

# Profinet++ – Funktionsdokumentation

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

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

**Protokolle:** S7/ISO-on-TCP · PROFINET TCP · DCP Device Discovery (UDP)

## Inhaltsverzeichnis

1. [App-Überblick & Einstieg](#1-app-überblick--einstieg)
2. [Navigation & Menüstruktur](#2-navigation--menüstruktur)
3. [Hauptbildschirm (MainScreen)](#3-hauptbildschirm-mainscreen)
4. [Geräteliste (DeviceListScreen)](#4-geräteliste-devicelistscreen)
5. [Gerätedetail (DeviceDetailScreen)](#5-gerätedetail-devicedetailscreen)
6. [Gerät anlegen/bearbeiten (DeviceFormScreen)](#6-gerät-anlegenbearbeiten-deviceformscreen)
7. [Modul anlegen/bearbeiten (ModuleFormScreen)](#7-modul-anlegenbearbeiten-moduleformscreen)
8. [DCP-Scan (ScanScreen)](#8-dcp-scan-scanscreen)
9. [Aufzeichnungen (LogScreen)](#9-aufzeichnungen-logscreen)
10. [Alarmer (AlarmScreen)](#10-alarmer-alarmscreen)
11. [Diagnose (DiagScreen)](#11-diagnose-diagscreen)
12. [IO-Device-Modus (IoDeviceScreen)](#12-io-device-modus-iodevicescreen)
13. [NFC (NfcScreen)](#13-nfc-nfcscreen)
14. [QR-Code (QrScreen)](#14-qr-code-qrscreen)
15. [GSDML-Viewer (GsdmlViewerScreen)](#15-gsdml-viewer-gsdmlviewerscreen)
16. [Einstellungen (SettingsScreen)](#16-einstellungen-settingscreen)
17. [Homescreen-Widget](#17-homescreen-widget)
18. [Wear OS](#18-wear-os)
19. [Datenmodelle](#19-datenmodelle)
20. [PROFINET-Protokollimplementierung](#20-profinet-protokollimplementierung)
21. [DCP Device Discovery](#21-dcp-device-discovery)
22. [Datenbank](#22-datenbank)
23. [Berechtigungen](#23-berechtigungen)

## 1. App-Überblick & Einstieg

Profinet++ ist eine vollständige PROFINET IO-Controller/IO-Supervisor-Anwendung für Android. Sie unterstützt den Zugriff auf PROFINET-Geräte (SPS, Antriebe, Feldgeräte) und bietet Datenvisualisierung, Aufzeichnung, Alarmierung und Diagnose.

**Zwei Einstiegswege beim App-Start:**

Einstieg	Beschreibung
Hauptbildschirm	Direktzugriff auf PROFINET ohne Konfiguration – für schnelle Tests
Geräteliste	Verwaltung gespeicherter Geräte/Anlagen – für strukturierte Arbeit

#### Unterstützte Verbindungsarten:

Typ	Beschreibung	Standard-Port
S7/ISO-on-TCP	Siemens S7-300/400/1200/1500 und kompatible Geräte	102
PROFINET TCP	Generische PROFINET-Gateways mit TCP-Support	34964
OPC UA	OPC UA Server (vorbereitet, Erweiterung geplant)	4840

#### Wichtige technische Einschränkung:

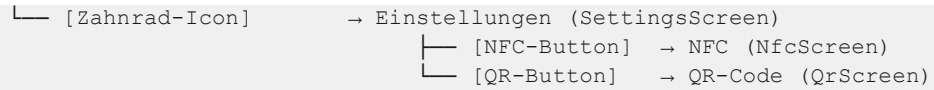
PROFINET Real-Time (RT/IRT) erfordert Layer-2-Ethernet-Zugriff (EtherType 0x8892), der auf Standard-Android **nicht** verfügbar ist. Profinet++ implementiert daher:

- **S7Comm** über ISO-on-TCP (RFC 1006, TCP Port 102) – für Siemens S7/PROFINET PLCs
- **PROFINET Application Layer Protocol** über TCP (für Gateways)
- **DCP** über UDP (vereinfachter Discovery-Ansatz für kompatible Gateways)

## 2. Navigation & Menüstruktur

Die App startet auf dem **Hauptbildschirm (MainScreen)**. Alle weiteren Screens werden von der **Geräteliste** aus erreicht.





