

FieldbusAnalyzer

Benutzerhandbuch Version: 1.0.0 Datum:

24.05.2026 Zielplattform: Windows 10 / 11 (x64) Laufzeit: .NET 8
(enthalten)

1. Einleitung

1.1 Anwendungsübersicht

FieldbusAnalyzer ist eine Desktop-Anwendung für Windows zur Konfiguration, Überwachung und Analyse industrieller Fieldbus-Protokolle. Die Software ermöglicht das Lesen und Schreiben von Prozessdaten, die Aufzeichnung von Messwertverläufen, die Überwachung mit konfigurierbaren Alarmschwellen sowie die Netzwerkanalyse auf Paketebene.

Die Anwendung unterstützt sechs Protokolle in einer einheitlichen Benutzeroberfläche und bietet eine MQTT-Integration zur Weiterleitung von Messwerten an übergeordnete Systeme.

1.2 Unterstützte Protokolle

- Modbus TCP / RTU-over-TCP (RFC 3453, IEC 61158)
- CANopen über TCP-Gateway, Seriell-Gateway oder virtuellen CAN-Bus (CiA 301)
- EtherNet/IP mit CIP (ODVA EtherNet/IP, IEC 61158 Typ 2)
- PROFINET / PROFIBUS (IEC 61158, IEC 61784)
- OPC UA (OPC Unified Architecture, IEC 62541)
- BACnet/IP (ASHRAE 135, ISO 16484-5)

1.3 Systemanforderungen

Komponente	Anforderung
Betriebssystem	Windows 10 oder Windows 11 (64-Bit)
Arbeitsspeicher	Mind. 512 MB RAM (empfohlen: 2 GB)
Festplatte	Ca. 250 MB (inkl. .NET 8 Runtime)
.NET Runtime	Enthalten, self-contained - keine separate Installation
Netzwerk	TCP/IP-Verbindung zum Zielgerät
Bildschirmauflösung	Mind. 1280 x 720 Pixel

2. Erste Schritte

2.1 Installation

FieldbusAnalyzer wird als ZIP-Archiv geliefert und benötigt keine Installation:

- ZIP-Archiv in ein beliebiges Verzeichnis entpacken (z.B. C:\Programme\FieldbusAnalyzer).
- Anwendung durch Doppelklick auf FieldbusAnalyzer.App.exe starten.
- Beim ersten Start werden Konfigurationsdateien automatisch unter %AppData%\FieldbusAnalyzer angelegt.

Eine Deinstallation ist nicht erforderlich - das Verzeichnis kann einfach gelöscht werden.

2.2 Datenspeicherung

Pfad	Inhalt
%AppData%\FieldbusAnalyzer\fieldbus.db	SQLite-Datenbank mit Aufzeichnungen und Alarmen
%AppData%\FieldbusAnalyzer\prefs.json	Benutzereinstellungen (Theme, MQTT, Intervalle)

%AppData%\FieldbusAnalyzer\modbus.json	Modbus-Gerätekonfigurationen
%AppData%\FieldbusAnalyzer\canopen.json	CANopen-Knotenkonfigurationen
%AppData%\FieldbusAnalyzer\eip.json	EtherNet/IP-Gerätekonfigurationen
%AppData%\FieldbusAnalyzer\profinet.json	PROFINET-Gerätekonfigurationen
%AppData%\FieldbusAnalyzer\opcua.json	OPC-UA-Serverkonfigurationen
%AppData%\FieldbusAnalyzer\bacnet.json	BACnet-Gerätekonfigurationen

2.3 Hauptfenster

Das Hauptfenster ist in drei Bereiche aufgeteilt:

- Linke Navigation: Protokollauswahl (Modbus, CANopen, EtherNet/IP, PROFINET, OPC UA, BACnet, Einstellungen).
- Linke Seitenleiste: Geräte-/Knotenliste und Editor für Geräteeigenschaften (protokollspezifisch).
- Rechter Hauptbereich: Daten-Tabs oben (Register/Tags/Module/Knoten, Dashboard) und Diagnose-Tabs unten (Tracer, Statistik, Log, Alarme). Getrennt durch einen horizontalen GridSplitter.

3. Einstellungen

Die Einstellungen sind über den Tab 'Einstellungen' erreichbar. Änderungen werden mit dem Speichern-Knopf übernommen.

