Department of Electrical and Computer Engineering

University of Colorado at Boulder

ECEN5833 - Low Power Embedded Design Techniques

Fall 2024, ECEN 5833

Course Project Report

GITHUB Project
GITHUB Code link

Submitted by
Parth Thakkar
Anagha Aditya
Akash Karoshi

Submitted on
September 28, 2024

Contents

List of Figures

List of Tables

PDF is clickable

1 Project Proposal

Team name:
Low Self Esteem Team

Student Name:
Parth Rajeshkumar Thakkar
ParthRajeshkumar.Thakkar@colorado.edu

Anagha Aditya
Anagha.Aditya@colorado.edu

Akash Karoshi
Akash.Karoshi@colorado.edu

1.1 Project Rationale and Goals

For the ECEN 5833 Low Power Embedded System Design course, our team is developing an advanced mechanical keyboard called "The Insane Keyboard". This project aims to create a high-performance input device that combines ergonomic design, customization options, and some cool features.

1.1.1 Project Rationale

Our analysis of the mechanical keyboard market revealed several issues:

- Ergonomic keyboards often lack additional features or are expensive
- Many feature-rich keyboards are wired, limiting mobility
- Affordable keyboards offer limited customization
- Few keyboards combine ergonomic design, wireless capability, programmable lighting, and smart functions
- Keyboards with displays or extra features often have poor power efficiency

1.1.2 Project Goals

We aim to create a mechanical keyboard with the following features:

- Split ergonomic design to reduce physical strain
- Wireless connectivity using Bluetooth Low Energy.
- Programmable RGB lighting with addressable LEDs
- Hot-swappable key switches for easy customization
- Low-power E-ink display for additional information
- Multi-device compatibility
- Advanced power management techniques
- Environment temperature and pressure sensing.
- Real Time clock module for timer, stopwatch and time features.
- Potential energy harvesting from typing (Yet to be seen)
- Open-source firmware for extensive customization
- Cost-effective design for market accessibility

1.1.3 Unique Features

"The Insane Keyboard" has unique features like:

- Integration of multiple desirable features in one device
- Optimized power management for extended use
- Potential energy harvesting from typing motions
- Open-source firmware for community-driven development
- Adaptable design for future upgrades

1.1.4 Expected Outcomes

Upon completion, this project could:

- Improve ergonomics in daily computer use
- Advance keyboard power management and energy harvesting
- Foster a community of keyboard enthusiasts and developers
- Demonstrate practical applications of low-power embedded design

1.2 Existing Products and Ideas from products

Figure 1: Split Keyboard without display(Wired)

Figure 2: Keyboard which is expensive and have a Display

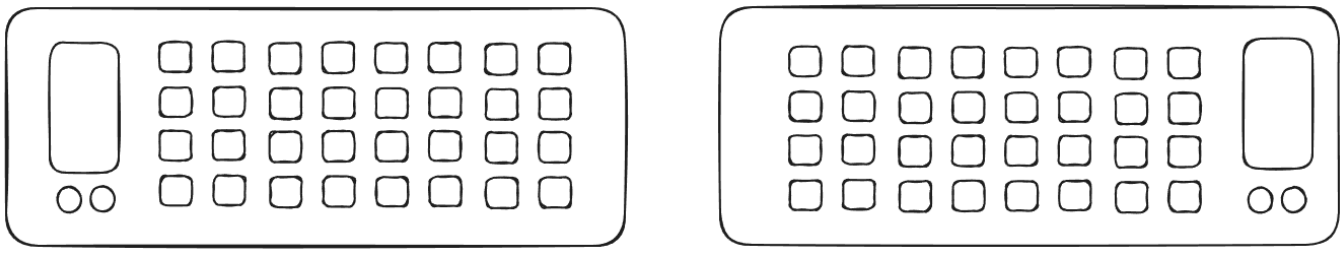


Figure 3: Conceptual Design of The Insane Keyboard

1.3 High Level Requirements

1.3.1 General Requirements

- Two EFR32BG13 boards (one per module)
- Should be Ergonomic
- 75% keyboard layout (TKL layout)
- IO expander for each MCU
- Temperature sensor
- Pressure sensor

- Real-Time Clock (RTC)
- Connectivity to three host devices
- Computer software for LED customization
- Individually customizable LEDs
- E-ink display on each module
- Energy harvesting on each module
- Charging circuit with Type-C connector
- Charging indicator
- Battery indicator
- Minimum 6-key rollover
- HID protocol communication between modules
- Smart power states (active, idle, deep sleep)
- RTOS: FreeRTOS (Yet to be decide)
- PWM-based RGB LED control
- Hot-swappable key switches
- Time, date, battery status, current profile, custom graphics
- Spotify integration for music control and now-playing information(yet to be seen)

1.4 Keyboard Layout and Design

1.4.1 Main Board Key Layout

Function Row						
ESC ~ ‘	F1 1 !	F2 2 @	F3 3 #	F4 4 $	F5 5 %	F6 6 ^
QWERTY Row						
TAB	Q	W	E	R	T	
Home Row						
CAPS	A	S	D	F	G	
Bottom Row						
SHIFT	Z	X	C	V	B	
Modifier Row						
CTRL	OPT	WIN/MAC	ALT	SPACE		

Additional Inputs:

- 2 × Rotary Encoders (Knobs)
- 3 × Extra Buttons

Additional Components

- E-Ink display
- Battery
- Battery Charging and Power Management Unit (Not in Secondary module)
- Tempurature Sensor (Not in Secondary module)
- Real time Clock (Not in Secondary module)

1.5 Secondary Module Key Layout

Function Row							
F7	F8	F9	F10	F11	F12	BKSP	HOME
QWERTY Row							
Y	U	I	P	{[	}]	—\	DEL
Home Row							
H	J	K	L	; :	” ’	ENTER	PGUP
Bottom Row							
N	M	, ¡	. ¿	/ ?	SHIFT	↑	PGDN
Modifier Row							
	SPACE	ALT/MAC	FN	CTRL	←	↓	→

Additional Inputs:

- 2 × Rotary Encoders (Knobs)
- 3 × Extra Buttons

Additional Components:

- E-Ink display
- Battery
- Power Management Unit

1.6 Challenges and Considerations

- Managing state machines with complex software like e-ink display drivers
- Integrating drivers with BLE firmware
- Implementing RTC driver and temperature control driver
- Ensuring reliable wireless connectivity
- Implementing multi-host communication and switching
- Developing HID profile in BLE stack
- Implementing anti-ghosting techniques
- Optimizing keyboard scan rate for low latency
- Achieving N-key rollover or 6-key rollover

