



ARATO INGEGNERIA

Modbus Tester

User manual

Modbus TCP / RTU master and slave · hexadecimal debug · flow analysis

Version 1.0.0 · 2026-10-17

Copyright © 2026 Ing. Roberto Arato · Arato Ingegneria · aratoingegneria.it

Contents

- Contents 2
- 1. Introduction 3
- 2. Installation 3
 - Requirements 3
 - Setup program 3
 - Where settings are stored 3
- 3. The main window 4
 - Header 4
 - Status bar 5
- 4. Master 5
 - 4.1 Connection 5
 - 4.2 Read 5
 - 4.3 Write 6
 - 4.4 Data table 6
 - 4.5 Formats and byte order 7
- 5. Monitor: traffic and analysis 8
 - 5.1 Traffic and Debug HEX 8
 - 5.2 Flow analysis 8
- 6. Slave 9
- 7. Settings 11
- 8. RLIB log 12
- 9. Typical procedures 13
- 10. Reference 13
 - Modbus exception codes 13
 - Supported Modbus functions 13
 - Support 14

1. Introduction

Modbus Tester is Arato Ingegneria's tool to test and commission Modbus devices. It works as a **master** (polling PLCs, instruments, gateways) and as a **slave** (simulating a device to be read by a PLC or SCADA), over **Modbus TCP** and **Modbus RTU** on a serial port or a serial/ethernet converter. It is built on the Modbus classes of the RLIB_NET_10 library.

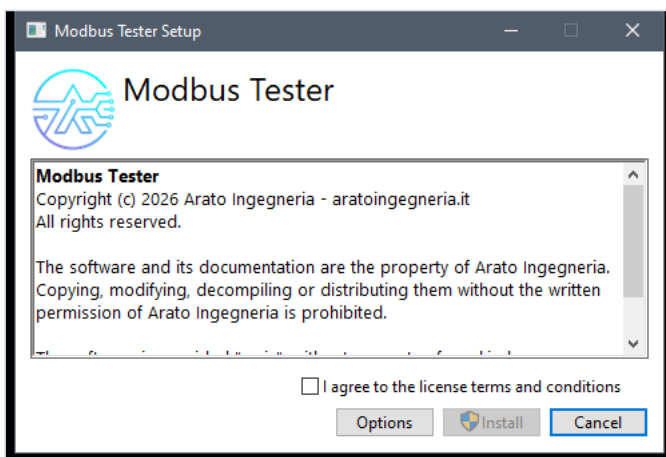
- FC01–FC04 reads, single or polled; FC05, FC06, FC15, FC16 writes; FC43 identification.
- 16, 32 and 64-bit values, **float on 2 registers** and double on 4, hexadecimal, binary, ASCII, in all four byte orders.
- **Debug HEX** mode: every byte sent and received, in hexadecimal, with the frame decoded.
- Flow **analyzer**: tells you what is wrong with the communication and what to check.
- Slave with editable tables and simulated values; RLIB log in the RLIB_GDI_NET_10 LogForm.
- Italian and English interface, light, dark or system theme.

2. Installation

Requirements

- 64-bit Windows 10 or Windows 11.
- **Microsoft .NET 10 Desktop Runtime (x64)**, available from dotnet.microsoft.com. It is not included in the program.
- For Modbus RTU: a serial port (e.g. a USB-RS485 adapter) or a serial/ethernet converter.

Setup program



Run **ModbusTester-1.0.0-Setup-en.exe** (or **-it.exe** for Italian). The setup:

- checks that the .NET 10 Desktop Runtime is present and, if missing, tells where to download it;
- **Options** lets you choose the folder (default: *Program Files\Arato Ingegneria\Modbus Tester*);
- creates Start menu and desktop shortcuts and can launch the program at the end.

The **.msi** packages can be used instead (e.g. for unattended installs with `msiexec /i ... /qn`). Newer versions automatically replace the installed one. The program is also a single **ModbusTester.exe** file that can be copied and run directly.

Where settings are stored

All settings (connections, reads, writes, language, theme, window) are saved in the file **ModbusTester.ini** in the user data folder `%APPDATA%\AratoIngegneria\ModbusTester`, written with RLIB's RConfig. They are saved automatically on exit and from the Settings page.

