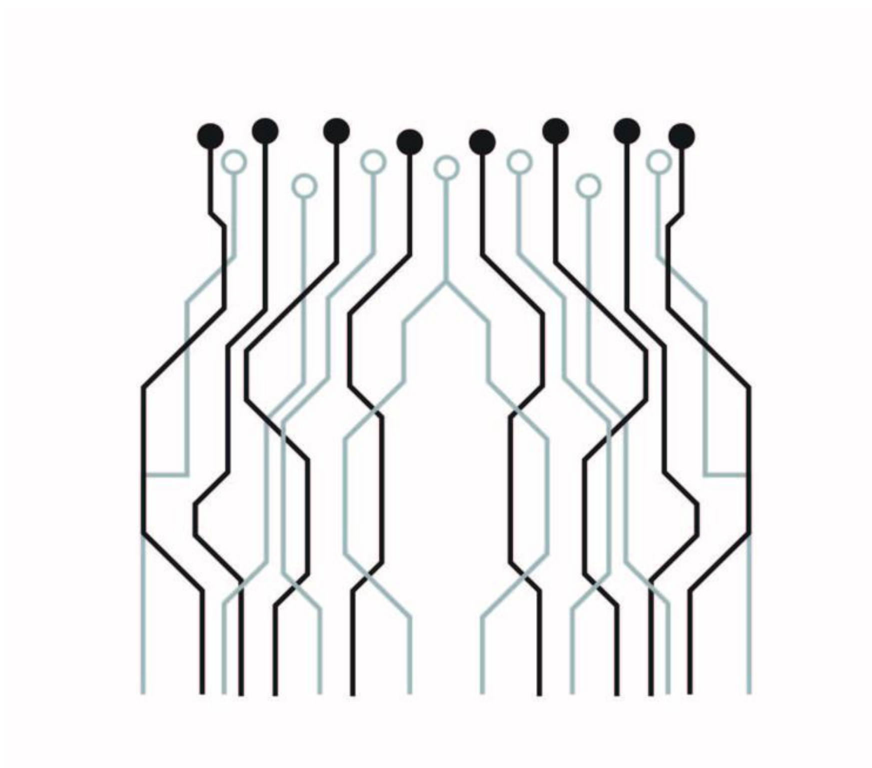
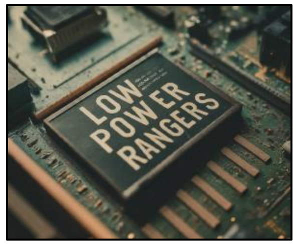
Low Power Embedded Gaming Application using Blue Gecko

ECEN 5833 Low Power Embedded Design Techniques



Team Members: Jayash Raulkar, Unmesh Phaterpekar & Tirth Patel



24.11.2024

Table of Content

Project Week 2 Update

Project Week 3 Update

Project Week 4 Update

Project Week 5 Update

Table of Figures

PROJECT PROPOSAL

PROJECT OVERVIEW

This project aims to design and develop a low-power gaming application using two Blue Gecko microcontrollers (MCUs) from Silicon Labs, leveraging Bluetooth Low Energy (BLE) communication. The project will involve creating a complete gaming system, where one microcontroller (Stick Gecko) acts as the BLE transmitter interfaced with various sensors and input devices, and the other microcontroller (Screen Gecko) acts as the BLE receiver, interfaced with a display to run the game application.

OBJECTIVES

- **Design & Development:** Create a functional gaming application using low-power embedded design techniques.
- **BLE Communication:** Implement BLE communication between two Blue Gecko MCUs.
- **Sensor Integration:** Interface various sensors (e.g., joystick, gyro sensor, ambient light sensor) with the MCUs to provide input to the game.
- **Power Management:** Ensure efficient power usage through energy harvesting and low-power design strategies.
- **PCB Design:** Design custom PCBs for both Stick Gecko and Screen Gecko to accommodate all components.
- **Firmware Development:** Write efficient firmware to handle sensor data, BLE communication, and display management.

SYSTEM DESCRIPTION

Stick Gecko (BLE Transmitter)

- **Microcontroller**: ARM Cortex-M4 core

  Components:

- **Joystick**: Interfaced using an ADC to capture user input for controlling the game.
- **Buttons**: Connected via GPIO for additional control functions.
- **Gyro Sensor**: Interfaced via I2C to detect motion, enhancing game interactivity.
- **Ambient Light Sensor**: Monitors the light in the surroundings and adjusts the LCD/OLED brightness as required.
- **Power Supply**: Battery powered with integrated energy harvesting, such as a solar panel, for extended operation.
- **Communication**: Sends sensor data and user inputs via BLE to the Screen Gecko.

Screen Gecko (BLE Receiver)

- **Microcontroller**: ARM Cortex-M4 core

  Components:

- **LCD/OLED Display**: Connected via SPI to display the game interface.
- **Power Supply**: Battery powered
- **BLE Communication**: Receives data from the Stick Gecko and processes it to update the game display.

BLOCK DIAGRAM

Block Diagram of the Stick Gecko

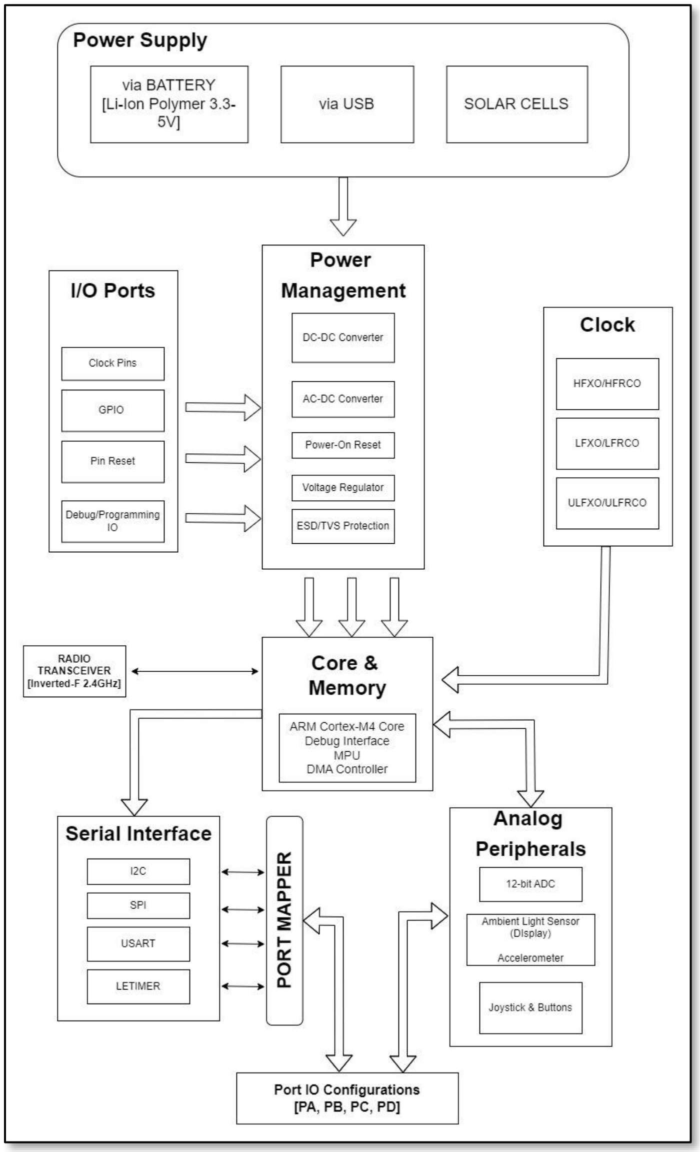


Figure 1 - Controller

The block diagram shows an embedded system powered by a Li-Ion battery, USB, or solar cells. A PMIC handles DC-DC and AC-DC conversion, along with ESD protection. The ARM Cortex-M4 processor with memory protection, debug interfaces, and a DMA controller manages processing. I/O components include GPIO, clock pins, and debug interfaces, with serial interfaces like I2C, SPI, USART, and LETIMER managed via a port mapper. Also supports analog peripherals - 12-bit ADC, sensors, and user controls like a joystick and buttons.

Block Diagram of the Screen Gecko

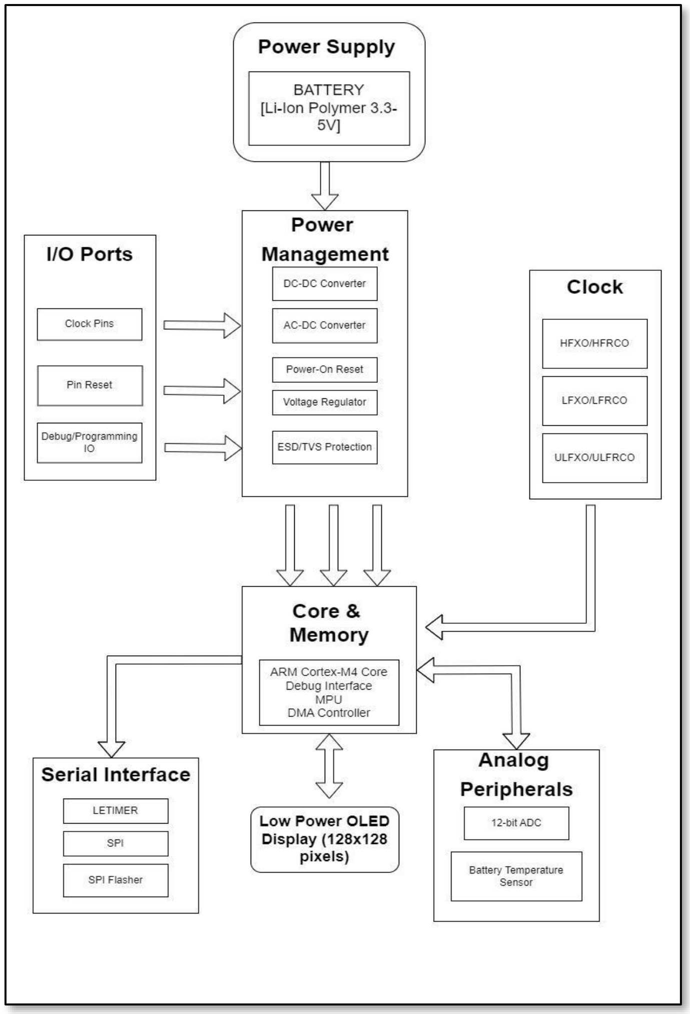


Figure 2 - Display

This block diagram shows an ARM Cortex-M4 processor at its core, powered solely by the Li-Ion Battery. Power management includes DC-DC/AC-DC conversion, voltage regulation, and ESD/TVS protection. The processor features a debug interface, MPU, and DMA controller, communicating via LETIMER and SPI. It controls a low-power 128x128 OLED display and supports analog peripherals like a 12-bit ADC and battery temperature sensor. A game logic will be implemented on this Blue GECKO and shall be controlled from the Stick Gecko (Above board) via BLE. The system's clock is managed by various oscillators (HFXO/HFRCO, LFXO/LFRCO, ULFXO/ULFRCO), and I/O includes clock pins, reset, and debug interfaces.

Block Diagram - BLE Communication

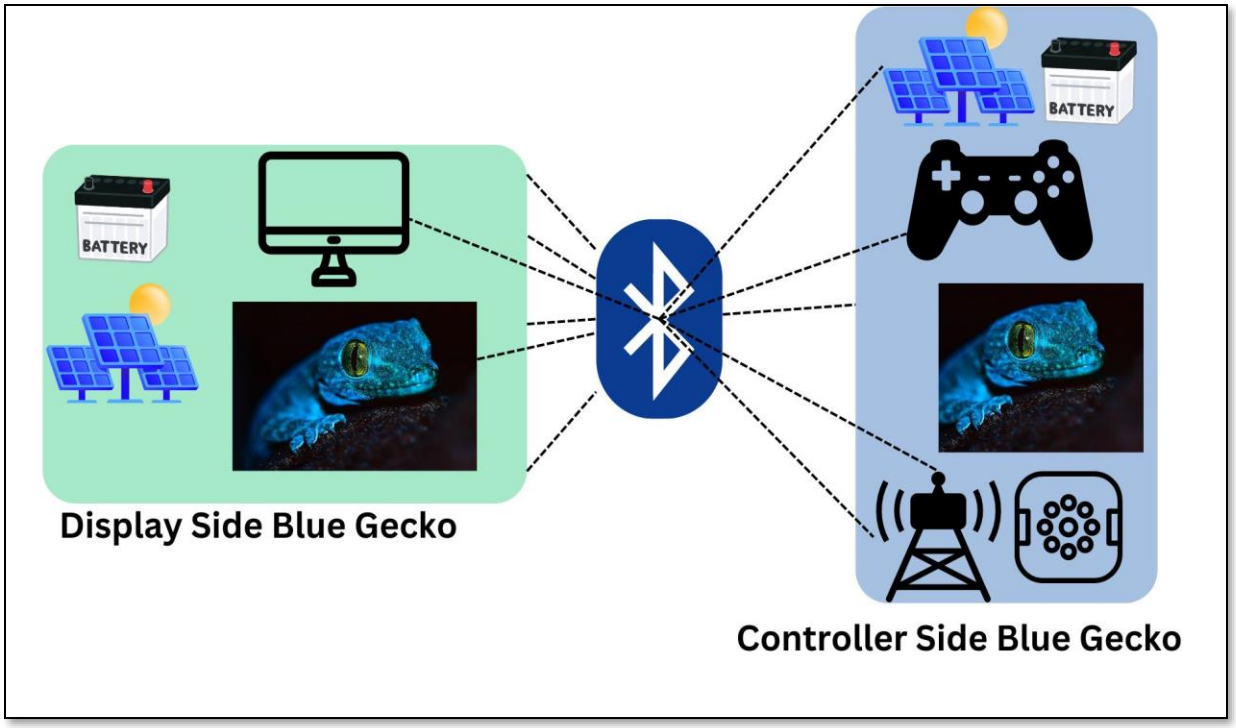


Figure 3 - Bluetooth Communication

KEY FEATURES & DESIGN LAYOUT

Low Power Consumption:

- Designed with energy-efficient components to maximize battery life, allowing for prolonged use without frequent recharging.
- Integrated power management ICs optimize power distribution across all components, ensuring minimal energy wastage.

Wireless Communication:

- Equipped with Bluetooth Low Energy (BLE) technology for seamless, low-latency communication between devices, ensuring responsive performance in gaming and other applications.
- Reliable BLE connectivity provides a stable link even in challenging environments, with a good range for remote operation.

Compact and Portable Design:

- The device is compact and lightweight, making it easy to carry and suitable for portable applications such as handheld gaming or wearable technology.
- Carefully designed PCB layout to minimize size without compromising performance, ideal for integration into small enclosures.

Precise Control Inputs:

- Includes an analog joystick for accurate directional control, ideal for gaming and remote-control applications.
- Features multiple buttons and an I2C gyro sensor, enabling precise motion sensing and user input for interactive applications.

Rich Display Capabilities:

- Integrated with an OLED display, providing clear and vibrant visuals for game interfaces or status information, even in low-light conditions.
- Supports dynamic content display with low power consumption, ensuring that the device remains functional for longer periods.

Advanced Sensor Integration:

- Equipped with high-precision sensors, including an ambient light sensor and gyro sensor, allowing for real-time data collection and monitoring.
- The sensors enable enhanced user interactions, such as motion-based controls and environmental brightness monitoring.

Energy Harvesting:

- Designed with support for energy harvesting technologies such as solar panels and piezoelectric elements, reducing dependency on traditional batteries and extending operational life.
- Efficient power conversion ensures that harvested energy is effectively stored and used, contributing to the device's sustainability.

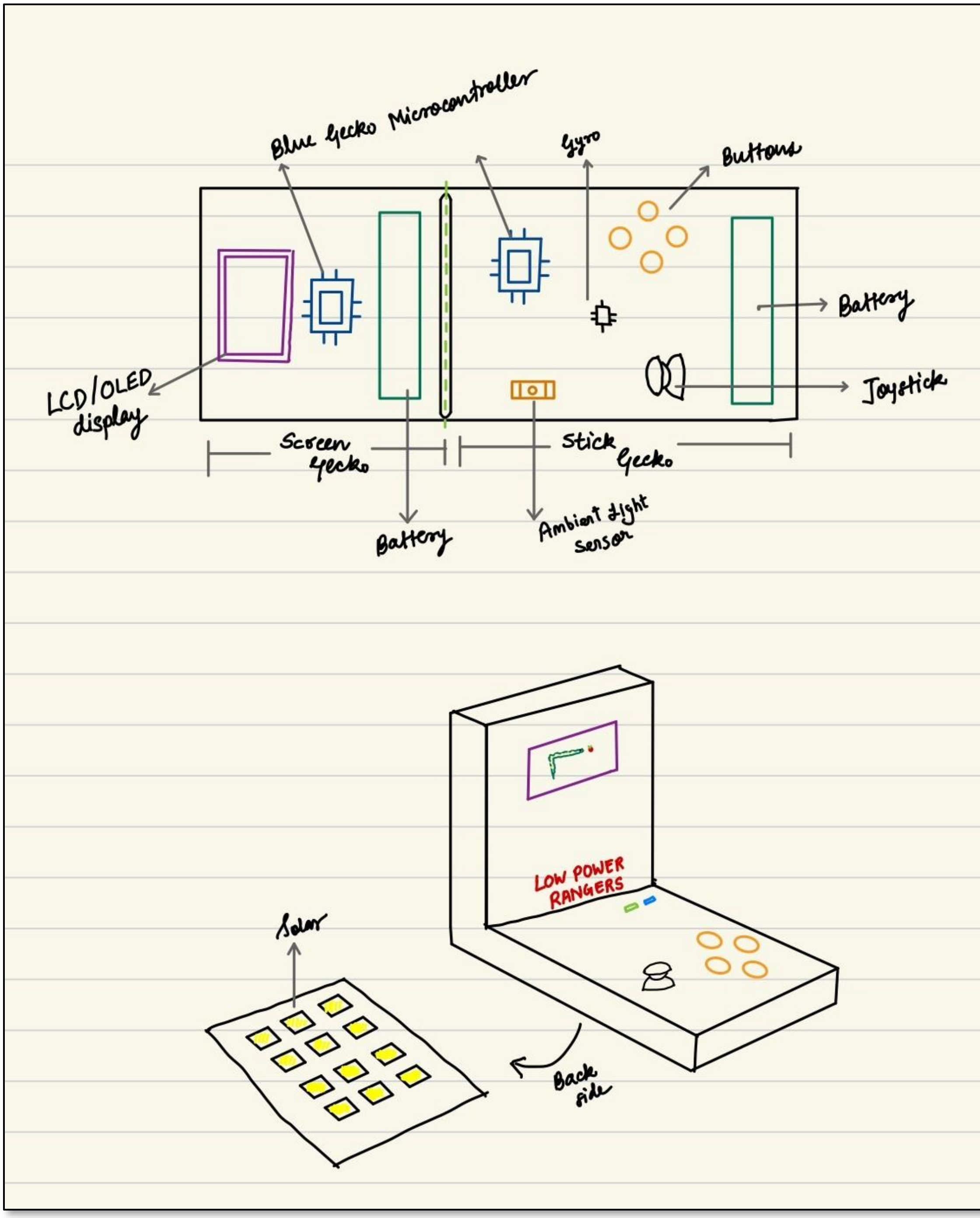


Figure 4 - Rough Sketch

EXTENDED IMPLEMENTATIONS & STRETCH GOALS

1) ***GPS Module for Device Safety***: Integrating a GPS module into your gaming device can significantly enhance its safety and functionality. By connecting a GPS module to your Blue Gecko microcontroller via UART or I2C, you can enable real-time location tracking. This integration allows for features such as geofencing, which alerts users if they move outside a designated area, and emergency location sharing, ensuring that the device's whereabouts are known in case of an incident. Developing firmware to handle GPS data, including latitude and longitude, will ensure reliable tracking and safety enhancements.

2) ***Multiplayer Capabilities***: Expanding your gaming device to support multiplayer functionality involves enabling local connectivity through BLE (Bluetooth Low Energy). By configuring multiple BLE receivers to communicate with each other, you can facilitate real-time multiplayer interactions. This requires implementing synchronization protocols to keep game states and player actions consistent across devices, ensuring a smooth multiplayer experience. Additionally, updating the OLED display and game logic to manage multiple players, including player identification and scoring, will enhance the overall gameplay experience.

3) ***Extended Connectivity with Cloud and Mobile App***: Enhancing your device's connectivity by integrating it with a cloud platform and a companion mobile app provides significant added value. By implementing cloud services, you can store and manage game data such as player statistics, high scores, and progress. Communication between the device and the cloud can be achieved using RESTful APIs or MQTT. Developing a companion mobile app allows users to interact with their gaming device remotely, adjust settings, and view game statistics. This integration not only extends the functionality of the device but also offers a synchronized experience across both the device and the mobile app.

