

Profinet++ – Feature Documentation

Platform: Android (minSdk 26 / compileSdk 35)

Technology stack: Kotlin 2.0 · Jetpack Compose · Material 3 · Room · Coroutines

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

Table of Contents

1. [App Overview & Getting Started](#1-app-overview--getting-started)
2. [Navigation & Menu Structure](#2-navigation--menu-structure)
3. [Home Screen (MainScreen)](#3-home-screen-mainscreen)
4. [Device List (DeviceListScreen)](#4-device-list-devicelistscreen)
5. [Device Detail (DeviceDetailScreen)](#5-device-detail-devicedetailscreen)
6. [Create/Edit Device (DeviceFormScreen)](#6-createedit-device-deviceformscreen)
7. [Create/Edit Module (ModuleFormScreen)](#7-createedit-module-moduleformscreen)
8. [DCP Scan (ScanScreen)](#8-dcp-scan-scanscreen)
9. [Recordings (LogScreen)](#9-recordings-logscreens)
10. [Alarms (AlarmScreen)](#10-alarms-alarmscreen)
11. [Diagnostics (DiagScreen)](#11-diagnostics-diagscreen)
12. [IO Device Mode (IoDeviceScreen)](#12-io-device-mode-iodevicescreen)
13. [NFC (NfcScreen)](#13-nfc-nfcscreen)
14. [QR Code (QrScreen)](#14-qr-code-qrscreen)
15. [GSDML Viewer (GsdmlViewerScreen)](#15-gsdml-viewer-gsdmlviewerscreen)
16. [Settings (SettingsScreen)](#16-settings-settingscreens)
17. [Home Screen Widget](#17-home-screen-widget)
18. [Wear OS](#18-wear-os)
19. [Data Models](#19-data-models)
20. [PROFINET Protocol Implementation](#20-profinet-protocol-implementation)
21. [DCP Device Discovery](#21-dcp-device-discovery)
22. [Database](#22-database)
23. [Permissions](#23-permissions)

1. App Overview & Getting Started

Profinet++ is a full-featured PROFINET IO-Controller/IO-Supervisor application for Android. It supports access to PROFINET devices (PLCs, drives, field devices) and provides data visualization, recording, alarms, and diagnostics.

Two entry points at app start:

Entry point	Description
Home Screen	Direct PROFINET access without configuration – for quick tests
Device List	Management of saved devices/plants – for structured work

Supported connection types:

Type	Description	Default port
S7/ISO-on-TCP	Siemens S7-300/400/1200/1500 and compatible devices	102
PROFINET TCP	Generic PROFINET gateways with TCP support	34964
OPC UA	OPC UA Server (prepared, extension planned)	4840

Important technical limitation:

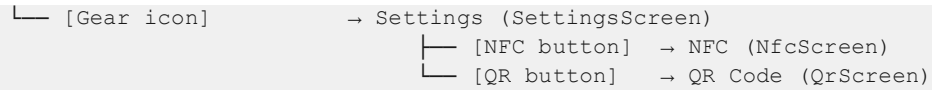
PROFINET Real-Time (RT/IRT) requires Layer-2 Ethernet access (EtherType 0x8892), which is **not** available on standard Android. Profinet++ therefore implements:

- **S7Comm** over ISO-on-TCP (RFC 1006, TCP port 102) – for Siemens S7/PROFINET PLCs
- **PROFINET Application Layer Protocol** over TCP (for gateways)
- **DCP** over UDP (simplified discovery approach for compatible gateways)

2. Navigation & Menu Structure

The app starts on the **Home Screen (MainScreen)**. All other screens are accessed from the **Device List**.





3. Home Screen (MainScreen)

Access: Launch app – the home screen appears automatically.

Provides quick access to PROFINET without prior device configuration.

