ECEN 5833 Low Power Embedded System Design Technique

Project Proposal

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 7^{th} September 2024

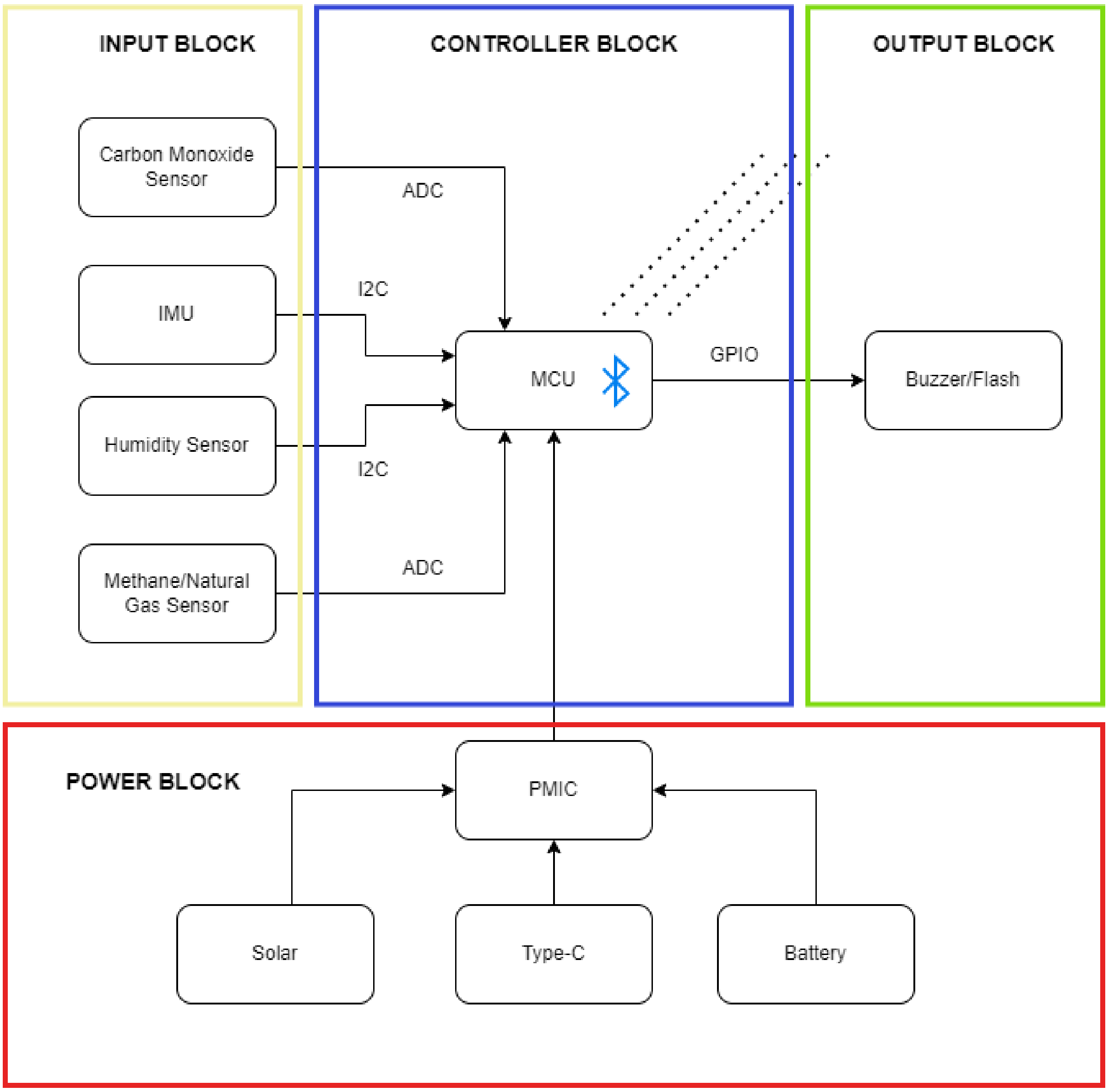
Introduction

The project focuses on prototyping a low power portable safety device to help people who work in close quarters in an outdoor area to harvest natural resources such as mineral mining, tree harvesting, and oil and gas extraction. This safety device would ideally be attached to a person on their vest/pant pockets and everyone working together in a site would be using this device. The safety device of every person shall be connected together in a Bluetooth mesh which would help the site supervisor to ensure that the maximum distance between two workers does not go beyond a threshold set by the supervisor. These safety devices will also house a variety of gas sensors to ensure that no worker is exposed to any toxic/flammable gases, and a IMU sensor for fall detection. An onboard buzzer and flasher are implemented to alert the users of any danger. The sensor information from each worker shall be available for viewing on the supervisor's mobile phone which shall be connected to the Bluetooth mesh network as well.

The final product should have the following capabilities:

- Must be able to run for about 24 hours on battery which can be charged using **Solar Power** or from an external 5V power supply source via type-c USB.
- The device must be able to connect to multiple of same devices providing virtual tethering between a group of miners using **BLE** and alter the group if a miner is moving away.
- Must be able to sense the environment to provide Air Quality information and fall detection.
- The over-all circuit should be of dimension 3.5x3.5 square inches.

Block Diagram



Protocols Used

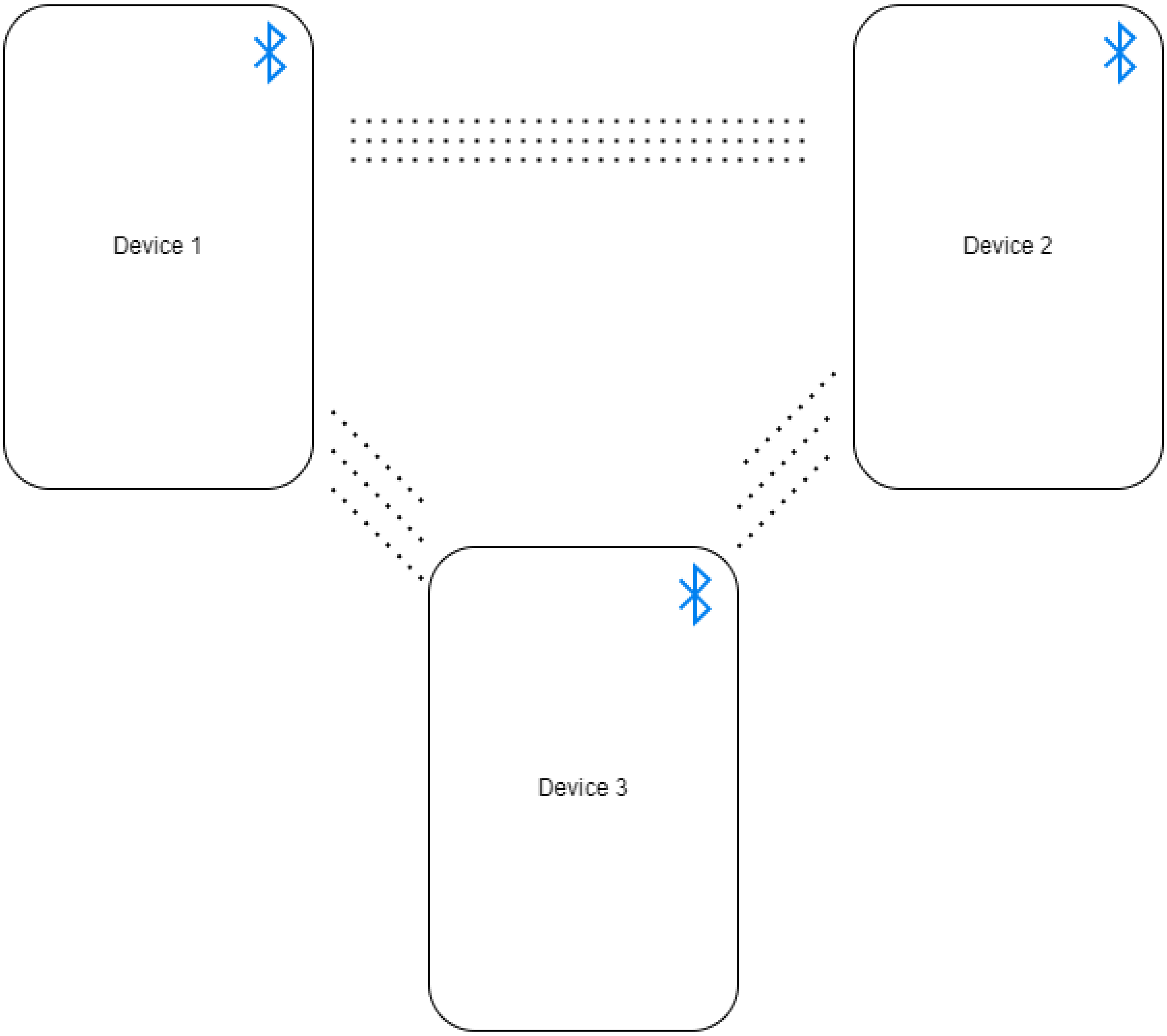
BLE: Bluetooth Low energy will be used for wireless communication between the devices.

ADC: Analog to Digital Converter will be used interface Gas sensors.

I2C: IMU and Humidity sensors will be interfaced using I2C.

GPIO: Buzzer and a flash indicator will be interfaced using GPIOs.

