YZW APPLICATION GUIDE / CUSTOMER EDITION

Three platforms

Data Acquisition and Reporting Guide

First setup • RS485 metering • PLC states • Checks at every step

Follow your use case from wiring to acquisition and platform ingestion. Each operation includes expected behavior, cautions, fault checks and evidence to save. Screenshot numbers match the field tables.

Connect
Match the physical interface

Verify readings
Start with one read-only point

Integrate
Map topics, fields and devices

Troubleshoot
Follow the data path

Workflow: connect → verify a read-only point → configure reporting → receive and display

English • V1.4 • 2026-11-05

MT7628 / MT7621 / MT7981

Contents

Page numbers refer to this PDF. Select a chapter to open it; use Contents in the footer to return.

01 | Understand the workflow 4
1.1 Who this guide is for 4
1.2 Common steps and model differences 4
1.3 Acquisition and reporting are separate steps 4
1.4 Choose the correct application 5
1.5 Choose your use case 5
02 | Connect equipment and prepare parameters 6
2.1 Information you need 6
2.2 Keep four kinds of address separate 7
2.3 Arrange the local lab network 7
2.4 Prepare evidence and test from the gateway 8
2.5 Inspect status and export gateway logs 10
03 | First exercise: connect a simulated Modbus meter 12
3.1 Start the supplied lab tools 12
3.2 Add a TCP device connection 13
04 | Add a command and verify one value 16
4.1 Fill in this read-only point 16
4.2 Why enter 0 when the manual says 40001? 17
4.3 Select function, data type and scale 18
05 | Configure MQTT reporting and receive data 19
5.1 Complete global settings 19
5.2 View received messages on the PC 21
5.3 Check all of these results 22
06 | Use case B: RS485 electrical metering to your platform 23
6.1 Check wiring and serial-port ownership 23
6.2 Add an RTU connection 24
6.3 Example: two slave IDs on one bus 27
07 | Use case C: MT7981 PLC states and separate reporting 28

7.1 When to use PLC Collect 28
7.2 Add an acquisition connection 28
7.3 Add one read-only point 30
7.4 Configure PLC Data Reporting 31
7.5 Additional checks 33
08 | Connect data to your platform 34
8.1 Identify the platform's ingestion endpoint 34
8.2 Obtain this information from your administrator 34
8.3 Enter platform parameters in the right application 36
8.4 Inspect raw messages before the dashboard 37
8.5 Turn raw payloads into displayed points 38
8.6 Plan topics for multiple devices 40
8.7 Resolve platform requirements absent from the current page 40
8.8 Local Broker, other reporting methods and DTU 41
09 | Move to the site and verify recovery 42
9.1 Save a recoverable site baseline 42
9.2 Verify restart and network recovery in a maintenance window 43
9.3 Submit evidence for the failed step 44
10 | Verify operation and keep recovery records 44
10.1 Complete acceptance using the recorded results 44
10.2 Project parameter record 45

01 | Understand the workflow

1.1 Who this guide is for

Use this guide with **MT7628, MT7621 and MT7981** products that provide the acquisition and reporting functions described here. **Modbus RTU/TCP acquisition and MQTT reporting follow the same basic steps across the three platforms.** Follow one common procedure, check the model-specific serial-port mapping in Chapter 02, and use Chapter 07 for the separate PLC application on applicable MT7981 products. Most operations use the gateway Web interface; Linux commands are not required.

1.2 Common steps and model differences

Item	How it applies	What to do
Modbus TCP/RTU connections, commands and global MQTT reporting	Basic fields and save sequence are common	Follow Chapters 02–06.
Serial-port quantity, electrical type and Serial1/Serial2 mapping	Depend on the exact model and delivered configuration	Check Chapter 02 before selecting a port.
PLC protocols and their reporting rules	Apply to MT7981 models with the PLC application	Follow Chapter 07; skip it for MT7628/MT7621.
Payloads and data types	Differ between the Modbus and PLC applications	Use a parser that matches the actual payload.

Current product range: MT7628 — YZW300X, YZW350L, YZW400, YZWK001 and YZW1010; MT7621 — YZW280, YZW380 and YZW800L; MT7981 — YZW001L, YZW300L, YZW800, YZW900, YZW900L and YZW1000. Check your model's interfaces, menus and delivered applications. YZW900 CPE has no physical serial port; its PLC applications also require the corresponding eMMC/preinstalled configuration, so the RTU wiring example does not apply to it.

Read the numbered screenshots as follows: red numbers restart at ① in every figure. They show the order of operations within that figure, not slave IDs, register addresses or values to enter. The “Step” column in the following table refers to the same numbers. Example values are not factory defaults.

1.3 Acquisition and reporting are separate steps

Acquisition: the gateway reads specified points from a meter or PLC. **Reporting:** it sends the results to your platform server. Verify one correct reading first, then verify reporting. A reachable server does not prove that a meter has been read successfully.

Stage	What happens	How to check it
① Data source	A meter, PLC or PC simulator provides a point value	Obtain a verified point definition and known value.
② Gateway acquisition	The gateway reads the specified slave and registers	Compare one read-only point.
③ Gateway reporting	The gateway sends the result to the configured server	Check the reporting method, server and topic.
④ MQTT Broker	The Broker receives and forwards MQTT messages	Subscribe using a client on the same Broker.
⑤ Your platform	Your platform parses and displays the data	Match a new message's value and time to the source.

1.4 Choose the correct application

Your equipment	Where to configure it	Information to obtain first
RS485 meter using Modbus RTU	Application > Modbus Collect; select RTU	Serial settings, slave ID and register table.
Ethernet meter/PLC using Modbus TCP	Application > Modbus Collect; select TCP	IP address, service port, slave ID and register table.
Equipment sending a custom serial byte stream	Your model's DTU function	The device protocol and a matching receiver/parser.
Equipment requiring a PLC-specific protocol	The matching protocol in your delivered PLC application	Exact CPU/series, access permission and engineering point table.

For Modbus, use **Modbus Collect**. Applicable MT7981 products also provide **PLC Collect** with separate reporting settings. Check the actual application and protocol list rather than selecting a driver by brand name alone.

1.5 Choose your use case

Use case	Goal	Where to start and what to verify
A: First sample setup	One PC simulates a meter and Broker; receive a new temperature=255 message	Chapters 02–05, without a SIM or Internet connection. Verify the PC first, then prove the gateway sent a new message.
B: Electrical-cabinet meter to your platform	Read voltage through RS485 and report over the existing LAN or cellular connection	After management access works, use Chapter 06, then Chapter 08. Compare the meter display, raw message and dashboard.

Use case	Goal	Where to start and what to verify
C: Production PLC running state	The MT7981 PLC application reads a running bit and reports it through its own rule	Chapter 07: verify one Bool point before Chapter 08. Use the PLC engineer's point table and online monitor as the reference.

Use case B illustrates the first voltage point of an Eastron SDM120M: the manufacturer's V2.9 document specifies FC04, reference 30001, PDU start 0 and Float. Confirm word order, actual serial settings and matching hardware against the manufacturer's protocol and meter; do not apply this example to another meter. In use case C, name the point pump_running if appropriate and obtain its address from your PLC project. DB1.0 in the figure illustrates address syntax only.

How to use each step: perform the operation, then read the expected behavior, files/pages and fault checks immediately below it. Meet that step's pass condition before proceeding. Troubleshoot at the failed operation rather than searching for one general table at the end.

02 | Connect equipment and prepare parameters

2.1 Information you need

Item	Required information	Where to obtain it
Gateway	Exact model, current LAN address, management credentials and terminal definitions	Device label, delivery records or technical support.
Meter/PLC	IP or serial settings, slave ID, function code, register address, type and scale	Meter manufacturer or PLC engineer.
MQTT server	Address, port, credentials, topic permissions and certificates	Your platform administrator.
Management PC	An address that can reach the gateway LAN; a browser	Your network administrator.

Check the actual serial port; do not select Serial2 solely by CPU platform.

Reference model/configuration	RS485 port in this guide	What to check
YZW1010 (MT7628)	Serial2	Serial1 is RS232 in the reference configuration.
YZW280 (MT7621)	Serial1	This example uses the RS485 configuration; check your terminals and options.

Reference model/configuration	RS485 port in this guide	What to check
001 series (MT7981)	Serial2	Serial1 is RS232 in the reference configuration.
Other models	Use the model manual and delivery record	Do not assume both illustrated ports exist; K001 serial type depends on the ordered configuration.

A software setting cannot add a missing physical interface or electrically convert RS232 to RS485.

Connect power and LAN according to the exact model manual. Log in using the gateway's **current management address and supplied credentials**. Demo IP addresses and test accounts in this guide are not factory defaults. All configuration described here uses the Web interface and requires no SSH/root access.

2.2 Keep four kinds of address separate

Name	What it identifies	Lab example
Gateway LAN address	The gateway whose management page you open	192.168.10.1.
Modbus TCP address	The slave providing register values	192.168.10.10, port 1502.
MQTT server address	The Broker receiving reports	192.168.10.10, port 1884.
Modbus slave ID/register start	A slave's unit ID/a location within its data area	Slave ID 1; register start 0. Neither is an IP address.

In this lab, one PC simulates both the Modbus meter and MQTT Broker, so their IP addresses match while their **ports differ**. In a production project, they may be separate systems.

2.3 Arrange the local lab network

Use an isolated test gateway and PC. The example network is gateway LAN **192.168.10.1/255.255.255.0**, PC **192.168.10.10/255.255.255.0**. Connect the PC to a LAN port and avoid duplicate addresses. Use a switch if additional ports are needed.

If the current network differs, either plan an isolated LAN according to the product manual or retain the current subnet and consistently replace all gateway, PC and slave IP examples with actual unused addresses. Ports 1502/1884, slave ID 1 and register start 0 do not change just because the subnet changes. The local lab needs no SIM, Internet access or cloud account.

Back up the configuration before changing LAN settings. A LAN change may interrupt management access. Do not apply this test subnet to an operating production gateway.

Check and troubleshoot this step | C01

Expected behavior: The PC opens the current gateway address and reaches the application after login. Internet access is unnecessary for the local exercise.

Pages/files and evidence to inspect: Run ipconfig /all on the PC and inspect the active physical adapter IPv4 address, mask and gateway; save pc_network.txt as described in 2.4. Record the actual management URL/port and save a post-login screenshot.

Observed problem	Checks in order	Condition to proceed
Browser timeout/unreachable	Check the actual LAN port and physical link. Inspect the active PC address/mask for subnet or duplicate-IP errors. Retry the current address with the correct HTTP/HTTPS scheme and port; a link light does not verify IP settings.	Continue only after stable login. Preserve the browser error and wiring; do not reset immediately.
Login page appears but rejects credentials	Confirm this is the intended gateway, then check supplied credentials, case, keyboard input and pasted spaces. Broker credentials do not log into the gateway.	Correct credentials open management. Preserve the error without exposing the password.
Old page disconnects after a LAN change	Use the recorded new address. Renew the PC address or apply the planned subnet; check its current ipconfig /all address rather than the previous setting.	Log in on the new subnet and recheck LAN settings before acquisition.

2.4 Prepare evidence and test from the gateway

Create a project folder on the management PC, for example `Site01`, and copy in the supplied `Forms` templates. The commands below create an `Evidence` subfolder inside that project. Names such as `pc_network.txt` and `mqtt_messages.txt` below refer to **evidence you save on the PC**, not internal gateway paths. Download gateway logs through the Web interface; no SSH/root is required.

Open `Forms/01_Project_Plan_EN.csv` and enter the actual model, gateway LAN/PC addresses, slave/meter address, Broker host/port, uplink and time zone. Only use case A uses 192.168.10.1/192.168.10.10; keep the planned site addresses for B and C.

In Windows PowerShell, open your project folder and save the PC's network information:

Copyable original PowerShell commands are supplied in Lab_Tools/Windows_ReadOnly_Commands_EN.txt

```
New-Item -ItemType Directory -Force .\Evidence | Out-Null
ipconfig /all |
  Out-File .\Evidence\pc_network.txt -Encoding utf8
```

Open **Application > Diagnostics** and follow Figure 2-1. The page provides Ping and Traceroute. Test the actual destination IP first, then the real hostname for the same service, and record the source IP used.

