Modbus++ – Feature Documentation

Platform: Android (minSdk 26 / compileSdk 35)

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

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

Table of Contents

1. App Overview & Getting Started

Modbus++ is a full-featured Modbus communication application for Android. It supports both master and slave modes and provides data visualization, recording, alarms, and diagnostics.

Two entry points at app start:

Entry point	Description
Home Screen	Direct Modbus protocol access without configuration – for quick tests
Structure List	Management of saved devices/plants – for

	structured work

Supported connection types:

Type	Description
Modbus TCP	Network connection (IP + port)
RTU/USB-Serial	Direct connection via USB-OTG
RTU/Bluetooth	Wireless connection (SPP profile)
RTU/WiFi-Bridge	RTU protocol over TCP bridge

2. Navigation & Menu Structure

The app starts on the **Home Screen (MainScreen)**. Primary navigation uses the **Bottom Navigation Bar** with six tabs plus icon buttons in the top bar (TopAppBar) and Floating Action Buttons (FAB, the "+" symbol).

Bottom Navigation Bar:

Tab	Icon	Screen
Home	House	Home Screen (MainScreen)
Devices	List	Structure List (StructureListScreen)
Slave	Memory	Slave Mode (SlaveScreen)
Log	History	Recordings (LogScreen)
Alarms	Bell	Alarms (AlarmScreen)
Settings	Gear	Settings (SettingsScreen)

```
App Start
└── Home Screen (MainScreen)                    ← Bottom Nav: House icon
│
├── Structure List (StructureListScreen)         ← Bottom Nav: List icon
│   ├── [+ FAB]          → Create new structure (StructureFormScreen)
│   ├── [Pencil icon]    → Edit structure (StructureFormScreen)
│   └── [Card tap]       → Open structure detail (StructureDetailScreen)
│       ├── [+ FAB]      → Create new detail (DetailFormScreen)
│       └── [Pencil icon]→ Edit detail (DetailFormScreen)
│
├── Slave Mode (SlaveScreen)                     ← Bottom Nav: Memory icon
├── Recordings (LogScreen)                       ← Bottom Nav: History icon
├── Alarms (AlarmScreen)                         ← Bottom Nav: Bell icon (badge)
└── Settings (SettingsScreen)                    ← Bottom Nav: Gear icon
    ├── [Bell icon]      → Alarms (AlarmScreen)
    ├── [Bug icon]       → Diagnostics (DiagScreen)
```

```
    ├── [NFC button]      → NFC (NfcScreen)
    └── [QR button]       → QR Code (QrScreen)
```

3. Home Screen (MainScreen)

Access: Launch app – the home screen appears automatically as the first screen.

The home screen provides quick, direct access to the Modbus protocol without prior structure configuration. All four areas are on a single scrollable screen.

3.1 Establishing a Connection

Path: Launch app → "Connection" area is immediately visible (first section).

Step	Control	Description
1	"Connection type" dropdown	Tap → Select: **Modbus TCP** · **RTU/USB-Serial** · **RTU/Bluetooth** · **RTU/WiFi**
2a	"Master/Slave" radio button	Visible for TCP only – select role
2b	IP + Port fields	For TCP and RTU/WiFi: enter IP address and port
2b	"USB device" dropdown + 🔄	For USB-Serial: select device, scan button to search, enter baud rate
2b	"BT device" dropdown + 🔄	For Bluetooth: select device, scan button to search
3	"Slave ID" field	Enter unit ID of target device (default: 1)
4	**"Connect"** button	Tap → connection is established; status appears below
—	**"Disconnect"** button	Tap → connection ends (appears after successful connection)

> ***Note:** All fields are locked during an active connection. Disconnect first, then change configuration.*

3.2 Reading Registers (Mode 0 – Raw)

Path: Launch app → scroll down → "Read Registers (Master)" area → select **"4 Registers"** segment.

