Software Description

8775 Gas Density Sensor

879x Hybrid Gas Density Monitor

Trafag sensors & controls
Switzerland

Industriestrasse 11
8608 Bubikon

+41 44 922 32 32
www.trafag.com

Content

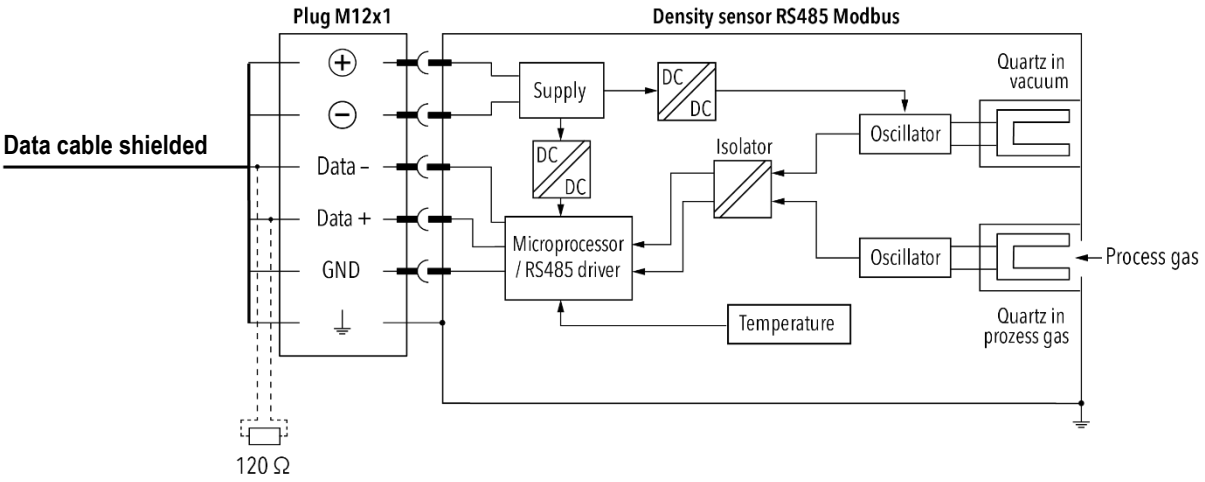
1 Introduction

This instruction manual covers the supported Modbus/RTU functionality and device-specific features of Trafag Gas Density Sensor 8775 and the Trafag Hybrid Gas Density Monitor 879x (8791, 8792, 8793).
This software instruction manual is dedicated to professionals with Modbus knowhow.
The installation instruction for both the Hybrid Density Monitor and the Density Sensor are available online:

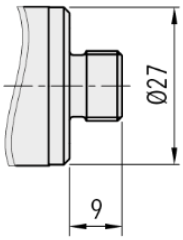
8775 Gas Density Sensor: www.trafag.com/H73519
879x Hybrid Gas Density Monitor: www.trafag.com/H73517

