



Besenix Modbus Studio

Modbus TCP diagnostics and automatic TIA Portal V21 engineering

USER MANUAL

Product version 0.5.2
Document revision 1.1
12 September 2026

Connect. Validate. Generate. Compile.

besenix.com

Document information

Item	Value
Product	Besenix Modbus Studio
Application version	0.5.2
Document	User Manual
Revision	1.1 - expanded PLC code and polling-rate guidance
Target engineering system	Siemens TIA Portal V21
Generated instruction	MB_CLIENT V6.0
Supported transport	Modbus TCP
Publisher	Besenix
Website	https://besenix.com

Important safety notice

Besenix Modbus Studio is an engineering and diagnostic tool. A Modbus write can operate physical equipment immediately. Confirm the target device, address, data type, scaling and plant state before writing any value.

- Version 0.5.2 can write manually to configured Coil and Holding Register tags after an explicit confirmation dialog.
- Generated PLC communication is read-only. The generated MB_CLIENT scheduler does not create automatic write jobs.
- Direct TIA Portal import does not save the project and does not download to the PLC.
- Always review and compile generated blocks before saving or downloading a PLC project.
- Use offline tests or a controlled commissioning environment before connecting to operating machinery.

Trademark notice

Besenix is an independent engineering software product. Siemens, SIMATIC, TIA Portal and related product names are trademarks of Siemens AG. Besenix is not affiliated with or endorsed by Siemens AG.

Contents

Document information	2
Important safety notice	2
Trademark notice	2
Contents	3
1. Product overview	6
Core capabilities	6
Current scope	6
2. System requirements and compatibility	7
Compatibility boundaries	7
3. Installation and first launch	8
Option A - standalone executable	8
Option B - portable PowerShell package	8
Package contents	8
4. TIA Portal Openness authorization	9
What the button changes	9
If authorization still fails	9
5. Main window tour	10
Recommended working order	10
6. Projects and connection settings	11
Create, open and save	11
Connection fields	11
Connect and disconnect	11
7. Network diagnostics	12
Ping	12
Run network diagnostics	12
Common socket results	12
8. Tag configuration reference	13
9. Data types, register length and byte order	14
Byte and word order	14
How to identify the correct order	14
10. Scaling, offset and engineering values	15
Generated data block behavior	15
11. Reading, polling and manual writes	16
Read once	16

Start polling	16
Write value	16
12. TIA Portal V21 - Full Client workflow	17
Pre-import validation	17
13. Modbus address conversion	18
Examples	18
14. Generated PLC blocks and architecture	19
Actual TIA Portal instruction version	19
Execution model	19
Important: Studio Poll ms is not the PLC delay	19
15. Generated TIA Portal code explained	20
Required OB1 call	20
FC_Besenix_Modbus_Call	20
Connection configuration	20
Job selection and MB_CLIENT call	20
Delay timer - the PLC polling-rate control	21
DONE and ERROR handling	21
16. Generated data members	22
Communication controls and diagnostics	22
Per-tag members	22
17. Interfaceld, Connection ID and timing	23
PROFINET Interfaceld	23
Connection ID	23
Request delay and cycle time	23
Where to slow polling in TIA Portal	23
18. Manual SCL package	24
Import in TIA Portal	24
19. Troubleshooting	25
Actual TIA Portal diagnostic examples	26
20. Commissioning checklist	27
21. Project file example	28
22. Validation and self-tests	29
Covered offline checks	29
Release file fingerprints	29
23. Reference and support	30
Release history	30

Official platform references 30

Support information to include in a request 30

Besenix Modbus Studio 30

1. Product overview

Besenix Modbus Studio combines a Modbus TCP test client, protocol-level diagnostics and a TIA Portal V21 code generator in one desktop workspace. It is designed to reduce the repetitive work between proving a device register map and building a usable Siemens PLC communication structure.

Step	Engineering result
1. Connect	A reachable Modbus TCP device and confirmed network path
2. Validate	Verified function codes, offsets, types, byte order and scaling
3. Generate	Reviewed symbolic DB, MB_CLIENT scheduler and wrapper call
4. Compile	TIA Portal project ready for controlled online commissioning

Core capabilities

- Connect to a Modbus TCP server using a numeric IPv4 address or host name for desktop testing.
- Select Automatic routing or bind the TCP socket to a specific local IPv4 network adapter.
- Ping the device and run detailed local adapter, route, ICMP and TCP port diagnostics.
- Read Coils, Discrete Inputs, Holding Registers and Input Registers.
- Decode Bool, 16-bit, 32-bit and 64-bit integer and floating-point values.
- Apply ABCD, BADC, CDAB or DCBA byte and word order.
- Apply engineering scaling and offset while retaining the raw hexadecimal response.
- Save reusable .bmsproj device projects.
- Generate a symbolic data block, MB_CLIENT V6.0 scheduler, instance DB and OB1 call wrapper.
- Import generated blocks directly into an open TIA Portal V21 project through Openness or export an SCL package for manual import.