Functional Diagram



Key Components

Microcontroller: Blue Gecko- EFR32BG

Radio Technology: Bluetooth Low Energy (Blue Gecko)

Inertial Measurement Unit: MPU6050

Carbon Monoxide Sensor: MQ-7

Methane & Natural Gas Sensor: MQ-9

Humidity Sensor: AM2320

Buzzer: Piezo Buzzer

Energy Harvesting: 0.5W Solar Panels

ECEN 5833 Low Power Embedded System Design Technique

Week 1 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 14th September 2024

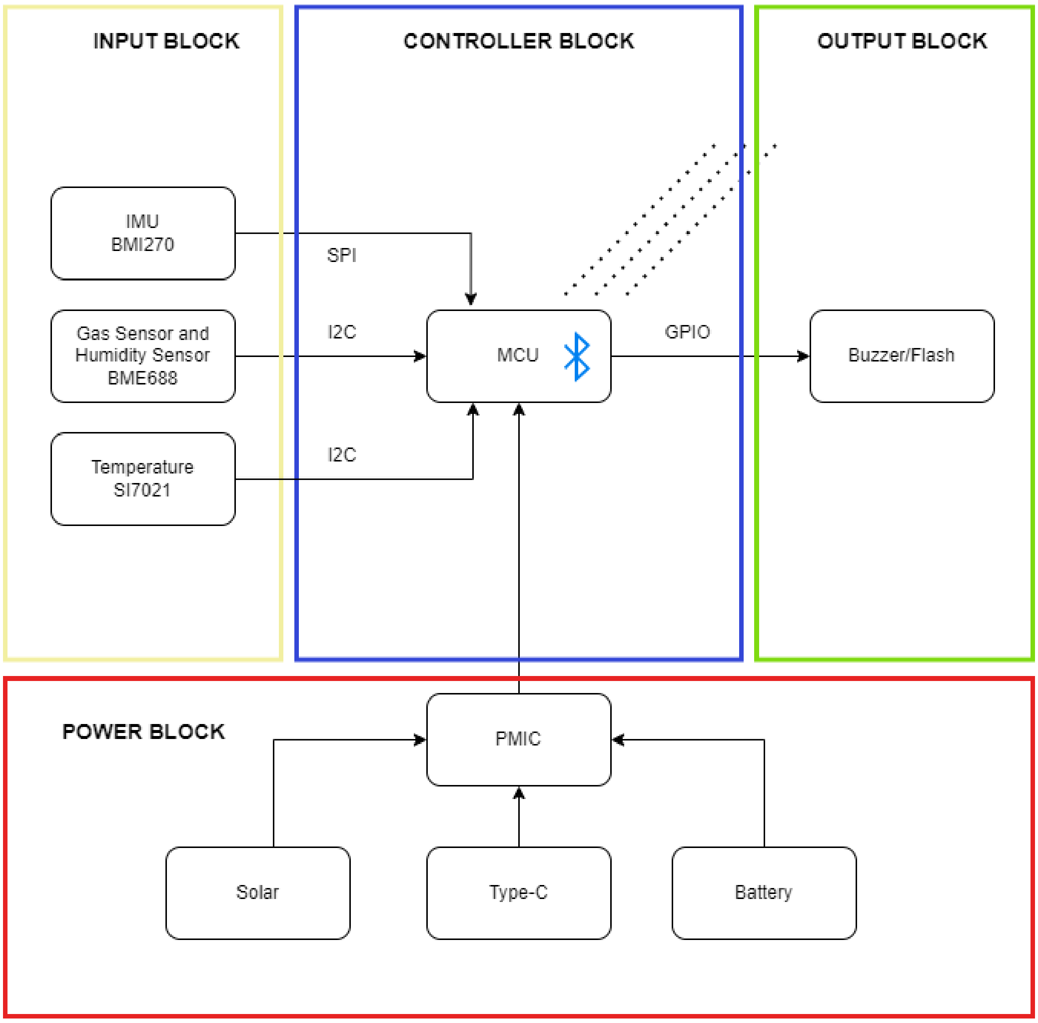
Previous review questions answered

- **How will you determine if a miner falls away from the group?Will you use the RSSI value or loss of connection?**
  We are planning to use RSSI value but, based on implementation difficulty we may switch to loss of connection method. IMU will be used to detect fall the BMI270 sensor has capabilities to detect type of motion.
- **Is this a mesh network or a sequential network?**
  We will be using Bluetooth mesh network.
- **May be able to find a better temperature and humidity sensor than the AM2320.**
  Yes, we found a better gas sensor BME688 which can be used for humidity sensing and Si7021 for temperature sensing.

Summary:

During week-1 we shortlisted the required components and got their part numbers. Using this information, we designed a use case timeline of how to schedule the tasks in period of the device's operation.

Updated Block Diagram with communication protocol:



There were some changes to the input block of the project due to availability and power requirements of the device.

BMI270: This is better than obsolete mpu6050 with 6 axis gyro and accelerometer, and optional magnetometer.

BME688: This sensor has various sensing capability to measure gas including Carbon Monoxide, Carbon Dioxide, and Hydrogen Sulphide. It's capable of measuring humidity and temperature.

SI7021: This is a temperature sensor chosen to read temperature parallelly instead of relying on BME688 which reduces the computation and on time of the BME688 sensor.

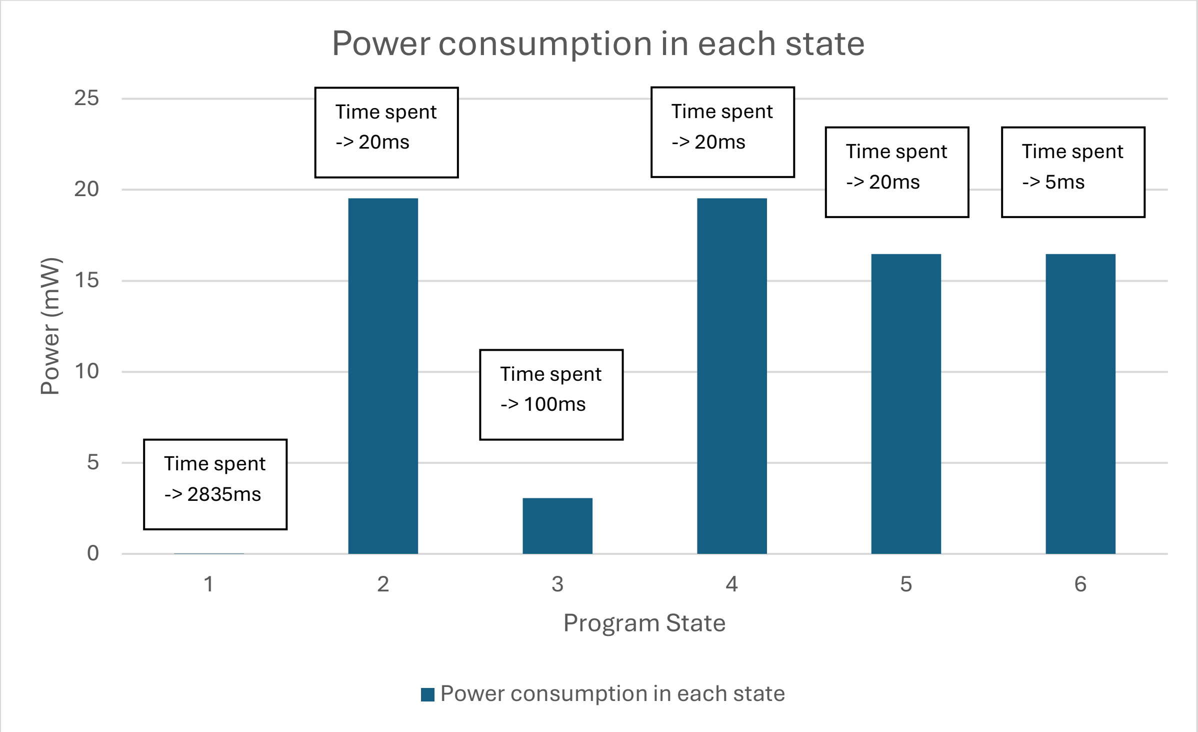
DigiKey Partnumbers:

Important Components	Use case	Digikey part number	Cost per component ($)
Blue Gecko	Microcontroller	336-EFR32BG13P632F512GM32-DRCT-ND	6.93
BMI270	Fall detection (IMU)	828-1091-1-ND	5.88
BME688	Gas sensor (CO, CO2 and H2S), Humidity	828-BME688CT-ND	11.83
Si7021	Temperature	336-4349-1-ND	10.91
Piezo Buzzer	Alerting	445-2525-1-ND	0.56

Use Case stages:

State	**Time spent in state (ms)**	**Task to achieve in state**	**Microcontroller state**
State 1: Deep Sleep	2835	Sleep	EM3
State 2: Startup	20	Start the MCU, supply power for all sensors, and start init sequence for the sensors if applicable	EM0
State 3: Sensor setup	100	Wait for the sensors to finish init sequence	EM2
State 4: Sensing	20	Get data from all sensors and turn them off	EM0
State 5: Computation	20	Determine if all sensor values are within thresholds; Determine distance using RSSI of nearby devices	EM0
State 6: Transmission	5	Transmit/receive data to/from other BLE devices	EM0

Power Consumption for the Use case state model:



Tasks for next week:

- Create a Gant Chart
- Select a PMIC
- Finalize use case timeline and optimize the timing and power management.

ECEN 5833 Low Power Embedded System Design Technique

Week 2 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 23rd September 2024

Previous week update review comments addressed

- **State plan excel sheet:** link
- **Power consumption calculation:** link
- **Power consumption chart:** link

Summary

- Changed the use case timeline from 3s to 500ms reduced the deep sleep time increasing the monitoring frequency of the parameters.
- Performed the power analysis and recalculated the power consumption for new use case timeline.
- Selected the PMIC based on the power requirements, battery and energy harvesting chosen.
- Selected the parts for power source components.
- Created Gantt chart: link
- Started with creating symbols and footprints (Altium Library).
- Update the proposed 24hr change cycle to 72 – 80hr cycle, due to available higher battery capacity.