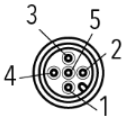
2 Wiring

2.1 Wiring of 8775 Gas Density Sensor



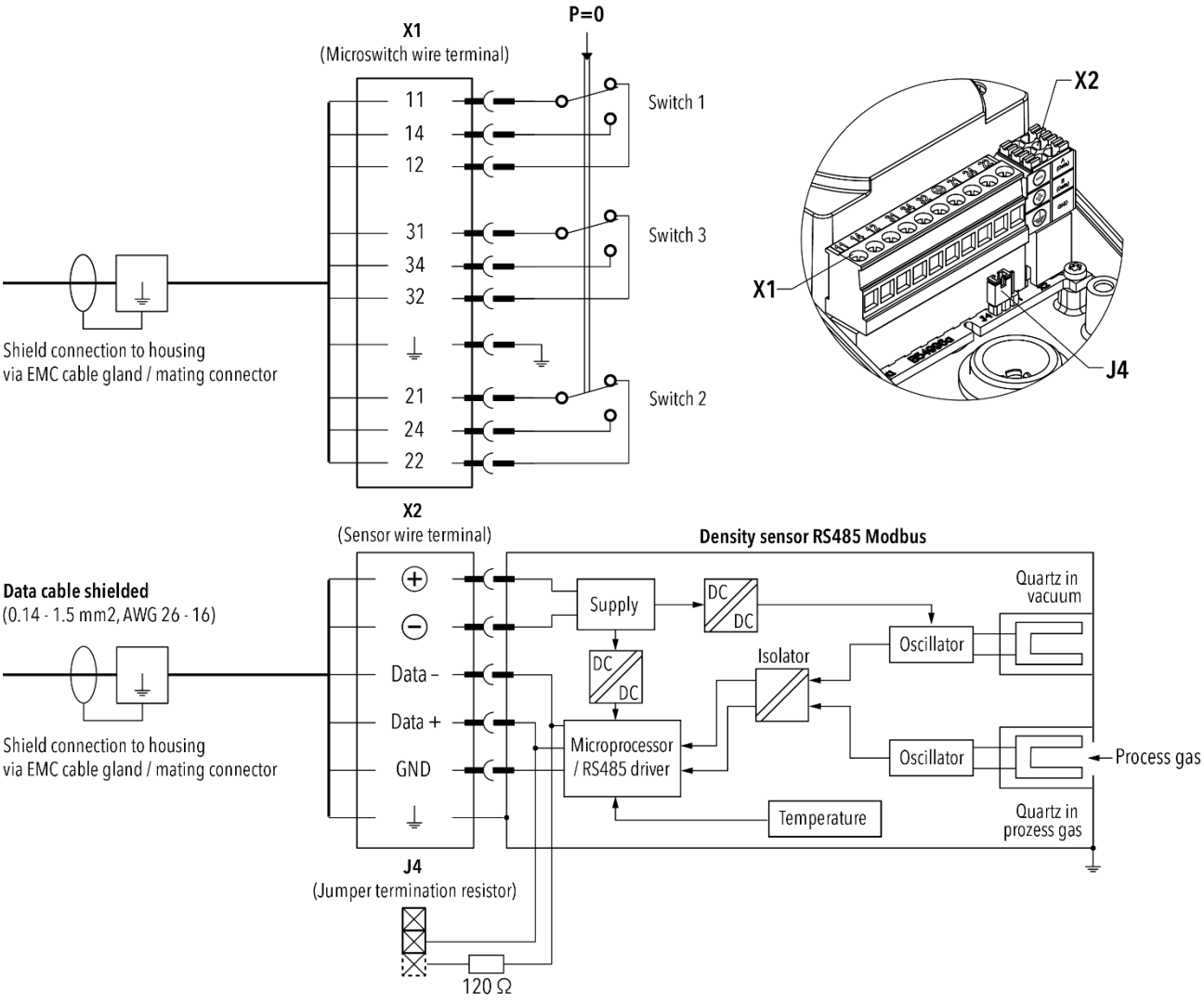
Electrical connector M12x1, 5-pole





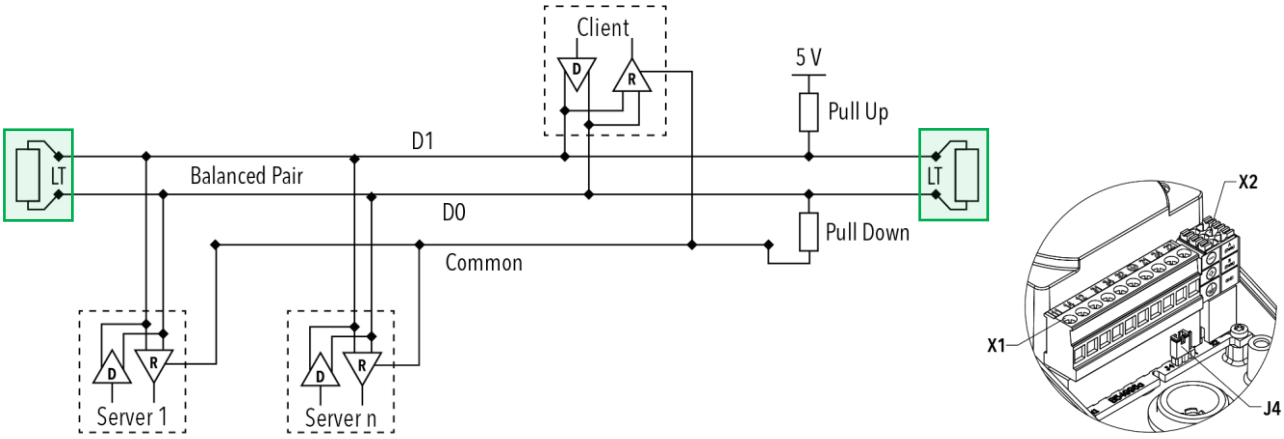
Pin 1: ⊕
Pin 2: ⊖
Pin 3: Data -
Pin 4: Data +
Pin 5: GND
PE: Shield on cable