3. The main window

The screenshot shows the Modbus Tester application in the MASTER tab. The left sidebar has three main sections: CONNECTION, READ, and WRITE. The CONNECTION section shows Modbus TCP settings (IP: 127.0.0.1, Port: 15502, Timeout: 1000, Retries: 0). The READ section shows Unit ID: 1, Function: 04 (Read Input Registers), Address: 0, Quantity: 12, and Polling: 500. The WRITE section shows Function: 16 (Write Multiple Registers), Address: 10, and Values: 3.14. The main area displays a data table with 12 registers (Address 0 to 11). The table shows raw hex values and their decimal equivalents. Below the table is a traffic monitor showing the sequence of data exchanges between the master and slave.

Address	PLC ref.	Raw	Value
0	30001	0x0023	3,2154685E-39
1	30002	0x0368	
2	30003	0xBF71	-0,943622172
3	30004	0x9139	
4	30005	0xBF05	-1,66551208
5	30006	0x2F80	
6	30007	0xFED5	-1,41597302E+38
7	30008	0x004C	
8	30009	0x0000	0
9	30010	0x0000	
10	30011	0x0000	0
11	30012	0x0000	

Master page while polling: connection, read and write on the left; the data table and the traffic monitor on the right.

Header

Item	Function
	Brand Click to open the aratoingegneria.it website in the browser.
	Pages MASTER, SLAVE and SETTINGS . Master and slave keep running when you switch page: the master can poll the program's own slave.
	Theme Cycles the theme on each click: System (follows Windows), Light, Dark.
	Debug HEX Turns the hexadecimal frame trace in the monitor on/off (dot = on).
	Log RLIB Opens the RLIB log window (see chapter 8).

Status bar

● Master connected · Modbus TCP 127.0.0.1:15502 · last response 0 ms | TCP slave running · port 15502 · 1 client · 10 requests

aratoingegneria.it

At the bottom: master state (green dot = connected, red = last request failed) and, if running, slave state with the number of requests served.

4. Master

The master polls a Modbus device: configure the connection, connect, then read and write data.

4.1 Connection

CONNECTION

Protocol

Modbus TCP

IP address

127.0.0.1

Port

15502

Timeout (ms)

1000

Retries

0

Connect

Field	Meaning
Protocol	Modbus TCP; Modbus RTU · serial; Modbus RTU · over TCP (RTU frames with CRC inside a TCP connection, typical of transparent serial/ethernet converters).
IP address, Port	TCP device or converter. Standard Modbus TCP port: 502.
Serial port, Baud, Parity, Bits	Serial RTU only. ↻ refreshes the COM list. They must match the slave (typical: 19200, Even, 8, 1).
Timeout (ms)	Maximum wait for the response.
Retries	Repeats after a timeout or an invalid response (0 = none).
RS485 echo	RTU only: enable it if the interface sends the transmitted bytes back.

Button	Name	Function
Connect	Connect	Opens the connection. Over TCP, if it drops, RLIB reopens it by itself every second.
Disconnect	Disconnect	Closes the connection and stops polling. While connected the connection fields are locked.

4.2 Read

READ

Unit ID

1

Function

04 · Read Input Registers

Address

0

Quantity

12

Polling (ms)

500

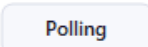
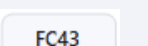
Read

Polling

FC43

Zero-based addresses, as on the wire: register 40001 in the manual is address 0.

Field	Meaning
Unit ID	Slave address (RTU 1–247; over TCP often 1 or 255, depending on the device).
Function	01 Coils, 02 Discrete inputs, 03 Holding registers, 04 Input registers.
Address	First address, zero-based as on the wire: register 40001 in the device manual is address 0.
Quantity	Up to 125 registers or 2000 bits per request.
Polling (ms)	Interval between polled reads.

Button	Name	Function
	Read	Performs one read and fills the data table.
	Polling	Repeats the read at regular intervals.
	Stop	Stops polling (the Polling button turns into Stop while active).
	FC43	Reads the device identification (vendor, product code, revision...) and writes it to the monitor.

