

FieldbusAnalyzer User Manual

Version: 1.0.0 Date: 2026-05-24 Target Platform: Windows 10 / 11
(x64) Runtime: .NET 8 (included)

1. Introduction

1.1 Application Overview

FieldbusAnalyzer is a Windows desktop application for configuring, monitoring, and analyzing industrial fieldbus protocols. The software enables reading and writing of process data, recording of measurement histories, monitoring with configurable alarm thresholds, and packet-level network analysis.

The application supports six protocols in a unified interface and provides MQTT integration for forwarding measured values to higher-level systems.

1.2 Supported Protocols

- Modbus TCP / RTU-over-TCP (RFC 3453, IEC 61158)
- CANopen via TCP gateway, serial gateway, or virtual CAN bus (CiA 301)
- EtherNet/IP with CIP (ODVA EtherNet/IP, IEC 61158 Type 2)
- PROFINET / PROFIBUS (IEC 61158, IEC 61784)
- OPC UA (OPC Unified Architecture, IEC 62541)
- BACnet/IP (ASHRAE 135, ISO 16484-5)

1.3 System Requirements

Component	Requirement
-----------	-------------

Operating System	Windows 10 or Windows 11 (64-bit)
Memory	Minimum 512 MB RAM (recommended: 2 GB)
Disk Space	Approx. 250 MB (incl. .NET 8 Runtime)
.NET Runtime	Included, self-contained - no separate installation
Network	TCP/IP connection to target device
Screen Resolution	Minimum 1280 x 720 pixels

2. Getting Started

2.1 Installation

FieldbusAnalyzer is delivered as a ZIP archive and requires no installation:

- Extract the ZIP archive to any directory (e.g., C:\Program Files\FieldbusAnalyzer).
- Launch the application by double-clicking FieldbusAnalyzer.App.exe.
- On first launch, configuration files are automatically created under %AppData%\FieldbusAnalyzer.

Uninstallation is not required - simply delete the directory.

2.2 Data Storage

Path	Content
%AppData%\FieldbusAnalyzer\fieldbus.db	SQLite database with recordings and alarms
%AppData%\FieldbusAnalyzer\prefs.json	User preferences (theme, MQTT, intervals)
%AppData%\FieldbusAnalyzer\modbus.json	Modbus device configurations

%AppData%\FieldbusAnalyzer\canopen.json	CANopen node configurations
%AppData%\FieldbusAnalyzer\eip.json	EtherNet/IP device configurations
%AppData%\FieldbusAnalyzer\profinet.json	PROFINET device configurations
%AppData%\FieldbusAnalyzer\opcua.json	OPC UA server configurations
%AppData%\FieldbusAnalyzer\bacnet.json	BACnet device configurations

2.3 Main Window

The main window is divided into three areas:

- Left Navigation: Protocol selection (Modbus, CANopen, EtherNet/IP, PROFINET, OPC UA, BACnet, Settings).
- Left Sidebar: Device/node list and property editor (protocol-specific).
- Right Main Area: Data tabs (upper - Registers/Tags/Modules/Nodes, Dashboard) and diagnostic tabs (lower - Tracer, Statistics, Log, Alarms), separated by a horizontal GridSplitter.

3. Settings

Settings are accessible via the 'Settings' tab. Changes are applied with the Save button.

3.1 Appearance

- System - follows Windows system settings (Light/Dark)
- Light - light theme
- Dark - dark theme (default)

3.2 MQTT Publisher

FieldbusAnalyzer can forward measured values to an MQTT broker during an active recording.

Parameter	Description	Default
MQTT enabled	Toggle the function on/off	No
Host	Hostname or IP address of the broker	localhost
Port	TCP port of the broker	1883
Topic Prefix	Prefix for all published topics	fieldbus

Topic format: {Prefix}/{DeviceName}/{TagName} Example:
fieldbus/Compressor1/Pressure

Values are only published during an active recording session (Rec) and only for successful reads.

3.3 General Settings

- Default Poll Interval (ms) - global default polling interval, default: 1000 ms
- CSV Separator - column delimiter for CSV exports, default: semicolon ;
- Database Path - read-only display of the SQLite database path. 'Open' reveals the folder in Explorer.
- Protocol Tracer enabled - globally enables/disables network capture for all protocols

3.4 Language

The user interface is available in German and English. The language can be changed at any time in the 'Settings' tab under 'Language' - no application restart is required.