### 3. Hauptbildschirm (MainScreen)

**Erreichen:** App starten – der Hauptbildschirm erscheint automatisch.

Ermöglicht schnellen Zugriff auf PROFINET ohne vorherige Geräte-Konfiguration.

#### 3.1 Verbindungseinstellung

Feld	Beschreibung
Connection Type	Dropdown: S7/ISO-on-TCP · PROFINET TCP · OPC UA
IP Address	IP-Adresse des PROFINET-Geräts
Port	TCP-Port (wird bei Connection-Type-Wahl automatisch gesetzt)
Rack	S7 Rack-Nummer (Standard: 0)
Slot (CPU)	S7 Slot-Nummer der CPU (Standard: 1)

#### 3.2 Daten lesen

Schritt	Bedienelement	Beschreibung
1	Dropdown „Area“	Speicherbereich: I · Q · M · DB · T · C · Record
2	Feld „DB#“	Datenbaustein-Nummer (nur bei Area=DB)
3	Feld „Byte offset“	Startbyte im Bereich
4	Feld „Count“	Anzahl Bytes zum Lesen
5	Button „Read“	Liest Bytes und zeigt Hex-Dump

#### 3.3 Daten schreiben

Schritt	Bedienelement	Beschreibung
1	Dropdown „Area“	Zielbereich auswählen
2	Felder DB#/Offset	Zieladresse
3	Textfeld „Hex bytes“	Zu schreibende Bytes als Hex, z. B. „00 FF 01 A5“
4	Button „Write“	Schreibt Bytes ans Gerät

### 3.4 Protocol Log










- Zeigt die letzten 20 Kommunikationsereignisse
- Zeitstempel im Format HH:mm:ss.SSS
- READ/WRITE + Hex-Dump

## 4. Geräteliste (DeviceListScreen)

**Erreichen:** Hauptbildschirm → List-Icon oben rechts

Die Geräteliste zeigt alle konfigurierten PROFINET-Geräte. Bei leerer Liste erscheint ein Hinweis.

### 4.1 Toolbar-Icons

Icon	Bezeichnung	Aktion
 (Kolben)	Demo laden	Lädt ein Demo-Gerät (S7-1200 Simulation)
 (Teilen)	JSON exportieren	Exportiert alle Geräte als JSON (Share-Dialog)
 (Ordner)	Import-Menü	JSON oder GSDML-Datei importieren
 (Radar)	DCP-Scan	Öffnet ScanScreen (DCP Device Discovery)
 (Glocke)	Alarmer	Öffnet AlarmScreen; Badge zeigt unbestätigte Alarmer
 (Bug)	Diagnose	Öffnet DiagScreen (Protokoll-Tracer)
 (Diagramm)	Aufzeichnungen	Öffnet LogScreen
 (Server)	IO-Device	Öffnet IoDeviceScreen
 (Zahnrad)	Einstellungen	Öffnet SettingsScreen

### 4.2 Neues Gerät anlegen

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

### 4.3 Importieren

Menüpunkt	Aktion
Import JSON	Dateiauswahl (JSON-Backup)
Import GSDML (XML)	GSDML-Dateiauswahl – Gerät wird automatisch mit Modulen angelegt

## 5. Gerätedetail (DeviceDetailScreen)

**Erreichen:** Geräteliste → Auf Geräte-Karte tippen

Herzstück der App – zeigt alle konfigurierten Module eines PROFINET-Geräts in vier Tabs.