Updated use case stages

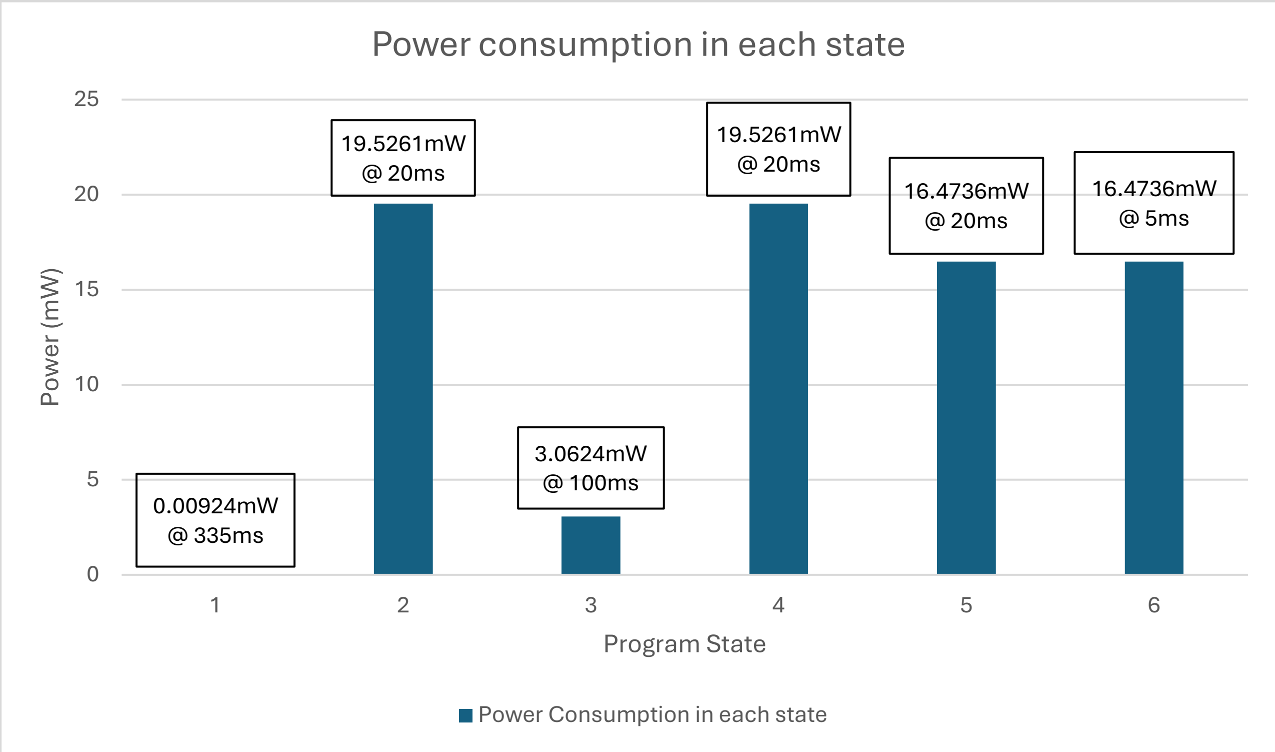
State	**Time spent in state (ms)**	**Task to achieve in state**	**Microcontroller state**
State 1: Deep Sleep	335	Sleep	EM3
State 2: Startup	20	Start the MCU, supply power for all sensors, and start init sequence for the sensors if applicable	EM0
State 3: Sensor setup	100	Wait for the sensors to finish init sequence	EM2
State 4: Sensing	20	Get data from all sensors and turn them off	EM0
State 5: Computation	20	Determine if all sensor values are within thresholds; Determine distance using RSSI of nearby devices	EM0
State 6: Transmission	5	Transmit/receive data to/from other BLE devices	EM0

The following values are for 24hr usage cycle.

Energy (Joule)	2.60E+02	J
Voltage	3.3	V

Charge	23.84	mAh

Updated power consumption chart



Updated DigiKey Part numbers:

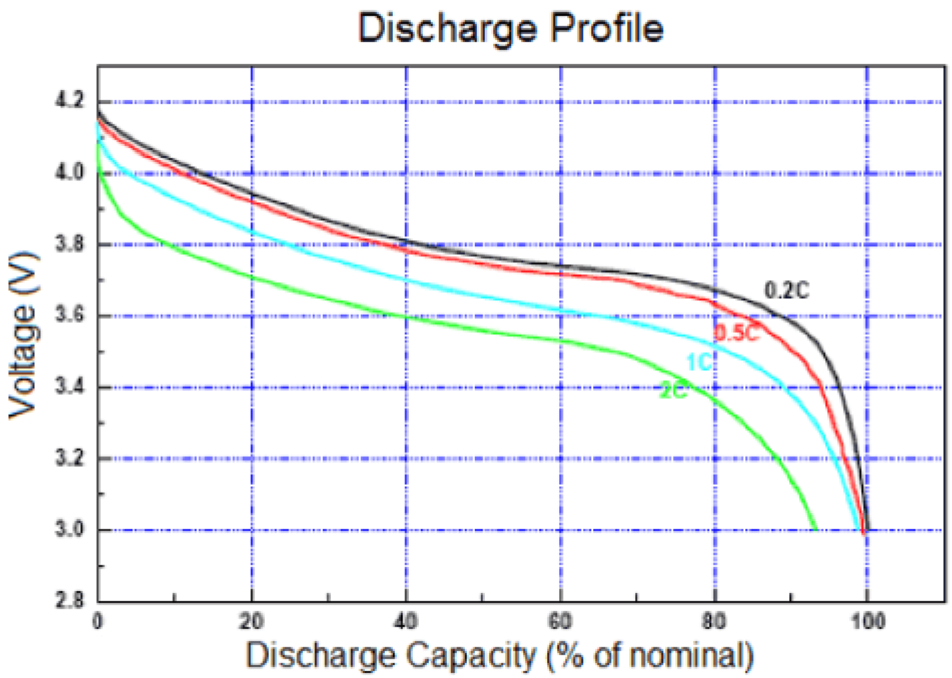
Components	Use case	Digikey part number	Cost per component ($)
Blue Gecko	Microcontroller	336-EFR32BG13P732F512GM32-DRCT-ND	6.93
BMI270	Fall detection (IMU)	828-1091-1-ND	5.88
BME688	Gas sensor (CO, CO2 and H2S), Humidity	828-BME688CT-ND	11.83
Si7021	Temperature	336-4349-1-ND	10.91
Piezo Buzzer	Alerting	445-2525-1-ND	0.56
BQ25570	PMIC	296-39934-1-ND	7.74
BATTERY LITH POLY 3.7V 350MAH	Battery	1528-4237-ND	5.95
KXOB141K06F-TR	Solar energy harvesting 4.15V 184.3mW	2994-KXOB141K06F-TRCT-ND	7.30

Part selection

PMIC – BQ25570

- Supports energy harvesting, battery charging and regulating the power supply
- System requires the operating voltage at 3.3V, but the battery provides greater than the 3.3 which can be maintained by using buck converter in PMIC
- Continuous Energy Harvesting from VIN as low as 100 mV which works for our application
- High Efficiency up to 93% for buck converter
- Supports Peak Output Current up to 110 mA buck converter

Battery – 350mAh 3.7V



- The battery chosen is rated at 0.2C for discharge, as observed in the above curve the discharge voltage remains above 3.3V until 80-90% charge is consumed.
- The battery discharge will be limited to 80-90 % to maintain battery health.
- The battery has a charging cut-off at 4.2V around which the PIMC will be designed to charge it to a 90% of full capacity.
- The capacity is 350mAh which will be discharged up to 80-90% of the capacity and charged to 90% percent of the full capacity. This provides a 72-80hr charge cycle while maintaining the battery health.

Solar Panel – KXOB141K06F-TR

- It is designed to be very compact (23mm x 42mm), making it ideal for small PCB boards

- Perform better under low-light or indoor lighting conditions compared to traditional silicon-based cells. So can be used in indoors or in environments where sunlight may be limited or diffused.
- It can be paired with an energy harvesting IC chosen (lBQ25570) to efficiently convert and store the harvested energy.
- Solar cell efficiency of 25%

Power Management Unit (PMU) selection checklist

- Will the circuitry run on an unregulated supply?
  No, because the system requires constant operating voltage of 3.3v
- Verify that all devices have wide voltage inputs
  - What is the minimum Vinmax of the ICs?
    3.6V
  - What is the maximum Vinmin of the ICs?
    1.8V
- What is the planned range of the unregulated voltage?
  Battery has an output voltage range of 3.6v – 4.2v for use case for which PMIC will be used to regulate it
- Does the digital / analog portion of the board require a fixed voltage?
  Yes, system will operate at 3.3V
- Will the Energy Source voltage always be above the circuitry voltage?
  Yes, the battery will always above the circuit voltage
- Use Buck Converter:
  Yes, from battery to power rail
- Will the Energy Source Voltage always be below the circuitry voltage?
  No
- Use Boost Charger
  Yes, from Solar panel to battery charger
- Will the Energy Source Voltage be above and below the circuitry voltage?
  No
- Use Buck/Boost Converter
  No

Determine if we need a super/bulk capacitor

- No, because we don't have any feature using rapid energy.
- Also, we need long time power because our use case requires about 72-80hrs.

Wireless range required

- Estimated about 5-10 m based on the Bluetooth and antenna design (not yet calculated).

Expected Operating Temperature range for product. (Important in energy storage selection.)

- Battery makes the operating temperature of product to 0 to 45℃.
- Operating temperature of battery 0 to 45℃ for charge and discharge battery -20 to 60℃.
- Battery storage temperature for 1 month is -20 to 45 ℃ and for 6 months it is -10 to 35 ℃ temperature.

Expected Operating humidity range for product.

90%-95RH%

Expected warranty for Product.

- Warranty of Battery is given to be 1 year in data sheet. So, we estimate warranty of product to be 1 year.

Dimension of product

- Estimated to be 90mm X 90mm (will be finalized after board design).

Tasks for next week:

- Select supporting components for individual ICs and components.
- Populate the component library.
- We are on schedule as per our planned timeline.

ECEN 5833 Low Power Embedded System Design Technique

Week 3 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 28th September 2024

Previous week update review comments addressed

- **What is your expected recharge time and how many cycles do you expect to utilize?**
  - The battery is charged using the Constant Current Constant Voltage method, with a charge voltage of 4.2V and a charge rate of 0.5C. The charge continues at constant current until the voltage reaches 4.2V
  - At 0.5C, the battery will be charged at half of its rated capacity per hour. For a battery with a capacity of 332mAh, charging at 0.5C means charging at 166mA
  - Initial charging time upto 80%:
    - Time = 332 mAh × 0.8/166 mA = 1.6 hours
    - The remaining 20% is charged during the constant voltage phase, which generally takes around 30 minutes to 1 hour. Therefore, the total recharge time is expected to be 2−2.5 hours
  - Expected number of cycles is 300 before the battery capacity drops below 80%.

- **You mention your expected warranty is dictated from the battery warranty of 1 year, but what calculations justify this for your product?**
  - For the lithium ion-battery our product uses can undergo >=300 charge cycles before its capacity drops to 80% when discharged to 3V at 0.2C rate
  - The cycle life is a major indicator for setting the warranty period since it reflects how many full charge/discharge cycles the battery can undergo before significant degradation.
  - As our use charge cycle is 1 cycle/3 per day
  - 300 cycles/ (1 cycle/3 per day) = 900days
  - As per system use case, we expect battery life to be 900 days if the temperature is 25 degree C
  - If the system gets into exposure to high temperatures like 55 degrees Celsius for 2 hours or low temperatures like -10 degree Celsius for 16-24 hours, the battery retains a significant amount of capacity because mining place could be in different temperature. This indicates that the battery can function well in a range of operating conditions, making it reliable for at least 1 year even when exposed to temperature extremes.