Step	Control	Description
1	Establish connection	(see 3.1)
2	**"4 Registers"** segmented button	Tap (left segment) – default
3	"Start address" field	Enter decimal start address
4	**"Read"** button	Tap → 4 registers from start address are read
—	4 read-only fields	Results shown directly in Reg 1–4 fields

3.3 Reading Registers (Mode 1 – Data Type)

Path: Launch app → scroll down → "Read Registers (Master)" area → select **"Data type"** segment.

Step	Control	Description
1	Establish connection	(see 3.1)
2	**"Data type"** segmented button	Tap (right segment)
3	"Data type" dropdown	Tap → select type (BOOLEAN, INT16, FLOAT32, etc.)
4	"Start address" field	Enter decimal start address
5	**"Read"** button	Tap → value is read and shown converted
—	Result box	Shows converted value (bold) + hex register info below

3.4 Writing Registers (Mode 0 – Raw)

Path: Launch app → scroll down → "Write Registers (Master)" area → **"4 Registers"** segment.

Step	Control	Description
1	Establish connection	(see 3.1)
2	**"4 Registers"** segmented button	Tap (left segment)
3	"Start address" field	Enter target address
4	"Reg 1" – "Reg 4" fields	Enter decimal values
5	**"Write"** button	Tap → values are written to device

3.5 Writing Registers (Mode 1 – Data Type)

Path: Launch app → scroll down → "Write Registers (Master)" area → **"Data type"** segment.

Step	Control	Description
1	Establish connection	(see 3.1)
2	**"Data type"** segmented button	Tap (right segment)
3	"Data type" dropdown	Select type
4	"Start address" field	Enter target address
5	"Value" field	Enter value to write
—	Register preview	Live display of raw register values (e.g. "Reg0: 0x0042 (66)")
6	**"Write"** button	Tap → converted value is written

3.6 Viewing the Protocol Log

Path: Launch app → scroll all the way down → "Protocol" area.

- Shows the last 10 communication events in hex format
- Updated automatically after each read/write
- No manual trigger needed

4. Structure List (StructureListScreen)

Access: Home Screen → System back button

(The structure list is the central management interface of the app.)

The structure list shows all configured Modbus devices/plants. An empty list shows the hint "Tap + to create a new structure".

4.1 Toolbar Icons (left to right)

Icon	Name	Action
⚗ (Flask)	Load demo	Loads sample data
↗ (Share)	Export JSON	Share all structures as JSON file (share dialog)
📂 (Folder)	Import menu	Dropdown opens (see 4.3)

🔔 (Bell)	Alarms	Opens AlarmScreen; red badge shows unacknowledged alarms
🐛 (Bug)	Diagnostics	Opens DiagScreen (protocol tracer)
⚙ (Gear)	Settings	Opens SettingsScreen

4.2 Create New Structure

Path: Structure List → **"+"** button (blue round button, bottom right) → StructureFormScreen

4.3 Importing

Path: Structure List → **Folder icon** in toolbar → dropdown appears:

Menu item	Action
Import JSON	File picker opens (JSON or text file)
Import JSON (Text)	Dialog with text field opens – paste/type JSON directly
Import EDS	File picker for CANopen EDS files
Import GSD	File picker for PROFIBUS GSD files

After successful import, a short confirmation snackbar appears.

4.4 Open Structure

Path: Structure List → Tap on structure card → StructureDetailScreen

4.5 Edit Structure

Path: Structure List → Pencil icon (✏) on right of structure card → StructureFormScreen

4.6 Delete Structure

Path: Structure List → Trash icon (🗑) on right of structure card → Confirmation dialog → tap "Delete"

4.7 Export (JSON)

Path: Structure List → Share icon (↗) in toolbar → System share dialog → select target app

5. Structure Detail (StructureDetailScreen)

Access: Structure List → Tap on structure card

Core of the app – shows all configured details of a plant in four tabs. The top bar shows: **structure name** + **IP:Port**. The back arrow (←) returns to the structure list.