3.1 Darstellung (Theme)

- System - folgt den Windows-Systemeinstellungen (Hell/Dunkel)
- Hell - helles Theme
- Dunkel - dunkles Theme (Standard)

3.2 MQTT-Publisher

FieldbusAnalyzer kann Messwerte während einer aktiven Aufzeichnung an einen MQTT-Broker weiterleiten.

Parameter	Beschreibung	Standard
MQTT aktiviert	Funktion ein-/ausschalten	Nein
Host	Hostname oder IP-Adresse des Brokers	localhost
Port	TCP-Port des Brokers	1883
Topic-Präfix	Präfix aller veröffentlichten Topics	fieldbus

Topic-Format: {Präfix}/{Gerätename}/{Tagname} Beispiel:
fieldbus/SPS1/Temperatur

Werte werden nur während aktiver Aufzeichnung (Rec) und nur bei fehlerfreien Lesevorgängen publiziert.

3.3 Allgemeine Einstellungen

- Standard-Poll-Intervall (ms) - globales Abfrageintervall, Standard: 1000 ms
- CSV-Trennzeichen - Spaltentrennzeichen für CSV-Exporte, Standard: Semikolon ;
- Datenbank-Pfad - Anzeige des SQLite-Datenbankpfads (schreibgeschützt). 'Öffnen' zeigt Ordner im Explorer.
- Protocol Tracer aktiviert - aktiviert/deaktiviert die globale Netzwerkaufzeichnung

3.4 Sprache

Die Benutzeroberfläche ist in Deutsch und Englisch verfügbar. Die Sprache kann jederzeit im Tab 'Einstellungen' unter 'Sprache' gewechselt werden - ein Neustart der Anwendung ist nicht erforderlich.

- Deutsch - vollständige deutsche Benutzeroberfläche
- English - vollständige englische Benutzeroberfläche

4. Modbus-Protokoll

Modbus ist das am weitesten verbreitete Protokoll in der Industrieautomatisierung. FieldbusAnalyzer unterstützt Modbus TCP sowie RTU-over-TCP (Modbus RTU-Rahmen über TCP-Verbindung).

4.1 Gerätekonfiguration

Feld	Beschreibung	Standard
Name	Anzeigename (Pflichtfeld)	-
IP-Adresse	IPv4-Adresse des Modbus-Servers	-
Port	TCP-Port	502
Unit ID	Modbus-Geräteadresse (1-247)	1
Beschreibung	Optionale Freitextbeschreibung	-
RTU over TCP	Aktiviert RTU-Rahmen über TCP-Verbindung	Nein

4.2 Register und Spulen

Feld	Beschreibung
Name	Anzeigename des Datenpunkts
Funktionscode	FC01 Coils, FC02 Discrete, FC03 Holding, FC04 Input, FC05/06/15/16 Schreiben
Adresse	Registeradresse (0-65535)
Datentyp	Boolean, Int16/32/64, UInt16/32, Float32/64, String, Raw
Anzahl	Array-Größe (Standard: 1)
Byte-Reihenfolge	Big-Endian oder Little-Endian
Skalierung	Anzeigewert = (Rohwert x Skalierung) + Offset
Offset	Additionskonstante nach Skalierung

Einheit	Anzeigeeinheit, z.B. °C, bar, rpm
Warnschwelle	Grenzwert für WARN-Alarm
Alarmschwelle	Grenzwert für ALARM
Änderungsalarm	Alarm bei jeder Wertänderung
Widget	Auto, Wert, LED, Manometer, Diagramm
IP-Überschreibung	Abweichende IP/Port für diesen Datenpunkt
Schreibwert	Manuell eingebbarer Schreibwert

4.3 Daten lesen und schreiben

- Einmal lesen - liest alle Datenpunkte einmalig aus
- Poll starten / Poll stoppen - startet/stoppt das zyklische Polling
- Poll-Intervall (ms) - Abfrageintervall (50 bis 60000 ms)
- Nur Änderungen - nur geänderte Werte in die Datenbank schreiben
- Alle schreiben - schreibt alle Datenpunkte mit eingegebenem Schreibwert
- Schreiben bestätigen - nach Schreiben Wert zurücklesen und vergleichen
- W-Knopf (pro Zeile) - nur den ausgewählten Datenpunkt schreiben

4.4 Dashboard

Registerkarte 'Dashboard' zeigt alle Datenpunkte als visuelle Karten.

