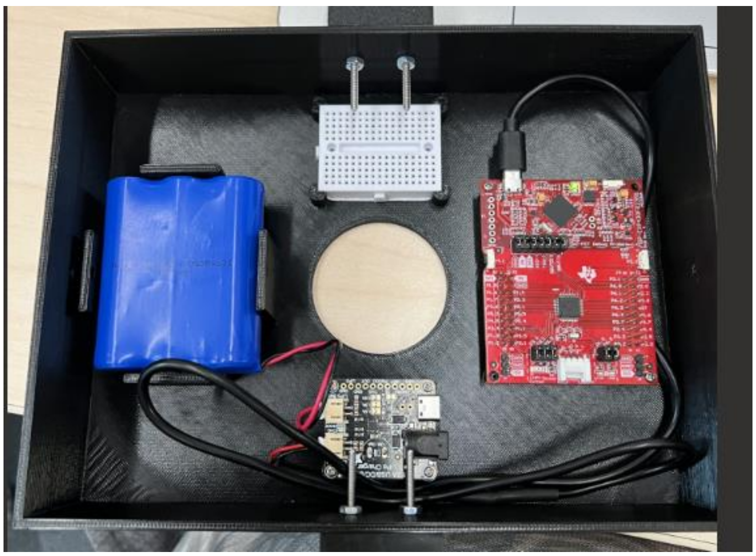
Design Report: Weather Monitoring System Power Supply Subsystem

Binh Ho

Tony Leggiero

Steven Vele

Michael Rabanales



Abstract:

The New York State Department of Environmental Conservation is implementing a distributed weather network statewide to better comprehend climate change's impact on microclimates. This initiative involves strategically using weather monitoring stations to gather comprehensive data on local weather patterns. The network will enable real-time monitoring, in-depth analysis, and predictive modeling, empowering policymakers, and stakeholders with valuable information for proactive climate adaptation. Through collaboration, this project strives to support sustainable ecosystem management and mitigate climate change effects on New York State's diverse environments.

Commented [JM1]: good

1. The problem:

- ## Problem statement:

The New York State Department of Environmental Conservation needs a distributed weather network across New York State so that the impact of climate change on microclimates can be better understood.

- ## Problem Introduction:

The NYSDEC is seeking to expand its understanding of the variations of microclimates across the entire states. The method by which this can be achieved is by making an affordable home weathering system that homeowners and business owners can acquire and monitor the type of weather around their homes/businesses. The importance of home and business owners possessing their own home weathering station is to obtain precise climate readings specific to their exact location. This will provide an advantage from using weather apps. Weathering apps often gather data from broader, less specific areas such as centralized locations or airports. With a personal weathering station, the owner can access accurate, real-time information specific to their immediate environment, which can help decision-making for activities that depend on the condition of the weather. It would be highly beneficial for the NYSDEC as it would allow the collection of extensive data from many individuals. This approach will provide specific readings from sensors in various parts of a city or town, as a result allowing the NYSDEC to gain a detailed understanding of the impact of microclimates in NY.

2. Inspiration:

The inspiration for this project was drawn from various sources, one of which was research conducted via Google. This influenced the decision-making process significantly. Initially, there was limited understanding of the structure and components of a home weathering system. However, upon reviewing existing models available in the market, insights into their functionalities and designs were gained. This facilitated the analysis of disparities in sensor technologies across these systems and evaluation of various microcontroller options based on power requirements and output capacities necessary for sensor operation. Among the prominent weathering systems examined was the model developed by Ambient Weather.

Monitoring & Alerts." *Ambient Weather*, ambientweather.com/ws-2902-smart-weather-station. Accessed 12 Mar. 2024]

Upon initial observation of the system, a notable aspect was the arrangement of sensors atop the pole. Subsequently, attention was directed towards the LCD display, which presented multiple readings of diverse sensor information. This prompted consideration of additional sensors beyond those visibly integrated. Moreover, upon accessing the system's webpage, information regarding its support for wireless sensors was discovered, inspiring the idea of enhancing the system's wireless functionality.

["Kestrelmet 6000 Weather Station - Professional Weather Station." Ambient Weather, ambientweather.com/kestrelmet-6000-weather-station. Accessed 12 Mar. 2024.]

The main distinctions between this system and the previous one are that this home weathering station has a rain gauge, and the wind sensor is more elevated. Additionally, a significant innovation in the current system is the incorporation of a solar panel, which is intended to supply power to the system. It should be noted that the specific functions and efficiency of the solar panel aren't detailed in on the page. The concept of using the solar panel to power the system and enhancing its wireless capabilities will be adopted in our system.

Another source of inspiration was the slide information from the in-class lecture, which provided valuable information the influenced our decision making. They initially helped with deciding the type of communication transfer between the system storage and the sensors whether it would be 802.11ac Wi-Fi or Radio Frequency (RF) signals. They then contributed to the decision of how the data would be stored within the system storage from the sensors either being XML or CSV files.

Commented [JM2]: Keep it formal. What key lessons-learned were determined from these examples? Are there gaps or opportunities that existing solutions do not address that can be addressed by your system?

Following information gathering and research done through online web browsing, as well as thorough examination of areas for potential improvement, the decision was made to use the best features from both systems. Those best features were the wireless capabilities of the first home weathering system and the second one's capability of charging the home weathering system at a more efficient way using solar panels and feeding back into its own battery.

