CSCE 5612 Embedded Hardware/Software Design

Spring 2026

Lab #2: Data Logger and Power Consumption

Due: 1/21/2026 @ 11:59pm

Lab Description

The Smart Environmental and Activity Data Logger is a portable embedded system that reads IMU and temperature data at a fixed sampling frequency (same frequency for both sensors). You will use an RTC to timestamp data. The main goal of this lab is to quantify the power cost of SD logging by comparing power consumption in two firmware modes using a USB power analyzer. You will estimate the runtime using a 500 mAh, 3.7 V LiPo battery.

Mini OLED USB-C Power Analyzer used in this activity.

Operating Modes

You will implement **exactly two modes** and switch between them using a button and/or BLE (BLE is allowed for control/status only, not streaming). Use the Arduino BLE library to get started using BLE. You can use the example code "LED" under the peripheral section, and please install nRF Connect on your phone so you can interact with BLE.

Mode 1 — Sample Only (No SD Writes)

- IMU + temperature are sampled at the **same frequency** (e.g., 100 Hz) – Start with the code that you used in the previous lab.
- RTC timestamps each sample (or each record) – Start with the code that you used in the previous lab.
- **No SD card writes**.
- OLED shows: `MODE` – Start with the same code you used in the previous lab.

Purpose: baseline sensor + processing power without logging.

Mode 2 — Sample + Log (SD Logging Enabled)

- Same IMU + temperature sampling frequency as Mode 1
- RTC timestamps each record
- Data is written to the **SD card** (CSV or txt) – Use the SD card functions that you used in the previous lab.
- OLED shows: `MODE`

Purpose: measure the added energy cost of SD card logging.

USB Power Analyzer Measurements (What you measure)

You will measure the **average current** in both modes.

Procedure (Do this for each mode)

1. Connect the USB power analyzer inline between the USB power and the device.
2. Run the mode for **at least 60–120 seconds**.
3. Record:
   - V_{USB}
   - $I_{USB(avg)}$

Compute:

$$P_{USB} = V_{USB} \times I_{USB(avg)}$$

Battery Runtime Estimate (500 mAh @ 3.7 V)

Assume a regulator efficiency $\eta = 0.85$.

1. Convert measured USB power to estimated battery current:

$$I_{BAT(avg)} \approx \frac{P_{USB}}{3.7 \times 0.85}$$

2. Compute runtime for a **500 mAh** battery:

$$t(\text{hours}) \approx \frac{0.5}{I_{BAT(avg)}}$$

(0.5 Ah = 500 mAh)

You will report the estimated runtime for:

- **Mode 1 (Sample Only)**
- **Mode 2 (Sample + Log)**

And you will compute the **logging overhead**:

$$\Delta I = I_{USB(avg,log)} - I_{USB(avg,sample)}$$
$$\Delta P = P_{USB(log)} - P_{USB(sample)}$$

Data Format Requirement

Each record must include:

- Timestamp (RTC)
- IMU accel (Ax, Ay, Az)
- IMU gyro (Gx, Gy, Gz)
- Temperature

Example CSV header:
`hr,min,sec,ax,ay,az,gx,gy,gz,temp`

Deliverables

1. **Working firmware**

    - Two modes (Sample Only / Sample + Log)
    - Fixed sampling rate (same for IMU + temp)
    - OLED indicates mode + frequency
    - SD file contains correctly timestamped records in Mode 2

2. **Power + Battery Report**

    - A small table for both modes: V_{USB}, $I_{USB(avg)}$, P_{USB}
    - Battery runtime estimate (hours) for both modes (500 mAh @ 3.7 V)
    - Logging overhead (ΔI, ΔP) and a 2–3 sentence interpretation

3. **Example data file**

    - **2 minutes** of logged data from Mode 2

4. **Short demo video**

    - Show switching between modes and the analyzer current difference

Evaluation

- **System works + correct data format (50%)**
- **Clean power measurements (25%)**
- **Correct battery-life math + logging overhead analysis (25%)**