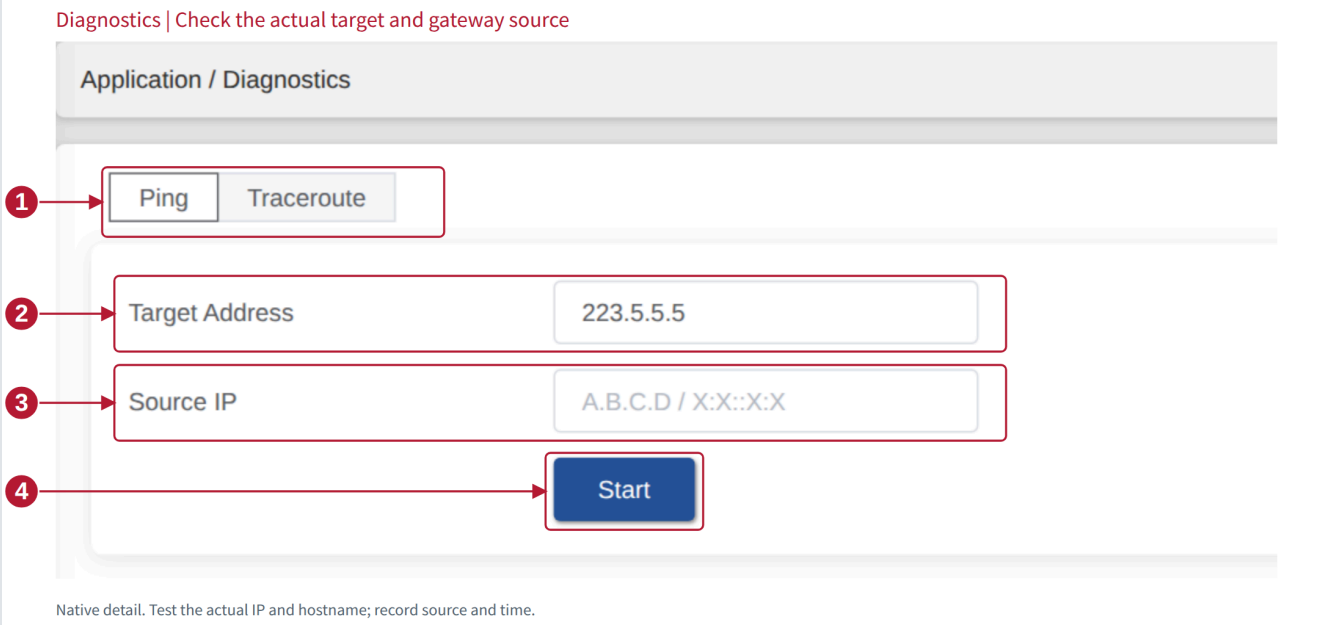


Figure 2-1. Diagnostics: ① select Ping, ② enter the actual target, ③ check the source IP, ④ start and save results. The address shown is a native UI example.

Step	UI location	What to do	Expected behavior
①	Ping / Traceroute	Start with Ping; use Traceroute when you need path information	Diagnostics do not change the acquisition connection.
②	Target Address	For A, first test PC 192.168.10.10. For production, test the server IP, then its real hostname	ICMP-enabled targets reply; hostname resolution matches the project plan.
③	Source IP	If specifying a source, use the gateway's actual local IP on that uplink. If unsure, leave blank for the default route and record this	Identifies the test source. Do not enter the meter IP or an interface name.
④	Start / result	Start, record time/target/source/result, then stop before changing the target	The result area shows replies or errors. Copy to `gateway_network.txt` or save a screenshot.

If a terminal has no Export control, copy the complete output for this attempt into a text editor and save as UTF-8 under the filename specified at the step. Include target/time/errors in screenshots and keep separate attempts for comparison.

A PC-side check of the lab slave port can be saved as follows:

```
Test-NetConnection 192.168.10.10 -Port 1502 |
  Out-File .\Evidence\tcp_1502.txt -Encoding utf8
```

Inspect `TcpTestSucceeded` in `tcp_1502.txt`: True means this PC established a TCP connection to the port; for False check host/listener/firewall first. It does not verify the gateway path, Modbus ID or registers.

Ping proves only ICMP reachability on that path; some servers do not reply. Verify the application port, authentication and topics separately in Chapters 05/08. A PC port test does not replace checking the gateway's actual uplink.

Check and troubleshoot this step | C02

Expected behavior: An ICMP-enabled target replies from the selected source. Use the real Broker hostname and record target, source IP and result together.

Pages/files and evidence to inspect: Save first-hop/PC, server-IP and server-hostname results in gateway_network.txt, including whether Source IP was specified.

Observed problem	Checks in order	Condition to proceed
PC has Internet, gateway cannot reach server	Check the gateway uplink/address in Overview/Network and retest from gateway Diagnostics. A specified source must belong to the gateway, not the PC or meter.	Identify the gateway path; a successful PC test is comparison evidence only.
Server IP reachable but hostname fails	Check hostname spelling without scheme/path and the gateway DNS settings. Ask the administrator to verify DNS records. PC Resolve-DnsName provides comparison, not proof of gateway DNS.	Gateway resolves the actual hostname; retain errors and test source.
No Ping reply or intermediate * in Traceroute	Check ICMP policy and verify the actual MQTT connection with Broker logs. An intermediate router may ignore probes while forwarding traffic.	Verify the application or identify a specific network fault; leave register settings alone.

2.5 Inspect status and export gateway logs

Open **System Management > Log** and follow Figure 2-2. Record gateway and PC time first. If they differ, check the time zone/time synchronization page before correlating fault times.

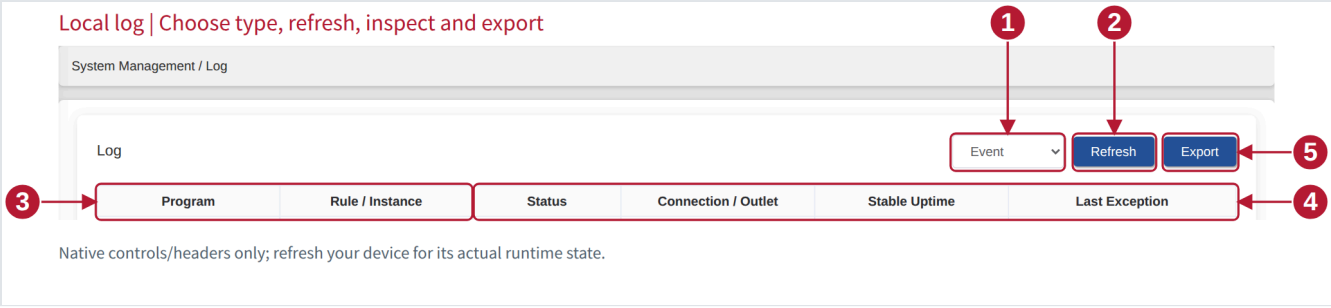


Figure 2-2. Local log controls and headers: ① choose Event/System, ② Refresh, ③ locate the relevant program/instance, ④ inspect status/exceptions, ⑤ Export. Only the native controls and table headers are shown.

Step	Action	What to inspect/save	How to interpret it
①	Select Event first	The program runtime table on the reference version	This is not a live meter-value table or a complete payload history.
②	Click Refresh	Status after the current operation, with its time	Do not use an old page as the latest result.
③	Find the relevant program/instance, if listed	Data Collection, Data Report, Mobile, Link Manager or the application used	Inspect your own service. An M2M Cloud connection issue is not automatically the acquisition MQTT issue.
④	Check Status, Connection/Outlet and Last Exception	Starting, waiting, connecting, running or abnormal state and actual uplink	A normal program state does not verify point values or new reports.
⑤	Export; then select System, Refresh and Export again	Reference downloads: `event_log.txt` and `system_log.txt`; use the actual downloaded names	Save under this attempt's Evidence folder and check time to avoid reusing an old download.

Open logs in a text editor. Check time first, then search for the relevant program, instance, target and error. Keep context around the first error. Event exports may differ from the on-screen table. System logs may not include every application's detailed communication records. If details are absent, retain both files and the step's test results; ask support for the version-specific application-log export. Empty logs do not establish that a service is healthy.

Use only view, Refresh and Export here. The Remote Log switch below is a separate function; exporting does not require enabling it or clicking its Save button.

Check and troubleshoot this step | C03

Expected behavior: Refresh shows available runtime/log information. Both exports open in a text editor. Running status alone does not verify every register or dashboard.

Pages/files and evidence to inspect: Save event_log.txt, system_log.txt and the Event-page screenshot. Check download/file times and timestamps inside; retain context around the failed operation.

Observed problem	Checks in order	Condition to proceed
Relevant program row is absent	Confirm global/rule enable and saves, then refresh. Some versions list only instances with runtime records. Verify real readings/receiver results without assuming an Online indicator exists.	Record the row if present; otherwise state its absence and keep configuration/test evidence.
Export empty or contains old data	Refresh after selecting each type, export again, and check duplicate download names and gateway clock. If application details are unavailable, retain tests and request the version-specific log entry from support.	Files match this attempt, or the missing coverage is explicitly recorded.
Log read/export fails	Save the exact error/time. Check the login session and re-enter the log page; inspect blocked downloads. Remote Log is not required for export.	Export succeeds, or provide the page/error and actual download result.

03 | First exercise: connect a simulated Modbus meter

3.1 Start the supplied lab tools

Extract **Lab_Tools** supplied with this guide. On a Windows PC with Python 3 installed, double-click **1_start_lab.cmd**. Keep its window open after `LAB READY` appears. The PC now provides a Modbus TCP slave on port 1502: slave ID 1, holding register start 0, initial value **255**. The MQTT Broker listens on port 1884.

If the tool cannot bind to the address, run `ipconfig` and check that the PC uses 192.168.10.10. If retaining another subnet, first edit **lab_address.cmd**. Close duplicate lab windows if a port is occupied. Allow TCP 1502 and 1884 only as required for your test network; keep the PC firewall enabled.

Double-click **2_check_modbus.cmd**. Expect `values [255]`. Stop the simulator with Ctrl+C when finished; keep its service window running during gateway acquisition. This verifies the PC's simulated slave, not gateway acquisition. If you already have a real meter, skip the simulator and use its verified parameters in the following pages.

Check and troubleshoot this step | C04

Expected behavior: LAB READY appears and the service window stays open. PC self-check returns values [255]; this verifies the PC service, not gateway access.

Pages/files and evidence to inspect: Copy startup output to lab_start.txt and the complete self-check to modbus_selfcheck.txt. Record PC IP/start time.

Observed problem	Checks in order	Condition to proceed
Python missing/window closes	Read Lab_Tools/README_EN.txt and install Python 3. Launchers try py -3 then python. Run them on the PC, not the gateway.	Rerun and keep the LAB READY window open.
Cannot assign requested address	Check that the PC actually owns the address in lab_address.cmd. Set YZW_LAB_IP to its real test IP and consistently update slave/Broker examples on the gateway.	Service binds to the correct adapter and self-check reads 255.
Address already in use	Close duplicate lab instances and check whether required services already own 1502/1884. Keep existing business services and the firewall running.	One lab instance runs and self-check passes; otherwise use an arranged isolated PC.
LAB READY but no 255 in self-check	Check self-check host/port and request time in the service window. Fix the PC-local test before gateway networking; retain the gateway register definition.	modbus_selfcheck.txt contains values [255] for this attempt.

3.2 Add a TCP device connection

Use ①→②→③ in Figure 3-1 to open Add Device, then complete Figure 3-2.