- Deutsch - complete German user interface
- English - complete English user interface

4. Modbus Protocol

Modbus is the most widely used protocol in industrial automation. FieldbusAnalyzer supports Modbus TCP and RTU-over-TCP (Modbus RTU frames over TCP connection).

4.1 Device Configuration

Field	Description	Default
Name	Display name (required)	-
IP Address	IPv4 address of the Modbus server	-
Port	TCP port	502
Unit ID	Modbus device address (1-247)	1
Description	Optional free-text description	-
RTU over TCP	Enables RTU framing over TCP connection	No

4.2 Registers and Coils

Field	Description
Name	Display name of the data point
Function Code	FC01 Coils, FC02 Discrete, FC03 Holding, FC04 Input, FC05/06/15/16 Write
Address	Register address (0-65535)
Data Type	Boolean, Int16/32/64, UInt16/32, Float32/64, String, Raw
Count	Array size (default: 1)
Byte Order	Big-endian or Little-endian
Scale	Display value = (raw value x scale) + offset
Offset	Additive constant after scaling

Unit	Display unit, e.g. degC, bar, rpm
Warn Threshold	Limit for WARN alarm
Alarm Threshold	Limit for ALARM
Change Alarm	Alarm on any value change
Widget	Auto, Value, LED, Gauge, Chart
IP Override	Different IP/port for this data point
Write Value	Manually entered write value

4.3 Reading and Writing Data

- Read Once - reads all data points once
- Start Poll / Stop Poll - starts/stops cyclic polling
- Poll Interval (ms) - polling interval (50 to 60000 ms)
- On-Change Only - only changed values written to the database
- Write All - writes all data points with entered write values
- Confirm Write - read back value after each write and compare to target
- W button (per row) - writes only the selected data point

4.4 Dashboard

The 'Dashboard' tab shows all data points as visual cards. Widget types:

- Auto - automatic selection (LED for Boolean, Value for numbers)
- Value - numeric display value with unit
- LED - green/red display for Boolean values (On/Off)
- Gauge - circular instrument with configurable min/max range
- Chart - time series of the last 100 measurements (sparkline)

4.5 Modbus Slave Server (Test Mode)

The 'Slave' tab contains a built-in Modbus TCP server for testing purposes.

Parameter	Description	Default
Port	TCP listen port of the server	502
Unit ID	Device address	1
Delay (ms)	Artificial response delay to simulate slow devices	0
Force Exception	Server responds with error code for all requests	No

The slave server logs all write accesses in the event log.

5. CANopen Protocol

CANopen (CiA 301) is a communication protocol for embedded systems. FieldbusAnalyzer communicates via TCP gateway, serial gateway, or virtual CAN bus.

5.1 Node Configuration

Field	Description	Default
Name	Display name of the node	-
Node ID	CANopen device address (1-127)	-
IP Address	Gateway IP address	-
Port	Gateway TCP port	3000
Baud Rate	CAN bus baud rate (bps)	250000
Connection Type	TCP gateway, serial gateway, or virtual CAN	TCP Gateway
EDS File	Reference to Electronic Data Sheet	-

NMT State	Current NMT status (updated on read)	Unknown
-----------	--------------------------------------	---------

5.2 Object Dictionary

CANopen objects are addressed via Index (0x0000-0xFFFF) and Sub-Index (0-255).

- Index / Sub-Index - object address in the CANopen object dictionary
- Data Type - Unsigned8/16/32, Signed8/16/32, Float, Double, String
- Access Type - RO (read-only), WO (write-only), RW (read-write)
- PDO Mapping - flag as Process Data Object for cyclic data transfer
- Scale, offset, unit, thresholds

5.3 NMT Commands

- Start Remote Node (0x01) - puts node into operation
- Stop Remote Node (0x02) - stops the node
- Enter Pre-operational (0x80) - preparatory state
- Reset Communication (0x82) - resets communication

5.4 Node Scan

'Scan Nodes' button sends a discovery request and lists all reachable nodes with their device address and NMT status.

6. EtherNet/IP Protocol

EtherNet/IP uses the Common Industrial Protocol (CIP) from ODVA, commonly used in Rockwell/Allen-Bradley controllers.