Summary

- Completed the component creation of all major components
- All tasks are on track
- Started with dev kit
- PMIC analysis and Battery analysis done
- Updated Gantt chart: link

Communication bus timing analysis

- We will be using this information when interfacing the sensor with the microcontroller and to verify the signal integrity in the bus while assembling the PCB board
- Component assumptions
  - BMI270
    - V_{DDIO} >=1.62V
  - BME688
    - V_{DDIO} = 1.62v

<u>SPI</u>

Device	T_{CSB_SETUP}	T_{CSB_HOLD}	T_{SCKL}	T_{SCKH}	T_{SDI_SETUP}	T_{SDI_HOLD}	T_{SDO_OD}
BMI270	40 ns	40 ns	45 ns	45 ns	20 ns	20 ns	30 ns

<u>I2C – Standard Mode</u>

Device	T_{SCL_LOW}	T_{SCL_HIGH}	T_{SDA_SETUP}	T_{SDA_HOLD}	T_{SDA_VALID}	T_{STOP_SETUP}	T_{START_SETUP}	T_{START_HOLD}	T_{BUF}
BME688	160 nS	Not available in Datasheet	160 nS	80 nS - C_b <= 100pF 90 nS - C_b <= 400pF	Not available in Datasheet	Not available in Datasheet	Not available in Datasheet	Not available in Datasheet	Not available in Datasheet
Si7021	1.3 uS	0.6 uS	100 nS	100 nS	0.9 uS	0.6 uS	0.6 uS	0.6 uS	1.3 us

Device	T_{STOP_SETUP}	T_{START_SETUP}	T_{START_HOLD}
BME688	Not available in Datasheet	Not available in Datasheet	Not available in Datasheet
Si7021	0.6 uS	0.6 uS	0.6 uS

When will you be using this communication information?
We will be using this information when interfacing the sensor with the microcontroller and to verify the signal integrity in the bus while assembling the PCB board

Battery and PMIC Analysis

- **What is the C-rate of your specified battery?**
    - 0.2C
- **What is the peak discharge rate out of the battery in your application?**
    - The peak discharge rate for the battery would be 525mA.
    - For our application, the peak discharge rate would be 4.992mA for 25mS.
- **Based on your lithium battery discharge curve, what is the lowest nominal voltage?**
    - The lowest nominal voltage would be around 3.7V
- **What will be the battery cut-off voltage of your circuit?**
    - We would be setting the cut-off voltage at 3.75V, Although the discharge cut-off voltage built-in the battery is 3.0V
- **Will this nominal voltage work using a buck only solution or is a buck-boost required?**
  We assume that this nominal voltage will work for us since all of our components operate at 3.3V**.**
- **Does your PMU support a low battery discharge cut-off voltage?**
  Yes, Our PMIC does supports a low battery discharge cut-off voltage which is permanently set to 1.95V.
- **If so, what is the cut-off voltage that you will program or set?**
    - Since the nominal discharge voltage of the battery is 3.7V, and the discharge cut-off at the battery side is 3.0V, the internally set voltage would not be of any use for this application.
    - On reading the datasheet, we found that we can set the output of VBAT_OK pin from the PMU by selecting appropriate resistors for ROK1, ROK2, and ROK3. Hence, we are discussing the possibility of using this design element to make the system enter EM4 energy mode or even shutdown if the battery voltage drops below 3.8V considering system losses and design tolerances**.**
- **sizing storage based on recharge requirements**
  The device predicted average current consumption is about 9.2mA and our required operation hours are around 35 hours hence we chose a battery of around 350mAh which also fits in the required dimensions of the prototype
- **tolerance issues**
  Battery, sensors and resistors [1% tolerance] are operable in the range of -20C to +65C which is in the operating range required for our protype.
- **Calculate the maximum discharge**
  Average current required is 9.2mA and battery capacity is about 350mAh so it is predicted to work for 35 hrs if operated continuously without sleep.

Risk Management

What are the high-risk development items? And, what are the planned mitigation plans?

- System will be using Bluetooth mesh network when ever the miner falls away from the group, we plan to use RSSI value to determine it. This can be mitigated by switching into loss of connection method.
- System expects to collect data from all three sensors together at same time. If this doesn't work, we try to mitigate this problem by collecting data from sensors at different intervals.

Next week planned work

- Getting component library reviewed from SA
- Planned to work on Schematic
- Also getting ready for sensor demonstration.
- We are on schedule as per our planned timeline.

ECEN 5833 Low Power Embedded System Design Technique

Week 4 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 08th October 2024

Summary

- We had review with SA for component library and updated the library accordingly.
- We completed the schematics for our project.
- Completed the bulk capacitance simulation.
- Installed simplicity studio IDE and verified board working by uploading the sample codes.
- Evaluated usage of zephyr OS with EFR32BG13 dev kit.
- Updated Gantt chart: [link]

What features, components, will need to be added to your schematic to enable programming of your micro controller or SoC?

- We are planning on using a 10-pin ARM debugger port to program the EFM32 board. This port will be connected by using the connection diagram given in an0002.1 Pg. 22 as reference.

Describe the process of how you will compile, connect, and download code to your board design, the target

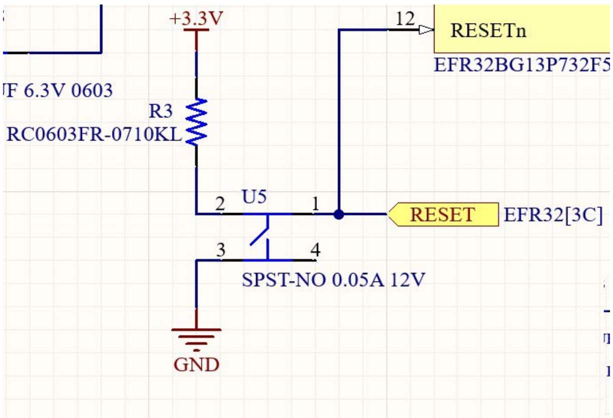
- The firmware shall be written using Simplicity Studio IDE v5 on a computer and shall be compiled using the same tool.
- The BRD4104A dev kit for the EFR32BG13 MCU shall be connected to the computer using a Micro USB B cable. This dev kit shall NOT have the radio board connected.
- The debug mode shall be set to "DEBUG OUT" mode in the IDE application settings.
- The ARM debugger 20pin port onboard the dev kit shall be connected to the ARM debugger 10pin port on the fabricated PCB (aka target board) which is connected to the MCU
- On Board power supply shall be disconnected and external power supply shall be provided from EFR32BG13 dev kit
- The code shall be flashed to the MCU by using the debug option in the IDE.
- External Power supply shall be disconnected and on board supply will be connected.

Which signals will have test points?

- We have added TPs for buses, interrupts and important nets for debugging

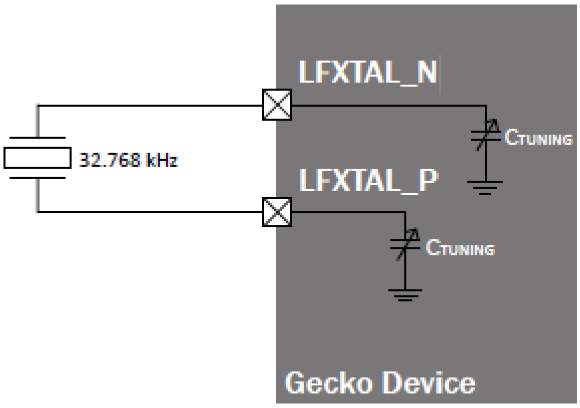
Reset Circuit Description

- Reset circuit is simple design where the Reset in is connected to 10Kohm pull-up resistor and button to short it to ground. As shown below:

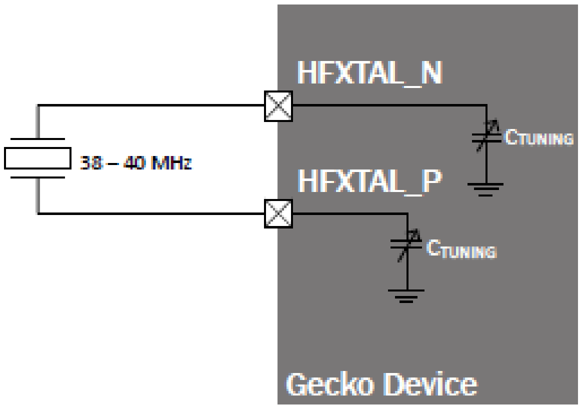


Clock Generation Description

- External clk is required to be generated and produced to the EFR MCU for both low/high frequency using
  - crystal/ceramic resonator – low and high freq.
  - an external clk source – low freq.
  - external sine/square wave – High freq.
- Low Frequency
  - A resonator of 32.768KHz is connected to the LFXTAL_N and LFXTAL_P pins of the EFR32BG13 board
  - The tuning caps are on-chip and can be configured via on-device registers (Ref doc “an0002.1” Pg. 23 for more information)



- High Frequency
  - A resonator of 38-40MHz is connected to the HFXTAL_N and HFXTAL_P pins of the EFR32BG13 board



Provide alternative energy source other than the energy harvester.

- We are using the micro-USB as our alternative source of energy and is compatible with system PMIC

Jump Start method to charge the energy storage element in the event that the Energy Harvester circuity is not functional or takes too long.

- Since we are using the Micro-USB as our alternative source of energy that can be used for jump starting of our circuit.

What is the maximum charging current allowed by the PMU circuitry?

- Peak current output that the PMIC can handle is 110mA.

What is the maximum charging current allowed by the energy storage unit specs?
What will the maximum current of the jump start power source be set to?
Where will the jump start power and ground signals connect to?

- For safe and long-lasting operation of the battery it should be charged at constant current of 0.5C and a constant voltage 4.15V. So, we can limit the jump start current to 1C and bring it down to 0.5C to preserve the battery life.

5	Standard Charge	Constant Current 0.5C Constant Voltage 4.2V 0.01 C cut-off	Charge Time : Approx 4.0h

- We'll be using a USB Type-C, which will jump start charging when the battery is dead. The power and ground pin will be connected to the PMIC to handle the jump start and charging.

Ensuring that there is enough energy / current to program the flash of the MCU

How much current will the programming of the MCU flash require?

Ref: efr32bg13 datasheet, Pg. 88

- Supply voltage req for flash erase and write = 1.62V – 3.6V
- Erase current = 2.0 mA per page
- Program current = 3.5mA

How much current will the energy storage element and the PMU be able to provide?

- Our battery will provide 525mA current and PMU about 110mA

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

- The power output from the PMU shall be given to the digital / MCU portion of the board via external jumper pins so that the PMU can be disconnected, and an external power supply can be connected to the MCU to provide power during programming/debugging.

Next week planned work

- Getting Schematic reviewed from SA
- Planned to work on component placement and Layout design
- Also getting ready for sensor demonstration (configuring sensors)
- We are on schedule as per our planned timeline.