COMPONENT SELECTION

Elements	Component Name	Part Name	DigiKey/Mouser Link
Microcontroller	Blue Gecko	EFR32BG13P732F512GM48-D	Link
Battery	Lith-Ion Battery	PRT-18286	Link
Energy Harvester	Solar Cell	KXOB25-12X1F-TB	Link
	Power Management Integrated Circuit	BQ25504RGTT	Link

Elements	Component Name	Part Name	DigiKey/Mouser Link
Sensors	Switch Joystick	COM-09032	Link
	Ambient Light Sensor	TEMT6000X01	Link
	Gyro Sensor	MPU-6050	Link

Elements	Component Name	Part Name	DigiKey/Mouser Link
Actuators	OLED Display	OLED 128x64 0.96" SPI	Link

PROJECT WEEK 1 UPDATE

STATUS REPORT

Addressing Proposal feedback

I] Power Management Block Clarification:

The Power Management Block in our design represents the PMIC. We do not plan to have line voltage going into our design. The AC-DC converter was initially added as a precautionary measure but has been removed based on its irrelevance to our current design. Apologies for any lack of clarity in the block diagram.

II] PMIC Selection (BQ25504 vs. BQ25570):

After evaluating both the BQ25504 and BQ25570, we chose the BQ25504 due to its lower quiescent current (IQ < 330nA), compared to the BQ25570 (IQ = 488nA). However, we are flexible and open to switching to BQ25570 if it is recommended based on reliability or previous use.

III] Radio Block on Screen Gecko:

A Radio block is indeed required on the Screen Gecko for BLE communication between the two boards. Initially, both Stick Gecko and Screen Gecko will be designed on the same PCB and later separated, as outlined in the rough sketch.

IV] Estimated Dimensions:

We will explore dimensions further. As per Prof. Bogatin's suggestion, a 3.9 mils x 3.9 mils PCB can be manufactured without extra cost. We aim for a rectangular PCB, likely 3.9 mils x ‘X’ mils, and will work on designing an ergonomic enclosure that suits the final PCB size.

V] Battery Life Estimation:

We are still calculating the device's runtime before requiring a recharge. Currently, our goal is to have a small battery that will allow approximately one hour of operation. This estimate will be refined as the project progresses.

Activity Accomplished

1. Completed the project proposal.
2. Completed major component selection (sensors/actuators).
3. Conducted power and current calculations for initial stages.
4. Started working on a GANTT chart for the project timeline.
5. Developed an energy model graph to visualize power consumption.

Planned Activities

1. Create a detailed sketch of the product, especially focusing on the Blue Gecko EFR32BG13.
2. Start initial firmware development to test peripherals (we require development kits for this).
3. Project Week 2 Update

Schedule & Obstacles

We are currently on schedule, as this is the first week. However, we face a couple of minor obstacles:

- Development Kits: We need development kits for firmware testing of different peripherals.
- Battery Level Measurement: We are looking for a solution to track battery percentage remaining. We haven't done much research yet, but we are curious if there's an IC for battery monitoring or a method to display the remaining battery percentage.

USE CASE MODEL & TIME SLOTS

The application will transition between several states:

1. **OFF State**: Both Stick Gecko and Screen Gecko are turned off.
2. **Connection State**: Both devices turn on and establish a connection.
3. **Menu Screen State**: Once the connection is established, the menu screen appears.
4. **Game ON State**: During gameplay, all components are active, and the microcontrollers remain in the EM0 Active state, consuming maximum power.
5. **Game Pause State**: The game is paused, and the system enters low-power mode.
6. **Game Abandon State**: If the device is idle for too long, it transitions into this low-power mode.
7. **Low Battery State**: If we implement battery percentage tracking, this state will turn everything off to conserve power.

Gantt Chart LINK here

PART SELECTION: Major components

Module	**Reason for Selection**
EFR32BG13P732F512GM48-D Blue Gecko	➢ Powered by a 32-bit ARM Cortex-M4 core with a maximum clock speed of **38.4 MHz**, offering a balance between performance and power efficiency. ➢ It has **512 kB Flash** and **64 kB RAM**, providing ample storage for application code and data. ➢ Excellent low-power performance, with **1.4 µA** in deep sleep mode and **63 µA/MHz** in active mode.
PRT-18286 Lithium Ion Battery	➢ It has a capacity of 125**0 mAh**, providing a balance between size and energy storage for small projects. ➢ This battery has a low self-discharge rate, meaning the PRT-18286 can retain charge for long periods when not in use, reducing the need for frequent recharges. ➢ The battery supports efficient recharging with minimal energy loss, maximizing energy use and lowering power wastage.
KXOB25-12X1F-TB Solar Cell	➢ It has an efficiency of up to **25%**, which is exceptional for its size, allowing for more power generation in low-light conditions. ➢ It performs well in low-light conditions, making it reliable even in indoor environments or shaded outdoor locations. ➢ Its low internal resistance ensures minimal power loss, making it highly efficient in converting sunlight into usable energy.

SENSOR/ACTUATOR SELECTION

Sensor	Reason for Selection	Lowest Energy Mode Supported
TEMT6000 Ambient Light Sensor	1. Compatible with **ADC** 2. In systems with adaptive brightness, the TEMT6000 reduces power consumption by enabling automatic dimming in low-light environments. 3. The TEMT6000 consumes very little power making it ideal for battery-powered and energy-sensitive applications.	EM2
COM-09032 Joystick	1. It provides analog outputs for X and Y axis movement, allowing direct connection to a microcontroller's **ADC**, reducing the need for additional components and saving power. 2. The COM-09032 joystick has low standby current, making it suitable for battery-powered applications that require minimal power consumption when idle. 3. With its simple potentiometer-based design, the joystick consumes power only during active use, which helps conserve energy in low-power systems.	EM2
MPU-6050 Gyroscope Sensor	1. Works on **Inter-Integrated Circuit (I2C)** Protocol 2. This sensor has a low-power sleep mode, allowing it to conserve energy when motion sensing is not required, ideal for battery-powered applications 3. It features an integrated Digital Motion Processor (DMP), reducing the processing load on the microcontroller, which helps save power in motion-tracking systems 4. It supports motion-activated wake-up, allowing the device to stay in low-power mode until significant movement is detected, further reducing power consumption.	EM1

Actuator	Reason for Selection	Lowest Energy Mode Supported
1.5inch OLED Display Module	1. Supports both **SPI** and **I2C** communication protocols, giving flexibility for interfacing with different microcontrollers and embedded systems. 2. Since OLED pixels generate their own light, no backlight is required, leading to lower power consumption, especially in dark-themed UIs or screens with minimal white elements. 3. The module typically includes a sleep or power-down mode, consuming minimal power when not actively displaying content	EM1/EM2

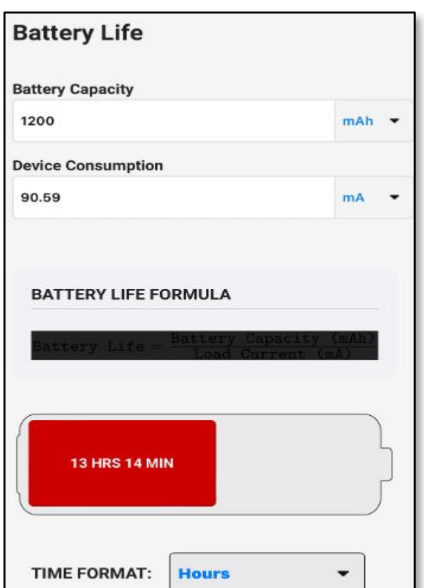
POWER & CURRENT CALCULATIONS

Component	Part Number	Operating Voltage	Current (Active Mode)	Power (Active)	Current (Sleep)	Power (Sleep)
Soc	EFR32BG13P732F512GM48-D	3.3V	11mA	36.3mW	1.9µA	6.27µW
Joystick	COM-09032	3.3V	25mA	1.65mW	-	-
Ambient Light Sensor	TEMT6000X01	3.3V (max 6V)	0.991mA [600uA Max Illuminace]	3.27mW	60µA	198µW
Gyro Sensor	MPU-6050	2.375V-3.46V	3.6mA	11.8mW	110µA	363µW
OLED Display	128x128 1.5inch OLED	3.3V	50mA (50% pixies on)	165mW	5mA (dISPLAY OFF)	16.5mA
			Total: 90.59mA	Total: 218.02 mW	Total: 5.0719mA	Total: 17.06727 mW
Power Related Components						
Solar Cell	KXOB25-12X1F-TB	5.58V	44mA	24.5mW	-	-
PMIC	BQ25504RGTT	3.3V	300mA	1.8mW	-	-
Battery	PRT-18286	3.7V	6µA		0.1µA	-

Sheet and References Link.

Power Consumption Table-

STATE:	Total Power Consumption (mW)
Connection State	53.36
Menu Screen State	201.86
GAME ON State	218.02
GAME PAUSE State	201.86
GAME Abandon State	165.56
Low Battery State	17.07



Digi-Key Battery Calculator

- Battery Capacity: 1200 mAh

- Device Consumption: 90.59 mA

Using the formula displayed:

Battery Life (hours) = Battery Capacity (mAh) / Load Current (mA)

The result shown is 13 hours and 14 minutes of battery life.

ENERGY MODEL GRAPH

TABLE	SoC/µC	Buttons	Joystick	Light	Gyro	Display
Connection state	EM0	✓	✗	✗	✗	✗
Menu Screen State	EM0	✓	✗	✗	✗	✓
Game ON* state	EM0	✓	✓	✓	✓	✓
Game Pause State	EM0	✓	✗	✗	✗	✓
Game Abandon State	EM2	✓	✗	✗	✗	✓
Low Battery State **	EM2	✗	✗	✗	✗	✗

* = We are planning to add a switch to control game input i.e., at a time only Button/Joystick/Gyro will be 'ON'

** = Depending on Lipo-fuel gauge IC incorporation

The table above outlines different operational states of the product, detailing which components are active during each state. These states represent different phases in the product's lifecycle, with specific components powered on or off depending on the state of operation. The columns represent the various hardware components (SoC/MCU, Buttons, Joystick, Light, Gyro, and Display), while the rows correspond to the operational states (Connection, Menu Screen, Game On, Game Pause, Game Abandon, Low Battery).

Connection State (EM0): During this state, only the SoC and buttons are active.

- This state corresponds to establishing communication between devices and initial setup.
- **Menu Screen State (EM0)**: Here, the buttons and the display are active, enabling user interaction with the menu.
- **Game On State (EM0)**: In this state, all components (buttons, joystick, gyro, and display) are active, as this is when the game is fully functional and operational.
- **Game Pause State (EM2)**: In the pause state, only the buttons and display are active, while other peripherals like the joystick and gyro are off, reducing power consumption.
- **Game Abandon State (EM2)**: During this state, most components are powered down except the display, and the power mode switches to EM2 (a deeper sleep mode for energy savings).

- **Low Battery State (EM2)**: In this state, none of the components are active, conserving as much energy as possible until the battery is recharged.

The table also includes two notes:

1. **Input Control Switch**: We plan to implement a switch that controls which inputs (Button/Joystick/Gyro) are active at any given time. This means that only one of these peripherals will be "on" to further reduce power usage.

Lipo Fuel Gauge IC Incorporation: The low-battery state may vary depending on whether the Lipo fuel gauge IC is incorporated. This gauge helps monitor battery levels and could trigger the Low Battery State.

POWER VS TIME GRAPH

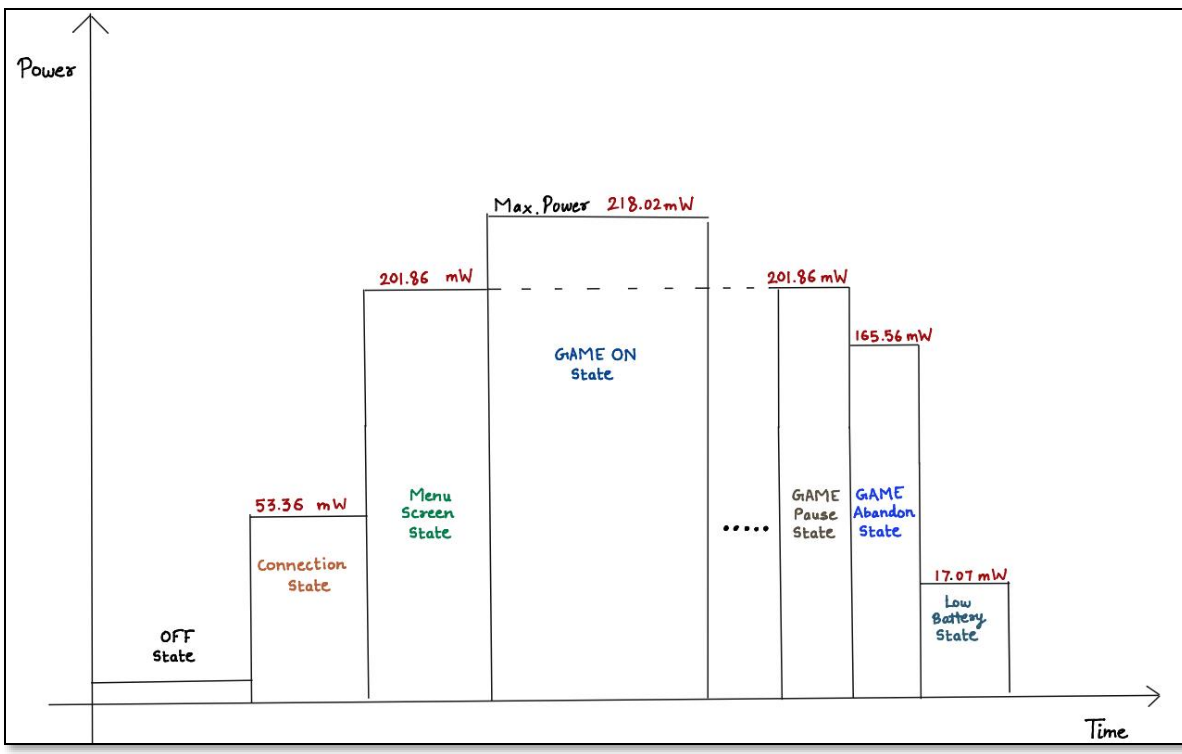


Figure 5 – Power Vs Time Graph

The above image is a power vs. time graph, where the power consumption for different states is displayed along the Y-axis, while the time spent in each state is displayed along the X-axis. Although the time spent in each state isn't fixed, the graph provides a general representation of how power consumption varies across states:

- **OFF State**: In this state, the product is completely turned off, consuming negligible power.
- **Connection State (53.36 mW)**: This represents the energy consumed during the initial connection phase, where only the SoC and buttons are active.
- **Menu Screen State (201.96 mW)**: Here, power consumption increases as more components are active (buttons and display).
- **Game On State (Max Power 218.02 mW)**: This state consumes the most power, as all components (SoC, buttons, joystick, gyro, and display) are in use.
- **Game Pause State (201.96 mW)**: Power consumption during pause is lower than in the Game On state but still considerable, as the display remains on while other components are turned off.
- **Game Abandon State (165.56 mW)**: When the game is abandoned, the device consumes less power as most peripherals are turned off.
- **Low Battery State (17.07 mW)**: In this state, the product conserves as much energy as possible, powering down nearly all components.

Explanation for Using Power vs. Time Instead of Energy vs. Time

In the report, we were asked to provide an energy vs. time graph for different states. However, the nature of our project makes it difficult to define fixed time durations for each state, as the user can move between states (e.g., Game On, Game Pause, or Game Abandon) at any moment. For instance:

- The **Game On state** could last for any duration depending on how long the player plays.
- The **Game Pause state** could be entered at any time, and its duration is user dependent.
- Similarly, the **Game Abandon state** could happen at any moment and last for varying periods.
- Due to these unpredictable transitions, we’ve created a power vs. time graph instead. The widths of each state in the graph represent tentative times the system could remain in each state. This graph helps visualize the system's power consumption in each state, even if the exact durations are unknown. By analysing this, the overall energy consumption can still be estimated based on average usage scenarios.

PROJECT WEEK 2 UPDATE

STATUS REPORT

Addressing Proposal feedback

Power Management IC (PMIC):
We have decided to go with the BQ25570 PMIC, instead of the BQ25504, after considering the feedback and evaluating the benefits. This change should help us better align with power management goals for the product.

Power vs. Time Analysis:
We appreciate the acceptance of our power vs. time analysis. As mentioned, we intend to break down the power consumption levels for different operating states like sleep modes, Bluetooth transmission/receiving, and sensor polling to enhance the granularity of our analysis.

Hand-Drawn Diagrams:
We acknowledge the feedback regarding the hand-drawn diagrams and will be updating all tables and graphs with professional tools in the next submission. The change can be seen here in the **TABLE.**

Fuel Gauge/ADC Battery Percentage Tracking:

After reviewing the options for battery percentage tracking, we have decided to drop the idea of incorporating a fuel gauge/coulomb counter.

Activity Accomplished

This week, we focused on finalizing and completing several critical aspects of the project:

- Completed the footprint and symbol for major components:
    - **EFR32BG13 (Blue Gecko)**
    - **OLED 1.5’ Display**
    - **Joystick (COM09032)**
    - **MPU6050 (Gyro/Accelerometer)**
    - **TEMT6000 (Ambient Light Sensor)**
    - **PMIC (BQ25570)**
    - **Solar Cell (KXOB24)**
- **Energy Storage Element Selection and Justification**:
  We selected a suitable energy storage element based on the expected power requirements and overall system design.
- **Finalization of PMIC and Energy Harvesting Method**:
  We finalized the BQ25570 PMIC and confirmed the solar-based energy harvesting method using the selected solar cell.
- **Estimation of Product Dimensions**:
  Estimated the overall dimensions of the product, ensuring that the chosen components fit within the form factor.

- **Estimation of Operating Temperature Range**:
  Defined the operating temperature range based on component datasheets and environmental factors.
- **Estimated Warranty Time**:
  Calculated the estimated warranty period based on the expected component lifetimes and product usage patterns.

Planned Activities

Last week we planned to start basic schematic design/ making a paper napkin sketch but haven't completed it. Also, we said we will start firmware but didn't start as we are yet to acquire DEV kits. This week we only did the things above mentioned in the activity accomplished section.

Schedule & Obstacles

Despite the challenges, we are generally on schedule, as we have completed the footprints and symbols for major components. However, we recognize that there is still a considerable amount of work remaining. One of the main obstacles we are currently facing is the lack of an internal ADC in the EFR32 microcontroller. This has complicated the process of interfacing an external ADC for battery voltage measurement.

Another question is about battery charging. How are going to achieve battery charging? Is a charging circuitry expected or the battery will be charged separately. After going through the PMIC BQ25570 datasheet we found that it has Battery Charging and Protection features.

Update of the project Gantt Chart

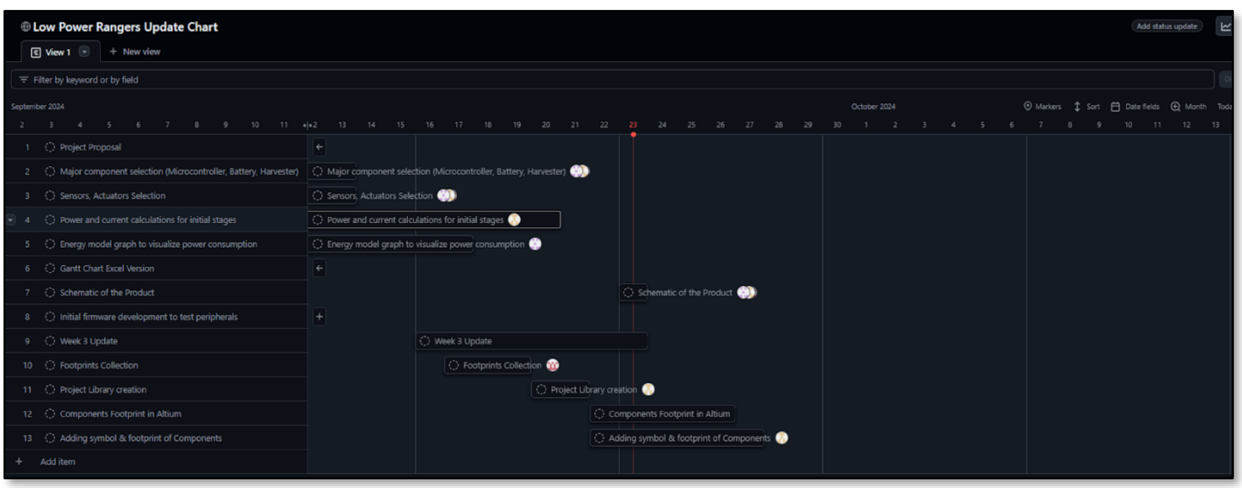


Figure 6 - Gantt Chart Overview

The following is the current progress as shown on the Gantt Chart. As stated above, the next task to accomplish is the Footprint & Symbol addition in the Altium library for the main components.