2.2 Wiring of 879x Hybrid Gas Density Monitor



3 Modbus Topology

3.1 Termination Resistor

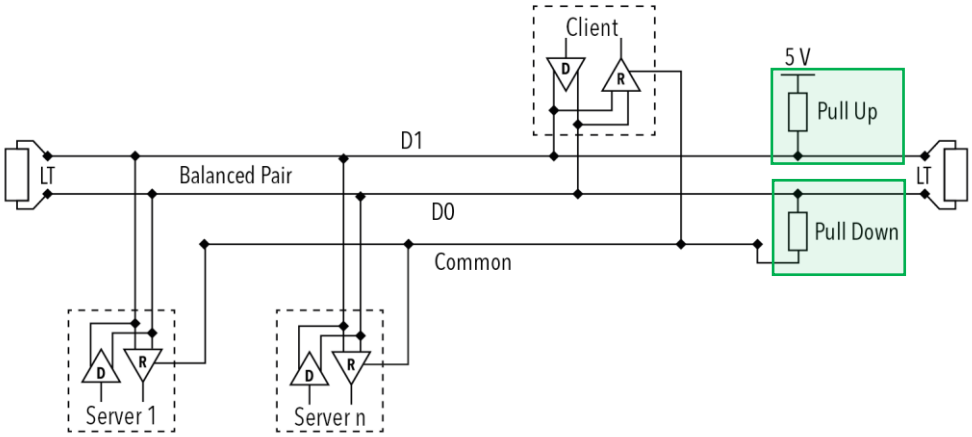


Recommended bus configuration according to the Modbus specification Chapter 3.3.2. [1]

Termination resistor: At the first and last participant of the bus, a termination resistor (120 Ω) needs to be placed. This is normally the client and the last server device. For the hybrid devices of Trafag, the resistor can be enabled by bridging Jumper (J4).

Stub length: The Modbus specification declares that a stub may not be longer than 20m. We recommend keeping the stub as short as possible, i.e. < 1m.

3.2 Line Polarisation (Pull-Up/Down Resistors)

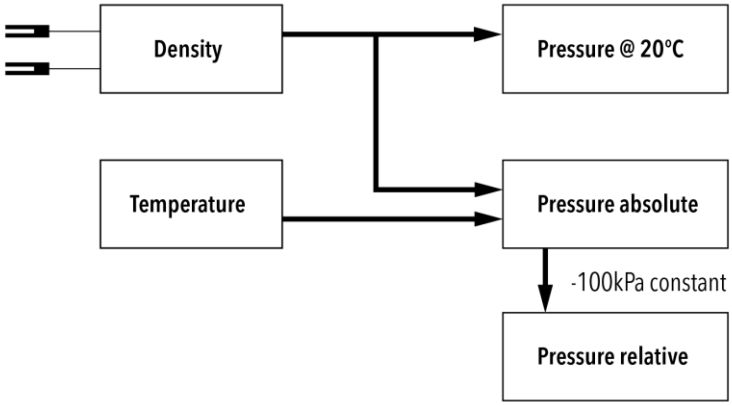


Recommended bus configuration according to the Modbus specification Chapter 3.3.2. [1]

We strongly recommend implementing pull-up and pull-down resistors (line polarisation). With these resistors, a proper idle level even in harsh environments is guaranteed. The recommended value of these resistors is 560 Ω (0.5 W).
In the idle state, the voltage level of Data- (A or D0) is lower than Data+ (B or D1).

4 Device Description

4.1 Signal generation



4.2 Signal description

4.2.1 Density

Density is measured by a pair of quartz oscillators, where one acts in contact with the gas, and the other as a reference in vacuum. The two quartz crystals are counted independently, and their frequency difference is the measurement with which the density is calculated.