ECEN 5833 Low Power Embedded System Design Technique

Week 5 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 12th October 2024

Previous week update review comments addressed

- **test points, the purpose of this question is to identify which nets in your schematic will have a test point**
  SPI MISO, SPI MOSI, SPI_SCK, SPI_CS_BMI270,I2C_SCL, I2C_SDA, VSTOR, VBAT_OK, 3V3, V_Solar
- **In regards to your reset circuit, consider adding a debounce capacitor (RC) filter. This will help filter out high frequency noise when pressing the button.**
  Yes, we have updated in our recent schematic after SA review.
- **When selecting out your crystals, make sure the tuning caps support the crystal's load capacitance.**
  Yes, we have chosen the 38.4MHz crystals according to it.

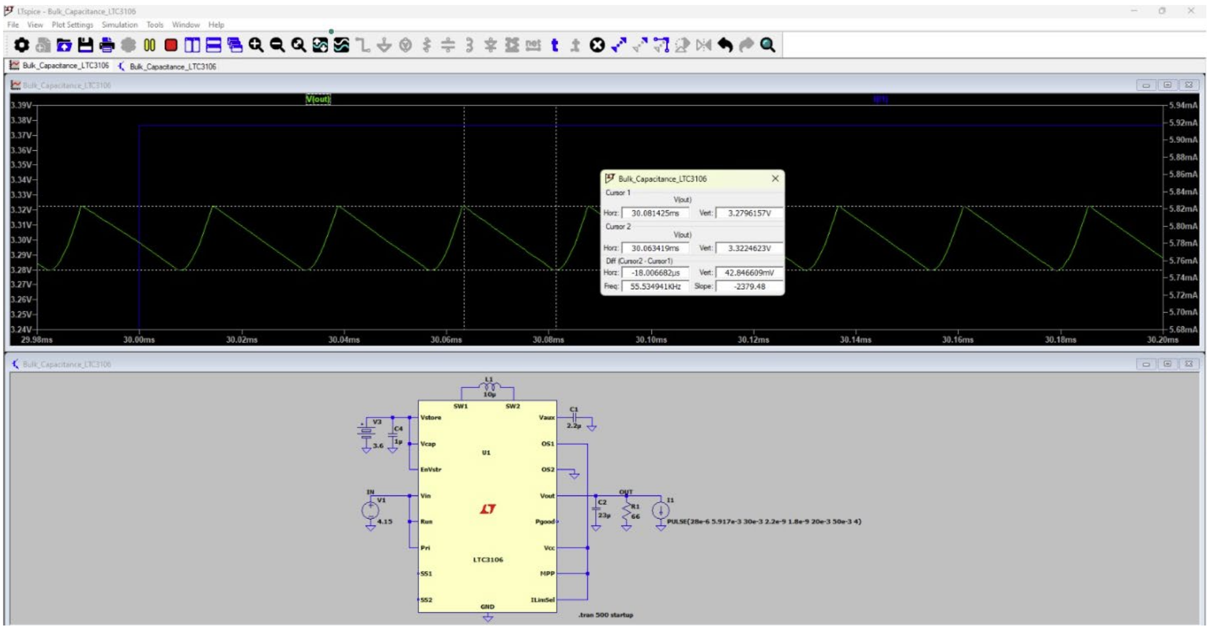
Summary

- We had a review with SA for schematic and updated the schematic accordingly.
- After updating, completed component placement
- Working on the sensor interfaces.
- Setting up everything for sensor demo.
- PMIC BQ25570 calculation: link
- Updated Gantt chart: link

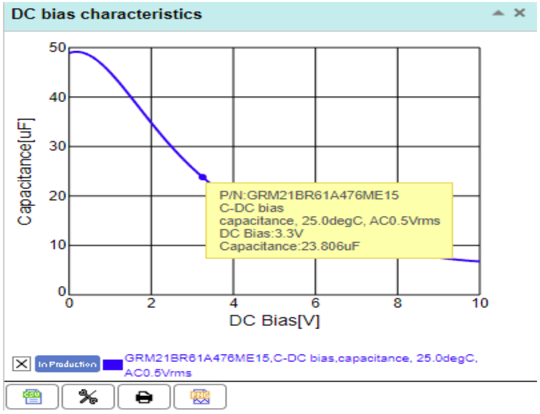
Based on the PMU / Capacitance simulation assignment, what is the actual bulk capacitance required for each of your power planes/voltages?

23uF

To achieve this actual capacitance, what are the part numbers chosen and provide data that verifies the effective capacitance at voltage DC bias of your system



490-GRM21BR61A476ME15KCT-ND
We are using 47uF capacitor because at normal temperature at 3.3 V bias it acts as 23uF capacitor.



List any I/O ports in your design

- **Power connections**
- **I/O devices**
- **USB**

Choose ESD protection for these I/O ports.

We are using ESD protection wherever there is exposed metal like USB and Push button Switch in our system

What is the p/n and why was it chosen?

SP1003-01DTG , P/n: F3377CT-ND, SP1003 Series - 30pF 30kV Unidirectional Discrete TVS, This part was chosen because we only want to protect only power lines.

Are you using VBUS only? No USB data lines?
Yes, we are using VBUS only no data lines from USB
Next week planned work

- Getting component placement reviewed from SA
- Also getting ready for sensor demonstration.
- Also starting with the routing after review from SA.
- We are on schedule as per our planned timeline.

ECEN 5833 Low Power Embedded System Design Technique

Week 6 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 26th October 2024

Previous week update review comments addressed

- **Please be more specific with where you are going to place your ESD protection devices instead of saying wherever there is exposed metal like USB and buttons.**
  We shall be placing ESD protection diodes on the USB port and the Reset button as they are only components that will be in contact with the user.

Summary

- We had a review with SA for component placement, and made necessary changes as suggested in the review meeting.
- We have also completed layout and had a review with SA and professor where we got a comment to modify the PMIC layout as the layout recommended in the datasheet is inaccurate. We shall be incorporating this change before the next design review.
- Interfaced all sensors to work on I2C protocol and demonstrated the same in class.
- Initially, we planned on using SPI protocol to interface the BMI270 sensor, but we are facing some difficulties in writing the firmware. Hence, we have temporarily interfaced the sensor with I2C protocol as fallback. We are planning to work on the SPI firmware for another week, but if we are unable to make considerable progress on this task, we shall be switching over to I2C protocol for this sensor.
- Updated Gantt chart: link

Next week planned work

- Getting updated layout reviewed from SAs and professor.
- Prepare board to send to fab
- Prepare the BOM and plan to all components
- Integrating and developing code for the whole system.
- We are on schedule as per our planned timeline.

ECEN 5833 Low Power Embedded System Design Technique

Week 7 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 11th November 2024

Summary

- Shared Digikey and mouser cart for components
- We had final review for layout and schematic from SA and professor and updated the changes.
- Verified layout with JLCPCB and submitted files for fab
- Started Bluetooth stack implementation
- Updated Gantt chart: link
- Created verification plan: link

Next week planned work

- Board assembly and bring up testing
- Finish Bluetooth stack implementation
- Start software integrating with the sensors
- Start making a 3D model for PCB.
- We are on schedule as per our planned timeline.

ECEN 5833 Low Power Embedded System Design Technique

Week 8 Update

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 22nd November 2024

Summary

- Inspected the received board and components and prepared them for assembly.
- We were able to bring up the board and run blinky to verify functionality.
- Issues faced:
    - A few capacitors near the sensors required rework.
    - The board was not being detected in Simplicity Studio.
    - The PMIC did not provide 3.3 V at the output.
    - The PMIC heats up when a USB Type-C connection is used on all three boards.
- Workaround to fix issues:
    - Resoldered the capacitors and verified connectivity.
    - Removed incorrectly connected ESD diodes, resolving the board detection problem.
    - Identified a mislabeled net in the PMIC between VOUT_EN and VSTORE, which caused the absence of 3.3 V at VOUT. A thin insulated wire was soldered to correct the issue.
    - PMIC heating persists and requires further debugging.
- We'll be using FreeRTOS for our project to manage our tasks, which we have ported from an example project and tested for blinky and buzzer tasks.
- Modelled enclosures for our prototype for 3D printing.
- Updated Gantt chart: link

Next week's planned work

- We'll be working on the Bluetooth firmware for the prototype.
- We'll be testing code for the sensors individually to verify its functionality.
- Parallelly, we'll debug and fix the PMIC issue we are facing.
- Finally, once everything is working as expected we'll integrate each individual feature as FreeRTOS tasks.
- We'll validate and evaluate our prototype by following the verification plan: link and check if it meets our proposed requirements.
- We are on schedule as per our planned timeline.

ECEN 5833 Low Power Embedded System Design Technique

Final Report

Title: Miner Safety Gear

Team Name: Tech Musketeers

Team Members:

Ajay Joy

Suhas Reddy

Vishnu Kumar

Date: 15th December 2024

Project Overview

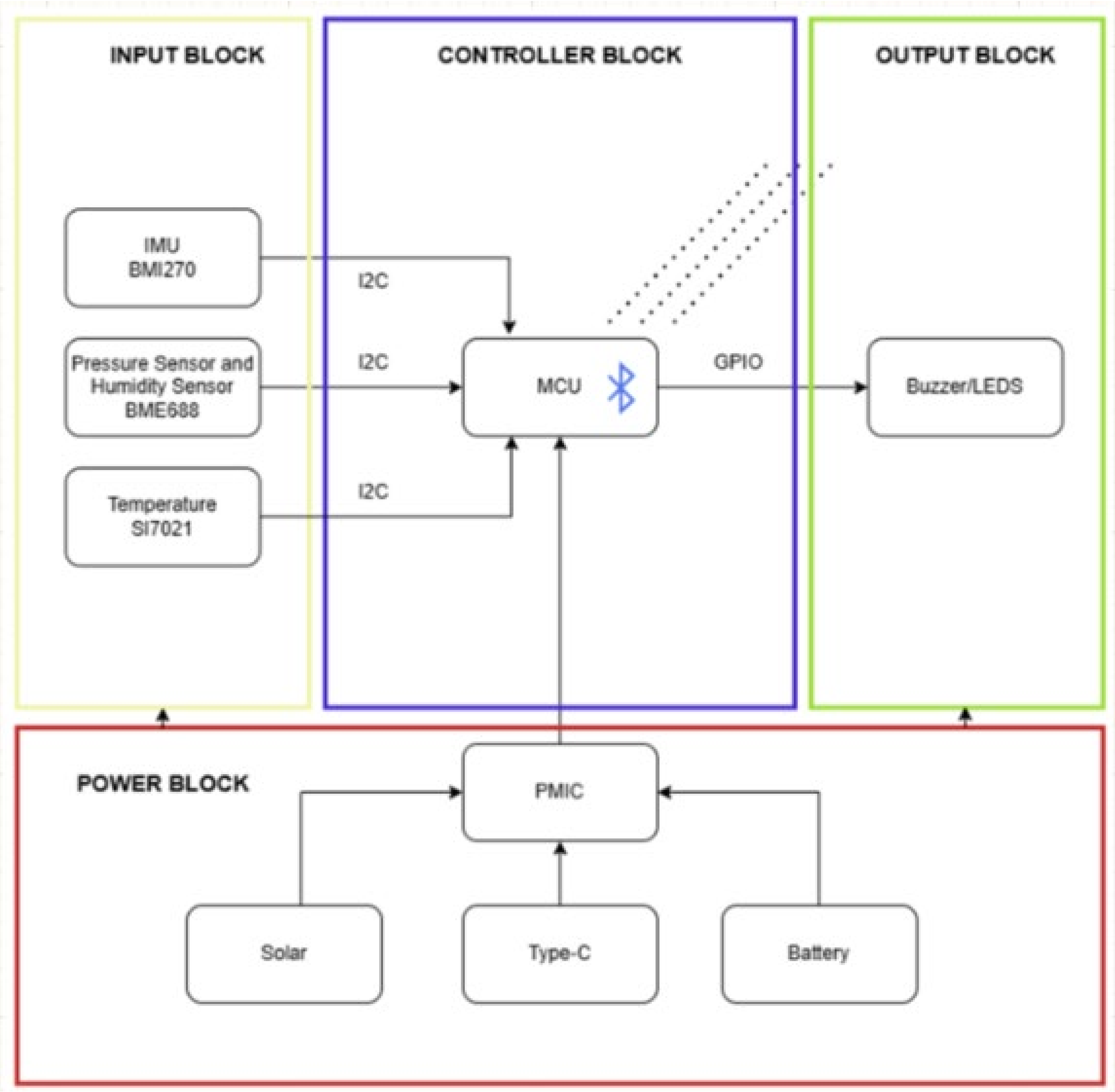
- Objective
  - Develop a low-power, portable safety device designed for workers in hazardous environments such as mines, quarries, and tunnels. The device ensures worker safety by monitoring environmental conditions, detecting hazards, and providing virtual tethering through Bluetooth mesh connectivity and RSSI values between devices. It also delivers real-time alerts to both workers and supervisors.
- Target Audience
  - Designed for workers operating in close-quarters outdoor sites involved in activities such as:
    - Mineral Mining
    - Quarry
    - Tree Harvesting
    - Oil /Gas Extraction
- Form Factor
  - A compact, wearable device that can be conveniently attached to a vest or pocket for easy access and portability.
- Connectivity: Bluetooth Mesh Network
  - All devices communicate in a Bluetooth mesh network for seamless data sharing and safety coordination.
  - Ensures continuous connectivity for workers in remote or underground locations.
  - Supervisors can monitor and manage worker safety as a team, even in isolated or low-visibility conditions.
- Key Features
  - Temperature, pressure and humidity sensors - Provide real-time updates on ambient environmental conditions.
  - Integrated IMU sensor detects falls and triggers automatic alerts.
  - Alert Mechanism - Buzzer and flasher activate upon detecting hazards or falls.
  - Supervisor Monitoring - Alerts sent to supervisors for quick action in harsh conditions.
  - Detect toxic or flammable gases to ensure safe breathing conditions.
- Project Use Cases
  - Oil and Gas Extraction
    - Monitor flammable gas levels to minimize the risk of explosions.
    - Maintain connectivity and oversee safety protocols for teams in isolated locations.
  - Tunnel Work
    - Enhance safety and communication in confined spaces.

    - Prevent overcrowding in hazardous zones through real-time monitoring.
- Disaster Response Teams
    - Assist in rescue missions under poor visibility or harsh environmental conditions.
    - Enable safe navigation and continuous connectivity for responders.
- Mineral Mining
    - Enforce safe distances between workers to prevent accidents.
    - Monitor and alert workers about toxic gas levels or other potential hazards

What problem does the project solve?

The Miner Safety Gear project addresses the critical need for enhanced safety and connectivity for workers in hazardous environments such as mines, tunnels, and quarries. These settings often expose workers to risks like toxic or flammable gases, extreme temperatures, falls, overcrowding, and limited communication, especially in remote or confined spaces. By integrating environmental monitoring, fall detection, and real-time alert systems into a compact, wearable device with Bluetooth mesh connectivity, the project ensures constant supervision, rapid hazard detection, and seamless communication. This reduces the likelihood of accidents, improves emergency response times, and provides workers and supervisors with the tools needed to maintain safety and efficiency in high-risk conditions.

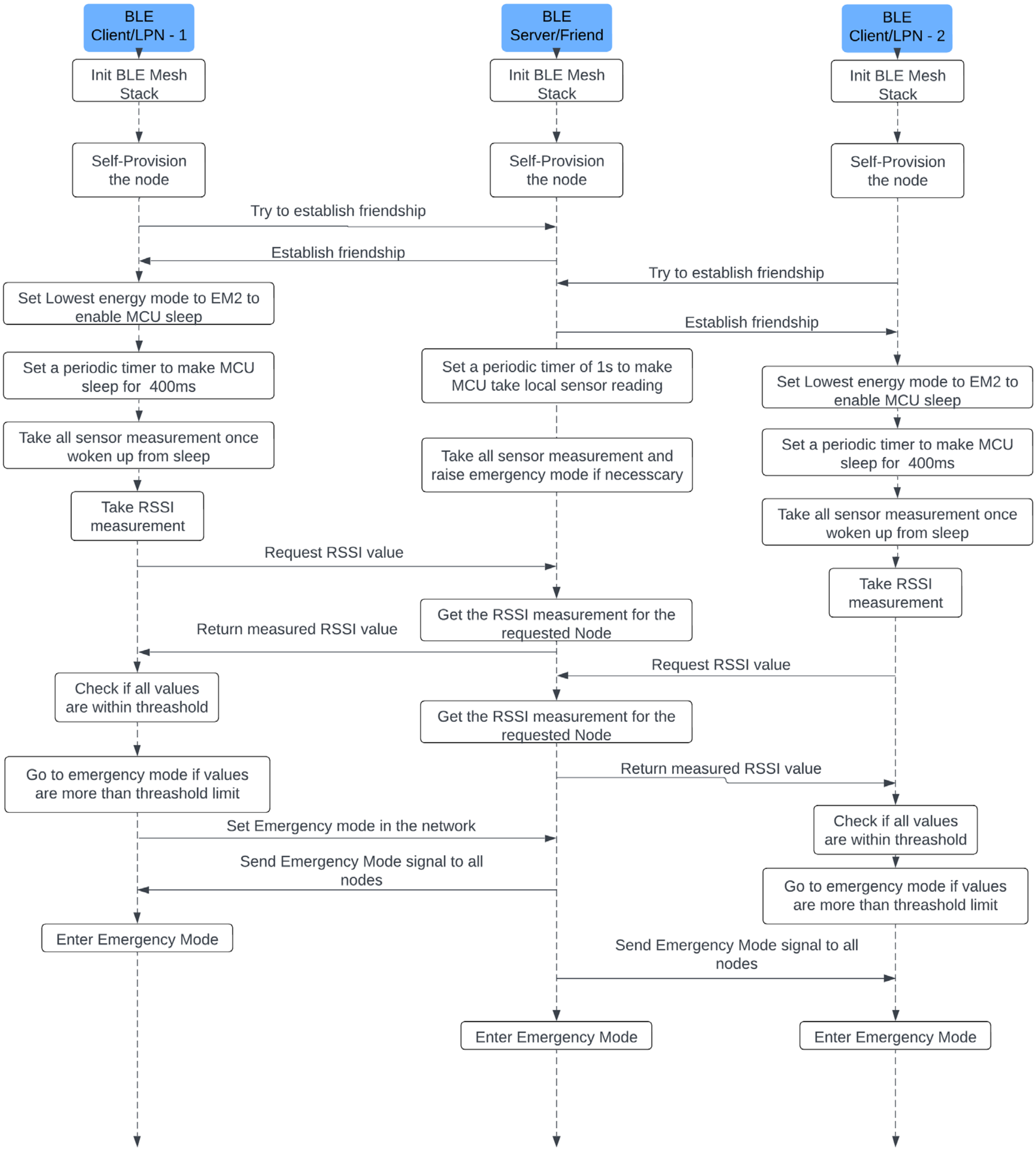
Hardware block diagram



Key components

Components	Description	Digikey part number
Blue Gecko (EFR32BG13)	Microcontroller to integrate all the sensors and acquire data with ability to interface an antenna for Bluetooth communication	336-EFR32BG13P732F512GM32-DRCT-ND
BMI270	This is used as an IMU to gather acceleration and gyroscope data and perform fall detection	828-1091-1-ND
BME688	This sensor measures humidity, pressure and gas index to map the state environment	828-BME688CT-ND
Si7021	This is used as a temperature sensor	336-4349-1-ND
Piezo Buzzer	PWM based audio alerts	445-2525-1-ND
BQ25570	Functions as a PMIC to power the MCU with stable 3.3V via battery. Also configured to charge the battery via USB-C and solar cell for energy harvesting	296-39934-1-ND
BATTERY LITH POLY 3.7V 350MAH	Battery to power the system	1528-4237-ND
KXOB141K06F-TR	Solar cell to harvest solar energy (4.15V /184.3mW)	2994-KXOB141K06F-TRCT-ND

Software flow organizational or block diagram chart



List of commands

Device	Event	Commands called in response to event	Notes
Events Common to Both Master/Server and Slave/Client	sl_bt_evt_system_boot_id		This event is generated on stack boot up
		sl_btmesh_node_init()	Init the Btmesh stack
		sl_bt_system_get_identity_address()	Get the BT address of the device

	sl_btmesh_evt_node_initialized_id		This event is generated on bt mest initilization
		sl_btmesh_vendor_model_init()	Init the vendor modedl for Btmesh data transfer
		sl_btmesh_node_set_provisioning_data()	Command to start device self-provisioning
		sl_simple_timer_start()	Create a simple timer that works with the bt mesh
		sl_bt_system_reset()	Performs a system reset
		sl_btmesh_test_get_local_model_pub()	Get a local model's publication address, key, and parameters.
		sl_btmesh_test_add_local_key()	Add a network key and application key locally.
		sl_btmesh_test_bind_local_model_app()	Bind a Model to an Appkey locally.
		sl_btmesh_test_set_local_model_pub()	et a local model's publication address, key, and parameters.
		sl_btmesh_test_add_local_model_sub()	Add an address to a local model's subscription list.
		sl_btmesh_test_set_relay()	Set the relay state and the relay retransmit state of a node locally.
		sl_btmesh_test_set_nettx()	Set the network transmit state of a node locally.
	sl_btmesh_evt_node_provisioned_id		The node has received provisioning data (address allocation and a network key) from the Provisioner
	sl_btmesh_evt_node_provisioning_failed_id		Provisioning the node has failed
	sl_btmesh_evt_node_provisioning_started_id		Provisioner has started provisioning this node.
	sl_btmesh_evt_node_key_added_id		Received when a Configuration Client has deployed a new network or application key to the node.
	sl_btmesh_evt_node_config_set_id		Configuration Client changes the State in the Configuration Server Model.

