03 00 16 00 01 70 10
Response : FF 03 02 01 01 51 C0 //EVEN, 1 STOPBIT

Modbus RTU protocol to set parity & stop bit

Fct code	Address	Size	Range
Write Single Holding Register (6)	0x16	16bits	*range values

*range values:

Parity	Stop bit number
PARITY_NONE = 0, PARITY_EVEN = 1, PARITY_ODD = 2,	NBR_STOP_BIT_ONE = 1, NBR_STOP_BIT_DEUX = 2,

Example Write Odd 1 stop bit
Request : FF 06 00 16 02 01 BD 70 //ODD, 1 STOPBIT

Modbus RTU protocol to get device unit

Fct code	Address	Size	Range
Read Holding register (3)	0x31	16bits	1 = liter, 2 = ml

Example
Request : FF 03 00 31 00 01 C0 1B
Response: FF 03 02 00 01 50 50 //01= Liter

Modbus RTU protocol to set device unit

Fct code	Address	Size	Range
Write Single Holding Register (6)	0x31	16bits	1 = liter, 2 = ml

Example Write ml (2)
Request : FF 06 00 31 00 02 4C 1A

Computation of the Digital I/O data

Mass Flow

There are two types of mass flow data: raw and scaled. The raw data is the raw values coming out from the embedded sensor used during calibration process in production. The scaled data are the raw values after conversion, it represents **the mass flow rate**.

Command	Type of Data
MFSR	Scaled
MFSW	Scaled
RMFR	Raw (only use for production mode)
SMFR	Scaled

$$Mass\ Flow = \frac{Mass\ Flow\ FS \cdot Scaled\ Data}{Digital\ FS}$$

Mass Flow	ls/min standard conditions 20°C, 1.013 bar, calibrated for Air
Mass Flow Full Scale	ls/min standard conditions 20°C, 1.013 bar, calibrated for Air
Scaled Data	command digital value
Digital Full Scale	4095

Example

The device is a 10 ls/min MFC. After sending the SMFR command (Scaled Mass Flow Read) the returned value is 2000. Thus, the mass flow is:

$$Mass\ Flow = \frac{10 \cdot 2000}{4095} = 4.884\ ls/min$$

Valve current

There are two types of valve current data: raw and scaled. The raw data is the raw values coming out from the embedded ADC used during calibration process in production. The scaled data are the raw values after conversion, it represents **the current supplying the valve**.

Command	**Type of Data**
VCSR	Scaled
VCSW	Scaled
RVCR	Raw (only use for production mode)
SVCR	Scaled

$$Valve\ Current = \frac{Valve\ Current\ FS \cdot Scaled\ Data}{Digital\ FS}$$

Valve Current	mA
Valve Current Full Scale	110 mA
Scaled Data	command digital value
Digital Full Scale	4095

Example

After sending the SVCR command (Scaled Valve Current Read) the returned value is 1000. Thus, the valve current is:

$$Valve\ Current = \frac{110 \cdot 1000}{4095} = 26.9\ mA$$

Drive Pwm

The raw data represents the **register value of the PWM duty cycle of the power drive supplying the valve**.

Command	**Type of Data**
DPSR	Raw
DPSW	Raw
RDPR	Raw

$$Duty\ Cycle = \frac{Raw\ Data}{Digital\ FS} \cdot 100$$

Duty Cycle	%
Raw Data	command digital value
Digital Full Scale	4000

Example

After sending the RDPR command (Raw Drive Pwm Read) the returned value is 2500. Thus, the duty cycle is:

$$Duty\ Cyle = \frac{2500}{4000} \cdot 100 = 62.5\%$$

Adc Setpoint

There are two types of ADC setpoint data: raw and scaled. The raw data is the raw values coming out from the embedded ADC used during calibration process in production. The scaled data are the raw values after conversion, it represents **the mass flow setpoint of the analog input**.

Command	Type of Data
RASR	Raw (only use for production mode)
SASR	Scaled

$$Mass\ Flow\ Setpoint = \frac{Mass\ Flow\ FS \cdot Scaled\ Data}{Digital\ FS}$$

Mass Flow Setpoint	ls/min standard conditions 20°C, 1.013 bar, calibrated for Air
Mass Flow Full Scale	ls/min standard conditions 20°C, 1.013 bar, calibrated for Air
Scaled Data	command digital value
Digital Full Scale	4095

Example

The device is a 10 ls/min MFC. After sending the SASR command (Scaled Adc Setpoint Read) the returned value is 2000. Thus, the mass flow setpoint is:

$$Mass\ Flow\ Setpoint = \frac{10 \cdot 2000}{4095} = 4.884\ ls/min$$

Effective Setpoint

The read data gives the last setpoint sent by the user regardless the command ‘CTRW’ and ‘SISW’. Thus, the nature of the setpoint can be:

Mass Flow
Valve Current
Drive Pwm
Mass Flow Analog Input (Adc Setpoint)

To interpret the read data, see the information of the same chapter.

