

Portfolio: Garmin Connect IQ Apps

Open-source applications for smart home control and EV energy management

March 2026

Over the past years, we have been actively involved in the **open-source community** and have developed two major applications for Garmin devices using the **Connect IQ platform**.

openHAB is an **open-source home automation platform**. The app we developed under the umbrella of the openHAB Foundation enables users to access and control their smart home directly from their Garmin device. This includes controlling lights, adjusting air conditioning or blinds, managing music playback, and many other use cases.

evcc is an **open-source energy management system** designed for photovoltaic-optimized electric vehicle charging. Our application provides a concise overview of the system status, allowing users to quickly view information such as energy production, consumption, and EV charging activity.

Both projects rely heavily on the Connect IQ communications module. They integrate REST APIs and process complex data structures while operating **within the platform's strict resource constraints**. The applications also combine native and custom UI components to deliver an intuitive and user-friendly interface that runs smoothly **on both button-based and touch-based watches and bike computers**.

FURTHER READING

The broader technical and commercial context is discussed in our white paper

Developing for Garmin Devices, available at:

<https://metiq.com/developing-for-garmin-devices> ↗

openHAB is an open-source home automation platform designed to integrate hundreds of smart home technologies and devices from many different vendors into one unified system.

The **openHAB Garmin app is built around openHAB's sitemap concept**, which allows users to define a customized view of their smart home. The app renders this structure as a **recursive menu using native Connect IQ UI components**, as shown in the images below. This approach aligns the interface with the look and feel of the Garmin device it runs on and provides a navigation model that works well on **small wearable screens**.

KEY CHALLENGES

- **Flexible, customizable UI**
- Works across **touch and button devices** with **round and rectangular displays**
- **Large, recursive data structures** operating **at the limits of Connect IQ** (memory, storage, networking, CPU)
- Supports **90+ Garmin devices**



Some items can be controlled directly from the menu, for example simple toggle switches. More complex interactions open **dedicated views**. Some of these controls are **generic and reusable across different item types**. Examples include a text selection control (first picture below) and a numeric picker that can be used for setting values such as temperature, audio volume, or dimmer levels (second picture below). Other views are designed for specific use cases, such as media player controls.

Following **Garmin's UI conventions**, the interaction model adapts to the device type. On **button-based devices**, the interface provides button input hints (third picture). On **touch-enabled devices**, the same functions are presented using touch buttons (fourth picture).



Technical Architecture

Networking and Data Processing

The app communicates with the openHAB server through its **REST API** using the **Connect IQ communications module**. It retrieves item states and sends commands over HTTP while processing JSON responses within **strict memory and performance limits**.

Execution Model

The Connect IQ sandbox enforces **strict limits on consecutive CPU execution time**. To operate within these limits, the app uses a **lightweight scheduler** that executes work in small chunks.

This approach also helps maintain **UI responsiveness in a single-threaded runtime**. It **enables incremental processing of large data structures** and avoids deep recursion, which would exceed the platform's **limited call stack**.

UI Architecture

The UI is built around a **custom implementation based on Connect IQ's CustomMenu**, which forms the core navigation structure. Image resources are used extensively for icons, while some UI elements are rendered programmatically using **drawing primitives**.

Platform Compatibility

The app supports devices starting with **Connect IQ 3.4**. Features from **Connect IQ 4.0 and 5.1** are used where available, with compatibility layers for earlier devices.

Connectivity Constraints

Connect IQ imposes strict networking limitations. Continuous networking is only available via a **Bluetooth phone connection**. **Wi-Fi** can only be used in a dedicated sync mode with restricted requests and interrupted user flow. If no phone is available, the app automatically falls back to a **degraded Wi-Fi mode** that still allows commands to be sent.

CONNECT IQ COMPONENTS

- **Connect IQ SDK** and **Monkey C**
- **Communications Module** (HTTP / REST API integration)
- **Wi-Fi Sync** for direct device connectivity
- **Storage Module**
- **App settings** for user configuration
- **Monkey Jungle** build configuration
- **Graphics Module** for custom UI rendering

FURTHER READING

App Store: <https://apps.garmin.com/apps/d83b5e1b-4a12-4eda-bbad-26f88952a959> ↗

User Manual: <https://next.openhab.org/docs/apps/garmin> ↗

Source Code: <https://github.com/openhab/openhab-garmin> ↗

evfcc is an open-source energy management system designed to optimize energy usage based on locally generated photovoltaic power. It integrates EV chargers, solar inverters, home batteries, and household consumption data to intelligently control when and how **electric vehicles and other surplus-controlled devices** are powered. By prioritizing surplus solar energy, evfcc helps maximize self-consumption and reduce grid dependency, while providing users with a clear overview of energy production, consumption, and charging activity.