2 Project Update: Week 1

2.1 Technical Considerations

2.1.1 Functional hardware block diagram

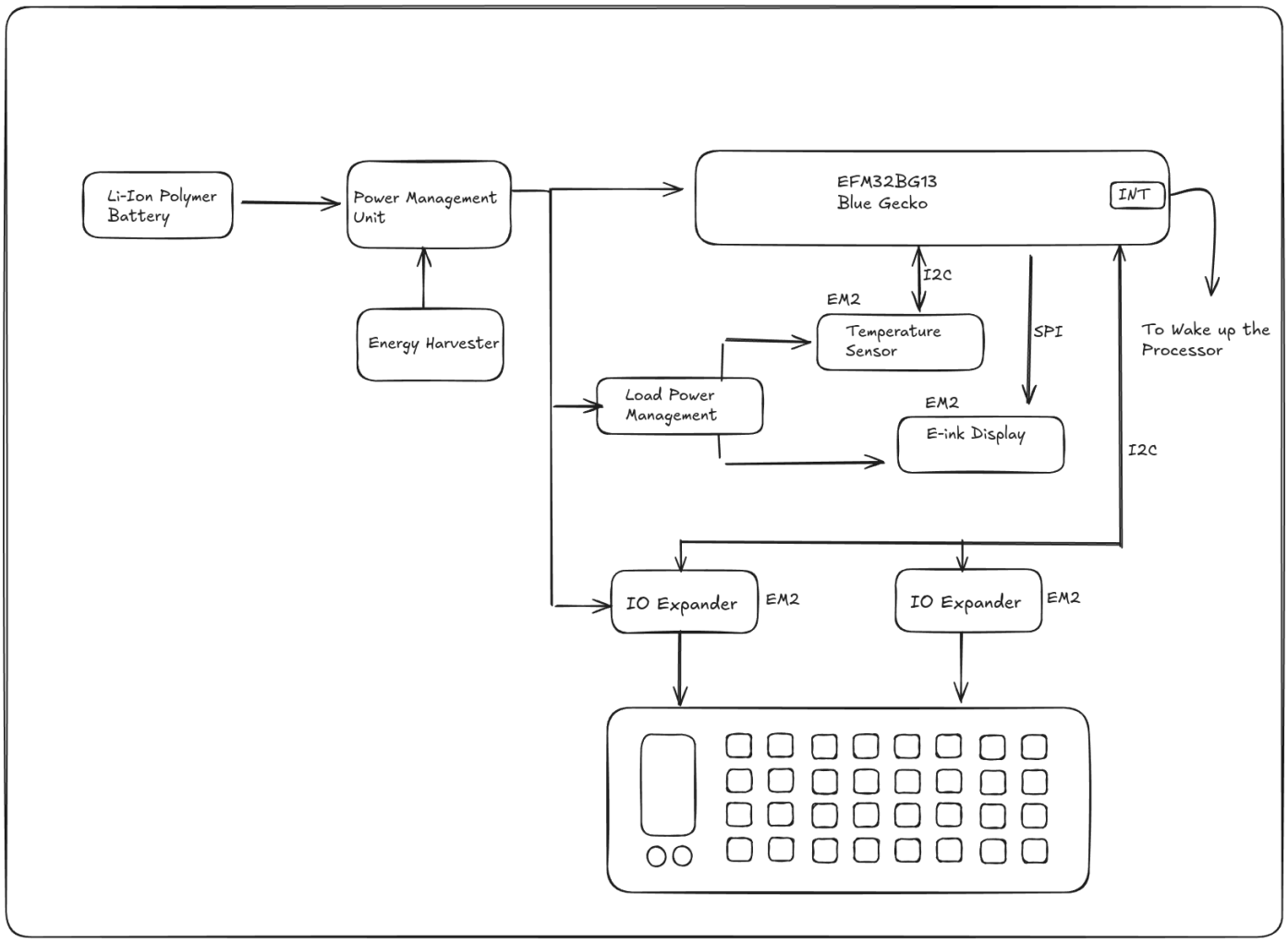


Figure 4: Hardware block diagram

0-Key Rollover (Normal Matrix)