Dac User

There are two types of DAC user data: raw and scaled. The raw data is the raw values written to the embedded DAC used during calibration process in production. The scaled data are the raw values after conversion, it represents **the voltage that the user wants to set at the analog output**.

Command	Type of Data
RDUR	Raw (only use for production mode)
RDUW	Raw (only use for production mode)
SDUR	Scaled
SDUW	Scaled

$$Set\ Analog\ Output = \frac{Set\ Analog\ Output\ FS \cdot Scaled\ Data}{Digital\ FS}$$

Set Analog Output	V
Set Analog Output Full Scale	5 V
Scaled Data	command digital value
Digital Full Scale	4095

Example

After sending the SDUW command (Scaled Dac User Write) with the value of 2000, the set output voltage is:

$$Set\ Analog\ Output = \frac{5 \cdot 2000}{4095} = 2.442\ V$$

Analog Output

There are two types of analog output data: raw and scaled. The raw data is the raw values coming out from the embedded ADC used during calibration process in production. The scaled data are the raw values after conversion, it represents **the approximate voltage at the analog output**.

Command	**Type of Data**
RAOR	Raw (only use for production mode)
SAOR	Scaled

$$Analog\ Output = \frac{Analog\ Output\ FS \cdot Scaled\ Data}{Digital\ FS}$$

Analog Output	V
Analog Output Full Scale	5.1 V
Scaled Data	command digital value
Digital Full Scale	4095

Example

After sending the SAOR command (Scaled Analog Output Read) the returned value is 1500. Thus, the approximate output voltage is:

$$Analog\ Output = \frac{5.1 \cdot 1500}{4095} = 1.868\, V$$

Drive Voltage

There are two types of drive voltage data: raw and scaled. The raw data is the raw values coming out from the embedded ADC used during calibration process in production. The scaled data are the raw values after conversion, it represents **the approximate voltage of the power drive**.

Command	**Type of Data**
RDVR	Raw (only use for production mode)
SDVR	Scaled

$$Drive\ Voltage = \frac{Drive\ Voltage\ FS \cdot Scaled\ Data}{Digital\ FS}$$

Drive Voltage	V
Drive Voltage Full Scale	39.6 V
Scaled Data	command digital value
Digital Full Scale	4095

<u>Example</u>

After sending the SDVR command (Scaled Drive Voltage Read) the returned value is 1768. Thus, the approximate output voltage is:

$$Drive\ Voltage = \frac{39.6 \cdot 1768}{4095} = 17.097\ V$$

Gas Temperature

There are two types of gas temperature data: raw and scaled. The raw data is the raw values coming out from the embedded sensor used during calibration process in production. The scaled data are the raw values after conversion, it represents **the gas temperature**.

Command	**Type of Data**
RGTR	Raw (only use for production mode)
SGTR	Scaled

$$Gas\ Temperature = \frac{Gas\ Temperature\ FS \cdot Scaled\ Data}{Digital\ FS}$$

Gas Temperature	°C
Gas Temperature Full Scale	81.9 °C
Scaled Data	command digital value
Digital Full Scale	4095

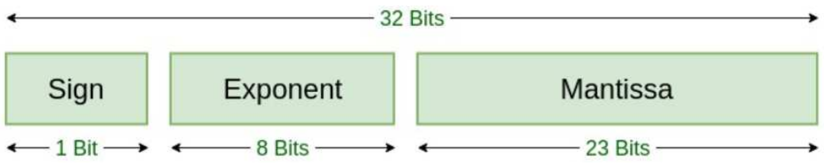
Example

After sending the SGTR command (Scaled Gas Temperature Read) the returned value is 1800. Thus, the gas temperature is:

$$Gas\ Temperature = \frac{81.9 \cdot 1800}{4095} = 30\ °C$$

User Gas Coefficient

The **User Gas Coefficient** allows to apply a correction factor (if needed) to the mass flow output computed by the MFC. By default, the factor is 1.0. The format value is a float32 (single precision) type following the standard IEEE 754. For R/W operation the value must be encoded/decoded on 32 bit for the communication frame. Any programming language allows to perform this common task.



Single Precision IEEE 754 Floating Point Standard

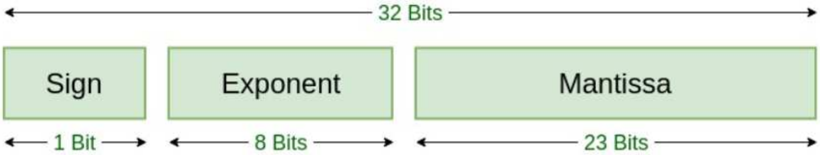
Example

Number to Encode	Single Precision Representation
1.00	0x3f800000
1.05	0x3f866666

Single Precision Representation	Decoded Number
0x3f7d70a4	0.99
0x3f733333	0.95

User Pid Parameters

The **Mass Flow User PID** allow to customize the Pid controller for mass flow control (if needed). By default, PID values are already stored in NVM. For fine tuning, it is recommended to start from these values. The format value is a float32 (single precision) type following the standard IEEE 754. For R/W operation the value must be encoded/decoded on 32 bit for the communication frame. Any programming language allows to perform this common task.