The **evfcc Garmin app** provides this overview directly on the device, presenting key information in a set of simple, well-structured screens that adapt to the user's configuration. The design is intentionally minimalistic, using iconography aligned with the evfcc web interface.

Key information is already visible in the watch's **glance carousel**, where users can see the current state of charge and charging activity of both house batteries and electric vehicles (**top left**). The **main view** provides a more detailed overview of the system, including energy production, grid import or feed-in, battery and vehicle state of charge, charging activity, and household consumption (**top middle**).

Additional views focus on **individual load points (top right)**, **production forecasts (bottom left)**, **grid price forecasts (bottom middle)**, and the **solar charging share (bottom right)**, giving users further insight into how effectively solar energy is being used for vehicle charging and other surplus-controlled loads.

KEY CHALLENGES

- **Simple, glanceable UI** that adapts to different device resolutions and screen types
- **Meaningful information within the limited glance interface**
- **Support for older devices** with as little as **64 kB of memory** of available app memory
- **Compatibility with multiple backend versions**
- Supports **85 Garmin devices**



Technical Architecture

Networking and Data Processing

The app communicates with the evcc server through its REST API using the Connect IQ communications module. On devices with only **64 kB of available app memory**, processing API responses required aggressive reduction of payload size. Server-side **JQ filtering** is used to strip the JSON responses down to only the fields required by the application.

Execution Model

To stay within **Connect IQ's strict limits on consecutive CPU execution time**, JSON processing and UI updates are split into small tasks executed by a **lightweight scheduler**, keeping processing within platform limits and the interface responsive.

UI Architecture

The UI is built around a **custom layout framework** that adapts to the available screen space and the user's device environment. Where supported, **vector fonts** are used to achieve precise sizing across different resolutions.

Icons are generated at build time for each device class to match the respective font sizes, avoiding expensive bitmap scaling on the device. UI views are also **pre-rendered in the background** where possible to improve perceived responsiveness.

Platform Compatibility

The app supports devices starting with **Connect IQ 3.0**. Features from later versions are used where available, while compatibility layers ensure correct behavior on earlier devices.

Backend Service

On some older devices, only **32 kB of memory** is available when the app runs as a glance. This is insufficient for both data processing and rendering. To address this, a **background service** retrieves and prepares the data in advance, allowing the glance component on these devices to focus solely on rendering.

CONNECT IQ COMPONENTS

- **Connect IQ SDK** and **Monkey C**
- **Communications Module** (HTTP / REST API integration)
- **Storage Module**
- **Background Service** for periodic data updates
- **App settings** for configuration
- **Monkey Jungle** build configuration
- **Graphics Module** for custom UI rendering

FURTHER READING

App Store: <https://apps.garmin.com/apps/2bc2ba9d-b117-4cdf-8fa7-078c1ac90ab0> ↗

User Manual: <https://evccg.metiq.com> ↗

Source Code: <https://github.com/METIQ-Solutions/evcc-garmin> ↗

About the Author

Robert has over 20 years of experience in the ICT industry, with a background in software development, IT architecture, and project management. He is a member of the management team at METIQ.

Alongside his professional work, Robert has a strong personal interest in residential solar systems and home automation. This led to the development of Garmin Connect IQ apps integrating with evcc, an open-source energy management system for photovoltaic-optimized electric vehicle charging, and openHAB, an open-source home automation platform.

If you are interested in discussing Garmin development, embedded platforms, or smart energy integrations, feel free to get in touch.

Robert Pollai
whitepaper-inquiry@metiq.com
www.metiqa.com



METIQ

METIQ Solutions s.r.o.
Karadžičova 8/A
821 08 Bratislava
Slovakia

Copyright © 2026 METIQ Solutions s.r.o. All rights reserved.

This document is provided for informational purposes only. The information contained herein is believed to be accurate at the time of publication but is provided without warranty of any kind, express or implied. METIQ Solutions s.r.o. shall not be liable for any errors, omissions, or for any damages arising from the use of this document. No part of this publication may be reproduced or distributed without prior written permission.