Tab Navigation

The four tabs are accessible via the tab bar below the toolbar:

Tab	Icon	Function
Read	↓	Read values, polling, recording
Write	↑	Write values
Table	☰	All values in tabular form
Dashboard	⊞	Visual widgets, drag-and-drop

5.1 "Read" Tab – Reading a Single Value

Path: Structure Detail → "Read" tab → Detail card → **"Read"** button

Step	Control	Description
1	Tap "Read" tab	If not already active
2	**"Read All"** button (top)	Read all details at once
— or —		
2	**"Read"** button (on detail card)	Read only this one detail
—	Result box	Green background = OK, red background = error

5.2 "Read" Tab – Starting Polling (Cyclic Reading)

Path: Structure Detail → "Read" tab → enter interval → "Start Polling"

Step	Control	Description
1	**"Interval (s)"** field	Enter desired seconds (e.g. 5)
2	**"Only on change"** toggle	Optional: update UI only when value changes
3	**"Start Polling"** button	Tap → cyclic reading begins
—	Status banner	Appears during active polling: "Polling active – every X s"
4	**"Stop"** button (red)	Stop polling

5.3 "Read" Tab – Starting/Stopping Recording

Path: Structure Detail → "Read" tab → "Recording" toggle

Step	Control	Description
1	**"Recording"** toggle	On → recording starts (red dot

		icon in toggle)
—	Status text	"Values are being saved" during active recording
2	**"Recording"** toggle	Off → session is saved to database

Recorded sessions can be retrieved in LogScreen.

5.4 "Write" Tab – Writing a Value

Path: Structure Detail → "Write" tab → enter value → "Write"

Step	Control	Description
1	Tap **"Write"** tab	
2	**"Write confirmation"** toggle	Optional: automatically read back after writing (✓ if matches)
3	**"Enter value"** text field	Enter value in the respective detail field
4	**"Write"** button (on detail card)	Write only this detail
— or —		
4	**"Write All"** button (top)	Write all changed values at once

5.5 "Table" Tab – Table View

Path: Structure Detail → "Table" tab

- No interaction needed – automatically shows all last-read values
- Columns: **Name** · **Address** · **Raw value** (hex + dec) · **Decoded** (converted value)
- Alternating row colors, threshold color coding in the "Decoded" column
- Scroll for many details

5.6 "Dashboard" Tab – Showing Widgets

Path: Structure Detail → "Dashboard" tab

- Shows details as visual widgets (LED, value, gauge, chart)
- Widgets are automatically populated from the last read operation

Changing widget order (drag-and-drop):

Step	Control	Description
1	**Long-press widget**	After ~0.5 s, drag mode begins
2	**Drag widget up/down**	Widget changes position once threshold is crossed
3	Release	New order is saved

Edit or delete a widget:

- Pencil icon (✏) at top right of widget card → DetailFormScreen
- Trash icon (🗑) at top right of widget card → confirmation dialog

5.7 Add Detail

Path: Structure Detail (Read, Write, or Table tab) → **"+"** button (bottom right) → DetailFormScreen

> *The FAB is not visible in the Dashboard tab.*

5.8 Edit Detail

Path: Structure Detail → Detail card → Pencil icon (✏) → DetailFormScreen

(Available in Read, Write, and Dashboard tabs)

5.9 Delete Detail

Path: Structure Detail → Detail card → Trash icon (🗑) → Confirmation dialog → "Delete"

5.10 Edit Structure

Path: Structure Detail → Pencil icon (✏) in TopAppBar (top right) → StructureFormScreen

6. Create/Edit Structure (StructureFormScreen)

Access (new): Structure List → "+" button

Access (edit): Structure List → Pencil icon on card

or: Structure Detail → Pencil icon in toolbar

All fields are on a scrollable screen. The back arrow (←) discards changes. The **"Save"** button at the bottom saves and navigates back.

Field	Required	Description
Name	✓	Plant name (error if empty)
Description		Free text note
IP Address	✓	Default IP for all details

Port	✓	Default port (default: 502)
Unit ID		Modbus unit/slave ID (default: 1)

7. Create/Edit Detail (DetailFormScreen)

Access (new): Structure Detail → "+" button

Access (edit): Structure Detail → Pencil icon on detail card

or: StructureDetailScreen Dashboard → Pencil icon on widget card

The screen is divided into sections, scrolled through from top to bottom. **"Save"** stores and navigates back.