Single Precision IEEE 754 Floating Point Standard

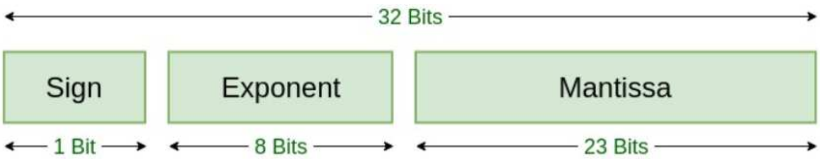
Example

Number to Encode	Single Precision Representation
0.1	0x3dcccccd

Single Precision Representation	Decoded Number
0x3d75c28f	0.06

Multi Gas Factor and Selection

The **Multi Gas Factor** is the factor (or ratio) linking the **Device Gas** and the **Calibration Gas** (see IDER/W command for details). It depends on several parameters such as: density, specific heat and viscosity. This value is set in factory during the calibration process. The format value is a float32 (single precision) type following the standard IEEE 754. For R/W operation the value must be encoded/decoded on 32 bit for the communication frame. Any programming language allows to perform this common task.



Single Precision IEEE 754 Floating Point Standard

The **Multi Gas Selection** allows to work either with the **Device Gas** or the **Calibration Gas**. To select the gas, the right gas code must be sent. By default, the **Device Gas** is selected.

<u>Example</u>

The IDER command of an MFC shows the following information:

Calibration Gas : 8 → Air
Calibration Full Scale : 10
Device Gas : 25 → CO2
Device Full Scale : 4.93
Unit : 1 → l_s/min

And MGFR (multi gas factor read) command shows:

Multi Gas Factor = 0.493

That means is a 4.93 l_s/min CO2 MFC, but calibrated with Air at 10 l_s/min. In fact:

$$Calibration\ Gas\ Full\ Scale = \frac{Device\ Gas\ Full\ Scale}{Multi\ Gas\ Factor} = \frac{4.93}{0.493} = 10\ ls/min$$

If the user wants to switch on **Calibration Gas**, the number 8 (Air) must be sent through the command MGSW and to return to the **Device Gas**, the number 25 (CO2) must be sent.

Boost Initial Value

The Boost Initial Value is the parameter which allows to accelerate the response time in Mass Flow Control Mode, especially when the valve current is near zero (after setting the setpoint to 0 or a power reset). This parameter could be dependent on the inlet pressure and temperature.

As this is already optimized during the calibration process in factory, it is not advisable to change the value (please contact your technical support for further details).

Troubleshooting

LED behaviour

- The LED green blinks about 1 Hz, the system is alive, the firmware is running. The protocol communication is in FAS protocol.
- The LED green blinks about 2 Hz, the system is alive, the firmware is running. The protocol communication is in Modbus RTU protocol.
- The LED red is switched to ON (permanently, green led continues blinking):
  - The Chipreg cannot reach the setpoint due to no input pressure, or valve has problem.
    If the error is due to the no input pressure or the pressure is too low, the red LED should disappear as soon as you provide enough pressure to reach the setpoint.
  - The sensor has a problem (no contact with the PCB board or sensor is defect).
- When the red LED is ON, please use the command HWSR to get more information.
- The LED red toggles at 4Hz when the system is in fatal error. You cannot communicate with the chipreg (since fw 1.07.04).

ERROR codes

Following a master command, if an error occurs during communication, the device sends back an error code, designed by 'ERRN' describing the type of error with the following structure:

Command Receive

Device		Command						Data		CRC16			
A0	A1	'-'	'>'	'E'	'R'	'R'	'N'	D0	D1	R0	R1	R2	R3

Data Receive

Parameter	Type	Min	Max	Char	Notice
Communication Error	Uint8	0x00 (0d0)	0x09 (0d9)	0..1	1)

1) 0x01 : Reserved
   0x02 : Reserved
   0x03 : Error CRC16 (the computation of the CRC16 is incorrect)
   0x04 : Error Integrity (number in hex format has an incorrect character: g, h, i…)
   0x05 : Error Range (the range of a number is out of bounds)
   0x06 : Reserved
   0x07 : Error Password (wrong factory password)
   0x08 : Error Control Disable (operation not possible, because control disabled)

0x09 : Error Control Enable (operation not possible, because control enabled)

Examples

Command Send

Device		Command						Data		CRC16			
‘0’	‘1’	‘-‘	‘>’	‘U’	‘U’	‘M’	‘W’	‘0’	‘3’	‘8’	‘b’	‘c’	‘7’

Error Receive

Device		Command				Data		CRC16			
‘0’	‘1’	‘E’	‘R’	‘R’	‘N’	‘0’	‘5’	‘c’	‘a’	‘2’	‘6’

- Error Range

Back to default Modbus RTU (since fw 1.07.04)

If communication with the chipreg via Modbus RTU is unsuccessful, you have the option to revert to the default Modbus RTU configuration using the Analog Input method.
To initiate this process, set the Analog Input to > 4.5V for a duration of 30 seconds, without applying any input pressure.
Following this action, the system will undergo a restart, and the communication settings will be configured to Modbus RTU at 115200 bauds, Even parity, and with 1 stop bit.