Current scope

Area	v0.5.2 status
Modbus TCP client	Supported
Modbus RTU	Not included
Desktop reads	FC01, FC02, FC03, FC04
Desktop writes	Single Coil and Holding Register writes with confirmation
Generated PLC reads	One sequential read job per selected verified tag
Generated PLC writes	Not generated
TIA Portal	V21 direct import and manual SCL package
MB_CLIENT	V6.0

2. System requirements and compatibility

Component	Requirement
Operating system	Windows 10 or Windows 11, 64-bit recommended
Application runtime	Standalone Besenix .exe or Windows PowerShell 5.1 package
Network	IPv4 access to a Modbus TCP server
Default TCP port	502, configurable
TIA direct import	TIA Portal V21 with TIA Portal Openness installed and enabled
Windows authorization	Current account in the local Siemens TIA Openness group
PLC target	S7-1200 or S7-1500 CPU/firmware compatible with MB_CLIENT V6.0
Permissions	Administrator approval only when adding the Windows user to the Openness group

Compatibility boundaries

- Direct import is locked to the TIA Portal V21 Openness assemblies.
- The generated TCON_IP_v4 connection requires a numeric IPv4 server address. Host names are valid for desktop testing but not for generated PLC code.
- PROFINET Interfaceld is hardware-specific and must be confirmed for the selected PLC or communication module.
- Connection ID must be unique inside the PLC and must be between 1 and 4095.
- The application does not configure the physical PLC Ethernet interface or download hardware settings.

The application can be used as a Modbus TCP diagnostic tool without TIA Portal. TIA Portal and Openness are required only for direct PLC import.

3. Installation and first launch

Option A - standalone executable

- 1 Download Besenix_Modbus_Studio_v0.5.2.exe from the official Besenix distribution page.
- 2 Place the file in a normal user folder, for example Documents\Besenix.
- 3 Start the executable. If Windows displays a security prompt, verify that the file came from the official Besenix source before continuing.
- 4 Confirm that the footer shows v0.5.2 | MB_CLIENT V6.0.

Option B - portable PowerShell package

- 1 Extract the entire ZIP package. Do not start it from inside the compressed archive.
- 2 Double-click Start_Besenix_Modbus_Studio.vbs.
- 3 If the window does not open, run Start_Besenix_Modbus_Studio_Debug.cmd and read the reported error.
- 4 Keep the Modules, Assets, Examples and Tests folders next to the main script.

Package contents

Path	Purpose
Besenix_Modbus_Studio.ps1	Main Windows PowerShell/WPF application
Modules\ModbusTcp.psm1	Modbus TCP transport, encoding and decoding
Modules\TiaDataBlock.psm1	Tag validation and data block package generation
Modules\TiaMbClient.psm1	MB_CLIENT V6.0 scheduler and SCL generation
Modules\TiaPortalV21.psm1	TIA Portal V21 Openness discovery and import
Examples\Demo_Device.bmsproj	Example connection and tag project
Tests\Run_Self_Tests.ps1	Offline conversion and generator tests

Keep the original download as a backup. A project file contains connection settings and tag definitions, but it does not contain PLC credentials.

4. TIA Portal Openness authorization

Direct import requires the current Windows account to be a member of the local Siemens TIA Openness group. Version 0.5.2 includes an always-visible authorization button in the application header.

- 1 Start Besenix Modbus Studio under the Windows account that will run TIA Portal.
- 2 Click AUTHORIZE WINDOWS USER.
- 3 Approve the Windows administrator prompt.
- 4 The application checks existing membership and adds the current account only when required.
- 5 Restart Windows if the account was newly added.
- 6 Open TIA Portal V21 and the required project after the restart.

What the button changes

- It identifies the current Windows user and SID.
- It runs one administrator-approved local group operation.
- It does not create a Windows account and does not change TIA project permissions.
- It reports success, existing membership or the exact failure in the Diagnostics panel.

If authorization still fails

Symptom	Action
Local group not found	Verify that TIA Portal Openness is installed for V21.
Access denied	Approve the administrator prompt or contact the PC administrator.
User added but import denied	Restart Windows, then restart both TIA Portal and Besenix.
No TIA project found	Open TIA Portal V21 and load a project containing a PLC.
Wrong TIA version	Use TIA Portal V21. The importer intentionally loads V21 assemblies.

Official Siemens setup reference: [TIA Portal Openness Getting Started and Demo V21](#).

5. Main window tour

The main window is divided into six functional areas. This manual intentionally uses only screenshots captured from the real application or TIA Portal; schematic UI reconstructions are not used.