LINK: Gantt Chart

ENERGY STORAGE ELEMENT SELECTION

During our evaluation of the energy storage options, we considered two primary battery choices based on their voltage, capacity, and physical size constraints. Initially, we planned to use a 3.7 V Lithium-Ion Battery (PRT-18286), offering a capacity of 1.25Ah. However, concerns arose regarding the discharge curve and cutoff voltage (3V), particularly in conjunction with a buck converter regulating down to 3.3V. This raised concerns that the device might not maintain optimal operation as the battery voltage approaches its cutoff threshold.

Our alternative, a 7.2 V Lithium-Ion Battery (LI18650JP2S1P) with a capacity of 3.25Ah, provides a higher nominal voltage and substantially greater capacity. However, its larger size poses design challenges.

After a discussion we had with TA, he said the choice 1: PRT-18286 will suffice.

JUSTIFICATION of ENERGY STORAGE ELEMENT SELECTION

All our components operate at 3.3V, so energy storage solutions must consistently provide this voltage without risk of premature device shutdown. Both battery choices exceed the minimum operational voltage of 3.3V, but we had concerns about the operational efficiency and longevity when nearing the discharge threshold voltage value of the first choice.

Energy Storage Element selection math: Our max current consumption is 90.59mA (Use case peak current). Considering it to be 100mA for easier calculation and a buffer of value. Our goal operation time is 7 hours per day. So at least we need a battery of rating 700mAh. Assuming we discharge at rated 0.2C value, the max discharge current will be 140mA, which will suffice. Both choices are adequately rated in terms of capacity (1200mAh and 3250mAh respectively), comfortably above our minimum requirement of 700mAh considering our estimated current draw of up to 100mA. Thus, either battery can handle our estimated current of 100mA at a 0.2C rate.

USE CASE ENERGY CALCULATION

To determine the total energy requirement for a 7-hour operation period at a constant current draw of 100mA, we calculate the following:

Energy Required (Wh) = Voltage (V) × Current (A) × Time (hours)

For 3.3V and 100mA:
Energy Required = 3.3V × 0.1A × 7h = 2.31Wh

RECHARGE TIME and ENERGY REQUIRED

For Choice 1 (PRT-18286):

- **Charging Voltage:** 4.2V
- **Charging Current:** 240mA (0.2C rate)
- **Charging Time:** Approximately 5 hours to full charge (Refer Page 4 of Datasheet)

- Charging Time = Capacity (Ah) / Charging Current (A)

  = 1.25Ah / 0.24A = 5.2 hours ≈ 5 hours

- Energy Required (Wh) = Voltage (V) × Current (A) × Time (hours)
  Energy Required = 4.2V × 240mA × 4h = 4.032Wh

For Choice 2 (LI18650JP2S1P):

- **Charging Voltage:** 9.4V
- **Charging Current:** 650mA (0.2C rate)
- **Energy Capacity:** 3.25Ah

Charging Time = Capacity (Ah) / Charging Current (A) = 3.25Ah / 0.65A ≈ 5 hours

- Energy Required (Wh) = Voltage (V) × Current (A) × Time (hours)
  Energy Required = 9.4V × 650mA × 5h = 30.55Wh

SIZING STORAGE BASED on USAGE & CHARGE CYCLE

- **Choice 1 (PRT-18286):**
  This battery is rated for 400 cycles at 60% depth of discharge (DoD), meaning it can use 40% of its 1250mAh capacity (i.e., 500mAh per cycle). Given our estimated usage of 700mAh per day (7 hours of operation), we would need to charge the battery twice per day. Therefore, the estimated lifespan of this battery is approximately 200 days.

- **Choice 2 (LI18650JP2S1P):**
  This battery is rated for 300 cycles at 70% DoD, meaning it can use 30% of its 3250mAh capacity (i.e., 975mAh per cycle). Given the same usage of 700mAh per day, this battery would require only one charge per day. As a result, its estimated lifespan is about 300 days.

CONCLUSION on ENERGY STORAGE

- After discussions with the TA, we decided to go with **Choice 1: PRT-18286**. Considering all the calculations, this battery will work for our application, and its size is better suited for our product.

- **Why didn't we use a supercapacitor?**
  Our application does not require a burst of current but rather a continuous current during the GAME ON state. Thus, a battery is a more appropriate solution for our needs.

POWER MANAGEMENT UNIT (PMU) SELECTION AND JUSTIFICATION

Initially, we considered using the **BQ25504 PMIC** due to its lower quiescent current compared to the **BQ25570**. However, after further evaluation, we finalized the **BQ25570** as our choice. One of the primary reasons for this decision is that the BQ25570 is a more tried and tested component—previous teams, including our TAs, have successfully used it and possess substantial knowledge about it.

Another key advantage of using the BQ25570 is its integrated **Step-Down Regulated Output (Buck Converter)**, which will be used to step down the battery voltage from its nominal 3.7V to the required 3.3V. Additionally, the BQ25570 features **Maximum Power Point Tracking (MPPT)** and a **Low Power DC-DC Boost Charger**, both of which will be employed in our solar cell (KXOB25) for energy harvesting. The PMIC also provides **battery charging and protection features**, making it well-suited for our application.

POWER MANAGEMENT UNIT (PMU) BLOCK DIAGRAM

1. Controller GECKO Board:

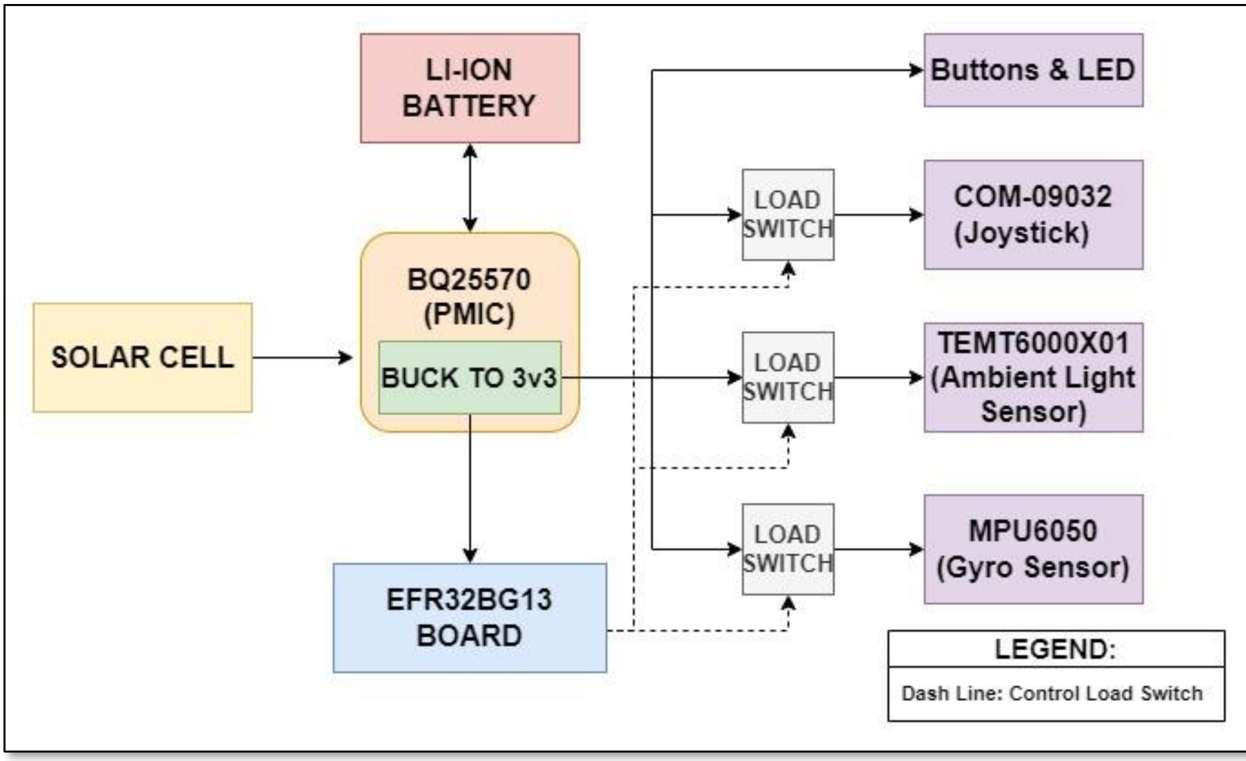


Figure 7 - Controller

2. Screen GECKO Board:

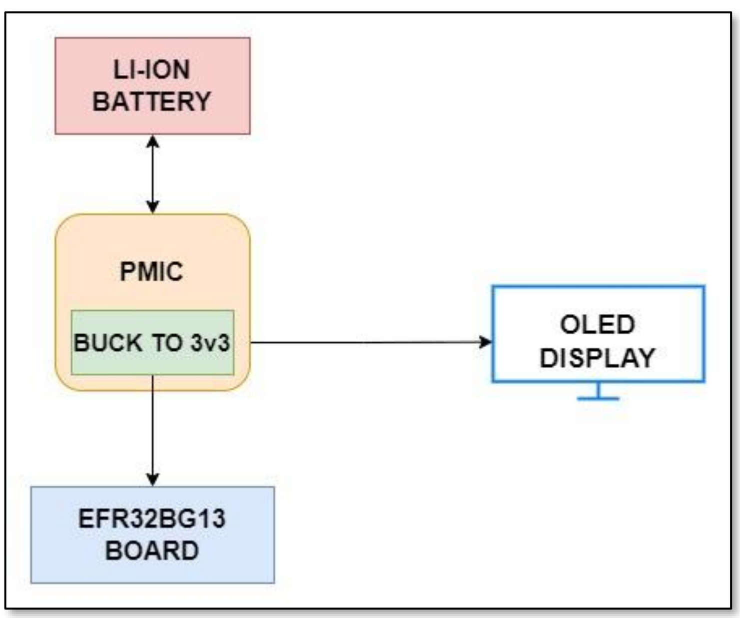


Figure 8 - Screen

CONSIDERATION OF AN UNREGULATED POWER SUPPLY

The digital portion of our board requires a fixed 3.3V supply. The battery we've selected operates with a maximum voltage of 4.2V and a cutoff voltage of 3V, making an unregulated power supply unsuitable. Components such as the gyro and ambient sensor function more reliably with a stable, fixed voltage. Therefore, we need a **buck converter** to regulate and step the voltage down to 3.3V, a feature that our selected PMIC, the BQ25570, conveniently provides.

CONCLUSION for PMIC

We have chosen to use the **BQ25570 PMIC**. Based on the above considerations, we are confident that it will meet all our product requirements.

GOAL DIMENSION OF THE PRODUCT

Controller-based Design:

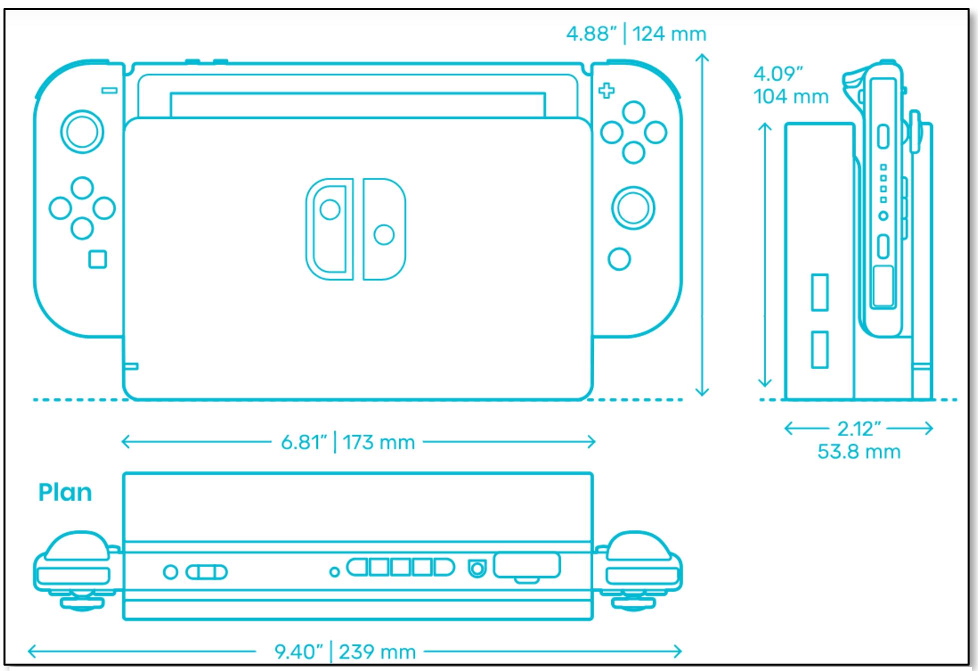


Figure 9 - Controller Design

- Dimensions: 6.81" x 4.88" x 2.12" (173 mm x 124 mm x 53.8 mm)
- Functionality is given priority in this design, which has a bigger footprint for docking mechanisms and hand controllers alike. With this magnitude, the goal is to lower the total volume while maintaining essential input functions like sensors, buttons and joystick controls.
- Actual dimensions will be smaller to enhance compaction.

Screen-based Design:

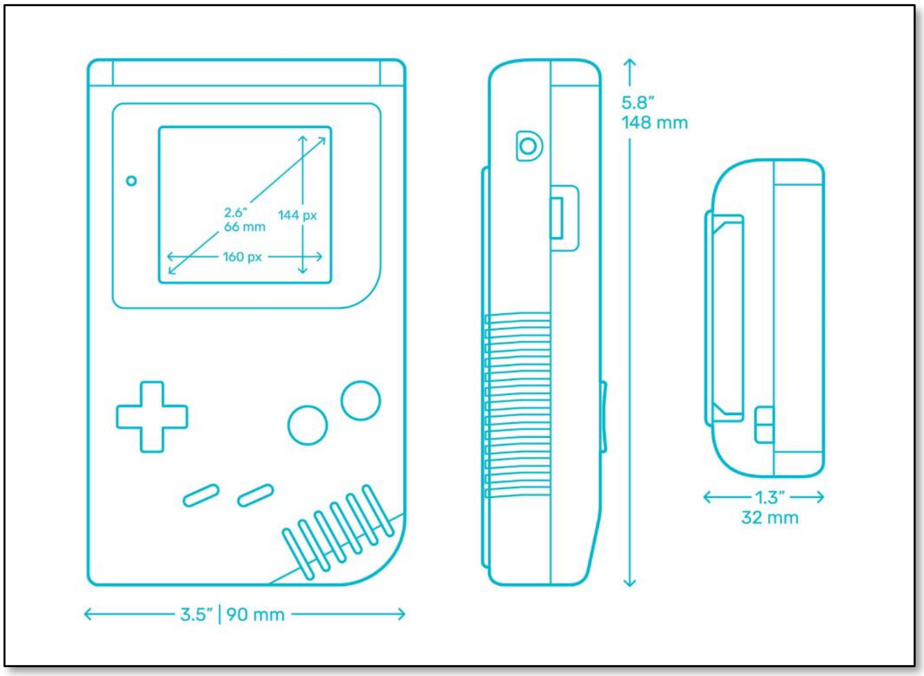


Figure 10 - Screen Design

- Dimensions: 3.5" x 5.8" x 1.3" (90 mm x 148 mm x 32 mm)
- This design is more compact because it places more emphasis on the interaction between the screen and buttons. Nonetheless, by improving the internal components—particularly by reducing their thickness (1.3")—further downsizing is achievable while preserving battery life and screen clarity.
- More emphasis on the OLED Display and portability.

We confirmed with the university's available 3D printers that their build platforms can accommodate a maximum size of 21mm x 23mm. Fortunately, our product's dimensions fall well within this limit.

WIRELESS RANGE

Compared to older BLE versions, the EFR32BG13's long-range mode with Bluetooth 5's LE Coded PHY allows for a greater range. Under perfect circumstances, Bluetooth 5's typical range can extend up to 200 meters when in line of sight.

Reference: [Paper Link](), [Community Link]()

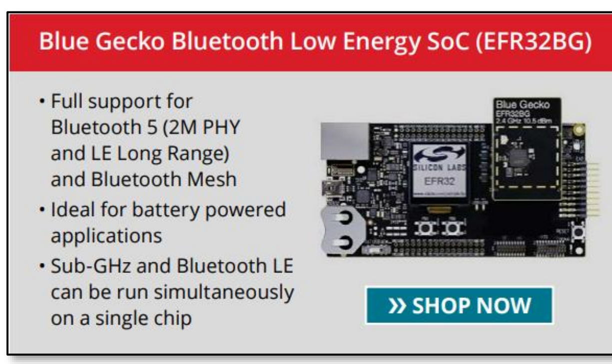


Figure 11 - Range

EXPECTED OPERATING TEMPERATURE & HUMIDITY RANGE

1. **SoC: EFR32BG13P732F512GM48-D**
   - **Operating Temperature**: -40°C to 85°C
   - **Relative Humidity**: Not specified - standard operating conditions
2. **Joystick: COM-09032**
   - **Operating Temperature**: -40°C to 100°C
   - **Relative Humidity**: < 60%
3. **Ambient Light Sensor: TEMT6000X01**
   - **Operating Temperature**: -40°C to 85°C
   - **Relative Humidity**: 60% to 90%
4. **Gyro Sensor: MPU-6050**
   - **Operating Temperature**: -40°C to 85°C
   - **Relative Humidity**: Not specified - standard operating conditions
5. **OLED Display: 128x128 1.5inch OLED**
   - **Operating Temperature**: -40°C to 85°C
   - **Relative Humidity**: Not specified in the datasheet

Power Related Components:

6. **Solar Cell: KXOB25-12X1F-TB**
   - **Operating Temperature**: -40°C to 90°C
   - **Relative Humidity**: Not specified in the datasheet
7. **PMIC: BQ25570**
   - **Operating Temperature**: -40°C to 125°C
   - **Relative Humidity**: Not specified in the datasheet
8. **Battery: PRT-18286**
   - **Operating Temperature**: Charge: 10°C to 45°C, Discharge: -20°C to 60°C
   - **Relative Humidity**: Not specified in the datasheet

Based on the above observations, we are estimating the temperature range of our product to be 10°C to 40° and the relative humidity to be around 60%.

Find the entire list of components in-detail: Spreadsheet Link

EXPECTED WARRANTY OF THE PRODUCT

- Based on the analysis of the battery life cycles provided in the usage and charge cycle section, the PRT-18286 battery, rated for 400 cycles at 60%, is expected to last approximately 200 days. This calculation assumes an estimated daily usage of 700mAh over a 7-hour period.

- Therefore, the anticipated warranty for the application, without replacing the battery, is 200 days. This estimate assumes that other components will not degrade within this time frame, as no specific warranty information was available in their respective datasheets.

PROJECT WEEK 3 UPDATE

STATUS REPORT

Addressing Week 2 Feedback

We have taken the following actions to incorporate the recommendations:

- **USB Input for BQ25570:** We have decided to incorporate the suggested USB input into the design. This will be used in conjunction with our solar cell to assist in jump-starting the battery charging circuitry.
- **Internal ADC Confirmation:** We appreciate the confirmation regarding the EFR32's internal ADC. We no longer need an external ADC for voltage readings as the internal 12-bit 1Msps SAR ADC suffices. We will ensure we use a voltage divider to bring the input voltage within the acceptable range.
- **Design Aesthetic:** The resemblance to existing consoles was intentional. We aimed to draw inspiration from popular designs like the Nintendo Switch and Gameboy. After further discussion, we have decided to finalize a form factor closer to the **Gameboy** for the final design.
- **Component Voltage Verification:** All our components have been verified, and 3.3V is the only voltage rail we will need for the system.

Activity Accomplished

Footprints Completion: Last week, most of the component footprints were completed. This week, we finished adding all the required components for our product. We are ready for the first **SA Review** and have uploaded the zip file of our integrated library.

Worst-Case Timing Analysis: We conducted a worst-case timing analysis for our communication bus to ensure reliable performance under different conditions.

High-Risk Development Mitigation Plan: We identified and analyzed the high-risk development areas of the project. A mitigation plan has been devised to minimize the impact of potential risks.

Planned Activities

Schematic Design: We plan to start working on the schematic design of the system.