### Tab-Navigation

Tab	Funktion
Read	Werte lesen, Polling, Aufzeichnung
Write	Werte schreiben (Output-Module)
Table	Alle Werte tabellarisch (Name / Adresse / Wert)
Dashboard	Visuelle Widgets (LED, Gauge, Value)

### 5.1 Tab „Read“ – Wert lesen

Schritt	Bedienelement	Beschreibung
1	Button „Read All“	Alle Module auf einmal lesen
– oder –		
1	Button ▼ (Read) auf Modul-Karte	Nur dieses Modul lesen
–	Karten-Hintergrund	Grün = OK, Rot = Fehler

### 5.2 Polling starten/stoppen

Schritt	Bedienelement	Beschreibung
1	Feld „Interval (s)“	Sekunden eingeben
2	Button „Start Polling“	Zyklisches Lesen beginnt
–	Status-Banner	„Polling active – every X s“
3	Button „Stop“	Polling beenden

### 5.3 Aufzeichnung

Schritt	Bedienelement	Beschreibung
1	Toggle „Recording“	An = Aufzeichnung startet
2	Toggle „Recording“	Aus = Sitzung wird gespeichert

### 5.4 Tab „Write“ – Wert schreiben

Zeigt nur Module mit IO-Richtung Output oder Bidirectional.

Schritt	Bedienelement	Beschreibung
1	Textfeld „New value“	Wert eingeben (passend zum Datentyp)

2	Button „Write“	Wert wird ans Gerät geschrieben
---	----------------	---------------------------------

## 6. Gerät anlegen/bearbeiten (DeviceFormScreen)

**Pflichtfelder:** Name, IP-Adresse

Feld	Beschreibung
Device Name	Frei wählbarer Name
Description	Optionale Beschreibung
Connection Type	S7/ISO-on-TCP · PROFINET TCP · OPC UA
IP Address	IP-Adresse des Geräts
Port	TCP-Port (wird automatisch gesetzt)
Rack	S7 Rack-Nummer (nur bei S7 TCP)
Slot (CPU)	S7 Slot-Nummer der CPU (nur bei S7 TCP)
Name of Station	PROFINET NameOfStation (optional, für DCP)

## 7. Modul anlegen/bearbeiten (ModuleFormScreen)

Module sind die PROFINET-Äquivalente von Modbus-Registern.

### Adressierung

Area	Adressierung	Beispiele
<b>I (Inputs)</b>	Byte-Offset · Bit-Offset	I0.0, I1.3
<b>Q (Outputs)</b>	Byte-Offset · Bit-Offset	Q0.0, Q2.7
<b>M (Merker)</b>	Byte-Offset · Bit-Offset	M10.0, M20.1
<b>DB (Data Block)</b>	DB-Nummer · Byte-Offset · Bit-Offset	DB1.DBB0, DB2.DBW4
<b>T (Timer)</b>	Timer-Nummer	T0, T5
<b>C (Counter)</b>	Counter-Nummer	C0, C10
<b>Record (Acyclic)</b>	Slot · Subslot · Index	Slot0/Sub1/Idx0x0030

### Datentypen (IEC 61131-3 / PROFINET)

Typ	Größe	Beschreibung
BOOL	1 Bit	Boolescher Wert
BYTE	1 Byte	8-Bit unsigned
WORD	2 Byte	16-Bit unsigned

DWORD	4 Byte	32-Bit unsigned
LWORD	8 Byte	64-Bit unsigned
INT	2 Byte	16-Bit signed
DINT	4 Byte	32-Bit signed
LINT	8 Byte	64-Bit signed
UINT	2 Byte	16-Bit unsigned (alias WORD)
UDINT	4 Byte	32-Bit unsigned (alias DWORD)
REAL	4 Byte	IEEE 754 Single
LREAL	8 Byte	IEEE 754 Double
STRING	variabel	S7-String (Length-Byte + MaxLen-Byte + Chars)
CHAR	1 Byte	Einzelnes Zeichen
TIME	4 Byte	Millisekunden
DATE	2 Byte	Tage seit 1990-01-01
RAW	variabel	Hex-Dump

## 8. DCP-Scan (ScanScreen)

**Erreichen:** Geräteliste → Radar-Icon

DCP (Discovery and Configuration Protocol) sucht PROFINET-Geräte im Netzwerk.

