

# CANopen++ – Funktionsdokumentation

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

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

**Protokolle:** CANopen SDO/NMT/PDO · DeviceNet · TCP-Gateway · SocketCAN/TCP · USB-Serial · Bluetooth

## Inhaltsverzeichnis

1. [App-Überblick & Einstieg](#1-app-überblick--einstieg)
2. [Navigation & Menüstruktur](#2-navigation--menüstruktur)
3. [Hauptbildschirm (MainScreen)](#3-hauptbildschirm-mainscreen)
4. [Knotenliste (NodeListScreen)](#4-knotenliste-nodelistscreen)
5. [Knotendetail (NodeDetailScreen)](#5-knotendetail-nodedetailscreen)
6. [Knoten anlegen/bearbeiten (NodeFormScreen)](#6-knoten-anlegenbearbeiten-nodeformscreen)
7. [Objekt anlegen/bearbeiten (ObjectFormScreen)](#7-objekt-anlegenbearbeiten-objectformscreen)
8. [Node-Scan (ScanScreen)](#8-node-scan-scanscreen)
9. [NMT-Steuerbefehle (NmtScreen)](#9-nmt-steuerbefehle-nmtscreen)
10. [Aufzeichnungen (LogScreen)](#10-aufzeichnungen-logscreen)
11. [Alarmer (AlarmScreen)](#11-alarmer-alarmscreen)
12. [Diagnose (DiagScreen)](#12-diagnose-diagscreen)
13. [NFC (NfcScreen)](#13-nfc-nfcscreen)
14. [QR-Code (QrScreen)](#14-qr-code-qrscreen)
15. [EDS-Viewer (EdsViewerScreen)](#15-eds-viewer-edsviewerscreen)
16. [Einstellungen (SettingsScreen)](#16-einstellungen-settingscreen)
17. [Homescreen-Widget](#17-homescreen-widget)
18. [Wear OS](#18-wear-os)
19. [Datenmodelle](#19-datenmodelle)
20. [CANopen-Protokollimplementierung](#20-canopen-protokollimplementierung)
21. [Datenbank](#21-datenbank)
22. [Berechtigungen](#22-berechtigungen)

## 1. App-Überblick & Einstieg

CANopen++ ist eine vollständige CANopen/DeviceNet-Monitor-Anwendung für Android. Sie ermöglicht den Zugriff auf CANopen-Geräte über TCP-Gateways, bietet Objektverzeichnisverwaltung, NMT-Steuerung, Datenvisualisierung, Aufzeichnung, Alarmierung und Diagnose.

## Zwei Einstiegswege beim App-Start:

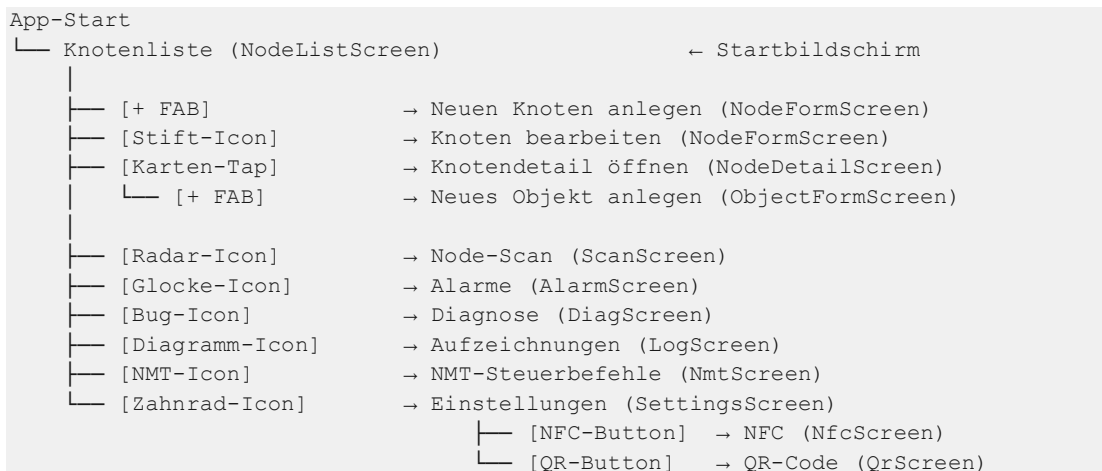
Einstieg	Beschreibung
Hauptbildschirm	Direktzugriff auf SDO-Protokoll ohne Konfiguration – für schnelle Tests
Knotenliste	Verwaltung gespeicherter CANopen-Knoten – für strukturierte Arbeit