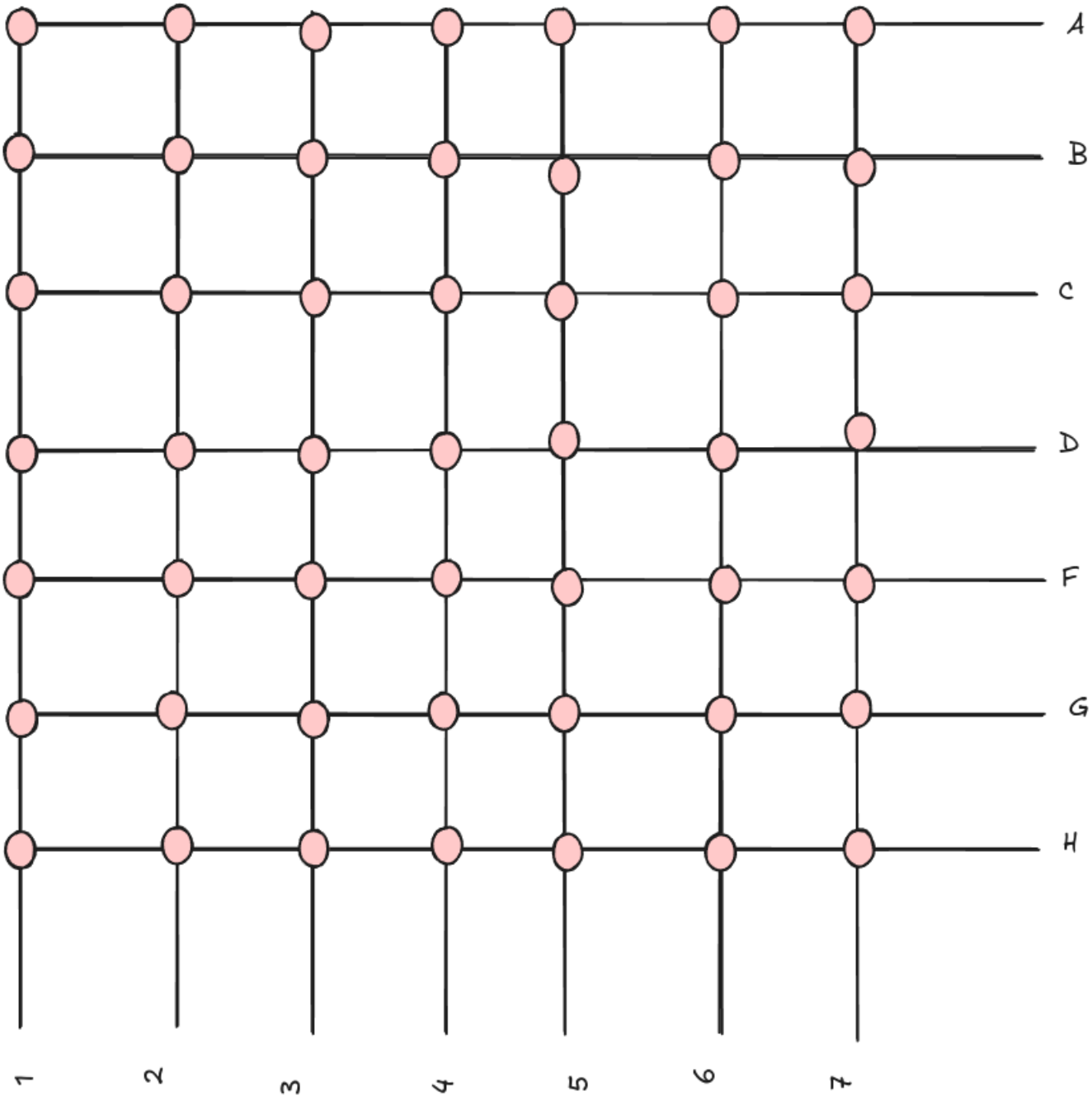


Figure 5: Normal Matrix for no key roll over

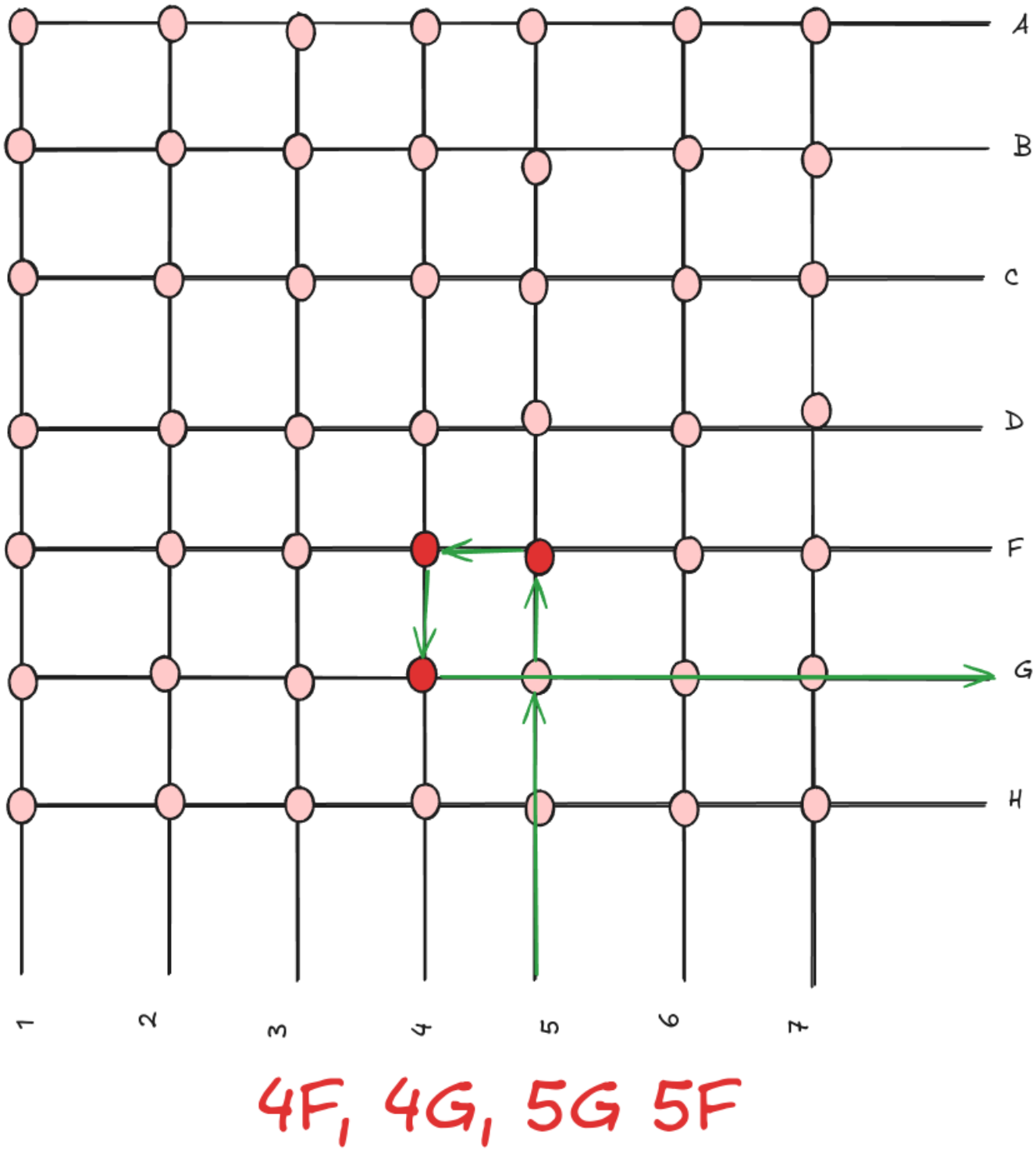


Figure 6: Decoding wrong button press

N-Key Rollover feature with Diodes

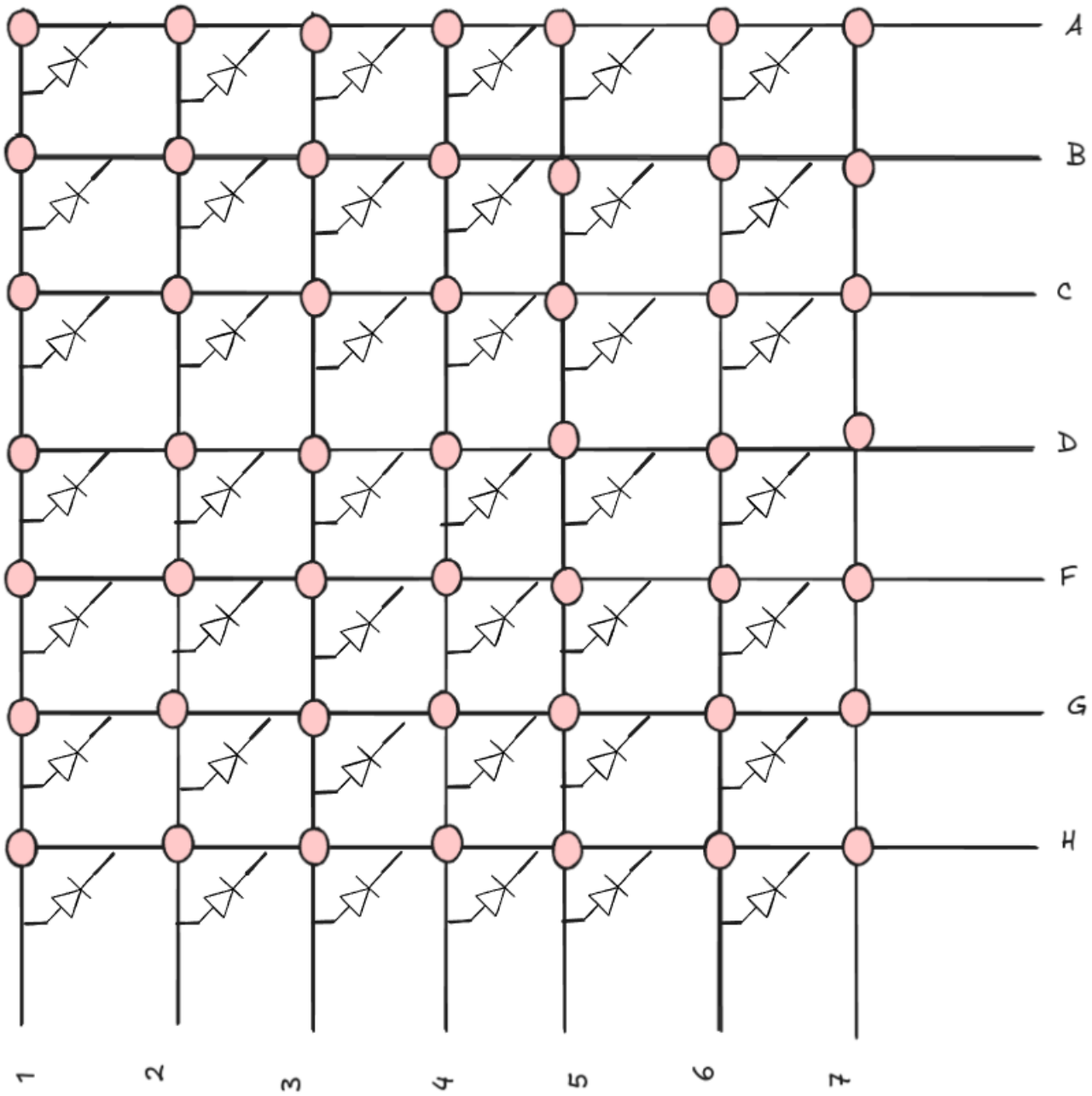


Figure 7: N-Key Rollover

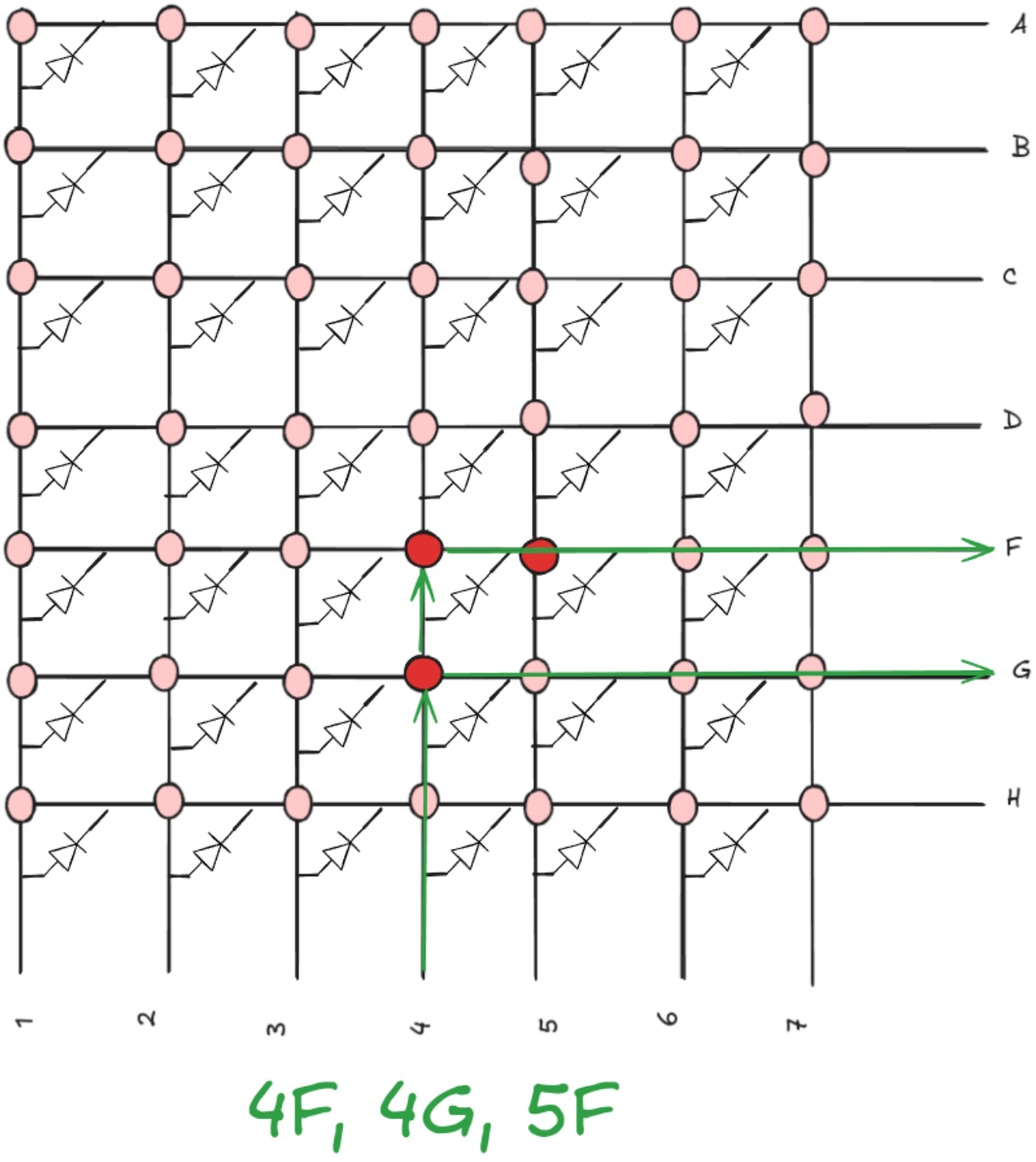


Figure 8: Multiple keys pressed

The keyboard uses a matrix layout, which is an efficient way to manage multiple key inputs with fewer microcontroller pins. In the images, we see an 8x7 matrix (rows A-H, columns 1-7).
Scanning Process:

The microcontroller scans this matrix by activating one column at a time and reading the state of all rows. When a key is pressed, it connects a row and column, which the microcontroller detects.

Ghosting and Its Prevention:
Ghosting is a problem in simple matrix designs where pressing multiple keys can lead to false key registrations. Example (Image 1):

If keys at 4F, 4G, and 5F are pressed simultaneously, the matrix might also falsely register 5G.

Solution (Images 2 and 3):
Diodes are added to each key switch. These diodes allow current to flow in only one direction, preventing the "phantom" current paths that cause ghosting. With diodes, when 4F, 4G, and 5F are pressed, 5G is not falsely registered.

N-Key Rollover (NKRO):
NKRO allows the keyboard to correctly register any number of simultaneous key presses.
Implementation:

Each key gets its own diode, ensuring independent registration. The microcontroller scans the entire matrix rapidly, detecting all pressed keys without conflicts.

Hardware Components:
Central Microcontroller:
EFM32BG13 Blue Gecko: This MCU manages all keyboard functions and interfaces with other components.

Sensors:
TMP117: Measures temperature, humidity, and pressure, potentially for environmental adaptation or user information.
DS3231 RTC Module: Provides accurate timekeeping, useful for time-based functions or logging.