The sensor provides three different timescales for the measurements:

- Fast: The fast registers measure the signal over 80ms
- Full: The full registers measure the signal over 1.3s
- Auto (between Fast and Full): The auto registers have a dynamic adaptation of the measurement time depending on the change in signal. When the signal is rapidly changing, it switches to fast measurement, and when the signal is stable, it shifts back to full measurement. The current status of the auto registers is displayed in the “Measurement Status Flags” register 104, bit 2.

4.2.2 Temperature

Ambient temperature is measured to calculate absolute and relative pressure. The temperature sensor is located in the device and mathematically compensated for the difference to the gas temperature.

4.2.3 Absolute Pressure @20°C of the specified gas

This value is calculated based on the density and the gas type, using the equation of state of the respective gas (mixture) for 20°C reference temperature.

4.2.4 Absolute Pressure

This value is calculated based on the density and the gas type, using the equation of state of the respective gas (mixture) for the measured temperature.

4.2.5 Relative Pressure

This value is calculated as absolute pressure minus 100kPa. As such, it is not a precise measurement and should only be used for rough calculations. Only use the absolute pressure and external relative pressure measurement to calculate the precise relative pressure.

4.3 Configuration

Holding registers are provided to configure the sensor. These configurations are saved to EEPROM.

NOTE:
Excessive writing of holding registers could potentially damage the memory and may affect the functionality of the sensor.

5 Modbus

5.1 General

5.1.1 Frame description

Server Adress	Function Code	Data	CRC
1 Byte	1 Byte	0 ... 252 Bytes	2 Bytes

5.1.2 Modbus registers

Supported function codes	Modbus registers
FC 04	Read Input Registers
FC 06	Write Single Register
FC 16	Write Multiple Registers
FC 03	Read Holding Registers
FC 02	Read Discrete Inputs

5.2 Modbus Configuration Legacy Sensor

Input registers hold the sensor data. FC4 is the function code to request these values.

5.2.1 Register Map Input Register *Legacy*

Variable	Addr. #	Scale	Unit	Data range	Datatype
Gas density [1]	0	Value * 0.01	kg/m³	0 … max.	U16
Gas pressure (abs, @20°C) [1]	1	Value * 0.1	kPa	0 … max.	U16
Gas temperature	2	Value * 0.1	K	min … max.	U16
Gas pressure (abs, var °C) [1]	3	Value * 0.1	kPa	0 … max.	U16
Server ID	4	-		1 … 247	U16
Serial Number	5 + 6	-			U32 [2]
FW Release Version	7	Value * 0.01			U16
Quartz frequency difference	8	1000 (fix) [3]		1000 (fix)	U16

[1] Change to 0xFFFF if value range is exceeded
[2] Big-endian
[3] Fixed dummy value of 1000 for firmware versions V2.0 and higher.
Value * 0.01 with data range 0 … max. for firmware versions lower than V2.0

Examples

Client Request "Read Density (FC04)"

ID	FC	Start Addr hi	Start Addr lo	Quantity hi	Quantity lo	CRC hi	CRC lo
0x05	0x04	0x00	0x00	0x00	0x01	0x30	0x4E

Server Response

ID	FC	Byte Count	Input Register [0] hi	Input Register [0] lo	CRC hi	CRC lo
0x05	0x04	0x02	0x00	0x89	0x89	0x56
			137 Dec → 1.37 kg/m³ density			

Client Request "Read Serial Number (FC04)"

ID	FC	Start Addr hi	Start Addr lo	Quantity hi	Quantity lo	CRC hi	CRC lo
0x05	0x04	0x00	0x05	0x00	0x02	0x60	0x4E

Server Response

ID	FC	Byte Count	Input Register [0] hi	Input Register [1] lo	Input Register [0] hi	Input Register [0] lo	CRC hi	CRC lo
0x05	0x04	0x02	0x07	0x5B	0xCD	0x15	0xD3	0xBC
			$0x075BCD15_{hex}$ → 123456789_{Dez} Serial Number 123456-789					

5.2.2 Register Map Holding Register *Legacy*

Variable	Addr #	Data range	Datatype
Server ID	2	1 … 247	U16
Baudrate	3 + 4	1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600	U32 [1)]
Parity	5	0 = none parity (2 stop bits) 1 = odd parity (1 stop bit) 2 = even parity (1 stop bit)	U16
Mode	6	non-functional	U16
Change to bootloader	7	1 = Enter bootloader mode for 10s	U16
Write permissions (read only)	8	bit # / Server ID / Baudrate / Parity 0 → write permission 1 → no write permission	U16