4.3 Write

WRITE

Function

16 · Write Multiple Registers

Address

10

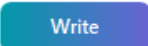
Values

3.14

Write

Values separated by space or ';'. FC06/16 use the format and order chosen above the table (e.g. a float on 2 registers). Bits: 1/0, ON/OFF. Double-click the

Field	Meaning
Function	05 one coil, 06 one register, 15 multiple coils, 16 multiple registers.
Address	First address to write (zero-based).
Values	Separated by space or ';'. For bits: 1/0 or ON/OFF. FC06/FC16 use the format and order chosen above the table: with <i>Float 32 bit</i> the value 3.14 becomes two registers.

Button	Name	Function
	Write	Sends the write; if the table holds a read, it repeats it to show the result.

Quick write from the table: **double-click** a holding register value to edit it (FC06, or FC16 if the format uses several registers); double-click a coil to toggle it (FC05).

4.4 Data table

DATA	Format	Float 32 bit (2 reg)	Order	ABCD · big endian	FC04 · 12 registers from 0 · 0 ms · 21:28:10
Address	PLC ref.	Raw	Value		
0	30001	0x0023	3,2154685E-39		
1	30002	0x036B			
2	30003	0xBF71	-0,943622172		
3	30004	0x9139			
4	30005	0x8FD5	-1,66551208		
5	30006	0x2F80			
6	30007	0xFED5	-1,41597302E+38		
7	30008	0x0D4C			
8	30009	0x0000	0		
9	30010	0x0000			
10	30011	0x0000	0		
11	30012	0x0000			

Input registers read as Float 32 bit: the value is shown on the first row of each register pair.

Field	Meaning
Address	Modbus address (zero-based).
PLC ref.	Manual notation: 0xxxx coil, 1xxxx discrete input, 3xxxx input register, 4xxxx holding register (one-based).
Raw	Raw register value in hexadecimal (or 1/0 for bits).
Value	Value decoded according to Format and Order . Changing them recomputes the table without reading again.

4.5 Formats and byte order

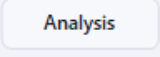
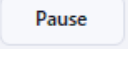
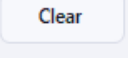
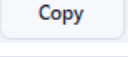
A Modbus register is 16 bits. 32-bit values (integers and floats) take **2 registers**, 64-bit values (integers and doubles) **4**. Each manufacturer arranges them differently: the **Order** tells how to put them back together. A = most significant byte.

Order	Meaning	123.456 as float on the wire
ABCD	Big endian, as in the Modbus specification	42F6 E979
CDAB	Word swap: registers swapped (very common in PLCs)	E979 42F6
BADC	Byte swap: bytes swapped inside each register	F642 79E9
DCBA	Little endian	79E9 F642

Format	Registers	Notes
Unsigned / Signed 16 bit	1	0...65535 / -32768...32767
Hexadecimal, Binary	1	When writing 0x1A2B, 1A2B, 0001 1010... are accepted
ASCII	1+	2 characters per register
Signed / Unsigned 32 bit	2	
Float 32 bit	2	IEEE 754 single precision: the classic PLC "double on two registers"
Signed / Unsigned 64 bit	4	
Double 64 bit	4	IEEE 754 double precision

If a float looks absurd (e.g. 1.2E-38 or 5.4E+21), try another **Order**: it is almost always CDAB instead of ABCD. When writing, decimals can use either a comma or a dot.

5. Monitor: traffic and analysis

Button	Name	Function
	Traffic	Shows the log: events, errors and, with Debug HEX on, every TX/RX frame.
	Analysis	Shows the flow analyzer (5.2).
	Pause	Pauses the log (turns into Resume). Useful to read calmly.
	Clear	Clears the log, or resets the analysis if the Analysis tab is open.
	Copy	Copies the selected lines (Ctrl/Shift to select several) or the whole log to the clipboard.