Display:
E-Ink Display (800x600, 4.3inch): Offers a low-power way to show keyboard status, settings, or other information.

Expansion:
MCP23017 IO Expander: Increases the number of available pins for the key matrix, allowing for more keys or other inputs.

Power Management:
Power Management Unit: Manages power distribution and consumption.
Li-Ion Polymer Battery: Provides portable power.
Energy Harvester: Potentially extends battery life by capturing ambient energy.

Key Switches and Caps:
Gateron G Black Pro 2.0 switches: Known for smooth linear action. Cherry MX key caps: Industry-standard keycaps for customization.

Additional Inputs:
Rotary encoders: Provide alternative input methods, possibly for volume control or menu navigation.

PCB Design:
4-layer PCB: Allows for more complex routing and better signal integrity.
Two versions:

Main Module with all sensors and charging capabilities.
Simplified Secondary Module without temperature, humidity, RTC, and LiPo charger.

2.2 Wireless Communication and Software Architecture

We have made significant progress in defining our wireless communication protocol and software architecture:

2.2.1 HID Profile Characteristics

We've detailed the HID (Human Interface Device) profile for our keyboard, including:

- Keyboard Input Report: 8-byte structure for key presses
- Keyboard Output Report: 1-byte for LED status
- Protocol Mode: Supports both Boot and Report protocols
- HID Information and Control Point: For device management
- Report Map: Defines input/output report formats

2.2.2 Software Architecture

We've developed a functional software block diagram (Figure 9) that outlines the interaction between various software components, including the BLE stack, keyboard matrix scanning, and power management modules.

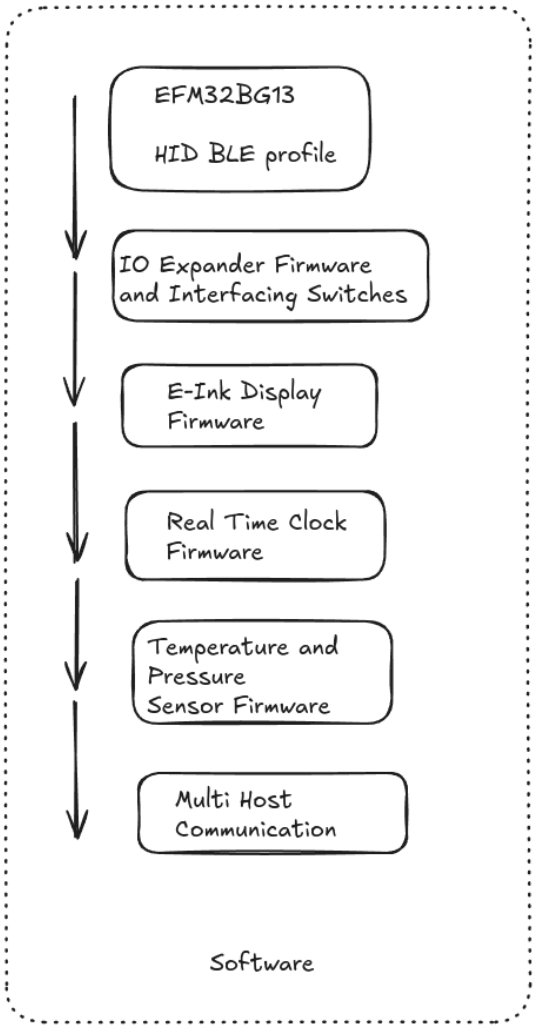


Figure 9: Software Block Diagram

2.3 Project Management

We've set up a robust project management system:

1. Timeline: We've created a detailed project timeline (Figure 10) to track our progress and milestones.
2. Version Control: Our project is now on GitHub, allowing for better collaboration and code management.
   - Project Board: GitHub Project
   - Code Repository: GitHub Code Repository
3. Task Tracking: We're using a Kanban board (Figure 11) to manage our tasks and workflow efficiently.

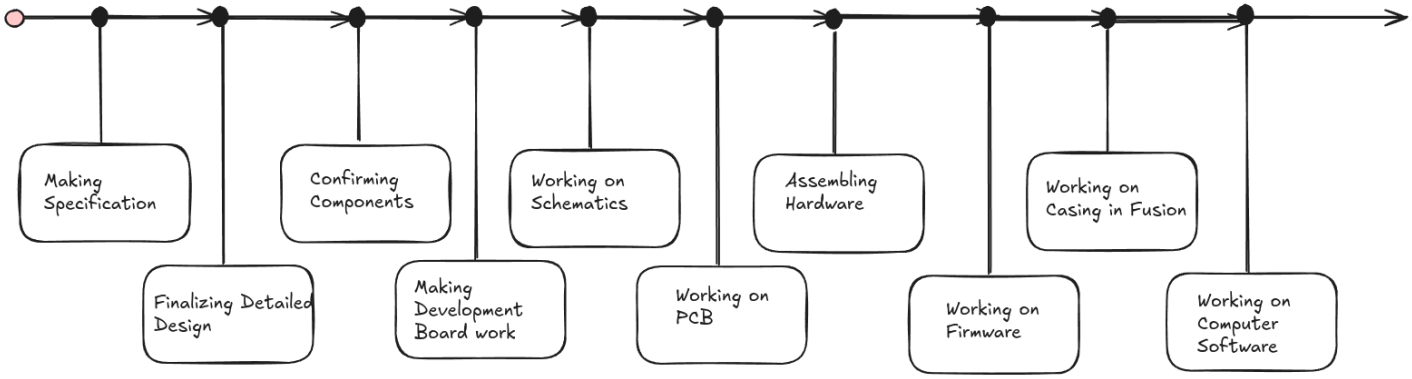


Figure 10: Project Timeline

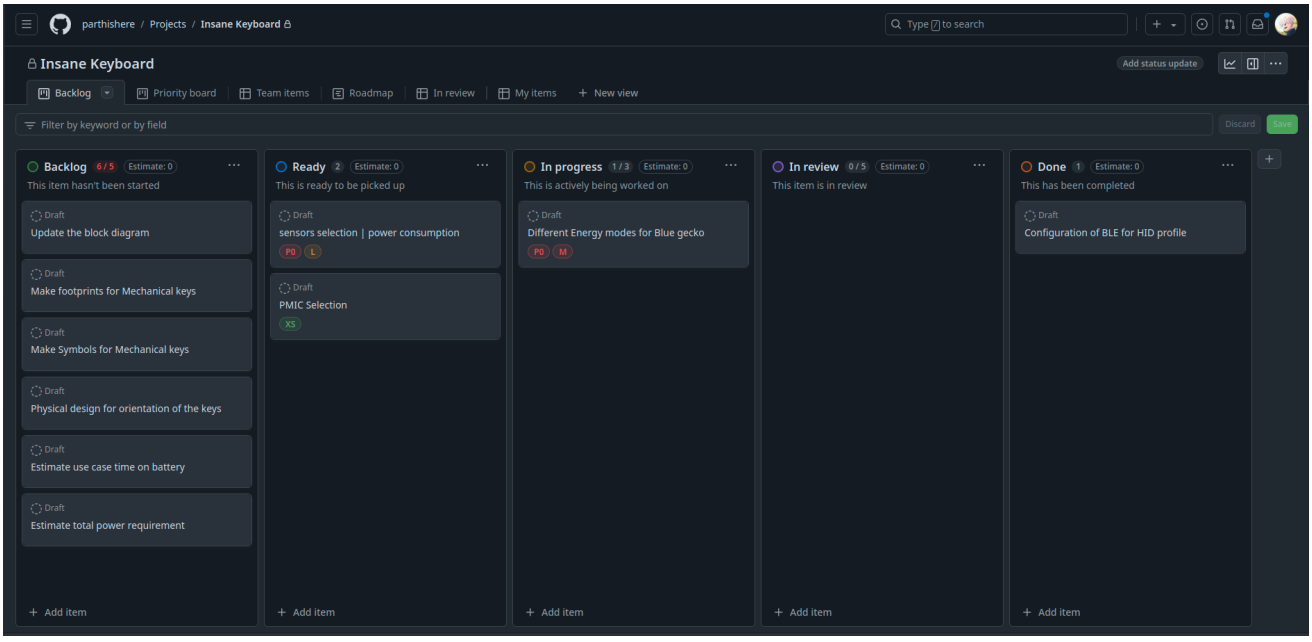


Figure 11: Kanban Board for Task Management

2.4 Energy Analysis and Power Management

We've conducted initial energy mode analysis and made decisions on power management:

1. Energy Calculations: We've performed preliminary calculations (Figure 12) to estimate power consumption in various modes.

2. Energy Harvesting: We've chosen to incorporate a solar panel for energy harvesting, enhancing the keyboard's power autonomy.

3. Interface Selections:

   - Display: SPI interface chosen for faster data transfer
   - Temperature Sensor and IO Expander: I2C interface for power efficiency

4. Load Power Management: Implemented a load switch to disable the temperature sensor and display when not in use, reducing standby power consumption.

5. Multiple Energy Modes: Our design now supports various energy modes to optimize power usage based on the keyboard's current state.