Important
For any master command the device must reply, except when the following errors happen:

- The specified device address is not found on the communication bus.
- The command doesn’t exist for the specified device Address.
- If the communication frame takes more than 1 sec (from the start to the end) to be transmitted.

Scripts

Default State

The Chipreg MFC can be used either through the analog setpoint input (ADC) or the digital setpoint input (RS232/RS485). After production calibration (or System Reset) the device default state is the following:

- DADR : 0xff (Device Address)
- BDRR : 115200 (Baud Rate = 115200)
- CTRR : Mass Flow (Control in Mass Flow)
- CTLR : Fast PID (Controller in Fast PID)
- SISR : Adc (Analog Setpoint)
- AOSR : Mass Flow (Mass Flow on Analog Output)
- UGCR : 1.0 (User Gas Coefficient = 1.0)
- UUMR : No Mode (User Unit Mode not used)
- TCSR : Activated (Temperature Compensation activated)
- MFAR : 0x20 (Mass Flow Average = 32)

Examples

Scenario1: Change the device address

The following example starts from the default state (first use). The device must be the only connected device on the serial line.

Command Send	**Command Receive**
'ff->DADRae19'	'ff->DADRffa621'

Read the device address. As expected, the address is 0xff.

Command Send	**Command Receive**
'ff->DADW01f94f'	'ff->DADWadd9'

Write the new address of the device: 0x01

The next step is to store the new address in the non-volatile memory. Before using the non-volatile memory command, the control CTR must be disabled (security purpose).

Command Send	**Command Receive**
'ff->CTRW000586'	'ff->CTRW7dc7'

The CTR is disabled.

Command Send	**Command Receive**

‘ff->NMWM8d96’	'ff->NMWM8d96'

The new address is stored. The NMWM command performs an automatic system reset, thus it restarts from the default state but with a new device address.

Now, it can be used with other devices on the same serial line with the address: 0x01.

Scenario2: Check the main parameters

The following example starts from the end of Scenario1.

Command Send	Command Receive
’01->CTRRada4’	'01->CTRR02a82e'

Read the CTR. The device is in mass flow control mode.

Command Send	Command Receive
’01->CTLR0dad’	01->CTLR0482a8'

Read the CTL. The device is in fast PID mode.

Command Send	Command Receive
’01->SISRfb31’	'01->SISR01c781'

Read the SIS. The setpoint is in analog input.

Command Send	Command Receive
’01->AOSR82d4’	'01AOSR02b44a'

Read the AOS. The analog output is selected for the mass flow.

Scenario3: Enter in digital mode for mass flow control

The following example starts from the end of Scenario2. The purpose of this scenario is to show how to change the setpoint input selection and the controller.

Command Send	Command Receive
’01->SISW02c7d1’	'01->SISWf8f1'

Write the SIS to switch on digital input (RS232/RS485).

As CTR is already in mass flow mode, no change is needed. The next step is to change the CTL type if needed. This example switches on medium PID.

Command Send	Command Receive
’01->CTLW0341f9’	'01->CTLW0e6d'

Write the CTL in medium PID mode.

After sending the SISW and CTLW commands, this new configuration is immediately activated. To keep this configuration in memory after a system reset, it needs to write it in the non-volatile memory. Before using the non-volatile memory command, the control CTR must be disabled (security purpose).

Command Send	Command Receive
’01->CTRW0068bf’	'01->CTRWae64'

The CTR is disabled.

Command Send	Command Receive
’01->NMWM5e35’	'01->NMWM5e35'

The new configuration is stored. The NMWM command performs an automatic system reset, thus it restarts from the default state but in digital input mode and medium PID mode.

Scenario4: Setpoint and mass flow reading

The following example starts from the end of Scenario3. The purpose of this scenario is to show how to write a setpoint, read the mass flow and temperature. In this example the device range is 10 ls/min.

Command Send	Command Receive
'01->MFSW09c4a73a'	'01->MFSWd3c7'

Write the MFS for the following value: 0x09c4 → 0d2500 → 6.105 ls/min

Command Send	Command Receive
'01->SMFRaa7e'	'01->SMFR09a6834e'

Read the MFR: 0x09a6 → 0d2470 → 6.032 ls/min

Command Send	Command Receive
'01->SGTR0852'	'01->SGTR0526021b'

Read the SGT: 0x0526 → 0d1318 → 26.36°C

Scenario5 : Read and write the user unit mode.

