

# ModbusApp – Funktionsdokumentation

**Plattform:** Android (minSdk 26 / compileSdk 35)

**Technologiestack:** Kotlin 2.0 · Jetpack Compose · Material 3 · Room · Coroutines

**Protokolle:** Modbus TCP · RTU/USB-Serial · RTU/Bluetooth · RTU/WiFi-Bridge

## Inhaltsverzeichnis

1. [App-Überblick & Einstieg](#1-app-überblick--einstieg)
2. [Navigation & Menüstruktur](#2-navigation--menüstruktur)
3. [Hauptbildschirm (MainScreen)](#3-hauptbildschirm-mainscreen)
4. [Strukturliste (StructureListScreen)](#4-strukturliste-structurelistscreen)
5. [Strukturdetail (StructureDetailScreen)](#5-strukturdetail-structuredetailscreen)
6. [Struktur anlegen/bearbeiten (StructureFormScreen)](#6-struktur-anlegenbearbeiten-structureformscreen)
7. [Detail anlegen/bearbeiten (DetailFormScreen)](#7-detail-anlegenbearbeiten-detailformscreen)
8. [Aufzeichnungen (LogScreen)](#8-aufzeichnungen-logscreen)
9. [Alarmer (AlarmScreen)](#9-alarmer-alarmscreen)
10. [Diagnose (DiagScreen)](#10-diagnose-diagscreen)
11. [Slave-Modus (SlaveScreen)](#11-slave-modus-slavescreen)
12. [NFC (NfcScreen)](#12-nfc-nfcscreen)
13. [QR-Code (QrScreen)](#13-qr-code-qrscreen)
14. [Einstellungen (SettingsScreen)](#14-einstellungen-settingscreen)
15. [Homescreen-Widget](#15-homescreen-widget)
16. [Wear OS](#16-wear-os)
17. [Datenmodelle](#17-datenmodelle)
18. [Modbus-Protokollimplementierung](#18-modbus-protokollimplementierung)
19. [Datenbank](#19-datenbank)
20. [Berechtigungen](#20-berechtigungen)

## 1. App-Überblick & Einstieg

Die ModbusApp ist eine vollständige Modbus-Kommunikationsanwendung für Android. Sie unterstützt sowohl den Master- als auch den Slave-Modus und bietet Datenvisualisierung, Aufzeichnung, Alarmierung und Diagnose.

**Zwei Einstiegswege beim App-Start:**

Einstieg	Beschreibung
Hauptbildschirm	Direktzugriff auf Modbus-Protokoll ohne

	Konfiguration – für schnelle Tests
<b>Strukturliste</b>	Verwaltung gespeicherter Geräte/Anlagen – für strukturierte Arbeit

### Unterstützte Verbindungsarten:

Typ	Beschreibung
Modbus TCP	Netzwerkverbindung (IP + Port)
RTU/USB-Serial	Direktverbindung über USB-OTG
RTU/Bluetooth	Kabellose Verbindung (SPP-Profil)
RTU/WiFi-Bridge	RTU-Protokoll über TCP-Brücke

## 2. Navigation & Menüstruktur

Die App startet auf dem **Hauptbildschirm (MainScreen)**. Die primäre Navigation erfolgt über die **Bottom Navigation Bar** mit sechs Tabs sowie über Icon-Buttons in der oberen Leiste (TopAppBar) und Floating Action Buttons (FAB, das „+“-Symbol).

### Bottom Navigation Bar:

Tab	Icon	Screen
Start	Haus	Hauptbildschirm (MainScreen)
Geräte	Liste	Strukturliste (StructureListScreen)
Slave	Memory	Slave-Modus (SlaveScreen)
Protokoll	Verlauf	Aufzeichnungen (LogScreen)
Alarmer	Glocke	Alarmer (AlarmScreen)
Einstellungen	Zahnrad	Einstellungen (SettingsScreen)

```

App-Start
├── Hauptbildschirm (MainScreen)           ← Bottom Nav: Haus-Icon
├── Strukturliste (StructureListScreen)    ← Bottom Nav: Listen-Icon
│   ├── [+ FAB] → Neue Struktur anlegen (StructureFormScreen)
│   ├── [Stift-Icon] → Struktur bearbeiten (StructureFormScreen)
│   └── [Karten-Tap] → Strukturdetail öffnen (StructureDetailScreen)
│       ├── [+ FAB] → Neues Detail anlegen (DetailFormScreen)
│       └── [Stift-Icon] → Detail bearbeiten (DetailFormScreen)
├── Slave-Modus (SlaveScreen)             ← Bottom Nav: Memory-Icon
├── Aufzeichnungen (LogScreen)           ← Bottom Nav: Verlauf-Icon
├── Alarmer (AlarmScreen)                 ← Bottom Nav: Glocken-Icon
└── (Badge)

```

└─ Einstellungen (SettingsScreen)	← Bottom Nav: Zahnrad-Icon
└─ [Glocke-Icon]	→ Alarme (AlarmScreen)
└─ [Bug-Icon]	→ Diagnose (DiagScreen)
└─ [NFC-Button]	→ NFC (NfcScreen)
└─ [QR-Button]	→ QR-Code (QrScreen)



### 3. Hauptbildschirm (MainScreen)

**Erreichen:** App starten – der Hauptbildschirm erscheint automatisch als erster Screen.

Der Hauptbildschirm ermöglicht schnellen, direkten Zugriff auf das Modbus-Protokoll ohne vorherige Struktur-Konfiguration. Alle vier Bereiche befinden sich auf einem scrollbaren Screen.