Component	Description			Current	Start up	Active	Sleep	Deep Sleep
BlueGecko								
	EM3	RTCC	Using ULFRCO	0.000003		1	1	1
	EM2		Using PLFRCO	0.0000033		0	1	
	EM0		38.4MHz clock	0.0049152		1		
Power Supply								
	Quiescent			0.0000007	1	1	1	1
Eink display								
	RefreshPower			0.03		1	1	
	Standby			0.000017		1	1	
	Quiesent Current							
Load Switch x2								
	Quiescent			0.00003	1	1	1	1
	Resistance ON State		0.2 OHMS					
Si7021								
	Standby			0.00000006	1	1	1	1
	Startup			0.004	1			
	I2C transaction			0.004				
	Humidity and temp reading			0.00018				
	Average Current during reading			0.0008		1	1	
IO expander								
TCA6408A	Quiscent current			0.0000015	1	1	1	1
				Current(A)	0.00403226	0.0308493	0.030849	3.23E-05
				Power(W)	0.01330646	0.1018026	0.101803	0.000106
				Time (s)	105	86400	86400	403200
				%Time	0.00018226	0.1499727	0.149973	0.699872
				Energy(J) = Power x %Time	2.4252E-06	0.0152676	0.015268	7.45E-05
				Average power(W)	0.03061213			
				Energy required for 24 hour(J)	2644.88832			
				Min capacitance (F)	0.38265167			
				Energy needs of the device and application (mins)	1.9047619			

Figure 12: Energy Calculations

3 Project Update: Week 2

3.1 Selection Criteria

We chose components based on:

1. Availability: Multiple suppliers to ensure steady supply.
2. Power efficiency: Low standby and active currents, multiple power-saving modes.
3. Documentation: Comprehensive datasheets, application notes, and support.
4. Sleep modes: Various low-power states with quick wake-up capabilities.
5. Temperature range: Suitable for varied environments.
6. EMC/EMI: Built-in interference mitigation.
7. ongevity: Components not near end-of-life.
8. Cost: Balanced features and performance with price.

3.1.1 Chosen Components

Component	Quantity	Function	Interface	DigiKey Part Number
EFR32BG13	2	Microcontroller	-	EFR32BG13P632F512GM32-D-ND
TCA6408A	2	IO Expander	I2C	296-24328-1-ND
Waveshare E-ink Display	1	Display	SPI	External Link
TPS22919	1	Load Switch	-	296-53421-1-ND
BQ25570	2	PMIC	-	296-37014-1-ND
Battery 1000mAh	2	Energy Storage	-	1939-BL1200P4054481S1PCAC-ND
SI7021	1	Temperature and Humidity Sensor	I2C	336-4379-1-ND
Schottky Diodes	1	Power Management	-	Have not decided yet
Gateron G Black Pro 2.0	68	Key Switches	-	External link
Mechanical Key Hotswap	68	Hotswap Sockets	-	External Link
Cherry MX Key Caps	68	Key Caps	-	External link
Rotary Encoders	2	Input Device	-	PEC12R-4222F-S0024-ND
Solar Panel SM141K06L	1	Energy Harvesting	-	SM141K06L-ND

Table 1: Component List for The Insane Keyboard with DigiKey Part Numbers

3.2 Justification for Each Component

3.2.1 EFR32BG13 Microcontroller

- Bluetooth 5.0 capable
- Energy-efficient: 60 uA/MHz at 40 MHz
- Five power modes: 1.4 uA in deep sleep, 20 nA in shutoff
- Low-energy peripherals for efficient keyboard scanning
- Comprehensive development tools

3.2.2 TCA6408A I/O Expander

- 0.1 uA standby current
- 400 kHz I2C interface for quick I/O updates
- 8 programmable I/O pins, each sinking 25 mA
- Interrupt output for event-driven operation
- Chosen over PCA9538 and SX1509 for lower power and simpler programming

3.2.3 Waveshare E-ink Display

- Power used only during updates
- Enables dynamic key labeling
- SPI interface for fast updates
- 200x200 pixel resolution
- Preferred over OLED for lower power consumption

3.2.4 TPS22919 Load Switch

- 12 nA quiescent current
- 15 us response time
- 2A current capacity
- 1.5V to 5.5V input range
- Built-in protection features
- Chosen over TPS22860 and SiP32431 for balanced performance

3.2.5 BQ25570 Power Management IC

- 100 mV to 5.1 V input range
- Integrated Maximum Power Point Tracking
- 488 nA quiescent current
- Built-in boost charger and buck converter
- Programmable voltage thresholds
- Selected over SPV1050 and ADP5090 for solar power optimization

3.2.6 SI7021 Temperature and Humidity Sensor

- ±0.4°C and ±3%RH accuracy
- 150 nA standby current
- I2C interface
- 3x3 mm package
- 1.9V to 3.6V supply range
- Chosen over HDC1080 and BME280 for accuracy and power efficiency

3.2.7 Solar Panel (SM141K06L)

- 42mm x 23mm size
- 184 mW maximum output
- 3.35V at maximum power point

3.2.8 Battery (1000mAh)

- Size: 2.09" x 1.89" x 0.16"
- 525mA charge current, 1.05A discharge current
- 3.7V nominal voltage

3.3 Energy Analysis

3.3.1 Overall Energy Efficiency

We have made the following assumption for our energy calculations: The keyboard will be operational 24/7, with the following daily usage pattern: 8 hours of active use per day

- 4 hours in active mode
- 4 hours in sleep mode

16 hours in deep sleep mode

Energy Consumption Breakdown Based on our calculations, the energy consumption for different operational states is as follows:

- Start-up: 0.000101970 W
- Active: 0.0166035556 W
- Transmit: 0.129140610 W
- Sleep: 0.000122760 W
- Deep Sleep: 0.00011187 W

Average Power Consumption The weighted average power consumption, considering the time spent in each mode, is calculated to be:

$$P_{avg} = 0.003976168 \text{ W} \tag{1}$$

This remarkably low average power consumption is achieved through intelligent power management and efficient component selection.

Daily Energy Requirement Based on our average power consumption, the total energy required for 24 hours of operation is:

$$E_{daily} = 343.5409345 \text{ J} = 0.095428 \text{ Wh} \tag{2}$$

Battery Capacity Calculation For sustained operation over a week, we calculated the required battery capacity:

$$E_{weekly} = 2404.786542 \text{ J} = 0.667996262 \text{ Wh} \quad \text{Adjusted Energy} = \frac{0.667996262}{0.85 \times 0.8} = 0.982347443 \text{ Wh} \quad \text{Battery Capacity}$$

This calculation takes into account a battery efficiency of 85% and a maximum depth of discharge of 80% to ensure long-term battery health.

Energy Harvesting Efficiency Our solar panel and power management system are designed to harvest energy efficiently even in low-light indoor conditions. The BQ25570 PMIC, with its Maximum Power Point Tracking (MPPT) capability, ensures optimal energy extraction from the solar panel.

Power Budget Analysis

- Average current draw: 1.07464 mA
- Peak current (during transmission): 39.0296560 mA
- Quiescent current (deep sleep): 33.9 uA

The significant difference between peak and quiescent current underscores the effectiveness of our power-saving modes.

Capacitor Sizing To support peak current demands, we calculated a minimum capacitance of:

$$C_{min} = 0.049702103 \text{ F} \tag{3}$$

This capacitance ensures stable operation during high-current events while allowing the solar harvesting system to replenish energy during low-power periods. With an average power consumption of just 3.976 mW and a weekly energy requirement of less than 1 Wh, our keyboard is well-positioned for long-term, self-sustained operation using solar energy harvesting, even in typical indoor environments.

3.4 Battery and Power Management

3.4.1 Power Management IC (PMIC) Selection

We have selected the BQ25570 PMIC as it provides the necessary features for efficient solar harvesting and battery management. Key features include:

- Wide input range: 100 mV to 5.1 V, ideal for solar harvesting
- Integrated Maximum Power Point Tracking (MPPT)
- Ultra-low quiescent current: 488 nA typical
- Integrated boost charger and buck converter
- Programmable voltage thresholds for optimal battery management

3.4.2 Energy Storage Element Selection

Choice: SM141K06L Solar panel

- Power Output: 184 mW
- Voltage at Maximum Power Point (Vmpp): 3.35 V
- Current at Maximum Power Point (Impp): 55.1 mA
- Open Circuit Voltage (Voc): 4.15 V
- Short Circuit Current (Isc): 58.6 mA
- Size: 42.00mm x 23.00mm x 2.10mm (1.654" x 0.906" x 0.083")

Reasons for selection:

- Significantly higher power output compared to previous options
- Larger but still manageable size for a keyboard design
- Higher voltage output, beneficial for power management
- Good power-to-size ratio

Energy Calculations (assuming 8 hours of typical indoor lighting per day):

Energy per day $= Power * Time$ Energy per day $= 184mW * 8hours$ $\boxed{\text{Energy per day} = 1,472mWh}$

Power Density:

$$\text{Power Density} = \frac{\text{Power Output}}{\text{Area}} = \frac{184 \text{ mW}}{42 \text{ mm} \times 23 \text{ mm}} = \boxed{0.191 \text{ mW/mm}^2}$$

3.5 Energy Requirements Analysis

3.5.1 Unregulated Supply Performance

Constant Power Case:

- Regulated power supply providing power for microcontroller and sensors which require input voltage kept in limited range.
- As voltage out of energy source drops from Vworking to Vmin, current increases to keep power through supply constant.