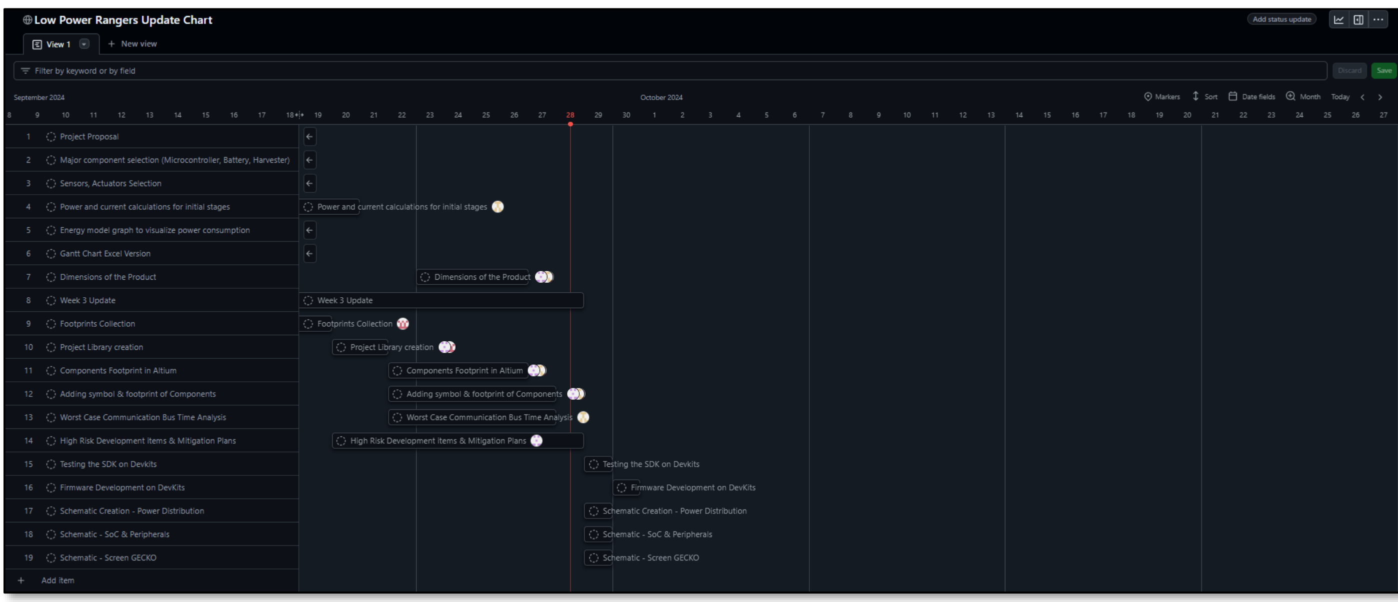
Firmware Development: With the development kits in hand, we will begin the firmware development. We installed **Simplicity Studio** and successfully ran the temperature example project to verify that the boards are working as expected.

Schedule & Obstacles

Schedule: We are currently on track with the project timeline and making steady progress.

Obstacles: No significant obstacles were encountered this week. In the current system design, the BQ25570 PMIC is primarily powered by a solar cell, which charges the Li-ion battery. As per your feedback, we plan to introduce a USB port to the VIN_DC of PMIC along with solar input. However, we still need to ensure that the USB input takes priority over the solar cell when both are available, a power switching mechanism needs to be introduced in the next week's update.

Update of the project Gantt Chart



Link: [Gantt Chart Link]

VERIFY C-RATE AND BATTERY SOLUTION

We performed a comprehensive battery analysis in the previous week, and the following details address the queries related to the C-rate, discharge characteristics, and integration with our Power Management Unit (PMU).

1. C-Rate of the Specified Battery

The C-rate measures how quickly a battery is discharged relative to its capacity. Our chosen battery has a **0.2C discharge rate**, meaning it can safely discharge 20% of its total capacity per hour. For our design, we opted for the **PRT-18286** battery (1200mAh) thus the discharge current of 240mA.

2. Peak Discharge Rate in Our Application

The peak current consumption for our application is **90.59mA**. For ease of calculation and a buffer, we rounded this up to **100mA**. This results in a peak discharge rate of approximately **0.084C**, which is well below the rated 0.2C, ensuring operational efficiency without stressing the battery.

3. Battery Discharge Curve and Cut-Off Voltage

Based on the lithium battery discharge curve for our PRT-18286:

- **Lowest Nominal Voltage:** The lowest nominal voltage during discharge is approximately **3.3V**.
- **Battery Cut-Off Voltage:** The cut-off voltage for our circuit is set to **3.0V**, ensuring that the battery isn't over-discharged, and that the lifespan is preserved.

4. Buck vs. Buck-Boost Solution

Given that the nominal voltage of the battery drops down to **3.3V** and our components operate at a steady **3.3V**, we evaluated whether we could use a **buck-only solution**.

- Since the voltage range of the battery is **3.0V to 4.2V**, a **buck converter** (step-down regulator) is sufficient. The PMU (BQ25570) has a built-in buck converter, which steps the voltage down from **3.7V nominal** to **3.3V**.
- A **buck-boost converter** is not required as the battery's operating voltage never falls below 3.0V during normal conditions. [We had a small doubt about what will happen when the battery goes below 3.3V till 3.0V, but after a discussion we had with TA, he said the battery will suffice.]

5. Maximum Discharge Rate of the Battery

For our PRT-18286 battery, the maximum discharge rate at **0.2C** is **240mA**. Given our peak current consumption of **100mA**, the battery will be operating well within safe limits, allowing it to power the system efficiently while maintaining a long cycle life.

In summary, the **PRT-18286** battery, coupled with the **BQ25570 PMU**, satisfies our product's power requirements. We have confirmed that the battery's **C-rate**, discharge rate, and cut-off voltage align with the design, and a **buck-only solution** will suffice. Further details on the battery and PMU integration will be provided in the next project update.

WORST CASE TIME ANALYSIS - COMMUNICATION BUS

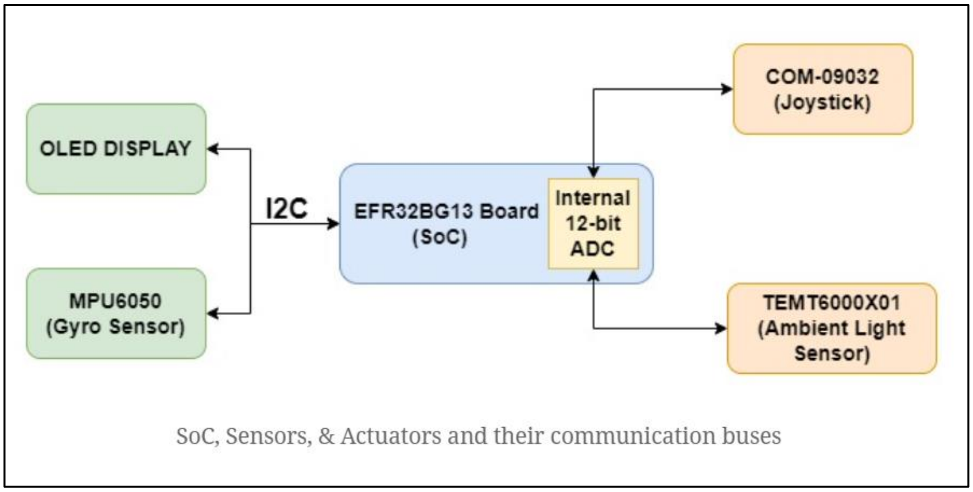


Figure 12 - Communication Bus

MPU-6050 - I2C Timing Specifications:

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

Parameters	Conditions	Min	Typical	Max	Units	Notes
I^2C TIMING	**I^2C FAST-MODE**					
f_{SCL}, SCL Clock Frequency				400	kHz	
$t_{HD.STA}$, (Repeated) START Condition Hold Time		0.6			µs	
t_{LOW}, SCL Low Period		1.3			µs	
t_{HIGH}, SCL High Period		0.6			µs	
$t_{SU.STA}$, Repeated START Condition Setup Time		0.6			µs	
$t_{HD.DAT}$, SDA Data Hold Time		0			µs	
$t_{SU.DAT}$, SDA Data Setup Time		100			ns	
t_r, SDA and SCL Rise Time	C_b bus cap. from 10 to 400pF	20+0.1C_b		300	ns	
t_f, SDA and SCL Fall Time	C_b bus cap. from 10 to 400pF	20+0.1C_b		300	ns	
$t_{SU.STO}$, STOP Condition Setup Time		0.6			µs	
t_{BUF}, Bus Free Time Between STOP and START Condition		1.3			µs	
C_b, Capacitive Load for each Bus Line			< 400		pF	
$t_{VD.DAT}$, Data Valid Time				0.9	µs	
$t_{VD.ACK}$, Data Valid Acknowledge Time				0.9	µs	

Note: Timing Characteristics apply to both Primary and Auxiliary I^2C Bus

Clock Frequency (fSCL): **400 kHz**

START (Repeated) Condition Hold Time (tHD;STA)l: 0.6 µs

- This is the minimum time the master must hold the START condition before the first clock pulse.

START (Repeated) Condition Setup Time (tSU;STA): **0.6 µs**

- Minimum time needed to set up a repeated START condition.

Data Hold Time (tHD;DAT):

- Max: 0 µs; Min: 100 ns
- The data hold time indicates the time the master must maintain the SDA line after the clock signal.

Data Setup Time (tSU;DAT): **100 ns**

- Time required for setting up data on the SDA line before the clock signal.

STOP Condition Setup Time (tSU;STO): **0.6 µs**

- Time needed to set up the STOP condition.

Timing values apply to both the **Primary and Auxiliary** I2C Bus.

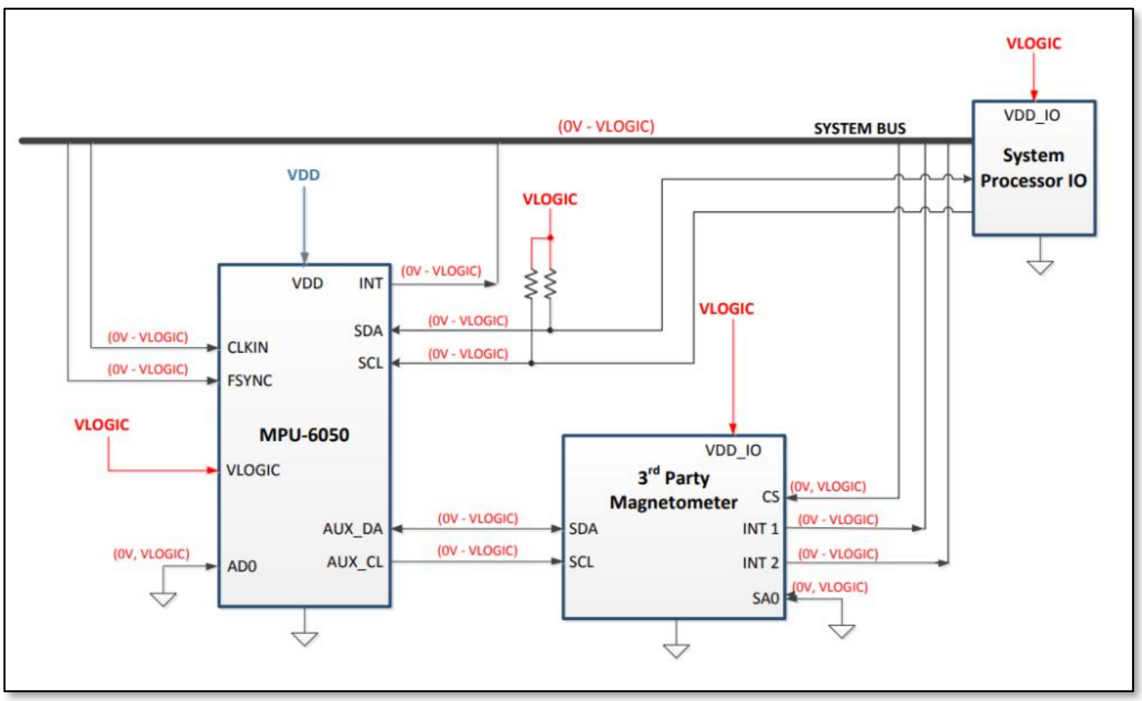


Figure 13 - Logic Levels and I/O

Source: **Gyro Datasheet Link** **- Page 18**

Steps involved in the I2C communication between the MPU-6050 and the SoC:

1. **Powering Components:** To ensure that all devices share a single logic level for communication, the MPU-6050, magnetometer, and SoC are powered by the same voltage rail (VLOGIC).
2. **Communication with MPU-6050:** The SoC addresses and sends a start condition to the MPU-6050 to initiate communication. It then uses the shared I2C bus to read or write data (gyroscope) to and from the device.
3. **Auxiliary I2C (for Magnetometer):** The MPU-6050 connects to the magnetometer via its auxiliary I2C lines. The MPU-6050 retrieves data from the magnetometer upon request from the SoC and transmits it back via the primary I2C channel.
4. **Data Flow:** The SoC starts the communication, gets information from the magnetometer or MPU-6050, and processes the sensor data for motion or orientation sensing applications.
5. **System-Level connection:** To ensure synchronized data sharing between the sensors and the SoC, all components rely on the I2C bus for direct connection.

OLED Display - I2C Timing Specifications:

(V_{DD} = 1.65 - 3.5V, TA = +25°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
f_{SCL}	SCL clock frequency	DC	-	400	kHz	
T_{LOW}	SCL clock Low pulse width	1.3	-	-	µs	
T_{HIGH}	SCL clock H pulse width	0.6	-	-	µs	
$T_{SU:DATA}$	data setup time	100	-	-	ns	
$T_{HD:DATA}$	data hold time	0	-	0.9	µs	
T_R	SCL、SDA rise time	20+0.1Cb	-	300	ns	
T_F	SCL、SDA fall time	20+0.1Cb	-	300	ns	
Cb	Capacity load on each bus line	-	-	400	pF	
$T_{SU:START}$	Setup timefor re-START	0.6	-	-	µs	
$T_{HD:START}$	START Hold time	0.6	-	-	µs	
$T_{SU:STOP}$	Setup time for STOP	0.6	-	-	µs	
T_{BUF}	Bus free times between STOP and START condition	1.3	-	-	µs	

This OLED Display uses **SSD11547**. I2C operates at the fast-mode (**400 kHz**) speed.

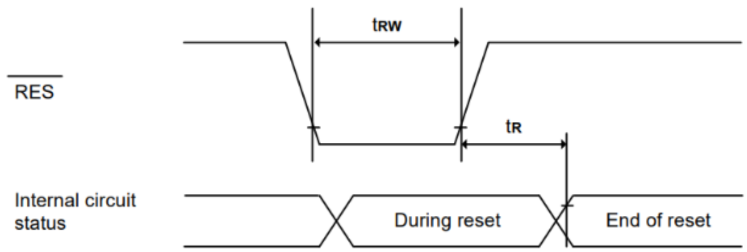
Clock Cycle Time (tCycle): The minimum clock cycle time is **2.5 µs**, dictating the overall speed of data transmission across the I2C bus.

Start Condition Hold Time (tHSTART) and Setup Time (tSUSTART):

- Critical for ensuring the stability of the start condition during data transmission.
- Both require a minimum time of **0.6 µs**.

Data Hold Time (tHD) and Data Setup Time (tSD): Data must be held steady for at least **100 ns** before being sent and can be held for a maximum of **0.9 µs** for valid communication.

(6) Reset Timing



(V_{DD} = 1.65 - 3.5V, T_A = +25°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Condition
t_R	Reset time	-	-	2.0	µs	
t_{RW}	Reset low pulse width	10.0	-	-	µs	

Reset Timing diagram shows the timing parameters for the reset signal:

- **RES Line**: This is the reset signal that stays low for a minimum of **10 µs** to ensure the system resets.
- **Internal Circuit Status**: During the time the **RES** line is low, the internal circuit remains in the reset state. The reset completes after **tR** (**2 µs**) once the reset signal goes high again.

The reset signal must be low for at least **10 µs** to trigger a reset, and the system will fully recover within **2 µs** after the signal goes high again.

Source: OLED Datasheet Link - Page 54, 55

Joystick and Ambient Light Sensor - ADC Sampling Specifications:

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
ADC clock frequency	f_{ADCCLK}		—	—	16	MHz
Throughput rate	$f_{ADCRATE}$		—	—	1	Msps
Conversion time[5]	$t_{ADCCONV}$	6 bit	—	7	—	cycles
		8 bit	—	9	—	cycles
		12 bit	—	13	—	cycles

Time Analysis with Internal 12-bit ADC, we know:

- ADC Clock: **16 MHz**
- Conversion Time: **13 clock cycles for a 12-bit resolution**
- Throughput Rate: **1 Msps (1 million samples per second)**

ADC Conversion Time:

- T_{conv} = Clock Cycles/ADC Clock Speed = 13/16MHz = 812.5ns
- Each 12-bit conversion takes **812.5 nanoseconds.**

Throughput Rate:

- The throughput rate is 1 million samples per second (1 Msps)
- ADC can process **one sample every 1 µs**.
- Conversion time is 812.5 ns, the system has approximately **187.5 ns** to handle overhead (switching channels or processing interrupts), before the next sample is taken to maintain the 1 Msps rate.

For two channels, the total time required for one cycle of sampling both signals is:

- T_{cycle} = 812.5ns × 2 =1.625µs
- Both signals can be sampled every **1.625 µs.**

Multiplexing and Interrupt Handling:

- The multiplexer switching time is negligible compared to the ADC conversion time, so we can approximate the total time per cycle as 1.625 µs.
- Assuming efficient interrupt service routines, the overhead for processing each ADC result could be in the range of **100-200 ns**.
- This would still fit comfortably within the available time between samples.

Total Timing Overview:

- Conversion Time per Channel: **812.5 ns (for 12-bit resolution).**
- Total Time for Two Channels: **1.625 µs.**
- Interrupt Processing and Switching Overhead: **~100-200 ns.**

The throughput of the ADC (1 Msps) is fast enough to allow sampling multiple sensors while maintaining a good refresh rate. Since each ADC conversion takes 812.5 ns, even when sampling two different sensors, it would still be under **1.625 µs for both**, which is extremely fast.

This allows the system to alternate between the joystick and the ambient light sensor without missing significant changes in sensor data.

Source: BG13 Datasheet - Page 92, 93

HIGH RISK DEVELOPMENT ITEMS

1. Integration of PMIC (Power Management Integrated Circuit)

- **Risk:** The integration of the BQ25570 PMIC is critical to ensure proper power regulation and energy harvesting from the solar cells. Given its complexity and multiple functionalities, improper configuration could lead to system failures, inefficient power management, or battery depletion.
- **Potential Impact:** System malfunction or inefficient energy harvesting, leading to shorter battery life and potential device shutdowns.

2. Communication Between the Two Blue Gecko Microcontrollers

- **Risk:** Our system relies on stable and efficient communication via BLE (Bluetooth Low Energy) between the two Blue Gecko microcontrollers. Issues in establishing or maintaining a reliable BLE connection could severely affect data transfer and real-time gaming responsiveness.
- **Potential Impact:** Delays or loss of data during gameplay, poor user experience, and potential failure to meet project deadlines due to extended debugging.

3. Joystick Footprint Alignment

- **Risk:** The joystick we selected does not come with a datasheet specifying the pad placement and dimensions. We downloaded its footprint from SnapEDA, but there is a risk that the footprint may not match the actual physical dimensions of the joystick legs.
- **Potential Impact:** If the footprint is incorrect, the joystick may not fit properly onto the PCB, leading to misalignment or connectivity issues.

MITIGATION PLANS

We have developed the following mitigation strategies to address the high-risk development items:

1. Integration of PMIC (Power Management Integrated Circuit)

- **Mitigation Plan:** We will carefully review the BQ25570 datasheet and application notes to ensure the correct configuration of the PMIC. Early testing of the power management system will be conducted on a development board to ensure functionality before integrating it into the final design. We will also consult with TAs who have experience with this component to verify that all settings are optimal.

2. Communication Between the Two Blue Gecko Microcontrollers

- **Mitigation Plan:** We will develop a robust BLE communication protocol and run extensive testing using both simulated and real-world conditions. This will involve creating test cases for varying signal strengths and distances between the two microcontrollers.

3. Joystick Footprint Alignment

Mitigation Plan: Since the joystick is a through-hole component, even if the pad placement does not perfectly match the legs of the joystick, we can manually adjust the positioning during assembly to ensure proper connectivity. Additionally, we will verify the physical dimensions as soon as the components arrive and adjust the PCB, if necessary, before final production. This will ensure minimal delays and prevent major redesigns.

UPDATED COMPONENTS

Component Name	Part No.	Specifications	Digikey/Mouser link
Load Switch	NLAST4599	Channel Select Input Over−Voltage Tolerant to 5.5 V Fast Switching and Propagation Speeds Low Power Dissipation: ICC = 2 A (Max) at TA = 25°C	Link
USB Mini-B Port			Link

PROJECT WEEK 4 UPDATE

STATUS REPORT

Addressing Week 3 Feedback

Hi team, if you are using a Schottky diode OR approach to your USB input to VIN_DC, whenever you plug your USB cable into your board, it will reverse bias the diode to your solar cell blocking it. This will ensure your USB always takes priority over your solar cell as long as your solar voltage is not higher than your V_USB (which it shouldn't since the max input for VIN_DC is 5.1V).