5.1 Traffic and Debug HEX

21:28:07.267	RX	00 04 00 00 00 1B 01		04 18 00 1C 02 BC BE B3 99 DC BF ED F7 74 03 C1 1A 5F 00 00 00 00 00 00 00	33 B · tid 4 · unit 1 · FC04 Re
21:28:07.771	TX	00 05 00 00 00 06 01		04 00 00 00 0C	12 B · tid 5 · unit 1 · FC04 Read Input Registers · addr 0 · n 12
21:28:07.771	RX	00 05 00 00 00 1B 01		04 18 00 1D 02 D5 BE ED FA 1B BF EC 55 54 A3 61 51 12 00 00 00 00 00 00 00	33 B · tid 5 · unit 1 · FC04 Re
21:28:08.275	TX	00 06 00 00 00 06 01		04 00 00 00 0C	12 B · tid 6 · unit 1 · FC04 Read Input Registers · addr 0 · n 12
21:28:08.275	RX	00 06 00 00 00 1B 01		04 18 00 1E 02 EE BF 12 51 D8 BF EA 42 05 B2 86 67 F7 00 00 00 00 00 00 00	33 B · tid 6 · unit 1 · FC04 Re
21:28:08.791	TX	00 07 00 00 00 06 01		04 00 00 00 0C	12 B · tid 7 · unit 1 · FC04 Read Input Registers · addr 0 · n 12
21:28:08.791	RX	00 07 00 00 00 1B 01		04 18 00 20 03 20 BF 41 BD CF BF E4 EA A6 06 DB 24 C1 00 00 00 00 00 00 00	33 B · tid 7 · unit 1 · FC04 Re

Modbus TCP frames in hexadecimal: TX in cyan, RX in violet, decoding on the right.

Each line shows time, direction, bytes and a decoding. The □ separator splits the frame parts:

Protocol	Example	Meaning
Modbus TCP	00 05 00 00 00 06 01 04 00 00 00 0C	MBAP: transaction id 0005, protocol 0000, length 0006, unit 01 □ PDU: FC04, address 0000, quantity 000C (12)
Modbus RTU	05 04 00 00 00 03 B1 8F	Slave address 05 □ PDU: FC04, address 0, quantity 3 □ CRC (low byte first). A wrong CRC is flagged in the decoding.

Over RTU the monitor shows the bytes **as they arrive** from the port, including split frames, echo and noise that the protocol then discards: it is the right view to understand wiring and baud rate problems.

5.2 Flow analysis

The analyzer watches every frame and the outcome of every request and summarizes the communication: at the top the overall result and statistics (requests, successful, timeouts, exceptions, invalid frames, min/avg/max response times), below the list of problems, each with an explanation, the likely cause (→) and how many times it happened.

 No problems detected Requests 13 · successful 13 (100,0%) · timeout 0 · exceptions 0 · invalid frames 0 · response time min/avg/max 0 / 3 / 16 ms

Normal communication.

Communication problems

Requests 2 · successful 0 (0,0%) · timeout 0 · exceptions 2 · invalid frames 0 · response time min/avg/max —

Exception 02 · Illegal Data Address ERROR

×2

21:28:35

The slave answered with exception 02 (Illegal Data Address) to FC03.
→ Address or quantity outside the device map. Often it is the offset: the manual says 40001 and address 0 must be requested; try ±1 too. Reduce the quantity if the area ends earlier.

Read beyond the device map: exception 02 with the hint about the 40001 offset (dark theme).

Reported problem	What to check
The slave does not respond / Timeouts	Unit id, IP and port, baud rate and parity, A/B wires, power supply, 120 Ω termination, common ground.
Bytes arrive but not a valid response	Different baud/parity/stop bits, converter splitting or losing bytes.
Bad CRC	Serial settings, noise, termination, unshielded cable.
Transmission echo	RS485 adapter without echo suppression: enable the Echo option in the master.
Truncated frames / extra bytes	Timeout too short, slow converter, noise, two slaves with the same address.
Response from a different slave	Duplicate addresses on the bus or late responses.
Different transaction id / unit id	Late responses (increase the timeout) or a gateway rewriting the unit id.
Invalid MBAP header	The device does not speak Modbus TCP: try Modbus RTU over TCP.
Exceptions 01...0B	See the table in chapter 10.
Responses close to the timeout	Increase the timeout.
The master polls another address (slave)	The slave sees requests for other unit ids: align the address.

6. Slave

The slave simulates a Modbus device with four tables (coils, discrete inputs, holding registers, input registers) of 10000 items each, addresses 0...9999. An external master (PLC, SCADA, or this program's own master) reads and writes them.

