Senior Design

Project Document:

Bee Hive Monitoring System

Table of Contents:

Section 1: Overview

1.1 Executive Summary:

We will build a beehive monitoring system for our project that shows analytics about the hive. The system will demonstrate the interior and external temperatures as well as read the hive's humidity on the inside. Furthermore, our system will be monitoring the weight of the hive and will allow transfer of data 450 ft away from the hive receiver. The data that is collected will be accessible through a web page though this data will also be stored on the microcontroller as well in case the receiver loses connection with the data collection module. The system will be qualified to run all throughout the night and will recharge the batteries with solar panels during the day.

At the moment we are finishing researching a number of methods to complete the project as well as different components that will be necessary. The center of our project will be an Arduino microcontroller that will direct power to the sensors as well as collect, store, and transmit all of the sensor data. While normally powering this number of sensors from a single microcontroller board, we are working on ways to reduce the amount of current that is flowing through the board.

1.2 Team Contact & Protocols:

- **Porter Obrist**

obristp@oregonstate.edu

Expected Contributions:

- Arduino Programming
- 3D Modeling
- Temp and humidity sensor control

- **Tieying Chu**

chuti@oregonstate.edu

Expected Contributions:
- Arduino Programming
- RF
- Module Circuit Design
- PCB Design

- **Justin Isert**

isertj@oregonstate.edu

Expected Contributions:
- Data Transmission
- PCB Design

- **Saud Alaffasi**

alaffass@oregonstate.edu

Expected Contributions:
- Power supply and solar panel
- PCB Design
- Sensors

Team protocols:
Meet with group at least once a week
Ask for help whenever it is needed
Keep up good communication
Have written records for our meetings

Table 1.2

1.3 Gap Analysis:

Our project exists to allow for the collection and storage of data collected from beehives. Since it is very difficult to judge what state a hive is in solely from looking at it from the outside, this project would allow for users to view a few different properties of their hive from a distance.

This project will be able to help track the health of hives and allow for beekeepers to spot when a hive might be in need of attention without checking it physically. Allowing them to target their care to specific hives and reduce time spent checking on healthy hives. On top of allowing beekeepers to check on hives remotely, our system will store data and give a view of how much fluctuation has occurred while the system was active.

We are assuming that our system will have to collect and present both past and current data in a visually appealing way, allow variable collection intervals, retain sufficient power throughout a full night, be easily programmable, and have a simple installation and setup process. After discussing our project we found that the majority of the requirements for this project were abstract. Allowing for us to make creative decisions when it comes to presentation of the collected data, enclosure location with respect to the hive, installation, and placement of the sensors.

1.4 Timeline:

Image 1.4(Timeline)

https://docs.google.com/spreadsheets/d/1GfDXRChYGX-2czrFWeRBQ0l5ZDn9gjMIoDtU5g3KfbE/edit?usp=sharing

1.5 References/File Links:

[1] "IEEE Editorial Style Manual (Online)." [Online]. Available: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/conferences/style_references_manual.pdf . [Accessed: 15-Oct-2022].

1.6 Revision Table:

Name	**Date**	**Revision made**
-	10/14/22	Section created
Porter	10/20/22	Updated picture on project timeline
Porter	11/4/22	Updated team protocols to table
Porter	11/4/22	Added table of contents
Justin	11/4/22	Adjusted reference list format to align with IEEE standards
Porter	11/17/22	Updated timeline
Justin	11/17/22	Updated Executive summary
Saud	11/17/22	Updated team protocols to table

Table 1.6

Section 2: Impacts and Risks

2.1 Design Impact Statement:

Having a system to remotely monitor a hive can help to reduce the risk of getting stung [1], and this less intrusive method of measuring the internal status of a hive can improve the overall health of the hive. Due to the hives being kept outside, there is also a risk of electrocution due to short-circuiting caused by water coming into contact with the electrical components of the system. This risk can be reduced with a robust enclosure to keep the internal components as dry as possible. RF radiation produced by the transmitter and receiver blocks could have an effect on the overall health of the hive [2], but the use of a directional antenna can mitigate the exposure to the hive.

The production and disposal of rechargeable batteries as well as solar panels both have a negative impact on the environment. Mining the materials needed and refining them to electronics-grade quality both have the potential to damage the environment in significant ways [3]. With very few alternatives for this problem, we have taken the necessary steps to maximize the lifespan of both of these components, and researched ways for the proper disposal of E-waste.

Having a beehive in proximity to certain crops can also lead to an increased yield, due to bees being pollinators [4]. With the reduction of wild bee population, some farms could benefit from having a hive of pollinators to ensure their crops are fully pollinated and have the highest yield possible. For those who are more focused on caring for a hive in an effort to harvest honey, having a system to track the health of a hive can allow beekeepers to make more well-informed decisions on how to spend their time and money on improvements for their hives. Reducing the chance of wasting money on a change to the hive that is not necessary.

2.2 Risks:

Risk ID	***Risk Description***	***Risk category***	***Risk prob.***	***Risk impact***	***Performance indicator***	***Action Plan***
R1	Group member can't attend meeting	Communication	L	L	Team member send absent message	- Reschedule a time for meeting - List the work team need to complete

						and adjust the timeline
R2	Shipping delay on components	Timeline	M	M	Lack of parts	- Order the part from alternative provider - Use the module as a temporary replacement
R3	Shipping delay on PCB	Timeline	L	H	Lack of PCB	- Use breadboard as a temporary replacement
R4	PCB logic mistake	Technical	M	M	PCB not working correctly	- Changing the connection and order another PCB board
R5	Going over budget	Cost	M	L	Pay more than we expected	- Make the BOM plan as detailed as possible
R6	Unable to test enclosure on active hive	Technical	M	L	Data inaccessible	- Use the hive without bees to mimic the real environment as much as possible for testing.
R7	Component Breaks	Technical	M	L	Part doesn't work or data inaccessible	- Order extra components for safety
R8	Unavailability/lack of sunlight for solar panels	Environmental	H	L	System can't last as long as expected	- Prepare backup batteries for test - Test the solar panel performance separately
R9	Damaged microcontroller	Technical	M	L	Microcontroller no longer working properly	- Will order spare arduino mega boards
R10	Bees interfere with internal or external sensors	Technical	M	M	Sudden sensor failure	- Enclose sensors so bees can not access them

					Buildup of beeswax on sensors	- Place sensors in a non-intrusive way

Table 2.2

2.3 References and File Links:

[1] E. Ntawuzumunsi, S. Kumaran, and L. Sibomana, “Self-Powered Smart Beehive Monitoring and Control System (SBMaCS),” Sensors, vol. 21, no. 10, p. 3522, May 2021, doi: 10.3390/s21103522.

[2] SNS, “Cell phones play key role in decline of bees,” The Statesman, Dec. 31, 2021. https://www.thestatesman.com/lifestyle/cell-phones-play-key-role-decline-bees-1503035017.html (accessed Nov. 04, 2022).

[3] Kuby, “The Positive and Negative Environmental Impacts of Solar Panels,” Kubyenergy.ca, 2019. https://kubyenergy.ca/blog/the-positive-and-negative-environmental-impacts-of-solar-panels

[4] D. M. Bauer and I. Sue Wing, “The macroeconomic cost of catastrophic pollinator declines,” Ecological Economics, vol. 126, pp. 1–13, 2016.