6.1 Device Configuration

Field	Description	Default
Name	Display name of the device	-
IP Address	IPv4 address of the controller	-

Port	TCP port (CIP)	44818
Slot	Module slot for multi-slot devices	0

6.2 Tag Configuration

- CIP Path - format '1.1' (Port.Link) or '4.1.3' (Class.Instance.Attribute)
- Logix Tag Name - simple tag name for Allen-Bradley Logix controllers
- Service Code - GetAttributeAll, GetAttributeSingle, SetAttributeSingle
- Data Type - Dint, Int, Uint, Bool, Byte, Real, LReal, String
- IP override, scale, offset, unit, thresholds

6.3 Read Device Identity

The 'Read Identity' button queries vendor, model, serial number, and firmware version from the controller.

7. PROFINET Protocol

PROFINET (IEC 61158) is the standard fieldbus of PROFIBUS & PROFINET International (PI). FieldbusAnalyzer supports S7-TCP, PROFIBUS gateway, and PROFINET TCP connections.

7.1 Device Configuration

Field	Description	Default
Name	Display name	-
IP Address	IPv4 address of the controller	-
Port	TCP port (S7: 102)	102
Device Name	PROFINET NameOfStation	-
Vendor ID	VendorID (from GSDML)	-

Device ID	DeviceID (from GSDML)	-
Rack	Rack number (for S7 connection)	0
Slot	Slot number (for S7 connection)	1
Station Address	PROFIBUS station address (gateway)	-
Connection Type	S7 TCP / PROFIBUS Gateway / PROFINET TCP	S7 TCP
GSD File	Reference to device description file	-

7.2 Modules and Memory Areas

Area	Abbreviation	Description
Input	I / E	Process input image
Output	Q / A	Process output image
Flag / Marker	M	Flag registers
Timer	T	Timer registers
Counter	C / Z	Counter registers
Data Block	DB	Data block with number and byte offset
I/O Record	Rec	PROFINET I/O record (Slot/Subslot/Index)

Per module configurable: data type, byte address, bit address (for Bool), array size, scale, unit, thresholds.

7.3 GSDML Import

GSDML files (IEC 61784-2) describe PROFINET devices in XML format.
 Button: '↓ GSDML...'

- Device data (vendor, Vendor ID, Device ID) are automatically imported
- All I/O records (slot, subslot, index, data type, direction) created as modules
- New device immediately shown in the device list

7.4 GSD Import (PROFIBUS)

GSD files describe PROFIBUS devices in INI-like format. Button: '↓ GSD...'

- Model name and vendor read from the file
- Modules automatically detected and created
- Connection type automatically set to 'PROFIBUS Gateway'

8. OPC UA Protocol

OPC Unified Architecture (IEC 62541) is the platform-independent standard for industrial data communication. FieldbusAnalyzer uses the OPC Foundation .NET Standard SDK.

8.1 Server Configuration

Field	Description	Example
Name	Display name of the server	My OPC UA Server
Server URL	OPC UA endpoint address	opc.tcp://192.168.1.100:4840
Security	Sign & Encrypt (TLS encryption)	No
Username	Login name (empty = anonymous)	admin
Password	Login password	-

8.2 Nodes (NodeId Addressing)

OPC UA nodes are addressed via NodeId - two formats:

- Numeric ID: ns=2;i=1001 (namespace index + integer identifier)
- String ID: ns=2;s=Temperature (namespace index + string identifier)

Data type 'Auto' automatically detects the type from server metadata (Bool, Byte, Int16, Int32, Float, Double, String).

8.3 Namespace Browser

'Browse' button: opens an interactive dialog to navigate the OPC UA namespace. The selected NodeId is automatically copied into the current row.

8.4 Security

With 'Use Security' enabled, the connection uses Sign-and-Encrypt (TLS). Username and password can additionally be configured for authentication.

9. BACnet Protocol

BACnet/IP (ASHRAE 135, ISO 16484-5) is the standard for building automation systems. FieldbusAnalyzer implements BACnet/IP per Annex J over UDP port 47808.

9.1 Device Configuration

Field	Description	Default
Name	Display name of the device	-
IP Address	IPv4 address of the BACnet device	-
Port	UDP port	47808
Device Instance	BACnet Device Instance (0-4194302)	-
Description	Optional free-text description	-

9.2 BACnet Objects

Field	Description
-------	-------------

Name	Display name of the data point
Object Type	AnalogInput, AnalogOutput, AnalogValue, BinaryInput, BinaryOutput, BinaryValue, MultiStateInput, MultiStateOutput, MultiStateValue
Instance	Object instance number (0-4194302)
Data Type	Bool, Int32, UInt32, Float32, String
Unit	BACnet Engineering Unit (e.g. degreesCelsius, percent, pascals)
Warn Threshold	Limit for WARN alarm
Alarm Threshold	Limit for ALARM
Change Alarm	Alarm on any value change
Write Value	Write value for AnalogOutput/Value, BinaryOutput/Value, MultiStateOutput/Value

9.3 Whols Discovery

BACnet devices on the network can be discovered automatically:

- Search: sends a Whols broadcast (or unicast to configured IP) and waits for IAm responses
- Found Devices: listed with IP address and device instance
- Import: selected device is imported as a new device configuration

Note: In BACnet networks with a BBMD (BACnet Broadcast Management Device), the BBMD IP address must be configured as the target.