3.5.2 Peak and Average Current Use Cases

We have analyzed the current requirements for each component in our system under both peak and average use conditions. Table 2 summarizes these findings:

Component	**Peak Current (A)**	**Average Current (A)**
Temperature Sensor	0.004	0.00071424
E-ink Display	0.008	8.05558E-05
MCU	0.0393152	0.000115536
PMIC	0.0000009	0.0000009
Load Switch	0.0000201	0.0000201
IO Expander	0.0000015	0.0000015
Total	0.0513377	0.000932832

Table 2: Component Current Requirements

3.5.3 Total Energy Required Between Charging

Given:

- Battery Capacity: 1000 mAh
- Assumed Battery Voltage: 3.7 V (typical for Li-ion)
- Daily Energy Consumption: 343.5409345 J

Calculations:

1. **Convert Battery Capacity to Joules:**

$$\text{Energy (J)} = \text{Capacity (mAh)} \times \text{Voltage (V)} \times 3.6 = 1000 \times 3.7 \times 3.6 = 13,320 \text{ J} \tag{4}$$

2. **Calculate Number of Days:**

$$\text{Days} = \frac{\text{Total battery energy}}{\text{Daily energy consumption}} = \frac{13,320 \text{ J}}{343 \text{ J/day}} \approx 38.83 \text{ days} \tag{5}$$

3. **Convert to Weeks:**

$$38.83 \text{ days} \approx 5.55 \text{ weeks} \tag{6}$$

3.5.4 Recharge Time Calculation

We are using a 1000mAh battery with a 0.5C charging rate:

- Theoretical charging current: 0.5C = 500mA
- BQ25570 PMIC maximum charging current: 230mA (typical)

Since the PMIC can't provide 500mA, it will charge at its maximum rate of 230mA.

$$\text{Estimated charging time} = \frac{\text{Battery Capacity}}{\text{Charging Current}} = \frac{1000\text{mAh}}{230\text{mA}} \approx 4.35 \text{ hours} \tag{7}$$

3.6 Supercapacitor vs Battery Comparison: Mathematical Analysis

We'll compare a 4F Vishay supercapacitor with our 1000mAh Li-ion battery, focusing on quantitative aspects.

3.6.1 Energy Storage

Battery: Energy stored in the battery:

$$E_{\text{battery}} = \text{Capacity} \times \text{Voltage} = 1000 \text{ mAh} \times 3.7 \text{ V} = 3700 \text{ mWh} = 13320 \text{ J} \tag{8}$$

Supercapacitor: Energy stored in the supercapacitor (assuming 2.7V max voltage):

$$E_{\text{supercap}} = \frac{1}{2}CV^2 = \frac{1}{2} \times 4 \text{ F} \times (2.7 \text{ V})^2 = 14.58 \text{ J} = 0.00405 \text{ Wh} \tag{9}$$

3.6.2 Power Density

Battery: Assuming a discharge time of 1 hour at maximum current (1.05A):

$$P_{\text{battery}} = 3.7 \text{ V} \times 1.05 \text{ A} = 3.885 \text{ W} \tag{10}$$

Supercapacitor: Assuming a discharge time of 1 second (typical for supercapacitors):

$$P_{\text{supercap}} = \frac{E_{\text{supercap}}}{\text{time}} = \frac{14.58 \text{ J}}{1 \text{ s}} = 14.58 \text{ W} \tag{11}$$

3.6.3 Current Handling

Battery: Maximum discharge current: 1.05A

Supercapacitor: Maximum current can be calculated using the ESR (Equivalent Series Resistance). Assuming an ESR of 0.1 Ohm:

$$I_{\text{max,supercap}} = \frac{V_{\text{max}}}{ESR} = \frac{2.7 \text{ V}}{0.1 \ \Omega} = 27 \text{ A} \tag{12}$$

3.6.4 Self-Discharge

Battery: Assuming 5% self-discharge per month:

$$E_{\text{loss,battery}} = 0.05 \times 3700 \text{ mWh} = 185 \text{ mWh/month} \tag{13}$$

Supercapacitor: Self-discharge in supercapacitors can be modeled as:

$$V(t) = V_0 e^{-t/\tau} \tag{14}$$

Where $\tau = RC$, R is the leakage resistance (assume 100 kOhm), and C is 4F:

$$\tau = 100 \text{ k}\Omega \times 4 \text{ F} = 400000 \text{ s} \approx 4.63 \text{ days} \tag{15}$$

After one month (2,592,000 seconds):

$$V(1 \text{ month}) = 2.7 \text{ V} \times e^{-2592000/400000} \approx 0.48 \text{ V} \tag{16}$$

Energy remaining:

$$E_{\text{remaining}} = \frac{1}{2} \times 4 \text{ F} \times (0.48 \text{ V})^2 = 0.4608 \text{ J} \tag{17}$$

Energy lost:

$$E_{\text{loss,supercap}} = 14.58 \text{ J} - 0.4608 \text{ J} = 14.1192 \text{ J} = 0.003922 \text{ Wh/month} \tag{18}$$

3.6.5 Charging Time

Battery: With a charging current of 525mA:

$$t_{\text{charge,battery}} = \frac{1000 \text{ mAh}}{525 \text{ mA}} = 1.90 \text{ hours} \tag{19}$$

Supercapacitor: Assuming a charging current of 1A:

$$t_{\text{charge,supercap}} = \frac{C \times V}{I} = \frac{4 \text{ F} \times 2.7 \text{ V}}{1 \text{ A}} = 10.8 \text{ seconds} \tag{20}$$

3.6.6 Conclusion

This mathematical analysis confirms our previous conclusions:

- The battery stores $\frac{3700}{0.00405} \approx 914$ times more energy than the supercapacitor.
- The supercapacitor can deliver $\frac{14.58}{3.885} \approx 3.75$ times more power in short bursts.
- The supercapacitor can handle much higher peak currents (27A vs 1.05A).
- The battery retains charge much better, losing only 5% per month compared to the supercapacitor's 96.8
- The supercapacitor charges $\frac{1.90 \times 3600}{10.8} \approx 633$ times faster than the battery.

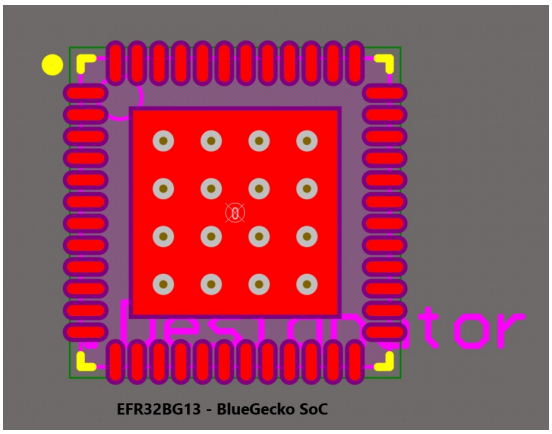
These calculations support using the battery as the main power source,

3.7 PCB Design Progress

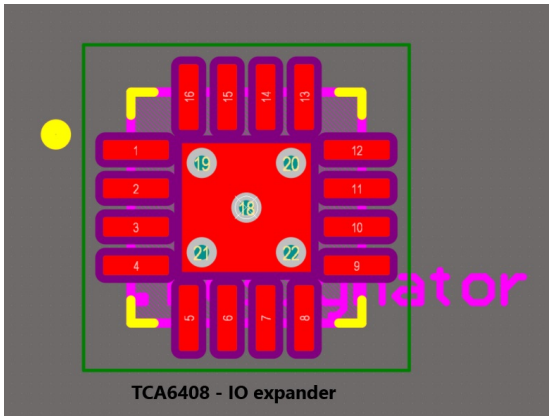
We've completed footprint creation for most components, with the following exceptions:
Solar Panel: Recently added, footprint pending 0603 components: Footprints for resistors and MLCC capacitors still needed

3.7.1 Footprint Generation

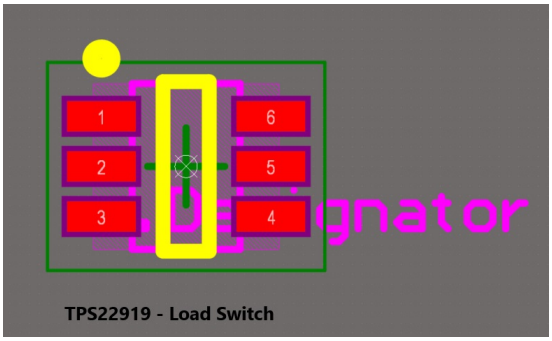
Footprints:



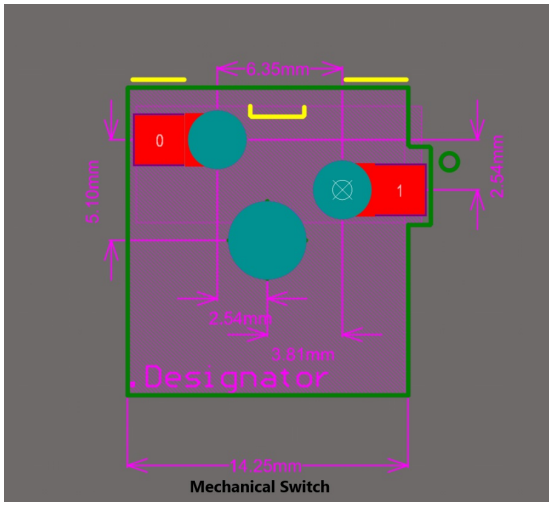
BlueGecko EFR32BG13 QFN48 package



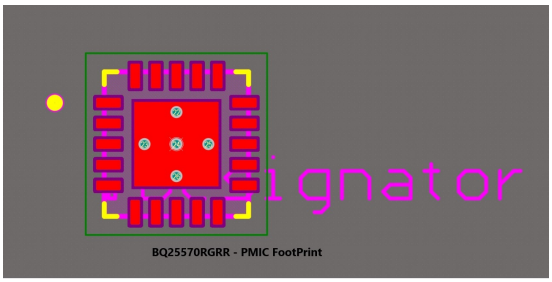
IO expander



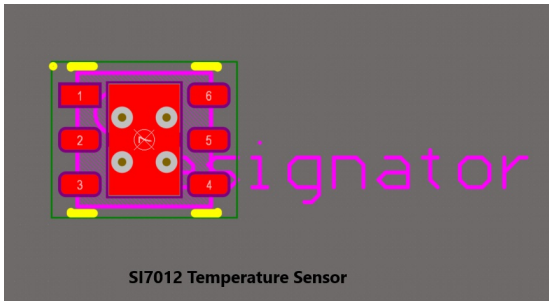
Loadswitch



Mechanical switch



PMIC



Temperature Sensor

Figure 13: Component Footprints

3.7.2 Power Supply Considerations

For components requiring a stable input voltage:
A regulated power supply will be used As the energy source voltage decreases, current will increase to maintain constant power

3.8 Mechanical Design

3.8.1 Dimensions

Left unit: 165mm x 120mm x 35mm
Right unit: 175mm x 120mm x 35mm

3.8.2 Material

Carbon fiber-filled Nylon selected for construction

3.8.3 Operating Conditions

Temperature range: 0°C to 50°C
Voltage range:

Voltage ranges				
	Min	Typ	Max	Units
BlueGecko	1.8		3.8	V
PMIC	0.1		5.1	V
eInk Display	2.2		3.7	V
Load Switch	1.6		5.5	V
Temperature sensor	1.9		3.6	V
IO Expander	1.65		5.5	V

Figure 14: Voltage Range

3.9 Project Progress and Future Plans

3.9.1 Recent Updates

- Added solar panel for energy harvesting
- removed Supercapacitor
- Added links to components
- created Libraries

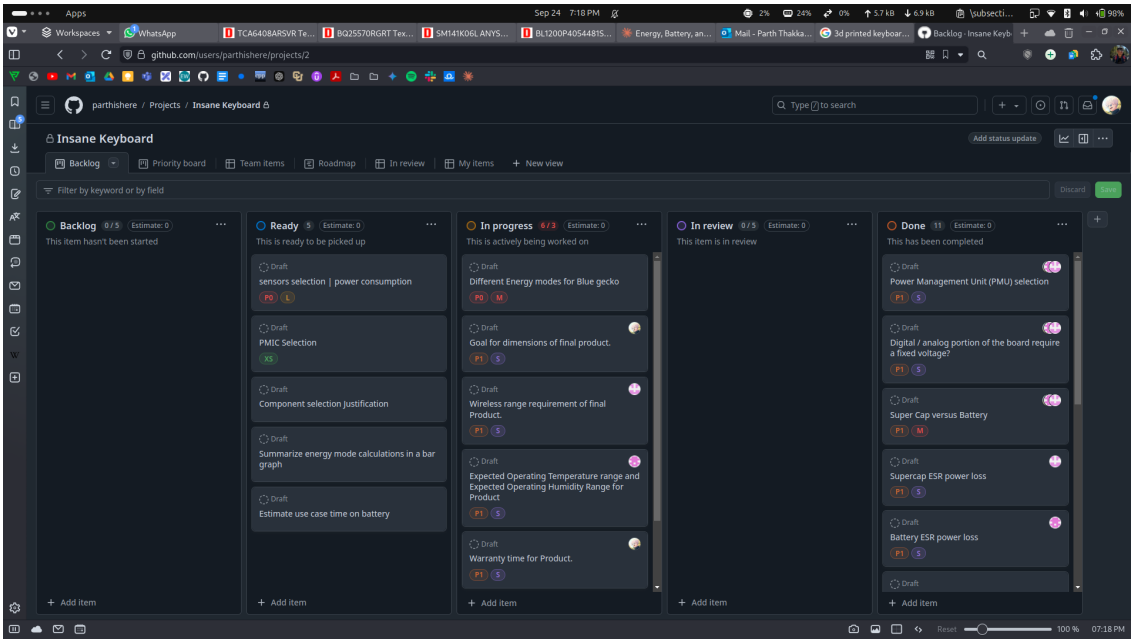


Figure 15: Kanban for update 2

3.9.2 Upcoming Tasks

1. Component Selection:
    - Choose appropriate decoupling capacitors
2. Design Refinement:
    - Optimize key size and spacing for ergonomics
    - Finalize keyboard layout
3. Prototyping Preparation:
    - Design 3D case model using Fusion 360
    - 3D print prototype case

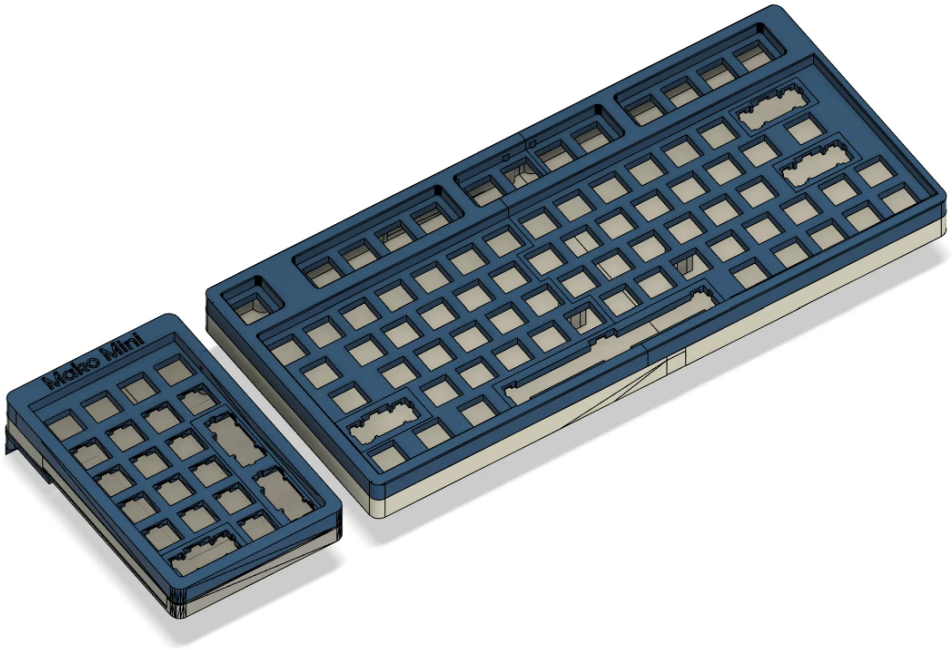


Figure 16: Design a Case and PCB

4 Update 3

4.1 Energy Storage Analysis and Charging Characteristics

4.1.1 Battery Specifications and Usage Profile

Parameter	Value
Internal Resistance (RDS)	0.18 Ω
Capacity	1.2 Ah
C-rate	0.2
Discharge Current at 0.2C	0.24 A
Discharge Time at 0.2C	5 hours
Charge Cycles	300

Table 3: Battery Specifications

Mode	Current (A)	Time Fraction
Start-up	0.00003240	0.0001808911
Active	0.0050311560	0.1488477220
Transmit	0.0390311560	0.007500960
Sleep	0.00003870	0.1488477220
Deep Sleep	0.0000354	0.694622704

Table 4: Current Draw and Time Distribution in Different Modes

4.1.2 Power Consumption Analysis

Based on our usage profile and battery characteristics:

- Average Current: 1.07618 mA
- Sleep Current: 38.7824 µA
- Peak Current: 47.224281 mA
- Weighted Average ESR Loss: 2.73528 µW
- On-mode ESR Loss: 0.208469 µW
- Off-mode ESR Loss: 0.270733 nW
- Peak Current ESR Loss: 401.424 µW

The peak current requirement (47.2 mA) is well below the battery's discharge current capability (240 mA at 0.2C), the battery can comfortably handle peak demands.

