# NEXASENSEE

## EE40190 Senior Design II

## Design Review 2

March 4th, 2025

| Name | NDID |
| --- | --- |
| Jeff M. Mwathi | 902182754 |
| Katherine Davila | 902167325 |
| AnnahMarie Behn | 902172819 |
| Kyle Crean | 902180910 |
| Jeffrey Yang | 902179486 |

**List of subsystems:**

- Sensor Subsystem
    - Demonstrating proper and accurate collection of temperature and light
        - Microphone arrived same day as demo
- Optical Communication Subsystem
    - Demonstrating ability to transmit information
- Data Decoding and Logging, Ethernet Communications, and GUI
    - Demonstrating ability to record data and save to .xlsx or .csv file

# Introduction

This Design Review 2 (DR2) demonstration for NexaSensee evaluates the functionality of each subsystem to ensure proper operation. The review assesses subsystem performance based on demonstration completeness and system integration. A fully functional system with all subsystems working earns the highest score, while incomplete or missing components lower the evaluation. This demonstration is a key step in validating NexaSensee's design before final integration. In this document, we take a closer look at the power subsystem, analyzing its design, performance, and impact on overall system reliability.

**Power Subsystem Detailed Design:**

## 1. Introduction

The power subsystem provides stable and efficient power to all system components. Since our PCB is not yet fabricated, we present a detailed design of the power system, including schematics, component selection, and calculations.

Because the receiver hub will connect to an ethernet jack to connect back to the central console, we also assume that a power plug port is nearby. Thus, the receiver hub board is powered with 5V over USB. USB 5V is converted to 3.3V and to ±3V to power the ESP32 microcontroller, components of the optical receiver, and ethernet hardware.

Our design for the sensor board is powered by a 3.7V lithium-ion battery. The battery's voltage directly powers the components of the optical transmitter and is converted to 3.3V to power the ESP32 microcontroller and sensors.

This design also includes the circuitry needed for two-way communication (i.e. both the receiver hub and sensor board have an optical transmitter and receiver). The designed powering scheme of the optical transmitter and receiver are adapted for the two source voltages (3.7V and 5V). However, detailed power calculations are performed only for one-way communication, as two-way communication requires significantly more current.

## 2. Power Subsystem Circuit Schematics

The power specific components are:

Sensor Board (Transmitter)

- 3.7V Lithium-Ion Battery (Primary Power Source)
- Buck-Boost Converter (3.7V → 3.3V) for Efficient Voltage Regulation
- ESP32-S2 Microcontroller
  - For 115.2 kbps, single core at 160 MHz is sufficient for simultaneously
- Sensors
- Optical Communication Components
  - If two-way communication is implemented, a power IC is needed to convert 3.7V to ±3V for receiver circuitry. Additional circuitry is added to toggle the power to the op-amps and put the receiver in a low power, initial IR detection mode. In this mode, only the comparator is powered; a long IR pulse emitted by the transmitter will cause the receiver circuitry to turn on.
- Voltage Divider Battery Monitoring
  - 100 kΩ & 68 kΩ to minimize ADC input impedance and keep input in safe range:
    - (4.2 * (68/(68+100))) V, (3.7 * (68/(68+100))) V, (3.0 * (68/(68+100))) V

Receiver Hub Board

- USB port with 5V pin → power efficiency does not have to be optimized.
- LDO Regulator (5V → 3.3V)
- ESP32-S3 Microcontroller
  - Two cores can be used separately for the optical communication link (data decoding and coordinating communication if two-way communication is implemented) and ethernet.
- Ethernet Communication Components - W5500
- Optical Communication Components
  - Power IC needed to convert 3.7V to ±3V for receiver circuitry