3. System requirements:

- ## Use case Diagram:

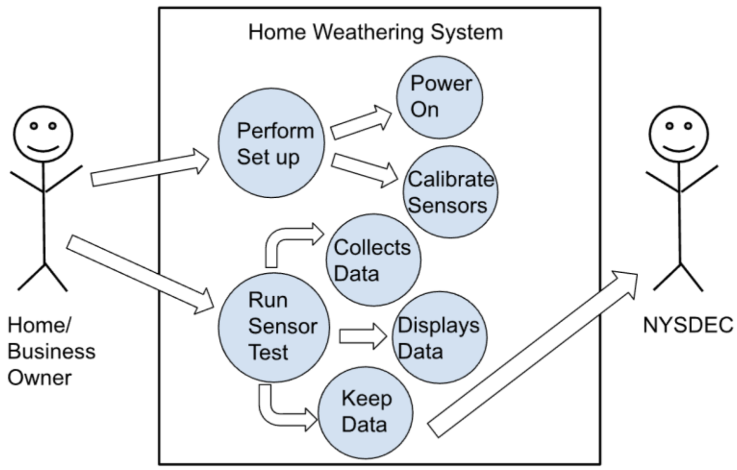


Figure 3.1: Home Weather Monitoring System Use Case Diagram.

The Use Case diagram for the Home Weather Monitoring System describes the interactions between the systems functionalities, the primary users, and external departments such as the NYDEC. The diagram identifies two principal users. A user who engages with the system to retrieve weather data, and the NYSDEC who will interact with the data transmitted from the system to analyze and monitor weather in that specific region. The primary user will perform the sensor test to collect, display, and store data. The primary user will also perform system setup like powering and calibrating sensors. NYSDEC will receive all that data that is retrieved from the sensors.

- ## Functional Requirements:

Data Processing– The system is must accurately record sensor readings and effectively interpret the readings to determine the weather conditions.

Data Transmission– The home unit must be able to send its data through Wi-Fi so that the NYDEC can access, store, and use it at their convenience.

Precision– the system must be able to read, interpret, and record data from the sensors continuously to provide NYDEC the most precise set of data.

- ## Subsystem Functional Requirements:

Power Management – The power subsystem should manage and distribute power to meet other subsystem requirements, making sure that the supply is stable and reliable without overloading the other subsystems.

Energy Storage – The power subsystem is going to store the excess energy from the solar panel in a battery for later use and to charge the battery. The excess energy can also be used as backup power when the solar panel isn't producing energy. Based on specifications, the battery should be able to store 10.05Ah after 10 hours charging at full sun. This will allow the system to operate at theoretically 1005 hours (about 1 and a half months) off the battery alone.

Reliability – Power system must minimize downtime and ensure that the rest of the weathering system will operate, even during harsh weather or home power outage.

- ## Non- Functional Requirements:

Durability – the field unit must be able to withstand harsh weather conditions over extended periods of time while ensuring that the components remain protected and undamaged.

Usability – the system should support ease of use for all homeowners, with a user-friendly UI to make the data easily accessible, and a simple installation/setup process to get the user started.

Reliability – The users should be able be able to depend on the system to consistently and accurately provide weather information from the sensors in the field unit.

- ## Subsystem Non- Functional Requirements

Efficiency – The power subsystem should work with high efficiency, lowering energy loss and optimizing power usage.

Availability – The power system should minimize downtime, ensure high availability, and maintain continuous operation of the weather monitoring system.

Compatibility -The power system should be compatible with the components of the other subsystems and a wide range of sensors.

- ## Design constraints.

Financial – There is a budget of $200. The systems are limited to a certain quality of components that can be used to build the home weathering system.

Weight – The weight of the subsystem should be reasonable. If it is excessively to light than it is susceptible to strong winds or heavy downfall. If its excessively heavy that it can cause challenges in positioning the system correctly and could potentially cause damage to the surrounding area.

Material – The selection of materials for constructing the system imposes certain constraints that must be considered.

- ## Subsystem Design constraints

Battery Capacity – The type of battery utilized determines the amount of energy that the subsystem can store and later supply power to the other subsystems.

Solar Panel – The type of solar panel will determine the available power output and voltage capacity. Also, the dimensions of the solar panel may impose a constraint on its installation and integration on the whole subsystem.

Financial – There is a limit of $200. This budget will limit us to buying certain components for the subsystem. A lower budget also means that there will be lower quality in components.

4. System Design

• Design Overview & Justifications

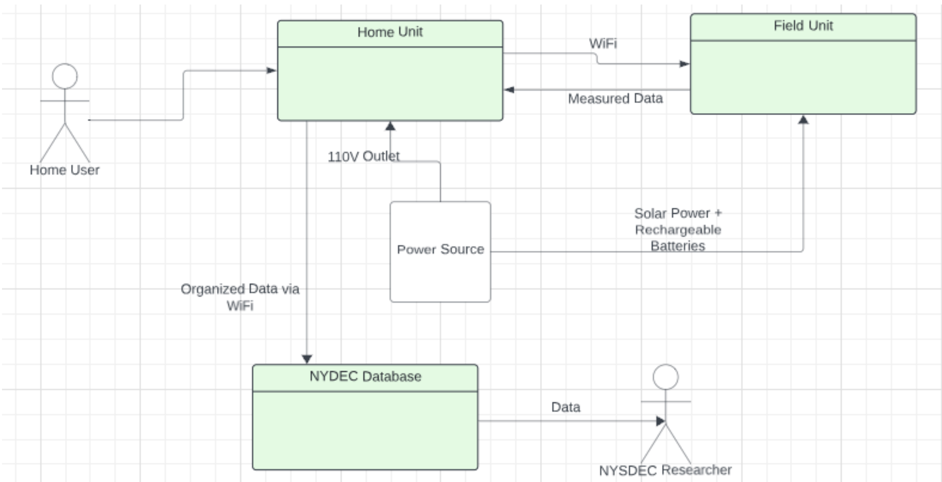


Figure 4.1: Weather Monitoring System

The design comprises a Home Unit, Field Unit, Power Source, and NYSDEC Database, facilitating efficient communication between homeowners and research scientists. Its user-friendly interface and ease of installation distinguish it, with data storage in the NYSDEC database. Potential drawbacks include heightened development costs for the power source and processing units of both Home and Field Units.