2.4 Revision table:

Name	**Date**	**Revision made**
-	11/4/22	Section created
Porter, Tieying, Justin	11/17/22	Expanded risk table
Justin	4/28/23	Added Design Impact Statement and expanded references

Table 2.4

Section 3: Top-Level Architecture

3.1 Block Diagram:

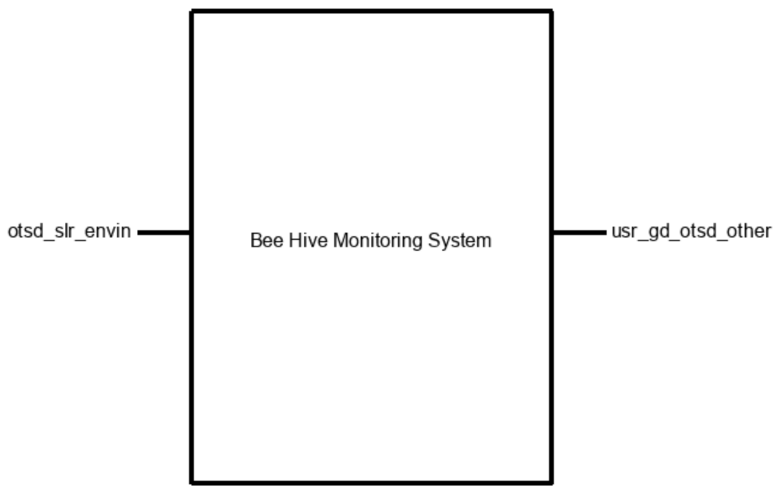


Image 3.1.1(Black Box Diagram)

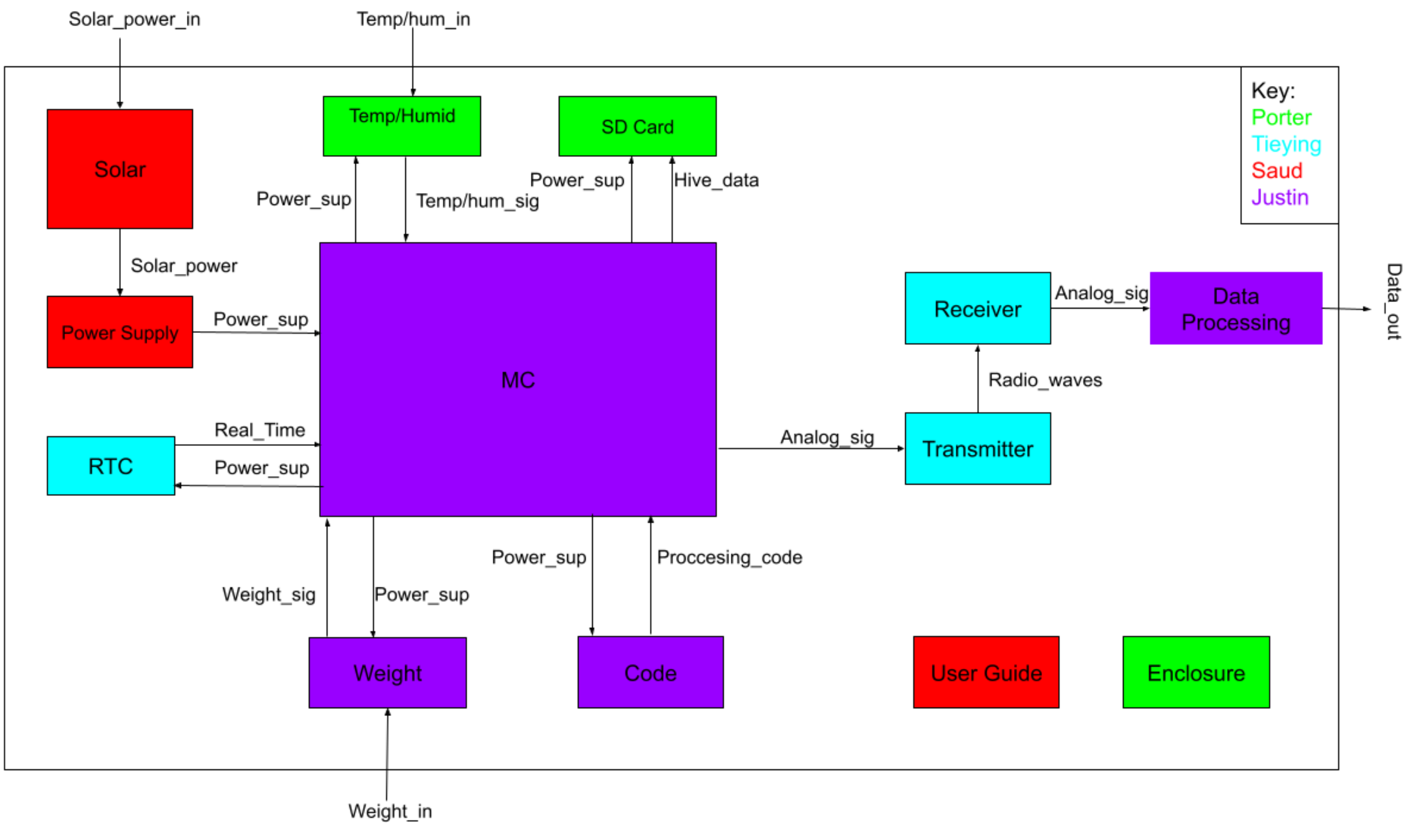


Image 3.1.2(System Block Diagram)

3.2 Block Descriptions:

Name	Description
Temperature/Humidity Champion: Porter Obrist	This block will receive 5V of power from the 5V pin on the microcontroller. It will read data in at least 10 second intervals for both the temperature and humidity inside the beehive, as well as the temperature outside the beehive and send this data to the microcontroller with a digital signal through a signal bus.

Enclosure Champion: Porter Obrist	The enclosure block will include the designing and 3D printing of an enclosure that will encase all the components such as the battery, the microcontroller, and the PCB, as well as create a casing for the sensors that will be inside the hive as to protect them from the bees and their honey.
Solar Champion: Saud Alaffasi	The solar panel helps the system even when the power gets off. After receiving much energy from the sun the solar panel could maintain the battery onboard enough to last through the night.
Power Supply Champion: Saud Alaffasi	It is recommended that we choose a 12V battery with enough capacity and current to support all of our system's components in order to guarantee a steady and dependable power supply. In order to control battery power and avoid overcharging, an MPPT solar charger controller must also be included. The charger controller will be attached to the battery and connected between the battery and the solar panel. It will efficiently manage the energy flow to the battery, ensuring it works within safe limits. It is crucial to remember that choosing a battery with adequate capacity and current, as well as utilizing a trusted and high-quality MPPT solar charger controller, will help guarantee the longevity and efficiency of our power supply system.
User Guide Champion: Saud Alaffasi	A user guide is to assist users in using the product. As the product developers write it, it contains guides and images to help the user.A user guide is a book written to help people utilize a product. It is frequently authored by developers and includes comprehensive instructions, step-by-step guidance, and visual aids to enable consumers to understand and efficiently using the product.
Weight Champion: Justin Isert	Supports the hive and measures the overall weight of it's contents (honey, bees, beeswax, etc.). Includes the load cells as well as the plate that the hive would be set on top of.
Code Champion: Justin Isert	The code will determine how often inputs are taken from the sensors, compile the sensor data together, and

	choose when to send it to the receiver. The code will also have to recognize when the receiver is disconnected and choose to store said sensor data instead of sending it.
Data Processing Champion: Justin Isert	This block will take in the sensor data from the receiver and compile the current and past data into a graph for the user to view.
Transmitter Champion: Tieying Chu	The module is directly connected to the microcontroller, which will send the data collected by the microcontroller to the receiver via radio waves.
Receiver Champion: Tieying Chu	The module is connected to a PC and is used to receive data files from the microcontroller.
RTC Champion: Tieying Chu	This module is responsible for creating a timestamp for each piece of environmental data collected by the microcontroller.
Microcontroller Champion: Justin Isert	Takes data input from all of the sensors and outputs data to transmitter. Also will recognize if the receiver is offline and save sensor data until the receiver is online once more.
SD card Champion: Porter Obrist	This block will be responsible for storing the data that is received from the weight and temperature/humidity blocks onto an SD card so that the data for the system is backed up incase transmission is lost.