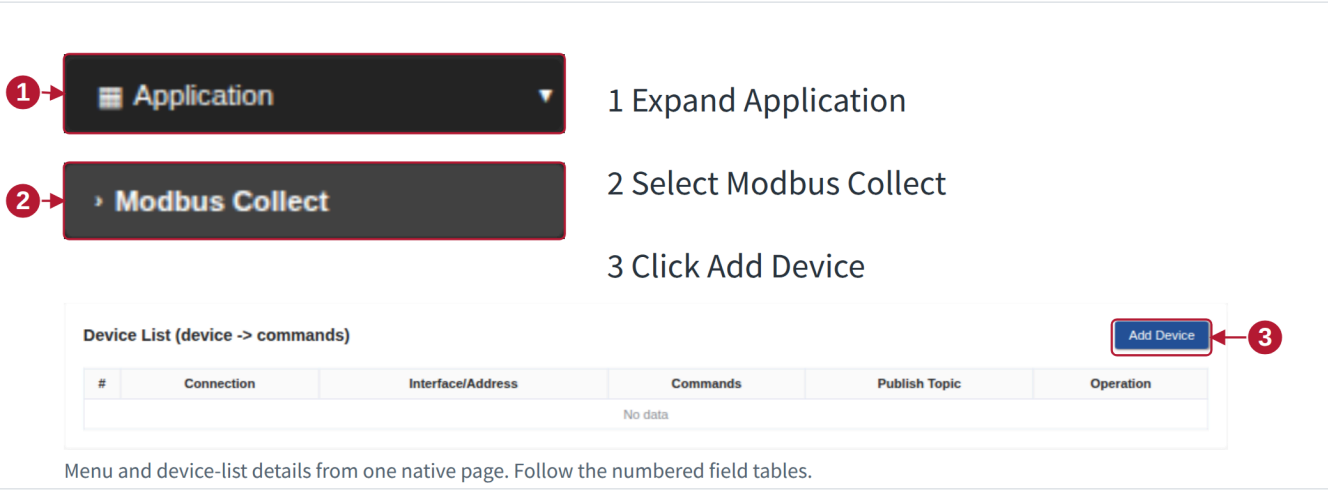


Figure 3-1. ① Expand Application, ② select Modbus Collect, ③ click Add Device. Two areas of the same native page are shown.

Order within this figure: ① Application; ② Modbus Collect; ③ Add Device in the device list.

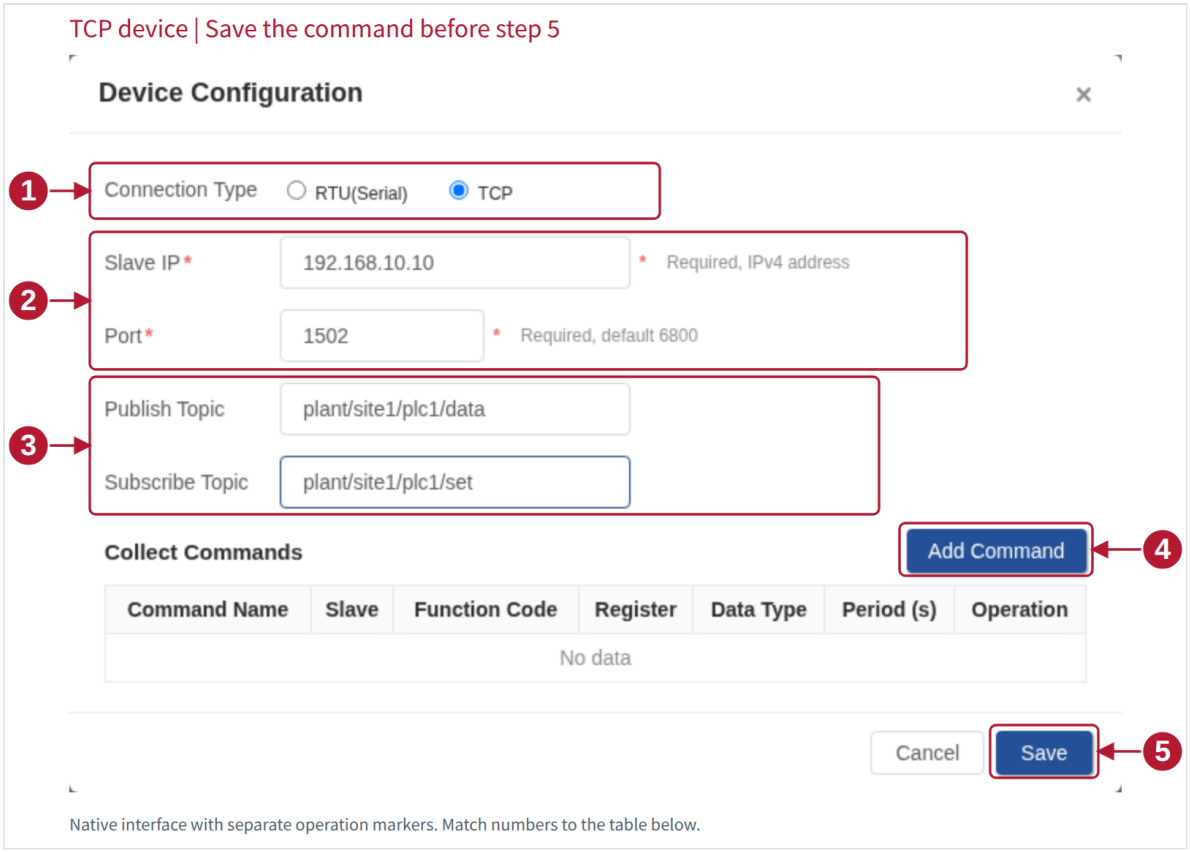


Figure 3-2. TCP device: complete ①–④ first. Return to ⑤ after saving the command in the next figure.

Order within this figure: ① TCP; ② slave IP/port; ③ publish/subscribe topics; ④ Add Command; ⑤ save the device after Figure 4-1.

Step	UI field	Lab value	Explanation / production value
①	RTU/TCP	TCP	The PC provides this data source over Ethernet.
②	TCP IP	192.168.10.10	Simulated meter address, not the gateway's management address.
②	TCP Port	1502	The supplied simulator listens here. A real device commonly uses 502; use its actual setting.
③	Publish topic	plant/site1/plc1/data	The gateway publishes readings here. The receiver subscribes to the identical topic.
③	Subscribe topic	plant/site1/plc1/set	Separate topic for downlink commands when required. Send no control commands in this exercise.
④	Command list	Add temperature in the next chapter	A connection alone does not specify what to read.
⑤	Save device	Click after saving the command	Finish the next figure, then return to this window.

Topics must match exactly, including case and slashes. Keep `data` and `set` separate, and avoid sharing control topics between unrelated devices.

Complete save sequence: Figure 3-2 ①–④ → Figure 4-1 ①–⑥, Save Command → return to Figure 3-2 ⑤, Save Device → Figure 5-1 ①–⑥, Save Global. Finish all three levels of configuration.

Check and troubleshoot this step | C05

Expected behavior: Save the command, then its device. The list shows the TCP slave and one command for this exercise. The ordinary Modbus list has no live value display; an entry proves configuration persistence only.

Pages/files and evidence to inspect: Save tcp_device.png and the device-list screenshot, showing slave IP, port 1502, command count and publish topic. Record both saves in 03_Step_Checks.

Observed problem	Checks in order	Condition to proceed
TCP row exists but command count is 0	Reopen the connection, add/save the command in Figure 4-1, then save the device. An empty connection cannot produce readings.	Count is 1; verify new reports after enabling global settings.
Connection disappears/settings revert after refresh	Check command then device saves. Correct red validation errors before refreshing/closing and reopen the same device to verify.	Reopened values match the intended configuration.
TCP acquisition mistaken for tcp reporting	TCP connection means reading a network slave; global mqtt means publishing results to a Broker. Select mqtt in Chapter 05 without changing verified slave settings.	TCP acquisition and MQTT reporting use their own host/port settings.

04 | Add a command and verify one value

4.1 Fill in this read-only point

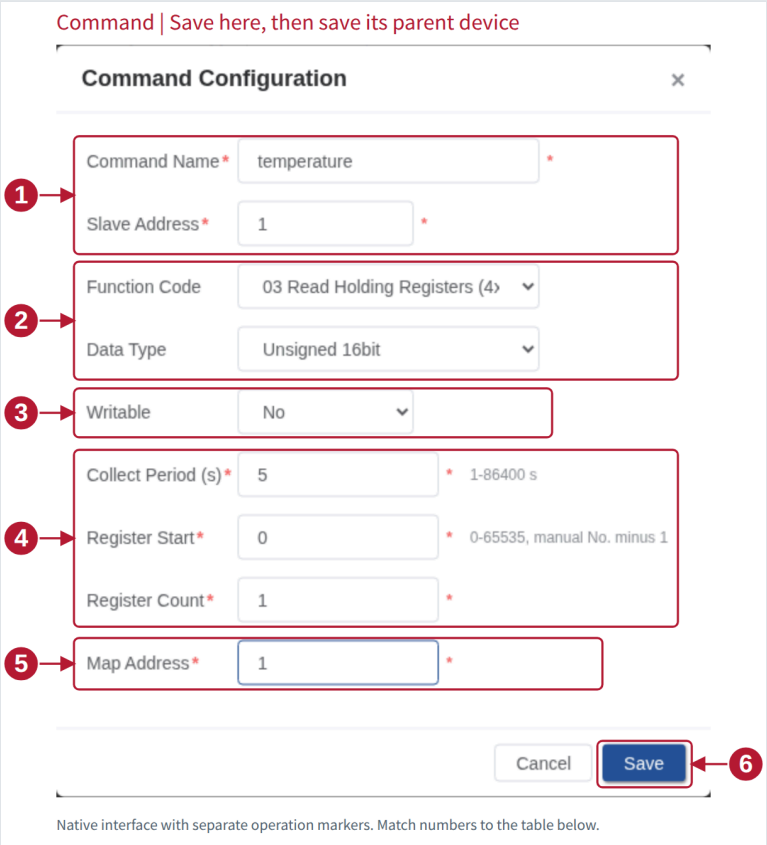


Figure 4-1. Acquisition command: complete ①–⑥ and save. Keep the first exercise read-only.

Order within this figure: ① name/slave ID; ② function/type; ③ Writable: No; ④ period/start/count; ⑤ map address; ⑥ Save Command, then return to Figure 3-2 ⑤.

Step	UI field	Lab value	What it means
①	Command name	temperature	A point identifier used as the payload key.
①	Slave address	1	Unit ID of the simulator. Use the real device's ID in production.
②	Function code	03 Read Holding Registers (4x)	Reads the simulator's holding-register area.
②	Data type	Unsigned 16bit	Also called uint16; this point occupies one register.
③	Writable	No	Read and verify first; perform no control writes.
④	Collect Period (s)	5	Polling interval for this command. Do not enter 5000.

Step	UI field	Lab value	What it means
④	Register start	0	Zero-based Modbus PDU address, not the display number 40001.
④	Register count	1	One 16-bit register.
⑤	Map address	1	A valid example for IEC104 mapping, not the MQTT key or Modbus start.
⑥	Save command	Click Save	Then save the parent device connection.

Click **Save** in the command window and check that the command appears in the device's command list. Click **Save** in the device window. Back on the main page, check the connection and command count, then save global reporting in Chapter 05. Saving only the innermost command is insufficient.

Check and troubleshoot this step | C06

Expected behavior: The read-only temperature command uses slave 1, FC03, start 0, count 1, Unsigned16bit and 5 s. It appears under the device after saving; verify new readings after global acquisition/reporting is configured.

Pages/files and evidence to inspect: Save modbus_command.png and record name, slave, function, PDU start, type, count and unit in 02_Point_Map. Lab raw value is 255.

Observed problem	Checks in order	Condition to proceed
Red field validation blocks Save	Check required fields, ranges and units. Command period 5 is seconds; the 1000 ms timeout belongs on the global page. Enter decimal start 0, not 40001 or 0x0000.	Command saves and reopened values match.
Tool reads 255 but gateway has no new messages	Verify command/device/global saves in 03_Step_Checks, then use C09/C10 to test reporting. The configuration list has no reading column.	A new temperature=255 arrives, or isolate acquisition versus reporting failure.
Considering Writable to fix failed reads	Keep Writable=No and check connection/function/address. Writable is not an acquisition-enable switch.	Read-only acquisition works without control writes.

4.2 Why enter 0 when the manual says 40001?

The gateway needs the address offset used in the Modbus request. Manufacturers use different numbering conventions; confirm the convention before converting it.