Area	Purpose
Header	Application identity, authorization and connection state
Connection row	Target, adapter, port, Unit ID, timeout, Ping and Connect
Project toolbar	New, Open, Save, tag editing, read/write and TIA generation
Live Tag Monitor	Configuration, values, quality, raw bytes and per-tag status
Diagnostics	Network tests, Modbus TX/RX trace, exceptions and TIA messages
Footer	Project name, counters, application version and MB_CLIENT version

Recommended working order

Set the connection first, add or load tags, prove the data with Read once or polling, then open TIA: Full Client. This order ensures that only registers with known-good responses become PLC jobs.

6. Projects and connection settings

Create, open and save

- New clears the current configuration after confirming unsaved changes.
- Open loads a .bmsproj JSON project.
- Save writes connection settings, tag definitions and TIA generator metadata.
- An asterisk in the window title indicates unsaved changes.

Connection fields

Field	Typical value	Meaning
IP address / host	192.168.0.10	Desktop Modbus TCP server target
Local network adapter	Automatic	Route selection or explicit local socket binding
TCP port	502	Server listening port
Unit ID	1	Modbus unit identifier in every request
Timeout	2000 ms	TCP connect, send and receive timeout

Connect and disconnect

Connect opens one TCP socket using the selected local interface. The connection indicator and Diagnostics panel report the result. Disconnect closes the stream and stops polling.

A successful TCP connection proves only that the server port is reachable. Use Read once to verify the Modbus protocol, Unit ID, function code and address.

7. Network diagnostics

The diagnostic workflow is designed to distinguish routing, TCP transport and Modbus protocol failures instead of reporting only a generic timeout.

Ping

- Runs four ICMP echo attempts and reports each result.
- A failed ping does not prove that Modbus TCP is unavailable; ICMP may be blocked.
- Ping uses the Windows routing table. The Modbus socket can still be explicitly bound to the selected adapter.

Run network diagnostics

- 1 Lists all active local IPv4 adapters, subnet masks and gateways.
- 2 Checks whether the target belongs to a local subnet.
- 3 Reports the selected adapter and local source address.
- 4 Attempts ICMP ping when possible.
- 5 Opens a TCP socket to the configured host and port using the configured timeout.
- 6 Suggests the next action based on the socket error category.

Common socket results

Result	Likely meaning
Connection timed out	No response: routing, firewall, cabling, wrong IP or server offline
Connection refused	Device is reachable but no service is listening on the selected port
Network unreachable	No valid Windows route to the target network
Host unreachable	The selected local network cannot reach the device
TCP reachable, Modbus read fails	Check Unit ID, FC, zero-based address, permissions and requested length

8. Tag configuration reference

Column	Editable	Description
On	Yes	Includes the tag in reads, polling and TIA preview
Tag name	Yes	Human-readable name; normalized for TIA identifiers
Area	Yes	Coil, Discrete Input, Holding Register or Input Register
Address	Yes	Zero-based protocol offset, 0 to 65535
Data type	Yes	Bool, signed/unsigned integer or floating-point type
Order	Yes	ABCD, BADC, CDAB or DCBA
Scale	Yes	Multiplier applied to the decoded raw value
Offset	Yes	Value added after scaling
Poll ms	Yes	Per-tag interval, 100 to 3,600,000 ms
Access	Yes	Read or Read/Write where supported
Value	No	Engineering value after conversion
Unit	Yes	Engineering unit shown as metadata
Quality	No	Idle, Good or Bad
Raw	No	Received bytes in hexadecimal order
Updated	No	Time of the last read attempt
Comment	Yes	Engineering description exported to the TIA map
Status	No	Per-tag success or detailed failure message

Coil and Discrete Input automatically use Bool. Discrete Input and Input Register automatically use Read access.

9. Data types, register length and byte order

Studio type	TIA type	Modbus words	Value class
Bool	Bool	1 bit job	Boolean
Int16	Int	1	Signed 16-bit
UInt16	UInt	1	Unsigned 16-bit
Int32	DInt	2	Signed 32-bit
UInt32	UDInt	2	Unsigned 32-bit
Float32	Real	2	IEEE 754 single
Int64	LInt	4	Signed 64-bit
UInt64	ULInt	4	Unsigned 64-bit
Float64	LReal	4	IEEE 754 double

Byte and word order

Order	32-bit byte sequence	Use
ABCD	A B C D	Network/register order unchanged
BADC	B A D C	Swap bytes inside every 16-bit word
CDAB	C D A B	Swap 16-bit words
DCBA	D C B A	Reverse all bytes

How to identify the correct order

- 1 Read a register whose real-world value is known.
- 2 Select the documented data type.
- 3 Compare ABCD, BADC, CDAB and DCBA while watching both Value and Raw.
- 4 Keep the order that produces the correct stable engineering value.
- 5 Document the chosen order in the tag comment before TIA generation.