Exploration of alternative designs, such as an All-in-one unit installable on user walls, revealed benefits such as reduced development costs for power sources and processing units, along with decreased installation expenses. However, this design may suffer from suboptimal placement for data accuracy.

- Decision Matrix

	Weight	Seperated Unit	2-in-1 Unit
Cost	20%	3	1
Durability	15%	3	4
Connectivity	15%	3	5
Accessibility	20%	3	2
Accuracy	30%	5	3
Total Weighted Scores		3.6	2.85

Figure 4.2: Decision Matrix

Commented [JM3]: This would be more organized. For instance, as a single table.

While brainstorming ideas for how the system would be built whether it would be a separated unit between the home and the field units or if it would all be included together in a 2-in-1 system was ultimately based on multiple categories. Cost, Durability, Connectivity, Accessibility and Accuracy were the basis of our categories as seen in Figure 4.2. Taking cost into consideration, the system had to be affordable which is why the separated unit was more cost efficient seeing as building the field unit separately allowed for improvements whenever possible without taking a deeper cut into the fund for the project, having the system as a 2-in-1 would be more cost ineffective seeing as we would have to cut down a wall to connect both the field and home units together. Moving forward onto the durability of the system it was found that the separated unit was less effective compared to the 2-in-1, this was due to the fact that the 2-in-1 system would be connected to the house which allowed it to survive better throughout any type of weather whereas the separated system would only be as durable as the parts allowed it to be. Continuing onto connectivity the separated unit was also found to be less effective as it would require some form of wireless communication between the field unit and the home unit. The 2-in-1 unit would just connect the field unit and the home unit by simple wiring making it less complicated. Now taking accessibility into account the separated unit ended up being better in the fact that it is easier to access different microclimates since the separated field unit can be located anywhere the user wants making it easier to access different microclimates whereas the 2-in-1 system is always connected to the home unit not allowing the user to access anywhere they might want to find the microclimate of. Finally, accuracy is the biggest measure of the system since it is the most important part, the separated unit won this decision by a big margin since being allowed to put the field unit anywhere allows for sensors to poll data and not be constrained by any buildings or walls that the 2-in-1 have to go through since it is connected to the home unit.

- Data Flow Diagram

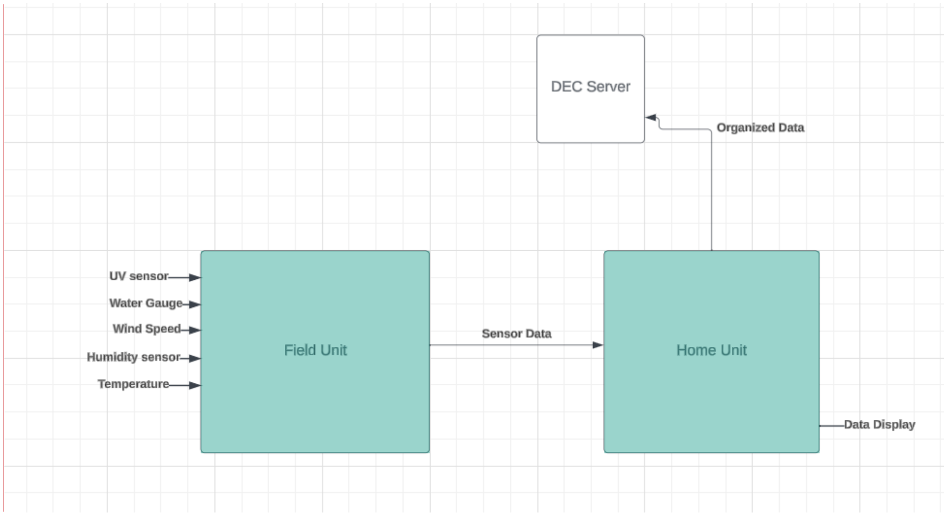


Figure 4.3: Data Flow Diagram

The system takes inputs from the various sensors and sends it to the home unit. From there, the data will be organized and verified, displayed on the screen for the homeowner to read, and sent to the DEC servers for them to use.

Commented [JM4]: You can probably just rename this data flow diagram (which is what it is). Otherwise, this looks reasonable.