What the device manual explicitly defines	What to enter
PDU address 0 / zero-based offset 0	Enter 0; do not subtract 1 again.
One-based numbering: the first register is 1	The first location corresponds to start 0.
40001 explicitly means the first holding register	Select function 03 and enter start 0.
Only “address 100,” with no numbering convention	Ask the manufacturer first; do not assume subtraction by 1 or 40001.

An on-page note about subtracting 1 applies to the corresponding numbering convention, not to every address. Slave ID, function code and start address identify the device, data area and location respectively.

Check and troubleshoot this step | C07

Expected behavior: Identify the manufacturer reference convention and PDU offset, select the correct data area and verify a known point. Do not subtract 1 universally.

Pages/files and evidence to inspect: Keep manufacturer reference, function, actual PDU start and point-table version in 02_Point_Map, with the relevant manufacturer row.

Observed problem	Checks in order	Condition to proceed
Exception 01 or 02	Keep the exception. For 01 check supported function; for 02 check address/count range. A uses FC03/start 0; the SDM120M voltage example uses FC04/start 0.	The point reads without exception and matches a known value; scaling is not an address fix.
Only “address 100” or mixed hex notation	Ask whether it is a PDU offset or display reference, decimal or hexadecimal, and confirm length. Avoid bulk probing adjacent addresses.	Record the confirmed decimal PDU start and repeat the same read.

4.3 Select function, data type and scale

Point definition	Selection / processing	Common issue
Coil / discrete input	Use 01/02 as defined by the point table	Do not read it as a 03 holding register.
Signed / unsigned 16-bit integer	Signed 16bit / Unsigned 16bit	If a negative value becomes a large positive number, check the type.
32-bit integer	Matching Signed/Unsigned 32 ABCD	Uses two consecutive registers, not one.

Point definition	Selection / processing	Common issue
Floating point	Float ABCD or Float CDAB, per the point table	Wrong word order can give meaningless values; correct it before applying a scale.
Manufacturer specifies scale 0.1	Convert raw 255 to 25.5	If no scale field exists here, apply the conversion on your platform; do not label raw 255 as 255°C.

The reference Modbus page provides six types: Signed 16bit, Unsigned 16bit, Signed 32 ABCD, Unsigned 32 ABCD, Float ABCD and Float CDAB. If your required type is absent, send the point table to technical support to determine an appropriate conversion.

Add further commands after one known value is correct. Multiple consecutive registers may be reported as an array; parse the structure actually received.

Check and troubleshoot this step | C08

Expected behavior: Value, sign, word order, scale and unit match the field reference for the same time. A 32-bit point occupies two 16-bit registers; apply scaling once at the platform if absent from the gateway UI.

Pages/files and evidence to inspect: Record manufacturer raw/decoded values, actual report and field display in 04_Readings_Recovery. Keep word order and conversion location in 02_Point_Map.

Observed problem	Checks in order	Condition to proceed
Negative value becomes a large positive	Check Signed versus Unsigned and required register count. Correct the type rather than subtracting a constant to disguise the error.	Sign and known value match.
Float is huge, tiny or meaningless	Verify ABCD/CDAB, start and count 2 against the protocol. Compare one known point with the manufacturer tool before scaling.	Confirmed type/order consistently matches the field value.
Platform differs by 10 or 100	Compare raw payload first. Apply 0.1 only when the point table requires it; do not reuse the temperature integer scale for the meter Float voltage. Check duplicate conversion.	Raw value, engineering value and unit have a documented consistent relationship.

05 | Configure MQTT reporting and receive data

5.1 Complete global settings

Return to **Application > Modbus Collect**, Global Configuration. The figure shows an unsaved example. Enable the function and save it on your device using the table below.

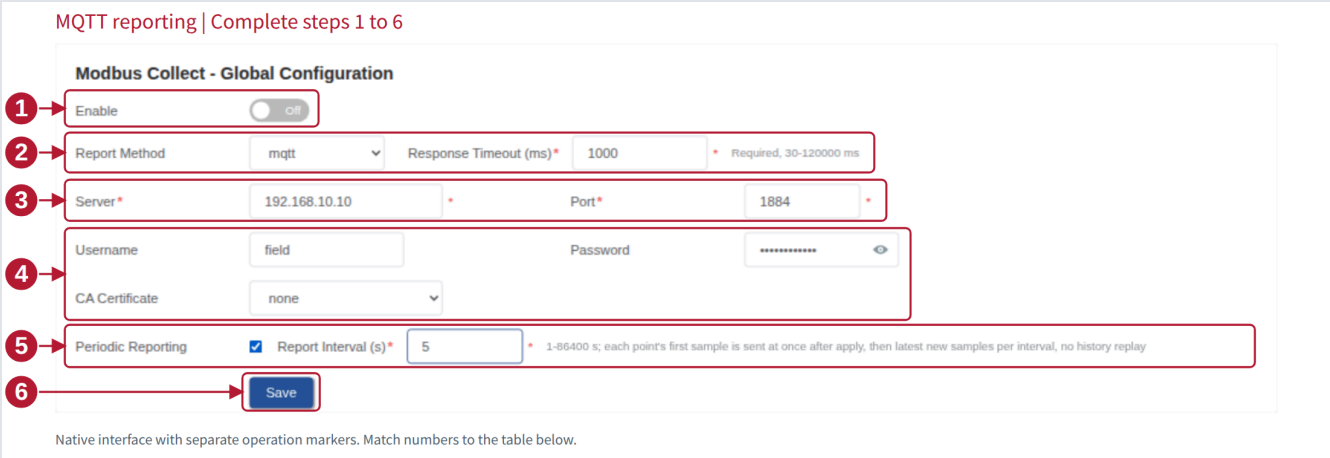


Figure 5-1. Global Modbus reporting: follow ①–⑥. Enable and save the settings on your gateway.

Order within this figure: ① enable; ② method/timeout; ③ Broker address/port; ④ credentials/CA; ⑤ reporting mode/period; ⑥ Save Global.

Step	UI field	Lab value	Explanation / production value
①	Enable	On	Save to activate acquisition; do not leave an unsaved draft.
②	Report method	mqtt	Sends readings to the test Broker.
②	Response timeout (ms)	1000	Wait time for one slave response: 1000 ms = 1 second. This is not the polling period.
③	Server	192.168.10.10	PC running the lab tools. Enter the host separately from the port; omit Web paths.
③	Port	1884	Test Broker port, not the Modbus port 1502.
④	Username	field	Test Broker account, not the gateway Web account.
④	Password	DemoOnly2026	Isolated lab password only. Use platform credentials in production.
④	CA certificate	none	No TLS in this isolated lab. Production settings must match the server and the gateway's supported configuration.
⑤	Periodic reporting	Checked	Displays the period field. If unchecked, reporting normally follows acquisition.
⑤	Report Interval (s)	5	Sending interval in seconds; separate from each command's polling interval.
⑥	Save Global	Click Save	Reopen/check settings, then verify new data at the receiver.

Confirm that the **command → device connection → global settings** have all been saved. Recheck that entries and enable states remain. A “saved” message alone does not verify reception.

Reporting after acquisition and periodic reporting are modes, not two duplicate reporting tasks. The current implementation may send the first valid sample immediately, then send new valid samples at the selected interval. Periodic publishing does not provide a strict real-time guarantee or replay all offline history.

Check and troubleshoot this step | C09

Expected behavior: Reopen global settings: enabled, mqtt, PC IP:1884 and lab credentials remain, with the device/command saved. Observe roughly 3–6 example cycles and inspect receiver/logs rather than a Save notification.

Pages/files and evidence to inspect: Save mqtt_global.png and current event_log.txt/system_log.txt. If application details are present, inspect mc_mqtt/mc_master and connection/read failures; retain new-connection evidence from lab_start.txt.

Observed problem	Checks in order	Condition to proceed
Disabled/waiting or no new acquisition after enable	Recheck enable, all save levels and at least one valid device/command. Inspect the collection program/Last Exception if present. If no row exists, record that and use C10.	Configuration is active and new-message validation can proceed, or preserve settings/logs for support.
Port 1884 timeout/refused	Keep the lab window running on the correct IP and allow 1884 for the lab subnet. Broker port is not slave port 1502. Check the gateway-to-PC path and service logs.	Gateway reaches the test Broker or a listening/firewall/path fault is isolated.
Lab Broker rejects authentication	Use the tool credentials field / DemoOnly2026, checking case/spaces. Do not substitute the admin Web account. CA is none for this isolated plain-MQTT lab.	Broker accepts the identity; verify new messages on the publish topic next.

5.2 View received messages on the PC

Keep **1_start_lab.cmd** running and double-click **3_watch_modbus.cmd**. It connects to the PC Broker on port 1884 and subscribes to `plant/site1/plc1/data`. Wait several polling cycles and inspect newly received raw messages. The tool shows `PASS` when it finds the expected value 255.

The actual payload depends on your gateway version/configuration. This example identifies the point to look for; its device identifier and timestamp are illustrative.

```
{"id":"DEMO_DEVICE","temperature":255,"ts":1700000000}
```

`temperature` matches the command name, and 255 is the simulator's raw reading. `id` identifies a device, while `ts` represents sampling time in Unix seconds. Do not require the example identifier or timestamp in your parser; additional fields may vary by version/configuration.

You may also use your own MQTT client with the same test server, port and credentials. Subscribe to the publish topic. Subscribing only to `.../set` will not show these reports. An old retained message is not a new report.

Check and troubleshoot this step | C10

Expected behavior: A new temperature=255 arrives on plant/site1/plc1/data and the tool reports PASS. Record receive time, effective topic and full JSON. Do not manually publish 255 to manufacture a pass.

Pages/files and evidence to inspect: Save watcher output to mqtt_messages.txt, including raw JSON rather than only the dashboard value. Check topic, temperature, ts and receive time.

Observed problem	Checks in order	Condition to proceed
No messages or wrong topic	Keep the lab service running and verify the same Broker/port. Copy the device publish topic exactly; subscribing to set or another project data topic will not show these reports.	A new message arrives on the exact topic; otherwise return to C09.
JSON arrives but no 255/PASS	Check the command name/case, array structure and whether the source is still the simulator. Compare the raw payload with 02_Point_Map rather than its last number.	A matches temperature=255; real equipment uses its own point definition/reference value.
Self-check reads 255 but subscriber stays empty	modbus_selfcheck.txt proves only the PC slave. Check the gateway-to-PC 1502 path, inbound firewall and read errors, then the MQTT connection.	Isolate the first failed stage with separate read/publish evidence.

5.3 Check all of these results

1. The simulator self-check reads 255, confirming that the data source works.
2. The gateway's connection and command are saved correctly, and acquisition is enabled.
3. A **new** message with temperature 255 arrives on the correct topic. A value left on screen after the source disconnects must not be treated as fresh acquisition.
4. For a real meter, compare the value, sign, scale and update time against its display or manufacturer tool for the same period.

For missing reports, use C10 here, then C09 to distinguish acquisition from publishing before changing registers.

Check and troubleshoot this step | C11

Expected behavior: Record several new messages and assess freshness from receive/sample time and source connectivity even if the value is constant. The first sample may send immediately; devices need not update on exactly the same second.

Pages/files and evidence to inspect: Use 04_Readings_Recovery for at least three receive times, ts values, readings and source states, plus a simultaneous field/manufacturer-tool screenshot.

Observed problem	Checks in order	Condition to proceed
A value appears immediately but never updates	Check retained data, wait several configured periods and record new receive times. In the isolated lab, observe after stopping the simulator; a value left on screen is not ongoing acquisition.	Distinguish stale display from fresh samples and record source loss.
Messages arrive but ts is wrong	Check gateway clock/time zone/synchronization in 2.5. Modbus ts is seconds; convert at the receiver if milliseconds are required. The sample timestamp is not real site time.	Sampling, receiving and field records can be correlated.
Broker connected but some points stop updating	Check each command/source and read errors by point name. Broker connectivity is not per-point health; compare a static field value with update time.	Freshness and source state are explained for each required point.

06 | Use case B: RS485 electrical metering to your platform

6.1 Check wiring and serial-port ownership

RS485 carries data; it is not a sensor power output. Wire the actual gateway and meter terminals according to their manuals and provide the correct meter power supply. Verify each device's A/B or polarity definitions instead of assuming that letters guarantee matching polarity. Follow equipment requirements for bus wiring, reference ground and termination.