#### 3.1 Verbindung herstellen

**Weg:** App starten → Bereich „Verbindung“ ist sofort sichtbar (erster Abschnitt).

Schritt	Bedienelement	Beschreibung
1	Dropdown „Verbindungstyp“	Tippen → Auswahl: <b>Modbus TCP · RTU/USB-Serial · RTU/Bluetooth · RTU/WiFi</b>
2a	Radio-Button „Master/Slave“	Nur bei TCP sichtbar – Rolle wählen
2b	Felder IP + Port	Bei TCP und RTU/WiFi: IP-Adresse und Port eingeben
2b	Dropdown „USB-Gerät“ + 	Bei USB-Serial: Gerät wählen, Scan-Button zum Suchen, Baudrate eingeben
2b	Dropdown „BT-Gerät“ + 	Bei Bluetooth: Gerät wählen, Scan-Button zum Suchen
3	Feld „Slave-ID“	Unit-ID des Zielgeräts eingeben (Standard: 1)
4	Button „Verbinden“	Tippen → Verbindung wird aufgebaut; Status erscheint darunter
—	Button „Trennen“	Tippen → Verbindung wird beendet (erscheint nach erfolgreicher Verbindung)

*\*\*Hinweis:\*\* Alle Felder sind während einer aktiven Verbindung gesperrt. Erst trennen, dann Konfiguration ändern.*

### 3.2 Register lesen (Modus 0 – Raw)

**Weg:** App starten → nach unten scrollen → Bereich „Register lesen (Master)“ → Segment „4 Register“ auswählen.

Schritt	Bedienelement	Beschreibung
1	Verbindung herstellen	(siehe 3.1)
2	Segmented Button „4 Register“	Tippen (linkes Segment) – ist Standard
3	Feld „Start-Adresse“	Dezimale Startadresse eingeben
4	Button „Lesen“	Tippen → 4 Register ab Startadresse werden gelesen
—	4 schreibgeschützte Felder	Ergebnisse werden direkt in den Reg 1–4 Feldern angezeigt

### 3.3 Register lesen (Modus 1 – Datentyp)

**Weg:** App starten → nach unten scrollen → Bereich „Register lesen (Master)“ → Segment „Datentyp“ auswählen.

Schritt	Bedienelement	Beschreibung
1	Verbindung herstellen	(siehe 3.1)
2	Segmented Button „Datentyp“	Tippen (rechtes Segment)
3	Dropdown „Datentyp“	Tippen → Typ wählen (BOOLEAN, INT16, FLOAT32 usw.)
4	Feld „Start-Adresse“	Dezimale Startadresse eingeben
5	Button „Lesen“	Tippen → Wert wird gelesen und konvertiert angezeigt
—	Ergebnisbox	Zeigt konvertierten Wert (fett) + Hex-Register-Info darunter

### 3.4 Register schreiben (Modus 0 – Raw)

**Weg:** App starten → nach unten scrollen → Bereich „Register schreiben (Master)“ → Segment „4 Register“.

Schritt	Bedienelement	Beschreibung
1	Verbindung herstellen	(siehe 3.1)

2	Segmented Button „ <b>4 Register</b> “	Tippen (linkes Segment)
3	Feld „Start-Adresse“	Zieladresse eingeben
4	Felder „Reg 1“ – „Reg 4“	Dezimalwerte eingeben
5	Button „ <b>Schreiben</b> “	Tippen → Werte werden ins Gerät geschrieben

### 3.5 Register schreiben (Modus 1 – Datentyp)

**Weg:** App starten → nach unten scrollen → Bereich „Register schreiben (Master)“ → Segment „Datentyp“.

Schritt	Bedienelement	Beschreibung
1	Verbindung herstellen	(siehe 3.1)
2	Segmented Button „ <b>Datentyp</b> “	Tippen (rechtes Segment)
3	Dropdown „Datentyp“	Typ wählen
4	Feld „Start-Adresse“	Zieladresse eingeben
5	Feld „Wert“	Zu schreibenden Wert eingeben
—	Register-Vorschau	Live-Anzeige der Rohregister-Werte (z. B. „Reg0: 0x0042 (66)“)
6	Button „ <b>Schreiben</b> “	Tippen → konvertierter Wert wird geschrieben

### 3.6 Protokoll-Log einsehen

**Weg:** App starten → ganz nach unten scrollen → Bereich „Protokoll“.

- Zeigt die letzten 10 Kommunikationsereignisse im Hex-Format
- Wird automatisch nach jedem Lesen/Schreiben aktualisiert
- Kein manuelles Auslösen nötig







## 4. Strukturliste (StructureListScreen)

**Erreichen:** Hauptbildschirm → System-Zurück-Taste

\*(Die Strukturliste ist die zentrale Verwaltungsoberfläche der App.)\*

Die Strukturliste zeigt alle konfigurierten Modbus-Geräte/Anlagen. Bei leerer Liste erscheint ein Hinweis „Tippe auf + um eine neue Struktur anzulegen“.