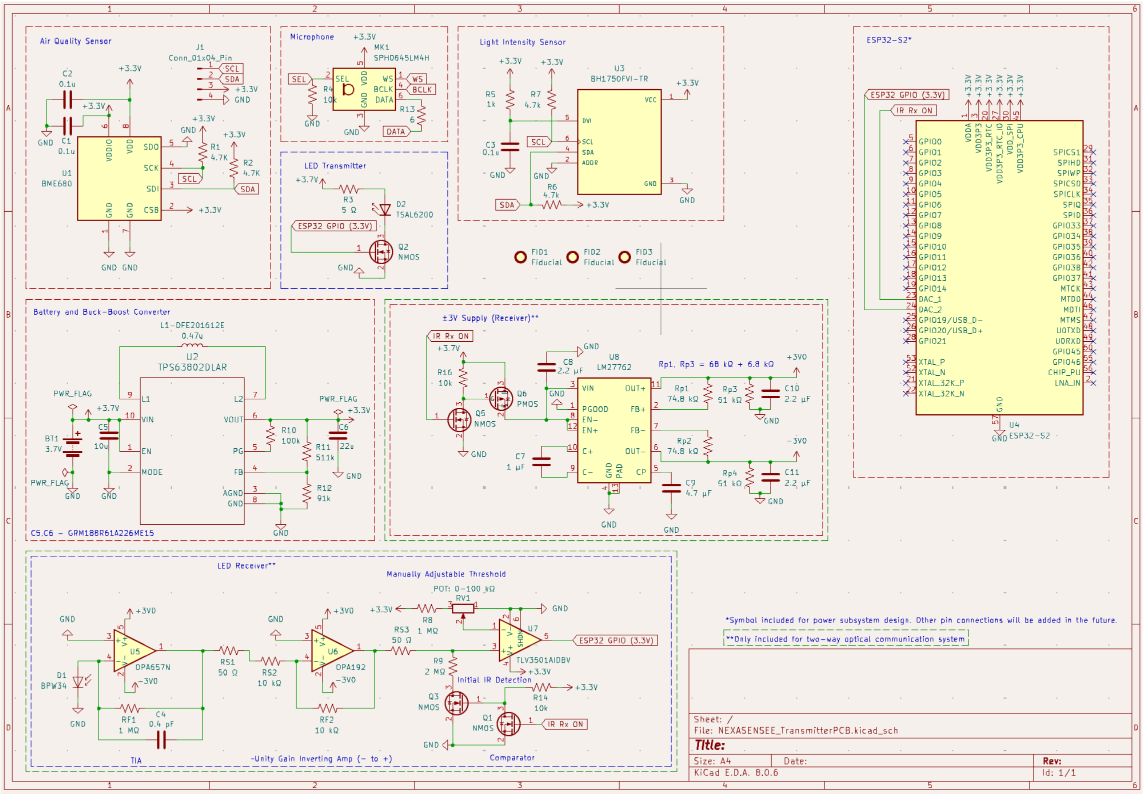

**Figure 1**. Sensor Board/Transmitter PCB

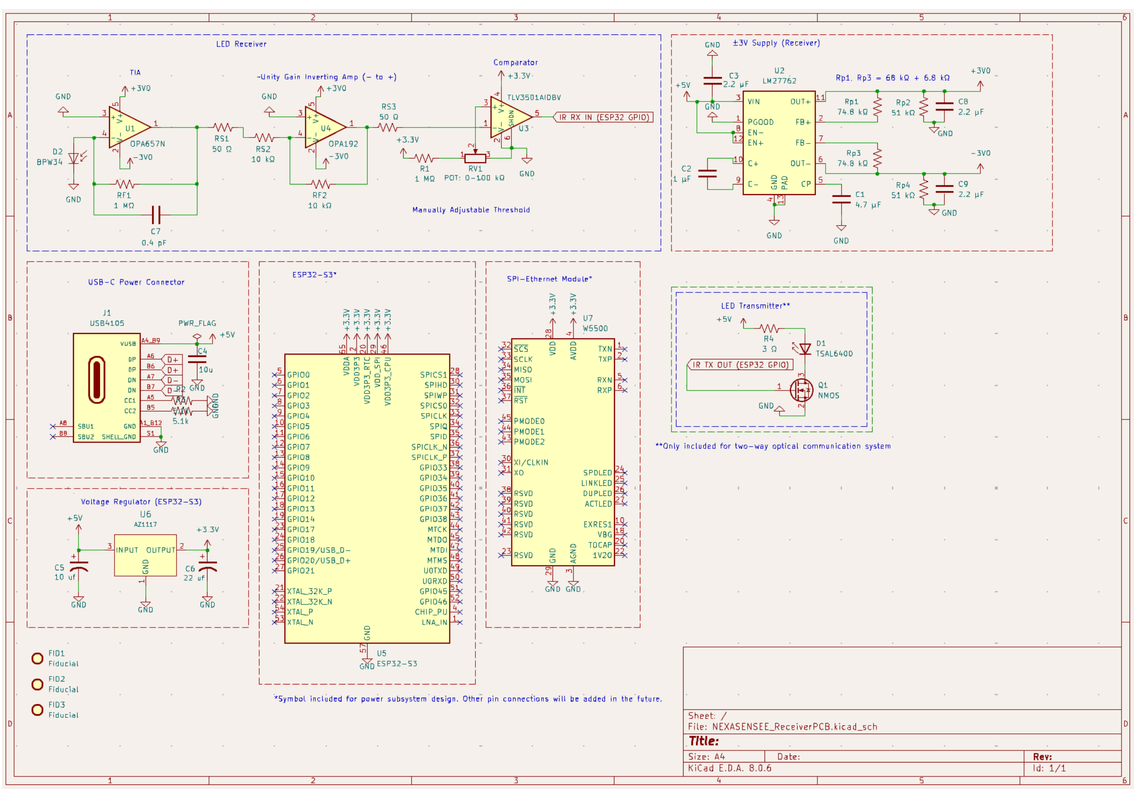

**Figure 2**. Receiver Hub PCB

## 3. Component Selection Justification

### 3.1 Buck Boost Converter for 3.7V → 3.3V Conversion

Why a Buck-Boost DC-DC Converter instead of an LDO regulator?

- Higher efficiency (~85-90%) than LDO regulators, dissipating excess voltage as heat.
  - Minimizes power loss, leading to longer battery life and reduced thermal impact.
- When nearly discharged, a 3.7V Li-ion battery's voltage drops to 3V, which is lower than the target supply voltage of 3.3V. An LDO regulator cannot provide a voltage higher than the input.
- These issues do not exist for USB 5V, so an LDO regulator is used for the receiver hub board.

| **LDO Power Loss:** | **Buck Converter Loss (~10%):** |
|---|---|
| $P_{loss} = (V_{in} - V_{out})\ x\ I_{load}$<br>$P_{loss} = (3.7V - 3.3V)\ x\ 0.5A = 0.2W$ | $P_{loss} = (V_{in} - V_{out})\ x\ I_{load}$<br>$P_{loss} = (3.3V\ x\ 0.5A\ x\ 0.1) = 0.165W$ |