The following example starts from the end of Scenario4. The purpose of this scenario is to show how to read and write the user unit mode. In this example the device unit is l_s/min (liter standard per minute) and the user wants to switch in l_n/min (liter normal per minute).

Command Send	Command Receive
'01->UUMR5ecd'	'01->UUMR008b97'

Read the UUM: 0x00 → No mode

No mode is applied, the unit is l_s/min. The next step is to to switch in l_n/min (liter normal per minute).

Command Send	Command Receive
'01->UUMW024b06'	'01->UUMW1639'

Write the UUM: 0x02 → Normal mode

After sending the UUMW command, the mode is immediately activated. To keep this coefficient in memory after a system reset, it needs to write it in the non-volatile memory. Before using the non-volatile memory command, the control CTR must be disabled (security purpose).

Command Send	Command Receive
'01->CTRW0068bf'	'01->CTRWae64'

The CTR is disabled.

Command Send	Command Receive
’01->NMWM5e35’	'01->NMWM5e35'

The mode is stored. The NMWM command performs an automatic system reset, thus it restarts from the default state but with the new mode.

Scenario 6 : Read and write the user gas coefficient

The following example starts from the end of Scenario5. The purpose of this scenario is to show how to read and write the user coefficient gas.

Command Send	Command Receive
’01->UGCR705d’	'01->UGCR3f800000c2af'

Read the UGC: 0x3f800000 → 1.00

Command Send	Command Receive
’01->UGCW3f8147ae0ce0’	'01->UGCW739d'

Write the UGC for the following value: 1.01 → 0x3f8147ae

After sending the UGCW command, the coefficient is immediately activated. To keep this coefficient in memory after a system reset, it needs to write it in the non-volatile memory. Before using the non-volatile memory command, the control CTR must be disabled (security purpose).

Command Send	Command Receive
’01->CTRW0068bf’	'01->CTRWae64'

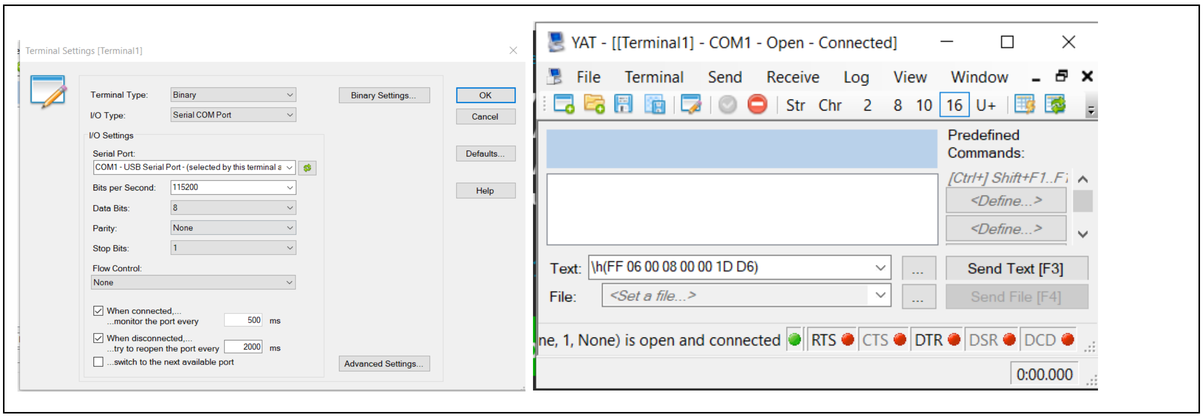
The CTR is disabled.

Command Send	Command Receive
’01->NMWM5e35’	'01->NMWM5e35'

The coefficient is stored. The NMWM command performs an automatic system reset, thus it restarts from the default state but with the new coefficient.

Modbus examples

You can just copy the example and past to the "Yat" application (https://sourceforge.net/projects/y-a-terminal/).
To connect, you have to go to the "Send"/"Open/Start".
To send in hexadecimal format you have to add "\h(.....)" example read an address:
\h(FF 03 00 01 00 01 C0 14).



