RS485 / MODBUS COMMUNICATIONS

CoolPro

Modbus RTU Communications Protocol

Doc · CP-MODBUS-RTU

Communication parameters, frame timing, RJ45 pinout and bus topology, supported function codes, the full register map, data encoding with worked CRC, and example transactions for integrating the CoolPro into a Modbus RTU network over RS485.

STANDARD	Modbus RTU
PHYSICAL	RS485 · RJ45
DEFAULT BAUD	9600 bps
FRAME	8·N·1
DEFAULT ADDRESS	40
CRC	CRC-16 (LSB first)

DOCUMENT	REVISION	EFFECTIVE	PRODUCT
CP-MODBUS-RTU	1	06 / 2026	CoolPro
AUDIENCE	**APPROVING OFFICER**	**FORMAT**	**LANGUAGE**
Integrators	Matthew Wynne	A4 · 8 pages	English (en-AU)

CONTENTS

CHAPTER 01

Bus, connection & settings

The CoolPro provides access to real-time measurements and device information via the standard **Modbus RTU** protocol over RS485.

Communication parameters

PARAMETER	VALUE
Baud rate	9600 (default); configurable: 9600 / 19200 / 38400 / 57600 / 115200
Data bits	8
Stop bits	1
Parity	None
Device address	40 (default); configurable 1–247
CRC	Modbus CRC-16 (transmitted LSB first)

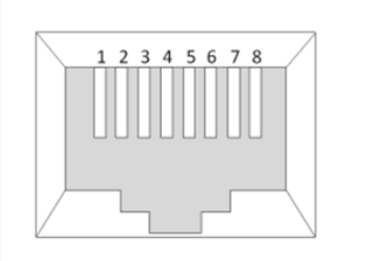
Frame timing

Modbus RTU frames are delimited by idle time on the bus: a frame is treated as complete once the bus has been silent for the inter-frame gap period.

The CoolPro uses a fixed inter-frame gap of approximately **8 ms**, independent of the configured baud rate. This is comfortably wider than the 3.5-character silent interval required by the Modbus specification at every supported baud rate, so a standard-compliant master will interoperate without adjustment.

Physical connection

The CoolPro uses an 8-pin RJ45 connector for its RS485 interface.



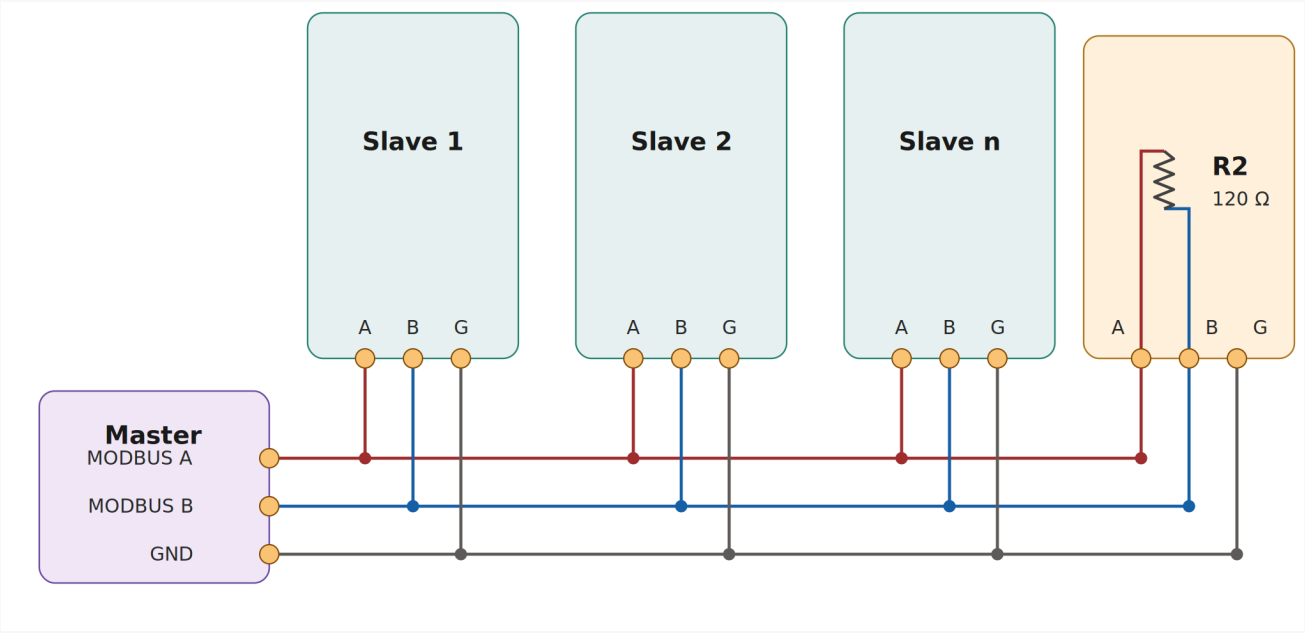
RJ45 connector viewed from the front — contacts numbered 1 to 8, left to right.