3.1 Connection Settings

Field	Description
Connection Type	Dropdown: S7/ISO-on-TCP · PROFINET TCP · OPC UA
IP Address	IP address of the PROFINET device
Port	TCP port (set automatically when connection type is selected)
Rack	S7 rack number (default: 0)
Slot (CPU)	S7 CPU slot number (default: 1)

3.2 Reading Data

Step	Control	Description
1	"Area" dropdown	Memory area: I · Q · M · DB · T · C · Record
2	"DB#" field	Data block number (only for Area=DB)
3	"Byte offset" field	Start byte in the area
4	"Count" field	Number of bytes to read
5	"Read" button	Reads bytes and displays hex dump

3.3 Writing Data

Step	Control	Description
1	"Area" dropdown	Select target area
2	DB#/Offset fields	Target address
3	"Hex bytes" field	Bytes to write as hex, e.g. "00 FF 01 A5"
4	"Write" button	Writes bytes to device

3.4 Protocol Log










- Shows the last 20 communication events
- Timestamp in HH:mm:ss.SSS format
- READ/WRITE + hex dump

4. Device List (DeviceListScreen)

Access: Home Screen → List icon top right

The device list shows all configured PROFINET devices. An empty list shows a hint message.

4.1 Toolbar Icons

Icon	Name	Action
 (Flask)	Load demo	Loads a demo device (S7-1200 simulation)
 (Share)	Export JSON	Exports all devices as JSON (share dialog)
 (Folder)	Import menu	Import JSON or GSDML file
 (Radar)	DCP Scan	Opens ScanScreen (DCP device discovery)
 (Bell)	Alarms	Opens AlarmScreen; badge shows unacknowledged alarms
 (Bug)	Diagnostics	Opens DiagScreen (protocol tracer)
 (Chart)	Recordings	Opens LogScreen
 (Server)	IO Device	Opens IoDeviceScreen
 (Gear)	Settings	Opens SettingsScreen

4.2 Create New Device

Path: Device List → "+" button (blue round button, bottom right) → DeviceFormScreen

4.3 Importing

Menu item	Action
Import JSON	File picker (JSON backup)
Import GSDML (XML)	GSDML file picker – device is automatically created with modules

5. Device Detail (DeviceDetailScreen)

Access: Device List → Tap on device card

Core of the app – shows all configured modules of a PROFINET device in four tabs.

Tab Navigation

Tab	Function
Read	Read values, polling, recording
Write	Write values (output modules)
Table	All values in tabular form (Name / Address / Value)
Dashboard	Visual widgets (LED, Gauge, Value)

5.1 "Read" Tab – Reading Values

Step	Control	Description
1	"Read All" button	Read all modules at once
— or —		
1	▼ (Read) button on module card	Read only this module
—	Card background	Green = OK, Red = Error

5.2 Starting/Stopping Polling

Step	Control	Description
1	"Interval (s)" field	Enter seconds
2	"Start Polling" button	Cyclic reading begins
—	Status banner	"Polling active – every X s"
3	"Stop" button	Stop polling

5.3 Recording

Step	Control	Description
1	"Recording" toggle	On = recording starts
2	"Recording" toggle	Off = session is saved

5.4 "Write" Tab – Writing Values

Shows only modules with IO direction Output or Bidirectional.

Step	Control	Description
1	"New value" field	Enter value (matching the data type)
2	"Write" button	Value is written to device

6. Create/Edit Device (DeviceFormScreen)

Required fields: Name, IP address

Field	Description
Device Name	Freely chosen name
Description	Optional description
Connection Type	S7/ISO-on-TCP · PROFINET TCP · OPC UA
IP Address	Device IP address
Port	TCP port (set automatically)
Rack	S7 rack number (S7 TCP only)
Slot (CPU)	S7 CPU slot number (S7 TCP only)
Name of Station	PROFINET NameOfStation (optional, for DCP)

7. Create/Edit Module (ModuleFormScreen)

Modules are the PROFINET equivalents of Modbus registers.

Addressing

Area	Addressing	Examples
I (Inputs)	Byte offset · Bit offset	I0.0, I1.3
Q (Outputs)	Byte offset · Bit offset	Q0.0, Q2.7
M (Merkers)	Byte offset · Bit offset	M10.0, M20.1
DB (Data Block)	DB number · Byte offset · Bit offset	DB1.DBB0, DB2.DBW4
T (Timer)	Timer number	T0, T5
C (Counter)	Counter number	C0, C10
Record (Acyclic)	Slot · Subslot · Index	Slot0/Sub1/Idx0x0030

Data Types (IEC 61131-3 / PROFINET)