Formatted: Centered

- Logical Design

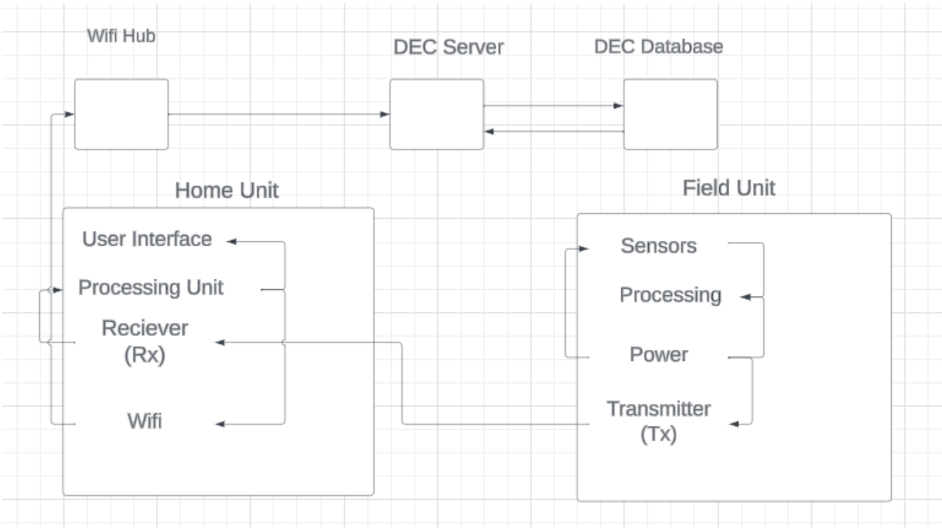


Figure 4.4: System Logical Design

The field unit will take in data from the sensors and process it to an extent, where it will then be sent via the transmitter to the home unit's receiver. After it is received, the data will be sent to the CPU for processing, displayed for the user, and sent to the DEC via Wi-Fi. The home unit is where the bulk of the data processing will take place, as the components are better protected, and it is therefore safer to use a stronger, more expensive processor.

- ## Logical Design (Power Subsystem Specific)

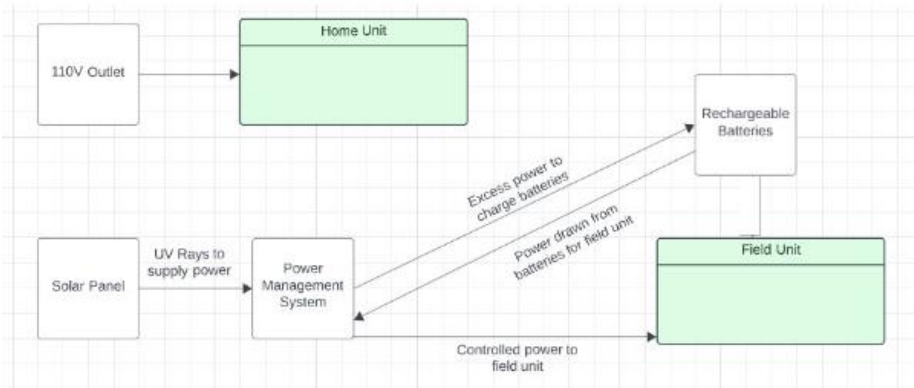


Figure 4.5: Subsystem Logical Design

The power system considers the system design, which consists of the Home and Field Unit. The Home Unit will be powered by a 110V Outlet while the Field Unit will use solar energy to power the system. The Field Unit power system consists of two parallel connected solar panels, a Power Management System (PMS) Unit and a Micro controller. The solar panels convert sunlight into 1A DC current and send it to the BMS at 3.7V. The BMS then uses the current to either charge up the battery or power up to the micro controller. The BMS also automatically switches between battery and panels to power up the micro controller when there is or isn't sufficient sunlight.

- ## Wireframe

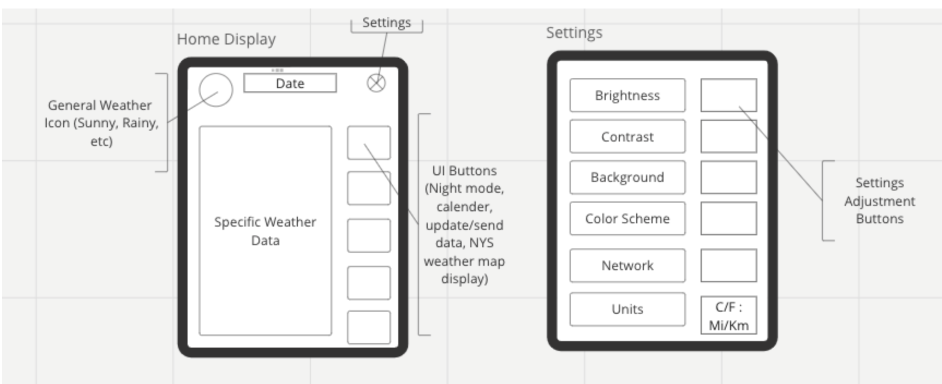


Figure 4.6: Wireframe diagram

The top left icon will be for general current conditions, such as a sun, a cloud with rain, or a cloud with snow. Support for importing a file to change the background can also be incorporated.