## 4. Power Calculations & Battery Life Estimation

### 4.1 | Power Requirements Summary

| **Power Requirements Summary** | | | |
|---|---|---|---|
| **Component** | **Voltage (V)** | **Current (Active, mA)** | **Current (Sleep, µA)** |
| **ESP32 Microcontroller (modem-sleep mode):**<br>**Sensor Board: ESP32-S2**<br>**Receiver Hub: ESP32-S3** | 3.3V | **Sensor Board: 24mA** at 160 MHz, CPU and all peripherals on<br>~~**Receiver Hub:** 64.1 mA at 160 MHz, Dual-core, all peripherals on~~<br>N/A, powered by USB | **Sensor Board: 10 µA**<br>~~**Receiver Hub: 15 µA**~~<br>N/A, powered by USB |
| **BME680 Sensor (Temp, Humidity, Pressure, Gas)** | 1.71V to 3.6V | 0.09 - 12 mA → **12 mA** | **0.15 µA** |
| **BH1750 Sensor (Light)** | 2.4V to 3.6V | 0.12 - 0.18 mA → **0.12mA** | **0.01µA** |
| **I²S SPH0645LM4H Microphone (Audio)** | 1.62V to 3.6V | 600 µA | **10µA** |
| **Infrared LED and associated circuitry**<br>**Sensor Board: VSLY5940**<br>**Receiver Hub: TSAL6400** | 3.7 V | **Sensor Board: 31.25 mA**<br>~~**Receiver Hub: 94 mA***~~<br>**N/A powered by USB*** | N/A, LED either on or off<br>NMOS leakage: **10 µA** |
| **BPW34 Photodiode and associated circuitry** | 5 V or 3.7V (amplification) and 3.3 V (comparator) | **Sensor Board: 20.2 mA***<br>~~**Receiver Hub: 20.2 mA**~~<br>**N/A, powered by USB** | Comparator always on, negligible IC turn-on times.<br>**Sensor Board: 3.2 mA***<br>~~**Receiver Hub: 3.2 mA**~~<br>**N/A, powered by USB** |
| **TPS63802 DC-DC Converter Efficiency (Sensor Board)** | 85% - 90% | N/A | N/A |

| Power Requirements Summary | | | |
|---|---|---|---|
| [W5500](#) Ethernet Module (Receiver Hub) | 3.3V | ~~132 mA, 100M transmitting mode~~ N/A, powered by USB | ~~180 µA, power-down mode~~ N/A, powered by USB |
| **Total (Sensor Board)** | | **67.37mA + 620µA= 67.37mA + 0.620 mA = = 67.99 mA** | **20.16µA = 0.02016 mA** |

*Only needed for a two-way communication system, where a transmitting LED is added to the receiving hub and a photodiode and receiver circuitry are added to the sensor board. Excluded highlighted currents for a one-way communication system.

### 4.2 | Battery life calculations without power optimization

*Estimate how long the battery will last under continuous operation at an average current draw of 345mA.*

| Lithium Ion Polymer Battery options | Battery Option 1: - 3.7v 500mAh | Battery Option 2: - 3.7v 1,200mAh | Battery Option 2: - 3.7v 2,500mAh |
|---|---|---|---|
| **Energy available from the battery** | Battery specification: - Capacity = 500 mAh (0.5Ah) - Voltage = 3.7V - Total energy stored: $E = V \times C$ - $3.7V \times 0.5Ah = 1.85\,Wh$ The battery can deliver 1.85 watt - hours of energy before depletion | Battery specification: - Capacity = 1,200 mAh (1.2Ah) - Voltage = 3.7V - Total energy stored : $E = V \times C$ - $3.7V \times 1.2Ah = 4.44\,Wh$ The battery can deliver 4.44 watt - hours of energy before depletion | Battery specification: - Capacity = 1,200 mAh (1.2Ah) - Voltage = 3.7V - Total energy stored : $E = V \times C$ - $3.7V \times 2.5Ah = 9.25\,Wh$ The battery can deliver 9.25 watt - hours of energy before depletion |
| **Power Consumption of System** | System operates at 3.3V so power usage: - Current draw : 68mA = 0.068A - Power consumption: $P = V \times I = 3.3V \times 0.068A = 0.224W$ Meaning that the system consumes 1.14 watts continuously | | |
| **Runtime without optimization** | $runtime = \frac{battery\ energy}{power\ consumption}$ $runtime = \frac{1.85\,Wh}{0.224W} = 8.25hrs$ Without any optimizations, the 500mAh battery would last about 8.25 hours before depletion | $runtime = \frac{battery\ energy}{power\ consumption}$ $runtime = \frac{4.44\,Wh}{0.224W} = 19.8hrs$ Without any optimizations, the 1,200mAh battery would last about 19.8 hours before depletion | $runtime = \frac{battery\ energy}{power\ consumption}$ $runtime = \frac{9.25\,Wh}{0.224W} = 41.29hrs$ Without any optimizations, the 2,500mAh battery would last about 41.29 hours/1.72 days before depletion |
| ***Battery Life*** | ***8.25 hours*** | ***19.8 hours*** | ***1.72 days*** |

### 4.3 | Battery life calculations with power optimization

*Estimate of how long the battery will last under power optimization and comparing different scenarios of frequency of data collection (determining when in active mode and when in sleep mode).*

**The total current draw of the system is a combination of:**

1. Active Mode: system wakes up, collects data, transmits, then goes back to sleep.
   a. Draws ~ 68 mA for a short duration (2 seconds per cycle).