Type	Size	Description
BOOL	1 bit	Boolean value
BYTE	1 byte	8-bit unsigned
WORD	2 bytes	16-bit unsigned
DWORD	4 bytes	32-bit unsigned
LWORD	8 bytes	64-bit unsigned
INT	2 bytes	16-bit signed

DINT	4 bytes	32-bit signed
LINT	8 bytes	64-bit signed
UINT	2 bytes	16-bit unsigned (alias WORD)
UDINT	4 bytes	32-bit unsigned (alias DWORD)
REAL	4 bytes	IEEE 754 single
LREAL	8 bytes	IEEE 754 double
STRING	variable	S7 string (length byte + max-len byte + chars)
CHAR	1 byte	Single character
TIME	4 bytes	Milliseconds
DATE	2 bytes	Days since 1990-01-01
RAW	variable	Hex dump

8. DCP Scan (ScanScreen)

Access: Device List → Radar icon

DCP (Discovery and Configuration Protocol) searches for PROFINET devices on the network.

Step	Control	Description
1	" Broadcast Address " field	Default: 255.255.255.255 or subnet broadcast
2	" Start DCP Scan (3s) " button	UDP broadcast sent, wait 3 s for responses
—	Device list	Found devices appear with IP, MAC, NameOfStation
3	+ (AddCircle) button	Add device to device list

Response fields per found device:



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

Note: Real DCP runs on Layer-2 with multicast (EtherType 0x8892). This implementation uses UDP broadcast. Compatibility depends on the PROFINET gateway or switch.

9. Recordings (LogScreen)

Access: Device List → Chart icon

Manages all recorded measurement sessions.

Icon	Action
 (DeleteSweep)	Delete all sessions
▼ (ExpandMore)	Show session entries
↓ (FileDownload)	Export as CSV
 (Delete)	Delete this session


CSV format:

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

10. Alarms (AlarmScreen)

Access: Device List → Bell icon

Alarm type	Color	Trigger
THRESHOLD_ALARM	Red	Measured value \geq alarm threshold
THRESHOLD_WARN	Yellow	Measured value \geq warning threshold
CONNECTION_LOST	Gray	Communication error
VALUE_CHANGE	Blue	Value has changed (alarmOnChange=true)
PROFINET_DIAGNOSIS	Purple	PROFINET diagnostic message

Icon	Action
✓ (Check)	Acknowledge single alarm
✓✓ (DoneAll)	Acknowledge all
 (DeleteSweep)	Delete all

11. Diagnostics (DiagScreen)

Access: Device List → Bug icon

Protocol tracer for PROFINET communication with 2-tab layout.

Tab 1 – Frames

Element	Description
REC badge	Blinking red badge in title during recording
Play/Pause button	Start/pause recording
Filter chips	All · OK · Errors – filters the displayed frame list
Frame card	Shows timestamp, direction (TX/RX), length, latency, status color
Hex dump	Tap to expand: Wireshark-style (offset · hex · ASCII)
ClearAll button	Delete all recorded frames
PCAP export	Export frames as <code>.pcap</code> file (Wireshark-compatible)

Ring buffer: Maximum 500 frames; oldest are discarded on overflow.

Tab 2 – Statistics

Element	Description
Overview	Total frames, error frames, success rate
Error breakdown	Timeouts vs. other errors
Latency sparkline	Canvas chart of recent latency values; red line at 200 ms threshold

12. IO Device Mode (IoDeviceScreen)

Access: Device List → Server icon

Simulates a PROFINET IO device (field device).

Note: Full PROFINET IO device support requires PROFINET-RT on Layer 2, which is not available on standard Android hardware. This mode is a simulation to demonstrate the IO device concept.

Area	Description
Input Data	Process data of this device (→ IO controller), configurable as hex bytes
Output Data	Received data from the IO controller

13. NFC (NfcScreen)

Access: Settings → NFC button

Writes the connection configuration of a device to an NFC tag.

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

14. QR Code (QrScreen)

Access: Settings → QR button

Displays a QR code with the connection configuration. Used for quickly sharing device configurations.