SLAVE

Protocol: Modbus TCP - server
 Listening port: 15502
 Unit ID: 1
☒ Answer any unit id
 Stop slave
 TCP: listening on all interfaces. RTU over TCP: the slave connects to the given serial/ethernet converter.

STATUS

Listening on port 15502
 1 clients connected
 20 requests served - 0 exceptions
☒ Simulate changing values
 Input register 0 counter, 1 sawtooth 0..1000, 2-3 float sine, 4-7 double cosine; discrete inputs 0..7 running.
 Clear tables

TABLES Input registers (3x) From 0 Rows 100 Format Unsigned 16 bit Order ABCD - big endian

Address	PLC ref.	Raw	Value
0	30001	0x002D	45
1	30002	0x007C	124
2	30003	0x0F1C	48924
3	30004	0x9735	38709
4	30005	0x3FE9	16361
5	30006	0x5088	20664
6	30007	0x63C4	25540
7	30008	0xC78F	51087
8	30009	0x0000	0
9	30010	0x0000	0
10	30011	0x0000	0
11	30012	0x0000	0
12	30013	0x0000	0
13	30014	0x0000	0
14	30015	0x0000	0

MONITOR Traffic Analysis Pause Clear Copy

21:28:12.329 TX 00 0E 00 00 00 18 01 04 18 00 27 03 CF BF 7C 9F 72 3F C4 88 FC 76 C2 5C 0F 00 00 00 00 00 00 127.0.0.1:58768 - 33 B - tid 14
 21:28:12.844 RX 00 0F 00 00 00 06 01 04 18 00 28 03 E8 BF 75 7C 10 3F D2 27 85 70 68 4A D9 00 00 00 00 00 00 127.0.0.1:58768 - 33 B - tid 15
 21:28:13.344 RX 00 10 00 00 00 06 01 04 00 00 00 0C 127.0.0.1:58768 - 12 B - tid 16 - unit 1 - FC04 Read Input Registers - addr 0 - n 12 127.0.0.1:58768 - 33 B - tid 16
 21:28:13.344 TX 00 10 00 00 00 18 01 04 18 00 29 00 18 BF 6A 84 04 3F D9 AA 06 C1 2B A6 E0 00 00 00 00 00 00 127.0.0.1:58768 - 33 B - tid 16
 21:28:13.846 RX 00 11 00 00 00 06 01 04 00 00 00 0C 127.0.0.1:58768 - 12 B - tid 17 - unit 1 - FC04 Read Input Registers - addr 0 - n 12 127.0.0.1:58768 - 33 B - tid 17
 21:28:13.846 TX 00 11 00 00 00 18 01 04 18 00 2A 00 31 BF 58 E3 22 3F E0 63 01 15 30 54 09 00 00 00 00 00 127.0.0.1:58768 - 33 B - tid 17
 21:28:14.361 RX 00 12 00 00 00 06 01 04 00 00 00 0C 127.0.0.1:58768 - 12 B - tid 18 - unit 1 - FC04 Read Input Registers - addr 0 - n 12 127.0.0.1:58768 - 33 B - tid 18
 21:28:14.361 TX 00 12 00 00 00 18 01 04 18 00 2B 00 4A BF 49 D3 D7 3F E3 AF 88 98 E3 48 31 00 00 00 00 00 00 127.0.0.1:58768 - 33 B - tid 18
 21:28:14.864 RX 00 13 00 00 00 06 01 04 00 00 00 0C 127.0.0.1:58768 - 12 B - tid 19 - unit 1 - FC04 Read Input Registers - addr 0 - n 12 127.0.0.1:58768 - 33 B - tid 19
 21:28:14.864 TX 00 13 00 00 00 18 01 04 18 00 2C 00 63 BF 34 9E 4A 3F E6 AD 6C 3C 07 D4 48 00 00 00 00 00 00 127.0.0.1:58768 - 33 B - tid 19
 21:28:15.365 RX 00 14 00 00 00 06 01 04 00 00 00 0C 127.0.0.1:58768 - 12 B - tid 20 - unit 1 - FC04 Read Input Registers - addr 0 - n 12 127.0.0.1:58768 - 33 B - tid 20
 21:28:15.365 TX 00 14 00 00 00 18 01 04 18 00 2D 00 7C BF 1C 97 35 3F E9 50 B8 63 C4 C7 8F 00 00 00 00 00 00 127.0.0.1:58768 - 33 B - tid 20

TCP slave running - port 15502 - 1 client - 20 requests | Master connected - Modbus TCP 127.0.0.1:15502 - last response 0 ms

Slave page with the TCP slave running and simulated values.