Assign each physical serial port to one application. DTU, remote serial, Modbus, PLC acquisition and other applications must not open the same port simultaneously. For several RTU slaves on one bus, add commands with different slave IDs under **one serial connection** rather than creating competing connections for the same physical port.

When using a manufacturer serial tool on the PC: keep one master on the Modbus RTU bus. In the maintenance window, pause gateway polling of that bus or disconnect its communication path before the read-only PC test. Record time, close/disconnect the tool, then restore gateway polling. Simultaneous PC/gateway polling can cause collisions that resemble a meter fault.

Check and troubleshoot this step | C12

Expected behavior: Gateway and meter have their specified power supplies and verified RS485 terminals. One application owns the physical port. Ethernet/power lights do not prove RS485 communication.

Pages/files and evidence to inspect: Save wiring.jpg with both communication terminal labels visible. Record the actual Serial mapping and meter communication menu in 01_Project_Plan.

Observed problem	Checks in order	Condition to proceed
All RTU points time out with no reply	Check meter power, actual terminals/Serial mapping, A/B definitions, loose wiring and slave ID before register changes. Qualified site personnel handle mains wiring.	Obtain one valid read-only response before adding points.
Port busy/intermittent, or acquisition stops when DTU starts	Check DTU, PLC and remote-serial ownership of the same physical port. Record settings/impact before removing competing ownership.	One application owns the port and communication is stable; retain logs/settings.
Short cable works, field bus has parity/CRC errors	Verify baud/parity/stop bits, then equipment requirements for topology, shielding/reference ground and termination. Change one item at a time.	Repeated reads of the same known point are stable; keep before/after results.

6.2 Add an RTU connection

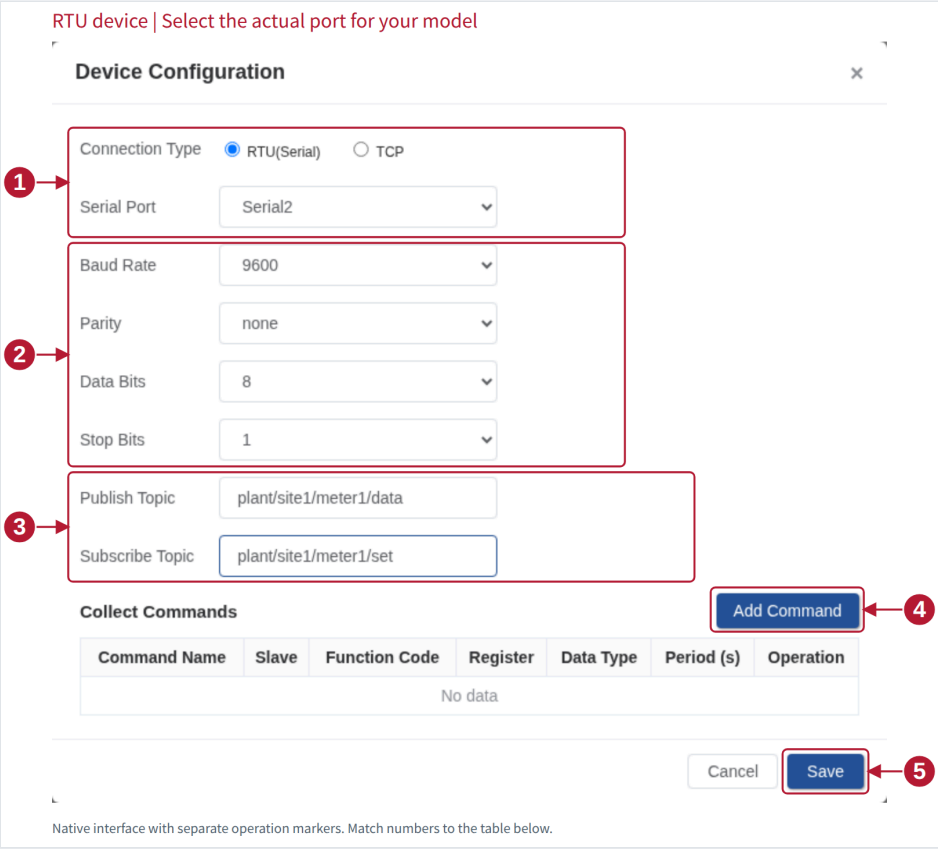


Figure 6-1. RTU connection: follow ①–⑤. Serial2 is the reference configuration; select your actual port using Chapter 02.

Order within this figure: ① RTU and actual port; ② meter serial settings; ③ meter topics; ④ add/save a read-only command; ⑤ return and save the device.

Step	UI field	Example meter	Production setting
①	RTU/TCP	RTU	Uses the physical serial port instead of a TCP slave IP.
①	Serial port	Select using Chapter 02	Figure: Serial2. YZW280 RS485 example: Serial1. Use actual terminals/configuration for other models.
①	Interface type, if shown	RS485	A fixed type may be hidden; software cannot create a missing electrical interface.
②	Baud rate	Reference 2400; use the actual menu	Copy the meter's setting; check its menu if unknown.
②	Parity	none	No parity. Use even/odd if required by the meter.
②	Data bits / stop bits	8 / 1	Must match the meter: the 8 and 1 in 8N1.
③	Publish topic	plant/site1/meter1/data	Separate from the TCP exercise to identify the source.
③	Subscribe topic	plant/site1/meter1/set	Separate downlink topic; send no control commands initially.
④	Add Command	One known read-only point	Use the real meter table, not the simulated 255 point.
⑤	Save device	Save after saving the command	Then check global settings and received results.

Use the real meter's slave ID, function code, start, count, type, word order and scale. The simulator's slave 1/start 0/value 255 does not define your meter. The global MQTT server still identifies the **receiving Broker**, even when the data source changes to RTU.

Add the use-case B read-only point below and save in order. Double-click **4_watch_rtu.cmd** to subscribe to `plant/site1/meter1/data`. Compare the message with the meter display before adding more points. This watcher does not require a fixed value of 255; use the actual meter reading.

Use case B | Verify one electrical-cabinet voltage point first. After wiring/settings are confirmed, retain the local test Broker to verify the meter before production integration. This is the matching SDM120M version's voltage point; use each other meter's own table. Select the actual gateway port from Chapter 02. Serial2/9600 in the figure illustrate the form; use the meter's real baud rate (the reference default is 2400).

Command field	Use case B entry/check	Verification basis
Command name	voltage	Find this key in raw reports, not the simulator's temperature.
Slave address	Example 1; use the meter menu	Record the actual ID and avoid duplicates on the bus.
Function code	04 Read Input Registers	The voltage point is an input parameter, unlike lab FC03.
Register start	0	Reference 30001 maps to PDU offset 0 for this table; do not enter 30001.
Register count	2	One 32-bit Float occupies two 16-bit registers.
Data type/order	Float; confirm ABCD against protocol/manufacturer tool	Correct order before applying a scale.
Writable / polling period	No / example 5 s	Read voltage without writing metering/reset settings.
Map address	Example 1 if required by the form	Not the MQTT key; IEC104 needs its own mapping plan.
Unit / scaling	V; actual point-table scale, applied once	Do not reuse the integer temperature scale of 0.1.

Save command, device and global MQTT in order. Use `4_watch_rtu.cmd` on `plant/site1/meter1/data`, without the simulator's fixed-255 pass condition. Compare simultaneous meter display and raw `voltage` before Chapter 08 production integration.

Allow for display resolution and sampling-time differences: extra payload decimals alone are not a decoding error. Use equipment/project tolerances. Observe the meter display alongside the gateway; test a manufacturer serial tool separately using the single-master sequence in the preceding section.

The point/settings reference is the manufacturer SDM120M V2.9 manual, communication settings and input-register table. Check separate documents for other suffixes/versions.

Check and troubleshoot this step | C13

Expected behavior: For B, start with voltage only: the SDM120M reference uses FC04, PDU 0, two registers and Float. Match actual serial settings and confirmed word order. Reported voltage should match the simultaneous meter V reading, not a fixed 230 or 255.

Pages/files and evidence to inspect: Save the meter communication menu, rtu_device.png, meter_command.png and mqtt_messages.txt. Record voltage/V/Float/order/scale/table version in 02_Point_Map and simultaneous readings in 04_Readings_Recovery.

Observed problem	Checks in order	Condition to proceed
Timeout after copying 9600/slave 1	Read actual settings from the meter. The reference SDM120M document lists default baud 2400; site settings may differ. Match baud/parity/stop bits/ID individually.	Gateway settings match this meter and the known voltage point replies.
Reply exists but voltage is wrong	Check for copied lab FC03/Unsigned16bit/count 1. For the verified voltage point use FC04/start 0/count 2 and confirmed Float order. Other meter models require their own table.	Decoded voltage matches the meter without an arbitrary 0.1 correction.
Broker receives temperature but no voltage	Check whether the subscriber/command still uses the TCP lab. Save the real meter connection/voltage command and inspect its publish topic. Reporting Server still points to the Broker.	A fresh real-meter voltage arrives and matches the display.

6.3 Example: two slave IDs on one bus

Suppose two meters use 9600/none/8/1 and slave IDs 1 and 2. Keep one serial connection and add commands `meter1_value` and `meter2_value` with their respective slave IDs. Set addresses and types from each meter's own table; they need not match. Avoid duplicate slave IDs on the bus.

Check and troubleshoot this step | C14

Expected behavior: Use one connection for the RS485 bus and distinct command names for slave IDs. Commands share that connection's publish topic; different IDs do not automatically create separate topics.

Pages/files and evidence to inspect: Map each physical meter to its slave settings/command in 02_Point_Map and record known values/update times individually in 04_Readings_Recovery.

Observed problem	Checks in order	Condition to proceed
Both become unstable after adding meter 2	Isolate meter 2 and verify meter 1 recovers. Check duplicate IDs, mismatched serial settings and wiring. Verify each independently before combining; keep one serial owner.	Both known values are stable with unique IDs.
One meter missing or values assigned to the wrong meter	Check each command ID and unique name against simultaneous displays. Map actual payload keys; duplicate keys can hide a second meter.	Each meter has an unambiguous point name and comparison record.
Updates slow after adding points	Record command count, periods, timeouts and failing slaves. Reduce to one known point and retest. Failed slaves may occupy bus timeout time; do not minimize every interval.	Add points gradually and record actual update times meeting project needs.

07 | Use case C: MT7981 PLC states and separate reporting

Skip this chapter for MT7628/MT7621. On MT7981, first look for the PLC Collect application. If it is absent, continue using the Modbus procedure where applicable. For other PLC protocols, send technical support the exact model and protocol information.

7.1 When to use PLC Collect

If your PLC uses a dedicated protocol available on the gateway, open **Application > PLC Collect**. This application is separate from Modbus: create a PLC data source, verify its points, then select that source in its own Data Reporting section. Modbus global MQTT settings do not automatically report PLC points.

Obtain the exact PLC CPU/series, protocol, access permission, network/serial settings and one point already verified with the manufacturer tool. Drivers and address syntax vary by PLC series. Start with a read-only point and leave cyclic writing disabled.

7.2 Add an acquisition connection

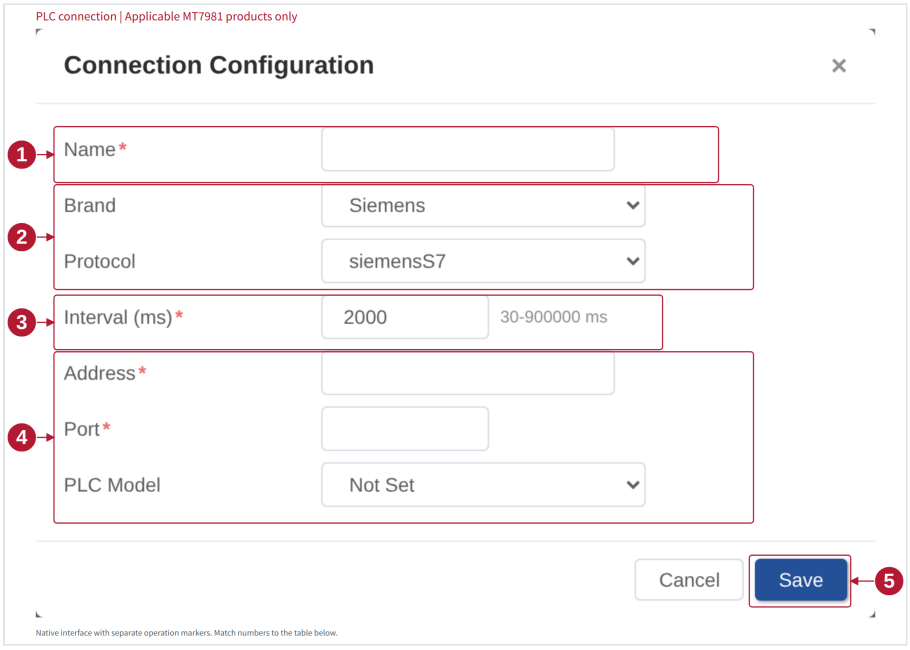


Figure 7-1. PLC connection on MT7981: complete ①–⑤. Save the connection before opening its point table.