Schritt	Bedienelement	Beschreibung
1	Feld „Broadcast Address“	Standard: 255.255.255.255 oder Subnetz-Broadcast
2	Button „Start DCP Scan (3s)“	UDP-Broadcast wird gesendet, 3s auf Antworten warten
—	Geräteliste	Gefundene Geräte erscheinen mit IP, MAC, NameOfStation
3	Button + ( <b>AddCircle</b> )	Gerät zur Geräteliste hinzufügen

**Antwortfelder je gefundenem Gerät:**


- IP-Adresse, MAC-Adresse
- NameOfStation
- VendorID, DeviceID (Hex)
- DeviceRole (IO-Device / IO-Controller / IO-Supervisor)
- TypeOfStation

**Hinweis:** Echtes DCP läuft auf Layer-2 mit Multicast (EtherType 0x8892). Diese Implementierung verwendet UDP-Broadcast. Kompatibilität hängt vom PROFINET-Gateway oder -Switch ab.

## 9. Aufzeichnungen (LogScreen)

**Erreichen:** Geräteliste → Diagramm-Icon

Verwaltet alle aufgezeichneten Messsitzungen.

Icon	Aktion
 (DeleteSweep)	Alle Sitzungen löschen
▼ (ExpandMore)	Sitzungseinträge anzeigen
↓ (FileDownload)	Als CSV exportieren
 (Delete)	Diese Sitzung löschen


**CSV-Format:**

```
Timestamp, DeviceID, ModuleID, ModuleName, RawValue, DisplayValue
1703000000000, uuid1, uuid2, Temperature, 41A00000, 20.0 °C
```

## 10. Alarmer (AlarmScreen)

**Erreichen:** Geräteliste → Glocke-Icon

Alarm-Typ	Farbe	Auslöser
THRESHOLD_ALARM	Rot	Messwert ≥ Alarm-Schwelle
THRESHOLD_WARN	Gelb	Messwert ≥ Warn-Schwelle
CONNECTION_LOST	Grau	Kommunikationsfehler
VALUE_CHANGE	Blau	Wert hat sich geändert (bei alarmOnChange=true)
PROFINET_DIAGNOSIS	Lila	PROFINET-Diagnosemeldung

Icon	Aktion
✓ (Check)	Einzelnen Alarm bestätigen
✓✓ (DoneAll)	Alle bestätigen
 (DeleteSweep)	Alle löschen

## 11. Diagnose (DiagScreen)

**Erreichen:** Geräteliste → Bug-Icon

Protokoll-Tracer für PROFINET-Kommunikation mit 2-Tab-Layout.

**Tab 1 – Frames**

Element	Beschreibung
REC-Badge	Blinkendes rotes Abzeichen im Titel während der Aufzeichnung
Play/Pause-Button	Aufzeichnung starten/anhalten
Filter-Chips	Alle · OK · Fehler – filtert die angezeigte Frame-Liste
Frame-Karte	Zeigt Zeitstempel, Richtung (TX/RX), Länge, Latenz, Statusfarbe
Hex-Dump	Antippen zum Aufklappen: Wireshark-Stil (Offset · Hex · ASCII)
ClearAll-Button	Alle aufgezeichneten Frames löschen
PCAP-Export	Frames als <code>.pcap</code> -Datei exportieren (Wireshark-kompatibel)

**Ringpuffer:** Maximal 500 Frames; älteste werden bei Überlauf verworfen.

**Tab 2 – Statistik**

Element	Beschreibung
Überblick	Gesamt-Frames, Fehler-Frames, Erfolgsrate
Fehleraufschlüsselung	Timeouts vs. sonstige Fehler
Latenz-Sparkline	Canvas-Diagramm der letzten Latenzwerte; rote Linie bei 200 ms Schwellenwert

## 12. IO-Device-Modus (IoDeviceScreen)

**Erreichen:** Geräteliste → Server-Icon

Simuliert ein PROFINET IO-Device (Feldgerät).

**Hinweis:** Vollständiger PROFINET IO-Device-Support erfordert PROFINET-RT auf Layer 2, der auf Standard-Android-Hardware nicht verfügbar ist. Dieser Modus ist eine Simulation zur Demonstration des IO-Device-Konzepts.

Bereich	Beschreibung
Input Data	Prozessdaten dieses Geräts (→ IO-Controller),

	konfigurierbar als Hex-Bytes
<b>Output Data</b>	Empfangene Daten vom IO-Controller

### 13. NFC (NfcScreen)

**Erreichen:** Einstellungen → NFC-Button

Schreibt die Verbindungskonfiguration eines Geräts auf ein NFC-Tag.