7.1 "General" Section

Path: DetailFormScreen → immediately visible at top

Field	Description
Name	Label (required)
Description	Free text note

7.2 "Connection" Section

Path: DetailFormScreen → scroll down → "Connection" section

Field	Description
IP Address	Leave empty = use structure IP
Port	Leave 0 = use structure port
Unit ID	Modbus slave ID for this detail
Register address	Register start address (decimal)
Array size	Number of elements for arrays/strings

7.3 "Data Type" Section

Path: DetailFormScreen → scroll down → "Data type" section

Field	Selection	Description

Data type	Dropdown	BOOLEAN · BYTE · INT16 · UINT16 · INT32 · UINT32 · INT64 · UINT64 · FLOAT32 · FLOAT64 · STRING · RAW
Function code	Dropdown	FC01–FC2B (read and write functions)
Byte order	Dropdown	Big Endian (ABCD) · Little Endian (DCBA) · Word Swap (CDAB) · Byte Swap (BADC)

7.4 "Conversion" Section

Path: DetailFormScreen → scroll down → "Conversion" section

Field	Description
Scale	Multiplier: display value = (raw value × scale) + offset
Offset	Addition after scaling
Unit	Displayed text (e.g. °C, bar, A)

7.5 "Visualization" Section

Path: DetailFormScreen → scroll down → "Visualization" section

Field	Selection / Description
Widget type	AUTO · Value · LED · Gauge · Chart
Gauge minimum	Lower end of gauge display
Gauge maximum	Upper end of gauge display
Warning threshold	Above this value → orange coloring
Alarm threshold	Above this value → red coloring

7.6 "Alerting" Section

Path: DetailFormScreen → scroll all the way down → "Alerting" section

Field	Description
"Alarm on value change" toggle	Creates an alarm event and system notification on every value change

8. Recordings (LogScreen)

Access: Structure List → **Chart icon** (📈) in toolbar

Shows all recorded measurement data sessions in a scrollable list.

8.1 Viewing Sessions

Path: Structure List → Chart icon → LogScreen

All sessions are listed automatically (newest first).

Each list entry shows:

- Start date and time
- Name of the recorded structure
- Duration (if session is ended)
- Number of saved measurements

8.2 Export Recording as CSV

Path: LogScreen → session in list → **Share icon** (↗) on session card → share dialog → select target app

CSV format:

```
Timestamp;Name;RawValue;Value;Unit
2024-01-15 14:23:05;Temperature;1234;12.34;°C
```

8.3 Delete Session

Path: LogScreen → session in list → **Trash icon** (🗑) → confirmation dialog → "Delete"

> *Deleting a session also removes all associated measurement entries.*

9. Alarms (AlarmScreen)

Access: Structure List → **Bell icon** (🔔) in toolbar

(Red number badge shows count of unacknowledged alarms)

9.1 Open Alarm Overview

Path: Structure List → Bell icon → AlarmScreen

- **"All" tab:** All alarm events in descending chronological order
- **"Unacknowledged" tab:** Only events not yet acknowledged

Switch tab: Tap the two tabs at the top.

9.2 Acknowledge Alarm

Path: AlarmScreen → alarm card → **"Acknowledge"** button

- The event is retained but removed from the "Unacknowledged" tab
- Badge counter in structure list decreases

9.3 Acknowledge All Alarms

Path: AlarmScreen → **"Acknowledge all"** button in toolbar (top right)

Visible only when unacknowledged events exist.

9.4 Delete Single Alarm

Path: AlarmScreen → alarm card → **"Delete"** button

9.5 Delete All Alarms

Path: AlarmScreen → **"Delete all"** button in toolbar

Alarm types and their triggers:

Type	Trigger	Notification priority
Warning	Warning threshold exceeded	Normal
Alarm	Alarm threshold exceeded	High
Value change	Value has changed (toggle enabled)	Low
Connection lost	Device no longer reachable	High

10. Diagnostics (DiagScreen)

Access: Structure List → **Bug icon** (🐛) in toolbar

Protocol analysis and statistics for all Modbus communication. Records automatically when enabled.