Widget-Typen:

- Auto - automatische Auswahl (LED für Boolean, Wert für Zahlen)
- Wert - numerischer Anzeigewert mit Einheit
- LED - grün/rote Anzeige für Boolean-Werte (Ein/Aus)
- Manometer - kreisförmig mit konfigurierbarem Min/Max-Bereich
- Diagramm - Zeitreihendarstellung der letzten 100 Messwerte (Sparkline)

4.5 Modbus-Slave-Server (Testmodus)

Registerkarte 'Slave': integrierter Modbus-TCP-Server für Testzwecke.

Parameter	Beschreibung	Standard
Port	TCP-Listenport des Servers	502
Unit ID	Geräteadresse	1
Verzögerung (ms)	Künstliche Antwortverzögerung zur Simulation langsamer Geräte	0
Exception erzwingen	Server antwortet auf alle Anfragen mit Fehlercode	Nein

Der Slave-Server protokolliert alle Schreibzugriffe im Ereignisprotokoll.

5. CANopen-Protokoll

CANopen (CiA 301) ist ein Kommunikationsprotokoll für eingebettete Systeme. FieldbusAnalyzer kommuniziert über TCP-Gateway, Seriell-Gateway oder virtuellen CAN-Bus.

5.1 Knotenkonfiguration

Feld	Beschreibung	Standard
Name	Anzeigename des Knotens	-
Knoten-ID	CANopen-Geräteadresse (1-127)	-
IP-Adresse	Gateway-IP-Adresse	-
Port	Gateway-TCP-Port	3000

Baudrate	CAN-Bus-Baudrate (bps)	250000
Verbindungstyp	TCP-Gateway, Seriell-Gateway oder virtueller CAN	TCP-Gateway
EDS-Datei	Referenz auf Electronic Data Sheet	-
NMT-Zustand	Aktueller NMT-Status (wird beim Lesen aktualisiert)	Unbekannt

5.2 Objektverzeichnis

CANopen-Objekte werden über Index (0x0000-0xFFFF) und Sub-Index (0-255) adressiert.

- Index / Sub-Index - Objektadresse im CANopen-Objektverzeichnis
- Datentyp - Unsigned8/16/32, Signed8/16/32, Float, Double, String
- Zugriffstyp - RO (nur lesen), WO (nur schreiben), RW (lesen/schreiben)
- PDO-Mapping - Markierung als Prozessdatenobjekt für zyklische Datenübertragung
- Skalierung, Offset, Einheit, Schwellenwerte

5.3 NMT-Befehle

- Start Remote Node (0x01) - versetzt Knoten in Betrieb
- Stop Remote Node (0x02) - stoppt den Knoten
- Enter Pre-operational (0x80) - vorbereitender Zustand
- Reset Communication (0x82) - Kommunikation zurücksetzen

5.4 Knoten-Scan

'Scan Nodes' sendet eine Erkennungsanfrage und listet alle erreichbaren Knoten mit ihrer Geräteadresse und ihrem NMT-Status auf.

6. EtherNet/IP-Protokoll

EtherNet/IP verwendet das Common Industrial Protocol (CIP) der ODVA und wird häufig in Rockwell/Allen-Bradley Steuerungen eingesetzt.

6.1 Gerätekonfiguration

Feld	Beschreibung	Standard
Name	Anzeigename des Geräts	-
IP-Adresse	IPv4-Adresse der Steuerung	-
Port	TCP-Port (CIP)	44818
Steckplatz	Modulsteckplatz bei mehrstufigen Geräten	0

6.2 Tag-Konfiguration

- CIP-Pfad - Format '1.1' (Port.Link) oder '4.1.3' (Klasse.Instanz.Attribut)
- Logix-Tagname - einfacher Tagname für Allen-Bradley Logix-Steuerungen
- Servicecode - GetAttributeAll, GetAttributeSingle, SetAttributeSingle
- Datentyp - Dint, Int, Uint, Bool, Byte, Real, LReal, String
- IP-Überschreibung, Skalierung, Offset, Einheit, Schwellenwerte

6.3 Geräteidentität auslesen

Knopf 'Identität lesen': fragt Hersteller, Modell, Seriennummer und Firmware-Version ab und zeigt sie in der Statusleiste an.