- **Format:** `profinet://<IP>:<Port>/<NameOfStation>`
- **MIME-Type:** `application/io.github.erginmusa.profinetplusplus.connection`

### 14. QR-Code (QrScreen)

**Erreichen:** Einstellungen → QR-Button

Zeigt einen QR-Code mit der Verbindungskonfiguration an. Verwendung zum schnellen Teilen von Gerätekonfigurationen.

### 15. GSDML-Viewer (GsdmlViewerScreen)

**Erreichen:** Wird nach GSDML-Import oder über direkten Link geöffnet

Zeigt die importierten Gerätedaten im strukturierten Format an:

- Geräte-Identität (VendorID, DeviceID, NameOfStation)
- Alle konfigurierten Module mit Adresse, Datentyp, Richtung
- Skalierung, Einheit, Alarmschwellen

### 16. Einstellungen (SettingsScreen)

Abschnitt	Einstellung	Beschreibung
Appearance	Dark Mode	Helles/dunkles Theme
Notifications	Enable Alarm Notifications	Push-Benachrichtigungen bei Alarmen
MQTT Bridge	Enable MQTT	PROFINET-Werte an MQTT-Broker senden
MQTT Bridge	Broker URL	z. B. <code>tcp://broker.hivemq.com:1883</code>
MQTT Bridge	Topic	z. B. <code>profinet/data</code>
Security	Biometric Lock	App-Start mit

		Fingerabdruck/Gesicht sichern
Security	License Key	Profinet++ Pro-Lizenzschlüssel (Format: PFNET-PRO-...)
Share & Transfer	NFC	Öffnet NFC-Screen
Share & Transfer	QR Code	Öffnet QR-Screen

## 17. Homescreen-Widget

Widget zeigt den App-Namen und kann um Live-Werte erweitert werden.

### Klassen:

- `ProfinetWidget` (`GlanceAppWidget`)
- `ProfinetWidgetReceiver` (`GlanceAppWidgetReceiver`)

## 18. Wear OS

Grundlegendes Wear OS Companion App-Gerüst vorhanden.

### Unterstützte Funktionen (vorbereitet):

- Werte vom Phone-Modul via Wearable Data Layer empfangen
- Schreibbefehle an Phone-App senden (Pfad: `/profinet/write`)

## 19. Datenmodelle

### ProfinetDevice

Feld	Typ	Beschreibung
id	String	UUID
name	String	Anzeigename
description	String	Optionale Beschreibung
ip	String	IP-Adresse
port	Int	TCP-Port
nameOfStation	String	PROFINET NameOfStation
vendorId	Int	PROFINET VendorID
deviceId	Int	PROFINET DeviceID
rack	Int	S7 Rack (0..7)
slot	Int	S7 CPU-Slot (0..31)
connectionType	ProfinetConnectionType	S7_TCP / PROFINET_TCP /

		OPCUA_TCP
modules	List<ProfinetModule>	Konfigurierte Module

### ProfinetModule

Feld	Typ	Beschreibung
id	String	UUID
name	String	Anzeigename
area	ProfinetArea	I / Q / M / DB / T / C / RECORD
dbNumber	Int	DB-Nummer (bei area=DB)
byteOffset	Int	Byte-Adresse im Bereich
bitOffset	Int	Bit-Offset (0-7, nur für BOOL)
arraySize	Int	Array-Größe / Stringlänge
slot	Int	PROFINET Slot (bei area=RECORD)
subslot	Int	PROFINET Subslot
recordIndex	Int	Record Data Index
dataType	ProfinetDataType	BOOL/BYTE/WORD/INT/REAL/...
ioType	ProfinetIoType	INPUT / OUTPUT / BIDIRECTIONAL
scale	Double	Multiplikator (raw * scale + offset = Anzeige)
offset	Double	Additivkonstante
unit	String	Einheit (z. B. „°C“, „rpm“)
widgetType	WidgetType	AUTO / VALUE / LED / GAUGE / CHART / TOGGLE
gaugeMin/Max	Double	Gauge-Skalierung
thresholdWarn	Double?	Warnschwelle
thresholdAlarm	Double?	Alarmschwelle
alarmOnChange	Boolean	Alarm bei Wertänderung