- Physical Design (Power Subsystem Specific)

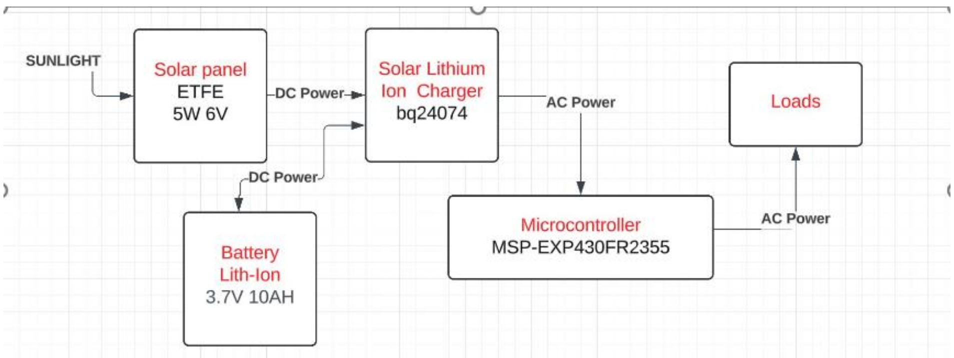


Figure 4.7: Physical Design Diagram

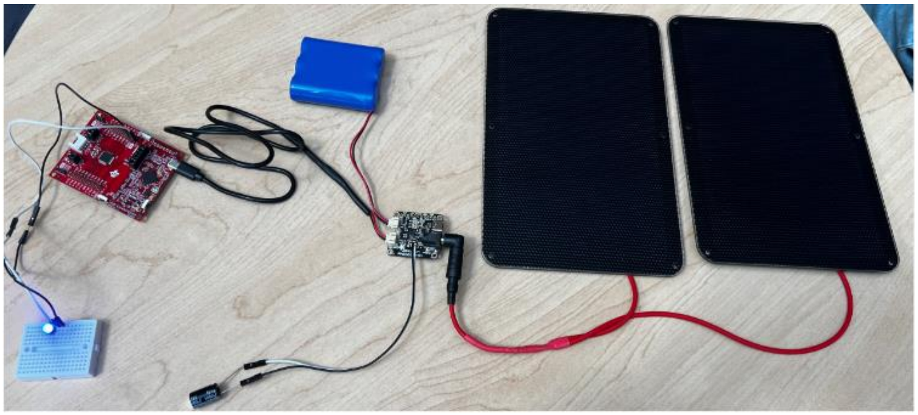


Figure 4.8: Physical Power Supply

After carefully researching the available components on the market in person and online we decided on the parts seen in figure 4.8. We start by selecting the Voltaic System 5W 6V ETFE Solar Panel as our main source of energy collection. It was decided that the ridged exterior for the solar panel would lead to a higher durability, which is especially important for facing multiple elements of weather. Regarding the decision using the Adafruit bq24074 Solar Lithium Ion Charger, inverting and storing the DC power collected from the solar panel and stored in the

Adafruit Lith-Ion 3.7v 10AH as AC power is the most efficient given the solar charger gave way to inverting power without the use of a breadboard or protoboard. It also allowed for an efficient way of connecting any load presented to the charger using any sort of microcontroller or a breadboard. Finally, the decision of using the Texas Instruments MSP430FR2355 microcontroller as the main source of powering and gathering data from any loads connected to it was based on the MSP430's power usage efficiency since it allows the user to control the amount of current drawn from it.

- ## Physical Casing Design / Prototype

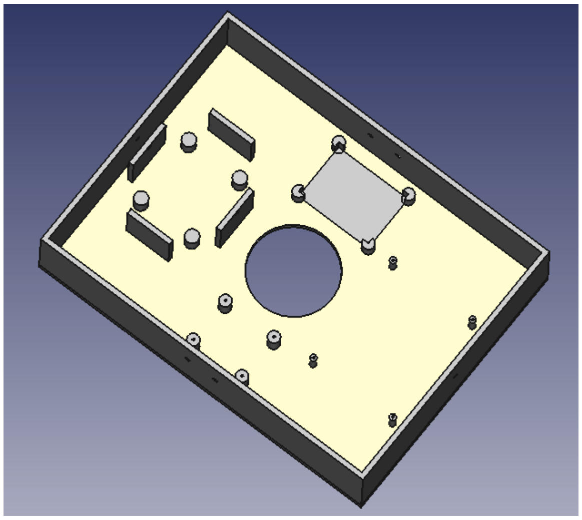


Figure 4.9: Component Box

Above is the CAD schematic for the main components of the system. It contains standoffs for the microcontroller and PMS PCBs, and compartments for the batteries and a small breadboard. The battery compartment also contains standoffs to allow more airflow to dissipate more heat.

- Final Physical Casing Design:

Figure 4.10: Physical Power Supply

The figure above describes the final casing for the subsystem. This design was chosen because of its portability and functionality aspects. This design allows the subsystem to be weather protected with enclosed casing while maximizing the power input by enabling the solar panels to receive the optimal amount of sunlight and reach the Maximum Power Point (MPP).

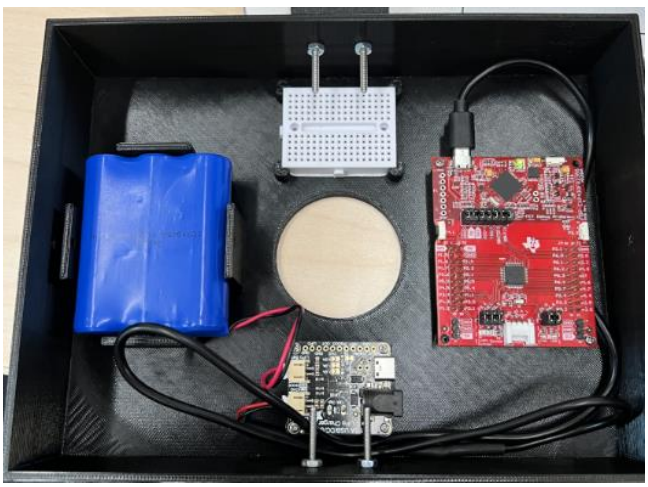
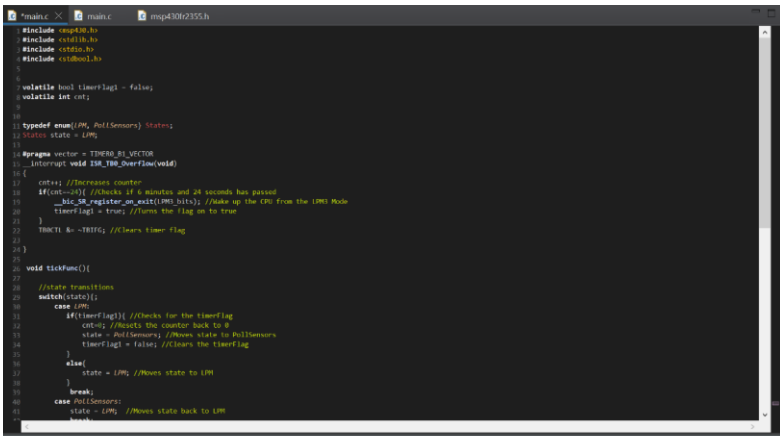