- We acknowledge that when the USB cable is plugged into the board, the Schottky diode connected to the solar cell input will be reverse-biased, ensuring the solar input is blocked. This guarantees that the USB input will always take priority over the solar cell input. The system's **maximum input for VIN_DC** is limited to **5.1V**, ensuring that the solar input voltage will never exceed the USB voltage in practical conditions. Therefore, this prioritization is reliable and robust in real-world scenarios.

Activity Accomplished

Component Footprints Update: Following a review of footprints and symbols as suggested by the SAs, several adjustments were made. We also added a significant number of passive components based on the requirements for each IC.

Schematic Completion: The design evolved from a basic napkin sketch into a full schematic library, including screen and controller sections. The overall schematic was divided into two key parts: PMIC and EFR32 SoC. Specific symbols for passive components were created for each section, ensuring clear functionality, power distribution, and GPIO connections.

Planned Activities

Firmware Development: The boards were tested for compatibility with different toolchains and GSDK versions. The next step involves initiating sensor testing for the I2C, ADC, and OLED interfaces. Additionally, testing for board-to-board BLE communication is planned to ensure readiness for the upcoming software demonstration. This will validate the complete system functionality and ensure all components are working properly before the demo.

PCB Placement & Design: After finalizing the schematic, we will proceed to the PCB design stage, focusing on optimal component placement, and board shape planning to ensure a well-organized, finished appearance. This step also includes designing the enclosure for an ergonomic shape suitable for the game controller and display.

Schedule & Obstacles

Obstacles:

Blue Gecko Devkit Bootloader Issue and Toolchain Compatibility

1. **Bootloader Issue:**
   The Blue Gecko Devkit entered bootloader mode unexpectedly and could not be reverted to debug mode. After updating the adapter firmware, the version information was missing. Upon investigation in forums, it was identified that the adapter firmware update issue could be resolved by manually installing the previous firmware version.

2. **Toolchain Compatibility:**

An additional issue occurred when trying to build previous projects:

- Error: Program 'make' not found in the path.
- Cause: The current setup was using SDK 3.2.9 instead of SDK 4.2.1, with GNU Toolchain 12.2.1, which is incompatible with the older projects that relied on GNU v10.2.1.

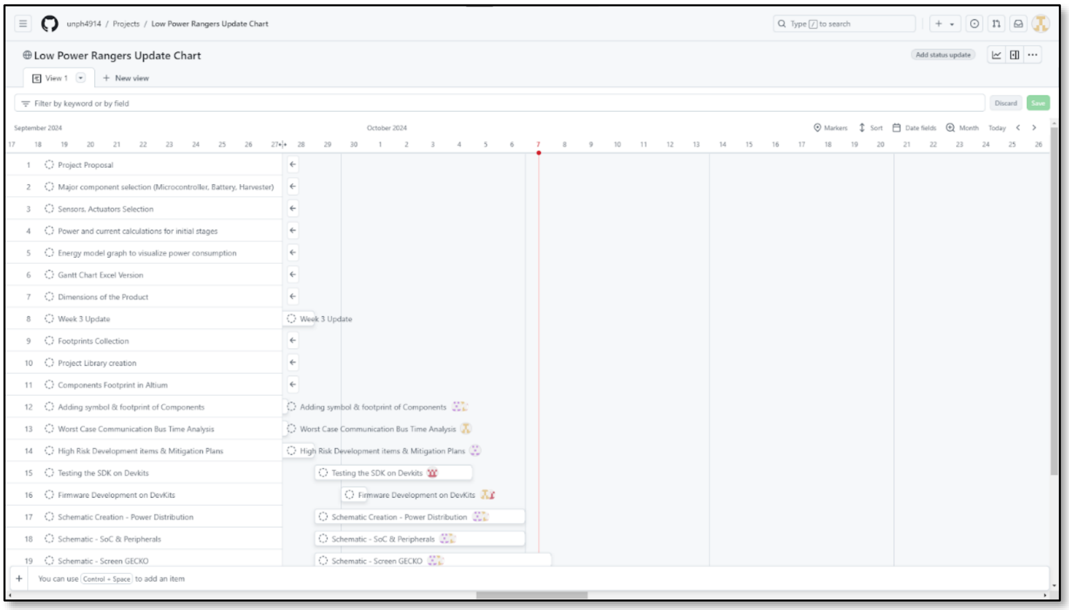
3. **Fix:**

- Reverted to GNU v10.2.1 and SDK 3.2.9 to ensure compatibility with existing projects.
- This setup will be used for flashing older codes and for further sensor testing and firmware development to ensure smooth project execution.

Dev Kit bootloader and SDK-Toolchain compatibility issues were encountered this week, now fixed. Schematic and symbol creation went smoothly with no significant obstacle.

Schedule: Despite the obstacles, we are currently on track with the project timeline and making steady progress.

Update of the project Gantt Chart



Link: [GanttChart Link]

USB-SOLAR INPUT TO PMIC

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted) (1)

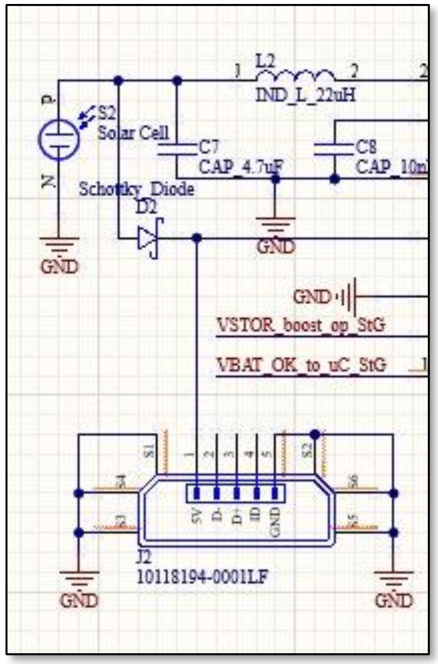
		MIN	MAX	UNIT
Input voltage	VIN_DC, VOC_SAMP, VREF_SAMP, VBAT_OV, VRDIV, OK_HYST, OK_PROG, VBAT_OK, VBAT, VSTOR, LBOOST, $\overline{\text{EN}}$, VOUT_EN, VOUT_SET, LBUCK, VOUT (2)	–0.3	5.5	V
Peak Input Power, PIN_PK			510	mW
Operating junction temperature, T_J		–40	125	°C
Storage temperature, T_{stg}		–65	150	°C

Source: [PMIC Link] | Page 5

WORKING:

When the USB cable is plugged in, the USB voltage (typically 5V) will be higher than the solar cell voltage, causing the Schottky diode connected to the solar cell to be reverse biased. This reverse biasing blocks the solar cell from powering the system, ensuring that the USB input takes priority.

The Schottky diode ensures that the solar cell is only used to power the system when the USB is not plugged in or when the USB voltage is lower than the solar voltage (which should not happen since the maximum VIN_DC input is around 5.1V). Further, The BQ25570 will ensure that the highest priority source (USB, when both connected) is used to charge the battery and supply power to the connected EFR32BG13.



FEATURES & COMPONENTS TO ENABLE PROGRAMMING OF MICROCONTROLLER

1. Mini Simplicity SWD Connector:

Essential for enabling in-system programming and debugging on custom PCBs. It provides a direct interface to the EFR32BG13 microcontroller through Serial Wire Debug (SWD) connections. This allows for flashing firmware, setting breakpoints, and stepping through code for debugging.

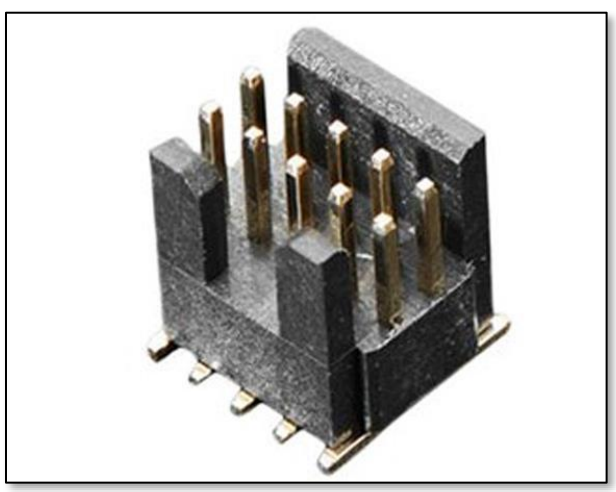
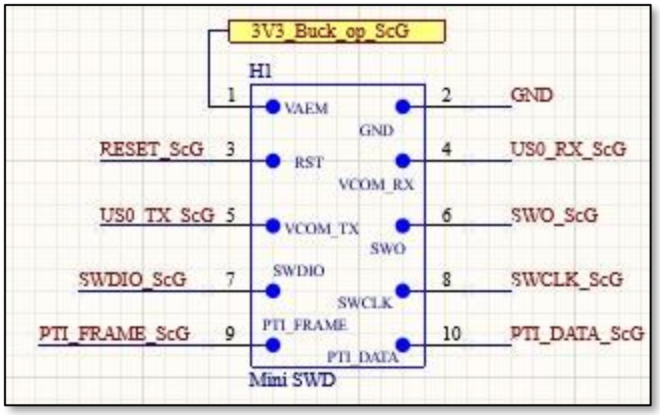


Figure 14 - MINI DEBUGGER



2. EFR32BG13 Dev Kit:

Serves as a comprehensive development platform for evaluating, prototyping, and debugging applications. It includes an onboard Simplicity Debugger, allowing flash firmware and advanced energy monitor (AEM) execution.

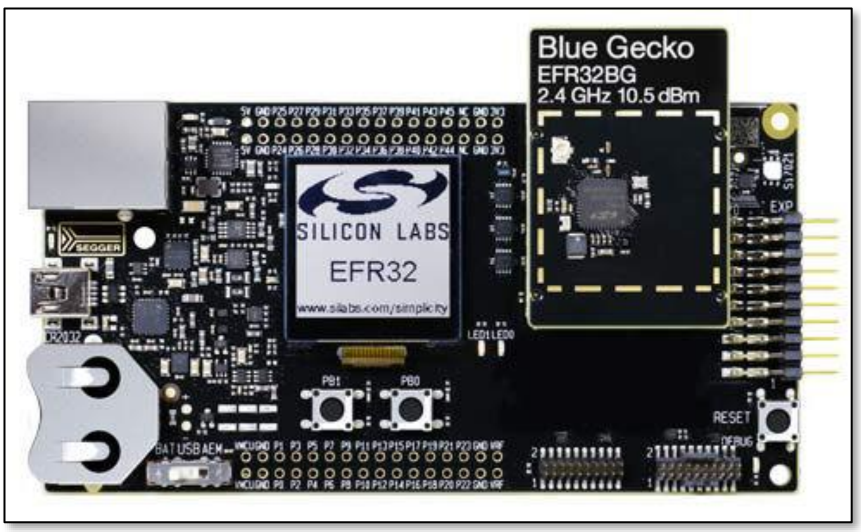


Figure 15 – BLUE GECKO

3. Simplicity Studio IDE:

It is an integrated development environment (IDE) designed specifically for Silicon Labs MCU. It integrates tools for code development, debugging, and wireless communication configuration.

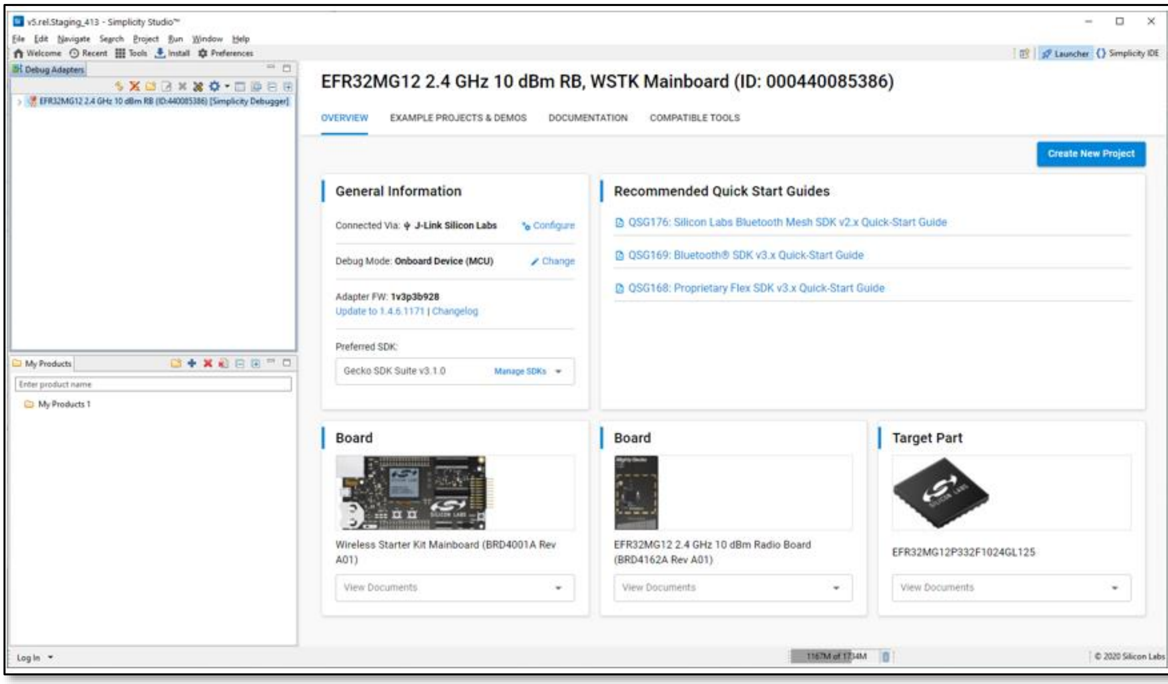


Figure 16 - IDE

PROCESS TO COMPILE, CONNECT, AND DOWNLOAD CODE

LPR Mini Debugger ⇥ Cable ⇥ EFR32BG13 Dev Kit ⇥ USB ⇥ Simplicity Studio IDE

1. **Install Board Support Packages**
   Launch Simplicity Studio and go to the **Install/Update** tab. Here, install the appropriate Wireless Starter Kit and SDK, which usually includes the Bluetooth SDK or the Simplicity Studio GSDK.
2. **Connect the EFR32BG13 Board to Your Computer**
   Use a micro-USB cable to connect the EFR32BG13 development kit to your computer. The board will receive power through the USB connection, and Simplicity Studio should automatically detect the development kit.
3. **Create a New Project**
   In Simplicity Studio, select **New Project**. From the list of connected devices, choose the EFR32BG13 Development Kit. You can opt for either a Bluetooth SoC – Empty Project or a relevant Bluetooth example project (such as SoC - Empty or Bluetooth SoC Thermometer). After configuring the project, click **Finish**.
4. **Configure the Project**
   If you created an empty project, you could begin coding in the provided `main.c` file or any other

necessary source files. This is the stage where you can configure the GATT or modify existing Bluetooth services as needed for your application.

5. **Build the Project**
   Click the Build button (represented by a hammer icon) or press **CTRL+B** to compile your code. Make sure that there are no errors during the build process.
6. **Flash the Code to the EFR32BG13**
   To upload the code to the EFR32BG13 board, click the Debug button (green bug icon) or press **CTRL+D**. Simplicity Studio will automatically download and flash the firmware onto your board.
7. **Run the Code**
   Once the flashing process is complete, the code should start running immediately. Utilize the Simplicity Studio's Debugger tools to monitor execution, set breakpoints, or step through the code if necessary.
8. **Monitor Output**
   To check for any serial output, open the Simplicity Studio Console or use a terminal application like PuTTY or Tera Term, if you have it configured. Ensure that the correct COM port (virtual serial port) is selected, and that the baud rate is set appropriately (usually 115200 baud for the EFR32BG13).
9. **Test the SoC Blinky Code**
   Load the SoC Blinky example in Simplicity Studio. Configure if needed, then build (CTRL+B) and flash (CTRL+D) the code to the EFR32BG13 board. The LED should blink, confirming the application is running correctly.
10. **Use SWD Debugger**: Connect the SWD debugger to flash code from the dev kit to the headers for Simplicity Mini-Debugger on the custom PCB. Once it is connected, the IDE will detect the SoC board, and the program can be flashed using the JLink Flash Programming Tool of the IDE.

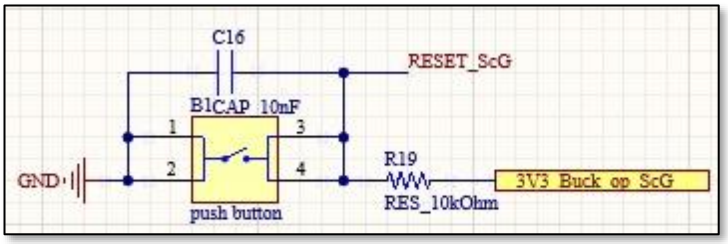
TEST POINTS

Following Test points will be required:

1. PMIC Output (VBAT_OK)
2. EFR32 Input (VMCU)
3. PMIC_VSTOR
4. Battery Output
5. I2C Connections - SDA, SCL
6. OLED Connections - CLK, DAT, VCC
7. ADC Pins
8. GPIO - IO Expander
9. Solar Cell Output
10. Reset Pin
11. GND Pins

RESET CIRCUIT DESCRIPTION

The EFR32BG13 features an external pin reset mechanism, allowing for manual or external control of system resets via the active-low RESETn pin.



Key Details

- **RESETn Pin Functionality:**
  - Pulling the RESETn pin low triggers a reset. Releasing it allows the system to restart.
  - This pin can connect to external hardware, like a push-button or controller, to initiate a reset.

- **How It Works:**
  - Asserting the RESETn pin (holding it low) halts all operations and resets internal registers to default.
  - Once released, the system undergoes a reset sequence similar to a Power-On Reset (POR).

- **Applications:**
  - **Manual Reset:** A physical button can pull the pin low to reset the microcontroller.
  - **External Controller Reset:** Other devices can control the reset line for fault recovery.

- **Glitch Protection:**
  - The RESETn pin includes protection against unintended short pulses, requiring a valid low signal for a reset.

- **Low-Power Systems:** Enables efficient resets without cycling power, saving energy during debugging or recovery.

Example Use Case

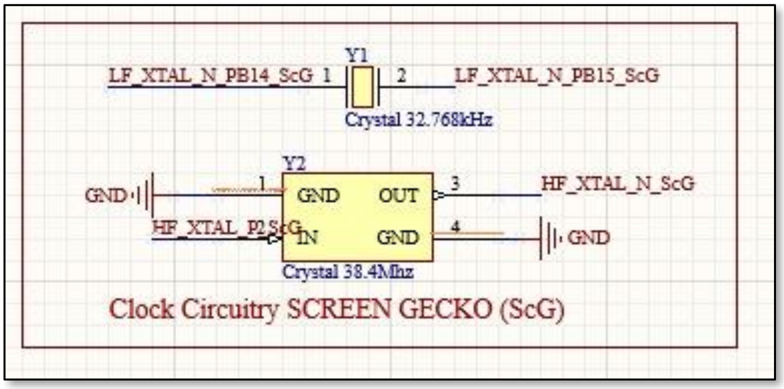
- **Reset Button:** Connect a button between the RESETn pin and ground. Pressing it pulls the pin low, triggering a reset; releasing it allows the device to restart.

Typical Configuration

- A pull-up resistor (10 kΩ) keeps the RESETn pin high during normal operation, with external devices able to pull it low to initiate a reset. This external pin reset mechanism is crucial for debugging, system recovery, and re-initialization without power cycling.

CLOCK GENERATION DESCRIPTION

The EFR32BG13 SoC from Silicon Labs features a configurable clock generation system designed for high-performance and low-power applications. Here's a concise overview:



3.5 Clocking

3.5.1 Clock Management Unit (CMU)

The Clock Management Unit controls oscillators and clocks in the EFR32BG13. Individual enabling and disabling of clocks to all peripherals is performed by the CMU. The CMU also controls enabling and configuration of the oscillators. A high degree of flexibility allows software to optimize energy consumption in any specific application by minimizing power dissipation in unused peripherals and oscillators.

3.5.2 Internal and External Oscillators

The EFR32BG13 supports two crystal oscillators and fully integrates five RC oscillators, listed below.