2. Deep-Sleep Mode: system is in low power mode.
    a. Draws ~ 0.02016 mA when sleeping.
3. Duty Cycle Impact: The more frequently it wakes up, the less time it spends in deep sleep
    a. Assuming time spent in active mode is estimated to be 2 seconds
    b. Asleep for the rest of the time

**Average current draw:**

$$I_{avg} = (duty\ cycle\ \times I_{active}) + ((1 - duty\ cycle) \times I_{sleep})$$

- $duty\ cycle = \frac{time\ spent\ in\ active\ mode}{total\ cycle\ time}$
- $I_{active} = 68mA = system\ draw\ during\ active\ mode$
- $I_{sleep} = 0.02016mA = system\ draw\ in\ deep\ sleep\ mode$

**Average Current Draw at varying wake up times:**

| **Frequency data collection, every:** | **Duty cycle:** **Amount of time active in % form** | **Average Current Draw:** $I_{avg}$ <br> $I_{avg} = (duty\ cycle\ \times I_{active}) + ((1 - duty\ cycle) \times I_{sleep})$ |
|---|---|---|
| **10 minutes (600 sec)** | $= \frac{2}{600} = 0.0033$ <br> 0.33% | $I_{avg}$= (0.0033 x 68mA) + (0.9967 x 0.02016mA) <br> = (0.2244) + (0.0201) <br> = 0.244mA |
| **5 minutes (300 sec)** | $= \frac{2}{300} = 0.0067$ <br> 0.67% | $I_{avg}$= (0.0067 x 68mA) + (0.9933 x 0.02016mA) <br> = (0.456) + (0.0201) <br> = 0.476mA |
| **3 minutes (180 sec)** | $= \frac{2}{180} = 0.0111$ <br> 1.11% | $I_{avg}$= (0.0111 x 68mA) + (0.9889 x 0.02016mA) <br> = (0.755) + (0.012) <br> = 0.767mA |
| **2 minutes (120 sec)** | $= \frac{2}{120} = 0.0167$ <br> 1.67% | $I_{avg}$= (0.0167 x 68mA) + (0.9833 x 0.02016mA) <br> = (1.14) + (0.02) <br> = 1.16mA |
| **1 minute (60 sec)** | $= \frac{2}{60} = 0.033$ <br> 3.33% | $I_{avg}$= (0.033 x 68mA) + (0.967 x 0.02016mA) <br> = (2.244) + (0.02) <br> = 2.264 mA |

| | | |
|---|---|---|
| **30 seconds** | $= \frac{2}{30} = 0.067$<br>6.67% | $I_{avg}$ = (0.067 x 68mA) + (0.933 x 0.02016mA)<br>= (4.556) + (0.02)<br>= 4.57mA |
| **10 seconds** | $= \frac{2}{10} = 0.2$<br>20% | $I_{avg}$ = (0.2 x 68mA) + (0.8 x 0.02016mA)<br>= (13.6) + (0.016)<br>= 13.62mA |