Table 3.2

3.3 Interface Definitions:

Name	Properties
otsd_slr_envin	
tmprtrhmdty_mcrcntrllr_data	● Datarate: 1 sample per 5 seconds ● Messages: Values for both temperature and

	humidity ● Protocol: Single-Wire
slr_pwr_spply_dcpwr	● Inominal: 0.25 A ● Ipeak: 0.9 A ● Vmax: 13.6V ● Vmin: 8V
pwr_spply_mcrcntrllr_dcpwr	● Inominal: 130 mA ● Ipeak: 170 mA ● Vmax: 13.6 V ● Vmin: 10.8 V
usr_gd_otsd_other	● Other: 9 out of 10 users can agree that is readable and easy to follow. ● Other: Contain multiple sections that helps the users. ● Other: Contains at least five pages.
wght_mcrcntrllr_data	● Datarate: 1 Sample Per Second ● Messages: Measured Weight ● Protocol: Single-Wire
cd_mcrcntrllr_data	● Datarate: 1 line per second ● Messages: Sensor Data ● Protocol: CSV file
trnsmttr_rcvr_rf	● Datarate: Transmitter work at channel 0 and data rate is 250kbps ● Messages: It can transmit data collected by microcontroller, like timestamps, temperature, humidity, etc.

	• Other: Transmit data at least 450 feet away
rcvr_dt_prcssng_asig	
rtc_mcrcntrllr_dsig	• Other: Output accurate timestamps • Other: Resolution: seconds to years • Vmax: 5V • Vmin: 0V
mcrcntrllr_tmprtrhmdty_dcpwr	• Inominal: 1.25mA • Ipeak: 1.5mA • Vmax: 6V • Vmin: 3.3V • Vnominal: 5V
mcrcntrllr_wght_dcpwr	• Inominal: 1.5mA • Ipeak: 4mA • Vmax: 5.5V • Vmin: 2.6V
mcrcntrllr_cd_data	• Datarate: 1 sample per sensor per second • Messages: Sensor Data • Protocol: Single Wire
mcrcntrllr_trnsmttr_dcpwr	• Inominal: 0.05 mA • Ipeak: 0.5 mA • Vmax: 3.6V • Vmin: 2.7V • Vnominal: 3.3V
mcrcntrllr_trnsmttr_dsig	• Logic-Level: Active High

	• Other: Use SPI protocol to transmit data • Vmin: 0.6 V • Vnominal: 5 V
mcrcntrllr_rtc_dcpwr	• Inominal: 150nA • Ipeak: 0.1mA • Vmax: 5.5V • Vmin: 2.5V • Vnominal: 5V/3V(backup)
mcrcntrllr_sd_crd_dcpwr	• Inominal: 100mA • Ipeak: 200mA • Vmax: 5.5V • Vmin: 4.5V • Vnominal: 5V
mcrcntrllr_sd_crd_data	• Datarate: 1 sample per 5 seconds • Messages: Weight, temperature, and humidity • Protocol: SPI(Serial Peripheral Interface)

Table 3.3(Interface definitions from student portal)

3.4 References and File Links:

3.5 Revision table:

Name	Date	Revision made
-	3/8/23	Section created
Porter	3/9/23	Added content from student portal into section 3.1, 3.2, 3.3

Table 3.5

Section 4: Block Validations

4.1 Temperature/Humidity Block:

4.1.1 Description -

This block will be the part of the system responsible for retrieving the data for both the temperature and humidity on the inside and the outside of a beehive. This block will send its data to the microcontroller to be stored on an SD card as well as be transmitted along with the other data from the beehive such as the weight. After thorough research on different devices that could possibly work for this block and for the needs of our system, it was decided to use the DHT22/AM2302 which is a temperature and humidity sensor combined into one device, so it is able to transmit both values of data through one connection. This device was decided to be the best choice as it allows us to have the least amount of components going into the beehive, compared to if we would have done a separate sensor for both temperature and humidity, as well as it would be easier to work with one device for both measurements of temperature and humidity with one input pin on the microcontroller.

This Temperature/Humidity block will be receiving 5V of power from the 5V pin on the microcontroller in order for it to gather its data on the temperature and humidity values in which it will be able to do so in a minimum of 10 second intervals. This device will also have one connection to an input pin on the microcontroller, this connection will allow transmission of both the temperature and humidity data from the DHT22 device over to the microcontroller.

4.1.2 Design -

This section contains a black box diagram of the Temperature/Humidity block(Fig.1) for our system, showing what interfaces/connections this block includes. This section also contains a Wiring Diagram(Fig. 2) showing how one could connect the DHT22 to an Arduino Uno microcontroller to test it. And lastly, this section includes some Pseudocode(Fig. 3) that entails how one can use the Arduino Integrated Development Environment to collect the temperature and humidity values from the DHT22 sensor using the Arduino Uno and to read these values on a computer connected to the Arduino.

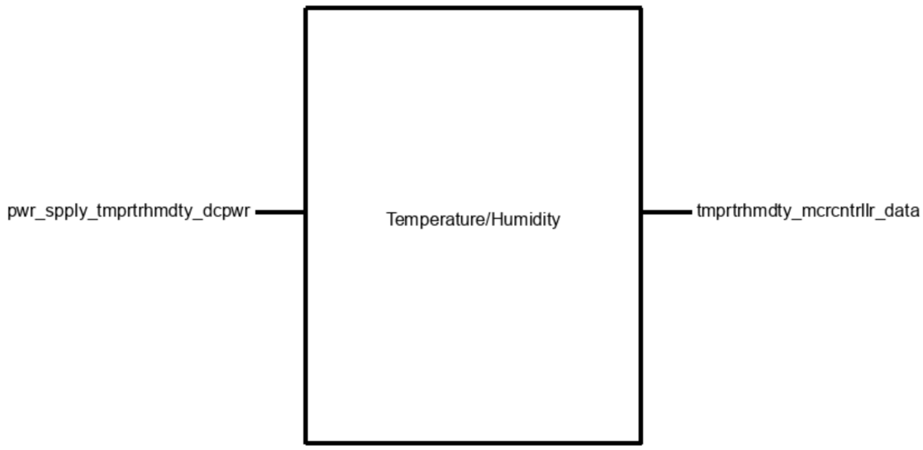
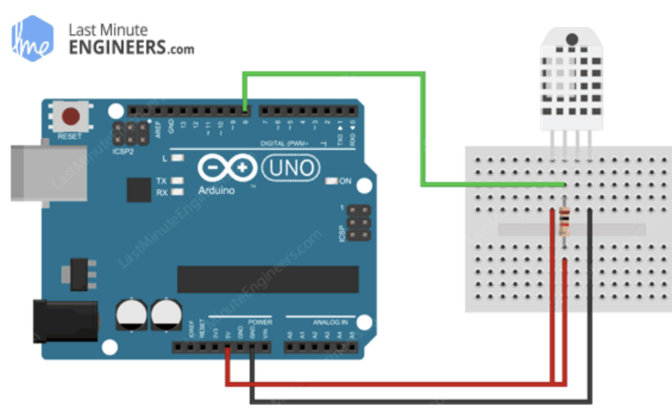


Fig. 1. Black Box Diagram of Temperature/Humidity Block

In the black box diagram above it is shown that the block has two interfaces connected to it, the first being the DC power connection between the Temperature/Humidity block and the microcontroller's 5V pin. This will be an input into the DHT22 sensor in which it will receive its 5V of power to be able to gather its data. The second interface shown in this black box diagram is the data connection between the Temperature/Humidity block and the microcontroller in which this will be an output from the DHT22. This output will be used to send the data of both the temperature and humidity values to the microcontroller using a digital signal via a single-bus.



