DATA MONITORING AND LOGGING SYSTEM FOR LITHIUM-ION BATTERY PACKS IN AGRICULTURAL ROBOTICS

Guido Soosaar, Tormi Lillerand
Estonian University of Life Sciences, Tartu, Estonia

Contact information:

Fr. R. Kreutzwaldi 56, 51006, Tartu, Estonia,
guido.soosaar@student.emu.ee

- **Raspberry Pi Zero 2 W** consumed
  - only **0.68 W** in *idle mode*
  - about **2.05 W** at *5-second polling intervals*
  - consistently **~50% less** than Raspberry Pi 5
- **E-ink display** had **minimal energy impact**
- **Dynamic polling** logic significantly **reduced data reads** during inactivity
- System supports **reliable operation** in **off-grid** field environments
- Enables **longer runtime**, **lower energy waste**, and **smarter monitoring**

These results confirm that our system is **well-suited** for energy-constrained **agricultural robotics**, combining low power use with real-time capability.

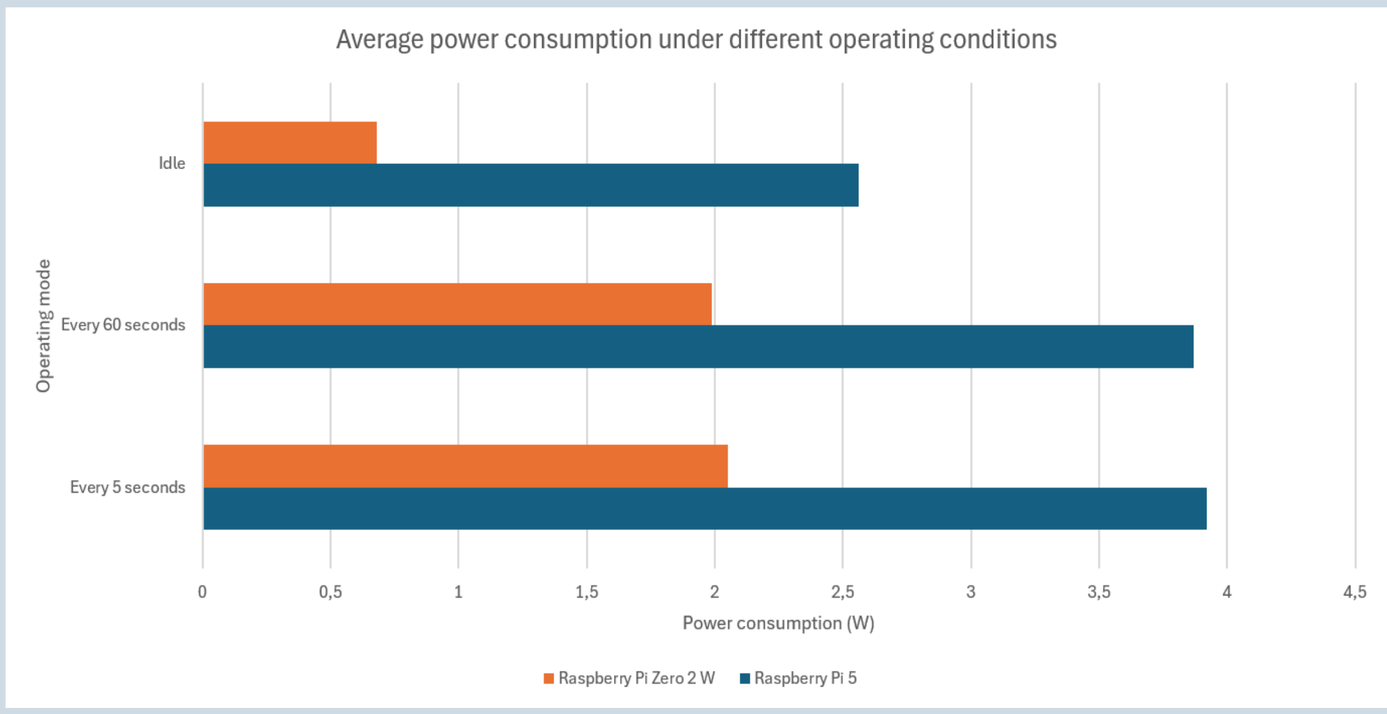

Fig 1. Average power consumption under different operating conditions.

Introduction

Battery-powered agricultural robots require efficient energy management to operate reliably in remote and off-grid environments. Continuous monitoring of battery parameters—voltage, current, temperature, and State of Charge (SoC)—is essential to prevent failures and extend runtime. This project presents a compact, energy-efficient battery monitoring system integrated with a CAN-based Battery Management System (BMS). It collects full-pack and cell-level data, logs it into an SQLite database, and offers both local and remote access through a web-based interface. The system's modular, low-power design enables long-term deployment in demanding field conditions.

Methodology

The system uses a Raspberry Pi Zero 2 W paired with a CAN interface to collect data from a lithium-ion Battery Management System (BMS). Key parameters—voltage, current, temperature, and SoC—are read via CAN bus and stored in an SQLite database. A Python-based monitoring script dynamically adjusts polling intervals based on battery conditions to reduce power usage. An e-ink display shows real-time data locally, while a web-based API enables remote access for diagnostics and performance tracking.