SLAVE

Protocol: Modbus TCP - server
 Listening port: 15502
 Unit ID: 1
☒ Answer any unit id
 Stop slave
 TCP: listening on all interfaces. RTU over TCP: the slave connects to the given serial/ethernet converter.

Field	Meaning
Protocol	Modbus TCP - server: listens on the given port, on all network interfaces. Serial RTU: on the COM port. RTU over TCP: the slave connects to the given serial/ethernet converter.
Unit ID	Slave address (1–247).
Answer any unit id	TCP only: answers every unit id; when off, others get exception 0B.
Response delay	RTU only: wait before answering, for RS485 converters slow to switch direction.

Button	Name	Function
Start slave	Start slave	Starts the slave with the given parameters.
Stop slave	Stop slave	Stops the slave and closes the connections.
Clear tables	Clear tables	Sets every value of the four tables to zero.

STATUS

Listening on port 15502
1 clients connected
20 requests served · 0 exceptions

☒ Simulate changing values

Input register 0 counter, 1 sawtooth 0..1000, 2-3 float sine, 4-7 double cosine; discrete inputs 0..7 running.

Clear tables

Status: listening port, connected clients, requests served, exceptions sent (over RTU also frames with bad CRC and frames seen on the bus).

Simulate changing values: input register 0 counter, 1 sawtooth 0...1000, 2-3 sine as float, 4-7 cosine as double; discrete inputs 0...7 switching on in turn.

TABLES

Input registers (3x)

From 0

Rows 100

Format Unsigned 16 bit

Order ABCD · big endian

Above the table you choose the area, the first address (**From**), the number of **Rows**, the **Format** and the **Order**. All areas are editable: double-click a value to write it, double-click a bit to toggle it. Rows written by a master briefly light up and the monitor reports *The master wrote...*

Windows may ask for firewall permission the first time the TCP slave starts listening: allow it so that network devices can connect.

7. Settings

Modbus Tester - Arato Ingegneria

ARATO INGEGNERIA
MODBUS TESTER TCP · RTU · master · slave

MASTER SLAVE SETTINGS

Theme: Light Debug HEX Log RLIB

GENERAL

Language English

The language changes when the program restarts.

Theme Light

☒ Show TX/RX frames in hexadecimal

Monitor lines 5000

Maximum lines kept in the master and slave traffic: the oldest are removed.

LOG RLIB

The RLIB log (connections, channel errors, TX/RX lines of the tester) is shown in the RLIB_GDI_NET_10 LogForm.

☐ Also write the RLIB log to a file

Open RLIB log

CONFIGURATION

Settings are saved automatically on exit; "Save" writes them now.

Save Defaults

ABOUT

Modbus Tester 1.0.0

Modbus TCP / RTU master and slave, hexadecimal debug and flow analysis, built on RLIB_NET_10.

Copyright © 2026 Ing. Roberto Arato - Arato Ingegneria. All rights reserved.

.NET 10.0.13

aratoingegneria.it

Master connected · Modbus TCP 127.0.0.1:15502 · last response 1 ms

TCP slave running · port 15502 · 1 client · 24 requests

aratoingegneria.it

GENERAL

LanguageEnglish

The language changes when the program restarts.