**Battery Life of Option1 and Option2 for each frequency:**

- $battery\ life = \frac{battery\ capacity\ (mAh)}{Avg.\ Current\ Draw\ (mA)}$

| **Frequency data collection, every:** | **Battery Option 1: [500mAh]**<br>- [3.7v 500mAh] | **Battery Option 2: [1,200mAh]**<br>- [3.7v 1,200mAh] | **Battery Option 3: [2,500mAh]**<br>- [3.7v 2,500mAh] |
|---|---|---|---|
| **10 minutes (600 sec)** | $B_{life} = \frac{500mAh}{0.244mA} = 2049.2\ hours$<br>Days = 2049. 2 /24 = **85.3 days** | $B_{life} = \frac{1{,}200mAh}{0.244mA} = 4918\ hours$<br>Days = 4918 /24 = **204 days** | $B_{life} = \frac{2{,}500mAh}{0.244mA} = 10,245.9\ hours$<br>Days =10, 245 /24 = **426 days** |
| **5 minutes (300 sec)** | $B_{life} = \frac{500mAh}{0.476mA} = 1050\ hours$<br>Days = 1050/24 = **43 days** | $B_{life} = \frac{1{,}200mAh}{0.476mA} = 2521\ hours$<br>Days = 2521/24 = **105 days** | $B_{life} = \frac{2{,}500mAh}{0.476mA} = 5252\ hours$<br>Days =5252/24 = **218 days** |
| **3 minutes (180 sec)** | $B_{life} = \frac{500mAh}{0.767mA} = 651\ hours$<br>Days = 651/24 = **27 days** | $B_{life} = \frac{1{,}200mAh}{0.767mA} = 1564\ hours$<br>Days = 1564/24 = **65 days** | $B_{life} = \frac{2{,}500mAh}{0.767mA} = 3259\ hours$<br>Days =3259/24 = **135 days** |
| **2 minutes (120 sec)** | $B_{life} = \frac{500mAh}{1.16mA} = 431\ hours$<br>Days = 431/24 = **17.9 days** | $B_{life} = \frac{1{,}200mAh}{1.16mA} = 1034\ hours$<br>Days = 1034/24 = **43 days** | $B_{life} = \frac{2{,}500mAh}{1.16mA} = 2155\ hours$<br>Days =2155/24 = **89 days** |
| **1 minute (60 sec)** | $B_{life} = \frac{500mAh}{2.264mA} = 220\ hours$<br>Days = 220/24 = **9 days** | $B_{life} = \frac{1{,}200mAh}{2.264mA} = 530\ hours$<br>Days = 530/24 = **22 days** | $B_{life} = \frac{2{,}500mAh}{2.264mA} = 1104\ hours$<br>Days =1104/24 = **46 days** |
| **30 seconds** | $B_{life} = \frac{500mAh}{4.57mA} = 109\ hours$<br>Days =109/24 = **4.5 days** | $B_{life} = \frac{1{,}200mAh}{4.57mA} = 262.5\ hours$<br>Days = 262.5/24 = **10 days** | $B_{life} = \frac{2{,}500mAh}{4.57mA} = 547\ hours$<br>Days =547/24 = **22.7 days** |
| **10 seconds** | $B_{life} = \frac{500mAh}{13.62mA} = 36.7\ hours$<br>Days = 36.7/24 = **1.5 days** | $B_{life} = \frac{1{,}200mAh}{13.62mA} = 88\ hours$<br>Days = 88/24 = **3.6 days** | $B_{life} = \frac{2{,}500mAh}{13.62mA} = 183.5\ hours$<br>Days =183. 5 /24 = **7.6 days** |

**4.4 | Note ULP: Ultra-Low-Power Coprocessor**

The Ultra-Low-Power (ULP) Coprocessor and RTC memory remain powered during deep sleep, allowing the system to perform background tasks without fully waking the ESP32. This enables:

- Event-driven wake-ups: The ULP coprocessor can monitor environmental changes, such as sudden shifts in temperature or humidity, and trigger immediate data collection.

- Threshold-Based Wake-Up: Instead of relying solely on fixed periodic sensing (e.g., every 10 minutes), the ESP32 can wake up when sensor readings exceed predefined thresholds.
- Adaptive Power Management: By leveraging ULP-driven event detection, battery life can be further extended beyond traditional deep-sleep cycles.

These approaches could be considered for future addition to ensure efficient power usage while maintaining responsiveness to critical environmental changes, making it ideal for long-term monitoring applications.