4.1.3 Charging Characteristics

We are using the BQ25570 Power Management IC for battery charging. Key charging parameters include:

- Battery Capacity: 1000 mAh
- Theoretical Charging Rate: 0.5C (500 mA)
- BQ25570 PMIC Maximum Charging Current: 230 mA (typical)

Charging Current and Time Calculations Theoretical charging current:

$$I_{charge} = 1000 \text{ mAh} \times 0.5 = 500 \text{ mA} \tag{21}$$

Actual charging current (limited by PMIC):

$$I_{actual} = 230 \text{ mA} \tag{22}$$

Estimated charging time:

$$t_{charge} = \frac{1000 \text{ mAh}}{230 \text{ mA}} \approx 4.35 \text{ hours} \tag{23}$$

4.1.4 Conclusion

- The chosen battery meets the peak current demands of our system.
- While capable of 0.5C charging, the actual charging rate is limited to 230 mA by the BQ25570 PMIC.
- A full charge from a depleted state takes approximately 4.35 hours.
- This charging configuration prioritizes battery longevity and safety over fast charging speeds, aligning with our low-power design goals.

4.2 Worst-Case Communication Bus Timing

Parameter	**IO Expander**	**Temperature Sensor**	**Blue Gecko**	**Worst Case**
SCL frequency	400 kHz	400 kHz	400 kHz	400 kHz
SCL high	0.6 µs	0.6 µs	0.6 µs	0.6 µs
SCL low	1.3 µs	1.3 µs	1.3 µs	1.3 µs
Spike time / Suppressed pulse width	50 ns	50 ns	-	50 ns
SDA setup	100 ns	0.1 µs	100 ns	0.1 µs
SDA hold	0 ns	0.1 µs	100-900 ns	0.1-0.9 µs
Input rise time	300 ns	-	-	300 ns
Input fall time	300 ns	-	-	300 ns
Output fall time (10pF - 400pF load)	300 ns	-	-	300 ns
Bus free time	1.3 µs	1.3 µs	1.3 µs	1.3 µs
Start setup time	0.6 µs	0.6 µs	0.6 µs	0.6 µs
Start hold time	0.6 µs	0.6 µs	0.6 µs	0.6 µs
Stop setup time	0.6 µs	0.6 µs	-	0.6 µs
Data valid time	1 µs	0.9 µs	-	1 µs
ACK valid time	1 µs	0.9 µs	-	1 µs

Table 5: I2C Fast Mode (400 kHz) Bus Timings

Parameter	IO Expander	Blue Gecko	Worst Case
SCL frequency	100 kHz	100 kHz	100 kHz
SCL high	4 µs	4 µs	4 µs
SCL low	4.7 µs	4.7 µs	4.7 µs
Spike time	50 ns	-	0 ns
SDA setup	250 ns	250 µs	250 µs
SDA hold	0 ns	100 µs	100 µs
Input rise time	1000 ns	-	-
Input fall time	300 ns	-	-
Output fall time (10pF - 400pF load)	300 ns	-	-
Bus free time	4.7 µs	4.7 µs	4.7 µs
Start setup time	4.7 µs	4 µs	4.7 µs
Start hold time	4 µs	4 µs	4 µs
Stop setup time	4 µs	-	4 µs
Data valid time	1 µs	-	1 µs
ACK valid time	1 µs	-	1 µs

Table 6: I2C Standard Mode (100 kHz) Bus Timings

Table 7: SPI Communication Parameters

Parameter	E-Ink Display				Blue Gecko			
	Condition	Min	Typ	Max	Condition	Min	Typ	Max
SCL freq	Write	-	-	20 MHz	Write	-	-	31.25 MHz
	Read	-	-	2.5 MHz	Read	6*tHFPERCLK		
CS_setup	Write	20 ns	-	-	Write	12.5 ns	-	-
	Read	100 ns	-	-	Read	4 ns	-	70 ns
CS_hold	Write	20 ns	-	-	Write	12.5 ns	-	-
	Read	50 ns	-	-	Read	4 ns	-	70 ns
CS_high	Both	100 ns	-	-	Both	-		
SCL_high	Write	25 ns	-	-	Write	2*tHFPERCLK		
	Read	180 ns	-	-	Read	2.5*tHFPERCLK		
SCL_low	Write	25 ns	-	-	Write	2*tHFPERCLK		
	Read	180 ns	-	-	Read	2.5*tHFPERCLK		
SDA_setup	Write	10 ns	-	-	Write	9 ns	-	-
	Read	-	50 ns	-	Read	12.5 ns	-	-
SDA_hold	Write	40 ns	-	-	Write	9 ns	-	-
	Read	-	0 ns	-	Read	13 ns	-	-

4.2.1 SPI Bus Timing

4.3 High-Risk Development Items

4.3.1 Hardware Challenges

- **PCB Design for Key Matrix:** Creating an accurate PCB layout for the keyboard matrix is significant challenge, as our design is based only on datasheet specifications without physical prototyping.

- **Ergonomic Key Spacing:** Determining the ergonomic spacing between keys is importanant. We are considering a 19.05 mm center-to-center distance, but the ergonomic efficacy of this choice remains uncertain.
- **Key Mounting Mechanism:** Developing a precise gasket or mounting system for the keys, whether PCB-based or 3D-printed, requires exact dimensional accuracy to ensure proper key fit and function.

4.3.2 Software Challenges

- **E-Ink Display Driver:** Development of a custom driver for the E-ink display is expected to be time-intensive.
- **Multi-Host Connectivity:** Implementing firmware to support seamless multi-host connections presents significant complexity.

4.4 Mitigation Plans

To address the identified risks and challenges, we have developed the following mitigation strategies:

4.4.1 Prototyping and Manufacturing

- We have access to the ITLL 3D printing facility,
  - Prototype cases and gaskets using PLA
  - Iterate designs rapidly and cost-efficiently.
  - Produce final components using high-quality materials once designs are finalized.

4.4.2 Hardware Development

- Initially utilize development boards, modules, and breadboards for proof-of-concept.
- Test and validate circuit functionality before proceeding to schematic design.
- Implement incremental improvements based on prototype performance.

4.4.3 Software Development

- Develop initial drivers for more commonly used platforms (e.g., STM32 or Arduino).
- Validate display functionality on these platforms.
- Port the verified code to our target EFR32BG13 platform.

4.5 Update

Despite these challenges, our project is progressing as scheduled. Key developments include:

- Firmware development is on track.
- Hardware integration is proceeding as planned.
- Schematic design is nearing completion.

4.6 gnatt chart

Figure 17 presents our updated Gantt chart, reflecting the current project timeline and milestones.

GITHUB

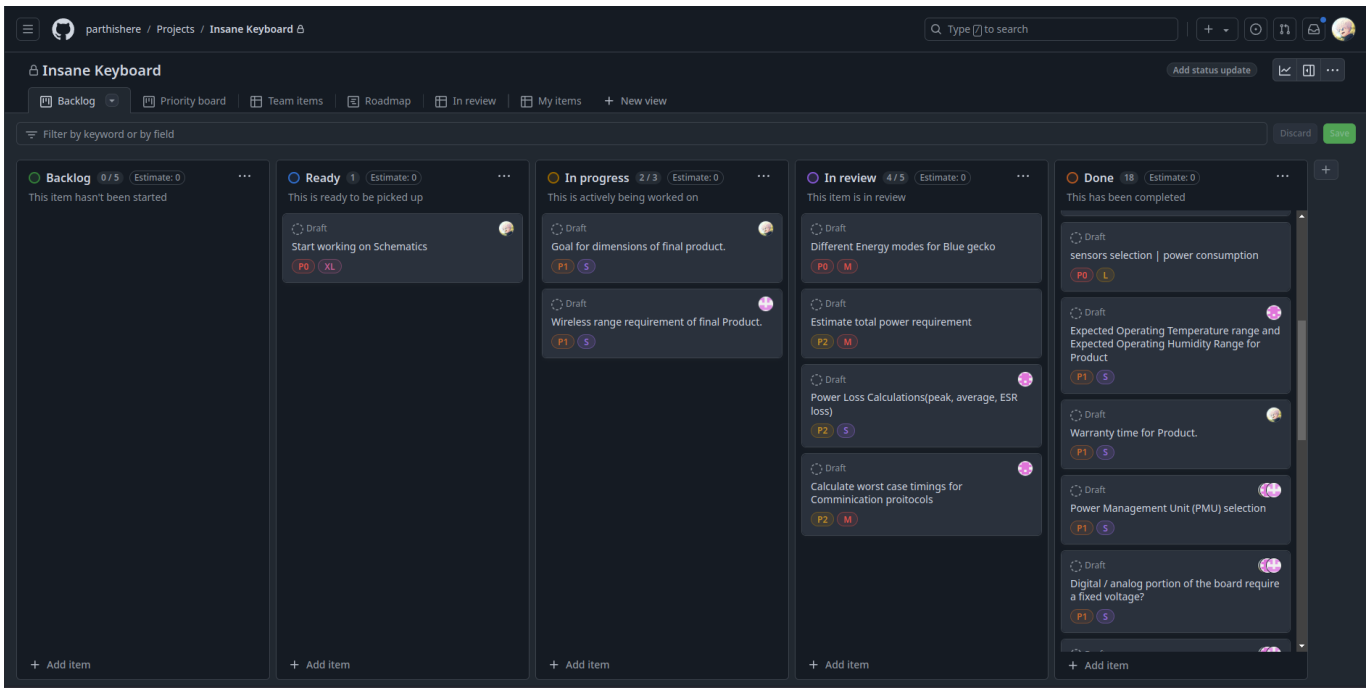


Figure 17: Updated Project Gantt Chart