[1)] Big-endian

Examples

Client Request "Read Server ID (FC03)"

ID	FC	Start Addr hi	Start Addr lo	Quantity hi	Quantity lo	CRC hi	CRC lo
0x05	0x03	0x00	0x02	0x00	0x01	0x24	0x4E

Server Response

ID	FC	Byte Count	Input Register [0] hi	Input Register [0] lo	CRC hi	CRC lo
0x05	0x03	0x02	0x00	0x05	0x89	0x87

Client Request "Write Server ID" (FC06)

ID	FC	Start Addr hi	Start Addr lo	Value hi	Value lo	CRC hi	CRC lo
0x05	0x06	0x00	0x02	0x00	0x01	0xE8	0x4E

Server Response

ID	FC	Start Addr hi	Start Addr lo	Value hi	Value lo	CRC hi	CRC lo
0x05	0x06	0x00	0x02	0x00	0x01	0xE8	0x4E

After the response, the Server ID is changed.

5.2.3 Discrete Registers

Variable	Bit	Description
Sensor Status	0	0 = sensor is working properly 1 = sensor error

Examples

Client Request "Read Sensor Status" (FC02)

ID	FC	Start Addr hi	Start Addr lo	Quantity hi	Quantity lo	CRC hi	CRC lo
0x05	0x02	0x00	0x00	0x00	0x01	0xB8	0x4E

Server Response

ID	FC	Byte Count	Value hi	CRC hi	CRC lo
0x05	0x02	0x01	0x00	0xA0	0xB8

5.3 Modbus configuration Standard sensor

5.3.1 Register Map Input Register *Standard*

Integration	Variable	Addr	Data range	Unit	Datatype	special values
Auto	Gas density	0	0 … max.	kg / m³	float	NaN = Error
	Gas pressure (@20°C)	2	0 … max.	kPa @ 20°C	float	NaN = Error
	Gas pressure (abs)	4	0 … max.	kPa abs	float	NaN = Error
	Gas pressure (rel)	6	-100 … max.	kPa rel	float	NaN = Error
	Gas temperature	8	min … max.	K	float	NaN = Error
Precise	Gas density	20	0 … max.	kg / m³	float	NaN = Error
	Gas pressure (@20°C)	22	0 … max.	kPa @ 20°C	float	NaN = Error
	Gas pressure (abs)	24	0 … max.	kPa abs	float	NaN = Error
	Gas pressure (rel)	26	-100 … max.	kPa rel	float	NaN = Error
	Gas temperature	28	min … max.	K	float	NaN = Error
Fast	Gas density	40	0 … max.	kg / m³	float	NaN = Error
	Gas pressure (@20°C)	42	0 … max.	kPa @ 20°C	float	NaN = Error
	Gas pressure (abs)	44	0 … max.	kPa abs	float	NaN = Error
	Gas pressure (rel)	46	-100 … max.	kPa rel	float	NaN = Error
	Gas temperature	48	min … max.	K	float	NaN = Error
	Hourmeter	100	0 … max.	h	U32	

Variable	Addr	Data range / Example	Datatype	special values
Common Device Status Flags	102	**Bit** **Description** 0 Error module initialization 1 Error hardware incompatibility 2 Error filesystem mounting 3 Error BootCfg file loading 4 Error CommunicationCfg file loading 5 Error ApplicationCfg file Loading 6 Error unexpected device reset 7 Error history file saving 8 Error history file loading 9 Error BootCfg fileSaving 10 Error CommunicationCfg file saving 11 Error ApplicationCfg file saving 12 Error quartz oscillator	U32	If larger than 0 = Error
Measuring Status Flags	104	**Bit** **Description** 0 Error gas frequency signal 1 Error vac frequency signal 2 Auto Register: Density signal with lower precision 3 Warning pressure overflow 4 Error density overflow 5 Error density underflow	U32	If larger than 0 = Warning/Error
Serial Number	200		U64	
Firmware Release	204	201	U16	
Full Typecode	300	8799.XX.XXXX	Char[64]	
Hardware Release	400	M02B44190	Char[10]	
Full Firmware Release	500	0M012-02.01-STD-4cc7ec3cf01-2025.06.02-13:38	Char[64]	

