CANopen++ – Feature Documentation

Platform: Android (minSdk 26 / compileSdk 35)

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

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

Table of Contents

1. App Overview & Getting Started

CANopen++ is a full-featured CANopen/DeviceNet monitor application for Android. It enables access to CANopen devices via TCP gateways, provides object dictionary management, NMT control, data visualization, recording, alarms, and diagnostics.

Two entry points at app start:

Entry point	Description
Home Screen	Direct SDO access without configuration – for quick tests
Node List	Management of saved CANopen nodes – for structured work

Supported connection types:

Type	Description	Default port
TCP Gateway	CANopen over TCP-to-CAN gateway	3000
SocketCAN/TCP	Linux SocketCAN over TCP tunnel	29536
USB-Serial	CAN-USB adapter (USB-OTG)	–
Bluetooth	Wireless CAN bridge (SPP)	–
DeviceNet TCP	DeviceNet over EtherNet/IP gateway	44818

2. Navigation & Menu Structure

The app starts on the **Node List (NodeListScreen)**. All other screens are accessed from the node list or via toolbar icons.

```
App Start
└── Node List (NodeListScreen)                    ← Start screen
    │
    ├── [+ FAB]              → Create new node (NodeFormScreen)
    ├── [Pencil icon]        → Edit node (NodeFormScreen)
    ├── [Card tap]           → Open node detail (NodeDetailScreen)
    │   └── [+ FAB]          → Create new object (ObjectFormScreen)
    │
    ├── [Radar icon]         → Node Scan (ScanScreen)
    ├── [Bell icon]          → Alarms (AlarmScreen)
    ├── [Bug icon]           → Diagnostics (DiagScreen)
    ├── [Chart icon]         → Recordings (LogScreen)
    ├── [NMT icon]           → NMT Commands (NmtScreen)
    └── [Gear icon]          → Settings (SettingsScreen)
                                   ├── [NFC button]  → NFC (NfcScreen)
                                   └── [QR button]   → QR Code (QrScreen)
```

3. Home Screen (MainScreen)

Access: Home screen via quick SDO access.

Provides quick SDO access without prior node configuration.

3.1 Connection Settings (Quick SDO)

Field	Description
Connection Type	Dropdown: TCP Gateway · SocketCAN/TCP · USB-Serial · Bluetooth · DeviceNet TCP
IP Address	IP address of the TCP gateway
Port	TCP port (set automatically)
Node ID	CANopen node ID (1–127)

3.2 Read SDO Object

Step	Control	Description
1	**"Index"** field	Object index (hex, e.g. 0x1018)
2	**"Subindex"** field	Subindex (0–254)
3	**"Data type"** dropdown	Expected type (UINT8, INT16, REAL32, etc.)
4	**"Read"** button	Send SDO Upload Request
—	Result box	Raw data (hex) and converted value

3.3 Write SDO Object

Step	Control	Description
1	Index / Subindex / Data type fields	Configure target address
2	**"Value"** field	Value to write
3	**"Write"** button	Send SDO Download Request

3.4 Protocol Log

- Shows the last communication events
- SDO requests/responses in hex format with timestamp

4. Node List (NodeListScreen)

Access: App launch – start screen

Shows all configured CANopen nodes. An empty list shows a hint message.

4.1 Toolbar Icons

Icon	Name	Action
⚗ (Flask)	Load demo	Loads a demo node
↗ (Share)	Export JSON	Exports all nodes as JSON

📁 (Folder)	Import menu	Import JSON or EDS file
📡 (Radar)	Node Scan	Opens ScanScreen
🔔 (Bell)	Alarms	Opens AlarmScreen
🐛 (Bug)	Diagnostics	Opens DiagScreen
📈 (Chart)	Recordings	Opens LogScreen
NMT	NMT commands	Opens NmtScreen
⚙ (Gear)	Settings	Opens SettingsScreen

4.2 Create New Node

Path: Node List → **"+"** button (bottom right) → NodeFormScreen

4.3 Importing

Menu item	Action
Import JSON	File picker (JSON backup)
Import EDS	EDS file picker – node is automatically created with objects

5. Node Detail (NodeDetailScreen)

Access: Node List → Tap on node card

Core of the app – shows all configured objects of a CANopen node in four tabs.