MFSW (2047) : FF 06 00 08 07 FF 5F A6
MFSW (2047, ad 1) : 01 06 00 08 07 FF 4A 78
MFSW (4095, ad 1) : 01 06 00 08 0F FF 4D B8
MFSW (2047, ad 2) : 02 06 00 08 07 FF 4A 4B
MFSW (4095, ad 2) :02 06 00 08 0F FF 4D 8B
MFSW (2047, ad 3) : 03 06 00 08 07 FF 4B 9A
MFSW (4095, ad 3) :03 06 00 08 0F FF 4C 5A
MFSW (2047, ad 4) :04 06 00 08 07 FF 4A 2D
MFSW (4095, ad 4) :04 06 00 08 0F FF 4D ED

MFSW (0) : FF 06 00 08 00 00 1D D6
MFSR : FF 03 00 08 00 01 10 16
MFSR (ad 1): 01 03 00 08 00 01 05 C8

SMFR : FF 03 11 10 00 01 95 2D
SMFR(ad 1) : 01 03 11 10 00 01 80 F3

SMFR(ad 3) :03 03 11 10 00 01 81 11
SMFR(ad 4) :04 03 11 10 00 01 80 A6

SISR : FF 03 1F 00 00 01 96 00
SISW(1) : FF 06 1F 00 00 01 5A 00
SYRN: FF 05 25 00 00 01 12 D8 // FF,05,25,00,00,01,12,D8 (restart system)
HWSR: FF 03 11 12 00 01 34 ED
DADR : FF 03 00 01 00 01 C0 14
DADW(1) :FF 06 00 01 00 01 0C 14
UUMR : FF 03 00 31 00 01 C0 1B
UUMW(2): FF 06 00 31 00 02 4C 1A
UUMW(1): FF 06 00 31 00 01 0C 1B

STYR: FF 03 11 11 00 01 C4 ED // FF,03,11,11,00,01,C4,ED
STYW (1): FF 06 11 11 00 01 08 ED
STYW (0): FF 06 11 11 00 00 C9 2D
DADW(1) :FF 06 00 01 00 01 0C 14
DADR(1) : 01 03 00 01 00 01 D5 CA
DADW(2): FF 06 00 01 00 02 4C 15
DADR(2) :02 03 00 01 00 01 D5 F9
DADW(3) :FF 06 00 01 00 03 8D D5
DADR(3) :03 03 00 15 00 01 94 2C
DADR(4) :04 03 00 15 00 01 95 9B

DADW(FF) : 01 06 00 01 00 FF 98 4A
DADR(FF) : FF 03 00 01 00 01 C0 14 or par modbus poll FF,03,00,01,00,01,C0,14

BDRR : FF 03 00 15 00 01 80 10
BDRR(1) :01 03 00 15 00 01 95 CE
BDRR(2) : 02 03 00 15 00 01 95 FD
BDRR(2) : 03 03 00 15 00 01 94 2C RESPONSE : 03 03 02 00 01 00 44 (00 01 = 1 9600BAUD)
BDRW(9600, ADRESS 255) : FF 06 00 15 00 01 4C 10
BDRW(115200, ADRESS 255) : FF 06 00 15 00 08 8C 16
BDRW(115200, ADRESS 2) : 02 06 00 15 00 08 99 FB
BDRW(9600, ADRESS 1) :01 06 00 15 00 01 59 CE
BDRW(9600, ADRESS 2) :02 06 00 15 00 01 59 FD

MODW(01 address FF) : FF 06 20 00 00 01 56 14
MODW(01address 2) :02 06 20 00 00 01 43 F9

MODW(01, Ad3) : 03 06 20 00 00 01 42 28
GetDeviceGas : FF 03 00 32 00 01 30 1B
MGSR GetGasSelection : FF 03 00 33 00 01 61 DB
MGSW SetGasSelection : FF 06 00 33 00 08 6D DD

Read parity&stopbit
FF 03 00 16 00 01 70 10 // FF 03 02 01 01 51 C0 //EVEN, 1 STOPBIT
Write Odd 1 stop bit:
FF 06 00 16 02 01 BD 70 // FF,06,00,16,02,01,BD,70

Write Even 1 stop bit:
FF 06 00 16 01 01 BD 80 //FF,06,00,16,01,01,BD,80

Write None 1 stop bit:
FF 06 00 16 00 01 BC 10 // FF,06,00,16,00,01,BC,10

Write None 2 stop bit:
FF 06 00 16 00 02 FC 11 // FF,06,00,16,00,02,FC,11

Read device unit:
FF 03 00 31 00 01 C0 1B
Response: FF 03 02 00 01 50 50 //01= Liter

Annex

User data description

Data application extended description

Offset Register Modbus starts at 0.

name	Type	Notice	Offset Register Modbus

Data system description

Offset Register Modbus starts at 4000.

Name	Type	Notice	Offset Register Modbus
Restart device	bool		4000

Calibration Data Description

Offset Register Modbus starts at 5000.