7. PROFINET-Protokoll

PROFINET (IEC 61158) ist der Standard-Feldbus der PROFIBUS & PROFINET International (PI). Unterstützt werden S7-TCP, PROFIBUS-Gateway und PROFINET-TCP-Verbindungen.

7.1 Gerätekonfiguration

Feld	Beschreibung	Standard
Name	Anzeigename	-

IP-Adresse	IPv4-Adresse der Steuerung	-
Port	TCP-Port (S7: 102)	102
Gerätename	PROFINET NameOfStation	-
Hersteller-ID	VendorID (aus GSDML)	-
Geräte-ID	DeviceID (aus GSDML)	-
Rack	Rack-Nummer (für S7-Verbindung)	0
Steckplatz	Steckplatznummer (für S7-Verbindung)	1
Stationsadresse	PROFIBUS-Stationsadresse (Gateway)	-
Verbindungstyp	S7 TCP / PROFIBUS-Gateway / PROFINET TCP	S7 TCP
GSD-Datei	Referenz auf Gerätebeschreibungsdatei	-

7.2 Module und Speicherbereiche

Bereich	Kürzel	Beschreibung
Eingang	I / E	Prozesseingangsabbild
Ausgang	Q / A	Prozessausgangsabbild
Merker	M	Flagregister
Timer	T	Timer-Register
Zähler	C / Z	Zähler-Register

Datenbaustein	DB	Datenbaustein mit Nummer und Byte-Offset
I/O-Record	Rec	PROFINET-Datensatz (Slot/Subslot/Index)

Pro Modul konfigurierbar: Datentyp, Byte-Adresse, Bit-Adresse (für Bool), Array-Größe, Skalierung, Einheit, Schwellenwerte.

7.3 GSDML-Import

GSDML-Dateien (IEC 61784-2) beschreiben PROFINET-Geräte im XML-Format. Knopf: '↓ GSDML...'

- Gerätedaten (Hersteller, Hersteller-ID, Geräte-ID) werden automatisch übernommen
- Alle I/O-Datensätze (Slot, Subslot, Index, Datentyp, Richtung) werden als Module angelegt
- Das neue Gerät wird sofort in der Geräteliste angezeigt

7.4 GSD-Import (PROFIBUS)

GSD-Dateien beschreiben PROFIBUS-Geräte im INI-Format. Knopf: '↓ GSD...'

- Modellname und Hersteller werden aus der Datei gelesen
- Module werden automatisch erkannt und angelegt
- Verbindungstyp wird auf 'PROFIBUS-Gateway' gesetzt

8. OPC-UA-Protokoll

OPC Unified Architecture (IEC 62541) ist der plattformunabhängige Standard für industrielle Datenübertragung. FieldbusAnalyzer verwendet die OPC Foundation .NET Standard SDK.

8.1 Serverkonfiguration

Feld	Beschreibung	Beispiel
Name	Anzeigename des Servers	Mein OPC-UA-Server
Server-URL	OPC-UA-Endpunktadresse	opc.tcp://192.168.1.100:4840
Sicherheit	Sign & Encrypt (TLS-Verschlüsselung)	Nein

Benutzername	Anmeldekennung (leer = anonym)	admin
Kennwort	Anmeldekennwort	-

8.2 Knoten (NodeId-Adressierung)

OPC-UA-Knoten werden über ihre NodeId adressiert - zwei Formate:

- Numerische ID: ns=2;i=1001 (Namespace-Index + Ganzzahl-Bezeichner)
- String-ID: ns=2;s=Temperatur (Namespace-Index + Zeichenketten-Bezeichner)

Datentyp 'Auto' erkennt den Typ automatisch aus den Server-Metadaten (Bool, Byte, Int16, Int32, Float, Double, String).

8.3 Namespace-Browser

Knopf 'Browser': öffnet interaktiven Dialog zum Durchsuchen des OPC-UA-Namensraums. Die ausgewählte NodeId wird automatisch in die aktuelle Zeile übernommen.

8.4 Sicherheit

Bei aktiviertem 'Sicherheit verwenden' wird die Verbindung mit Sign-and-Encrypt (TLS) aufgebaut. Benutzername und Kennwort können zusätzlich für die Authentifizierung konfiguriert werden.