#### 4.1 Toolbar-Icons (von links nach rechts)

Icon	Bezeichnung	Aktion
 (Kolben)	Demo laden	Tippt man darauf, werden Beispieldaten geladen
 (Teilen)	JSON exportieren	Alle Strukturen als JSON-Datei teilen (Share-Dialog)
 (Ordner)	Import-Menü	Dropdown öffnet sich (siehe 4.3)
 (Glocke)	Alarme	Öffnet AlarmScreen; rotes Badge zeigt Anzahl unbestätigter Alarme
 (Bug)	Diagnose	Öffnet DiagScreen (Protokoll-Tracer)
 (Zahnrad)	Einstellungen	Öffnet SettingsScreen

#### 4.2 Neue Struktur anlegen

**Weg:** Strukturliste → „+“-Button (blauer runder Button, unten rechts) → StructureFormScreen

#### 4.3 Importieren

**Weg:** Strukturliste → **Ordner-Icon** in der Toolbar → Dropdown erscheint:

Menüpunkt	Aktion
<b>JSON importieren</b>	Dateiauswahl öffnet sich (JSON- oder Textdatei)
<b>JSON (Text) importieren</b>	Dialog mit Textfeld öffnet sich – JSON direkt einfügen/eingeben
<b>EDS importieren</b>	Dateiauswahl für CANopen-EDS-Dateien
<b>GSD importieren</b>	Dateiauswahl für PROFIBUS-GSD-Dateien

Nach erfolgreichem Import erscheint eine kurze Bestätigungs-Snackbar.

#### 4.4 Struktur öffnen

**Weg:** Strukturliste → Auf die Struktur-Karte tippen → StructureDetailScreen

#### 4.5 Struktur bearbeiten

**Weg:** Strukturliste → Stift-Icon (✎) rechts auf der Struktur-Karte → StructureFormScreen

#### 4.6 Struktur löschen

**Weg:** Strukturliste → Papierkorb-Icon (🗑️) rechts auf der Struktur-Karte → Bestätigungsdialog erscheint → „Löschen“ tippen

#### 4.7 Exportieren (JSON)

**Weg:** Strukturliste → Teilen-Icon (↗) in der Toolbar → System-Share-Dialog → Ziel-App wählen

### 5. Strukturdetail (StructureDetailScreen)

**Erreichen:** Strukturliste → Auf Struktur-Karte tippen

Herzstück der App – zeigt alle konfigurierten Details einer Anlage in vier Tabs. Die obere Leiste zeigt: **Name der Struktur + IP:Port**. Der Zurück-Pfeil (←) führt zurück zur Strukturliste.

#### Tab-Navigation

Die vier Tabs sind über die Tab-Leiste unterhalb der Toolbar erreichbar:

Tab	Icon	Funktion
Lesen	↓	Werte lesen, Polling, Aufzeichnung
Schreiben	↑	Werte schreiben
Tabelle	☰	Alle Werte tabellarisch
Dashboard	☐	Visuelle Widgets, Drag-and-Drop

#### 5.1 Tab „Lesen“ – Einzelnen Wert lesen

**Weg:** Strukturdetail → Tab „Lesen“ → Detail-Karte → Button „Lesen“

Schritt	Bedienelement	Beschreibung
1	Tab „Lesen“ antippen	Falls nicht bereits aktiv
2	Button „Alle lesen“ (oben)	Alle Details auf einmal lesen
– oder –		
2	Button „Lesen“ (auf Detail-Karte)	Nur dieses eine Detail lesen
–	Ergebnisbox	Grüner Hintergrund = OK, roter Hintergrund = Fehler

#### 5.2 Tab „Lesen“ – Polling (zyklisches Lesen) starten

**Weg:** Strukturdetail → Tab „Lesen“ → Intervall eingeben → „Polling starten“

Schritt	Bedienelement	Beschreibung
1	Feld „Intervall (s)“	Gewünschte Sekunden

		eingeben (z. B. 5)
2	Toggle „Nur bei Änderung“	Optional: UI nur aktualisieren wenn Wert sich ändert
3	Button „Polling starten“	Tippen → zyklisches Lesen beginnt
—	Status-Banner	Erscheint bei aktivem Polling: „Polling aktiv – alle X s“
4	Button „Stoppen“ (rot)	Polling beenden

### 5.3 Tab „Lesen“ – Aufzeichnung starten/stoppen

**Weg:** Strukturdetail → Tab „Lesen“ → Toggle „Aufzeichnung“

Schritt	Bedienelement	Beschreibung
1	Toggle „Aufzeichnung“	Einschalten → Aufzeichnung startet (rotes Punkt-Icon im Toggle)
—	Status-Text	„Werte werden gespeichert“ bei aktiver Aufzeichnung
2	Toggle „Aufzeichnung“	Ausschalten → Sitzung wird in der Datenbank gespeichert

*Aufgezeichnete Sitzungen sind im LogScreen abrufbar.*

### 5.4 Tab „Schreiben“ – Wert schreiben

**Weg:** Strukturdetail → Tab „Schreiben“ → Wert eingeben → „Schreiben“

Schritt	Bedienelement	Beschreibung
1	Tab „Schreiben“ antippen	
2	Toggle „Schreib-Bestätigung“	Optional: Nach Schreiben automatisch zurücklesen (✓ bei Übereinstimmung)
3	Textfeld „Wert eingeben“	Wert im jeweiligen Detail-Feld eingeben
4	Button „Schreiben“ (auf Detail-Karte)	Nur dieses Detail schreiben
— oder —		
4	Button „Alle schreiben“ (oben)	Alle geänderten Werte auf einmal schreiben

### 5.5 Tab „Tabelle“ – Tabellenansicht

**Weg:** Strukturdetail → Tab „Tabelle“

- Keine Interaktion nötig – zeigt automatisch alle zuletzt gelesenen Werte
- Spalten: **Name** · **Adresse** · **Rohwert** (Hex + Dez) · **Dekodiert** (konvertierter Wert)
- Alternierend eingefärbte Zeilen, Schwellwert-Farbkodierung in der Spalte „Dekodiert“
- Scrollen bei vielen Details