Order within this figure: ① name; ② brand/protocol; ③ polling period in ms; ④ PLC address/port/model; ⑤ Save Connection, then open its point table.

Step	UI field	How to fill it in	What to check
①	Name	For example PLC_A, the data-source name	Select this same source later; it is not an MQTT topic.
②	Brand / protocol	Use the actual PLC series and engineering protocol	Match driver to CPU rather than brand alone.

Step	UI field	How to fill it in	What to check
③	Interval (ms)	Example: 2000 = 2 seconds	This field uses milliseconds; entering 5 does not mean the Modbus example's 5 seconds.
④	Address / port	Example IP 192.168.10.20; use the verified protocol port	Identifies the PLC, not the Broker. Do not assume every PLC uses 502 or 102.
④	PLC model, rack/slot, if shown	Copy actual CPU engineering settings	Fields vary by protocol; do not add settings absent from the page.
④	Serial settings, for a serial protocol	Match the physical port and PLC	Avoid sharing a port with Modbus, DTU or Node-RED.
⑤	Save Connection	Save after checking the settings	Confirm the source exists, then maintain its points.

192.168.10.20 is only an example address. Address syntax, port, model and rack/slot are protocol-specific. Copy verified settings from the manufacturer tool. If you only know the brand, obtain the engineering point table before configuring points.

Click **Save** in the connection window. Confirm PLC_A appears in the list, open that row's **Point Table**, then click **Add Point**. A connection must be saved before its points can be maintained.

Check and troubleshoot this step | C15

Expected behavior: The saved PLC connection appears with the intended name, driver, host and interval. For C, use one PLC whose access has been verified by its engineer. TCP success alone does not verify a point.

Pages/files and evidence to inspect: Save plc_connection.png and exact CPU/series, protocol, port and access requirements in 01_Project_Plan, plus a manufacturer-tool connection screenshot without passwords.

Observed problem	Checks in order	Condition to proceed
PLC application/driver absent	Check the applicable MT7981 model/menu. Use Modbus if supported by the source; otherwise supply model/version/protocol to support rather than assuming all drivers exist.	Proceed with an available matching application/driver.
Network works but PLC fails/reconnects	Verify the same IP/port/CPU/access in the manufacturer tool. Fill model/rack/slot only where provided and confirmed. Do not assume every PLC uses 502 or 102.	Use verified parameters and preserve the actual error.
Point Table requires a saved connection	Save and confirm the connection row before opening Point Table. Do not report from an unsaved temporary source.	Saved source exists and its point table opens.

7.3 Add one read-only point

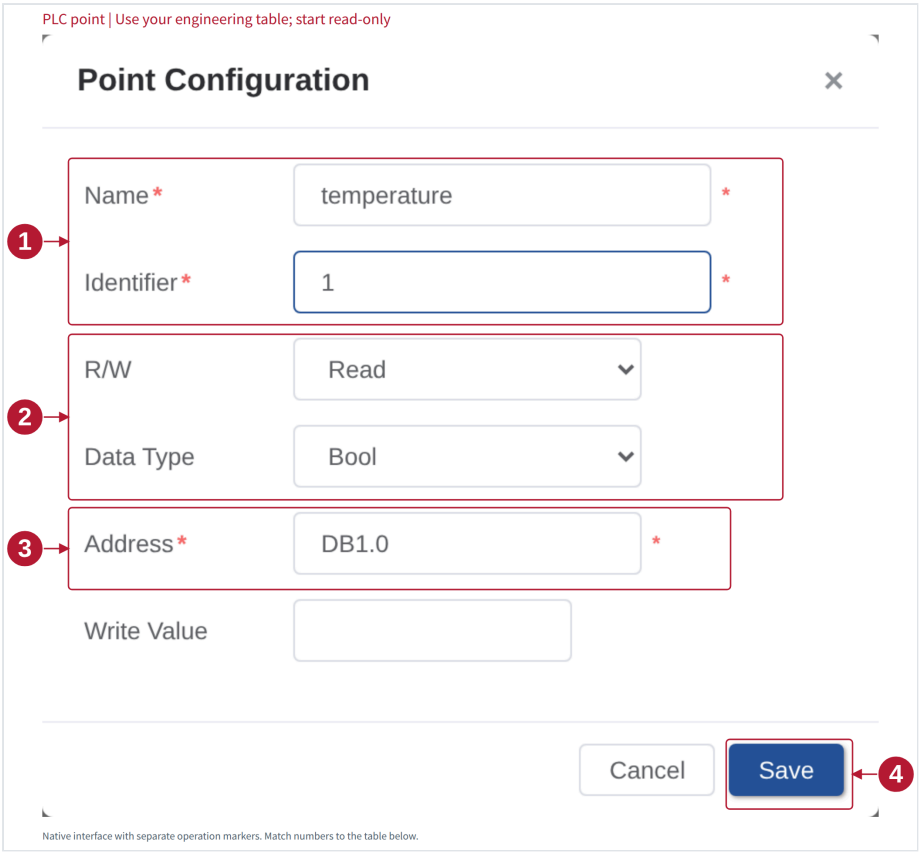


Figure 7-2. PLC point: follow ①–④. Bool and DB1.0 are native examples; use the type and full address from your engineering table.

Order within this figure: ① name/identifier; ② Read and the engineering data type; ③ full engineering address; ④ Save Point. A read-only point needs no control write value.

Step	UI field	Example / source	What to enter
①	Name	Use case C: pump_running	A readable name; avoid confusing duplicates.
①	Identifier	1	Identifies the point in reports; use a clear project convention.
②	Data type	Bool for C, with engineer-confirmed address	Bool, integers and floating point are not interchangeable.
②	R/W	Read	Choose Read from Read/Write/Read-Write for the first test.
②	Cyclic writing, if available	Off	This is not the polling interval. Configure no control action here.
③	Address	Exact address supplied by the PLC engineer	Preserve protocol syntax. Do not apply the Modbus 40001-to-0 rule.

Step	UI field	Example / source	What to enter
④	Save Point	Check and click Save	Close the point table and verify the value in Monitor.

Save the point and close its table. Check that the connection is enabled. Open that connection's **Monitor** and compare the value and updates with the manufacturer tool or field display for the same time period. A successful connection does not verify the address/type/length of every point.

The reference PLC page offers Bool, Byte, Int16, UInt16, Int32, UInt32, Int64, Float and Double. These differ from the six Modbus types. Match the chosen protocol's supported type to the PLC point definition.

Add a few points at a time. Use the gateway's own CSV template for import/export where available. Connection/point capacity does not guarantee polling speed.

Use case C entries: at Figure 7-2 ① change the example temperature name to pump_running; at ② select Read and the confirmed Bool type; at ③ use the real running-bit address. DB1.0 is not a universal PLC address. Do not display Bool as °C. Observe normal operation without control writes.

Check and troubleshoot this step | C16

Expected behavior: For C, use pump_running, a unique identifier, Read and the running bit's Bool type. Compare simultaneous run/stop states in Monitor and the PLC tool. temperature in the native figure is an example name, not this case's temperature value.

Pages/files and evidence to inspect: Save plc_point.png/plc_monitor.png, record the actual address, Bool, Read and identifier in 02_Point_Map, and simultaneous states in 04_Readings_Recovery.

Observed problem	Checks in order	Condition to proceed
Connection exists, no Monitor value/address error	Verify the bit, protocol access and full driver-specific address. DB1.0 is illustrative; Modbus subtraction does not apply.	Monitor reads the engineer-confirmed point.
Value disagrees with actual running state	Compare the same time in the PLC tool; check connection, identifier, bit offset and type. Do not force writes or start equipment merely to test.	Both agree during normal operation while read-only.
Monitor correct, platform always Running	Use C21 with Monitor evidence. Values may be strings; do not treat "0" as true merely because it is nonempty. Map actual encoding.	Platform state matches the field.

7.4 Configure PLC Data Reporting

In **Application > PLC Collect > Data Reporting**, click **+ Add**. If no data source can be selected, create and save the acquisition connection first. Expand **Advanced Settings** to enter the server, credentials and topics, then save the individual rule. Enable state and source selection are essential and separate

from Modbus global settings.

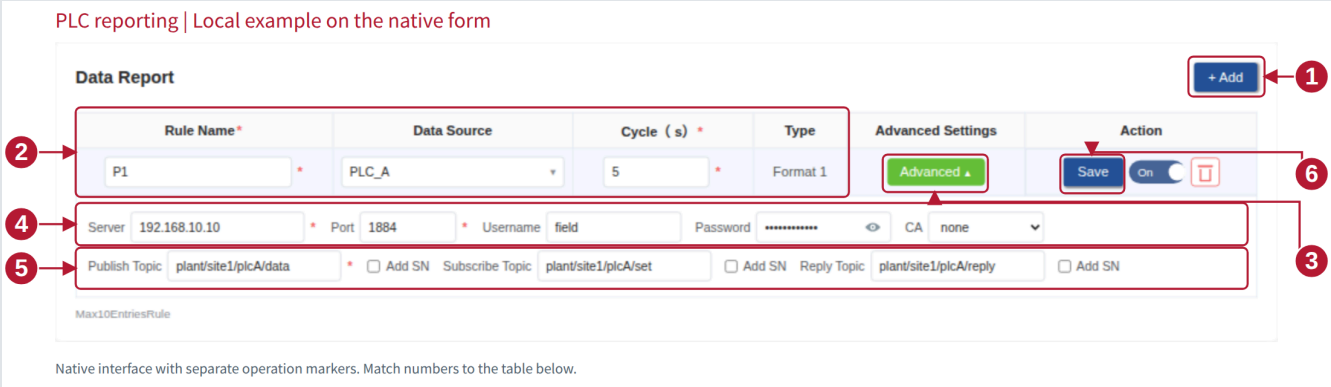


Figure 7-3. Native PLC reporting page with local example values. PLC_A demonstrates the form; save and verify your real source before configuring reporting.

Order within this figure: ① Add rule; ② name/source/period; ③ Advanced Settings; ④ server/credentials/CA; ⑤ topics and suffixes; ⑥ confirm On and save the rule.

Step	UI field	Example	What it does
①	Add rule	Click + Add	First save and verify the data source.
②	Rule name	P1	Distinguishes this reporting rule.
②	Data source	PLC_A	Select the connection verified in Monitor.
②	Cycle (s)	5	Separate from the acquisition connection's period in ms.
②	Format	Reference page displays Format 1	No selection is needed if no dropdown is provided. Follow your actual page for other versions.
③	Advanced Settings	Expand	Complete ④ and ⑤ before ⑥.
④	Server / port	192.168.10.10 / 1884	Reuses the PC's test Broker.
④	Username / password	field / DemoOnly2026	Lab only; replace for production.
④	CA certificate	none, local lab only	Match actual TLS/certificate requirements in production.
⑤	Publish topic	plant/site1/plcA/data	Subscribe here to receive this example's data.
⑤	Subscribe topic	plant/site1/plcA/set	PLC downlink topic; send no control commands in this exercise.
⑤	Reply topic	plant/site1/plcA/reply	Separate from periodic acquisition reports.
⑤	SN suffix	Initially disable for each topic	If enabled later, use the complete effective topic including the real suffix.