9. BACnet-Protokoll

BACnet/IP (ASHRAE 135, ISO 16484-5) ist der Standard für Gebäudeautomatisierungssysteme. FieldbusAnalyzer implementiert BACnet/IP gemäß Annex J über UDP-Port 47808.

9.1 Gerätekonfiguration

Feld	Beschreibung	Standard
Name	Anzeigename des Geräts	-
IP-Adresse	IPv4-Adresse des BACnet-Geräts	-

Port	UDP-Port	47808
Geräte-Instanz	BACnet Device Instance (0-4194302)	-
Beschreibung	Optionale Freitextbeschreibung	-

9.2 BACnet-Objekte

Feld	Beschreibung
Name	Anzeigename des Datenpunkts
Objekttyp	AnalogInput, AnalogOutput, AnalogValue, BinaryInput, BinaryOutput, BinaryValue, MultiStateInput, MultiStateOutput, MultiStateValue
Instanz	Objektinstanznummer (0-4194302)
Datentyp	Bool, Int32, UInt32, Float32, String
Einheit	BACnet Engineering Unit (z.B. degreesCelsius, percent, pascals)
Warnschwelle	Grenzwert für WARN-Alarm
Alarmschwelle	Grenzwert für ALARM
Änderungsalarm	Alarm bei jeder Wertänderung
Schreibwert	Schreibwert für AnalogOutput/Value, BinaryOutput/Value, MultiStateOutput/Value

9.3 Whols-Erkennung

BACnet-Geräte im Netzwerk können automatisch erkannt werden:

- Suchen: sendet einen Whols-Broadcast (oder Unicast an konfigurierte IP) und wartet auf IAm-Antworten
- Gefundene Geräte: werden mit IP-Adresse und Geräte-Instanz aufgelistet

- Importieren: ausgewähltes Gerät wird als neue Gerätekonfiguration übernommen

Hinweis: Bei BACnet-Netzwerken mit BBMD (BACnet Broadcast Management Device) muss die BBMD-IP-Adresse als Ziel konfiguriert werden.

9.4 BACnet-Slave-Server (Testmodus)

Registerkarte 'Slave': integrierter BACnet/IP-Server für Testzwecke.

- Geräte-Instanz - eigene BACnet Device Instance des Slave-Servers
- Port - UDP-Listenport (Standard: 47808)
- Verfügbare Objekte - alle konfigurierten Objekte werden simuliert und können von anderen BACnet-Clients gelesen werden

Unterstützte Dienste: ReadProperty, WriteProperty, Whos/IAm.

10. Gemeinsame Funktionen

Folgende Funktionen stehen bei allen sechs Protokollen identisch zur Verfügung.

10.1 Polling

- Poll starten / Poll stoppen - Polling aktivieren und deaktivieren
- Intervall - Abfrageintervall in Millisekunden (Minimum: 50 ms)
- Nur Änderungen (Polling) - identische aufeinanderfolgende Werte werden gefiltert und nicht erneut angezeigt

10.2 Aufzeichnung (Recording)

- Rec starten - öffnet eine neue Aufzeichnungssitzung in der SQLite-Datenbank
- Rec stoppen - schließt die Sitzung und setzt die Endzeit
- On-Change - nur geänderte Werte aufzeichnen (Fehler werden immer aufgezeichnet)

Jede Sitzung ist einem Gerät zugeordnet und enthält Zeitstempel, Tagname, Wert und Fehlerkennzeichen.

10.3 Schreiben mit Bestätigung

Bei aktiviertem 'Bestätigen' wird nach jedem Schreibvorgang der Wert zurückgelesen und mit dem Sollwert verglichen. Bei Abweichung erscheint eine Warnmeldung in der Statusleiste.

10.4 Export und Import

- ↑ Export - speichert die Gerätekonfiguration als JSON-Datei
- ↓ Datei... - lädt eine JSON-Datei vom Dateisystem
- ↓ Einfügen... - JSON-Text aus der Zwischenablage importieren

11. Tracer und Netzwerkanalyse

Der Tracer zeichnet alle Netzwerkpakete auf Protokollebene auf und ermöglicht die Fehleranalyse auf Paketebene.

