2025 TEEP Progress Report Week-16

1. Description

Following the system stability improvements achieved last week, this week's main objective was to continue refining the 3-cell (3S) Li-ion battery monitoring system and begin transitioning from research and troubleshooting to integration and feature enhancement. With the ADS1115 and INA219 modules now performing more consistently, the focus shifted toward improving data smoothing, enhancing measurement precision, designing a more reliable hardware layout, and starting the initial phase of dashboard integration for real-time visualization.

This week's work served as the bridge between the detailed technical study completed earlier and the next stage of system development, where the monitoring system will evolve into a more user-friendly and robust application.

---

2. Work Done

The main tasks completed during Week 16 include:

2.1 Software Optimization

- Implemented moving average filtering on cell voltages to reduce sudden spikes and analog noise.
- Experimented with median filtering and compared stability between different window sizes.
- Improved the sampling interval logic so that ESP32 collects data at consistent time steps.
- Adjusted ADS1115 PGA and data rate settings to find an optimal balance between speed and accuracy.

2.2 Hardware Improvements

- Redesigned the voltage divider network with corrected resistor ratios to accurately map 4.2 V cell voltage into the ADC range.

- Added decoupling capacitors near the ADS1115 and INA219 modules to suppress electrical noise.
- Reorganized ground wiring to reduce common-mode interference.
- Identified weak jumper wires and replaced them with thicker, low-resistance ones to improve overall signal integrity.

2.3 System Integration Work

- Successfully tested combined voltage + current + pack measurement through a synchronized read cycle.
- Started drafting the data exchange format that will later be used by the Flask dashboard (JSON output via serial or WiFi).
- Planned the structure for ESP32 → Flask → Web Dashboard communication.

2.4 Testing and Validation

- Conducted long-duration measurement tests (30–45 minutes).
- Recorded fluctuations before and after applying software filters.
- Verified that calibration settings remain consistent across multiple power cycles.
- Logged measurement results for future comparison and debugging.