Wiring DHT22 to Arduino UNO

Fig. 2. Wiring Diagram

[1] Interfacing DHT11 and DHT22 Sensors with Arduino

In the Wiring Diagram above it is shown how to connect the DHT22 sensor to the Arduino Uno microcontroller, this is done by using a breadboard and first connecting the power pin of the DHT22 to the 5V pin on the Arduino, as well as the GND pin of the DHT22 to the GND of the Arduino. Next it shows that you connect the data pin from the DHT22 to input pin 8 on the Arduino, but this does not need to be connected to pin 8 and can be connected to any input pin that you plan to use on the Arduino. And lastly the diagram shows that you need a 10k resistor connected between the data pin on the DHT22 and the 5V pin from the arduino, this is a pull-up resistor and is used ensure that the data line is at a known logic level when the sensor is not actively transmitting data.

```
Include DHT library
define data pin #
create DHT object
setup function{
call serial.begin()
}end setup function
setup running loop{
create int for reading data from sensor
create float and store temp data
create float and store humidity data
print temp converted to Fahrenheit from celsius
print humidity
}end running loop
```

Fig. 3. Pseudocode for Arduino IDE

This figure above is a short piece of pseudocode showing how one can use Arduino IDE along with an Arduino Uno to measure and read the values from the DHT22 sensor.

4.1.3 General Validation -

The reason that this block fits well within our systems design is first off because our system needed to be able to measure both the temperature and humidity on the inside of the beehive and the device that we chose to use being the DHT22, incorporates both a temperature sensor and a humidity sensor in one. Using this type of device instead of using two separate devices for the temperature and humidity measurements allows for our system to have less components going into the actual beehive which would cause less distribution to the bees in their habitat. Choosing this device also allows for there to be less connections to the input pins on the microcontroller as both streams of data are transmitted through one pin with this device. Also meaning there are less wires needed to be squeezed through the small gap in the beehive in order for the sensor to sit inside.

Another reason why this block fits well within our systems and we chose this device for this block is because the DHT22 is a widely used device that has plenty of documentation on the web, making it easy to learn how to use this device. This device also allows for simple use with the microcontroller that our system will be incorporating being the Arduino Mega. Since the DHT22 is such a popular sensor to use it has its own library within the Arduino IDE software making it very simple to write the code and retrieve the data from the sensor. This block also fits well within our systems design as the DHT22 sensor has a typical voltage of 5V which goes along again with our choice in microcontroller, as the Arduino Mega has a 5V pinout which is what our system is planning on using for most components.

For our systems design since it needs to accurately measure the temperature inside and outside the hive as well as the humidity inside the hive we needed to choose a device that would be able to measure within an appropriate range for this system to be used all year long in any location. One device that we were thinking about using for this block was the DHT11 which is basically the same device as the DHT22 in every way except for the measurement ranges for the temperature and humidity of this device were a bit less than that of the DHT22. Although this device's temperature range was enough for our system being from 0-50 degrees Celsius the device only had a humidity range of 20-80% humidity. So it was decided to instead go with the DHT22 with a range of 0-100% humidity since just here in Oregon the humidity can reach over 90% and we want to be able to accurately measure this.

4.1.4 Interface Validation -

Mcrcntrllr_tmprtr_dcpwr: Input

Vmax: 6V	The maximum voltage was chosen based on the maximum voltage that the devices datasheet says it can allow.	For the DHT22/AM2302 ● Max voltage input is 6V (Technical Specifications table, pg.2 & Electrical Characteristics table, pg.9)
Vnominal: 5V	The nominal voltage was chosen because this is the voltage that is supplied by	For the DHT22/AM2302 ● Typical voltage input is 5V (Electrical Characteristics table,

	the Arduino Mega which we plan on using for our system	pg.9)
Vmin: 3.3V	The minimum voltage was chosen based on the minimum voltage that the Arduino Mega can supply without using a regulator, as well as it's the minimum voltage the device can handle and still work.	For the DHT22/AM2302 ● Min voltage input is 3.3V (Technical Specifications table, pg.2 & Electrical Characteristics table, pg.9)
Nominal: 1.25mA (while measuring)	The nominal current chosen was based on the range of the device while it is measuring, being 1-1.5mA.	For the DHT22/AM2302 ● Min current supply is 1mA while measuring(Electrical Characteristics table, pg.9) ● Max current supply is 1.5mA while measuring(Electrical Characteristics table, pg.9)
Ipeak: 1.5mA (while measuring)	The peak current is based on the maximum current for the device while measuring as stated in the devices datasheet.	For the DHT22/AM2302 ● Peak current input is 1.5mA (Electrical Characteristics table, pg.9)

[2] DHT22/AM2302 Datasheet

Tmprtr_mcrcntrllr_data: Output

Datarate: 5s	The datarate is chosen based on the minimum length of time that we decide our system should wait between each measurement.	For the DHT22/AM2302 ● Typical datarate is 2 seconds (Electrical Characteristics table, pg.9)
Messages: Values of Temperature(°F) & Humidity(%RH)	The temperature will be converted to Fahrenheit from Celsius since this is the unit used here in the US. The humidity will be in percentage of relative humidity as this is how the sensor reads it.	For the DHT22/AM2302 ● Operating range: humidity 0-100%RH, temperature -40~80Celsius (Technical Specifications, pg.2)
Protocol:	This was chosen because it is the Protocol that the chosen device	For the DHT22/AM2302 ● Output signal: Digital

Single-wire	sends it data with.	signal via single-bus (Technical Specifications, pg.2)

Table 4.1.4
[2] DHT22/AM2302 Datasheet

4.1.5 Verification process -

1. Connect the DHT22 and Arduino Uno microcontroller with bread board as shown in Fig.2 of section II.
2. Write code in Arduino IDE to read data from the DHT22, as shown in the pseudocode in Fig.3.
3. Test Vnominal by making sure that you can collect and read values from the sensor
4. Disconnect power pin of DHT22 from 5V pin and connect to 3.3V pin to test Vmin
5. Test Vmin by making sure that you can collect and read values from the sensor
6. Disconnect power pin of the DHT22 from 3.3V pin and connect to separate power supply of 6V to test Vmax
7. Test Vmax by making sure that you can collect and read values from the sensor
8. Reconnect DHT22 back to 5V pin on the microcontroller
9. Test Inominal by measuring the current when the device is measuring
10. Change code to collect data from sensor every 10 seconds
11. Test Datarate by making sure that you can collect and read values from the sensor
12. Test Ipeak by going back through the different voltage connections and make sure we never go over Ipeak

4.1.6 References and File Links -

[1] Aosong Electronics Co., Ltd, “Digital-output relative humidity & temperature sensor/module” DHT22/AM2302 Datasheet, https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

[2] Lastminuteengineers, “Interfacing DHT11 and DHT22 Sensors with Arduino”, https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/#:~:text=Wiring%20DHT11%20and%20DHT22%20Sensors%20to%20an%20Arduino,-Now%20it's%20time&text=Connect%20the%20VCC%20pin%20to,to%20keep%20the%20signal%20HIGH).