10. Scaling, offset and engineering values

For numeric tags, Besenix calculates the displayed engineering value with:

$$\text{EngineeringValue} = \text{RawValue} * \text{Scale} + \text{Offset}$$

Raw	Scale	Offset	Result	Example
14500	0.1	0	1450.0	Motor speed in rpm
723	0.1	0	72.3	Temperature in deg C
500	0.01	-5	0.0	Calibrated process value
1.25	1	0	1.25	Float value unchanged

Generated data block behavior

- Unscaled numeric values keep their native TIA type for the engineering member.
- When Scale is not 1 or Offset is not 0, the engineering member is generated as LReal.
- The raw value, engineering value and QualityGood flag are separate symbolic members.
- Scale and offset are written into generated SCL conversion logic, not calculated only in the desktop application.

Use a period as the decimal separator in project values for maximum portability. Version 0.5.2 also validates numeric input against the current Windows culture.

11. Reading, polling and manual writes

Read once

Read once processes enabled tags through the active Modbus TCP connection and updates Value, Quality, Raw, Updated and Status. Use it before TIA generation to prove the register map.

Start polling

- The UI timer checks tags every 100 ms and reads one eligible tag per cycle.
- Each tag uses its own Poll ms interval.
- Polling stops when the connection is closed.
- Use longer intervals for slow devices or large tag sets.

Write value

Manual writes are available only when the selected tag is configured as Read/Write and belongs to Coil or Holding Register.

Target	Desktop Modbus function	Input
Single Coil	FC05	True/False, On/Off or 1/0
Single Holding Register word	FC06	Converted numeric value
Multiple Holding Register words	FC16	Converted 32-bit or 64-bit value

Before transmission, the application displays the exact target area and zero-based address and requires Yes/No confirmation. After a successful write, it reads the tag again.

12. TIA Portal V21 - Full Client workflow

The Full Client dialog is the review boundary between a proven desktop register map and generated PLC code. Verify every value in this dialog before import.

- 1 Connect to the device and verify the required tags until Quality is Good.
- 2 Open TIA Portal V21 and load the destination project.
- 3 Click TIA: Full Client in Besenix.
- 4 Click Refresh PLCs and select the required PLC software container.
- 5 Review the normalized TIA names, FC, offsets, types, lengths, byte order and validation messages.
- 6 Keep Generate MB_CLIENT V6.0 + scheduler enabled.
- 7 Enter the correct PROFINET Interfaceld and a unique Connection ID.
- 8 Keep overwrite disabled on the first import.
- 9 Click Import to TIA Portal V21.
- 10 Add FC_Besenix_Modbus_Call to OB1, compile, review and test.

Pre-import validation

- Invalid or duplicate normalized TIA names are excluded.
- Overlapping register ranges are reported.
- Coils and Discrete Inputs must use Bool.
- Tags without a successful Good read are not selected by default.
- An existing block is not overwritten unless the checkbox and separate confirmation are both accepted.

13. Modbus address conversion

The Live Tag Monitor uses zero-based protocol offsets. The generated MB_CLIENT block converts each offset to the Siemens MB_DATA_ADDR convention.

Studio area	Function	Studio offset 0	Generated MB_DATA_ADDR
Coil	FC01	0	1
Discrete Input	FC02	0	10001
Input Register	FC04	0	30001
Holding Register	FC03	0	40001

Examples

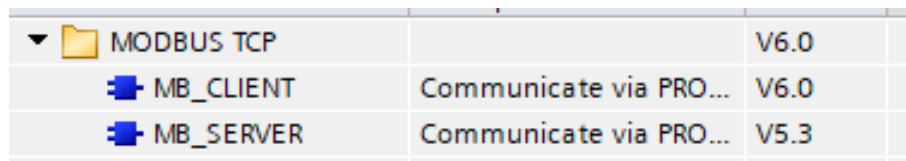
Vendor notation	Enter in Studio	Generated address
Holding Register 40001	Area: Holding Register, Address: 0	40001
Holding Register 40011	Area: Holding Register, Address: 10	40011
Input Register 30021	Area: Input Register, Address: 20	30021
Coil 00001	Area: Coil, Address: 0	1

Vendor manuals are inconsistent: some list zero-based offsets, others display 40001-style reference numbers. Confirm the convention before entering addresses.

14. Generated PLC blocks and architecture

Generated block	Purpose
DB_Besenix_Modbus_Data	Connection controls, diagnostics, receive buffers, raw values, engineering values and quality flags
FB_Besenix_Modbus_Client	MB_CLIENT V6.0 instance, TCON_IP_v4 settings, safe sequential scheduler and value decoding
DB_Besenix_Modbus_Client	Non-retain instance DB for the generated client FB
FC_Besenix_Modbus_Call	Stable zero-parameter cyclic wrapper called from OB1

Actual TIA Portal instruction version



MODBUS TCP	V6.0
MB_CLIENT	Communicate via PRO... V6.0
MB_SERVER	Communicate via PRO... V5.3

Figure 1. Actual TIA Portal V21 instruction catalog showing MB_CLIENT V6.0. The generator is locked to this library version.