15. GSDML Viewer (GsdmlViewerScreen)

Access: Opened after GSDML import or via direct link

Displays the imported device data in structured format:

- Device identity (VendorID, DeviceID, NameOfStation)
- All configured modules with address, data type, direction
- Scaling, unit, alarm thresholds

16. Settings (SettingsScreen)

Section	Setting	Description
Appearance	Dark Mode	Light/dark theme
Notifications	Enable Alarm Notifications	Push notifications for alarms
MQTT Bridge	Enable MQTT	Send PROFINET values to MQTT broker
MQTT Bridge	Broker URL	e.g. tcp://broker.hivemq.com:1883
MQTT Bridge	Topic	e.g. profinet/data
Security	Biometric Lock	Secure app start with fingerprint/face
Security	License Key	Profinet++ Pro license key (format: PFNET-PRO-...)
Share & Transfer	NFC	Opens NFC screen
Share & Transfer	QR Code	Opens QR screen

17. Home Screen Widget

Widget shows the app name and can be extended with live values.

Classes:

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

18. Wear OS

Basic Wear OS companion app scaffold is available.

Supported features (prepared):

- Receive values from the phone module via Wearable Data Layer
- Send write commands to phone app (path: `/profinet/write`)

19. Data Models

ProfinetDevice

Field	Type	Description
id	String	UUID
name	String	Display name
description	String	Optional description
ip	String	IP address
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>	Configured modules

ProfinetModule

Field	Type	Description
id	String	UUID

name	String	Display name
area	ProfinetArea	I / Q / M / DB / T / C / RECORD
dbNumber	Int	DB number (for area=DB)
byteOffset	Int	Byte address in area
bitOffset	Int	Bit offset (0-7, BOOL only)
arraySize	Int	Array size / string length
slot	Int	PROFINET slot (for 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	Multiplier (raw × scale + offset = display)
offset	Double	Additive constant
unit	String	Unit (e.g. "°C", "rpm")
widgetType	WidgetType	AUTO / VALUE / LED / GAUGE / CHART / TOGGLE
gaugeMin/Max	Double	Gauge scaling
thresholdWarn	Double?	Warning threshold
thresholdAlarm	Double?	Alarm threshold
alarmOnChange	Boolean	Alarm on value change

20. PROFINET Protocol Implementation

S7Client (ISO-on-TCP, port 102)

Implements the S7Comm protocol over ISO-on-TCP per 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: ...

```

Connection setup (2-stage):

24. COTP Connection Request (CR, TPDU 0xE0) with source/destination TSAP
25. S7 Setup Communication (function 0xF0) with PDU size and AMQ values

Read Variable (FC 0x04):

- Item specification: SyntaxID=0x10, TransportSize=BYTE, Length, DB no., area code, address
- Address encoded as 24-bit: byte address × 8 (bit address)

Write Variable (FC 0x05):

- Item specification identical to Read
- Data part: return code + transport size + length + bytes

S7 Area Codes:

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

ProfinetRecord (PROFINET TCP, port 34964)

Simplified PROFINET Application Layer Protocol for acyclic record access:

```
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

Class: ProfinetDcp

DCP Identify All Request:

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

DCP Identify Response (from devices):

- FrameID 0xFEFF
- Blocks: IP parameters, NameOfStation, DeviceID, DeviceRole, TypeOfStation

DeviceRole bits:

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

22. Database

Room database (`profinet_log.db`) with three tables:

Table	Description
<code>log_sessions</code>	Recording sessions (device, start/end time)
<code>log_entries</code>	Individual measurements (sessionID, timestamp, module, value)
<code>alarm_events</code>	Alarm history (type, message, timestamp, acknowledged)

Devices and modules are stored as **JSON** in SharedPreferences (no Room).

23. Permissions

Permission	Purpose
<code>INTERNET</code>	TCP/UDP connection to PROFINET devices
<code>ACCESS_NETWORK_STATE</code>	Check network status
<code>ACCESS_WIFI_STATE</code>	WiFi interface info for DCP
<code>CHANGE_WIFI_MULTICAST_STATE</code>	UDP multicast for DCP broadcast
<code>POST_NOTIFICATIONS</code>	Alarm push notifications (Android 13+)
<code>NFC</code>	Read/write NFC tags

Profinet++ – PROFINET IO-Controller for Android

Based on IEC 61158-6-10 · IEC 61784-2 · RFC 1006

2026-05-21 · Profinet++ v1.0