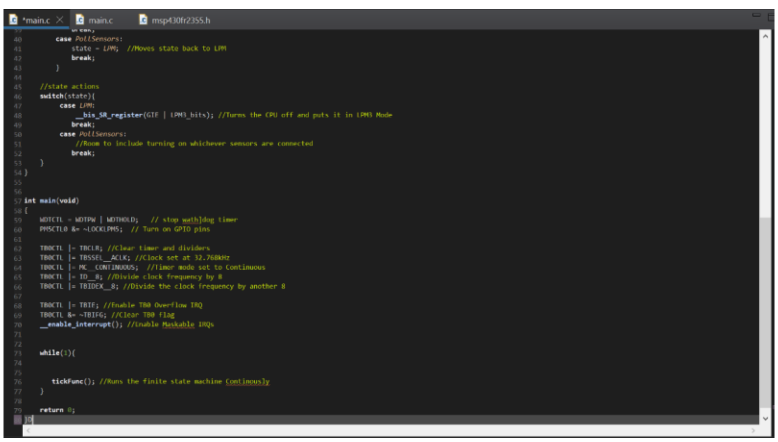
Figure 4.11: Inside View of the Casing

5. Analysis:

- ## Programming & Logic



```c
#include <msp430.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>


volatile bool timerFlag1 = false;
volatile int cnt;


typedef enum{LPM, PollSensors} States;
States state = LPM;

#pragma vector = TIMER0_B1_VECTOR
__interrupt void ISR_TB0_Overflow(void)
{
    cnt++; //Increases counter
    if(cnt==24){ //Checks if 6 minutes and 24 seconds has passed
        __bic_SR_register_on_exit(LPM3_bits); //Wake up the CPU from the LPM3 Mode
        timerFlag1 = true; //Turns the flag on to true
    }
    TB0CTL &= ~TBIFG; //Clears timer flag

}

 void tickFunc(){

    //state transitions
    switch(state){;
        case LPM:
           if(timerFlag1){ //Checks for the timerFlag
               cnt=0; //Resets the counter back to 0
               state = PollSensors; //Moves state to PollSensors
               timerFlag1 = false; //Clears the timerFlag
           }
           else{
               state = LPM; //Moves state to LPM
           }
            break;
        case PollSensors:
            state = LPM;  //Moves state back to LPM
```



```c
        case PollSensors:
            state = LPM;  //Moves state back to LPM
            break;
        }

    //state actions
    switch(state){
         case LPM:
             __bis_SR_register(GIE | LPM3_bits); //Turns the CPU off and puts it in LPM3 Mode
            break;
         case PollSensors:
             //Room to include turning on whichever sensors are connected
            break;
    }
}


int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;   // stop wath]dog timer
    PM5CTL0 &= ~LOCKLPM5;  // Turn on GPIO pins

    TB0CTL |= TBCLR; //Clear timer and dividers
    TB0CTL |= TBSSEL__ACLK; //Clock set at 32.768kHz
    TB0CTL |= MC__CONTINUOUS;  //Timer mode set to Continuous
    TB0CTL |= ID__8; //Divide clock frequency by 8
    TB0CTL |= TBIDEX__8; //Divide the clock frequency by another 8

    TB0CTL |= TBIE; //Enable TB0 Overflow IRQ
    TB0CTL &= ~TBIFG; //Clear TB0 flag
    __enable_interrupt(); //Enable Maskable IRQs


    while(1){


       tickFunc(); //Runs the finite state machine Continously
    }

    return 0;
}
```