	sl_btmesh_evt_node_model_config_changed_id		This event notifies that a remote Configuration Client has changed the configuration of a local model.
Events only for Master/Server	sl_btmesh_evt_vendor_model_receive_id		Stack generates this event when a vendor message with a valid opcode is received.
		sl_btmesh_node_get_rssi()	Get the latest RSSI value of a provisioned Bluetooth device
		sl_simple_timer_stop()	Stops a previously started simple timer
		sl_btmesh_vendor_model_send()	Send vendor-specific data to the specific requested node
		sl_btmesh_vendor_model_set_publication()	Set the vendor model publication message
		sl_btmesh_vendor_model_publish()	Publish the vendor model publication message to the entire node
		sl_bt_advertiser_start() - if slave device	
	sl_btmesh_evt_friend_friendship_established_id		Indicate that a friendship has been established.
	sl_btmesh_evt_friend_friendship_terminated_id		Indicate that a friendship that was successfully established has been terminated.
Events only for Slave/Client	sl_btmesh_evt_vendor_model_receive_id		Stack generates this event when a vendor message with a valid opcode is received.
		sl_btmesh_vendor_model_set_publication()	Set the vendor model publication message
		sl_btmesh_vendor_model_publish()	Publish the vendor model publication message to the entire node
	sl_btmesh_evt_friend_friendship_established_id		Indicate that a friendship has been established.
		sl_power_manager_add_em_requirement()	Adds requirement on given energy mode.

	sl_btmesh_evt_friend_friendship_terminated_id		Indicate that a friendship that was successfully established has been terminated.
		sl_simple_timer_stop()	Stops a previously started simple timer
		sl_power_manager_add_em_requirement()	Adds requirement on given energy mode.
	sl_btmesh_evt_lpn_friendship_failed_id		Indicate that the friendship establishment has failed.
		sl_power_manager_add_em_requirement()	Adds requirement on given energy mode.

Planned development schedule and when tasks were completed

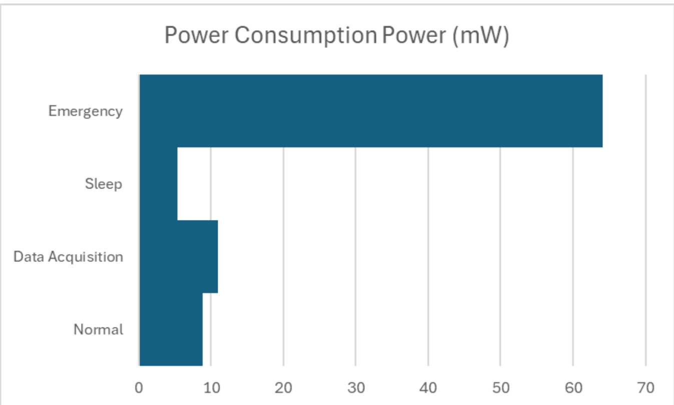
Please refer to this github link for the GANTT char for the planned development schedule and task completion: link

How the target microcontroller/SoC will be programmed?

We have used a 10-pin ARM debugger port to program the EFM32 board. This port will be connected by using the connection diagram given in an0002.1 Pg. 22 as reference. This debugger port shall be connected to the silicon labs brd4104a devkit using the Simplicity Debug Adapter Kit. This port shall be used for both programming the custom board, launching the debugger in the Simplicity Studio IDE as well as provide USART communications with the host computer.

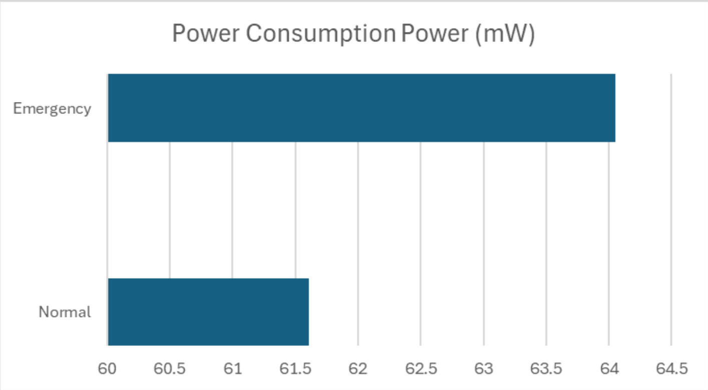
Current profile over time based on expected application usage

Current consumption for the Peripheral board (Client/LPN):



State	Current Draw (mA)
Normal	2.67
Data Acquisition	3.33
Sleep	1.62
Emergency	19.41

Current consumption for the Controller board (Server/Friend):

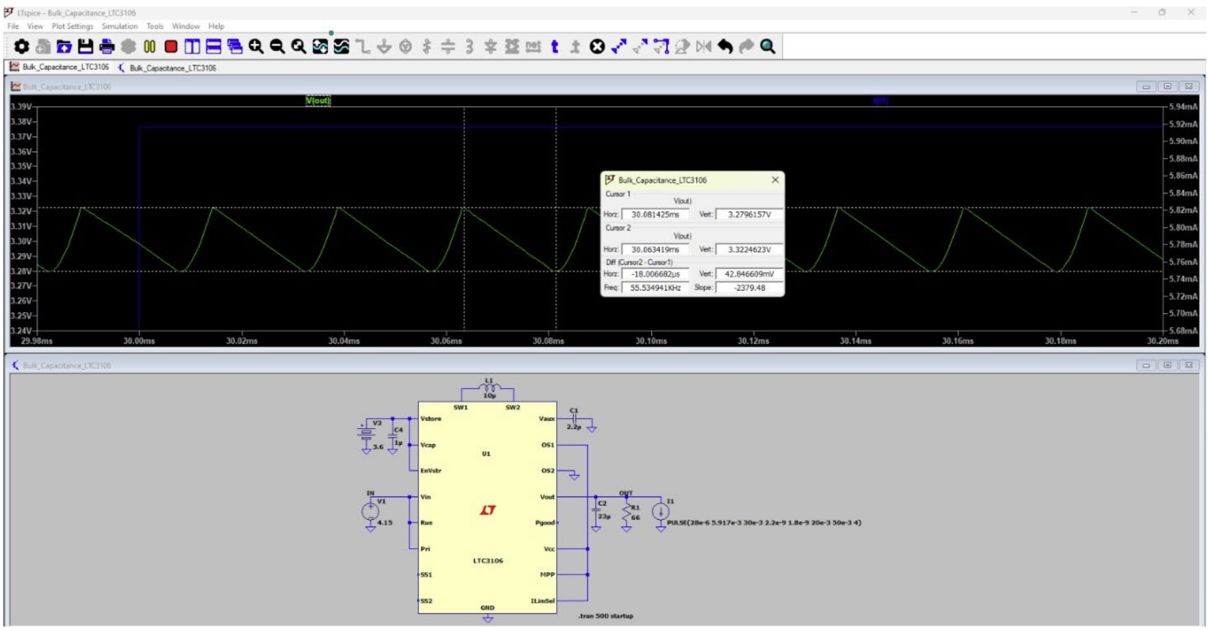


State	Current Draw (mA)
Normal	18.66
Emergency	19.41

Energy Storage Element selected and selection documentation

- The battery was selected based on the calculations performed in the following charts
- Power consumption calculation: [link]
- Power consumption chart: [link]
- Energy Storage selection
  - **Battery – 350mAh 3.7V**
    - The 350mAh battery operates within an 80-90% discharge range, maintaining a voltage above 3.3V for optimal performance and health.
    - Charging is capped at 90% capacity (4.2V cut-off), enabling a 72-80 hour charge cycle while preserving battery longevity.

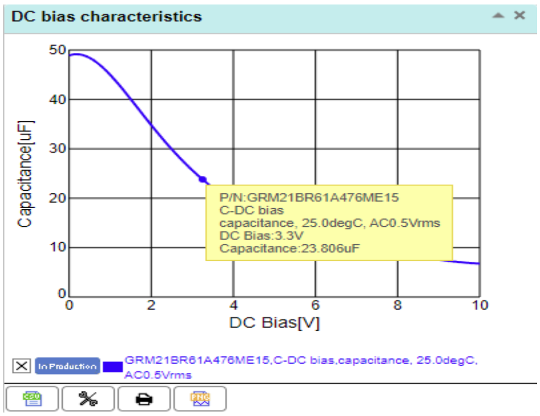
PMU simulation results / summary



- This simulation verified the PMIC configuration to harvest energy and if it provided stable 3.3V at buck converter output for the given battery voltage
- We also determined that bulk capacitance value to be 23uF
- LTC3106 was used as an alternative to simulate BQ25570 to work as a charger, energy harvester and buck converter
- **PMIC – BQ25570**
  - Supports energy harvesting from input voltages as low as 100 mV, with efficient battery charging and regulated 3.3V power supply using a buck converter (up to 93% efficiency).
  - Buck converter handles peak output current up to 110 mA, ensuring stable operation for the system.

Bulk or large decoupling capacitor selection and back up data

- From the above simulation of the PMIC we determine system would work with 23uF capacitor
- We are using 47uF capacitor because at normal temperature and at 3.3 V bias it acts as 23uF capacitor, which can be seen in the below plot



Will an external energy source be required to program the MCU?