- A high frequency crystal oscillator (HFXO) with integrated load capacitors, tunable in small steps, provides a precise timing reference for the MCU. Crystal frequencies in the range from 38 to 40 MHz are supported. An external clock source such as a TCXO can also be applied to the HFXO input for improved accuracy over temperature.
- A 32.768 kHz crystal oscillator (LFXO) provides an accurate timing reference for low energy modes.
- An integrated high frequency RC oscillator (HFRCO) is available for the MCU system, when crystal accuracy is not required. The HFRCO employs fast startup at minimal energy consumption combined with a wide frequency range.
- An integrated auxilliary high frequency RC oscillator (AUXHFRCO) is available for timing the general-purpose ADC and the Serial Wire Viewer port with a wide frequency range.
- An integrated low frequency precision 32.768 kHz RC oscillator (PLFRCO) can be used as a timing reference in low energy modes, with 500 ppm accuracy.
- An integrated low frequency 32.768 kHz RC oscillator (LFRCO) can be used as a timing reference in low energy modes, when crystal accuracy is not required.
- An integrated ultra-low frequency 1 kHz RC oscillator (ULFRCO) is available to provide a timing reference at the lowest energy consumption in low energy modes.

Source: EFR32BG13 Link |Page 12

Clock Sources

- **High-Frequency Oscillators:**
  - **HFXO:** External crystal oscillator (typically 38.4 MHz) for precise timing.
  - **HFRCO:** Internal RC oscillator with multiple frequency options (1 MHz to 38 MHz) for lower power.
- **Low-Frequency Oscillators:**
  - **LFXO:** External crystal oscillator (32.768 kHz) for real-time clock.

    - **LFRCO:** Internal RC oscillator (32.768 kHz) for low-power needs.
    - **ULFRCO:** Very low power (1 kHz) for deep sleep modes.

Clock Management

- **Clock Trees and Dividers:** Flexible routing and frequency adjustments for CPU and peripherals.
- **Clock Control Unit (CMU):** Manages clock selection, gating for power savings, and division for lower frequencies.

Clock Domains

- **High-Frequency Domain:** For high-speed tasks (e.g., Bluetooth).
- **Low-Frequency Domain:** For low-power tasks (e.g., RTC).

Energy Modes

- **EM0 (Active):** All clocks active.
- **EM1 (Sleep):** High-frequency peripherals active.
- **EM2 (Deep Sleep):** High-frequency clocks off; low-frequency peripherals active.
- **EM3 (Stop):** Only ULFRCO active for essential functions.
- **EM4 (Shutoff):** Most systems powered down; clocks disabled.

This system ensures efficient power management while meeting varying performance requirements.

ALTERNATIVE ENERGY SOURCE FOR THE ENERGY HARVESTER

Using USB as an alternative power source (only V_USB & GND) offers a practical and accessible way to supply energy to the PMIC during development and testing phases. This approach is especially useful when the primary energy harvester isn't operational. It enables us to troubleshoot and validate the PMU circuitry before fully relying on harvested energy for the final implementation.

JUMP START METHOD TO CHARGE ENERGY STORAGE ELEMENT

There are two main ways to jump-start charging of the energy storage element (battery):

1. **USB Power Method** with only the **Vcc (power)** and **GND** pins connected. This provides power directly to the PMIC, which will initiate the charging process. The battery will begin charging automatically, provided the input voltage meets the charging requirements of the PMIC.

2. **External Power Supply Method**: Connect a regulated external power supply to the battery's terminals using a header. Given that the supply matches the required battery charging voltage (4.2V). This allows it to directly power the PMIC and charge the battery.

MAXIMUM CHARGING CURRENT ALLOWED BY THE ENERGY STORAGE UNIT SPECS

Battery Model: KPL623450 LiPo Battery (1300 mAh, 3.7V nominal)

C-rating for Charging: The datasheet lists the recommended **charging current** as 0.5C and **maximum charging current** as 1C.

1. **Recommended Charging Current: 0.5C**
   I_{charge}=0.5 × Battery Capacity =0.5×1300 mAh=650 mA
2. **Maximum Charging Current: 1C**
   I_{max_charge}=1 × Battery Capacity =1 × 1300 mAh = 1300 mA (1.3 A)

Thus, the maximum charging current allowed for this battery is **1.3 A** (at 1C).

What will the maximum current of the jump start power source be set to?

- To calculate the appropriate power supply settings for charging the battery based on the calculated current:

Voltage Setting: The datasheet specifies the nominal voltage of the battery as **3.7V**, with the charging voltage ranging from **4.2V (full charge)** to a **cut-off voltage of 3.0V**. Therefore, the **power supply should be set to 4.2V** to safely charge the battery without overvoltage.

Current Setting: Based on the C rating calculated above, we can set the maximum current of the jump start power source to a value that is less than or equal to 1.3 A to ensure safe and controlled charging of the battery

Where will the jump start power and ground signals connect to?

The jump start power (4.2V) and ground signals will be connected via **header pins** on the circuit board. These header pins will interface the external power supply to the battery management circuit, allowing the system to receive charging current through standard connectors.

- **Power pin**: The positive lead from the power supply (set at 4.2V) will connect to the designated **VBAT**+ header pin.
- **Ground pin**: The ground lead from the power supply will connect to the **GND** header pin.

Using header pins allows easy attachment and detachment of the external power supply, which is especially useful for testing or replacing power sources.

ENERGY / CURRENT TO PROGRAM THE FLASH OF THE MCU

Table 4.44. Flash Memory Characteristics[1]

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Flash erase cycles before failure	EC_{FLASH}		10000	—	—	cycles
Flash data retention	RET_{FLASH}	T ≤ 85 °C	10	—	—	years
		T ≤ 125 °C	10	—	—	years
Word (32-bit) programming time	t_{W_PROG}	Burst write, 128 words, average time per word	20	26.3	30	µs
		Single word	62	68.9	80	µs
Page erase time[2]	t_{PERASE}		20	29.5	40	ms
Mass erase time[3]	t_{MERASE}		20	30	40	ms
Device erase time[4 5]	t_{DERASE}	T ≤ 85 °C	—	56.2	70	ms
		T ≤ 125 °C	—	56.2	75	ms
Erase current[6]	I_{ERASE}	Page Erase	—	—	2.0	mA
Write current[6]	I_{WRITE}		—	—	3.5	mA
Supply voltage during flash erase and write	V_{FLASH}		1.62	—	3.6	V

Source: EFR32BG13 Link | Page 88

The EFR32BG13 features onboard flash memory for storing program code. During the programming process, erasing a page of flash requires up to **2 mA** of current, while writing to the flash demands a maximum of **3.5mA.**

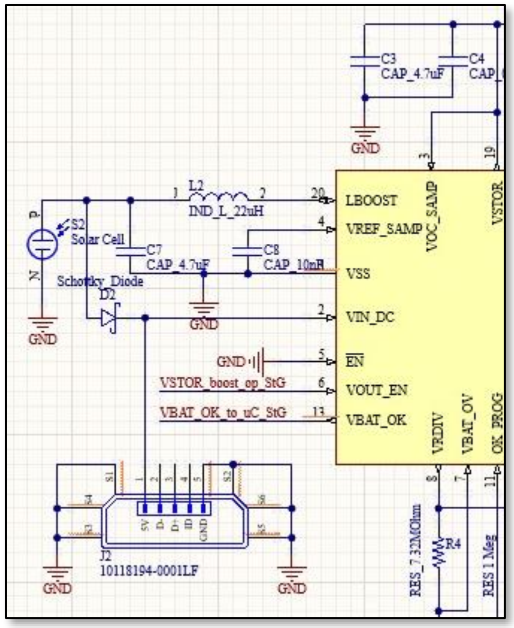
In addition to the boost charging front end, the bq25570 provides the system with an externally programmable regulated supply via the buck converter. The regulated output has been optimized to provide high efficiency across low output currents (< 10 µA) to high currents (~110 mA).

Source: PMIC Link | Page 13

Given that the PMIC **BQ25570** can supply a peak current of **110 mA**, the energy storage element is more than capable of supporting flash operations. According to the battery datasheet, the KPL623450 (3.7V, 1300mAh) can provide **1.3 A peak current**, making both the battery and PMIC sufficient for powering the flash operations without any bottleneck, as the required current is approximately 100 times lower than what the battery and PMIC can supply.

This ensures stable flash programming during development.

What are the connection points to enable external power to digital / MCU portion of the board



The EFR32BG13 microcontroller is powered through three distinct supply sections: analog, digital, and I/O. The digital supply is also the input for the internal buck regulator, which is connected to the main supply voltage. The input supply, VREGVDD, ranges from 1.8V to 3.8V, but this range is constrained by factors such as transient current load, operating temperature, and average current load over the device's lifetime.

The main supply voltage, VMCU, and a decoupling capacitor, CVDD (typically 10uF), are the two primary components necessary to power the digital section of the MCU. This also provides input to the internal buck regulator, which in turn powers other core functionalities of the board.

UPDATED COMPONENTS

Component Name	**Part No.**	**Specifications**	**Digikey Link**
Load Switch	TPS22919	Power Switch/Driver 1:1 N-Channel Input operating voltage range (VIN): 1.6 V to 5.5 V • Maximum continuous current (IMAX): 1.5 A	Link
Schottky Diode	SD103CWS-7-F	20 V 350mA - Low-Forward Voltage Drop • Guard Ring Construction for Transient Protection • Negligible Reverse-Recovery Time • Low Reverse Capacitance	Link

PROJECT WEEK 5 UPDATE

STATUS REPORT

Addressing Week 4 Feedback

We have not yet received any feedback for the previous update, so this section will be updated in the next week's project update.

Activity Accomplished

Schematic Completion: After a thorough review by the TAs, we have implemented their feedback, which included updating the designator names, adding a TVS diode for ESD protection, and incorporating indicator LEDs at the PMIC output. The schematic is now finalized and failure-proof. We are ready to proceed to the next step, focusing on the board shape and component placement.

Firmware Development: Progressing steadily with testing two out of four external hardware components on the Dev Kit. We aim to complete all testing in time for the software demo.

Planned Activities

Firmware Development: Our goal is to finalize firmware testing for the remaining sensors and actuators before the software demo. We have successfully tested the ambient light sensor and joystick. If time permits, we also plan to test BLE communication between the two Blue Gecko Dev Kits. All testing is targeted for completion by Monday.

PCB Placement & Design: Implement the changes as reviewed by the TAs. With the basic PCB dimensions established, we will move on to designing an ergonomic enclosure for the Stick GECKO (Controller pad). Multiple versions will be explored before finalizing the design. A similar design process will be applied to the Screen GECKO (Display Interface).

Routing: We will begin the initial routing process once the board design and component placements are finalized. The routing will be reviewed multiple times with the TA and professor to ensure there is no room for doubt or misinterpretation, repeating the process until we are completely confident in the design.

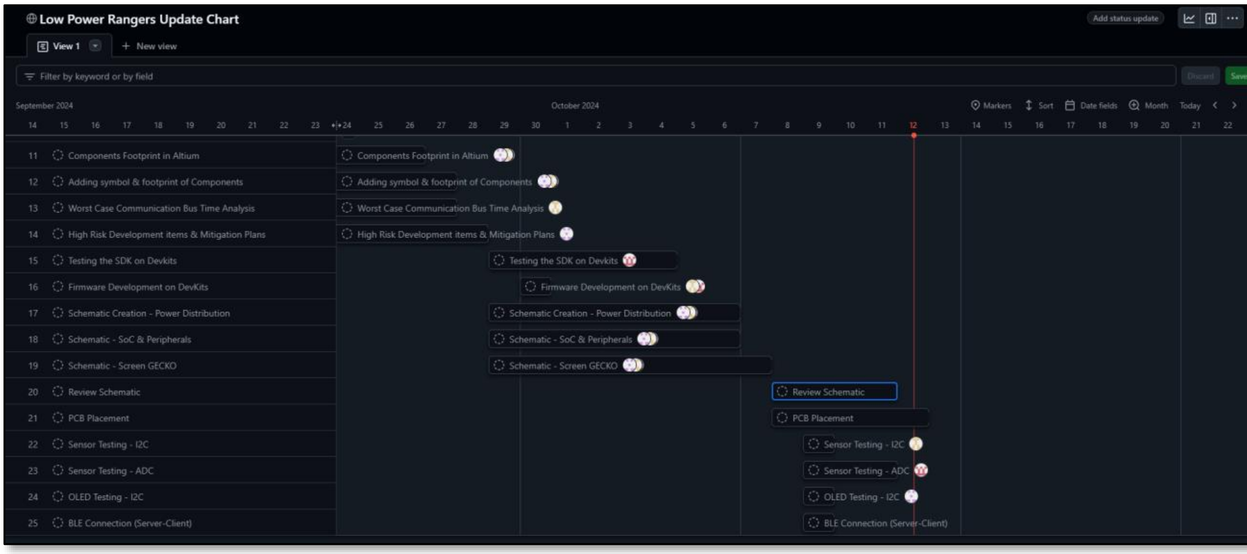
Schedule & Obstacles

1. MPU6050 Chip Obsolete

We looked through several sources, including DigiKey and Mouser, and discovered that the MPU6050 IC is no longer in production, along with comparable substitutes. Initially, we didn't expect stock problems because we thought the MPU6050 was widely accessible. Therefore, rather than directly integrating the IC, we have chosen to use the MPU6050 breakout board and connect it externally to the board.

Other than this, we are currently on track with the project timeline and making steady progress.

Update of the project Gantt Chart



Link: [GanttChart Link]

BULK CAPACITANCE - REQUIRED OR NOT?

In the simulation which we conducted, we used the LTC3106 as the PMIC, while our actual design will use the BQ25570, which has a higher VOUT range. Despite some differences, the simulation confirms that the BQ25570 will work effectively with our multiple ICs operating at 3.3V. The voltage stabilized at 3.3V during the simulation, demonstrating that the system is capable of maintaining a consistent power supply without additional capacitance or changes to the input voltage (VIN).

Moreover, the bulk capacitance values simulated were more than sufficient to meet the requirements of our actual PMIC, eliminating the need to rerun the simulation or make further adjustments. The BQ25570's robust output regulation further ensures stable operation across varying loads. This reinforces that the design, including the existing capacitive network, can effectively handle power fluctuations and noise, maintaining system stability without introducing additional components.

In summary, our simulation confirms that the BQ25570 will perform well in our design with the current configuration, and **there is NO need for any Bulk capacitance** for any voltage input adjustments or stabilization.

I/O PORTS IN THE DESIGN

1. Power: USB & Solar Cells
2. Buttons: RESET & GPIO (x4)
3. Indicator: Two LEDs at the output of PMIC; one for VBAT & one for VOUT_3v3.
4. IO Devices:
    - MPU6050: 3-axis accelerometer using I2C.
    - OLED 1.5’: 128x128 pixel blue/yellow display using I2C.
    - TEMT6000: Ambient light sensor using 12-bit ADC sampling.
    - Joystick: Sampled over the dual ADC channels.
    - Solar Cells: Energy harvester
    - Push Buttons: Reset circuit and User interface GPIO
    - Mini Simplicity Connector: Flashing and debugging code via Dev kit.
    - LEDs: Power On, VBAT, VOUT_3v3 and user LED.
    - BQ25570 PMIC
    - Li-Ion Battery

ESD PROTECTION AT I/O PORTS

We’ve chosen to install TVS diodes near our push buttons (W, A, S, D controls, and the reset buttons on both the boards) to protect the circuit from electrostatic discharge (ESD) and other voltage transients. Here are the key reasons for this decision:

Rationale for Using TVS Diodes

- **Protection Against ESD**:
  The buttons are regularly touched, making them vulnerable to ESD. TVS diodes help mitigate this risk by clamping high transient voltages that could damage sensitive components.
- **Guarding Sensitive Components**:
  The I/O pins connected to our microcontroller are particularly sensitive. By adding TVS diodes, we ensure that any voltage spikes don’t reach these crucial parts of the circuit.

Reliability of TVS Diodes for High Transient Voltages

- **Fast Response Time**:
  TVS diodes respond within picoseconds, effectively clamping kilovolt-level spikes introduced by button presses or touches.
- **Energy Dissipation**:
  Their ability to quickly dissipate energy makes them a reliable choice for protecting the circuit from potential damage.

Why We Selected the SP1003

- **Designed for ESD Protection**:
  The Littelfuse SP1003 is specifically engineered to handle ESD and transient voltages.
- **Compliance with Standards**:
  It meets the IEC 61000-4-2 standard, providing robust protection against:
  - Up to ±15 kV for contact discharge
  - Up to ±30 kV for air discharge
- **Performance Features**:
  - Clamps transients in less than 1 nanosecond
  - Low capacitance, ensuring no interference with button signals

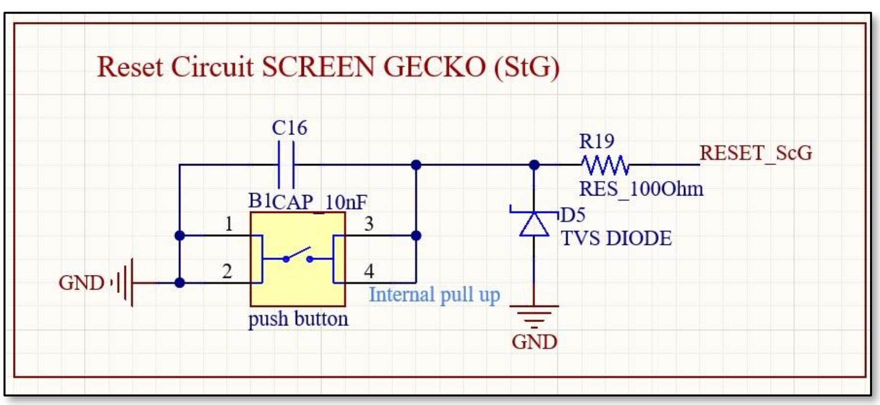
For USB Power:

We don't need an ESD diode for our USB power because we're only using it as a power source (VBUS) and not for data communication. Since no data lines are connected, the risk of transient voltages is much lower. The USB host already provides power protection, and any minor noise on the VBUS line can be handled by our power filtering capacitors, making the ESD diode unnecessary in this case.

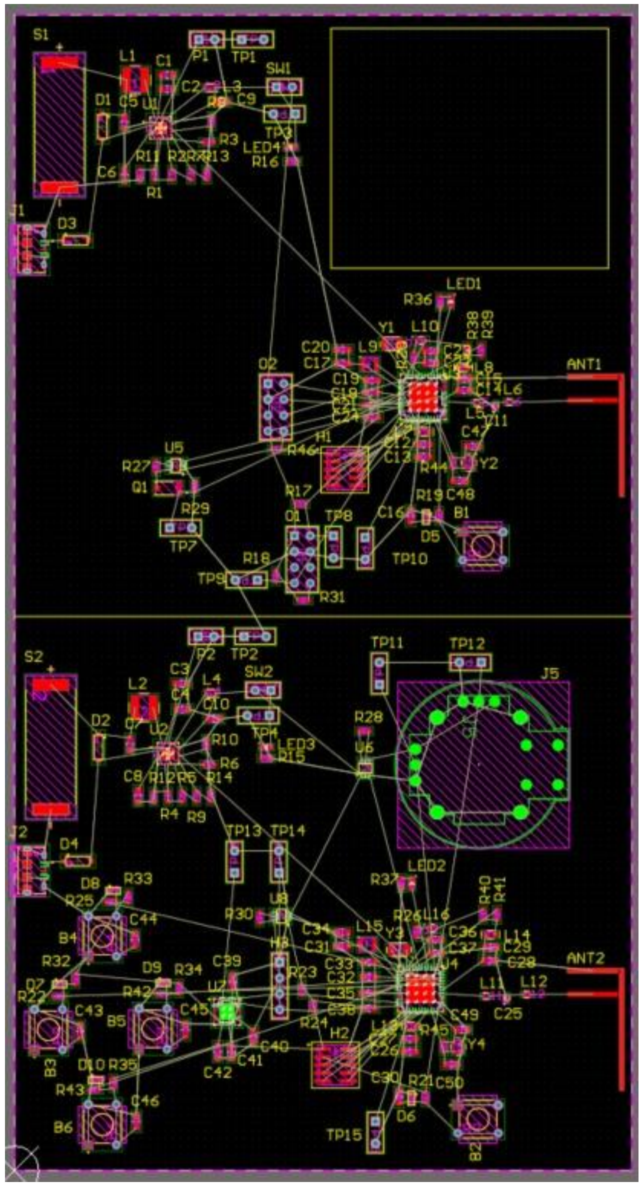
MISCELLANEOUS UPDATES

These changes were made in the schematic as per the TA's suggested after the schematic review.