Figure 5.1: MSP430FR2355 Code

For the system, the intention was to put the MSP430FR2355 Microcontroller into a Low Power Mode, more specifically Low Power Mode 3(LPM3) or “standby” to design the system as the most power efficient possible. After research the microcontroller was found drawing 350 µA when it is completely operational and 1.5 µA when it goes into the LPM3 mode. To implement the LPM3 mode via code, it simply followed a finite state machine in which the system starts with the MSP430 already in a LPM3 mode as seen in figure 5.1. After approximately 6 minutes and 24 seconds, the CPU is awakened from its low power mode. Once awakened, the system polls the sensors for another 6 minutes and 24 seconds or before automatically returning to LPM3. This cycle repeats indefinitely.

```c
#include <msp430.h>
#include <stdlib.h>
#include <stdio.h>
#include <stdbool.h>


volatile bool timerFlag1 = false;
volatile int cnt;


typedef enum{LPM, PollSensors} States;
States state = LPM;

#pragma vector = TIMER0_B1_VECTOR
__interrupt void ISR_TB0_Overflow(void)
{
    cnt++; //Increases counter
    if(cnt==24){ //Checks if 6 minutes and 24 seconds has passed
        __bic_SR_register_on_exit(LPM3_bits); //Wake up the CPU from the LPM3 Mode
        timerFlag1 = true; //Turns the flag on to true
    }
    TB0CTL &= ~TBIFG; //Clears timer flag

}

 void tickFunc(){

    //state transitions
    switch(state){;
        case LPM:
            if(timerFlag1){ //Checks for the timerFlag
                cnt=0; //Resets the counter back to 0
                state = PollSensors; //Moves state to PollSensors
                timerFlag1 = false; //Clears the timerFlag
            }
            else{
                state = LPM; //Moves state to LPM
            }
             break;
        case PollSensors:
             state = LPM;  //Moves state back to LPM
```

```c
             break;
        }

    //state actions
    switch(state){
        case LPM:
            P3OUT |= BIT5;  //Truns on Red LED
            P3OUT &= ~BIT2; //Turns off Blue LED
            P3OUT |= BIT1; //Turns on Green LED
            __bis_SR_register(GIE | LPM3_bits); //Turns the CPU off and puts it in LPM3 Mode
           break;
        case PollSensors:
            //Room to include turning on whichever sensors are connected
            P3OUT &= ~BIT5; //Turns off Red LED
            P3OUT |= BIT2; //Turns on Blue LED
            P3OUT &= ~BIT1; //Turns off Green LED
            __delay_cycles(500000); //Delay by 0.5 Seconds
            P3OUT &= ~BIT2;  //turns off Blue LED
            __delay_cycles(500000); //Delay by 0.5 Seconds
            P3OUT |= BIT2; //Turns on Blue LED
            __delay_cycles(500000); //Delay by 0.5 Seconds
            P3OUT &= ~BIT2; //Turns off Blue LED
           break;
    }
}


int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;   // stop wath]dog timer
    PM5CTL0 &= ~LOCKLPM5;  // Turn on GPIO pins

    TB0CTL |= TBCLR; //Clear timer and dividers
    TB0CTL |= TBSSEL__ACLK; //Clock set at 32.768kHz
    TB0CTL |= MC__CONTINUOUS;  //Timer mode set to Continuous
    TB0CTL |= ID__8; //Divide clock frequency by 8
    TB0CTL |= TBIDEX__8; //Divide the clock frequency by another 8

    TB0CTL |= TBIE; //Enable TB0 Overflow IRQ
    TB0CTL &= ~TBIFG; //Clear TB0 flag
    __enable_interrupt(); //Enable Maskable IRQs
```

```
P3DIR |= BIT5;//makes P3.5 an output
P3DIR |= BIT2;//makes P3.2 an output
P3DIR |= BIT1;//makes P3.1 an output
P3OUT &= ~BIT5; //turns off RLED
P3OUT &= ~BIT2; //turns off GLED
P3OUT &= ~BIT1; //turns off BLED


while(1){


    tickFunc(); //Runs the finite state machine Continously
}

return 0;
}
```

Figure 5.2: MSP430 Test Code

To test the system, a simple circuit connecting 3 LEDs to a breadboard and the MSP430 is needed. This allowed the system to be tested by simply making the red and green LED stay on while the system was in LPM3 mode, when the system woke up the red and green LEDs turned off and then the blue LED started to blink to show the time in which the sensors or anything connected would be turned on and ran. Afterwards the system went back into LPM3 mode and ran another cycle of the red and green LEDS turning on until the blue light started blinking. All this code is in figure 5.2.

- ## Specifications:

Solar Panel	
Prated(W)	5.49
Vrated(V)	6.07
PratedMax(W)	10.98

Figure 5.3: Solar Panel Specifications

The solar panel is rated for 5.49W with a voltage of 6.07V. Since the subsystem uses 2 panels to supply the other parts of the system, the total rated power will be 10.98W.

However, since the BMS can only take 4.4V from the panel, the battery can only draw 1A at 3.7V. The system output will become:

Vcharging(V)	3.7
Icharging (A)	1
Battery Capacity(Ah)	10.05
Tcharging(theoretical)(h)	10.05
VoutMax(V)	3.7

Figure 5.4: Subsystem Specifications

Vcharging: Voltage input/output of the battery.

Icharging: Current in/out of the BMS.

Battery Capacity: Measured in Ah, the amount of X (A) currents the battery will be able to supply for one hour.

Tcharging: Total charging time (assuming no shading). This is calculated by dividing the Battery Capacity by the Charging current.

VoutMax: Maximum voltage output of the system.

Based on collaboration with the other teams, (4B), the other subsystems will require approximately:

Sensor:	
Vsensor(V)	3.3
Isensor(A)	0.0025

Assuming 4 sensors	
Vsensor(V)	3.3
Isensor(A)	0.01
Theoretical uptime (h)	1005

Figure 5.5: Power Output Requirements

Based on the power supply specifications and anticipated power required from other subsystems, the power supply will be able to power the system for 1005 hours (about 1 and a half months) without sunlight. This satisfies the energy storage requirements.