Execution model

- One read job is generated for every selected tag.
- All jobs run sequentially through one MB_CLIENT instance.
- The default delay between completed jobs is T#100ms. For slower devices, increase Besenix_Comm_RequestDelay in DB_Besenix_Modbus_Data.
- Address, length and receive pointer are changed only when the client is not Busy.
- Reset is latched and applied safely after the active request completes.
- A successful job decodes its buffer, applies scaling and sets QualityGood.
- A failed job clears that tag's QualityGood and increments ErrorCount.

Important: Studio Poll ms is not the PLC delay

The Poll ms column controls only desktop polling inside Besenix Modbus Studio. Version 0.5.2 does not transfer each tag's Poll ms value into the generated PLC scheduler. In the PLC, all generated jobs use the single global DB value Besenix_Comm_RequestDelay.

The generated block names are fixed except for the main data block name. Fixed client names make the required OB1 call predictable.

15. Generated TIA Portal code explained

Required OB1 call

After import, call the generated wrapper once every PLC cycle:

```
"FC_Besenix_Modbus_Call"();
```

Do not place this call behind a slow pulse or call it only once per second. MB_CLIENT and the generated state machine must execute cyclically. Communication speed is controlled by Besenix_Comm_RequestDelay, not by slowing the OB1 call.

FC_Besenix_Modbus_Call

The wrapper calls the client instance DB and automatically returns the Reset command to FALSE. This keeps OB1 clean and turns Besenix_Comm_Reset into a one-cycle command.

```
"DB_Besenix_Modbus_Client"(
  Enable := "DB_Besenix_Modbus_Data".Besenix_Comm_Enable,
  Reset  := "DB_Besenix_Modbus_Data".Besenix_Comm_Reset);
"DB_Besenix_Modbus_Data".Besenix_Comm_Reset := FALSE;
```

Connection configuration

```
#Connection.InterfaceId := 64;
#Connection.ID := 1;
#Connection.ConnectionType := BYTE#16#0B;
#Connection.ActiveEstablished := TRUE;
#Connection.RemoteAddress.ADDR[1] := BYTE#192;
#Connection.RemoteAddress.ADDR[2] := BYTE#168;
#Connection.RemoteAddress.ADDR[3] := BYTE#0;
#Connection.RemoteAddress.ADDR[4] := BYTE#10;
#Connection.RemotePort := UINT#502;
#Client.MB_Unit_ID := BYTE#1;
#Client.RCV_TIMEOUT := 2.0;
```

Code item	Meaning
InterfaceId	Hardware ID of the Ethernet interface used by the PLC
ID	Unique connection ID inside this PLC
RemoteAddress.ADDR	Four IPv4 octets of the Modbus TCP server
RemotePort	TCP server port, normally 502
MB_Unit_ID	Modbus Unit ID sent with each request
RCV_TIMEOUT	Maximum wait for one response; this is not the polling interval

Job selection and MB_CLIENT call

JobIndex selects one generated CASE branch. Each branch contains the address, length and dedicated receive buffer for exactly one tested tag.

```

CASE #JobIndex OF
  0:
    #Client(
      REQ := #Request,
      DISCONNECT := NOT #Enable,
      MB_MODE := USINT#0,
      MB_DATA_ADDR := UDINT#40001,
      MB_DATA_LEN := UINT#1,
      DONE => #DoneInternal,
      BUSY => #BusyInternal,
      ERROR => #ErrorInternal,
      STATUS => #StatusInternal,
      MB_DATA_PTR := "DB_Besenix_Modbus_Data".MotorSpeed_Buffer,
      CONNECT := #Connection);
END_CASE;

```

Delay timer - the PLC polling-rate control

After DONE or ERROR, the scheduler sets Waiting to TRUE. The following TON holds the scheduler before the next job is released:

```

#DelayTimer(
  IN := #Waiting,
  PT := "DB_Besenix_Modbus_Data".Besenix_Comm_RequestDelay);

IF #Waiting AND #DelayTimer.Q THEN
  #Waiting := FALSE;
  #ReleaseCycle := TRUE;
END_IF;

```

Therefore, the setting to increase when a device is polled too quickly is DB_Besenix_Modbus_Data.Besenix_Comm_RequestDelay. The generated default is T#100ms.

DONE and ERROR handling

- DONE decodes the active buffer, applies byte order and scaling, sets the tag QualityGood flag and increments CompletedJobs.
- ERROR clears the active tag QualityGood flag and increments ErrorCount.
- Both outcomes lower Request, start Waiting and advance JobIndex to the next tag.
- After the last tag, JobIndex returns to zero and a new full scan begins.