PIN	SIGNAL
3	RS485-A (non-inverting)
6	RS485-B (inverting)
8	GND

All other pins are unused.

Network topology

The Modbus network is a linear bus, with short stubs branching from ‘T’ connectors on the main backbone to each device. The data lines must be terminated at each end of the bus with a **120 Ω** resistor between the RS485-A and RS485-B signals.



Linear two-wire RS485 bus — master daisy-chained to slave devices (A / B / G), with 120 Ω termination across A and B at the far end of the backbone.

CHAPTER 02

Supported function codes

0x03 Read Holding Registers

Read one or more configuration registers.

```
REQUEST
[Device ID] [0x03] [Start Address High] [Start Address Low] [Quantity High] [Quantity Low] [CRC Low] [CRC High]
RESPONSE
[Device ID] [0x03] [Byte Count] [Data…] [CRC Low] [CRC High]
```

FRAME	BYTES	NOTES
REQUEST	`28 03 00 00 00 01 [CRC]`	Read 1 register starting at 0 (device address)
RESPONSE	`28 03 02 00 28 [CRC]`	Byte count = 2, value = 40 (0x0028)
REQUEST	`28 03 00 00 00 03 [CRC]`	Read 3 registers starting at 0
RESPONSE	`28 03 06 00 28 00 00 00 00 [CRC]`	Byte count = 6, three register values

0x04 Read Input Registers

Read one or more input registers containing measurements or device information.

```
REQUEST
[Device ID] [0x04] [Start Address High] [Start Address Low] [Quantity High] [Quantity Low] [CRC Low] [CRC High]
RESPONSE
[Device ID] [0x04] [Byte Count] [Data…] [CRC Low] [CRC High]
```

FRAME	BYTES	NOTES
REQUEST	`28 04 00 00 00 01 [CRC]`	Read 1 register starting at 0 (input voltage in mV)
RESPONSE	`28 04 02 5F B4 [CRC]`	Byte count = 2, value = 24500 (0x5FB4 = 24.5 V)
REQUEST	`28 04 00 00 00 04 [CRC]`	Read 4 registers starting at 0 (Vin, Iin, Vout, Iout)
RESPONSE	`28 04 08 5F B4 0C B2 5D C0 0C 80 [CRC]`	Vin = 24.5 V, Iin = 3.25 A, Vout = 24.0 V, Iout = 3.20 A

0x06 Write Single Register

Write a configuration register (device address, baud rate, or reset trigger).

```
REQUEST
[Device ID] [0x06] [Register High] [Register Low] [Value High] [Value Low] [CRC Low] [CRC High]
RESPONSE
[Device ID] [0x06] [Register High] [Register Low] [Value High] [Value Low] [CRC Low] [CRC High]
```

FRAME	BYTES	NOTES
REQUEST	`28 06 00 00 00 2A [CRC]`	Write device address = 42
RESPONSE	`28 06 00 00 00 2A [CRC]`	Echo confirms write

CHAPTER 03

Modbus register map

Addressing

Register **reference numbers** (4xxxx for holding registers, 3xxxx for input registers) are a documentation convention only. The value placed on the wire is the zero-based **PDU address**, shown in the *Offset* column — that is, on-wire address = reference number − 40001 (holding) or − 30001 (input) = the offset.

- Holding register **40001** is addressed on the wire as `0x0000` (function 0x03 / 0x06).
- Input register **30001** is addressed on the wire as `0x0000` (function 0x04).