## Unterstützte Verbindungsarten:

Typ	Beschreibung	Standard-Port
TCP Gateway	CANopen über TCP-zu-CAN-Gateway	3000
SocketCAN/TCP	Linux SocketCAN über TCP-Tunnel	29536
USB-Serial	CAN-USB-Adapter (USB-OTG)	–
Bluetooth	Drahtlose CAN-Brücke (SPP)	–
DeviceNet TCP	DeviceNet über EtherNet/IP-Gateway	44818

## 2. Navigation & Menüstruktur

Die App startet auf der **Knotenliste (NodeListScreen)**. Alle weiteren Screens werden von der Knotenliste aus oder über die Toolbar-Icons erreicht.



## 3. Hauptbildschirm (MainScreen)

**Erreichen:** Hauptbildschirm über untere Leiste oder Schnell-SDO-Zugriff aus dem Menü.

Ermöglicht schnellen SDO-Zugriff ohne vorherige Knoten-Konfiguration.

### 3.1 Verbindungseinstellung (Quick SDO)

Feld	Beschreibung
Connection Type	Dropdown: TCP Gateway · SocketCAN/TCP · USB-Serial · Bluetooth · DeviceNet TCP
IP Address	IP-Adresse des TCP-Gateways
Port	TCP-Port (wird automatisch gesetzt)
Node ID	CANopen-Knoten-ID (1–127)

### 3.2 SDO-Objekt lesen

Schritt	Bedienelement	Beschreibung
1	Feld „Index“	Objektindex (Hex, z. B. 0x1018)
2	Feld „Subindex“	Subindex (0–254)
3	Dropdown „Datentyp“	Erwarteter Typ (UINT8, INT16, REAL32 usw.)
4	Button „Lesen“	SDO Upload Request absenden
—	Ergebnisbox	Rohdaten (Hex) und konvertierter Wert

### 3.3 SDO-Objekt schreiben

Schritt	Bedienelement	Beschreibung
1	Felder Index / Subindex / Datentyp	Zieladresse konfigurieren
2	Feld „Wert“	Zu schreibender Wert
3	Button „Schreiben“	SDO Download Request absenden

### 3.4 Protocol Log









- Zeigt die letzten Kommunikationsereignisse
- SDO Requests/Responses im Hex-Format mit Zeitstempel

## 4. Knotenliste (NodeListScreen)

**Erreichen:** App-Start – Startbildschirm

Zeigt alle konfigurierten CANopen-Knoten. Bei leerer Liste erscheint ein Hinweis.

#### 4.1 Toolbar-Icons

Icon	Bezeichnung	Aktion
 (Kolben)	Demo laden	Lädt einen Demo-Knoten
 (Teilen)	JSON exportieren	Exportiert alle Knoten als JSON
 (Ordner)	Import-Menü	JSON oder EDS-Datei importieren
 (Radar)	Node-Scan	Öffnet ScanScreen
 (Glocke)	Alarme	Öffnet AlarmScreen
 (Bug)	Diagnose	Öffnet DiagScreen
 (Diagramm)	Aufzeichnungen	Öffnet LogScreen
NMT	NMT-Befehle	Öffnet NmtScreen
 (Zahnrad)	Einstellungen	Öffnet SettingsScreen

#### 4.2 Neuen Knoten anlegen

**Weg:** Knotenliste → „+“-Button (unten rechts) → NodeFormScreen

#### 4.3 Importieren

Menüpunkt	Aktion
Import JSON	Dateiauswahl (JSON-Backup)
Import EDS	EDS-Dateiauswahl – Knoten wird automatisch mit Objekten angelegt

### 5. Knotendetail (NodeDetailScreen)

**Erreichen:** Knotenliste → Auf Knoten-Karte tippen

Herzstück der App – zeigt alle konfigurierten Objekte eines CANopen-Knotens in vier Tabs.