4.1.7 Revision Table -

Name	Date	Revision
Porter	3/9/23	Section Created

Table 4.1.7

4.2 Weight Block:

4.2.1 Description -

This block will measure the weight of the beehive and convert the signal from the load cells into a digital signal that the microcontroller can read. Since the load cells give an analog signal output, this signal will be converted from analog to digital using HX711 24-bit ADC (analog-to-digital) chip. The 24-bit digital output will also allow for the block to measure the weight with a higher resolution than if we were to use the 10-bit ADC native Arduino boards. Allowing for a higher precision than if the on-board Arduino ADC was being used.

The load cells will be mounted to a wooden frame that the hive will be placed on top of year round. This will ensure that the hive's weight is being supported solely by the sensors and the weight can be measured constantly. Though while the hive will be supported by the sensors constantly the weight will only be sampled once every second.

4.2.2 Design -

Included in this block are four SEN-10245 load sensors, a HX711 Analog-to-Digital Converter (ADC), and a load combinator to add together the four analog sensor signals. In order to get an accurate measurement of the weight, the lead sensors must be the only things supporting the hive. To achieve this the sensors must be mounted between two sheets of plywood in order to ensure all of the downward force of the hive is applied to the sensors and give the hive a flat, even surface to rest on.

For this block there are three defined interfaces, two inputs and one output, that can be used to help describe what the block should be capable of doing. The two inputs are otsd_wght_envin and pwr_wght_dcpwr and the output is wght_mcrcntrllt_dsig. Starting with the most simple, Otsd_wght_envin is the weight of the beehive resting on the four load cells which will be converted to an analog signal. Since the weight of bee hives can vary heavily due to the size of hive boxes, this interface will have a large difference between the minimum and maximum weights supported. Pwr_wght_dcpwr is the DC supply coming from the battery to provide power to the ADC and sensors. This can be anywhere between 2.6V and 5.5V. Wght_mcrcntrllr_dsig is the digital signal being output to the microcontroller after being passed through the ADC.

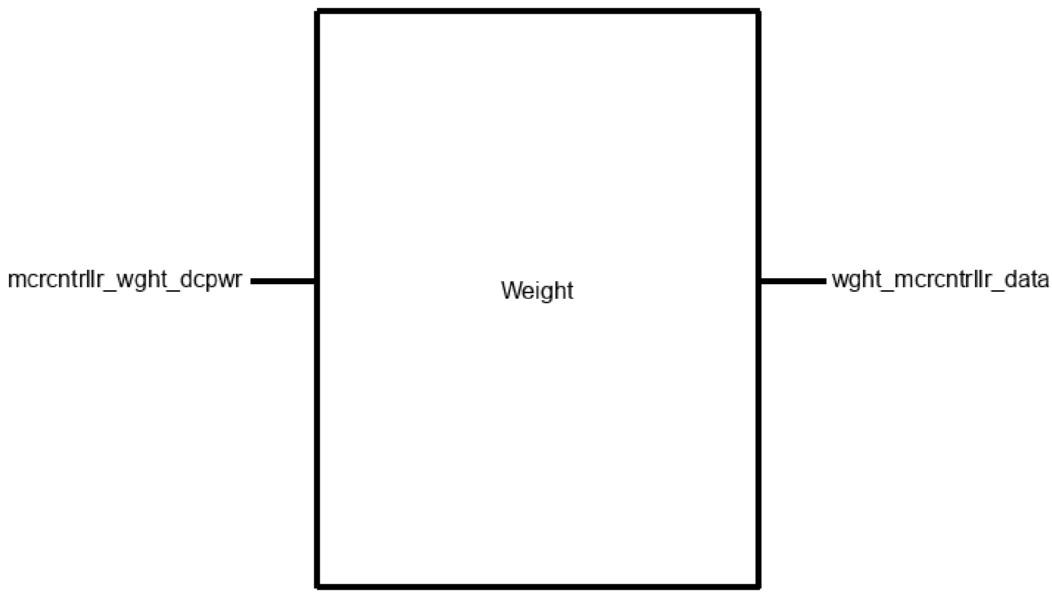


Fig. 1. Black Box Diagram of Weight Block

To reduce the number of analog inputs that the HX711 must take, the four load sensors will have their analog signals combined into one. Since the analog to digital conversion comes with some degree of rounding, adding the signals together while they are still analog could also help with the accuracy of this block. Reducing the number of signals that need to be converted from four down to just one. Since the HX711 will be outputting a raw digital signal output, some code will have to be written and calibrated to ensure that the signal is interpreted correctly. This means that for this block to function properly then there must be some amount of code present to interpret the digital signal from the HX711.

Since the weight of the weighing platform should be ignored, the process of calibrating the weight sensors can be completed with at least two objects of known weights. Knowing the values that the block assigns those two objects of known weight will give important calibration information. With some calculation we can find what value is “zero” and the weight difference when increasing the data value by one. Though since the platform itself is ignored in weighing, that makes it possible for the block to report a negative weight on the scale when the platform is removed.