9.4 BACnet Slave Server (Test Mode)

The 'Slave' tab contains a built-in BACnet/IP server for testing purposes.

- Device Instance - own BACnet Device Instance of the slave server

- Port - UDP listen port (default: 47808)
- Available Objects - all configured objects are simulated and can be read by other BACnet clients

Supported services: ReadProperty, WriteProperty, Whols/IAm.

10. Common Features

The following features are available identically across all six protocols.

10.1 Polling

- Start Poll / Stop Poll - enable and disable polling
- Interval - polling interval in milliseconds (minimum: 50 ms)
- On-Change Only - identical successive values within polling cycles are filtered

10.2 Recording

- Start Rec - opens a new recording session in the SQLite database
- Stop Rec - closes the session and sets the end time
- On-Change - only changed values recorded (errors always recorded)

Each session is associated with one device and contains timestamp, tag name, value, and error flag.

10.3 Write with Confirmation

With 'Confirm' enabled, after each write the value is read back and compared to the target. Discrepancy is shown as a warning in the status bar.

10.4 Export and Import

- ↑ Export - saves the device configuration as a JSON file
- ↓ File... - loads a JSON file from the file system
- ↓ Paste... - import JSON text from the clipboard

11. Tracer and Network Analysis

The Tracer records all network packets at the protocol level, enabling fault analysis at the packet level.

11.1 Capture Control

- ▶ Capture / ■ ■ Capture - start and pause recording (button in Tracer toolbar)
- Status: gray = inactive, orange/bold = active
- The Tracer is global - applies to all protocols simultaneously
- Note: The Tracer can also be permanently enabled in Settings

11.2 Display Filter

- All - shows all recorded frames
- OK - shows only successful communication
- Errors - shows only erroneous frames

Statistics (Stats tab) are always calculated from all frames, regardless of the selected filter.

11.3 Tracer Table Columns

Column	Content
Time	Timestamp HH:mm:ss.fff
Protocol	Protocol name (Modbus, CANopen, EIP, PROFINET, OPC UA, BACnet)
IP	Destination IP address
Port	TCP/UDP destination port
Request	Hex preview of request bytes (max. 24 bytes)
Response	Hex preview of response bytes or error message
ms	Latency in milliseconds
Info	Error message or exception code on failure

11.4 PCAP Export (Wireshark)

'Export PCAP' button: all recorded frames are exported to a PCAP file. Wireshark-compatible with simulated Ethernet/IP/TCP headers. Filename includes protocol name and timestamp.

12. Statistics

The 'Statistics' tab shows aggregated metrics from all recorded tracer frames.

12.1 Transactions

- Total - total number of all communication operations
- Success - number of error-free operations (shown in green)
- Errors - number of failed operations (orange-red)
- Success Rate (%) - proportion of successful operations with progress bar

12.2 Error Breakdown

- Timeouts - operations where no response was received within the timeout
- Other Errors - all other error categories

12.3 Latency (ms)

- Min - shortest measured response time
- Average - mean response time
- P95 - 95th percentile (95% of requests were answered faster)
- Max - longest measured response time

Latencies are calculated only for successful transactions (error frames are excluded).

13. Recording and Logging

13.1 Session Management

Each recording is stored as a session in the SQLite database. The left list in the Log tab shows all sessions with device name, start time, and entry count. Clicking a session loads its entries.

13.2 Log Entries

Each entry contains: timestamp (HH:mm:ss.fff), tag name, value, and error flag. Erroneous entries are color-highlighted.

13.3 CSV Export

Button 'Export CSV': save selected session as CSV file.

Columns: Timestamp, Protocol, DeviceId, Tag, Value, IsError (0/1), Message. Separator configurable in Settings (default: semicolon).