11.1 Capture-Steuerung

- ► Capture / ■ ■ Capture - Aufzeichnung starten und pausieren (Knopf im Tracer-Tab)
- Status: grau = inaktiv, orange/fett = aktiv
- Der Tracer ist global - gilt für alle Protokolle gleichzeitig
- Hinweis: Der Tracer kann auch in den Einstellungen dauerhaft aktiviert werden

11.2 Anzeigefilter

- Alle - zeigt alle aufgezeichneten Frames
- OK - zeigt nur erfolgreiche Kommunikation
- Fehler - zeigt nur fehlerhafte Frames

Die Statistik (Stats-Tab) wird immer aus allen Frames berechnet, unabhängig vom gewählten Filter.

11.3 Tracer-Tabelle

Spalte	Inhalt
Zeit	Uhrzeitstempel HH:mm:ss.fff
Protokoll	Protokollname (Modbus, CANopen, EIP, PROFINET, OPC UA, BACnet)
IP	Ziel-IP-Adresse
Port	TCP/UDP-Zielport

Anfrage	Hex-Vorschau der Anfrage-Bytes (max. 24 Bytes)
Antwort	Hex-Vorschau der Antwort-Bytes oder Fehlermeldung
ms	Latenz in Millisekunden
Info	Fehlermeldung oder Exception-Code bei Fehler

11.4 PCAP-Export (Wireshark)

Knopf 'Export PCAP': alle aufgezeichneten Frames werden in eine PCAP-Datei exportiert. Wireshark-kompatibel mit simulierten Ethernet/IP/TCP-Headern. Dateiname enthält Protokollname und Zeitstempel.

12. Statistik

Registerkarte 'Statistik' zeigt aggregierte Kennzahlen aller aufgezeichneten Tracer-Frames.

12.1 Transaktionen

- Gesamt - Gesamtanzahl aller Kommunikationsvorgänge
- Erfolg - Anzahl fehlerfreier Vorgänge (grün angezeigt)
- Fehler - Anzahl fehlerhafter Vorgänge (orange-rot)
- Erfolgsrate (%) - Anteil erfolgreicher Vorgänge mit Fortschrittsbalken

12.2 Fehleraufschlüsselung

- Timeouts - Vorgänge ohne Antwort innerhalb des Timeouts
- Sonstige Fehler - alle anderen Fehlerkategorien

12.3 Latenz (ms)

- Min - kürzeste gemessene Antwortzeit
- Mittelwert - durchschnittliche Antwortzeit
- P95 - 95. Perzentil (95% der Anfragen wurden schneller beantwortet)
- Max - längste gemessene Antwortzeit

Latenzen werden nur für erfolgreiche Transaktionen berechnet (Fehler-Frames werden ausgeschlossen).

13. Aufzeichnung und Protokollierung

13.1 Sitzungsverwaltung

Jede Aufzeichnung wird als Sitzung in der SQLite-Datenbank gespeichert. Die linke Liste im Log-Tab zeigt alle Sitzungen mit Gerätenamen, Startzeitpunkt und Anzahl der Einträge. Ein Klick auf eine Sitzung lädt die zugehörigen Einträge.

13.2 Log-Einträge

Jeder Eintrag enthält: Zeitstempel (HH:mm:ss.fff), Tagname, Wert und Fehlerkennzeichen. Fehlerhafte Einträge werden farblich hervorgehoben.

13.3 CSV-Export

Knopf 'Export CSV': ausgewählte Sitzung als CSV-Datei speichern.

Spalten: Zeitstempel, Protokoll, Geräte-ID, Tag, Wert, Fehler (0/1), Meldung. Das Trennzeichen ist in den Einstellungen konfigurierbar (Standard: Semikolon).

14. Alarmer

14.1 Alarmtypen

Typ	Farbe	Auslöser
ALARM	Orange-Rot (fett)	Messwert \geq Alarmschwelle
WARN	Orange	Messwert \geq Warnschwelle (kleiner als Alarmschwelle)
CHANGE	Kornblumenblau	Wert hat sich geändert (Änderungsalarm aktiviert)
CONNECTION	Dunkelrot (fett)	Alle Lesevorgänge für ein Gerät fehlgeschlagen

14.2 Schwellenwerte konfigurieren

Schwellenwerte werden pro Datenpunkt in der Register-/Tag-Tabelle konfiguriert. Alarm-Logik: ALARM überschreibt WARN. Bei Rückkehr

unter die Warnschwelle werden beide Alarme automatisch zurückgesetzt.