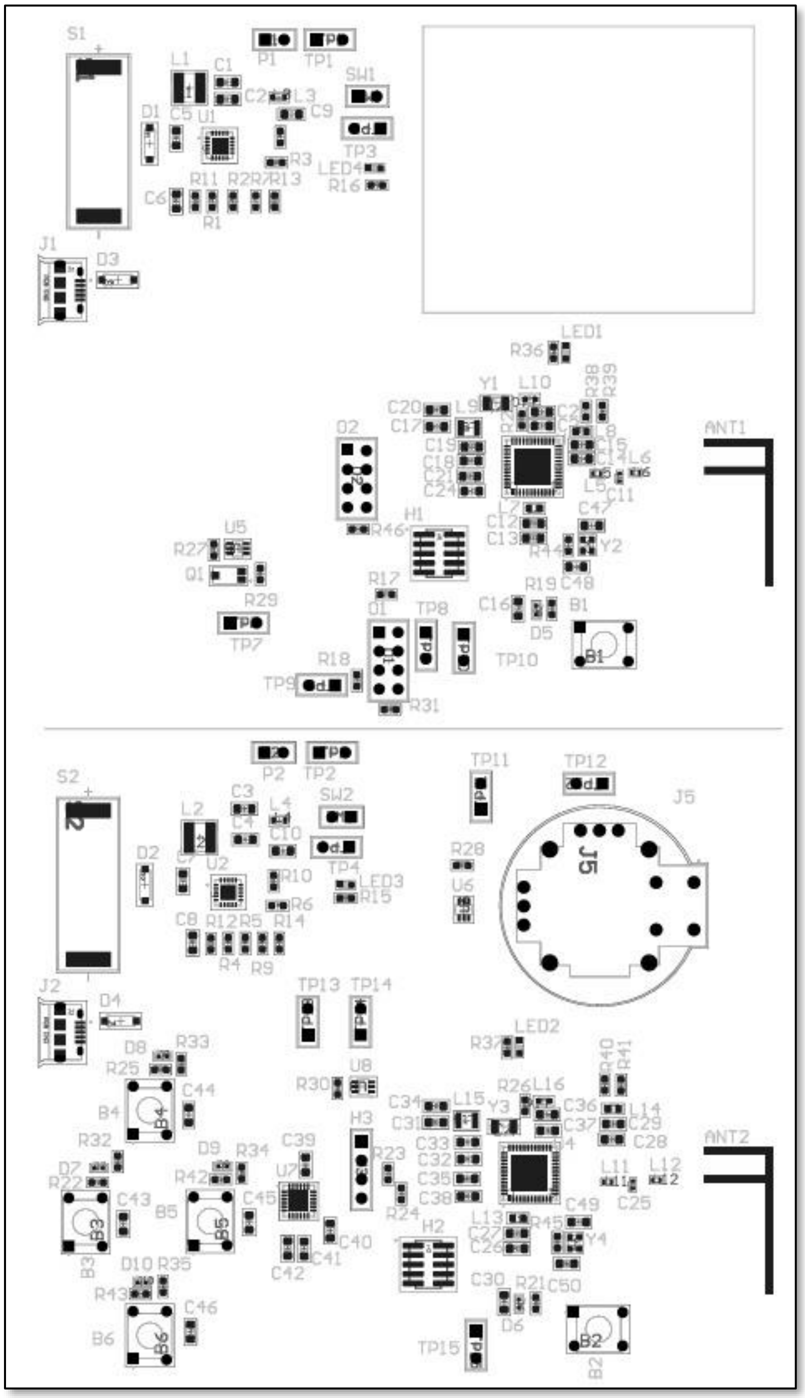
1. The values of the passive components were rewritten in the standard naming format.
2. Initially, the inductor we had put for the buck was 4.7uH as it was mentioned in the datasheet that the minimum value is 4.7uH. Later, we changed it to 10uH as a value more than recommended will not cause any harm.
3. We added an LED at the output of the PMIC - VOUT_3V3.
4. Removed the UFL connector and the corresponding resistors were removed since we are already using the F-Antenna, using both will cause impedance mismatching.
5. Added both SPI and I2C connections for our display.
6. Made changes to the reset circuitry as suggested by the TA's and added an ESD diode.



BOARD PLACEMENT LAYOUT



BOARD ASSEMBLY LAYOUT



PROJECT WEEK 6 UPDATE

STATUS REPORT

Addressing Week 4 & Week 5 Feedback

Week 4 Feedback:

I believe we discussed this already, but you should have two Schottky diodes, one for each input and they should come together to a single node.

- We have incorporated two Schottky diodes, one for each input, and both are now connected to a single node (VIN_DC) as discussed. This ensures proper handling of input signals as per your recommendation.

On your reset circuit, you want the resistor in series between the button and ground. This is to prevent pulling too much instantaneous current from the MCU, we aren't too worried about the 3v3 buck since it has decoupling caps on the output.

- The resistor has been added in series between the button and ground as suggested.

When you pick crystals, make sure the load capacitance is supported by the Blue Gecko.

- The load capacitance of the selected crystals has been verified to be compatible with the Blue Gecko, following thorough review by the SAs.

When thinking about adding external power to your board, if you are injecting power before the PMIC, consider the PMIC's max CONTINUOUS current limits.

- The BQ25570 can typically provide up to 100 mA continuous current on its VBAT and VSTOR outputs, depending on the power source and thermal conditions. The integrated buck converter output (VOUT) can support around 110 mA continuously under optimal conditions. For our application, the max continuous current allowable is around 100mA which is sufficient for the working of our sensors and actuators.

Week 5 Feedback:

Hi team, I see the MPU6050 is obsolete. I would consider looking at the MPU6500 or the BHI160B. Both are 3-axis I2C accelerometers.

- With the MPU6050 now obsolete, we are switching to the MPU6500 as the replacement. This IC supports the required 3-axis accelerometer functions and operates seamlessly with I2C, aligning with the project's requirements.

Identifying the ESD protection you need for your device. You still need to add power line USB ESD protection on your VBUS rail for USB. You can ignore the data lines since you are not using it.

- ESD protection has already been implemented on each pushbutton. Additionally, we will add power line ESD protection to the VBUS rail for the USB input, as suggested.
- Also, because we are using USB-C it has been advised to pull-down CC1 & CC2 pins using 5.1k ohm resistors. This is necessary to correctly detect the connection as a device, ensuring proper power negotiation, while the data lines remain unused in our design.

Activity Accomplished

Component Placement: After incorporating the schematic changes, we proceeded with the component placement, ensuring sufficient space for effective and efficient routing. Following a review by the SAs, we made necessary adjustments, such as modifying the F-antenna's distance to the board edge, refining designator names, and verifying RLC values for accuracy.

Component Routing: With component placement finalized, we moved on to routing. We maintained a 20-mil width for power traces and 9 mils for signal traces. Special attention was given to the F-antenna routing, where we carefully transitioned from 20 mils to 9 mils, accommodating the teardrop design while ensuring an uninterrupted GND plane beneath the F-antenna for optimal performance.

Board Design Shape Defined: The board shape was defined in accordance with our mechanical design specifications, ensuring the enclosure fits securely around the PCB.

Planned Activities

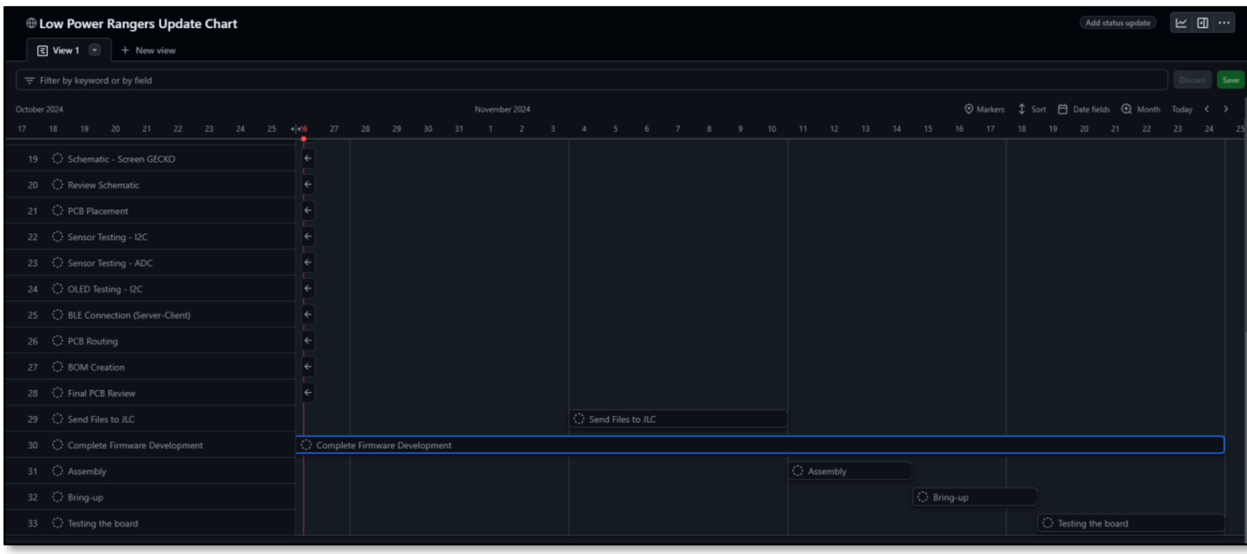
Overall PCB Review and Finishing: Before sending the design to the fabrication vendor, we are addressing minor adjustments to eliminate any potential design gaps. A final review session will be conducted with the SAs to ensure the design is ready to be sent.

Constructive Firmware Development: Firmware development is underway, with individual sensor testing already completed. The next steps involve integrating all sensors and actuators onto a single BG13 Dev-kit. We will also write GUI-specific drivers for the OLED and perform BLE testing across various GPIOs.

Schedule & Obstacles

We have replaced the MPU6050 with MPU6500 which is in stock on DigiKey. It is similar to the MPU6050, but we will have to write the code for interfacing with that IC. Also, we had to change the layout of our PCB to get the Joystick and the Buttons to align in a straight line. We were not able to configure some of the GPIO pins on the Development Kit to get to work as inputs for our push buttons. We are guessing this is due to the alternate functionality of the GPIO pins on the Dev Kit. Also, we were confused about which protocol to use for our OLED Display, so we have kept headers for both protocols, I2C and SPI respectively.

Update of the project Gantt Chart



Link: [GanttChart Link]

PROJECT WEEK 7 UPDATE

STATUS REPORT

Activity Accomplished

Final PCB Review by Professor:

After comprehensive reviews of the PCB with the professor and the TAs, the following changes were incorporated based on the feedback received:

1. Reduced the length of the cross-under to improve signal integrity.
2. Applied ground stapling for every three data signals crossing over the cross-under, ensuring a more efficient return current path to the vias.
3. Tented all vias to prevent solder wicking and ensure better electrical reliability.
4. Repositioned the joystick to avoid interference with the radio antenna, safeguarding the antenna's performance and range.
5. Enlarged the pad width for the solar panel to improve solder joint reliability.
6. Updated the footprint for the 1.9nH component for optimal fit and performance.
7. Relocated the decoupling capacitors closer to the ICs to reduce noise and enhance power delivery.
8. Shifted the trace from the noisy side of the inductor to the quiet side, eliminated cross-under, and minimized the switching loop to reduce EMI.
9. Added mice bites around the board cutout, with an accompanying keep out layer to ensure proper manufacturing.
10. Removed cross-under from the ADC signal lines to improve signal clarity and reduce potential interference.

DFM Check & Component Ordering:

We performed a DFM (Design for Manufacturability) check on our PCB Gerber files using JLCPCB and finalized the Bill of Materials (BOM). All components were added to the Digi-Key cart, and the order was placed for both the PCB and the components.

Planned Activities

Board Assembly: Upon receiving the PCBs, we will perform continuity tests to ensure there are no issues before starting assembly. The assembly process will follow a methodical approach to ensure smooth integration of components and prevent any errors. Structural testing will also be carried out as part of this process.

Firmware Development: Firmware development is in progress, with individual sensor tests already completed. The next steps involve integrating all sensors and actuators onto a single BG13 Dev-kit. We will also develop GUI-specific drivers for the OLED display and conduct BLE testing across various GPIOs using the latest 4.2.1 SDK.

Schedule & Potential Obstacles

We are currently adhering to the project timeline and making steady progress. As part of our risk management strategy, we have developed a contingency plan to address any unforeseen challenges that may arise.

Assembling Plan

Based on the Industry-Recommended PCB Assembly, we will keep our strategy aligned with the stated guidelines to ensure nuanced and smooth assembling.

1. **Preparation:**
    - Inspect PCB for defects; ensure components match BOM.
    - Verify orientation of polarized components (diodes, capacitors, ICs).

2. **Solder Paste Application:** Use stencil printing for even paste application; conduct an optical inspection for coverage accuracy. [**Reference: IPC-7525A]**

3. **SMT Component Placement:**
    - High-volume: Use pick-and-place machine for accurate alignment.
    - Prototypes: Manual placement with tweezers for precision. **[Reference: IPC-1601]**

4. **Reflow Soldering:** Ensure correct oven profile for proper solder joints, especially on QFNs/BGAs to avoid defects. **[Reference: IPC-610]**

5. **Inspection and Testing:** Check for solder defects. Verify continuity and functionality. **[References: IPC-A-610, IPC-2221]**

6. **Post-Assembly:** Clean PCB to remove flux; apply conformal coating if needed for protection. For the verification of the board, refer to our verification plan mentioned below.

VERIFICATION PLAN

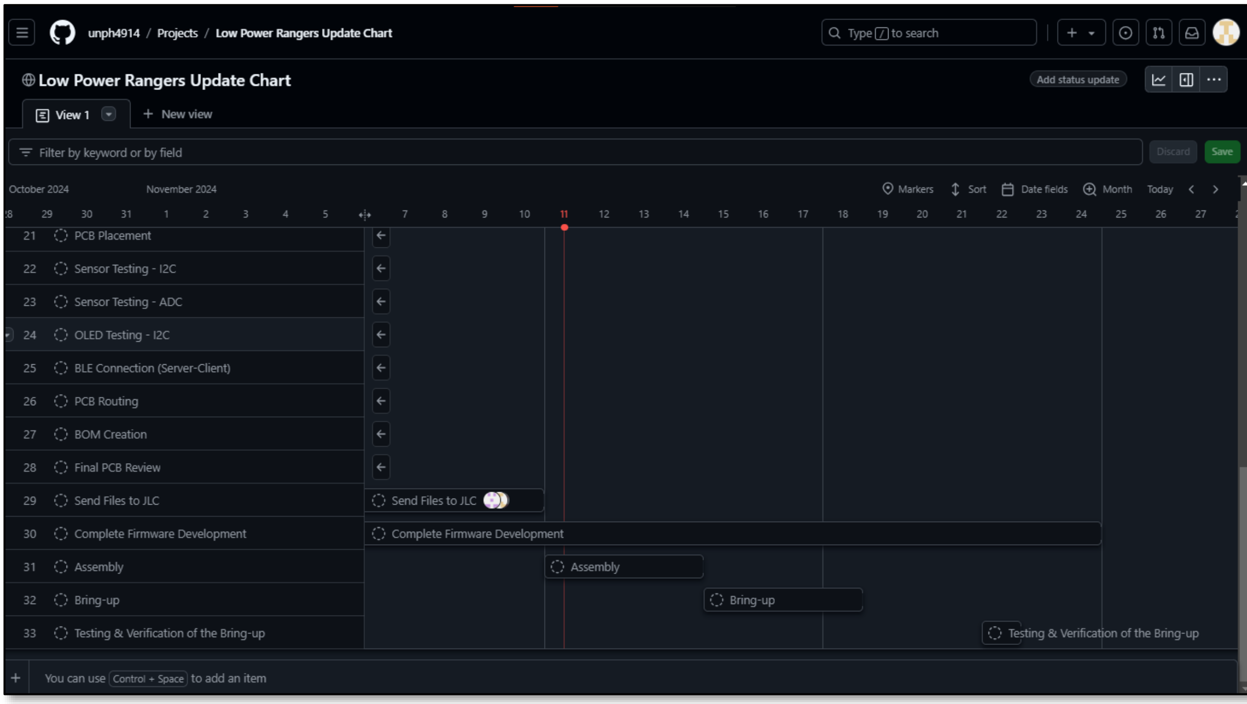
For the verification plan, we will be following this spreadsheet created on our GitHub Project Repository and keep committing the observations and changes we go through while verifying the board. Link to the same has been given below.

Verification Plan

Verification Item	Definition of Passing	Date Test Performed	Test Performed By	Measured Result	Pass (Y/N)	Remarks
1. Verify Each Power Supply Voltage (Min & Max)	Each power rail remains within ±5% of its specified value.					
2. Verify Signal Quality for Each Communication Bus (I2C, UART, SPI)	Signal levels meet expected thresholds and exhibit low noise. Scope screenshots show clear transitions.					
3. Verify Energy Storage Element Charging	Battery charges to its nominal voltage within the specified time at the standard charging current.					
4. Verify Power Good Signal and System Boot from Low to Sufficient Charge	System receives Power Good signal and boots up once energy storage element reaches sufficient charge level.					
5. Verify System Shutdown as Energy Storage Element Discharges Below	System gracefully shuts down once battery voltage reaches cut-off (3.0V).					

Link: [Verification Plan]

Update of the project Gantt Chart



Link: [GanttChart Link]

PROJECT WEEK 8 UPDATE

STATUS REPORT

Activity Accomplished:

Bare PCB from Fabrication Vendor

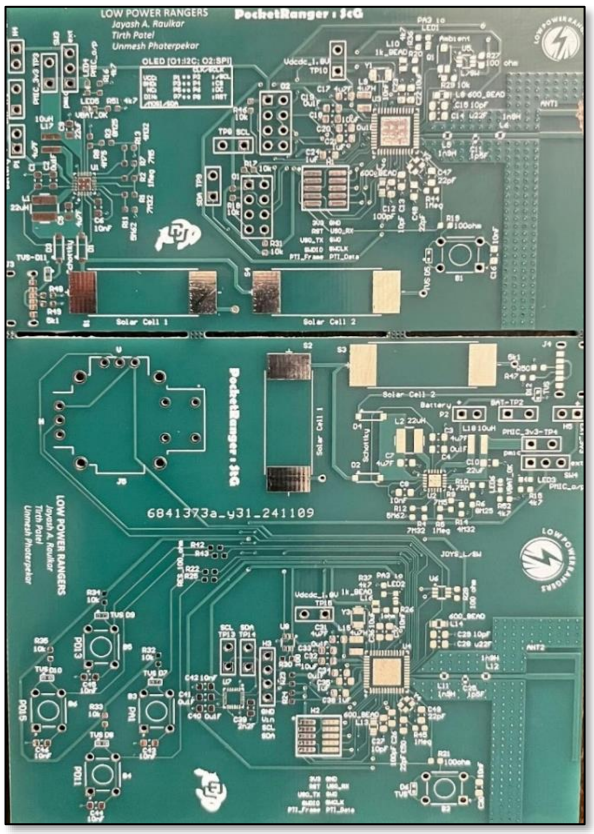


Figure 17- Bare PCB

Board Assemble

1. **Pick Place Machine:** Most components on the PCB, including integrated circuits (ICs), passive components (such as inductors, resistors, and capacitors), diodes, the USB-C connector, and the ambient light sensor, were assembled using a pick and place machine. After placement, these components were soldered by passing the board through a reflow oven, ensuring proper adherence and electrical connections.

2. **Manual:** Some large parts, such solar cells, and through-hole parts, like the buttons and joystick, were put together by hand using soldering techniques. Additionally, to ensure proper functioning and dependable connections, the USB-C connector needed to be manually re-solder due to continuity issues that we found

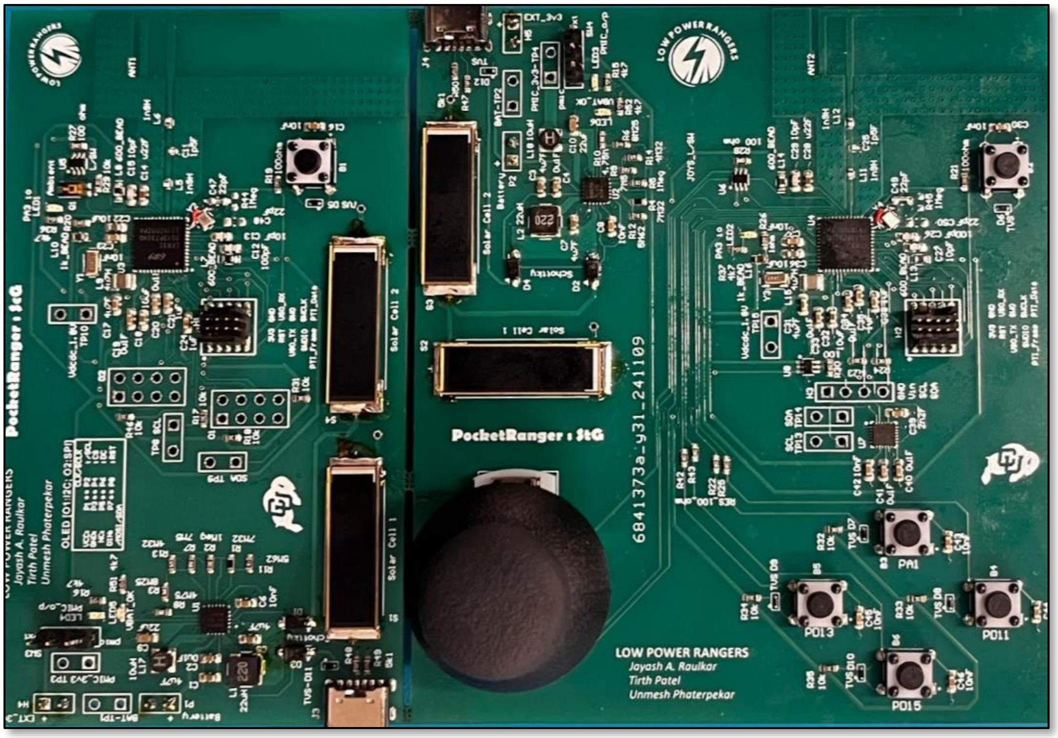


Figure 18 - Assembled PCB

Physical Connections:

1. **Solder joints Verification:** The solder joints were inspected under a microscope to ensure proper wetting and coverage. During inspection, a few cold solder joints were identified, primarily near the microcontroller IC and a capacitor. They were modified for proper electrical connectivity.