#### Tab-Navigation

Tab	Funktion
Read	Objektwerte lesen, Polling, Aufzeichnung
Write	Objekte schreiben
PDO	PDO-Konfiguration und Mapping
Chart	Werteverlauf als Diagramm

#### 5.1 Tab „Read“ – Objektwert lesen

Schritt	Bedienelement	Beschreibung
1	Button „Read All“	Alle Objekte auf einmal lesen

		(SDO Upload)
— oder —		
1	Button ▼ auf Objekt-Karte	Nur dieses Objekt 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 starten/stoppen

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

## 5.4 Tab „Write“ – Objektwert schreiben

Schritt	Bedienelement	Beschreibung
1	Textfeld „New value“	Wert eingeben (passend zum Datentyp)
2	Button „Write“	SDO Download Request absenden

## 6. Knoten anlegen/bearbeiten (NodeFormScreen)

**Pflichtfelder:** Name, IP-Adresse, Node-ID

Feld	Beschreibung
Node Name	Frei wählbarer Name
Description	Optionale Beschreibung
Connection Type	TCP Gateway · SocketCAN/TCP · USB-Serial · Bluetooth · DeviceNet TCP
IP Address	IP-Adresse des Gateways
Port	TCP-Port (wird automatisch gesetzt)
Node ID	CANopen-Knoten-ID (1–127)

Baud Rate	CAN-Baudrate (125k / 250k / 500k / 1M)
EDS File	EDS-Dateireferenz (optional)

## 7. Objekt anlegen/bearbeiten (ObjectFormScreen)

Objekte entsprechen Einträgen im CANopen-Objektverzeichnis.

### Adressierung

Feld	Beschreibung
Index	Objektindex (0x0000–0xFFFF)
Subindex	Subindex (0–254)
Access Type	RO (nur lesen) · WO (nur schreiben) · RW (lesen+schreiben)

### Datentypen

Typ	Größe	Beschreibung
BOOLEAN	1 Bit	Boolescher Wert
INT8	1 Byte	8-Bit vorzeichenbehaftet
INT16	2 Byte	16-Bit vorzeichenbehaftet
INT32	4 Byte	32-Bit vorzeichenbehaftet
INT64	8 Byte	64-Bit vorzeichenbehaftet
UINT8	1 Byte	8-Bit vorzeichenlos
UINT16	2 Byte	16-Bit vorzeichenlos
UINT32	4 Byte	32-Bit vorzeichenlos
UINT64	8 Byte	64-Bit vorzeichenlos
REAL32	4 Byte	IEEE 754 Single
REAL64	8 Byte	IEEE 754 Double
VISIBLE_STRING	variabel	ASCII-Zeichenkette
OCTET_STRING	variabel	Binäre Daten (Hex)
DOMAIN	variabel	Freier Binärblock

### Konvertierung & Visualisierung

Feld	Beschreibung
Scale	Multiplikator: Anzeigewert = (Rohwert × Scale) + Offset
Offset	Addition nach Skalierung
Unit	Einheitentext (z. B. „°C“, „rpm“)

Widget-Typ	AUTO · VALUE · LED · GAUGE · CHART
Warn-Schwellwert	Ab diesem Wert → orange Einfärbung
Alarm-Schwellwert	Ab diesem Wert → rote Einfärbung

## 8. Node-Scan (ScanScreen)

**Erreichen:** Knotenliste → Radar-Icon

Sucht aktive CANopen-Knoten im Netzwerk per NMT Heartbeat / Node Guarding.

Schritt	Bedienelement	Beschreibung
1	Felder IP + Port	Gateway-Adresse eingeben
2	Button „Start Scan“	Sendet NMT-Scan-Anfragen an alle Node-IDs (1–127)
—	Geräteliste	Antwortende Knoten erscheinen mit ID und NMT-Zustand
3	Button + ( <b>AddCircle</b> )	Knoten zur Knotenliste hinzufügen

## 9. NMT-Steuerbefehle (NmtScreen)

**Erreichen:** Knotenliste → NMT-Icon

Sendet NMT-Netzwerkverwaltungsbefehle an einzelne oder alle Knoten.