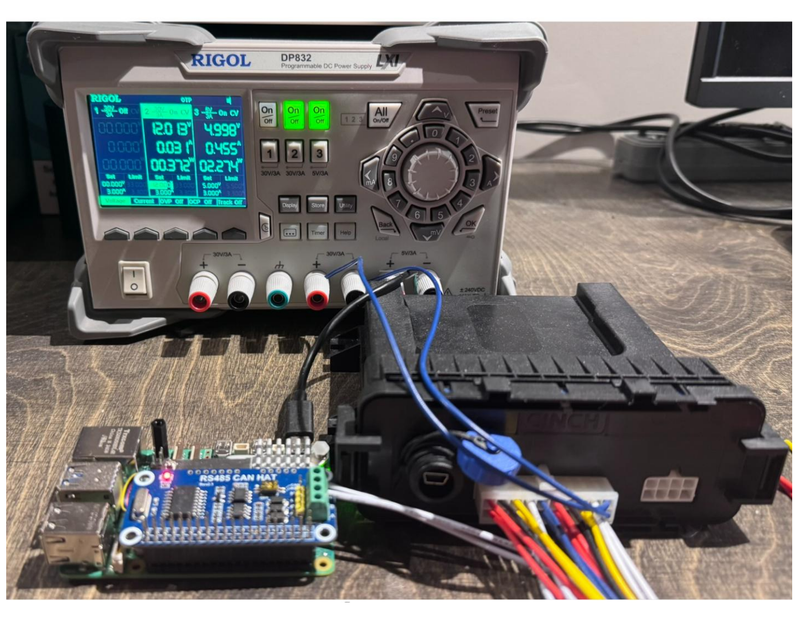

Fig 2. Measuring power draw

Results

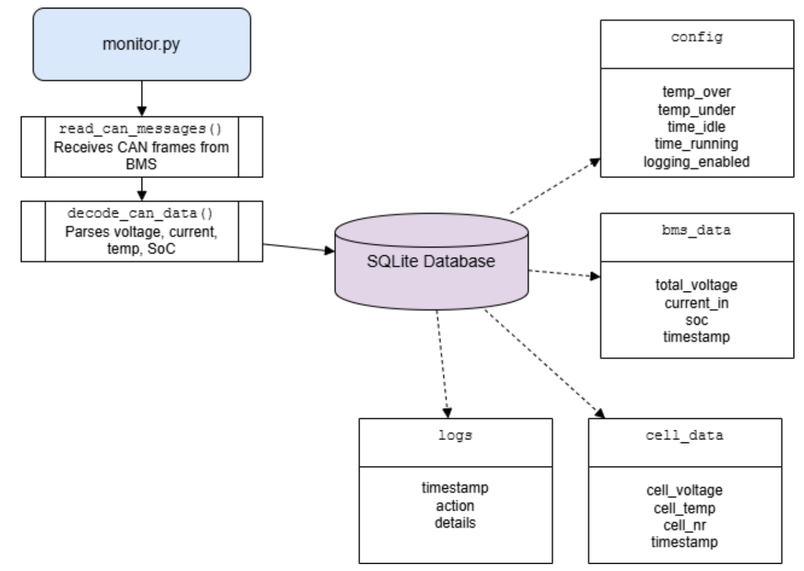

Fig 3. Battery data collection and storage architecture with SQLite and monitoring functions.

- The system **successfully logged** real-time battery data via **CAN bus**.
- Data included **voltage**, **current**, **temperature**, **SoC**, and **cell-level details**.
- All data was stored in a structured **SQLite database**, enabling efficient queries.
- System operated reliably in **field-like conditions** with **low energy use**.
- **Monitoring frequency** adapted to changing battery conditions.
- Local interface (e-ink) and remote API ensured **dual access** to data.
- The system architecture is ready for **AI-based analysis**, including:
  - **Battery degradation prediction**
  - **Charging pattern optimization**
  - **Remaining Useful Life (RUL)** estimation

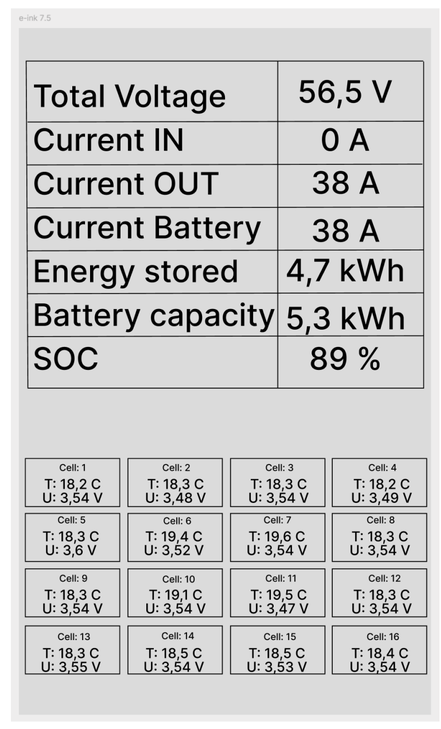

Fig 4. User interface on e-ink screen

These results demonstrate that the system not only performs **robust real-time monitoring** but also **lays the groundwork** for advanced analytics and **predictive maintenance** using machine learning.