

# MODBUS-RTU Communication Protocol

## Gas Meter Integration Guide | RS485 Serial Communications

### TECHNICAL PRODUCT INFORMATION | GasIQ Meter

This document describes the MODBUS-RTU communication information supplied for the gas meter. The master polls uniquely addressed slave devices over a half-duplex serial bus.

### Serial transmission format

|                      |                              |
|----------------------|------------------------------|
| Mode                 | Asynchronous MODBUS-RTU      |
| Start bits           | 1                            |
| Data bits            | 8 (LSB first)                |
| Parity               | None                         |
| Stop bits            | 1                            |
| RTU frame separation | At least 3.5 character times |
| Slave address range  | 1–247                        |

### RTU data frame

|               |                 |
|---------------|-----------------|
| Address code  | 1 byte          |
| Function code | 1 byte          |
| Data field    | Variable length |
| CRC16 check   | 2 bytes         |

### Supported function codes (as documented)

|           |                                         |
|-----------|-----------------------------------------|
| 03H / 04H | Read data registers                     |
| 10H       | Preset multiple registers               |
| 05H       | Valve open / close, start address 0001H |
| 01H       | Read valve status, start address 0001H  |

### Example: Read total gas consumption

Example for slave 01H, starting register 0000H, reading two registers (four bytes).

|                      |                                                                      |
|----------------------|----------------------------------------------------------------------|
| Master request       | 01 03 00 00 00 02 C4 0B                                              |
| Sample response      | 01 03 04 00 12 D6 87 44 34                                           |
| Value interpretation | 0x0012D687 = 1,234,567; scaled $\times 0.01 = 12,345.67 \text{ m}^3$ |

### Example: Write registers (10H)

Function 10H is described for writing communications address and baud rate parameters. The source indicates a maximum of 16 data points (32 bytes) per write. Example register 0015H is used in the source; its map also lists both address and baud rate at 0015H, which requires supplier clarification before configuration.

## Valve commands

|                           |                                        |
|---------------------------|----------------------------------------|
| Open valve request        | 01 05 00 01 00 FF DC 4A                |
| Close valve request       | 01 05 00 01 00 00 9C 0A                |
| Read valve status request | 01 01 00 01 00 01 AC 0A                |
| Valve status values       | FF = open; 00 = closed (as documented) |

For function 05H, the source describes the slave reply as an echo of the command frame. Confirm valve functionality on the actual delivered device.

## Register address map

|                          |                                                              |
|--------------------------|--------------------------------------------------------------|
| 0000H-0001H              | Total gas consumption   read-only   4 bytes                  |
| 0015H                    | Communication address   read/write   char                    |
| 0015H (source duplicate) | Communication baud rate   read/write   char — verify address |

## CRC calculation

Initialize a 16-bit CRC register to FFFFH. XOR each incoming byte into the low byte, then shift right eight times, XORing with polynomial A001H whenever the shifted-out least-significant bit is 1. Repeat for each frame byte. Append the calculated two-byte CRC in MODBUS wire order.

## Integration cautions

- Do not write to undocumented or read-only registers.
- Confirm baud rate, register scaling, byte order, valve support and addressing on the delivered unit.
- The source protocol contains an overlapping 0015H register assignment; do not assume this is correct without testing.

## Communication topology