### 5.6 Tab „Dashboard“ – Widgets anzeigen

**Weg:** Strukturdetail → Tab „Dashboard“

- Zeigt Details als visuelle Widgets (LED, Wert, Balken, Diagramm)
- Widgets werden automatisch aus der letzten Leseoperation befüllt

**Widget-Reihenfolge ändern (Drag-and-Drop):**

Schritt	Bedienelement	Beschreibung
1	Widget <b>lange gedrückt halten</b>	Nach ca. 0,5 s beginnt der Drag-Modus
2	Widget <b>nach oben/unten ziehen</b>	Widget wechselt Position sobald Schwellwert überschritten
3	Loslassen	Neue Reihenfolge wird gespeichert

**Widget bearbeiten oder löschen:**

- Stift-Icon (✎) rechts oben auf der Widget-Karte → DetailFormScreen
- Papierkorb-Icon (🗑) rechts oben auf der Widget-Karte → Bestätigungsdialog

### 5.7 Detail hinzufügen

**Weg:** Strukturdetail (Tab Lesen, Schreiben oder Tabelle) → „+“-Button (unten rechts) → DetailFormScreen

*Der FAB ist im Dashboard-Tab nicht sichtbar.*

### 5.8 Detail bearbeiten

**Weg:** Strukturdetail → Detail-Karte → Stift-Icon (✎) → DetailFormScreen

\*(Verfügbar in Tab Lesen, Schreiben und Dashboard)\*

## 5.9 Detail löschen

**Weg:** Strukturdetail → Detail-Karte → Papierkorb-Icon (🗑️) → Bestätigungsdialog → „Löschen“

## 5.10 Struktur bearbeiten

**Weg:** Strukturdetail → Stift-Icon (✎) in der TopAppBar (oben rechts) → StructureFormScreen

## 6. Struktur anlegen/bearbeiten (StructureFormScreen)

**Erreichen (neu):** Strukturliste → „+“-Button

**Erreichen (bearbeiten):** Strukturliste → Stift-Icon auf Karte

**oder:** Strukturdetail → Stift-Icon in der Toolbar

Alle Felder befinden sich auf einem scrollbaren Screen. Der Zurück-Pfeil (←) verwirft Änderungen. Der „**Speichern**“-Button unten speichert und navigiert zurück.

Feld	Pflichtfeld	Beschreibung
Name	✓	Anlagenbezeichnung (Fehler wenn leer)
Beschreibung		Freitextnotiz
IP-Adresse	✓	Standard-IP für alle Details
Port	✓	Standard-Port (Standard: 502)
Unit-ID		Modbus Unit-/Slave-ID (Standard: 1)

## 7. Detail anlegen/bearbeiten (DetailFormScreen)

**Erreichen (neu):** Strukturdetail → „+“-Button

**Erreichen (bearbeiten):** Strukturdetail → Stift-Icon auf Detail-Karte

**oder:** StructureDetailScreen Dashboard → Stift-Icon auf Widget-Karte

Der Screen ist in Abschnitte unterteilt, zwischen denen nach unten gescrollt wird. „**Speichern**“ sichert und navigiert zurück.

### 7.1 Abschnitt „Allgemein“

**Weg:** DetailFormScreen → sofort sichtbar oben

Feld	Beschreibung
------	--------------

Name	Bezeichnung (Pflichtfeld)
Beschreibung	Freitextnotiz

### 7.2 Abschnitt „Verbindung“

**Weg:** DetailFormScreen → nach unten scrollen → Abschnitt „Verbindung“

Feld	Beschreibung
IP-Adresse	Leer lassen = IP der Struktur verwenden
Port	0 lassen = Port der Struktur verwenden
Unit-ID	Modbus Slave-ID für dieses Detail
Registeradresse	Startadresse des Registers (dezimal)
Array-Größe	Anzahl Elemente bei Arrays/Strings

### 7.3 Abschnitt „Datentyp“

**Weg:** DetailFormScreen → nach unten scrollen → Abschnitt „Datentyp“

Feld	Auswahl	Beschreibung
Datentyp	Dropdown	BOOLEAN · BYTE · INT16 · UINT16 · INT32 · UINT32 · INT64 · UINT64 · FLOAT32 · FLOAT64 · STRING · RAW
Funktionscode	Dropdown	FC01–FC2B (Lese- und Schreibfunktionen)
Byte-Reihenfolge	Dropdown	Big Endian (ABCD) · Little Endian (DCBA) · Word Swap (CDAB) · Byte Swap (BADC)

### 7.4 Abschnitt „Konvertierung“

**Weg:** DetailFormScreen → nach unten scrollen → Abschnitt „Konvertierung“

Feld	Beschreibung
Skalierung	Multiplikator: Anzeigewert = (Rohwert × Skalierung) + Offset
Offset	Addition nach Skalierung
Einheit	Angezeigter Text (z. B. °C, bar, A)

## 7.5 Abschnitt „Visualisierung“

**Weg:** DetailFormScreen → nach unten scrollen → Abschnitt „Visualisierung“

Feld	Auswahl / Beschreibung
Widget-Typ	AUTO · Wert · LED · Balken · Diagramm
Balken-Minimum	Unterer Endwert der Balkenanzeige
Balken-Maximum	Oberer Endwert der Balkenanzeige
Warnschwellwert	Ab diesem Wert → orange Einfärbung
Alarmschwellwert	Ab diesem Wert → rote Einfärbung

## 7.6 Abschnitt „Alarmierung“

**Weg:** DetailFormScreen → ganz nach unten scrollen → Abschnitt „Alarmierung“

Feld	Beschreibung
Toggle „Alarm bei Wertänderung“	Erzeugt bei jeder Wertänderung einen Alarm-Event und eine Systembenachrichtigung

## 8. Aufzeichnungen (LogScreen)

**Erreichen:** Strukturliste → **Diagramm-Icon** (  ) in der Toolbar

Zeigt alle aufgezeichneten Messdaten-Sitzungen in einer scrollbaren Liste.