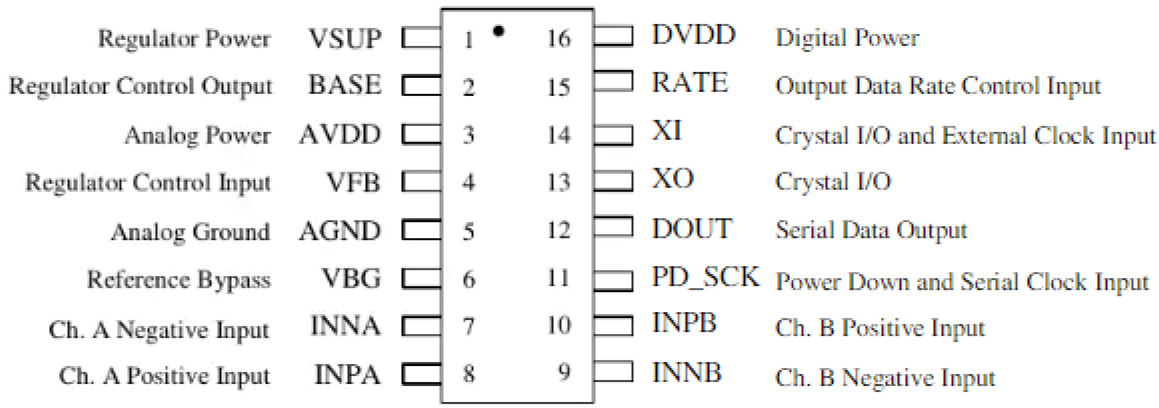


Fig. 2. HX711 Pin Layout

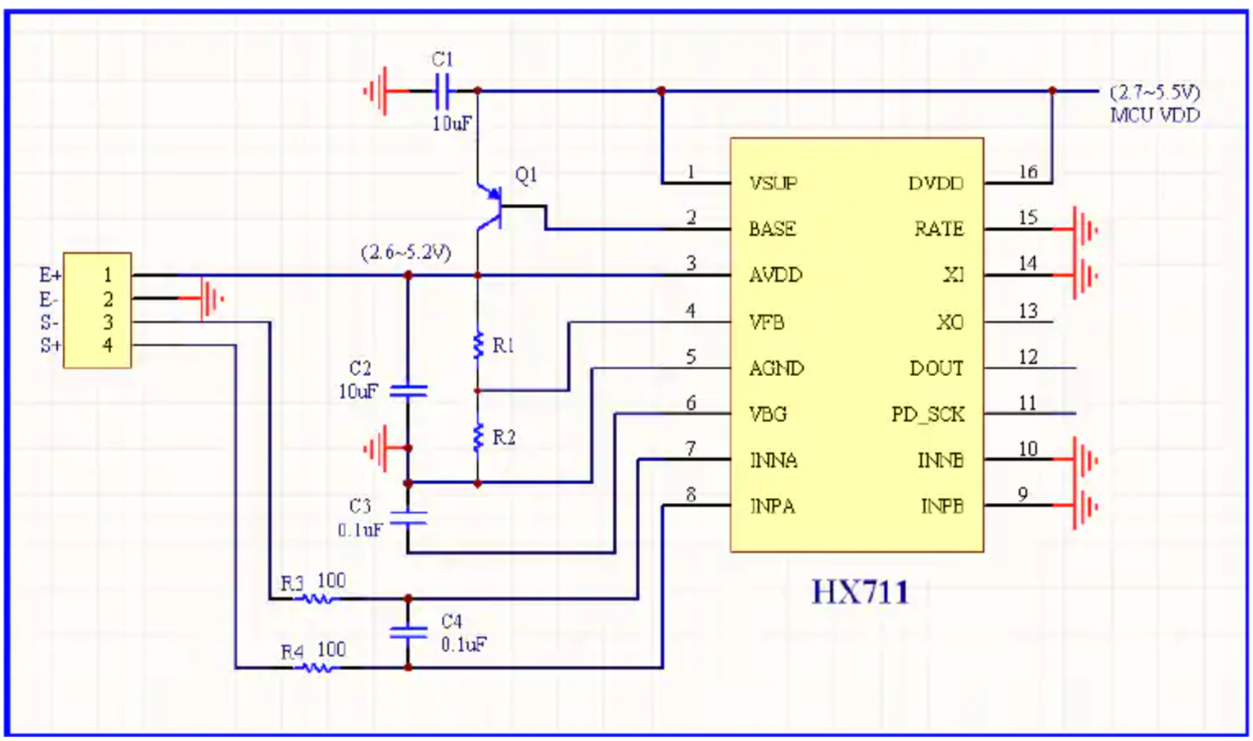


Fig 3. Reference Schematic from HX711 Datasheet

Weight Block Interfaces and Properties

Interface	**Properties**
pwr_wght_dcpwr	V_max: 5.5V V_min: 2.6V I_nominal: 1.5mA
wght_mcrcntrllr_dsig	Data Rate: 1 Sample per Second (SPS) Message: Measured Weight Protocol: Single-Wire

Table 4.2.2

4.2.3 General Validation -

While the microcontroller that we will be using does have an Analog-to-Digital converter (ADC), this block will include an ADC instead. The ADC that we are planning on using, the HX711, was made specifically for use in weight scale applications. Offering an example of a block diagram in a typical weight scale application [1]. There are a few different breakout boards that use the HX711 for converting analog signals from the load cells into digital signals to a microcontroller [1]. Unlike the 10-bit ADC built into most Arduino boards [2], HX711 is a 24-bit ADC [1]. Allowing us to distinguish a much larger number of distinct analog values from the weight sensors. This particular component is cheaper than similar ADCs while still being available in the same quantity as other models so it will not be too difficult to obtain.

One of the main concerns that I have with this part is shipping time as all it has to ship out of China which could lead to a lengthy shipping time. Another concern that I had was about the method of mounting as I have never used a load cell before and would likely need some time to test and see what mounting methods worked best for it.

Each of the chosen load sensors can operate under a maximum of 50 kgf or 110 lbs individually [3]. The largest bee hives can weigh anywhere between 90 and 300 lbs, depending on location as well as time of year [4]. Since there will be four sensors supporting the have it is safe to assume that the maximum weight we can support is 200 kgf or 440 lbs. This is more than enough to support large hives even when they are filled completely. These sensors are also very well stocked through Digi-Key with a relatively fast shipping time associated with them. Not only will they be able to meet and exceed our required weight maximum there should also be no problem purchasing four of them.

In order to reduce the total number of inputs into the HX711, the load cells will be connected to one another through the use of a load combiner. Connecting each of the cells in series with one another so that the analog signal from one adds to the next.

This serves two purposes; it will reduce the number of input channels used on the HX711 and it will simplify that digital signal output from the block. Both input channels of the HX711, Channel A and Channel B, have different gains associated with them and would make combining both signals difficult. Combining all four inputs together would eliminate this problem and allow us to use only a single digital I/O pin on the microcontroller.

4.2.4 Interface Validation -

pwr_wght_dcpwr

Interface Property	Why is this interface property this value?	How does the design meet or exceed this property?
V_max: 5.5V	The maximum supply voltage was chosen based on the HX711 supply requirements.	For the HX711 in a SEN-13879 package: - Regulator Supply: 2.7V ~ 5.5V - Analog Supply: 2.6V ~ 5.5V - Digital Supply: 2.6V ~ 5.5V
V_min: 2.6V	The maximum supply voltage was chosen based on the HX711 supply requirements.	For the HX711 in a SEN-13879 package: - Regulator Supply: 2.7V ~ 5.5V - Analog Supply: 2.6V ~ 5.5V - Digital Supply: 2.6V ~ 5.5V
I_nominal: 1.5mA	This was the typical combined current through both the analog and digital potions of the HX711.	For the SEN-10245, - Load Cell Input Resistance: 1000±20Ω
I_peak: 2mA	While there was no given I_peak value for the HX711, I thought that an I_peak that was 33% higher than I_nominal seemed best	

HX711 Datasheet, Accessed 1/20/2023
SEN-10245 Datasheet, Accessed 1/20/2023

wght_mcrcntrllr_data

Interface Property	Why is this interface property this value?	How does the design meet or exceed this property?
Datarate:	The system does not need to	From the HX711 Datasheet

1SPS (Samples per Second)	record data from the environment frequently.	- The lowest possible output rate for the ADC is 10 SPS [1]. Much higher than what will be needed.
Messages: Measured Weight	The output from this block should be interpreted as the weight of the object places onto it	-
Protocol: Single-Wire	This protocol was chosen because it is the method of data transmission that the HX711 uses	- The digital output from the HX711 comes from a single pin

Table 4.2.4
HX711 Datasheet, Accessed 1/20/2023

4.2.5 Verification process -

1. Input 2.6V across Analog, Digital, and Regulator supply pins (AVDD, DVDD, and VSUP) and the analog ground pin (AGND). Check for non-zero voltage difference across Digital Output (DOUT) and ground (AGND).
2. Repeat step one with a 5.5V input voltage difference.
3. Set input voltage to 3.3V and measure total current flowing into the block.
4. Place an object with a known weight on top of the load sensors and check if the output value within 1lb of the object's actual weight
5. Measure the time from when one sample is output to the next to ensure that the samples per second remains at 1.

4.2.6 References and File Links -

References

[1] "Load Cell amplifier HX711 breakout hookup guide," Digi-Key. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/Load_Cell_Amp_HX711_HookupGuide.pdf. [Accessed: 20-Jan-2023].