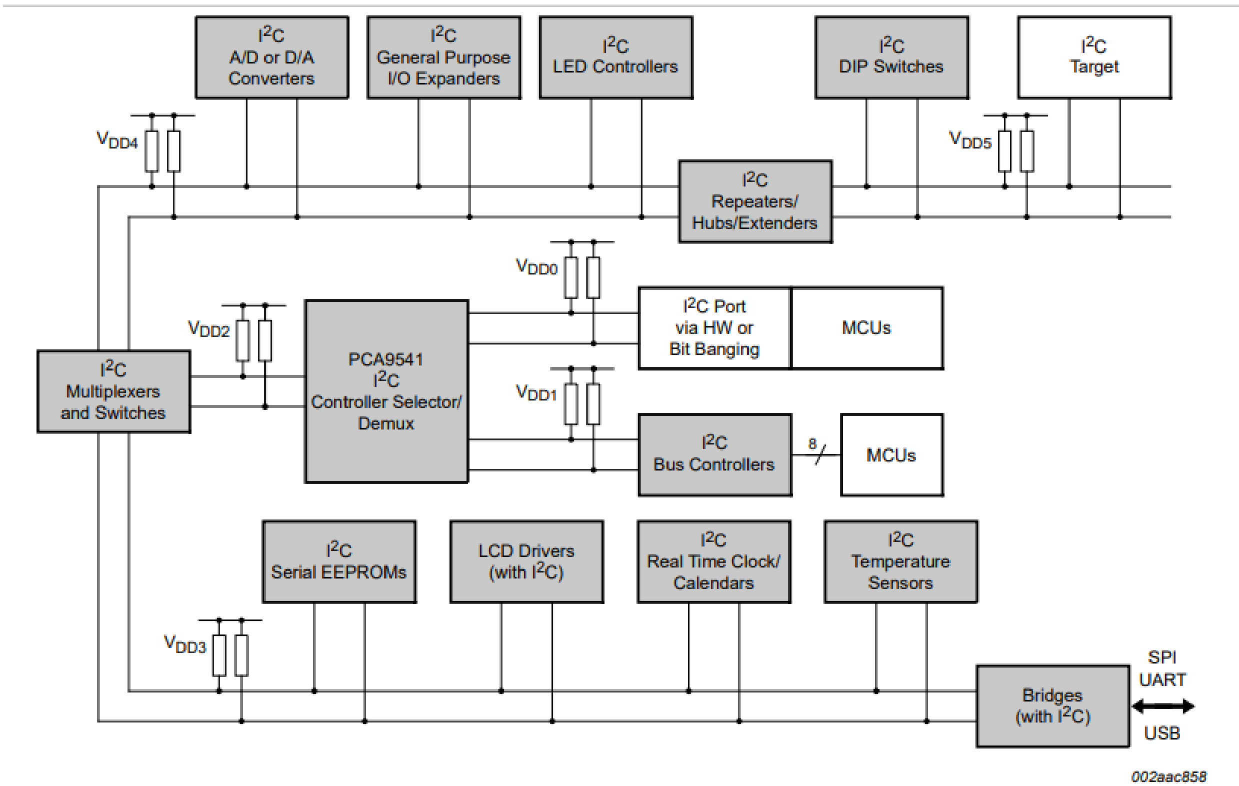
---

3. Observations and Findings

- Moving average filtering significantly improves display stability, especially for Cell-1 and Cell-3, which previously showed larger fluctuation bands.
- Hardware noise is still present at a very low level, but the combined effect of filtering and proper grounding keeps readings within acceptable tolerance.
- INA219 current measurements became more reliable once decoupling capacitors were added near the module.
- The ESP32 serial output is now much smoother and better suited for dashboard integration.

4. Challenges Faced

- Balancing filtering strength and responsiveness: heavy filtering stabilizes data but slows response to fast changes.
- Minor I2C delays appeared when both ADS1115 and INA219 were read rapidly at high sampling rates.



- Voltage divider precision depends heavily on resistor tolerance; even ±1% resistors can cause variations.
- ESP32 WiFi tasks occasionally interfered with I2C timing, requiring optimized delay management.

5. Next Steps

5.1 Complete Dashboard Integration

- Begin building the Flask dashboard UI using HTML, CSS, and Chart.js.
- Implement real-time data streaming from ESP32 to Flask (Serial first, WiFi later).
- Add graphs for:
  - Individual cell voltages
  - Total pack voltage
  - Charging/discharging current
  - Voltage vs. time plot

5.2 Advanced Filtering and Calibration

- Add Kalman filtering for higher accuracy on noisy current readings.
- Implement per-cell calibration constants inside ESP32 code.
- Store calibration values in EEPROM for persistence.

5.3 Hardware Enhancements

- Finalize PCB design for stable long-term operation.
- Add an external precision shunt for INA219 to increase measurement range.
- Improve thermal stability of voltage dividers using metal-film resistors.

5.4 System Documentation

- Prepare block diagrams and flowcharts for the final report.
- Summarize hardware and software revisions made so far.
- Plan the final demonstration video once full system stability is confirmed.

---

6. Resources Used

- TI ADS1115 datasheet and application notes
- Adafruit INA219 guide

- ESP-IDF and MicroPython documentation
- Tutorials on digital filtering (moving average, median, Kalman)
- Flask backend and Chart.js visualization examples

---

7. Reflection

This week marked an important transition from troubleshooting to structured development. The improvements in both software filtering and hardware noise reduction made the system significantly more stable and predictable. I also gained a deeper understanding of how precision sensors behave in real-world conditions and how small adjustments—both in code and hardware—can dramatically improve overall measurement quality.

Moving forward, the integration of the monitoring system with a Flask-based web dashboard will represent a major step toward completing the project. With the core sensing components now reliable, the upcoming weeks will focus on user interface design, backend communication, and final system validation.