5.3.2 Register Map Holding Register *Standard*

Variable	Addr	Data Range	Datatype	Additional Info
Server ID	0	1…247	U16	
Baudrate	1	only specific values allowed	U32	1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600, 115200
Parity	3	only specific values allowed	U16	0 = none parity (2 stop bits) 1 = odd parity (1 stop bit) 2 = even parity (1 stop bit)
Write permissions	4		U16	bit # | 0 | Server ID 1 | Baudrate 2 | Parity 0 → write permission 1 → no write permission

5.4 Change Server ID by broadcast

5.4.1 Write with single register FC06 (firmware versions lower than V2.0)

Write all registers separate with FC06. For the serial number, only 32-bit datatype is supported.

Register 0	Register 1	Register 2
serial high	serial low	new server ID

Examples

Serial number e.g. 123456-789 → 123456789_{Dez} → $075BCD15_{hex}$
Serial high = **075B**$CD15_{hex}$
Serial low = $075B$**CD15**$_{hex}$
New server ID = 4

Client Request (FC06)

ID	FC	Start Addr hi	Start Addr lo	Value hi	Value lo	CRC hi	CRC lo
0x00	0x06	0x00	0x00	0x07	0x5B	0xCB	0xD0
0x00	0x06	0x00	0x01	0xCD	0x15	0x4C	0x84
0x00	0x06	0x00	0x02	0x00	0x04	0x28	0x18

5.4.2 Write with multiple register FC16 (firmware versions V2.0 and higher)

Write all registers with FC16, order for 32 bit

Register 0	Register 1	Register 2
serial high	serial low	new server ID

Write all registers with FC16, order for 64 bit

Register 0	Register 1	Register 2	Register 3	Register 4
serial msb	...	...	serial lsb	new server ID

Examples

Client Request "Change ID by broadcast 32 bit" (FC16)

Calculation with the same values as in Chapter 6.1.1

ID	FC	Addr hi	Addr lo	Quan-tity hi	Quan-tity lo	Byte Count	Value [0] hi	Value [0] lo	Value [1] hi	Value [1] lo	Server ID hi	Server ID low	CRC hi	CRC lo
0x00	0x10	0x00	0x00	0x00	0x03	0x06	0x07	0x5B	0xCD	0x15	0x00	0x04	0xAE	0xD0

Client Request "Change ID by broadcast 64 bit" (FC16)

Client Request: 00 10 00 00 00 05 0A 00 00 00 00 07 5B CD 15 00 04 1A B9

5.5 Exceptions

	Exceptional Responses
01	**Illegal function** The function code received in the query is not an allowable action for the server. This may be because the function code is only applicable to newer devices and was not implemented in the unit selected. It could also indicate that the server is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	**Illegal data address** The data address received in the query is not an allowable address for the server. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed. A request with offset 96 and length 5 will generate exception 02.
03	**Illegal data value** A value contained in the query data field is not an allowable value for the server. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04	**Server device failure** An unrecoverable error occurred while the server was attempting to perform the requested action.
06	**Server device busy** Specialised use in conjunction with programming commands. The server is engaged in processing a long–duration program command. The client should retransmit the message later when the server is free.

Examples

The client sends a request to a server device with FC04 for the data address 0x09 (only 0x00 … 0x08 is supported).

ID	FC	Start Addr hi	Start Addr lo	Quantity hi	Quantity lo	CRC hi	CRC lo
0x05	0x04	0x00	0x09	0x00	0x01	0xE0	0x4C

In an exception response, the server sets the most significant bit (MSB) of the function code to 1. The function code value in an exception is exactly 80 hexadecimal higher than in a normal response.

Server Response

ID	FC	Exception Code	CRC hi	CRC lo
0x05	0x84 "Exception response for FC04" (MSB set)	0x02 "Illegal data address"	0x89	0x56

6 Release notes

Release	Date	Comments
0M012-02.01-STD / LGY	25.06.2025	First Firmware release in production

7 References

[1] Modbus Organization, «MODBUS over Serial Line Specification and Implementation Guide V1.02,» 2006. [Online]. Available: https://www.modbus.org/.

[2] Modbus Organization, «MODBUS Application Protocol Specification V1.1b3,» 2012. [Online]. Available: https://www.modbus.org/.