Step	UI field	Example	What it does
⑥	Enable	On, then save	An existing row is not necessarily active.
⑥	Save rule	Confirm On; click the blue Save button	Avoid the red Delete button. Verify new messages on the actual topic.

Replace PLC_A with your saved connection and verify its point value in Monitor first. After saving the reporting rule, double-click **5_watch_plc.cmd**, or subscribe to `plant/site1/plcA/data` with an MQTT client. You may also run this in the tool directory:

```
python site_lab.py watch --host 192.168.10.10 --topic plant/site1/plcA/data --seconds 60
```

Find the corresponding source, identifier 1 or point name in the raw payload and compare it with Monitor. **PLC Format 1 contains fields such as deviceId, deviceName, params and status, and point values may be strings.** Use a parser matching this structure rather than the flat Modbus numeric payload. Reporting status does not prove a control command was executed.

Check and troubleshoot this step | C17

Expected behavior: PLC_A reads the bit correctly in Monitor. Enabled P1 selects that source with 5 s period; its effective data topic receives a new identifiable payload.

Pages/files and evidence to inspect: Save plc_report.png, plc_monitor.png, mqtt_messages.txt and Event/System logs. Match source, effective topic/SN and hierarchy.

Observed problem	Checks in order	Condition to proceed
No source or selection is empty	Save/verify acquisition first, then add/select the reporting source. Close the selector and verify the name.	Rule shows the intended saved source.
PLC Monitor has data, Broker receives none	Check P1 enable/source/period and its own server/credentials/save. Modbus global MQTT does not configure PLC reports. Verify the selected source.	PLC rule publishes fresh Monitor-matched data.
Nothing on plcA/data, messages elsewhere	Check SN suffixes including double suffixing and subscribe to the complete effective publish topic. Reply is not periodic data.	Fresh messages arrive on the effective topic.

7.5 Additional checks

- Monitor has a value but the platform does not: check the reporting rule's enable state, source, effective publish topic and SN suffix, then Broker authentication.

- For Modbus-only acquisition, do not create a duplicate PLC connection owning the same serial port.
- Reporting to an external Broker does not require enabling the gateway's local MQTT Broker. The Broker, acquisition client and subscribing client have distinct roles.
- The basic procedure needs no Node-RED. Prefer processing payloads at your platform receiver when possible. For processing on the gateway, first check its actual Node-RED entry and required nodes/resources.

08 | Connect data to your platform

8.1 Identify the platform's ingestion endpoint

This chapter focuses on **MQTT ingestion**. The complete path is **meter/PLC → gateway acquisition → your MQTT Broker → platform receiver → database/dashboard**. The Broker receives and forwards messages. Your platform must also subscribe, parse fields and associate them with a device before values appear.

A platform's Web login URL, HTTP API endpoint and MQTT endpoint serve different purposes. Ask your platform administrator for the **MQTT device connection parameters**. If the platform only offers HTTP/HTTPS APIs, determine an adapter method using Section 8.7 instead of entering a Web/API URL in the MQTT server field.

Complete the local exercise in Chapters 03–05 first, or use one verified real point. Before changing reporting settings, export the current configuration from **System Management > Configuration Management**. Record the original server, topics and intervals so that you can restore them if needed.

8.2 Obtain this information from your administrator

Parameter	Information to obtain	Where it is used
Broker host and port	Domain/IP reachable from the gateway; actual MQTT listening port	Server/Port in Modbus global settings or a PLC reporting rule.
Connection transport	Plain MQTT TCP or MQTT over TLS; any other transport requirement	Match the gateway's available connection/certificate settings; do not infer transport from a port alone.
Authentication	Username, password and expiry; any additional Client ID or signing requirement	Enter credentials in the matching fields; check extra requirements using Section 8.7.
Certificates	CA file and how to obtain it; whether client certificate/private key are required	Import/select CA using the gateway's certificate procedure; verify mutual TLS separately.

Parameter	Information to obtain	Where it is used
Publish topic and permissions	Complete topic the gateway may publish to; topics the platform receiver may subscribe to	Device connection's publish topic or PLC rule's publish topic.
Downlink/reply topics	Separate command/reply topics and permissions if control is required	Enter separately. Send no control messages in a read-only project.
Device identification	How the platform identifies gateways/points using topic, identifier or SN	Platform device registration and field mapping; also check PLC SN suffixes.
Data requirements	One expected payload, field types, units, scale, time format and reporting interval	Configure parsing in Section 8.5 and check format compatibility.
Network conditions	Public, LAN or VPN server; DNS, source allowlist and routing requirements	Gateway Internet path, DNS, VPN or server access rules.

These are **MQTT credentials, not gateway Web login credentials**. The gateway and platform receiver may use separate accounts, with publish and subscribe permissions respectively. Transfer passwords, private keys and tokens using your project's agreed secure method.

Check and troubleshoot this step | C18

Expected behavior: Administrator supplies a reachable MQTT host/transport/port/identity, topic permissions and payload/device mapping requirements, separate from a Web login URL.

Pages/files and evidence to inspect: Use 01_Project_Plan for redacted parameters and platform_integration.txt for specification version, mandatory fields and confirmed Client ID/signing/mutual-TLS settings.

Observed problem	Checks in order	Condition to proceed
Only Web/HTTP endpoint supplied	Ask for an MQTT device endpoint and TCP/TLS port. Use 8.7 for HTTP/WebSocket-only adaptation, not a URL in Server.	Obtain a compatible real endpoint.
Required identity/signing/key field missing	Compare requirements with the actual page; ask support about gaps/model/version. Do not substitute unrelated username/CA/topic fields.	Confirm the method before production migration.
Broker on a private LAN/VPN	Verify gateway-to-private-network routing/VPN/DNS. Cellular addressing and an office PC connection do not prove the gateway path.	Validate the gateway path and Broker connection arrival.

8.3 Enter platform parameters in the right application

Keep the verified acquisition connection and read-only commands, and change reporting parameters first. **The meter address still identifies the meter; the reporting server becomes your Broker.** Do not replace both addresses with the platform host.

The following values only illustrate field mapping. `mqtt.example.com` is a documentation example, not a working project endpoint. Replace it with the real host supplied by your administrator. Port 1883 is only a plain-MQTT example.

Platform parameter / example	Modbus Collect: location	PLC Collect: location, applicable MT7981
MQTT protocol	Global Configuration > Report method: mqtt	The reference PLC reporting rule uses MQTT.
Host mqtt.example.com	Global Configuration > Server	Data Reporting > rule > Advanced Settings > Server.
Port 1883, plain-MQTT example only	Global Configuration > Port	Rule > Advanced Settings > Port.
Platform username/password	Global Configuration > Username/Password	Rule > Advanced Settings > Username/Password.
CA, when TLS is required	Prepare the CA using the gateway's procedure, then select it in global settings	Prepare the CA, then select it in the rule's advanced settings.
factory/site01/gateway01 /telemetry	Device connection > Publish topic	Rule > Publish topic; also check the SN suffix.
factory/site01/gateway01/ command	Device connection > Subscribe topic	Rule > Subscribe topic; keep the first test read-only.
factory/site01/gateway01/ reply, if needed	Do not assume the Modbus page has the PLC reply field	Rule > Reply topic; keep separate from periodic data.
Reporting interval 10 s, example	Check periodic reporting and enter 10	Rule > Cycle (s): 10.
Verified source PLC_A	Modbus uses commands under the device connection	Rule > Data source: your saved connection.

For Modbus: change that device's topics at Figure 3-2 ③ and save the device at ⑤. Set the platform host, credentials and interval at Figure 5-1 ③④⑤; confirm Enable at ① and save global settings at ⑥. Changing only the global server does not change old device topics.

For PLC Collect: select a valid source and interval at Figure 7-3 ②. Expand ③, enter the platform server/credentials at ④, then topics at ⑤. You can initially leave SN suffixes off. If enabling a suffix to distinguish devices, record the complete actual publish topic and subscribe to it on the platform. At ⑥

confirm the rule is On and save it.

Enter only the real hostname or IP in Server, without `mqtt://`, `https://`, a port or a Web path. Enter the port separately. For CA upload on the Modbus page, scroll to **MQTT CA Certificate**, click **Upload Certificate**, select the CA file supplied by your platform, and follow the on-page prompts. After the file appears in the list, select it in the global CA dropdown and save. For a PLC rule, use the certificate entry available on your device; if the required file is absent, send the current page and certificate requirements to technical support.

For TLS, check the gateway time first, obtain the correct CA from your administrator, and import/select it using the device's certificate procedure. **Changing 1883 to 8883 alone does not enable TLS.** Match the server's requirements to the gateway's actual supported configuration method. Do not bypass a production certificate error by disabling certificate validation.

Check and troubleshoot this step | C19

Expected behavior: Reopened settings match the platform sheet and Broker logs identify the gateway. TLS hostname/port/CA/time match; plain and TLS ports are separate.

Pages/files and evidence to inspect: Save production_mqtt.png, gateway_network.txt and matching logs. Obtain redacted Broker connection/auth logs and record CA name/source/validity without exposing keys.

Observed problem	Checks in order	Condition to proceed
Timeout/Connection refused	Check bare hostname/port and gateway path/DNS. Administrator checks listener/access rules and connection arrival. Preserve verified meter settings.	Isolate the specific network/service failure and retest.
Certificate/TLS configuration fails	Check clock, certificate hostname, CA chain and TLS port; reopen the saved CA selection. Client authentication needs its own supported entry; a private key is not a CA.	Correct credentials/certificates validate without bypass.
Authentication rejected/repeated disconnects	Inspect credentials, expiry, authorization and duplicate identities at the failure time. Verify MQTT versus Web accounts; ask support about identity when Client ID is absent.	Stable connection, then verify publishing in C20.

8.4 Inspect raw messages before the dashboard

Ask your administrator to use an existing MQTT client or the platform's receiver logs in this order. The supplied local lab tool is for the isolated exercise, not a production TLS client.

1. **Connect to the same Broker.** Enter the real host, port, transport and subscriber credentials. Confirm the connection; preserve the full error text if it fails.
2. **Subscribe to the effective publish topic.** Copy the complete gateway topic and check case, slashes, device hierarchy and SN suffix. Start with one exact topic. Obtain authorization before

using wildcard subscriptions.

3. **Wait for new samples.** Observe several polling/reporting intervals and record receive time, topic and raw payload. Compare natural field changes, or a controlled known value in an isolated lab. Do not count an old retained message as a successful new report.
4. **Match the source.** Compare with gateway Monitor, the manufacturer tool or the meter display. Confirm the device, point, raw value and time.
5. **Configure platform parsing.** Give this real payload to your administrator. Use Section 8.5 to associate the device, extract fields and configure units/time, then check dashboard updates.

If raw messages reach the Broker while the dashboard is empty, check platform subscriptions, device mapping and parser logs first. Keep the gateway's already verified register settings.

Check and troubleshoot this step | C20

Expected behavior: An authorized subscriber on the same Broker receives fresh data on the exact topic at the relevant time. This proves reception, not dashboard parsing.

Pages/files and evidence to inspect: mqtt_messages.txt includes Broker identity, full topic, receive time and JSON, with authorization/publish-denial evidence where available and no passwords.

Observed problem	Checks in order	Condition to proceed
Connected but no reports	Check source/enable/saves, then distinguish CONNECT from PUBLISH authorization. Inspect actual topic ACL.	Broker accepts new publishes, or return to the source step.
Administrator sees data, subscriber does not	Compare Broker/tenant/permissions/full topic/SN and use a distinct subscriber identity. Start with an authorized exact topic.	Subscriber receives the same raw message.
Only old data/lab tool fails against TLS	Wait for fresh receive times and verify the source. Use an appropriate production TLS client; the lab watcher is for local plain MQTT.	New data arrives through the intended transport/client.

8.5 Turn raw payloads into displayed points

The Modbus example from Chapter 05 is:

```
{"id":"DEMO_DEVICE","temperature":255,"ts":1700000000}
```

This illustrates the structure only. The identifier and time are examples; use a new payload from your own gateway. Configure the platform as follows.