10.1 Start/Stop Recording

Path: DiagScreen → **Play/Pause icon** in toolbar (top right)

- During active recording: red "REC" badge visible in title
- Ring buffer stores the last 500 frames

10.2 View Statistics

Path: DiagScreen → **"Statistics"** tab (default tab)

Shows automatically:

- Total transactions · Successes · Errors
- Success rate in %
- Latency: average / min / max / P95
- Error breakdown: timeouts · CRC errors · exceptions · other

10.3 View Individual Frames / Hex Dump

Path: DiagScreen → tap **"Frames"** tab

Step	Control	Description
1	Tap "Frames" tab	List of all recorded frames
2	Filter chips (All / OK / Errors)	Filter frames
3	Tap frame card	Expands → shows hex dump + MBAP info

10.4 Export PCAP File (for Wireshark)

Path: DiagScreen → **PCAP icon** in toolbar → share dialog → select target app

> *The exported `.pcap` file contains valid Ethernet + IPv4 + TCP + Modbus-MBAP headers and can be opened directly in Wireshark.*

10.5 Clear Frame Buffer

Path: DiagScreen → **Trash icon** in toolbar → confirmation dialog → "Delete"

11. Slave Mode (SlaveScreen)

Access: Bottom Navigation → **Memory icon** (Slave)

Runs the app as a Modbus TCP server. Useful for testing PLC programs or visualization software.

11.1 Start Slave Server

Path: SlaveScreen → **"Server"** tab → enter port and unit ID → **"Start Server"** button

Step	Control	Description
1	**"Port"** field	Enter TCP port (default: 502)
2	**"Unit ID"** field	Enter desired unit ID
3	**"Start Server"** button	Tap → server listens for incoming connections
—	Status badge	"ACTIVE :502" appears in title when active
4	**"Stop Server"** button	Tap → server stops

11.2 Configure Response Delay

Path: SlaveScreen → "Server" tab → **"Response delay"** slider

- Slider from 0 to 5000 ms
- Delay applies to all responses (simulates slow devices)

11.3 Force Exception Responses

Path: SlaveScreen → "Server" tab → "Simulate exception" section

Step	Control	Description
1	**"Enable exception"** toggle	Turn on
2	**"Exception code"** dropdown	01 Illegal Function · 02 Illegal Data Address · 03 Illegal Data Value · 04 Server Failure
3	**"Target"** radio	All addresses or address range only
4	"From" / "To" fields	For address range: enter range

11.4 Edit Holding Registers

Path: SlaveScreen → **"Registers"** tab

Step	Control	Description
1	**"From address"** / **"Count"** fields	Restrict the displayed range
2	Tap register row	Edit value inline
3	Check button	Tap → value is applied

Table shows: Address · Value in hex · Value in decimal

11.5 Edit Coils

Path: SlaveScreen → **"Coils"** tab

- 2-column overview of 16 coils
- Per coil: LED indicator (green = 1, gray = 0) + switch to toggle
- Flipping switch → coil state changes immediately

11.6 Using the Value Generator

Path: SlaveScreen → **"Generator"** tab

Step	Control	Description
1	**"Mode"** dropdown	OFF · RANDOM · RAMP · SINE
2	**"Target address"** field	Register start address
3	**"Min"** / **"Max"** fields	Value range

4	**"Period (s)"** field	Period duration for RAMP/SINE
5	**"Step size"** field	Increment per tick (RAMP)
6	**"Interval (ms)"** field	Update interval
7	**"Start generator"** button	Tap → registers are populated automatically
8	**"Stop generator"** button	Tap → stop generator

12. NFC (NfcScreen)

Access: Settings (SettingsScreen) → **"NFC"** button

Transfers connection profiles (name, IP, port, unit ID) via NFC tag.