The function code — not the reference range — selects the address space, so the same on-wire address `0x0000` reads the device address under 0x03 and the input voltage under 0x04.

Holding registers (FC 0x03, 0x06)

Configuration registers (read / write). **All values are uint16.**

ADDRESS	OFFSET	DESCRIPTION	RANGE	NOTES
40001	0	Device Address	1–247	Takes effect after reset
40002	1	Baud Rate Index	0–4	0 = 9600, 1 = 19200, 2 = 38400, 3 = 57600, 4 = 115200; takes effect after reset
40003	2	Reset Command	—	Write 0xAA55 to trigger a reset; always reads 0

> **RESET REGISTER READ-BACK**
> Register 40003 is a write-only command trigger. A read always returns **0** regardless of the value last written — this is expected behaviour, not a fault. The written value initiates the reset and is never stored.

Input registers (FC 0x04) — live telemetry (offsets 0–7)

All values are uint16 with milliunit scaling.

ADDRESS	OFFSET	DESCRIPTION	UNIT	CONVERSION
30001	0	Input Voltage	mV	voltage_v = value / 1000
30002	1	Input Current	mA	current_a = value / 1000
30003	2	Output Voltage	mV	voltage_v = value / 1000
30004	3	Output Current	mA	current_a = value / 1000
30005	4	Temperature	m°C	temp_c = value / 1000
30006	5	Uptime (Low Word)	s	Low 16 bits of uptime
30007	6	Uptime (High Word)	s	High 16 bits of uptime
30008	7	Status Flags	—	Bit field (see below)

Status Flags bit field (register 30008)

A 16-bit bit field. Each bit reports a specific system condition and can be set or cleared independently.

BIT	HEX	NAME	DESCRIPTION
0	0x0001	**System Startup**	Device is initialising (cleared once startup completes)
1	0x0002	**Daily Reset**	Daily reset has occurred (uptime counter reset)
2	0x0004	**Input Measurement Fault**	Fault communicating with the input measurement sensor
3	0x0008	**Output Measurement Fault**	Fault communicating with the output measurement sensor
4	0x0010	**Calibration Invalid**	Stored calibration data failed validation
5	0x0020	**Calibration Missing**	No calibration data stored
6	0x0040	**Temperature Sensor Fault**	Fault communicating with the temperature sensor
7–15	—	**Reserved**	Reserved for future use (currently 0)

Device information registers (offsets 100–107)

ADDRESS	OFFSET	DESCRIPTION	TYPE	NOTES
30101	100	Serial Number Low	uint16	Low 16 bits
30102	101	Serial Number High	uint16	High 16 bits
30103	102	Hardware Revision	uint16	e.g. 1, 2, 3
30104	103	Firmware Major Version	uint16	e.g. 1
30105	104	Firmware Minor Version	uint16	e.g. 0
30106	105	Firmware Patch Version	uint16	e.g. 0
30107	106	Manufacture Date Low	uint16	Unix timestamp low word
30108	107	Manufacture Date High	uint16	Unix timestamp high word

CHAPTER 04

Data encoding

16-BIT REGISTERS

- **Byte order:** big-endian (MSB first).
- **Example:** value 24500 (0x5FB4) is transmitted as `[0x5F, 0xB4]`.

MILLIUNIT SCALING

All measurements use **milliunits** so values carry as integers without floating point:

- **Voltage:** millivolts (mV) — divide by 1000 for volts.
- **Current:** milliamps (mA) — divide by 1000 for amps.
- **Temperature:** millidegrees Celsius (m°C) — divide by 1000 for °C.

For example, an input-voltage reading of `24500` is 24.5 V, and a temperature reading of `25800` is 25.8 °C.

32-BIT VALUES

Uptime, serial number and manufacture date span two uint16 registers:

- **Split:** low word first, high word second.
- **Byte order:** big-endian within each register.

Example — value 70000 (0x00011170) is stored as:

REGISTER	HEX	VALUE
N (low)	0x1170	4464
N+1 (high)	0x0001	1

Reconstruct: `(high << 16) | low = 70000`. All values are transmitted as integers; no floating-point encoding is used.