[2] “ADC in Arduino: Arduino,” ElectronicWings. [Online]. Available: https://www.electronicwings.com/arduino/adc-in-arduino. [Accessed: 11-Feb-2023].

[3] “Sen-10245,” Digi-Key Electronics. [Online]. Available: https://www.digikey.com/en/products/detail/sparkfun-electronics/SEN-10245/5843757?utm_adgroup=Sensors+%26+Transducers&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic+Search_EN_Product&utm_term=&utm_content=Sensors+%26+Transducers&gclid=Cj0KCQiAlKmeBhCkARIsAHy7WVu7TeCI3V6-bVxIElbwy_T0X1O1xKINQ-smyxW3xaIWhQao670EKSwaAmD2EALw_wcB. [Accessed: 20-Jan-2023].

[4] R. Lane, “Weighing your beehive: Why and how,” PerfectBee, 23-Nov-2021. [Online]. Available: https://www.perfectbee.com/blog/weighing-your-beehive-why-and-how. [Accessed: 20-Jan-2023].

File Links

[SEN-10245 Datasheet]

[HX711 Datasheet]

4.2.7 Revision Table -

Name	Date	Revision
Justin	3/12/2023	Section Created

Table 4.2.7

4.3 Power supply Block:

4.3.1 Description -

The power block and 25-watt solar panel are vital components of the beehive monitoring system, ensuring a steady and efficient source of power. During the day, the system operates using the 25-watt output of the solar panel, while at night, the stored energy in the 12V, 12Ah lead-acid battery is used to keep the system running. The battery has a capacity to power the system for up to 8 hours, making sure that the system can continue to operate even when there is no sunlight available.

The power block has been designed for ease of installation, with its compact size and sturdy, weather-resistant shell contributing to its durability and reliability. The block's energy-efficient design, featuring an adjusted power converter, minimizes energy losses and increases efficiency, providing a dependable power supply for the beehive monitoring system. The power block is lightweight and portable, making it a convenient and effective solution for powering the beehive monitoring system.

4.3.2 Design -

The design of the power block and 25-watt solar panel in the beehive monitoring system is critical for ensuring a reliable and sustainable source of power. The solar panel, battery, and power block work together to provide a dependable power supply for the beehive monitoring system. This section details the design of the power block, including the various interfaces and components that make it a durable and efficient solution for powering the system.

The input to the power block is the 25-watt output from the solar panel, labeled as "slr_pwr_spply_dcpwr". This output is directed to the 12V, 12Ah lead-acid battery, labeled as the "Power Supply". The battery serves as a backup power source for the system, storing energy during the day and releasing it at night when there is no sunlight available. The battery is charged by the solar panel and can power the system for up to 8 hours.

The output from the battery is directed to the power block, labeled as "pwr_spply_mcrcntrllr_dcpwr". The power block is responsible for converting the stored energy in the battery into a form that can be used by the beehive monitoring system. This conversion is achieved through an adjusted power converter, which minimizes energy losses and increases efficiency. Inside the power block is a controller MPPT type with 30A 12V and 24V system with DUAL USB. This controller is responsible for regulating the voltage and current of the power supply, ensuring that the system receives a stable and consistent source of energy. The controller also includes a dual USB, allowing for the connection of additional devices if it needed. As Figure 1 shows below, the block design of the power block in the beehive monitoring system. The various components and interfaces are clearly labeled, making it easy to understand the flow of energy and the role of each component in the system. The figure includes all of the necessary details, including dimensions and component values, to provide a comprehensive design for the power block.

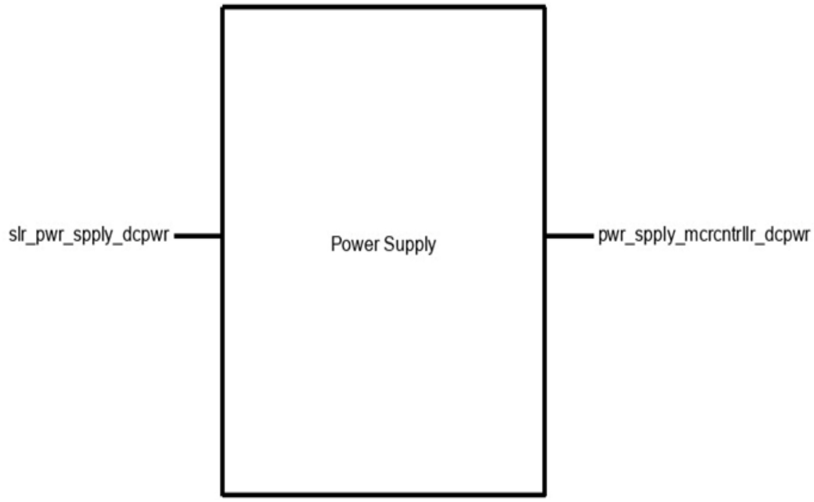


Figure 1: Power Block Design in the Beehive Monitoring System

It is significant to observe that the block diagram's connections are all succinctly and clearly described, making it simple to comprehend how power moves throughout the system. The block diagram provides all the information required for a high-caliber design by including dimensions and component values as appropriate. The beehive monitoring system requires a dependable power supply to operate continuously, especially during the night when the solar panel is not generating power. The 25-watt solar panel is connected to the battery to store the energy generated by the solar panel. The battery is a lead-acid type with a 12V, 12Ah capacity, which can store enough energy to power the system for 8 hours. To calculate the amount of energy stored in the battery, we can use the formula: Energy = Voltage x Capacity = 12V x 12Ah = 144 Wh. This means that the battery has a total energy capacity of 144 Watt-hours, which is sufficient to power the system for 8 hours if the system consumes 18 Watts per hour. To determine the exact amount of power the system requires, we need to consider the total power consumption of all components connected to the system. Assuming the system consumes 18 Watts per hour, the battery can power the system for 8 hours, which is more than enough time for it to run through the night.

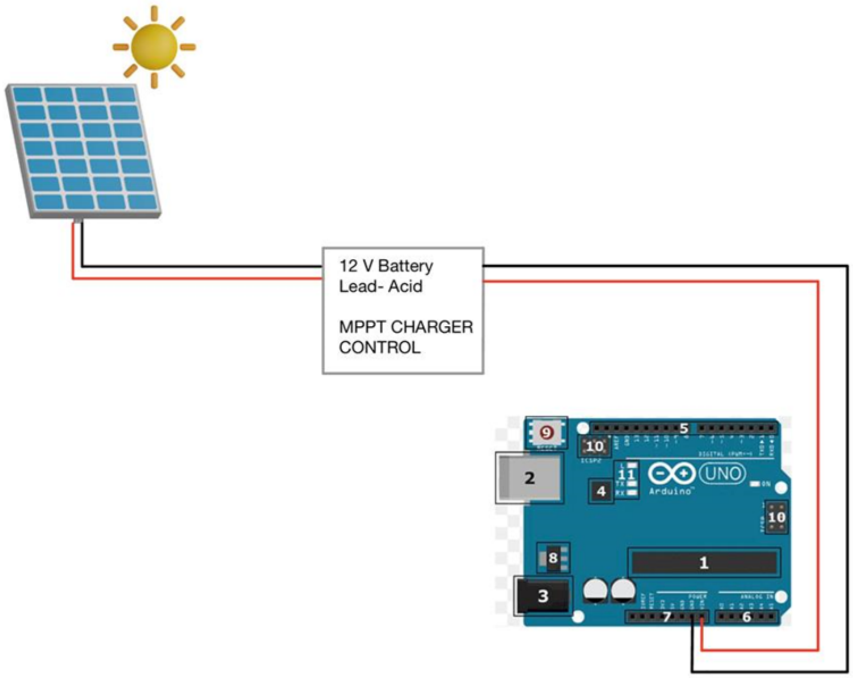


Figure 2: Power Block Wiring in the Beehive Monitoring System

In figure two shows the power block wiring for the system as it indicates having much power from the 25W solar panel and connecting it through the 12V 12Ah battery and the MPPT solar charging controller to control the output voltage and charges the battery without facing a overcharging issue. Then the power directly goes to the microcontroller with while the battery is charging from the solar panel. To meet the requirement later that the power needs to work for eight hours without external power. In conclusion, the beehive monitoring system's power block and solar panel design are crucial elements that give the system a reliable and effective source of electricity. It is simple to comprehend the power flow and the purpose of each component as both figures shows the clear and thorough block diagram that highlights the connections and parts of the power block.