> *Prerequisite: The Android device must support NFC and NFC must be enabled in Android system settings.*

12.1 Write Connection Profile to NFC Tag

Path: SettingsScreen → NFC button → NfcScreen → **"Write"** tab

Step	Control	Description
1	Tap **"Write"** tab	
2	**"Structure"** dropdown	Select structure to transfer
3	**"Write to tag"** button	Tap → NFC reader mode is activated
4	Hold NFC tag to device	Once detected → profile is written to tag
—	Status message	Success or error message appears

12.2 Read Connection Profile from NFC Tag

Path: SettingsScreen → NFC button → NfcScreen → **"Read"** tab

Step	Control	Description
1	Tap **"Read"** tab	NFC reader is automatically active
2	Hold NFC tag to device	Profile is automatically read
—	Display	Name + IP:Port of read profile appears

13. QR Code (QrScreen)

Access: Settings (SettingsScreen) → **"QR Code"** button

Shares connection profiles as QR codes (JSON content).

13.1 Display QR Code of a Connection Profile

Path: SettingsScreen → QR Code button → QrScreen → **"Display"** tab

Step	Control	Description
1	Tap **"Display"** tab	
2	**"Structure"** dropdown	Select structure
—	QR code	Generated immediately (512×512 pixels)
3	**"Export"** button	Share QR code image via share intent

13.2 Scan QR Code and Import Structure

Path: SettingsScreen → QR Code button → QrScreen → **"Scan"** tab

Step	Control	Description
1	Tap **"Scan"** tab	
2	Camera dialog	Opens automatically (ZXing scanner)
3	Point camera at QR code	
—	Automatically	Structure is imported from scanned JSON

14. Settings (SettingsScreen)

Access: Structure List → **Gear icon** (⚙) in toolbar

14.1 Change Color Scheme

Path: SettingsScreen → **"Appearance"** section → tap filter chips

Option	Description
Light	Light Material 3 theme
Dark	Dark Material 3 theme
System	Follows Android system setting (default)

Change is applied immediately without restart.

14.2 Change Language

Path: SettingsScreen → **"Language"** section → tap dropdown → select language

Option	Description
Deutsch (de)	German interface
English (en)	English interface

Change is applied immediately without restart.

14.3 Configure Home Screen Widget

Path: SettingsScreen → **"Home Screen Widget"** section → "Structure" dropdown

- Selects which structure is displayed on the home screen widget
- Widget updates after each read cycle of the selected structure

14.4 Open NFC Screen

Path: SettingsScreen → **"NFC"** button → NfcScreen

14.5 Open QR Screen

Path: SettingsScreen → **"QR Code"** button → QrScreen

15. Home Screen Widget

Setup: Android home screen → add widget → select "Modbus++" → place widget

Configure: SettingsScreen → "Home Screen Widget" section → select structure

The widget shows on the home screen:

- **Structure name** (top left, bold)
- **Last measured value** (large, bold)
- **Timestamp** of last update (bottom, light blue)

Background color: Blue (#1565C0)

Update: Automatically after each polling cycle of the selected structure – no manual action needed.

16. Wear OS

Setup: Pair Wear OS smartwatch with Android device → Modbus++ runs automatically on the watch as soon as data is sent.

The Wear OS app receives measurements automatically after each read cycle:

Display elements on the watch:

- Timestamp of last update (top)
- Scrollable list of all readings (ScalingLazyColumn)
- Per entry: **Name** (small) · **Value** (large) · **Unit**
- "No data" placeholder when no values have been sent yet

Data transfer: Wearable Data API, path `/modbus/readings` – triggered automatically after each successful `readAll()` operation.

17. Data Models

ModbusStructure

```
id           – UUID (auto-generated)
name         – Plant name
description  – Description
ip           – Default IP address (default: 192.168.1.1)
port         – Default port (default: 502)
unitId       – Default unit ID (default: 1)
details      – List of ModbusDetail entries
```