2. **Check for Shorts in connection:** Connections were checked for shorts using a multi-meter in continuity mode. A short connection was detected between the crystal pin and ground. This issue was resolved to restore proper functionality.

3. **Verify the supply on Power Traces:** Voltages on power traces were measured using a multi-meter and analysed for ripple with an oscilloscope. The 3.3V power rail was measured at 3.31V ± 20mV, and ripple was under 50mV under load. These measurements confirmed stable and consistent power delivery to all components.

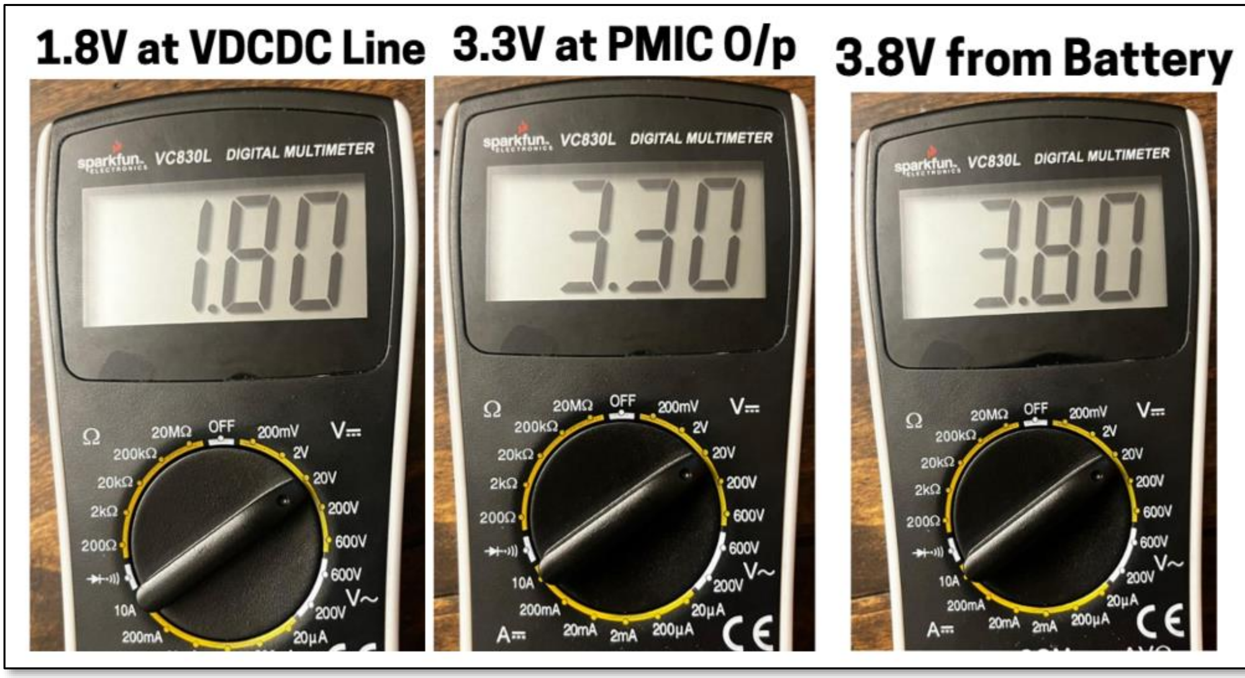


Figure 19 - Power O/P

As seen in Fig 19, we are getting a stable and constant power supply from the battery, at the output of PMIC and the VDCDC Line.

VBAT_OK & **PMIC_O/P** LEDs are functional, confirming the constant & stable supply as shown in Fig 20 below.

Figure 20 - VBAT_OK and PMIC O/P

Battery Power Up:

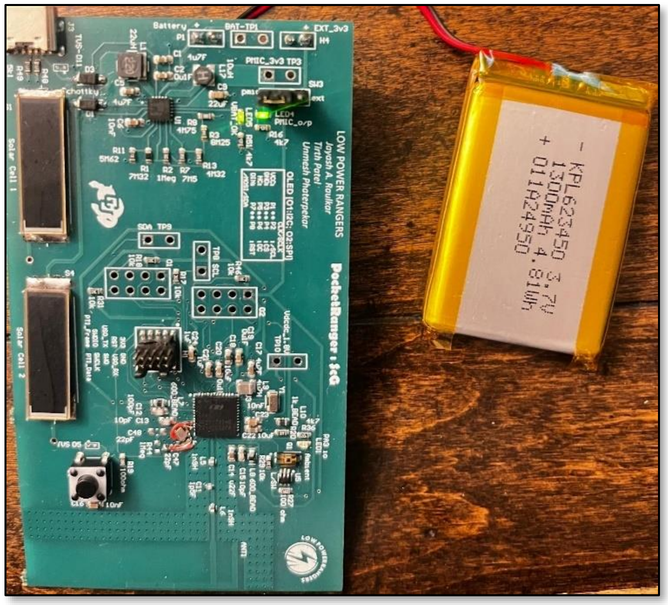


Figure 21- Battery Connection

Firmware Verification:

1. **SWD Connection:** The SWD connection was tested to ensure proper communication between the development board and the debugging tool. This step involved confirming that the SWD clock and data lines were functioning without any interruptions.

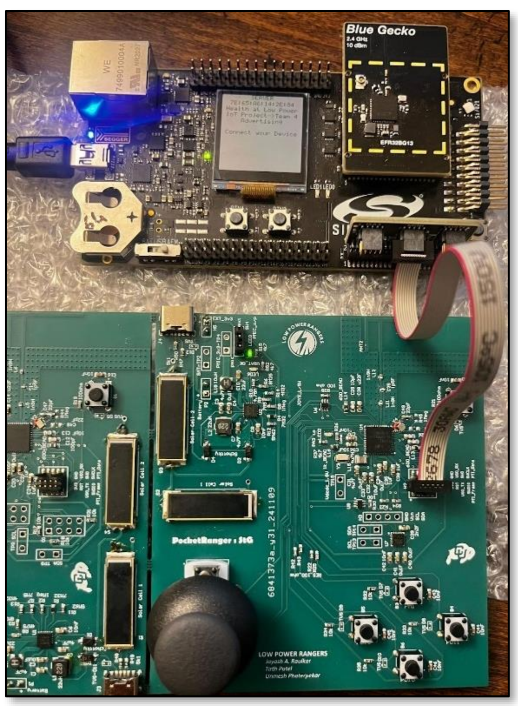


Figure 22 - SWD Connection

2. **Board detection on Simplicity Studio v5:** The board was successfully detected on Simplicity Studio v5, indicating that the hardware and firmware communication was established correctly. The detection process confirmed the proper configuration of the development environment.

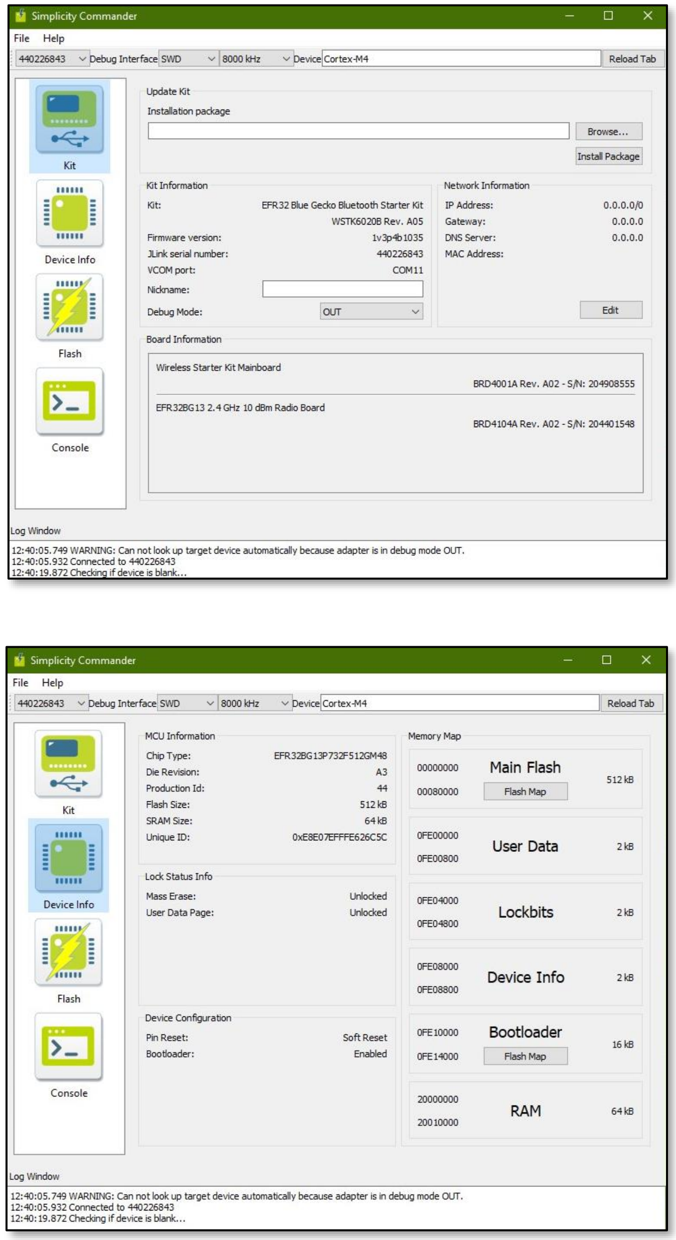


Figure 23- Simplicity Commander

Simplicity Commander successfully detected the target device (Pocket Ranger) and displayed the information of the connected board as shown in the Fig above.

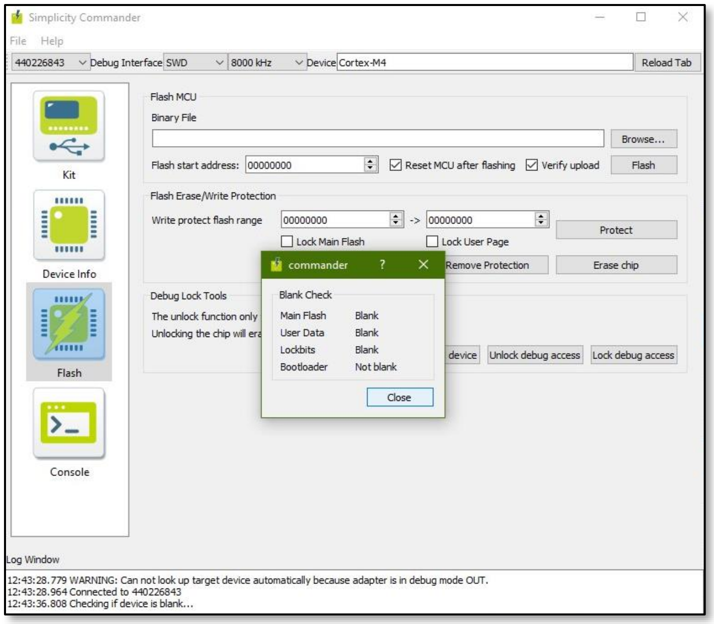


Figure 24 - Blank check

Blank check & successfully boot loaded.

3. **Flash Code to ensure Functionality:** The firmware was flashed onto the board to verify functionality. A simple test code was used to check if all peripherals, such as GPIOs, ADC, were operating as expected.

4. **Working Code (Energy Profiler):** The **Energy Profiler** in Simplicity Studio was used to monitor the working code. The expected voltage drop was observed when the reset button was pressed, confirming its proper functionality. Additionally, the GPIO functionality was successfully validated using a reference implementation from a previous course assignment.

Figure 25 - Energy Profiler

Currently flashed a simple LED toggle code on one of the GPIO pins. The random dips observed on the profiler are due to the behavior of the RESET pin. This confirms that the code has been flashed correctly and is functioning as expected.

Schedule & Obstacles:

Wrong 38.4MHz Crystal Footprint

Unfortunately, we overlooked cross-checking the footprint with the symbol of the crystal during the design phase. Instead of having pins on opposite ends, the crystal's pins were placed on the same side of the footprint. To resolve this issue, we had to modify the board by cutting a trace and connecting the other pad to the crystal using a small piece of wire. The original pad, where the crystal pin was previously connected, was then repurposed and connected to ground via a shorting 22pF capacitor. Might not look aesthetically appealing, but the continuity is complete without any shorts.

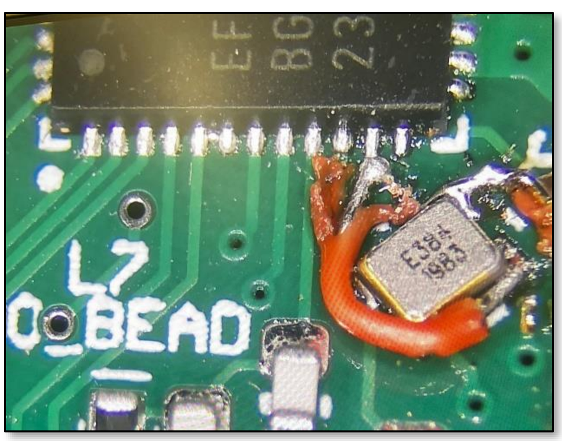


Figure 26 - Re-Soldered Crystal

Reverse TVS Diode

The TVS protection diode was inadvertently connected in reverse, which prevented our custom board from being detected by the Simplicity Commander. After identifying the issue as suggested by the professor and the TA's, we removed the diodes, allowing the board to be successfully recognized. This enabled us to flash the code onto the board without any further issues.

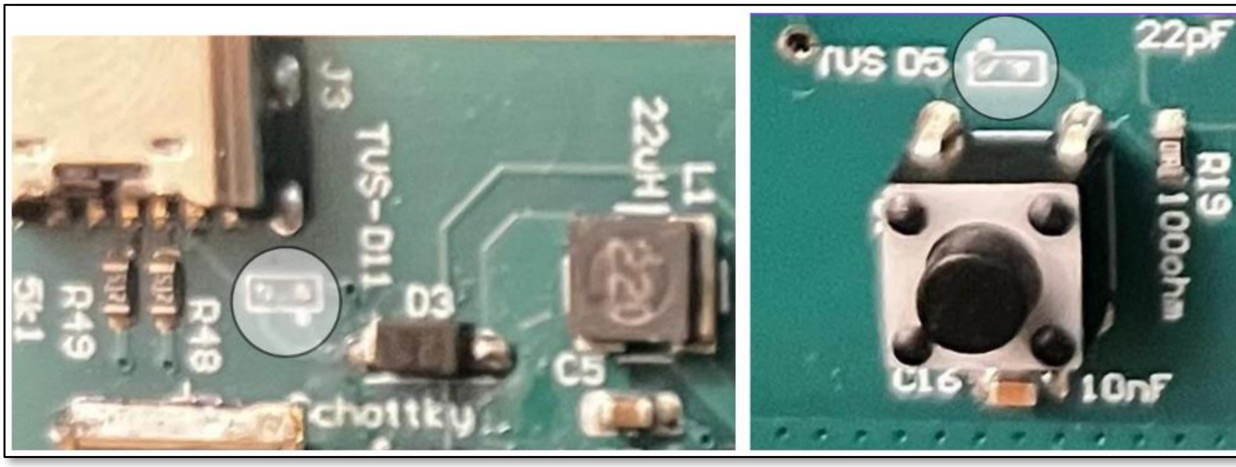


Figure 27 - Reverse TVS Diode

After removing the diodes, all 4 of the boards are **powering up correctly, detecting on IDE** & **able to flash the code.**

Planned Activities:

Run Event-driven code for Testing Sensors: This activity involves executing event-driven firmware to test and validate the functionality of specific sensors and peripherals integrated into the system. Each device will be tested for accurate response to events and compatibility with the system.

1. **Ambient Light Sensor with OLED Display**
2. **MPU Gyro Sensor**
3. **Push Buttons and Joystick**
4. **Load Switch configuration for low energy mode**

Verify the signals at each test point: To ensure the proper functioning of the circuitry, signals at designated test points will be verified using an oscilloscope or logic analyser. This step involves:

- Comparing measured signals against expected values.
- Identifying and resolving discrepancies due to noise, signal degradation, or hardware issues.
- Verifying continuity, voltage levels, and signal timing at critical nodes in the circuit.
- Documenting observations for future reference and debugging.

Bluetooth Testing: Bluetooth functionality will be thoroughly tested to validate connectivity and communication. This includes:

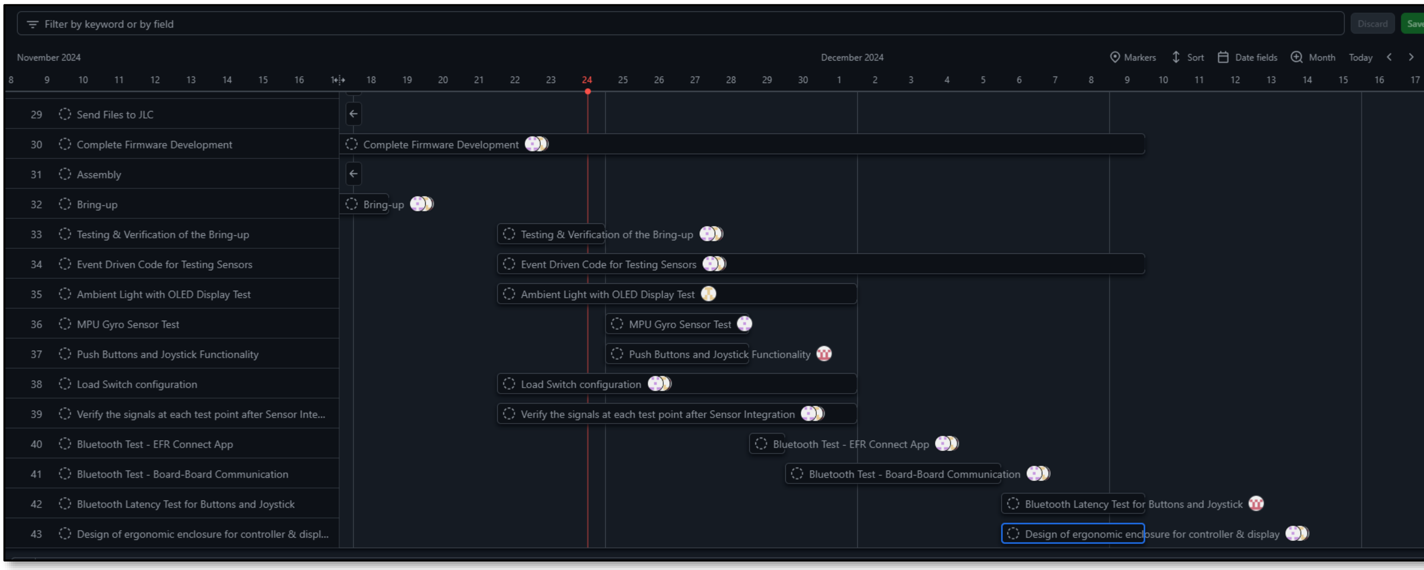
1. **With EFR Mobile App:** Pairing the board with the EFR Mobile App to test basic Bluetooth features like scanning, pairing, and data transfer. Additional features like reading sensor data or controlling peripherals via the app will be validated.
2. **Board-to-board communication:** Establishing communication between two identical boards via Bluetooth to test peer-to-peer communication. Data integrity, synchronization, and error handling in real-time scenarios will be analysed.
3. **Latency Testing with Push button & Joystick:** Measuring the latency between input events (push buttons & joystick movements) and corresponding Bluetooth responses.

Design ergonomic enclosure for controller & display: A custom enclosure will be designed to house the controller and display, focusing on ergonomic considerations and usability.

Figure 28 - Controller 3D Print Prototype

This prototype serves as a close reference to the concept we aim to develop. It demonstrates key functionalities and design principles that align closely with our objectives.

Update of the project Gantt Chart



Link: [Gantt Chart Link]