ThemeLight

☒ Show TX/RX frames in hexadecimal

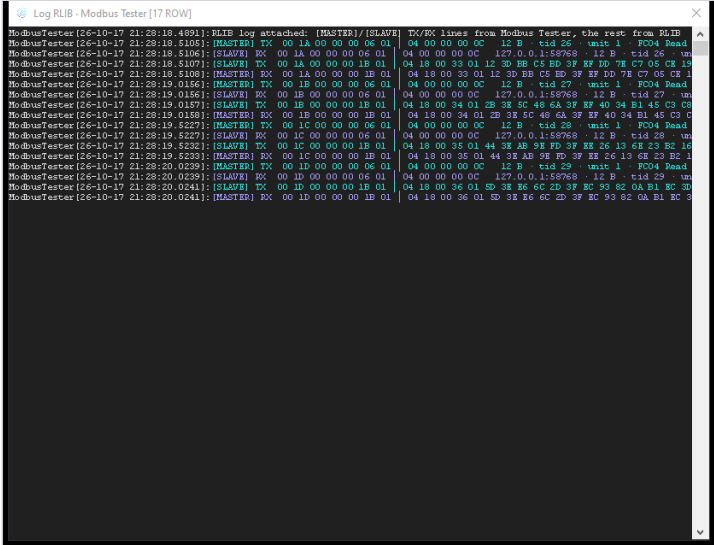
Monitor lines5000

Maximum lines kept in the master and slave traffic; the oldest are removed.

Field	Meaning
Language	Italiano or English. The change applies on restart: the program asks whether to restart now.
Theme	System, Light, Dark (same as the header button).
Hexadecimal frames	Same as the Debug HEX button.
Monitor lines	Lines kept in the master and slave traffic; the oldest are removed.

Button	Name	Function
Open RLIB log	Open RLIB log	Opens the RLIB log window. The option above also writes the log to a file (rlog.txt, rotated).
Save	Save	Writes all settings now (otherwise they are saved on exit).
Defaults	Defaults	After confirmation deletes the settings and restarts the program with default values.

8. RLIB log



The **RLIB log** window is the LogForm of the RLIB_GDI_NET_10 library. It shows the RLIB log: socket connections and drops, serial and TCP channel errors, and the same TX/RX lines as the monitor, prefixed with **[MASTER]** or **[SLAVE]**.

It is useful to attach everything that happened to a report; closing it just hides it and the lines are kept.

9. Typical procedures

Trying the program without devices

- **SLAVE** page: Modbus TCP · server, port 502 (or 5020), **Simulate values** on, **Start slave**.
- **MASTER** page: Modbus TCP, IP 127.0.0.1, same port, **Connect**.
- Read FC04, address 0, quantity 8, **Polling**: values change; with Format *Float 32 bit* register 2 shows the sine.

Reading a PLC or an instrument over Modbus TCP

- Enter the device IP and port, the Unit ID stated by the manufacturer, **Connect**.
- Convert the manual address: 40001 → FC03 address 0; 30010 → FC04 address 9.
- If floats look absurd change the **Order**; on exception 02 try ± 1 on the address.

RS485 bus with a USB adapter

- Modbus RTU · serial, choose the COM port (↻ to refresh), baud rate, parity and bits as on the slave.
- With Debug HEX on, check in the traffic whether bytes arrive: no bytes = wiring or address; garbage bytes = baud rate or parity.
- Open the **Analysis** tab and follow the hints (echo, CRC, truncated frames...).

10. Reference

Modbus exception codes

Code	Name	Typical cause
01	Illegal Function	Function not supported by the device (e.g. FC04 on a device with holding registers only).
02	Illegal Data Address	Address or quantity outside the map: often the 40001/0 offset.
03	Illegal Data Value	Quantity not allowed or value rejected.
04	Slave Device Failure	Internal device error.
05 / 06	Acknowledge / Busy	Device busy: slow down polling.
0A / 0B	Gateway Path / Target	TCP→RTU gateway: the serial slave is unreachable or does not answer.

Supported Modbus functions

FC	Name	Master	Slave
01 / 02	Read Coils / Discrete Inputs	✓	✓
03 / 04	Read Holding / Input Registers	✓	✓
05 / 06	Write Single Coil / Register	✓	✓
15 / 16	Write Multiple Coils / Registers	✓	✓
08 / 11 / 22 / 23	Diagnostics, Comm Event Counter, Mask Write, Read/Write	—	✓
43	Read Device Identification	✓	✓

Support

Arato Ingegneria · **aratoingegneria.it**. When reporting a problem attach the text copied from the monitor (Copy button) or the RLIB log.