ModbusDetail

```
id                – UUID
name              – Label
description       – Description
dataType          – Data type (12 options)
arraySize         – Array size
ip / port / unitId – Connection override (empty/0 = use structure)
registerAddress   – Modbus register address
functionCode      – FC01–FC2B
byteOrder         – Big/Little/WordSwap/ByteSwap
scale / offset    – Scaling: (raw value × scale) + offset = display value
unit              – Unit text
widgetType        – AUTO / VALUE / LED / GAUGE / CHART
gaugeMin/Max      – Range for gauge display
thresholdWarn     – Warning threshold (optional)
thresholdAlarm    – Alarm threshold (optional)
alarmOnChange     – Alarm on every value change
```

Data Types

Type	Description	Registers
BOOLEAN	1 bit (coil)	1
BYTE	8 bit	1
INT16	16-bit signed	1
UINT16	16-bit unsigned	1
INT32	32-bit signed	2
UINT32	32-bit unsigned	2
INT64	64-bit signed	4

UINT64	64-bit unsigned	4
FLOAT32	IEEE 754 single precision	2
FLOAT64	IEEE 754 double precision	4
STRING	ASCII (1 byte/register)	variable
RAW	Raw register 16-bit	1

18. Modbus Protocol Implementation

Connection Types (ModbusConnection)

TcpConnection

- New TCP socket per request
- MBAP framing: Transaction ID (2 bytes) · Protocol ID (0x0000) · Length (2 bytes) · Unit ID · PDU
- Timeout: 3000 ms

UsbSerialConnection

- USB-OTG via `hoho-android-usbserial`
- Configurable: baud rate, data bits, parity, stop bits
- DTR/RTS enabled

BluetoothConnection

- SPP profile (UUID: 00001101-0000-1000-8000-00805F9B34FB)
- Polling-based receive (10 ms delay)

RtuWifiConnection

- TCP socket to serial bridge
- RTU framing (CRC-16) over TCP

RTU Framing (ModbusRtu)

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

Supported Function Codes

Reading:

- 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)

Writing:

- 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 Orders

Name	Order	Example (value 0x12345678)
Big Endian (ABCD)	Standard network	12 34 56 78
Little Endian (DCBA)	Intel format	78 56 34 12
Word Swap (CDAB)	Mitsubishi format	56 78 12 34
Byte Swap (BADC)	PLC format	34 12 78 56

File Import

EDS (CANopen Electronic Data Sheet):

- INI format: `[Device]` → device name, `[Param###]` → parameter list
- Automatically maps to Modbus details with correct data types

GSD (PROFIBUS General Station Description):

- Key-value format: `Vendor_Name`, `Model_Name`, `Module "Name" data`
- Creates detail entries per module

19. Database

Room database (version 2) with three tables:

log_sessions

Recording sessions per structure

```
id            - UUID (PK)
structureId   - Reference to structure
structureName - Name at time of recording
startTime     - Unix timestamp
endTime       - Unix timestamp (null if active)
```

log_entries

Individual measurements of a session

```
id            – UUID (PK)
sessionId     – Reference to session
structureId   – Reference to structure
detailId      – Reference to detail
detailName    – Name at time of recording
timestamp     – Unix timestamp
rawValue      – Raw value as string (comma-separated for arrays)
displayValue  – Converted value with unit
unit          – Unit
```

alarm_events

Alarm events and their acknowledgement status

```
id            – UUID (PK)
timestamp     – Unix timestamp
type          – THRESHOLD_WARN / THRESHOLD_ALARM / VALUE_CHANGE /
CONNECTION_LOST
structureId   – Reference to structure
structureName – Name at time of alarm
detailId      – Reference to detail (optional)
detailName    – Name at time of alarm (optional)
value         – Triggering value (optional)
message       – Alarm description
acknowledged  – Boolean (false = unacknowledged)
```

20. Permissions

Permission	Purpose
`INTERNET`	Modbus TCP connections
`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 notifications (API 33+)
`USB_HOST` (feature)	USB-OTG communication
`NFC` (feature)	NFC (marked as optional)

> ***Note:** USB-Serial communication requires no explicit runtime permission – permission is granted via a system dialog for the respective USB device. Bluetooth permissions are requested automatically on first scan. Post-Notifications permission is requested automatically on first alarm (Android 13+).*

Documentation updated 2026-05-21 · Modbus++ v1.0