Befehl	Code	Beschreibung
Start Node	0x01	Knoten in den Zustand „Operational“ versetzen
Stop Node	0x02	Knoten in den Zustand „Stopped“ versetzen
Enter Pre-Operational	0x80	Knoten in Pre-Operational versetzen
Reset Node	0x81	Vollständigen Knoten-Reset auslösen
Reset Communication	0x82	Nur Kommunikation zurücksetzen

**Ziel-Auswahl:**

- Einzelner Knoten (Node-ID 1–127)
- Broadcast (Node-ID 0 = alle Knoten)

## 10. Aufzeichnungen (LogScreen)

Erreichen: Knotenliste → 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:

```
Zeitstempel;Knoten-ID;Objektindex;Subindex;Rohwert;Wert;Einheit
2024-01-15 14:23:05;1;0x6000;0x01;0042;66;rpm
```

## 11. Alarme (AlarmScreen)

Erreichen: Knotenliste → Glocken-Icon

Alarm-Typ	Farbe	Auslöser
THRESHOLD_ALARM	Rot	Messwert $\geq$ Alarm-Schwelle
THRESHOLD_WARN	Gelb	Messwert $\geq$ Warn-Schwelle
CONNECTION_LOST	Grau	Kommunikationsfehler
VALUE_CHANGE	Blau	Wert hat sich geändert (alarmOnChange=true)

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

## 12. Diagnose (DiagScreen)

Erreichen: Knotenliste → Bug-Icon

Protokoll-Tracer für CANopen-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

## 13. NFC (NfcScreen)

Erreichen: Einstellungen → NFC-Button

Schreibt die Verbindungskonfiguration eines Knotens auf ein NFC-Tag.

- Format: `canopen://<IP>:<Port>/node/<NodeID>`
- MIME-Type: `application/io.github.erginmusa.canopenplusplus.connection`

## 14. QR-Code (QrScreen)

Erreichen: Einstellungen → QR-Button

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

## 15. EDS-Viewer (EdsViewerScreen)

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

Zeigt die importierten Gerätedaten im strukturierten Format an:

- Geräte-Identität (Vendor, Produktname, Revision)
- Alle Objekte aus dem EDS mit Index, Subindex, Datentyp, Zugriffstyp
- Defaultwerte und Skalierungsinformationen

## 16. Einstellungen (SettingsScreen)

Abschnitt	Einstellung	Beschreibung
Appearance	Dark Mode	Helles/dunkles Theme
Notifications	Enable Alarm Notifications	Push-Benachrichtigungen bei Alarmen
MQTT Bridge	Enable MQTT	CANopen-Werte an MQTT-Broker senden
MQTT Bridge	Broker URL	z. B. tcp://broker.hivemq.com:1883
MQTT Bridge	Topic-Präfix	z. B. canopen/{node}/{index}
Security	Biometric Lock	App-Start mit Fingerabdruck/Gesicht sichern
Security	License Key	CANopen++ Pro-Lizenzschlüssel
Share & Transfer	NFC	Öffnet NFC-Screen
Share & Transfer	QR Code	Öffnet QR-Screen

## 17. Homescreen-Widget

Widget zeigt Knotenname und aktuellen Objektwert auf dem Android-Homescreen.

**Klassen:**

- CanopenWidget (GlanceAppWidget)
- CanopenWidgetReceiver (GlanceAppWidgetReceiver)

## 18. Wear OS

Grundlegendes Wear OS Companion App-Gerüst vorhanden.

### Unterstützte Funktionen (vorbereitet):

- Objektwerte vom Phone-Modul via Wearable Data Layer empfangen
- Schreibbefehle an Phone-App senden

**Datenpfad:** Wearable Data API – wird nach jeder erfolgreichen Leseoperation automatisch ausgelöst.

## 19. Datenmodelle

### CanopenNode

Feld	Typ	Beschreibung
id	String	UUID
name	String	Anzeigename
description	String	Optionale Beschreibung
ip	String	IP-Adresse des Gateways
port	Int	TCP-Port
nodeld	Int	CANopen Node-ID (1–127)
baudRate	Int	CAN-Baudrate in kBit/s
connectionType	CanopenConnectionType	TCP_GATEWAY / SOCKETCAN_TCP / USB_SERIAL / BLUETOOTH / DEVICENET_TCP
edsFile	String?	Referenz auf importierte EDS-Datei
nmtState	CanopenNmtState	UNKNOWN / INITIALISING / STOPPED / OPERATIONAL / PRE_OPERATIONAL
objects	List<CanopenObject>	Konfigurierte Objektverzeichnis-Einträge