Tab Navigation

Tab	Function
Read	Read object values, polling, recording
Write	Write objects
PDO	PDO configuration and mapping
Chart	Value history as chart

5.1 "Read" Tab – Reading Object Values

Step	Control	Description
1	**"Read All"** button	Read all objects at once (SDO upload)
— or —		
1	▼ button on object card	Read only this object
—	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 Starting/Stopping Recording

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

5.4 "Write" Tab – Writing Object Values

Step	Control	Description
1	**"New value"** text field	Enter value (matching the data type)
2	**"Write"** button	Send SDO Download Request

6. Create/Edit Node (NodeFormScreen)

Required fields: Name, IP address, Node ID

Field	Description
Node Name	Freely chosen name
Description	Optional description
Connection Type	TCP Gateway · SocketCAN/TCP · USB-Serial · Bluetooth · DeviceNet TCP
IP Address	Gateway IP address
Port	TCP port (set automatically)
Node ID	CANopen node ID (1–127)
Baud Rate	CAN baud rate (125k / 250k / 500k / 1M)
EDS File	EDS file reference (optional)

7. Create/Edit Object (ObjectFormScreen)

Objects correspond to entries in the CANopen object dictionary.

Addressing

Field	Description
Index	Object index (0x0000–0xFFFF)
Subindex	Subindex (0–254)
Access Type	RO (read-only) · WO (write-only) · RW (read+write)

Data Types

Type	Size	Description
BOOLEAN	1 bit	Boolean value
INT8	1 byte	8-bit signed
INT16	2 bytes	16-bit signed
INT32	4 bytes	32-bit signed
INT64	8 bytes	64-bit signed
UINT8	1 byte	8-bit unsigned
UINT16	2 bytes	16-bit unsigned
UINT32	4 bytes	32-bit unsigned
UINT64	8 bytes	64-bit unsigned
REAL32	4 bytes	IEEE 754 single
REAL64	8 bytes	IEEE 754 double
VISIBLE_STRING	variable	ASCII string
OCTET_STRING	variable	Binary data (hex)
DOMAIN	variable	Free binary block

Conversion & Visualization

Field	Description
Scale	Multiplier: display value = (raw value × scale) + offset
Offset	Addition after scaling
Unit	Unit text (e.g. "°C", "rpm")
Widget type	AUTO · VALUE · LED · GAUGE · CHART
Warning threshold	Above this value → orange coloring
Alarm threshold	Above this value → red coloring

8. Node Scan (ScanScreen)

Access: Node List → Radar icon

Searches for active CANopen nodes on the network via NMT heartbeat/node guarding.

Step	Control	Description
1	IP + Port fields	Enter gateway address
2	**"Start Scan"** button	Sends NMT scan requests to all node IDs (1–127)
—	Device list	Responding nodes appear with ID and NMT state
3	**+ (AddCircle)** button	Add node to node list

9. NMT Commands (NmtScreen)

Access: Node List → NMT icon

Sends NMT network management commands to individual or all nodes.

Command	Code	Description
Start Node	0x01	Set node to "Operational" state
Stop Node	0x02	Set node to "Stopped" state
Enter Pre-Operational	0x80	Set node to Pre-Operational
Reset Node	0x81	Trigger full node reset
Reset Communication	0x82	Reset communication only

Target selection:

- Single node (Node ID 1–127)
- Broadcast (Node ID 0 = all nodes)