16. Generated data members

Communication controls and diagnostics

Member	Type	Meaning
Besenix_Comm_Enable	Bool	Master enable; initial value TRUE
Besenix_Comm_Reset	Bool	Pulse TRUE to reset scheduler, counters and quality
Besenix_Comm_RequestDelay	Time	Delay between jobs; initial value T#100ms
Besenix_Comm_Connected	Bool	MB_CLIENT connection state
Besenix_Comm_Done	Bool	One-cycle successful request indication
Besenix_Comm_Busy	Bool	Request currently active
Besenix_Comm_Error	Bool	One-cycle request error indication
Besenix_Comm_Status	Word	Siemens MB_CLIENT status code
Besenix_Comm_ActiveJob	UInt	Zero-based current generated job index
Besenix_Comm_CompletedJobs	UDInt	Successful request counter
Besenix_Comm_ErrorCount	UDInt	Failed request counter

Per-tag members

```

MotorSpeed_Buffer : Array[0..0] of Word;
MotorSpeed_Raw : UInt;
MotorSpeed : LReal; // Raw * 0.1 + 0
MotorSpeed_QualityGood : Bool;

```

Bit areas use an Array[0..0] of Bool receive buffer. Numeric areas use one, two or four Word elements based on the selected type.

17. Interfaceld, Connection ID and timing

PROFINET Interfaceld

Interfaceld is the hardware identifier of the CPU Ethernet interface or selected CM/CP interface used by MB_CLIENT. The default value 64 is common but is not universal.

- 1 Open Device configuration in TIA Portal.
- 2 Select the Ethernet interface used for the Modbus TCP network.
- 3 Open its properties or system constants and locate the hardware identifier.
- 4 Enter that decimal value in the Full Client dialog.

Connection ID

Use a value from 1 to 4095 that is not used by another TCON, MB_CLIENT or related connection in the same PLC. For a second device, generate a separate client design with another ID and unique block names in a future multi-device workflow.

Request delay and cycle time

Approximate full scan time is the sum of response times plus the configured inter-request delay for every job. PLC scan transitions add a small additional overhead.

$$\text{FullScanTime} \approx \text{JobCount} * (\text{AverageResponseTime} + \text{RequestDelay})$$

Jobs	Response	Delay	Approx. scan
10	20 ms	100 ms	1.2 s
30	20 ms	100 ms	3.6 s
30	20 ms	500 ms	15.6 s
50	40 ms	100 ms	7.0 s

Where to slow polling in TIA Portal

- 1 Open PLC > Program blocks > DB_Besenix_Modbus_Data.
- 2 Open the DB in Data view and find Besenix_Comm_RequestDelay.
- 3 Change the Start value from T#100ms to a safer initial value such as T#500ms or T#1s.
- 4 Compile the software and download the changed DB value to the PLC.
- 5 For an online test, monitor the DB and modify the Actual value. Also update the Start value if the setting must survive a DB reinitialization or future download.
- 6 Watch Besenix_Comm_Status, Error, ErrorCount, CompletedJobs and ActiveJob while commissioning.

Recommended first test: T#1s for a slow or unknown device. When communication remains stable, reduce it gradually to T#500ms, T#250ms or the fastest interval permitted by the device documentation.

Do not change RCV_TIMEOUT to control polling speed. RCV_TIMEOUT defines how long MB_CLIENT waits for a response. Do not slow the FC call in OB1; change Besenix_Comm_RequestDelay instead.

18. Manual SCL package

Use Export SCL package when direct Openness import is unavailable or when the generated source must be reviewed before it enters a project.

File	Contents
DB_Besenix_Modbus_Data.scl	Data DB and, when enabled, the client FB, instance DB and call FC
DB_Besenix_Modbus_Data.tagmap.csv	Source-to-TIA name map, FC, address, type, length, order, scale and metadata
Besenix_TIA_Import.json	Machine-readable manifest with target version, connection and generated block list

Import in TIA Portal

- 1 Open the PLC tree and locate External source files.
- 2 Add the generated .scl file.
- 3 Choose Generate blocks from source.
- 4 Review the generated DB, FB, instance DB and FC.
- 5 Add the wrapper FC call to OB1.
- 6 Compile the PLC software and resolve every error before download.

Keep the CSV and JSON files with the project documentation. They record exactly how the tested desktop tags were translated into TIA objects.

19. Troubleshooting

Problem	Checks and corrective action
Ping fails	Confirm address and adapter. ICMP may be blocked; run TCP diagnostics before concluding that Modbus is unavailable.
Connect timeout	Check cable, subnet, route, firewall, server IP, TCP port and selected adapter.
Connection refused	Verify that Modbus TCP server mode is enabled and listening on the configured port.
Modbus exception 01	The device does not support the requested function code.
Modbus exception 02	The requested address or range is illegal for this device.
Modbus exception 03	The requested quantity or value is not accepted.
Value is incorrect	Check zero-based address, data type, register count, byte order, scale and offset.
Quality remains Bad	Open Diagnostics and inspect the exact per-tag protocol or conversion error.
No PLC in Refresh PLCs	Open TIA Portal V21, load a project with a PLC and confirm Openness authorization.
Openness access denied	Use AUTHORIZE WINDOWS USER and restart Windows after a new membership.
Invalid ARRAY tag in SCL	Use v0.5.1 or later; IPv4 octets must use RemoteAddress.ADDR[1..4].
MB_CLIENT will not compile	Confirm CPU support, MB_CLIENT V6.0 availability, TCON_IP_v4 version and hardware Interfaceld.
Block already exists	Keep overwrite disabled unless the existing generated block has been reviewed and backed up.
PLC polls too quickly / device errors	Open DB_Besenix_Modbus_Data and increase Besenix_Comm_RequestDelay from T#100ms to T#500ms or T#1s. Do not change the OB1 call rate.
PLC communicates slowly or times out	Confirm RCV_TIMEOUT, increase RequestDelay and remove unnecessary one-tag jobs.
Recover after repeated PLC errors	Set Enable FALSE, wait until Busy is FALSE, increase RequestDelay, pulse Reset TRUE once, then set Enable TRUE and monitor Status and ErrorCount.