- No, the power is provided through the SWD power pins from the programmer board

Planned test points

- The PCB had test points for I2C clock and data, PMIC's VSTOR, VOUT_EN, VOUT, VBAT_OK , Battery input, Type-C and GND.
- Did these work for bring up? Should there have been more test points?
  - These helped in bringing up the board, debug and fix issues

Photos of assembled board

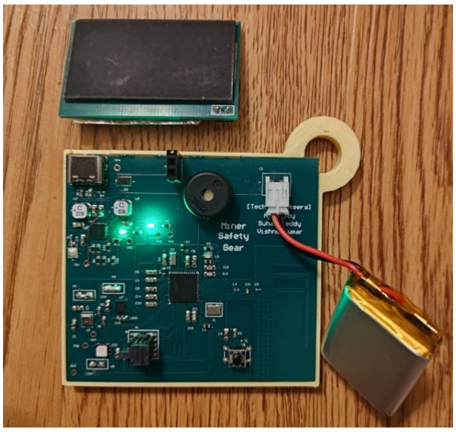


Figure 1 Assembled boards with battery connected

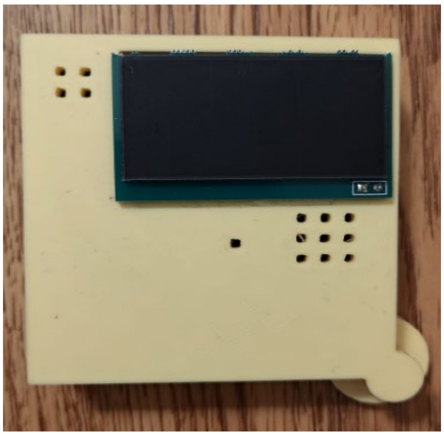


Figure 2 Board with casing with solar panel placed on top

Complete detailed verification report (spread sheet)

- verification plan: link completed
- Some of images taken while testing the board

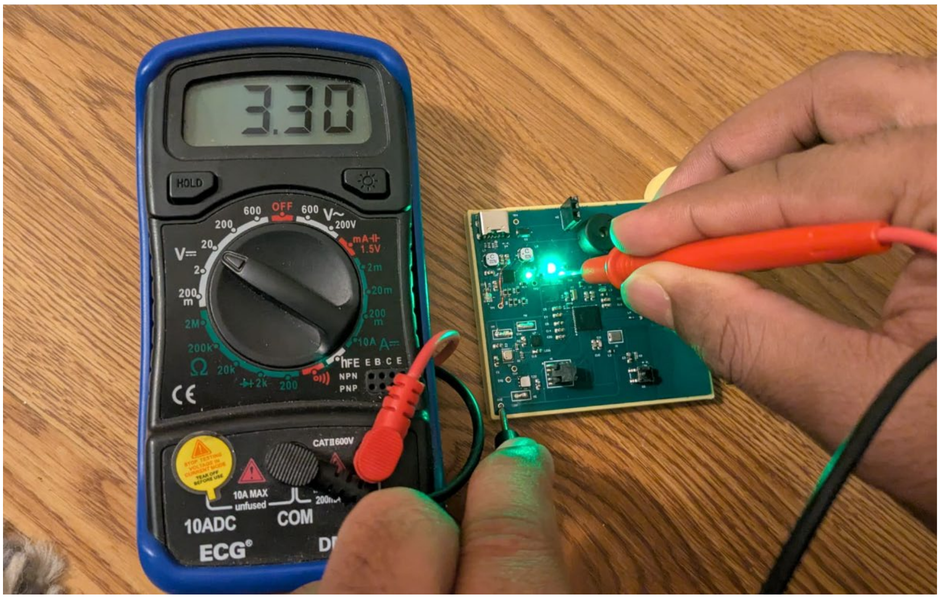


Figure 3 VOUT from PMIC

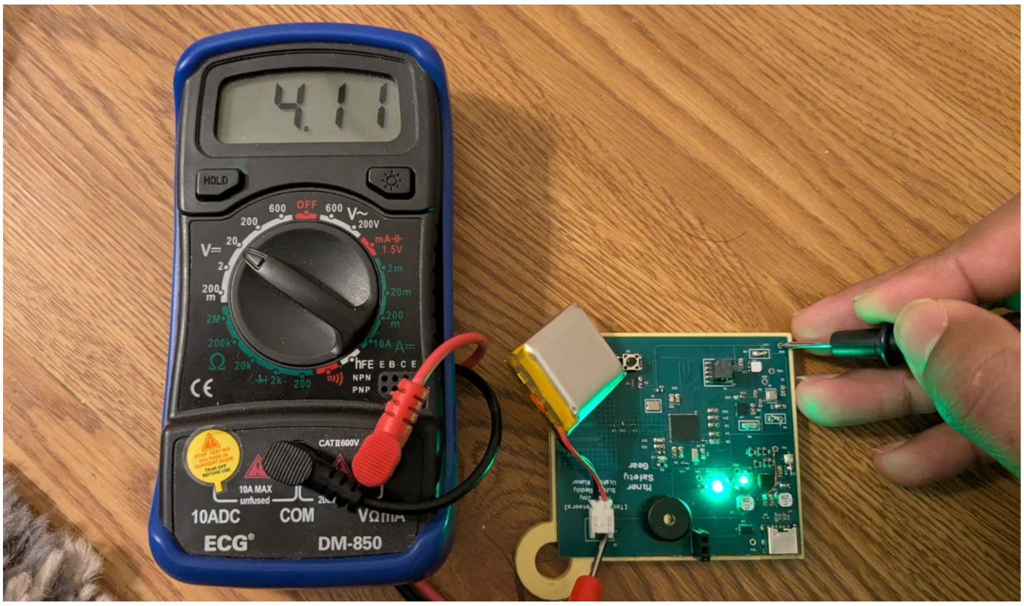


Figure 4 Battery Discharging

Time	Battery Voltage
11:30 AM	3.66V
12:30 PM	3.81V
1:30 PM	4.11V
2:00 PM	4.21V

Figure 5 Charging of battery via USB C

- Took almost 2.5hrs to charge complete battery from 3.6 to 4.21V
- Provides about 80 hrs of continuous operation.

Signal quality analysis of key signals

- The primary signals analyzed in the system are the I2C bus lines, SCL and SDA.
- Signal damping observed is attributed to the effects of the pull-up resistors on the bus lines.
- Below images are of SCL and SDA of I2C bus which is as expected

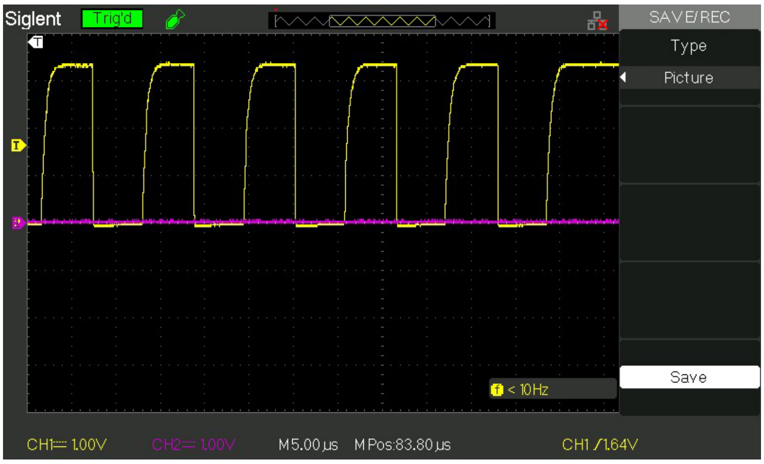


Figure 6 I2C-SCL from Scope

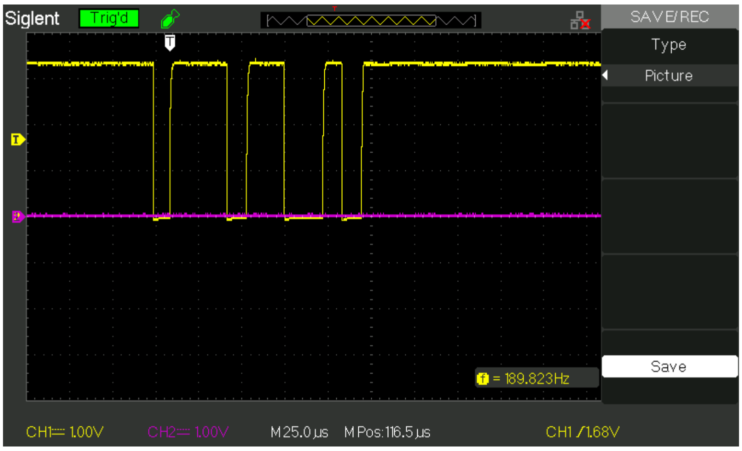


Figure 7 I2C-SDA from Scope

What were the difficulties encountered on the project?

- We faced difficulties during board bring up. Most of the issues were related to incorrect diode orientation which let to the PMIC not receiving power via battery or from the USB-C port
- We also faced problems with the firmware development as initially we planned on the Peripheral being the GATT server that will fetch the local sensor data and will pass the data to the Controller which will be the GATT client and shall process all data, but due to the way BLE mesh is setup, the Peripheral is set as GATT client and Controller as GATT server, so we had to make changes to the firmware and overall software flow to accommodate this change.

Summary of functionality of final project

- Successfully established communication between the 3 custom boards via Bluetooth mesh
- Acquired all the required data from the sensors via I2C bus through which we performed fall detection, detect hazardous environmental conditions and provide appropriate alerts via buzzer and LED
- Implemented virtual tethering using RSSI value and was verified by moving node away from group of nodes

Each team member – 5 lessons learned not taught in lecture but learned through doing the project. These individual lessons learned will be graded on quality

Ajay

- I learnt to debug the Power Management Unit which was very new to me
- Understood the energy management down to the protocol level
- Learnt about effect of signal integrity in low power boards
- Was able to experience system design of a custom board with antenna
- Learnt to make 3D CAD model

Suhas

- I learnt how to important it is to refer the datasheet to verify a circuit for an IC after layout design so, that there is no miss matched connections.
- I learnt a lot about simplicity studio and its capabilities and using the mainboard and SWD cable to program a custom board.
- I realized having several test points in a prototype board is very important which, makes the debugging and fixes very easy. We resolved one such issue of mis matched net and short a test point with a capacitor lead thus reducing the time and risk of waiting for a new PCB.
- Writing firmware for a complex and capable sensor is a hassle because of the large initialization sequence required. BMI270 IMU sensor required 8KB initialization data to be written.
- I learnt a lot about software and hardware debugging techniques while programming a custom board.

Vishnu

- I learnt how to implement a BLE mesh using the EFR32BG13 MCU using Bluetooth SDK v3.2.9.
- I learnt how the BLE stack is implemented for the mesh networking protocol and how much it deviates from the normal BLE stack such as if a node is designed to go to sleep for some amount of time, it should be initialized as a Low Power Node and it should be connected to a Friend node that can buffer the messages intended for the LPN.
- I learnt how to assemble a custom board using a semi-automatic pick-place machine, and a reflow oven and how important it is to select the solder paste based on the component with the lowest thermal tolerance.
- I learnt how to use the soldering tweezers, and hot air reflow station to rework the solder connections on some components and techniques to ensure that oscillators do not get affected by this rework.
- I learnt how to flash a bootloader to a newly received MCU chip, how to enable debugging on this chip and how to remap pins so that default USART functionality is routed through the debug pins.

WOW/Impressive factor of report (Instructor discretion.)