### 8.1 Sitzungen anzeigen

**Weg:** Strukturliste → Diagramm-Icon → LogScreen

Alle Sitzungen werden automatisch aufgelistet (neueste zuerst).

Jeder Listeneintrag zeigt:

- Startdatum und -uhrzeit
- Name der aufgezeichneten Struktur
- Dauer (wenn Sitzung beendet)
- Anzahl gespeicherter Messwerte

## 8.2 Aufzeichnung als CSV exportieren

**Weg:** LogScreen → Sitzung in der Liste → **Teilen-Icon** (↗) auf der Sitzungskarte → Share-Dialog → Ziel-App wählen

### CSV-Format:

```
Zeitstempel;Name;Rohwert;Wert;Einheit  
2024-01-15 14:23:05;Temperatur;1234;12.34;°C
```

## 8.3 Sitzung löschen

**Weg:** LogScreen → Sitzung in der Liste → **Papierkorb-Icon** (🗑️) → Bestätigungsdiallog → „Löschen“

*Das Löschen einer Sitzung entfernt auch alle zugehörigen Messwert-Einträge.*

## 9. Alarme (AlarmScreen)

**Erreichen:** Strukturliste → **Glocken-Icon** (🔔) in der Toolbar

\*(Rotes Zahl-Badge zeigt Anzahl unbestätigter Alarme)\*

### 9.1 Alarmübersicht aufrufen

**Weg:** Strukturliste → Glocken-Icon → AlarmScreen

- **Tab „Alle“:** Alle Alarm-Events chronologisch absteigend
- **Tab „Unbestätigt“:** Nur noch nicht bestätigte Events

**Tab wechseln:** Die beiden Tabs oben antippen.

### 9.2 Alarm bestätigen

**Weg:** AlarmScreen → Alarm-Karte → Button „**Bestätigen**“

- Der Event bleibt erhalten, wird aber aus dem Tab „Unbestätigt“ entfernt
- Das Badge-Zähler in der Strukturliste verringert sich

### 9.3 Alle Alarme bestätigen

**Weg:** AlarmScreen → „**Alle bestätigen**“-Button in der Toolbar (oben rechts)

*Nur sichtbar wenn unbestätigte Events vorhanden sind.*

### 9.4 Einzelnen Alarm löschen

**Weg:** AlarmScreen → Alarm-Karte → Button „**Löschen**“

### 9.5 Alle Alarme löschen

**Weg:** AlarmScreen → „**Alle löschen**“-Button in der Toolbar

## Alarm-Typen und ihre Auslöser:

Typ	Auslöser	Benachrichtigungspriorität
Warnung	Warnschwellwert überschritten	Normal
Alarm	Alarmschwellwert überschritten	Hoch
Wertänderung	Wert hat sich geändert (Toggle aktiviert)	Niedrig
Verbindungsverlust	Kein Gerät mehr erreichbar	Hoch

## 10. Diagnose (DiagScreen)

**Erreichen:** Strukturliste → **Bug-Icon** (  ) in der Toolbar

Protokollanalyse und Statistik aller Modbus-Kommunikation. Zeichnet automatisch auf wenn aktiviert.

### 10.1 Aufzeichnung starten/stoppen

**Weg:** DiagScreen → **Play/Pause-Icon** in der Toolbar (oben rechts)

- Bei aktiver Aufzeichnung: rotes „REC“-Badge im Titel sichtbar
- Der Ringpuffer speichert die letzten 500 Frames

### 10.2 Statistiken anzeigen

**Weg:** DiagScreen → Tab „**Statistik**“ (Standard-Tab)

Zeigt automatisch:

- Gesamt-Transaktionen · Erfolge · Fehler
- Erfolgsrate in %
- Latenz: Durchschnitt / Min / Max / P95
- Fehler-Aufschlüsselung: Timeouts · CRC-Fehler · Exceptions · Sonstige

### 10.3 Einzelne Frames / Hex-Dump anzeigen

**Weg:** DiagScreen → Tab „**Frames**“ antippen

Schritt	Bedienelement	Beschreibung
1	Tab „Frames“ antippen	Liste aller aufgezeichneten Frames
2	Filter-Chips (Alle / OK / Fehler)	Frames filtern
3	Frame-Karte antippen	Klappt auf → zeigt Hex-Dump + MBAP-Info

## 10.4 PCAP-Datei exportieren (für Wireshark)

**Weg:** DiagScreen → **PCAP-Icon** in der Toolbar → Share-Dialog → Ziel-App wählen

*Die exportierte \.pcap`-Datei enthält valide Ethernet + IPv4 + TCP + Modbus-MBAP-Header und kann direkt in Wireshark geöffnet werden.*

## 10.5 Frame-Buffer leeren

**Weg:** DiagScreen → **Papierkorb-Icon** in der Toolbar → Bestätigungsdialo → „Löschen“

## 11. Slave-Modus (SlaveScreen)

**Erreichen:** Bottom Navigation → **Memory-Icon** (Slave)

Betreibt die App als Modbus TCP-Server. Nützlich zum Testen von SPS-Programmen oder Visualisierungssoftware.