10. Recordings (LogScreen)

Access: Node 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;NodeID;ObjectIndex;Subindex;RawValue;Value;Unit
2024-01-15 14:23:05;1;0x6000;0x01;0042;66;rpm
```

11. Alarms (AlarmScreen)

Access: Node List → Bell icon

Alarm type	Color	Trigger
THRESHOLD_ALARM	Red	Measured value ≥ alarm threshold
THRESHOLD_WARN	Yellow	Measured value ≥ warning threshold
CONNECTION_LOST	Gray	Communication error
VALUE_CHANGE	Blue	Value has changed (alarmOnChange=true)

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

12. Diagnostics (DiagScreen)

Access: Node List → Bug icon

Protocol tracer for CANopen 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 `.pcap` 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

13. NFC (NfcScreen)

Access: Settings → NFC button

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

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

14. QR Code (QrScreen)

Access: Settings → QR button

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

15. EDS Viewer (EdsViewerScreen)

Access: Opened after EDS import or via direct link

Displays the imported device data in structured format:

- Device identity (vendor, product name, revision)
- All objects from the EDS with index, subindex, data type, access type
- Default values and scaling information

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 CANopen values to MQTT broker
MQTT Bridge	Broker URL	e.g. `tcp://broker.hivemq.com:1883`
MQTT Bridge	Topic prefix	e.g. `canopen/{node}/{index}`
Security	Biometric Lock	Secure app start with fingerprint/face
Security	License Key	CANopen++ Pro license key
Share & Transfer	NFC	Opens NFC screen
Share & Transfer	QR Code	Opens QR screen

17. Home Screen Widget

Widget shows node name and current object value on the Android home screen.

Classes:

- `CanopenWidget` (GlanceAppWidget)
- `CanopenWidgetReceiver` (GlanceAppWidgetReceiver)

18. Wear OS

Basic Wear OS companion app scaffold is available.

Supported features (prepared):

- Receive object values from the phone module via Wearable Data Layer
- Send write commands to phone app

Data path: Wearable Data API – triggered automatically after each successful read operation.

19. Data Models

CanopenNode

Field	Type	Description
id	String	UUID
name	String	Display name
description	String	Optional description
ip	String	Gateway IP address
port	Int	TCP port
nodeId	Int	CANopen node ID (1–127)
baudRate	Int	CAN baud rate in kBit/s
connectionType	CanopenConnectionType	TCP_GATEWAY / SOCKETCAN_TCP / USB_SERIAL / BLUETOOTH / DEVICENET_TCP
edsFile	String?	Reference to imported EDS file
nmtState	CanopenNmtState	UNKNOWN / INITIALISING / STOPPED / OPERATIONAL / PRE_OPERATIONAL
objects	List\<CanopenObject\>	Configured object dictionary entries

CanopenObject

Field	Type	Description
id	String	UUID
name	String	Display name
index	Int	Object index (0x0000–0xFFFF)
subindex	Int	Subindex (0–254)
dataType	CanopenDataType	BOOLEAN/INT8/.../REAL64/VISIBLE_STRING/...
accessType	CanopenAccessType	RO / WO / RW / RWR / RWW
scale	Double	Multiplier
offset	Double	Additive constant
unit	String	Unit
widgetType	WidgetType	AUTO / VALUE / LED / GAUGE / CHART
thresholdWarn	Double?	Warning threshold
thresholdAlarm	Double?	Alarm threshold
alarmOnChange	Boolean	Alarm on value change

NMT States (CanopenNmtState)

State	Description
UNKNOWN	State not known
INITIALISING	Node is starting up
STOPPED	Node stopped (NMT only)
OPERATIONAL	Node active (SDO + PDO)
PRE_OPERATIONAL	Node pre-configured (SDO, no PDO)

20. CANopen Protocol Implementation

SDO (Service Data Object)

Class: `CanopenClient`

Implements the SDO channel over TCP socket:

```
SDO Upload Request (Read):
  Command Specifier: 0x40
  Index: 2 bytes (little endian)
  Subindex: 1 byte
  Reserved: 4 bytes

SDO Upload Response:
  CS: 0x4F (1 byte) / 0x4B (2 bytes) / 0x47 (4 bytes) / 0x42 (segmented)
  Index: 2 bytes
  Subindex: 1 byte
  Data: 1–4 bytes

SDO Download Request (Write):
  CS: 0x2F (1 byte) / 0x2B (2 bytes) / 0x27 (3 bytes) / 0x23 (4 bytes)
  Index: 2 bytes
  Subindex: 1 byte
  Data: 1–4 bytes (little endian)

SDO Abort: CS 0x80 + 4-byte abort code
```

- Timeout: 3000 ms
- Byte order: little endian (CANopen standard)

NMT (Network Management)

Class: `CanopenController`

NMT command format:

```
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 = all nodes)
```

SDO Abort Codes (most common)

Code	Meaning
0x05030000	Toggle bit not alternating
0x05040000	SDO protocol timeout
0x06010000	Attempt to access non-existent object
0x06010001	Write access to read-only object
0x06010002	Read access to write-only object
0x06090011	Subindex does not exist
0x08000000	General error

Sealed Result Type

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

21. Database

Room database with three tables:

Table	Description
`log_sessions`	Recording sessions (node, start/end time)
`log_entries`	Individual measurements (sessionID, timestamp, object index, value)
`alarm_events`	Alarm history (type, message, timestamp, acknowledged)

Nodes and objects are stored as **JSON** in SharedPreferences (no Room).

22. Permissions

Permission	Purpose
`INTERNET`	TCP connection to CANopen gateways
`ACCESS_NETWORK_STATE`	Check network status
`BLUETOOTH` / `BLUETOOTH_ADMIN`	Bluetooth (API ≤ 30)
`BLUETOOTH_CONNECT`	Bluetooth connection (API 31+)

`BLUETOOTH_SCAN`	Bluetooth device scan (API 31+)
`NFC`	Read/write NFC tags
`POST_NOTIFICATIONS`	Alarm push notifications (Android 13+)
`USB_HOST` (feature)	USB-OTG for CAN-USB adapters

CANopen++ – CANopen/DeviceNet Monitor for Android

Based on CiA 301 (CANopen Application Layer) · CiA 302 · ISO 11898

2026-05-21 · CANopen++ v1.0