Parameter	Type	Char	Notice	

floatNvmScaledAdcSetpointSlope floatNvmScaledAdcSetpointOffset floatNvmRawDac1Slope floatNvmRawDac1Offset int16NvmMflsBound[0] int16NvmMflsBound[1] int16NvmMflsBound[2] int16NvmMflsBound[3] int16NvmMflsBound[4] int16NvmMflsBound[5] floatNvmMflsMathFuncCoeff[0][0] floatNvmMflsMathFuncCoeff[0][1] floatNvmMflsMathFuncCoeff[0][2] floatNvmMflsMathFuncCoeff[1][0] floatNvmMflsMathFuncCoeff[1][1] floatNvmMflsMathFuncCoeff[1][2] floatNvmMflsMathFuncCoeff[2][0] floatNvmMflsMathFuncCoeff[2][1] floatNvmMflsMathFuncCoeff[2][2] floatNvmMflsMathFuncCoeff[3][0] floatNvmMflsMathFuncCoeff[3][1] floatNvmMflsMathFuncCoeff[3][2] floatNvmMflsMathFuncCoeff[4][0] floatNvmMflsMathFuncCoeff[4][1] floatNvmMflsMathFuncCoeff[4][2] floatNvmMflsGlobalMathFuncCoeff[0] floatNvmMflsGlobalMathFuncCoeff[1] int16NvmMflsRawTemperatureReferenc e int16NvmMflsScaledTemperatureRefere nce floatNvmMflsTemperatureCoefficient	Uint32 Uint32 Uint32 Uint32 Int16 Int16 Int16 Int16 Int16 Int16 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Uint32 Int16 Int16 Uint32	0..7 8..15 16..23 24..31 32..35 36..39 40..43 44..47 48..51 52..55 56..63 64..71 72..79 80..87 88..95 96..103 104..111 112..119 120..12 7 128..13 5 136..14 3 144..15 1 152..15 9 160..16 7 168..17 5 176..18 3 184..19 1 192..19 5 196..19 9 200..20 7	Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2 Float3 2	

			Float3 2 Float3 2 Float3 2 Float3 2	

Configuration Data Description

Offset Register Modbus starts at 6000.

Parameter	Type	Char	Notice	
floatNvmCtrlValveCurrentSlowPid[0]	Uint32	0..7	Float32	
floatNvmCtrlValveCurrentSlowPid[1]	Uint32	8..15	Float32	
floatNvmCtrlValveCurrentSlowPid[2]	Uint32	16..23	Float32	
floatNvmCtrlValveCurrentMediumPi	Uint32	24..31	Float32	
d[0]	Uint32	32..39	Float32	
floatNvmCtrlValveCurrentMediumPi	Uint32	40..47	Float32	
d[1]	Uint32	48..55	Float32	
floatNvmCtrlValveCurrentMediumPi	Uint32	56..63	Float32	
d[2]	Uint32	64..71	Float32	
floatNvmCtrlValveCurrentFastPid[0]	Uint32	72..79	Float32	
floatNvmCtrlValveCurrentFastPid[1]	Uint32	80..87	Float32	
floatNvmCtrlValveCurrentFastPid[2]	Uint32	88..95	Float32	
floatNvmCtrlMassFlowSlowPid[0]	Uint32	96..103	Float32	
floatNvmCtrlMassFlowSlowPid[1]	Uint32	104..111	Float32	
floatNvmCtrlMassFlowSlowPid[2]	Uint32	112..119	Float32	
floatNvmCtrlMassFlowMediumPid[0	Uint32	120..12	Float32	
]	Uint32	7	Float32	
floatNvmCtrlMassFlowMediumPid[1	Uint32	128..13	Float32	
]	Uint32	5	Float32	
floatNvmCtrlMassFlowMediumPid[2	Uint32	136..14	Float32	
]	Uint32	3	Float32	
floatNvmCtrlMassFlowFastPid[0]	Uint32	144..15	Float32	
floatNvmCtrlMassFlowFastPid[1]	Uint32	1	Float32	
floatNvmCtrlMassFlowFastPid[2]	Uint32	152..15	Float32	
floatNvmCtrlValveCurrentUserPid[0]	Uint8	9		
floatNvmCtrlValveCurrentUserPid[1]	Uint8	160..16		
floatNvmCtrlValveCurrentUserPid[2]	Uint8	7		
floatNvmCtrlMassFlowUserPid[0]	Uint8	168..17		
floatNvmCtrlMassFlowUserPid[1]	Uint32	5	Float32	
floatNvmCtrlMassFlowUserPid[2]	Uint32	176..18	Float32	
uint8NvmSetpointInputSelection	Uint32	3	Float32	
uint8NvmControlType	Uint32	184..19	Float32	
uint8NvmControllerType	Uint32	1	Float32	
uint8NvmAnalogOutputDac1Selecti	Uint32	192..19	Float32	
on	Unint8	3		
floatNvmMflsGasCoefficient[0]	Int16	194..19		
floatNvmMflsGasCoefficient[1]	Int16	5		
floatNvmMflsGasCoefficient[2]	Int16			