14. Alarms

14.1 Alarm Types

Type	Color	Trigger
ALARM	Orange-Red (bold)	Value >= Alarm threshold
WARN	Orange	Value >= Warning threshold (less than Alarm threshold)
CHANGE	Cornflower Blue	Value changed (Change Alarm enabled)
CONNECTION	Dark Red (bold)	All read operations for a device failed

14.2 Configuring Thresholds

Thresholds are configured per data point in the register/tag table. Alarm logic: ALARM overrides WARN. When value returns below warning threshold, both alarms are automatically cleared.

14.3 Alarm Management

- Acknowledge - marks an alarm as confirmed (Ack)
- Acknowledge All - all alarms acknowledged at once
- Delete - permanently delete the selected alarm
- Clear All - delete all alarms (with confirmation prompt)
- Export CSV - save alarm list as CSV file

The badge in the tab title shows the number of unacknowledged alarms (orange-red number).

15. MQTT Integration

15.1 Configuration and Connection

Forward measured values to an MQTT broker (e.g., Mosquitto, HiveMQ, EMQX). Configure host, port, and topic prefix in Settings.

Use 'Connect' and 'Disconnect' to control the broker connection. Status shown in green when connected.

15.2 Topic Format

Topics: **{Prefix}/{DeviceName}/{TagName}**

Examples:

fieldbus/Compressor_1/Pressure fieldbus/PLC_Hall2/Temperature

- Values are only published during an active recording session (Rec)
- Only error-free values are published
- Message format: plain text value, e.g. '23.4' or 'True'

16. Data Export

16.1 CSV Export

Log entries and alarms can be exported as CSV files. Separator configurable in Settings.

- Recording CSV columns: Timestamp, Protocol, DeviceId, Tag, Value, IsError (0/1), Message
- Alarm CSV columns: Timestamp, Protocol, DeviceId, Tag, Level, Value, Message, Acknowledged (0/1)

16.2 PCAP Export (Wireshark)

All tracer frames can be exported as a PCAP file ('Export PCAP' button in Tracer tab). Contains complete Ethernet/IPv4/TCP frames, openable in Wireshark or TShark.

Appendix A: Supported Data Types

Data Type	Modbus	CANopen	EIP	PROFINET	OPC UA	BACnet
Boolean / Bool	X	-	X	X	X	X
Byte (8-bit unsigned)	-	X	X	X	X	-

Int8 / Signed8	-	X	-	X	-	-
Int16 / Signed16	X	X	X	X	X	-
UInt16 / Unsigned16	X	X	X	X	-	-
Int32 / Dint	X	X	X	X	X	X
UInt32 / Udint	X	X	X	X	-	X
Int64	X	-	-	-	-	-
Float32 / Real	X	X	X	X	X	X
Float64 / Double / LReal	X	X	X	-	X	-
String	X	X	X	X	X	X
Raw (Byte Array)	X	-	-	X	-	-
Auto (Type Detection)	-	-	-	-	X	-

Appendix B: Technical Specifications

Parameter	Value
Platform	Windows 10 / 11 (x64)
.NET Version	.NET 8.0, self-contained
UI Framework	WPF with ModernWpfUI 0.9.6
Database	SQLite via Microsoft.Data.Sqlite 10.x

MQTT Client	MQTTnet 4.3.x
OPC UA SDK	OPC Foundation .NET Standard SDK 1.5.x
BACnet Stack	Custom BACnet/IP implementation (UDP)
Tracer Buffer Size	1000 frames (ring buffer)
Max. Log Entries (view)	500 per session
Max. Alarms (view)	300
Chart History	100 data points per data point (sparkline)

Appendix C: Protocol Feature Comparison

Feature	Modbus	CANopen	EtherNet/IP	PROFINET	OPC UA	BACnet
Read / Polling	X	X	X	X	X	X
Write	X	X	X	X	X	X
Write Confirmation	X	X	X	X	X	X
Dashboard (Widgets)	X	-	-	-	-	-
Slave Server (Test)	X	-	-	-	-	X
Whols Discovery	-	-	-	-	-	X
NMT Commands	-	X	-	-	-	-
Node Scan	-	X	-	-	-	-

Device Identity	-	-	X	-	-	-
GSDML Import	-	-	-	X	-	-
GSD Import (PROFIBUS)	-	-	-	X	-	-
Namespace Browser	-	-	-	-	X	-
Tracer / PCAP Export	X	X	X	X	X	X
Statistics	X	X	X	X	X	X
Recording / Log	X	X	X	X	X	X
Alarms	X	X	X	X	X	X
MQTT Forwarding	X	X	X	X	X	X
CSV Export	X	X	X	X	X	X