14.3 Alarmverwaltung

- Quittieren - Alarm als bestätigt markieren (Ack)
- Alle quittieren - alle Alarme auf einmal quittieren
- Löschen - ausgewählten Alarm dauerhaft löschen
- Alle löschen - alle Alarme löschen (mit Rückfrage)
- Export CSV - Alarmliste als CSV-Datei speichern

Das Badge-Symbol im Tab-Titel zeigt die Anzahl unquittierter Alarme (orange-rote Zahl).

15. MQTT-Integration

15.1 Konfiguration und Verbindung

Weiterleitung von Messwerten an einen MQTT-Broker (z.B. Mosquitto, HiveMQ, EMQX). Konfigurierbar in den Einstellungen (Host, Port, Topic-Präfix).

Mit 'Verbinden' und 'Trennen' wird die Verbindung gesteuert. Verbindungsstatus wird grün angezeigt.

15.2 Topic-Format

Topics werden nach folgendem Schema aufgebaut:

{Präfix}/{Gerätename}/{Tagname}

Beispiele:

fieldbus/Kompressor_1/Druck fieldbus/SPS_Halle2/Temperatur

- Werte werden nur während aktiver Aufzeichnung (Rec) publiziert
- Nur fehlerfreie Werte werden veröffentlicht
- Nachrichtenformat: reiner Textwert, z.B. '23.4' oder 'True'

16. Datenexport

16.1 CSV-Export

Log-Einträge und Alarme können als CSV-Dateien exportiert werden. Das Trennzeichen ist in den Einstellungen konfigurierbar.

- Aufzeichnungs-Csv-Spalten: Zeitstempel, Protokoll, Geräte-ID, Tag, Wert, Fehler (0/1), Meldung

- Alarm-CSV-Spalten: Zeitstempel, Protokoll, Geräte-ID, Tag, Level, Wert, Meldung, Quittiert (0/1)

16.2 PCAP-Export (Wireshark)

Alle Tracer-Frames können als PCAP-Datei exportiert werden (Knopf 'Export PCAP' im Tracer-Tab). Die Datei enthält vollständige Ethernet/IPv4/TCP-Rahmen und kann in Wireshark oder TShark geöffnet werden.

Anhang A: Unterstützte Datentypen

Datentyp	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 (Typ-Erkennung)	-	-	-	-	X	-
----------------------	---	---	---	---	---	---

Anhang B: Technische Spezifikationen

Parameter	Wert
Plattform	Windows 10 / 11 (x64)
.NET-Version	.NET 8.0, self-contained
UI-Framework	WPF mit ModernWpfUI 0.9.6
Datenbank	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	Eigene BACnet/IP-Implementierung (UDP)
Tracer-Puffergröße	1000 Frames (Ring-Puffer)
Max. Log-Einträge (Ansicht)	500 pro Sitzung
Max. Alarme (Ansicht)	300
Diagramm-Historie	100 Messwerte pro Datenpunkt (Sparkline)

Anhang C: Funktionsvergleich Protokolle

Funktion	Modbus	CANopen	EtherNet/IP	PROFINET	OPC UA	BACnet
----------	--------	---------	-------------	----------	--------	--------

Lesen / Polling	X	X	X	X	X	X
Schreiben	X	X	X	X	X	X
Bestätigung Schreiben	X	X	X	X	X	X
Dashboard (Widgets)	X	-	-	-	-	-
Slave-Server (Test)	X	-	-	-	-	X
Whols-Erkennung	-	-	-	-	-	X
NMT-Befehle	-	X	-	-	-	-
Knoten-Scan	-	X	-	-	-	-
Geräteidentität	-	-	X	-	-	-
GSDML-Import	-	-	-	X	-	-
GSD-Import (PROFIBUS)	-	-	-	X	-	-
Namespace-Browser	-	-	-	-	X	-
Tracer / PCAP-Export	X	X	X	X	X	X
Statistik (Stats)	X	X	X	X	X	X
Aufzeichnung / Log	X	X	X	X	X	X
Alarme	X	X	X	X	X	X
MQTT-Weiterleitung	X	X	X	X	X	X
CSV-Export	X	X	X	X	X	X