floatNvmMflsGasCoefficient[3] floatNvmMflsGasCoefficient[4] floatNvmMflsUserGasCoefficient uint8NvmMflsUserGasUnit int16NvmCtrlBoostParameter[0] int16NvmCtrlBoostParameter[1] int16NvmCtrlBoostParameter[2] int16NvmCtrlBoostParameter[3] Regulation period time in ms DP raw data sliding average int16NvmCtrlBoostParameter[6] int16NvmCtrlBoostParameter[7] int16NvmCtrlBoostParameter[8] int16NvmCtrlBoostParameter[9] int16NvmCtrlBoostParameter[10] Scale sliding average uint8NvmDeviceAddress uint8NvmRs485Impedance uint32NvmBaudRate	Int16 Int16 Int16 Int16 Int16 Int16 Int16 Int16 Int16 Uint8 Uint8 Uint32	196..197 198..199 200..207 208..215 216..223 224..231 232..239 240..247 248..249 250..253 254..257 258..261 262..265 266..269 270..273 274..277 278..281 282..285 286..289 290..293 294..297	 [5;255] [1;32]	 REGW/R DPAW/R

		298..299 300..301 302..309		

Identification Data Description

Offset Register Modbus starts at 7000.

Parameter	Type	Min	Max	Char	Notice	Offset Register Modbus
Part Number	Char			0..12	13	7000
Suffix	Char			13..20	8	7008
Description	Char			21..52	32	7012
Serial Number	Char			53..74	22	7030
SW Version	Char			75..83	9	7042
HW Version	Char			84..92	9	7048
Calibration Date	Char			93..106	1) 14	7054
Calibration Gas	Uint8	0x00 (0d0)	0xFF (0d255)	107..108	2)	7062
Calibration Full Scale Integer Part	Uint16	0x0000 (0d0)	0xFFFF (0d65535)	109..112	3)	7063
Calibration Full Scale Decimal Part	Uint16	0x0000 (0d0)	0xFFFF (0d65535)	113..116	3)	
Device Gas	Uint8	0x00 (0d0)	0xFF (0d255)	117..118	1)	7065
Device Full Scale Integer Part	Uint16	0x0000 (0d0)	0xFFFF (0d65535)	119..122	3)	7066
Device Full Scale Decimal Part	Uint16	0x0000 (0d0)	0xFFFF (0d65535)	123..126	3)	
Device Unit	Unit8	0x00	0xFF	127..128	4)	7068

		(0d0)	(0d255)			
Pressure Reference	Unit16	0x0000 (0d0)	0xFFFF (0d65535)	129..132	mbar	7069
Temperature Reference	Uint16	0x0000 (0d0)	0xFFFF (0d65535)	133..136	m°C	7070
Calibration Pressure	Unit16	0x0000 (0d0)	0xFFFF (0d65535)	137..140	mbar	7071
Calibration Temperature	Unit16	0x0000 (0d0)	0xFFFF (0d65535)	141..144	m°C	7072
Full Scale Accuracy	Unit16	0x0000 (0d0)	0xFFFF (0d65535)	145..148	m%	7073
Reading Accuracy	Unit16	0x0000 (0d0)	0xFFFF (0d65535)	149..152	m%	7074

1)

		Example
Format	: YYYYMMDDHHMMSS	20190221153623
Year	: YYYY	2019
Month	: MM	02
Day	: DD	21
Hour	: HH	15
Minute	: MM	36
Second	: SS	23

2)

Format : Uint8

Helium (He) : 1
Argon (Ar) : 4
Air : 8
Nitrogen (N2) : 13
Oxygen(O2) : 15
Carbon Dioxide (CO2) : 25

According to the Semi E52-0703 standard

3)

Example

Format : Uint16, Uint16 8, 500

Full Scale = integer part + decimal part/1000 = 8 + 0.5 = 8.5

4)

Format : Uint8

liter standard per minute	**l_s/min**	: 1	(1013 mbar, 20°C)
milliliter standard per minute	**ml_s/min**	: 2	(1013 mbar, 20°C)
liter normal per minute	**l_n/min**	: 3	(1013 mbar, 0°C)
millilitre normal per minute	**ml_n/min**	: 4	(1013 mbar, 0°C)

Device Address

The default address is 0xFF. If the user changes the device address (from 0x00 to 0xFE), 0xFF can still be used as rescue address. It is important to underline that the default address can't be used when several devices are connected on the same serial line (RS485).

To change the default address, the user must connect individually each device on the communication bus and follow the scenario in 'Scripts' chapter.

Security Mode

If the MFC is momentarily deprived of its pressurized gas source (following a breakdown for example), the maximum power will be applied at the valve to reach (but in vain) the current flow setpoint. To avoid the valve overheating, the MFC shuts it down, the red led is activated and the HWSR command will report a control saturation error.
Once the issue solved (gas source), to return in the normal flow control process, the user can set once the setpoint at 0 or proceed a power reset.

By default, this security mode is activated, but the user can easily deactivate it using the STYW command. In this case, the red led and the HWSR command will be in error mode but leaving the valve full powered ready to return in the normal flow control process.

Important: the valve can be full powered for a limited time only. Please contact your technical support for further details.