Actual TIA Portal diagnostic examples

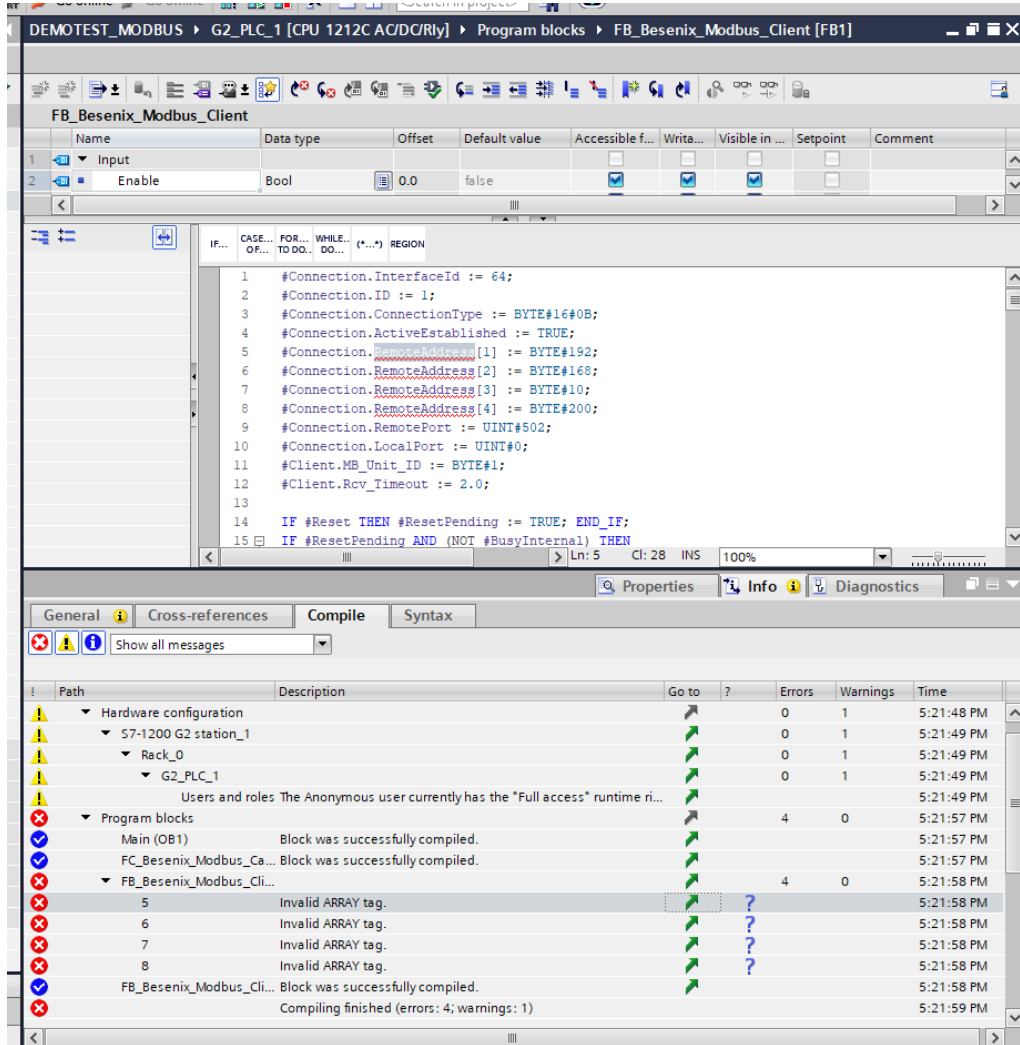


Figure 2. Actual TIA Portal compiler screen from the early v0.5.0 test. The four Invalid ARRAY tag errors identified direct RemoteAddress indexing. Since v0.5.1, the generator uses RemoteAddress.ADDR[1..4].