### 11.1 Slave-Server starten

**Weg:** SlaveScreen → Tab „**Server**“ → Port und Unit-ID eingeben → Button „**Server starten**“

Schritt	Bedienelement	Beschreibung
1	Feld „ <b>Port</b> “	TCP-Port eingeben (Standard: 502)
2	Feld „ <b>Unit-ID</b> “	Gewünschte Unit-ID eingeben
3	Button „ <b>Server starten</b> “	Tippen → Server lauscht auf eingehende Verbindungen
—	Status-Badge	„ <b>AKTIV :502</b> “ erscheint im Titel wenn aktiv
4	Button „ <b>Server stoppen</b> “	Tippen → Server wird beendet

### 11.2 Response-Delay konfigurieren

**Weg:** SlaveScreen → Tab „**Server**“ → Slider „**Response-Delay**“

- Slider von 0 bis 5000 ms
- Verzögerung gilt für alle Antworten (simuliert langsame Geräte)

### 11.3 Exception-Antworten erzwingen

**Weg:** SlaveScreen → Tab „**Server**“ → Abschnitt „**Exception simulieren**“

Schritt	Bedienelement	Beschreibung
1	Toggle „ <b>Exception aktivieren</b> “	Einschalten
2	Dropdown „ <b>Exception-Code</b> “	01 Illegal Function · 02 Illegal Data Address · 03 Illegal Data Value · 04 Server Failure
3	Radio „ <b>Ziel</b> “	Alle Adressen oder nur

		Adressbereich
4	Felder „Von“ / „Bis“	Bei Adressbereich: Bereich eingeben

#### 11.4 Holding-Register bearbeiten

Weg: SlaveScreen → Tab „Register“

Schritt	Bedienelement	Beschreibung
1	Felder „Von Adresse“ / „Anzahl“	Anzuzeigenden Bereich einschränken
2	Register-Zeile antippen	Wert direkt inline editieren
3	Haken-Button	Tippen → Wert wird übernommen

Tabelle zeigt: Adresse · Wert in Hex · Wert in Dezimal

#### 11.5 Coils bearbeiten

Weg: SlaveScreen → Tab „Coils“

- 2-spaltige Übersicht von 16 Coils
- Pro Coil: LED-Anzeige (grün = 1, grau = 0) + Switch zum Umschalten
- Switch kippen → Coil-Zustand wechselt sofort

#### 11.6 Wertegenerator verwenden

Weg: SlaveScreen → Tab „Generator“

Schritt	Bedienelement	Beschreibung
1	Dropdown „Modus“	OFF · RANDOM · RAMP · SINE
2	Feld „Zieladresse“	Startadresse der Register
3	Felder „Min“ / „Max“	Wertebereich
4	Feld „Periode (s)“	Periodendauer bei RAMP/SINE
5	Feld „Schrittgröße“	Inkrement pro Tick (RAMP)
6	Feld „Intervall (ms)“	Aktualisierungsintervall
7	Button „Generator starten“	Tippen → Register werden automatisch befüllt
8	Button „Generator stoppen“	Tippen → Generator anhalten

## 12. NFC (NfcScreen)

Erreichen: Einstellungen (SettingsScreen) → Button „NFC“

Überträgt Verbindungsprofile (Name, IP, Port, Unit-ID) per NFC-Tag.

*Voraussetzung: Das Android-Gerät muss NFC unterstützen und NFC muss in den Android-Systemeinstellungen aktiviert sein.*

### 12.1 Verbindungsprofil auf NFC-Tag schreiben

**Weg:** SettingsScreen → NFC-Button → NfcScreen → Tab „Schreiben“

Schritt	Bedienelement	Beschreibung
1	Tab „Schreiben“ antippen	
2	Dropdown „Struktur“	Zu übertragende Struktur auswählen
3	Button „Auf Tag schreiben“	Tippen → NFC-Reader-Modus wird aktiviert
4	NFC-Tag ans Gerät halten	Sobald erkannt → Profil wird auf den Tag geschrieben
—	Status-Meldung	Erfolg oder Fehlermeldung erscheint

### 12.2 Verbindungsprofil von NFC-Tag lesen

**Weg:** SettingsScreen → NFC-Button → NfcScreen → Tab „Lesen“

Schritt	Bedienelement	Beschreibung
1	Tab „Lesen“ antippen	NFC-Reader ist automatisch aktiv
2	NFC-Tag ans Gerät halten	Profil wird automatisch ausgelesen
—	Anzeige	Name + IP:Port des gelesenen Profils erscheint

## 13. QR-Code (QrScreen)

**Erreichen:** Einstellungen (SettingsScreen) → Button „QR-Code“

Teilt Verbindungsprofile als QR-Code (JSON-Inhalt).