Conversion examples

QUANTITY	REGISTER VALUE	MEANING
24.5 V	24500	24500 mV
3.25 A	3250	3250 mA
25.8 °C	25800	25800 m°C
Uptime 70000 s	offset 5 = 4464, offset 6 = 1	Combined: 0x0001_1170

CRC-16

Every frame ends with a 16-bit CRC computed over all preceding bytes using polynomial `0xA001` with an initial value of `0xFFFF`, and transmitted **low byte first**.

FRAME (WITHOUT CRC)	CRC-16	ON THE WIRE (CRC APPENDED, LOW BYTE FIRST)
`28 04 00 00 00 01`	`0x3336`	`28 04 00 00 00 01 36 33`
`28 04 02 5F B4`	`0x71DD`	`28 04 02 5F B4 DD 71 (response)`

Exception codes

Standard Modbus exception responses:

CODE	NAME	DESCRIPTION
`0x01`	**Illegal Function**	Function code not supported
`0x02`	**Illegal Data Address**	Register address invalid
`0x03`	**Illegal Data Value**	Value out of range
`0x04`	**Slave Device Failure**	Device cannot process the request

EXCEPTION RESPONSE

```
[Device ID] [Function Code + 0x80] [Exception Code] [CRC Low] [CRC High]
```


Example transactions

All requests are addressed to device ID `0x28 (40)`. CRC bytes are shown as `[CRC]`; worked CRC values are given in § Data encoding.

Example transactions — telemetry & identity

FRAME	BYTES	NOTES
READ INPUT VOLTAGE — FC 0X04, OFFSET 0		
REQUEST	28 04 00 00 00 01 [CRC]	8 bytes total
RESPONSE	28 04 02 5F B4 [CRC]	7 bytes; value 0x5FB4 = 24500 mV = 24.5 V
READ INPUT CURRENT — FC 0X04, OFFSET 1		
REQUEST	28 04 00 01 00 01 [CRC]	Read 1 register at offset 1
RESPONSE	28 04 02 0C B2 [CRC]	Value 0x0CB2 = 3250 mA = 3.25 A
READ TEMPERATURE — FC 0X04, OFFSET 4		
REQUEST	28 04 00 04 00 01 [CRC]	Read 1 register at offset 4
RESPONSE	28 04 02 64 C8 [CRC]	Value 0x64C8 = 25800 m°C = 25.8 °C
READ SERIAL NUMBER — FC 0X04, OFFSETS 100–101 (SPLIT LOW / HIGH WORD)		
REQUEST	28 04 00 64 00 02 [CRC]	Option 1 — read both registers at offset 100
RESPONSE	28 04 04 30 39 00 00 [CRC]	Byte count = 4; Low = 0x3039, High = 0x0000
REQUEST	28 04 00 64 00 01 [CRC]	Option 2 — read offset 100 individually
RESPONSE	28 04 02 30 39 [CRC]	Low word = 0x3039 = 12345
REQUEST	28 04 00 65 00 01 [CRC]	Read offset 101 individually
RESPONSE	28 04 02 00 00 [CRC]	High word = 0x0000. Reconstruct: serial = 12345

Example transactions — configuration writes

FRAME	BYTES	NOTES
READ DEVICE ADDRESS — FC 0X03, HOLDING REGISTER 0		
REQUEST	28 03 00 00 00 01 [CRC]	Read 1 register at offset 0
RESPONSE	28 03 02 00 28 [CRC]	Value 0x0028 = 40 (device address)
WRITE DEVICE ADDRESS — FC 0X06, HOLDING REGISTER 0		
REQUEST	28 06 00 00 00 2A [CRC]	New device address = 42 (0x002A)
RESPONSE	28 06 00 00 00 2A [CRC]	Echo. Changes take effect after a device reset
TRIGGER DEVICE RESET — FC 0X06, HOLDING REGISTER 2		
REQUEST	28 06 00 02 AA 55 [CRC]	Reset trigger value 0xAA55
RESPONSE	28 06 00 02 AA 55 [CRC]	Echo. Device resets immediately after sending the response

CHAPTER 07

Revision history

REV.	DATE	CHANGE	APPROVING OFFICER
1	2026-06-08	Initial customer release	Matthew Wynne