Mapping item	How to handle the Modbus example	How to verify it
Gateway/device identity	Associate the actual topic and device identifier; whether id is present depends on the payload	Different gateways do not overwrite one device's values.
Point key	Read temperature, derived from the acquisition command name	Match case exactly; update parsing when renaming a command.
Value type	This example is numeric 255; multiple registers may form arrays and digital points may be Boolean	Match the received type rather than forcing every point to one integer.
Scale and unit	Only if the real point table specifies scale 0.1: 255 × 0.1 = 25.5	Compare with the field reading; use °C only if the point represents temperature.
Time	ts uses Unix seconds; convert at the receiver if the platform requires milliseconds	Check units, time zone and gateway time. Do not use the sample timestamp.
Freshness	Consider sampling time together with platform receive time	A value remaining after a source disconnects is not marked as a fresh normal sample.

Use separate parsing for PLC reporting. Reference Format 1 contains fields such as `deviceId`, `deviceName`, `params` and `status`; point values may be strings. Obtain one real PLC payload and extract data according to its source, point identifiers and hierarchy. Do not reuse the flat `temperature` path above. A reporting status does not prove that a control command executed.

Apply scaling at one defined stage to avoid multiplying again after a gateway/receiver adapter has already converted the value. If your platform requires a fixed JSON hierarchy, field names or a dedicated device model, complete the conversion in Section 8.7 before dashboard mapping.

Check and troubleshoot this step | C21

Expected behavior: A real message maps to the correct device/point with field-consistent value/state, unit and time; parse each application's actual hierarchy/types.

Pages/files and evidence to inspect: Save mqtt_messages.txt, platform_parser.txt, platform_dashboard.png and 04_Readings_Recovery; document key/path/type/scale/conversion stage in 02_Point_Map.

Observed problem	Checks in order	Condition to proceed
Broker receives data, dashboard empty	Inspect subscriptions, registration/topic mapping, parser and ingestion logs. PLC params hierarchy is not flat Modbus temperature.	The same message reaches the correct device and update time advances.

Observed problem	Checks in order	Condition to proceed
Running bit always true/Float wrong/array ignored	Inspect actual Boolean/number/string/array types. String "0" is not true merely because nonempty. Handle the real layout rather than forcing integers.	Converted value/state matches the source.
Engineering value/time wrong	Check raw value/unit, single scaling, seconds/milliseconds and time zone; record gateway/receive/dashboard times together.	Values, units and times correlate.

8.6 Plan topics for multiple devices

Give each gateway a clear, unique device level, such as `factory/site01/gateway01/telemetry` and `factory/site01/gateway02/telemetry`. For multiple meters on one gateway, add meter-specific levels if required by your platform registration scheme.

MQTT topics are case-sensitive. Enter an actual topic in a publish field; do not use the subscription wildcards `+` or `#` there. In a platform subscription such as `factory/site01/+/telemetry`, `+` matches one level. Account permissions still apply. If PLC SN suffixes are enabled, inspect the actual published topic to determine the hierarchy and separator instead of assuming how the suffix is appended.

Plan data, command and reply topics separately. Verify read-only reporting first. For remote writing, commission the relevant application's command format and device permissions as a separate task.

Check and troubleshoot this step | C22

Expected behavior: Each gateway/meter is uniquely identified by topics, command names or actual identifiers, with matching permissions and separated control topics.

Pages/files and evidence to inspect: Record effective topics/SN rules per gateway in 01_Project_Plan and keep a raw message from each.

Observed problem	Checks in order	Condition to proceed
Device 2 overwrites/disconnects device 1	Separate topic/device mapping conflicts from duplicate client identity using platform/Broker logs.	Both operate simultaneously with correct mapping.
Exact topic works, wildcard fails	Count actual levels including SN; + matches one level and needs permission. Publish topics contain no + or #.	Authorized subscription matches the real hierarchy.

8.7 Resolve platform requirements absent from the current page

Platform requirement	Recommended action
HTTP/HTTPS API or WebSocket endpoint only	Ask for an MQTT-over-TCP/TLS endpoint. If unavailable, deploy a platform-side MQTT receiver/API adapter. Do not enter an HTTP URL in the MQTT server field.

Platform requirement	Recommended action
Fixed Client ID, product key, dynamic signature or token refresh	Send the integration specification, exact model and current settings page to technical support to check configurable options. Do not substitute unrelated username/topic fields for missing settings.
Mutual TLS requiring client certificate and private key	Client authentication support is required in addition to CA trust. Ask technical support to identify the available configuration method for your version, then follow the platform specification.
Fixed JSON schema, device model or special time format	Add mapping/conversion in the platform receiver first. For gateway-side processing, verify the available applications and resources, then use the confirmed method.
Offline history replay, buffering or fixed latency	Specify the points, retention duration and recovery behavior needed, and ask technical support to check a concrete solution. Periodic publishing alone does not provide history storage/replay.

Keep field names, algorithms and schema visible in integration documents, but remove production passwords, private keys and tokens. Continue using verified local acquisition while the integration method is established, then replace production reporting parameters.

Check and troubleshoot this step | C23

Expected behavior: Requirements match available settings/the confirmed adapter; one real read-only point is received and parsed before expansion.

Pages/files and evidence to inspect: Record gaps, method and input/output payloads in platform_integration.txt with administrator-maintained adapter/platform logs.

Observed problem	Checks in order	Condition to proceed
MQTT connects, proprietary auth/model fails	List identity/signing/certificate/model requirements and verify an agreed adapter/delivered app rather than installing unknown sample software.	One real point is authenticated, received and parsed.
Adapter outputs data, platform still empty	Keep input/output JSON/errors and check identity/required fields/types/time using one fixed read-only point.	Trace the same point through adapter and platform.

8.8 Local Broker, other reporting methods and DTU

Local Broker: Application > MQTT Broker, or MQTT Service on some versions, provides a Broker on the gateway. The acquisition application is a client sending readings. You do not need to enable this local service when reporting to an existing remote Broker. If using the local Broker, configure its listening port and authentication first, then connect the acquisition client and platform client to that service. From a PC, enter a reachable gateway address, not the PC's own `127.0.0.1`. Listening address, access rules and account permissions must allow the connection.

Report method	What the platform provides	How to verify it
MQTT	Broker, credentials and topic permissions	Receive new raw messages on the effective publish topic, then parse them.
TCP	Continuously listening TCP receiver and parser	Check new payloads in receiver logs. This host is not the Modbus slave.
UDP	UDP receiver and parser	Check actual UDP receive records; a TCP port test is insufficient.
IEC104	Matching master, common address, information-object addresses and point types	Verify points/mapping at the master; map address is not the Modbus register start.

These are methods visible in the Modbus application; configure PLC reporting on its separate page. After changing transport, also verify the payload format expected by the receiver.

DTU: use your model's DTU function to forward a custom raw serial byte stream, and parse it at the receiver according to the device protocol. For named register values, use the acquisition procedure above. Only one application should own a physical serial port.

Check and troubleshoot this step | C24

Expected behavior: Receiver and transport agree. DTU forwards bytes; a polled meter does not start sending named register values merely because TCP connects.

Pages/files and evidence to inspect: Record acquisition/pass-through roles, protocol/ports and labelled raw receiver frames/payloads in project evidence.

Observed problem	Checks in order	Condition to proceed
DTU TCP connects, no named JSON	Use Modbus Collect for named points; a matching DTU peer polls/parses raw protocol with one serial owner.	Obtain the valid raw response or acquired point required by the design.
Port open, receiver rejects/no record	Check raw TCP sent to MQTT, UDP tested as TCP, and IEC104 role/mapping mismatch in receiver logs.	The matching service receives new data, then verify its meaning.

09 | Move to the site and verify recovery

9.1 Save a recoverable site baseline

After A passes, do not copy simulator IDs, registers and credentials into production. Save the lab results, export the current backup from **System Management > Configuration Management**, and record model, current version, export time and applications used. Keep the actual backup filename and record its controlled storage location.

Migrate in order: ① replace the source using the real point table and verify one point; ② change Broker, authentication and topics; ③ verify new raw messages; ④ compare the dashboard; ⑤ disable unused test connections and temporary rules. Update `Forms/03_Step_Checks_EN.csv` after each stage. At a failure, retain the last passing configuration and use the corresponding section's checks rather than changing everything at once.

Check and troubleshoot this step | C25

Expected behavior: Change one category at a time; verify source, production Broker and platform before removing the lab. Retain configuration and point/rule records.

Pages/files and evidence to inspect: Keep the actual backup and project/point/check records under access control, recording only credential storage location.

Observed problem	Checks in order	Condition to proceed
Migration fails with unclear cause	Restore the most recent documented setting and retest; do not roll back network, points and topics together.	Reproduce the last good state and first failing change.
Considering another device/version backup	Check model/configuration compatibility. Do not edit or assume cross-device backups; use the supported restore method or re-enter project records.	Verify restored settings and fresh messages.

9.2 Verify restart and network recovery in a maintenance window

Use the project's agreed maintenance window and avoid interrupting critical operation. Record the last good message time, value, topic and gateway state in `Forms/04_Readings_Recovery_EN.csv`. Restart using the model manual. For a network-loss test, interrupt only the agreed data path and retain power. Do not unplug a PoE-powered device's power/data cable just to simulate a network outage.

After recovery, check management login → saved commands/connections/rules → enabled acquisition → readable source → new Broker message → updated platform time/value. Record the first recovered message and elapsed time. This proves new sampling resumed, not that offline history was replayed. Verify any required buffering/history-replay solution separately.

Check and troubleshoot this step | C26

Expected behavior: Settings remain, sources read and fresh reporting resumes with elapsed recovery time recorded; current data does not establish history replay.

Pages/files and evidence to inspect: Record last pre-outage/first recovered messages, state, elapsed time and gaps in 04_Readings_Recovery with matching logs.

Observed problem	Checks in order	Condition to proceed
Management returns but business does not	Recheck saves/enable/source, then C09 or C17 → C20 → C21; power lights/login alone are insufficient.	Repair the first failed stage and observe new data.
Broker repeatedly reconnects	Check uplink/DNS/clock/identity logs; restore time if needed. Preserve the first error rather than repeatedly rebooting.	Stable connection and new data with recovery time recorded.
Current data returns, history does not	Check the separate buffering/replay solution; periodic reporting cannot recreate unstored data.	Current reporting works and gaps are recorded or agreed replay separately tested.

9.3 Submit evidence for the failed step

Start with the failed step record. Include the project plan, checkpoint ID, relevant screenshots, matching `event_log.txt` / `system_log.txt`, local read result, raw payload on the effective topic and platform parser log. Identify the last passing step and first failing step, and keep the changes attempted and recovery actions.

Your platform administrator supplies production Broker connection/authentication/publish logs and platform ingestion logs. Service directories and container layouts differ; this guide does not assign a universal internal server path. Remove passwords, tokens, private keys and sensitive point tables. Keep controlled originals and share redacted copies.

10 | Verify operation and keep recovery records

10.1 Complete acceptance using the recorded results

Check	Pass condition
Wiring and points	Actual interface used; no serial conflict; correct point table, type, word order and scale.
Saved settings	Commands, device connections and global/reporting rules saved and enabled as needed.
Broker reception	New messages on the effective topic; identifiers match the data source.
Platform display	Correct device mapping, fields, units and time; dashboard follows new samples.
Recovery	In an approved maintenance window, verify configuration and new reports after restart/network recovery; assess operational impact first.
Records and backup	Configuration exported; points, rules, topics and certificates recorded under access control.

Use Section 9.1 for backup and test cleanup; keep detailed step results in Forms.

10.2 Project parameter record

Record actual project values and store them under access control. Keep detailed step records in the Forms templates. Record where passwords are held, not the passwords themselves, in a shared handover sheet.

Item	Actual project value
Gateway model/current version	_____
Current LAN management address	_____
Acquisition application and connection type	_____
Actual serial settings or slave IP/port	_____
Slave ID, function, start, count, type, order and scale	Attach the point table with version/date.
Polling period/timeout/report interval and units	_____
Broker host, port and transport	_____
Controlled location of credentials, certificates and private keys	_____
Publish/downlink/reply topics and suffix rules	_____
PLC source and identifiers, if used	_____
Platform fields, scale, units and time conversion	_____
Last successful time/configuration and point-table backup location	_____