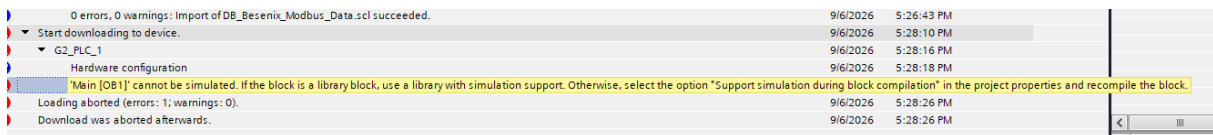


Figure 3. Actual PLCSIM download message. This concerns the project's simulation support setting, not Modbus addressing or polling speed.

20. Commissioning checklist

Stage	Required verification
Device	Server IP, TCP port, Unit ID and register documentation confirmed
PC network	Correct adapter, local IP, subnet and route confirmed
Desktop test	Every required tag has Good quality and credible value
Data model	Type, order, scale, offset, unit and comment documented
Write safety	Writable tags tested only under an approved safe plant condition
TIA target	Correct project and PLC selected
Connection	Correct InterfaceId and unique Connection ID
Preview	No errors, overlaps understood, normalized names accepted
Import	Generated block list reviewed before overwrite
Program	FC_Besenix_Modbus_Call added once to OB1
Compile	PLC software compiles without unresolved errors
Online test	Connected, Busy, Status, counters and QualityGood monitored
Handover	SCL, CSV map, JSON manifest and .bmsproj archived

Do not treat a successful import as a commissioning result. Commissioning is complete only after online PLC behavior and the physical process have been verified safely.

21. Project file example

A .bmsproj file is UTF-8 JSON. The format stores connection settings, tags and TIA generator metadata.

```
{
  "schema": "besenix.modbus-studio.project",
  "version": 1,
  "connection": {
    "transport": "ModbusTCP",
    "host": "192.168.0.10",
    "port": 502,
    "unitId": 1,
    "timeoutMs": 2000,
    "adapterName": "Automatic"
  },
  "tags": [{
    "enabled": true,
    "name": "MotorSpeed",
    "area": "Holding Register",
    "address": 0,
    "dataType": "UInt16",
    "byteOrder": "ABCD",
    "scale": 0.1,
    "offset": 0.0,
    "pollMs": 500,
    "access": "Read",
    "unit": "rpm",
    "comment": "Drive actual speed"
  }],
  "tiaMetadata": {
    "targetVersion": "V21",
    "mode": "FullMbClient",
    "mbClientVersion": "6.0"
  }
}
```

Edit project JSON manually only when necessary. Invalid values are rejected when tags are read or generated.

22. Validation and self-tests

The portable package contains Tests\Run_Self_Tests.ps1. Run it in Windows PowerShell 5.1 from an extracted package.

```
cd "C:\Path\To\Besenix-Modbus-Studio"
powershell.exe -NoProfile -ExecutionPolicy Bypass `
-File ".\Tests\Run_Self_Tests.ps1"
```

Covered offline checks

- Round-trip conversion for 16-bit, 32-bit and 64-bit numeric types.
- ABCD, BADC, CDAB and DCBA byte/word order.
- Modbus register count by data type.
- TIA identifier normalization and reserved word protection.
- Modbus offset to MB_DATA_ADDR conversion.
- Data block member generation.
- MB_CLIENT V6.0 declaration and RemoteAddress.ADDR syntax.
- Duplicate normalized TIA name detection.

Self-tests do not replace compiling the generated source in the exact target CPU and TIA Portal project.

Release file fingerprints

File	SHA-256
Besenix_Modbus_Studio_v0.5.2.exe	69b425aec15a59eb05ea33de3d69359026993eb82de5fbd6cf2af6dc7a96b326
Besenix-Modbus-Studio-v0.5.2.zip	b9f5b4d03e41d0168580bdd14b5da8e878831c7b31af16f21641b725792da9f3

23. Reference and support

Release history

Version	Key change
0.4.0	Tags + Data Block generation and TIA Portal V21 integration
0.5.0	MB_CLIENT V6.0, scheduler, buffers, conversion and diagnostics
0.5.1	Correct TCON_IP_v4 RemoteAddress.ADDR[1..4] syntax
0.5.2	AUTHORIZE WINDOWS USER workflow for TIA Openness

Official platform references

- Siemens TIA Portal Openness V21 Getting Started and Demo: support.industry.siemens.com
- Siemens MB_CLIENT V6.0 documentation: docs.tia.siemens.cloud

Support information to include in a request

- Besenix Modbus Studio version and Windows version.
- TIA Portal version, CPU order number and firmware version.
- Device type, server IP, port and Unit ID with sensitive data redacted if required.
- Selected adapter and the complete copied Diagnostics output.
- Tag area, zero-based offset, type, byte order and requested register count.
- Screenshot of the TIA compile error and generated block name.

Besenix Modbus Studio

Engineering diagnostics and repeatable TIA Portal integration.

Documentation and product information: besenix.com

Copyright (c) 2026 Besenix. All rights reserved. Distribution and license terms are governed by the license delivered with the software.