### CanopenObject

Feld	Typ	Beschreibung
id	String	UUID
name	String	Anzeigename
index	Int	Objektindex (0x0000–0xFFFF)
subindex	Int	Subindex (0–254)
dataType	CanopenDataType	BOOLEAN/INT8/.../REAL64/VISIBLE_STRING/...

accessType	CanopenAccessType	RO / WO / RW / RWR / RWW
scale	Double	Multiplikator
offset	Double	Additivkonstante
unit	String	Einheit
widgetType	WidgetType	AUTO / VALUE / LED / GAUGE / CHART
thresholdWarn	Double?	Warnschwelle
thresholdAlarm	Double?	Alarmschwelle
alarmOnChange	Boolean	Alarm bei Wertänderung

### NMT-Zustände (CanopenNmtState)

Zustand	Beschreibung
UNKNOWN	Zustand nicht bekannt
INITIALISING	Knoten startet
STOPPED	Knoten gestoppt (nur NMT erreichbar)
OPERATIONAL	Knoten aktiv (SDO + PDO)
PRE_OPERATIONAL	Knoten vorkonfiguriert (SDO, kein PDO)

## 20. CANopen-Protokollimplementierung

### SDO (Service Data Object)

**Klasse:** CanopenClient

Implementiert den SDO-Kanal über TCP-Socket:

```

SDO Upload Request (Lesen):
  Command Specifier: 0x40
  Index: 2 Bytes (Little Endian)
  Subindex: 1 Byte
  Reserved: 4 Bytes

SDO Upload Response:
  CS: 0x4F (1 Byte) / 0x4B (2 Byte) / 0x47 (4 Byte) / 0x42 (segmented)
  Index: 2 Bytes
  Subindex: 1 Byte
  Data: 1-4 Bytes

SDO Download Request (Schreiben):
  CS: 0x2F (1 Byte) / 0x2B (2 Byte) / 0x27 (3 Byte) / 0x23 (4 Byte)
  Index: 2 Bytes
  Subindex: 1 Byte
  Data: 1-4 Bytes (Little Endian)

SDO Abort: CS 0x80 + 4-Byte Abort Code

```

- Timeout: 3000 ms
- Byte-Reihenfolge: Little Endian (CANopen-Standard)

### NMT (Network Management)

**Klasse:** CanopenController

NMT-Befehlsformat:

```
Byte 0: Command Specifier
  0x01 = Start Node
  0x02 = Stop Node
  0x80 = Enter Pre-Operational
  0x81 = Reset Node
  0x82 = Reset Communication
Byte 1: Node-ID (0 = alle Knoten)
```

### SDO Abort Codes (häufigste)

Code	Bedeutung
0x05030000	Toggle-Bit nicht alternierend
0x05040000	SDO-Protokoll-Timeout
0x06010000	Zugriff auf nicht existierendes Objekt
0x06010001	Schreibzugriff auf schreibgeschütztes Objekt
0x06010002	Lesezugriff auf schreibgeschütztes Objekt
0x06090011	Subindex existiert nicht
0x08000000	Allgemeiner Fehler

### Sealed Result Type

```
sealed class CanopenResult {
  data class Success(val data: ByteArray) : CanopenResult()
  data class Error(val message: String) : CanopenResult()
  object Timeout : CanopenResult()
}
```

## 21. Datenbank

Room-Datenbank mit drei Tabellen:

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

Knoten und Objekte werden als **JSON** in SharedPreferences gespeichert (kein Room).

## 22. Berechtigungen

Permission	Zweck
INTERNET	TCP-Verbindung zu CANopen-Gateways
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-Push-Benachrichtigungen (Android 13+)
USB_HOST (Feature)	USB-OTG für CAN-USB-Adapter

\*CANopen++ – CANopen/DeviceNet Monitor für Android\*

\*Basierend auf CiA 301 (CANopen Application Layer) · CiA 302 · ISO 11898\*

\*2026-05-21 · CANopen++ v1.0\*