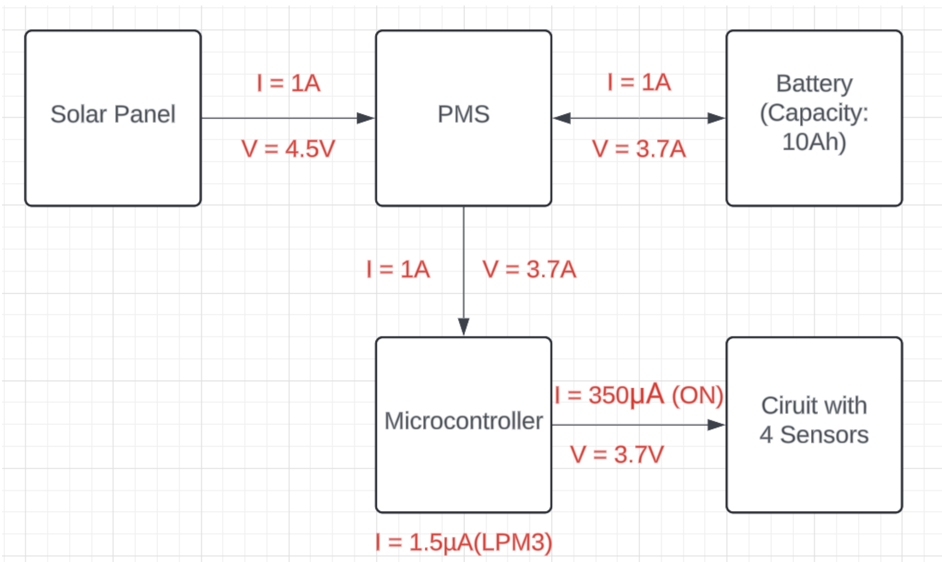


Figure 5.6: Power Supply Subsystem Specification.

Above is the final specification of the power supply subsystem, despite being rated for 6V, the solar panels were consistently output 4.5V and since the battery only rated for 3.7V, the system is limited to output 3.7V at 1A. This theoretically results in the system being able to supply the other systems for 1005 hours (about 1 and a half months) without needing to be recharged by sunlight.

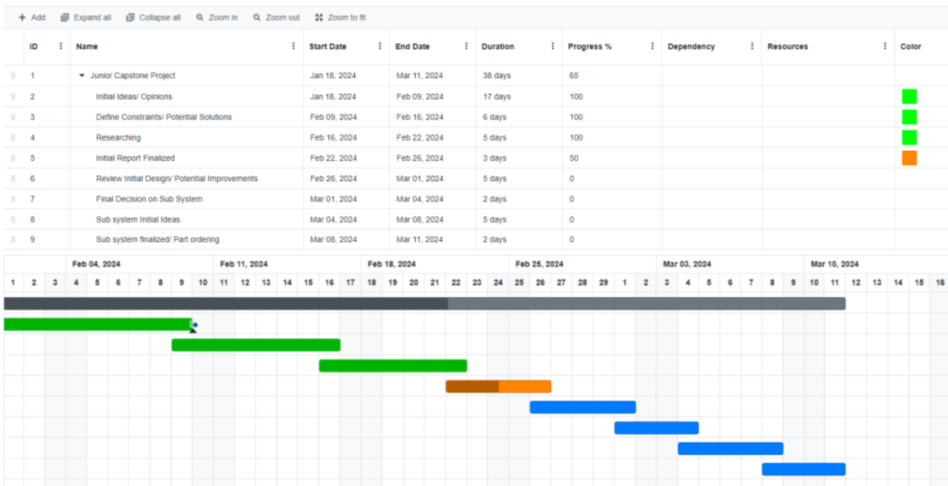
6. Bill of Materials

Part	Purpose	Cost
Adafruit Universal USB/DC/Solar Lithium Ion/Polymer Charger	Inverts the DC power from the Solar Panel and converts it into AC to charge up the battery	$14.95
JST PH 2-Pin Cable – Female Connector 100mm	Soldered to a Micro USB cable to turn on the MCU from the the Solar Chargers LOAD OUT	$0.75
USB A to Micro B	Used in conjunction with the JST cable	$2.99
Battery LITH-ION 3.7V 10AH	Connected to the Solar charger to store the power	$29.95
2x 5 Watt 6 Volt Solar Panel – ETFE	Solar Panel to power the system	$70.00
Texas Instruments MSP430FR2355	The MCU to power on and control the sensors	$12.99
4700 µF Electrolytic Capacitor	Used to stabilize the DC input into the solar charger	$1.95
	TOTAL	$133.58

Figure 6.1: Bill of Materials

7. Semester Planning

Gannt Chart



ID	Name	Start Date	End Date	Duration	Progress %	Dependency	Resources	Color
1	Junior Capstone Project	Jan 18, 2024	Mar 11, 2024	38 days	65			
2	Initial Ideas/ Opinions	Jan 18, 2024	Feb 09, 2024	17 days	100			
3	Define Constraints/ Potential Solutions	Feb 09, 2024	Feb 16, 2024	6 days	100			
4	Researching	Feb 16, 2024	Feb 22, 2024	5 days	100			
5	Initial Report Finalized	Feb 22, 2024	Feb 26, 2024	3 days	50			
6	Review Initial Design/ Potential Improvements	Feb 26, 2024	Mar 01, 2024	5 days	0			
7	Final Decision on Sub System	Mar 01, 2024	Mar 04, 2024	2 days	0			
8	Sub system Initial Ideas	Mar 04, 2024	Mar 08, 2024	5 days	0			
9	Sub system finalized/ Part ordering	Mar 08, 2024	Mar 11, 2024	2 days	0			