## 20. PROFINET-Protokollimplementierung

### S7Client (ISO-on-TCP, Port 102)

Implementiert das S7Comm-Protokoll über ISO-on-TCP nach RFC 1006:

```

TPKT Header (4 Bytes): Version(1) + Reserved(1) + Length(2)
COTP Header (3 Bytes): LI(1) + PDU-Type(1) + TPDU-NR/EOT(1)
S7Comm PDU:

```

```

Header: ProtID(1=0x32) + MsgType(1) + RedID(2) + PDU-Ref(2) + ParamLen(2) +
DataLen(2)
Parameter: Function(1) + ItemCount(1) + Items...
Data: ...

```

### Verbindungsaufbau (2-stufig):

24. COTP Connection Request (CR, TPDU 0xE0) mit Quell/Ziel-TSAP
25. S7 Setup Communication (Funktion 0xF0) mit PDU-Größe und AMQ-Werten

### Read Variable (FC 0x04):

- Item-Spezifikation: Syntax-ID=0x10, Transport-Size=BYTE, Länge, DB-Nr., Bereich-Code, Adresse
- Adresse kodiert als 24-Bit: Byte-Adresse × 8 (Bit-Adresse)

### Write Variable (FC 0x05):

- Item-Spezifikation identisch mit Read
- Datenteil: Return-Code + Transport-Size + Länge + Bytes

### S7-Bereichscodes:

Bereich	Code
I (Inputs)	0x81
Q (Outputs)	0x82
M (Merker)	0x83
DB	0x84
Timer	0x1D
Counter	0x1C

### ProfinetRecord (PROFINET TCP, Port 34964)

Vereinfachtes PROFINET Application Layer Protocol für acyclischen Record-Zugriff:

```

RecordDataReadReq (BlockType 0x0006):
  BlockType(2) + BlockLength(2) + BlockVersion(2) + SeqNumber(2) +
  ArUUID(8) + Slot(2) + Subslot(2) + Index(2) + Length(2)

RecordDataReadRes (BlockType 0x8006):
  BlockType(2) + BlockLength(2) + ... + Data(n)

```

## 21. DCP Device Discovery

**Klasse:** ProfinetDcp

DCP Identify All Request:

```

FrameID:      0xFEFE (DCP-All-Ident.req)
ServiceID:    0x05 (Identify)
ServiceType:  0x00 (Request)
XID:          4 Bytes (Transaktions-ID)
ResponseDelay: 2 Bytes
DCPDataLength: 0x0004
Option:       0xFF (All)
Suboption:    0xFF (All)
BlockLength:  0x0000

```

#### DCP Identify Response (von Geräten):

- FrameID 0xFEFF
- Blöcke: IP-Parameter, NameOfStation, DeviceID, DeviceRole, TypeOfStation

#### DeviceRole-Bits:

Bit	Rolle
0	IO-Device
1	IO-Controller
2	IO-Supervisor
3	IO-Multicast-Controller

## 22. Datenbank

Room-Datenbank (profinet\_log.db) mit drei Tabellen:

Tabelle	Beschreibung
log_sessions	Aufzeichnungssitzungen (Gerät, Start/Endzeit)
log_entries	Einzelne Messwerte (SessionID, Zeitstempel, Modul, Wert)
alarm_events	Alarmhistorie (Typ, Nachricht, Zeitstempel, bestätigt)

Geräte und Module werden als **JSON** in SharedPreferences gespeichert (kein Room).

## 23. Berechtigungen

Permission	Zweck
INTERNET	TCP/UDP-Verbindung zu PROFINET-Geräten
ACCESS_NETWORK_STATE	Netzwerkstatus prüfen
ACCESS_WIFI_STATE	WLAN-Interface-Info für DCP

CHANGE_WIFI_MULTICAST_STATE	UDP-Multicast für DCP-Broadcast
POST_NOTIFICATIONS	Alarm-Push-Benachrichtigungen (Android 13+)
NFC	NFC-Tag lesen/schreiben

\*Profinet++ – PROFINET IO-Controller für Android\*

\*Basierend auf IEC 61158-6-10 · IEC 61784-2 · RFC 1006\*

\*2026-05-21 · Profinet++ v1.0\*