### 13.1 QR-Code eines Verbindungsprofils anzeigen

**Weg:** SettingsScreen → QR-Code-Button → QrScreen → Tab „Anzeigen“

Schritt	Bedienelement	Beschreibung
1	Tab „Anzeigen“ antippen	
2	Dropdown „Struktur“	Struktur auswählen

—	QR-Code	Wird sofort generiert (512×512 Pixel)
3	Button „Exportieren“	QR-Code-Bild per Share-Intent teilen

### 13.2 QR-Code scannen und Struktur importieren

**Weg:** SettingsScreen → QR-Code-Button → QrScreen → Tab „Scannen“

Schritt	Bedienelement	Beschreibung
1	Tab „Scannen“ antippen	
2	Kamera-Dialog	Öffnet sich automatisch (ZXing-Scanner)
3	Kamera auf QR-Code richten	
—	Automatisch	Struktur wird aus dem gescannten JSON importiert

## 14. Einstellungen (SettingsScreen)

**Erreichen:** Strukturliste → Zahnrad-Icon (⚙) in der Toolbar

### 14.1 Farbschema ändern

**Weg:** SettingsScreen → Abschnitt „Erscheinungsbild“ → FilterChips antippen

Option	Beschreibung
Hell	Helles Material-3-Theme
Dunkel	Dunkles Material-3-Theme
System	Folgt der Android-Systemeinstellung (Standard)

*Änderung wird sofort ohne Neustart übernommen.*

### 14.2 Sprache ändern

**Weg:** SettingsScreen → Abschnitt „Sprache“ → Dropdown antippen → Sprache wählen

Option	Beschreibung
Deutsch (de)	Deutsche Oberfläche
English (en)	Englische Oberfläche

*Änderung wird sofort ohne Neustart übernommen.*

### 14.3 Homescreen-Widget konfigurieren

**Weg:** SettingsScreen → Abschnitt „Homescreen-Widget“ → Dropdown „Struktur“

- Wählt aus, welche Struktur auf dem Homescreen-Widget angezeigt wird
- Das Widget aktualisiert sich nach jedem Lese-Zyklus der gewählten Struktur

### 14.4 NFC-Screen öffnen

**Weg:** SettingsScreen → Button „NFC“ → NfcScreen

### 14.5 QR-Screen öffnen

**Weg:** SettingsScreen → Button „QR-Code“ → QrScreen

## 15. Homescreen-Widget

**Einrichten:** Android-Homescreen → Widget hinzufügen → „ModbusApp“ wählen → Widget platzieren

**Konfigurieren:** SettingsScreen → Abschnitt „Homescreen-Widget“ → Struktur auswählen

Das Widget zeigt auf dem Homescreen:

- **Strukturname** (oben links, fett)
- **Letzter Messwert** (groß, fett)
- **Zeitstempel** der letzten Aktualisierung (unten, hellblau)

**Hintergrundfarbe:** Blau (#1565C0)

**Aktualisierung:** Automatisch nach jedem Polling-Zyklus der gewählten Struktur – kein manuelles Eingreifen nötig.

## 16. Wear OS

**Einrichten:** Wear-OS-Smartwatch mit dem Android-Gerät koppeln → ModbusApp läuft automatisch auf der Uhr sobald Daten gesendet werden.

Die Wear-OS-App empfängt Messwerte automatisch nach jedem Lese-Zyklus:

**Anzeigeelemente auf der Uhr:**

- Zeitstempel der letzten Aktualisierung (oben)
- Scrollbare Liste aller Readings (ScalingLazyColumn)
- Pro Eintrag: **Name** (klein) · **Wert** (groß) · **Einheit**
- „Keine Daten“ Platzhalter wenn noch keine Werte gesendet wurden

**Datentransfer:** Wearable Data API, Pfad `/modbus/readings` – wird nach jeder erfolgreichen `readAll()`-Operation automatisch ausgelöst.

## 17. Datenmodelle

### ModbusStructure

```

id          - UUID (automatisch generiert)
name       - Anlagename
description - Beschreibung
ip         - Standard-IP-Adresse (Standard: 192.168.1.1)
port      - Standard-Port (Standard: 502)
unitId    - Standard Unit-ID (Standard: 1)
details   - Liste der ModbusDetail-Einträge
    
```

### ModbusDetail

```

id          - UUID
name       - Bezeichnung
description - Beschreibung
dataType   - Datentyp (12 Optionen)
arraySize  - Array-Größe
ip / port / unitId - Verbindungsüberschreibung (leer/0 = Struktur nutzen)
registerAddress - Modbus-Registeradresse
functionCode - FC01-FC2B
byteOrder   - Big/Little/WordSwap/ByteSwap
scale / offset - Skalierung: (Rohwert × scale) + offset = Anzeigewert
unit       - Einheitentext
widgetType - AUTO / VALUE / LED / GAUGE / CHART
gaugeMin/Max - Bereich für Balkenanzeige
thresholdWarn - Warnschwelle (optional)
thresholdAlarm - Alarmschwelle (optional)
alarmOnChange - Alarm bei jeder Wertänderung
    
```

### Datentypen

Typ	Beschreibung	Register
BOOLEAN	1 Bit (Coil)	1
BYTE	8 Bit	1
INT16	16 Bit vorzeichenbehaftet	1
UINT16	16 Bit vorzeichenlos	1
INT32	32 Bit vorzeichenbehaftet	2
UINT32	32 Bit vorzeichenlos	2
INT64	64 Bit vorzeichenbehaftet	4
UINT64	64 Bit vorzeichenlos	4
FLOAT32	IEEE 754 einfache Genauigkeit	2
FLOAT64	IEEE 754 doppelte Genauigkeit	4
STRING	ASCII (1 Byte/Register)	variabel
RAW	Rohregister 16 Bit	1

## 18. Modbus-Protokollimplementierung

### Verbindungstypen (ModbusConnection)

#### TcpConnection

- Neue TCP-Socket pro Request
- MBAP-Framing: Transaction-ID (2 Byte) · Protocol-ID (0x0000) · Länge (2 Byte) · Unit-ID · PDU
- Timeout: 3000 ms

#### UsbSerialConnection

- USB-OTG über `hoho-android-usbserial`
- Konfigurierbar: Baudrate, Datenbits, Parität, Stoppbits
- DTR/RTS aktiviert

#### BluetoothConnection

- SPP-Profil (UUID: 00001101-0000-1000-8000-00805F9B34FB)
- Polling-basiertes Empfangen (10 ms Delay)

#### RtuWifiConnection

- TCP-Socket zu Serial-Brücke
- RTU-Framing (CRC-16) über TCP

### RTU-Framing (ModbusRtu)

```
CRC-16: CCITT Polynomial 0xA001  
Frame: [Unit-ID] [PDU...] [CRC-Lo] [CRC-Hi]
```

### Unterstützte Funktionscodes

#### Lesen:

- FC01 (0x01): Read Coils
- FC02 (0x02): Read Discrete Inputs
- FC03 (0x03): Read Holding Registers
- FC04 (0x04): Read Input Registers
- FC07 (0x07): Read Exception Status
- FC0B (0x0B): Get Comm Event Counter
- FC0C (0x0C): Get Comm Event Log
- FC11 (0x11): Report Server ID
- FC18 (0x18): Read FIFO Queue
- FC2B (0x2B): Read Device ID (MEI 0x0E)

#### Schreiben:

- FC05 (0x05): Write Single Coil
- FC06 (0x06): Write Single Register

- FC08 (0x08): Diagnostics
- FC0F (0x0F): Write Multiple Coils
- FC10 (0x10): Write Multiple Registers
- FC17 (0x17): Read/Write Multiple Registers

### Byte-Reihenfolgen

Bezeichnung	Reihenfolge	Beispiel (Wert 0x12345678)	
Big Endian (ABCD)	Standard Netzwerk	12 34 56 78	
Little Endian (DCBA)	Intel-Format	78 56 34 12	
Word Swap (CDAB)	Mitsubishi-Format	Mitsubishi	56 78 12 34
Byte Swap (BADC)	PLC-Format	34 12 78 56	

### Dateiimport

#### EDS (CANopen Electronic Data Sheet):

- INI-Format: [Device] → Geräte name, [Param###] → Parameterliste
- Mappt automatisch auf Modbus-Details mit korrekten Datentypen

#### GSD (PROFIBUS General Station Description):

- Key-Value-Format: Vendor\_Name, Model\_Name, Module "Name" data
- Erstellt Detail-Einträge pro Modul

## 19. Datenbank

Room-Datenbank (Version 2) mit drei Tabellen:

### log\_sessions

Aufzeichnungssitzungen pro Struktur

```

id          - UUID (PK)
structureId - Referenz zur Struktur
structureName - Name zum Zeitpunkt der Aufzeichnung
startTime   - Unix-Timestamp
endTime     - Unix-Timestamp (null wenn aktiv)

```

### log\_entries

Einzelne Messwerte einer Sitzung

```

id          - UUID (PK)
sessionId   - Referenz zur Sitzung
structureId - Referenz zur Struktur
detailId    - Referenz zum Detail
detailName  - Name zum Zeitpunkt der Aufzeichnung
timestamp   - Unix-Timestamp
rawValue    - Rohwert als String (kommagetrennt bei Arrays)

```

displayValue - Konvertierter Wert mit Einheit  
 unit - Einheit

## alarm\_events

### Alarm-Events und deren Bestätigungsstatus

id - UUID (PK)  
 timestamp - Unix-Timestamp  
 type - THRESHOLD\_WARN / THRESHOLD\_ALARM / VALUE\_CHANGE / CONNECTION\_LOST  
 structureId - Referenz zur Struktur  
 structureName - Name zum Zeitpunkt des Alarms  
 detailId - Referenz zum Detail (optional)  
 detailName - Name zum Zeitpunkt des Alarms (optional)  
 value - Auslösender Wert (optional)  
 message - Beschreibung des Alarms  
 acknowledged - Boolean (false = unbestätigt)

## 20. Berechtigungen

Berechtigung	Zweck
INTERNET	Modbus TCP-Verbindungen
ACCESS_NETWORK_STATE	Netzwerkstatus prüfen
BLUETOOTH / BLUETOOTH_ADMIN	Bluetooth (API ≤ 30)
BLUETOOTH_CONNECT	Bluetooth-Verbindung (API 31+)
BLUETOOTH_SCAN	Bluetooth-Gerätescan (API 31+)
NFC	NFC-Tags lesen/schreiben
POST_NOTIFICATIONS	Alarm-Benachrichtigungen (API 33+)
USB_HOST (Feature)	USB-OTG-Kommunikation
NFC (Feature)	NFC (als optional markiert)

*\*\*Hinweis:\*\* USB-Serial-Kommunikation erfordert keine explizite Laufzeit-Berechtigung – die Erlaubnis wird über einen System-Dialog für das jeweilige USB-Gerät erteilt. Bluetooth-Berechtigungen werden beim ersten Scan automatisch angefragt. Post-Notifications-Berechtigung wird beim ersten Alarm automatisch angefragt (Android 13+).*

\*Dokumentation aktualisiert am 2026-05-21 · Modbus++ v1.0\*