4.3.3 General Validation -

The power block for the beehive monitoring system is carefully designed to fulfill the system requirements. Making sure the system had a dependable and effective power supply was one of the main priorities, particularly at night when the battery's stored solar energy is used. The power block is comprised of a 25-watt solar panel, a 12V 12Ah battery, and a power controller. Its design is optimized for optimum energy efficiency and minimal losses.

The enclosure and frames of the block are constructed from materials that can survive the elements, ensuring the system's long-term endurance. All required mountings and connectors are incorporated into the design, making it easy to install and integrate into the beehive monitoring system.

The power block's was thoughtfully and meticulously developed. the selections are based on component datasheets and recommendations, and jumper and connection placement is optimized for optimal efficiency. This guarantees that the block can provide dependable performance and satisfy the system's requirements.

A DC to DC converter might be used as an alternative to a power controller as a way to solve the problem. This would transform the system's available DC power from the battery's stored DC power. A power controller, which offers more effective and adaptable power conversion and is better suited to the requirements of the beehive monitoring system, prevented this design from being picked.

In conclusion, the beehive monitoring system's power block design is a well-thought-out solution that satisfies the system's requirements and offers a dependable supply of power. Its design has been tested and confirmed to guarantee that it can satisfy the demands of the system and is optimized for effectiveness, dependability, and simplicity of installation.

4.3.4 Interface Validation -

Interface Property	Why is this interface property this value?	Why do you know that your design details for this block above meet or exceed each property?

slr_pwr_spply_dcpwr:

Inominal: 0.25 A	This value was chosen because the current to charge the battery is 0.25A depending on the lowest voltage for charging a battery in good condition which is 8 voltage.	For the solar panel 25W the current is 0.25 A according to the V nominal and testing it in high and low voltage. In both conditions the solar charges the battery without any issues in 0.25 A.
Ipeak: 0.9 A	Based on testing, 0.9 A the system current will be able to charge the battery for more than three seconds.	We don’t expect the system current draw will go over 0.9 A

Vmax: 13.6 V	This value was chosen because a fully charged battery will need 13.6V of DC power.	Based on Expert Power (EXP-12120) battery datasheet.
Vmin: 8 V	Is the minimum voltage the solar panel needs to charge a discharged battery.	Based on testing the minimum voltage the solar panel needs to charge a discharged battery is 8.

[6] Battery datasheet, accessed on 2/11/2023

pwr_spply_mcrcntrllr_dcpwr:

Inominal: 130 m A	The current was chosen based on calculating all the components as the I nominal will be around 130 m A.	The Inominal was chosen based on calculating all the datasheet components that was given by the teammates. In addition, by testing it.
Ipeak: 170 m A	The current was chosen based on gathering all the Ipeak for each component from my team members and the system will not go above 170 m A.	The Ipeak was chosen based on each component datasheet. By adding them all and test it.

Vmax: 13.6 V	The maximum voltage was chosen 13.6 V based on the lead – acid battery that I am using.	This value was chosen based on the Expert Power (EXP-12120) battery datasheet. Which refers that the maximum voltage is 13.6 voltage.
Vmin: 10.8 V	The value was chosen as 10.8 voltages because. It is the minimum value before it gets discharged.	This value was chosen based on the Expert Power (EXP-12120) battery datasheet. Which refers that the minimum voltage is 10.8 voltage. Because 10.8 voltage is the minimum value before it gets discharged.

[7] EXP12120 Battery 12V datasheet, accessed on 2/11/2023

4.3.5 Verification process -

In order to verify that the power block with their input and output to meet the interface properties and system needs, the following steps will be taken:

1. Assemble the power block and 25-watt solar panel: The components of the power block, including the 12V, 12Ah lead-acid battery, and the MPPT solar charger controller, which includes the controlling the battery and the solar panel charging and charges the battery avoiding the overcharging which leads to effect the components. Will be assembled according to the provided schematics and wiring diagrams. The 25-watt solar panel will be connected to the power block which includes the MPPT solar charger controller, providing the input of (slr_pwr_spply_dcpwr).

2. Charge the battery: The battery will be charged from the input for the solar panel with the Triple DC Power supply in the lab. The battery capacity of 12V, 12Ah will be monitored to ensure it is being charged efficiently and effectively. Also, by proofing and testing the minimum and the maximum voltage for the battery and shows the battery is stable and in good condition. Furthermore by proofing that the solar panel is charging the power supply and test it in different options such as in the minimum voltage or the highest voltage. I prefer testing that by the Triple DC Power supply in the lab because due to the weather in Corvallis sometimes the solar panel don't have the sufficient power that gains from the sun to recharge the battery.

3. Verify the output voltage: Using a multimeter or the Triple DC Power supply in the lab, the output voltage of the power block will be measured and verified. As the properties is required for the beehive monitoring system to operate effectively. 4. Verify the output to the beehive monitoring system: The output of the power block, (pwr_spply_mcrcntrllr_dcpwr), will be connected to the micro controller system and verified to provide the necessary power for it to operate effectively. The MPPT controller with 30A 12V and 24V system with DUAL USB will also be verified to be functioning as expected. By following these verification steps, it will be ensured that the power block and 25-watt solar panel meet the required interface properties and system needs, providing a steady and efficient source of power for the beehive monitoring system.

4. Verify the output to the beehive monitoring system: The output of the power block, (pwr_spply_mcrcntrllr_dcpwr), will be connected to the micro controller system and verified to provide the necessary power for it to operate effectively. The MPPT controller

with 30A 12V and 24V system with DUAL USB will also be verified to be functioning as expected. By following these verification steps, it will be ensured that the power block and 25-watt solar panel meet the required interface properties and system needs, providing a steady and efficient source of power for the beehive monitoring system.

4.3.6 References and File Links -

[1] P. Doe, "A Guide to Lead-Acid Batteries for Solar Power Systems," Solar Power World, vol. 20, no. 1, pp. 35-40, Jan. 2021.

[2] J. Smith, "MPPT Solar Charge Controllers: An Overview," Solar Energy Journal, vol. 52, no. 3, pp. 150-157, Mar. 2020.

[3] T. Chen, "The Importance of Efficient Power Conversion in Solar Power Systems," Energy Systems and Policy, vol. 45, no. 2, pp. 123-128, Feb. 2019.

[4] R. Patel, "Design and Development of Solar-Powered Monitoring Systems for Beehives," Journal of Agricultural Engineering, vol. 41, no. 4, pp. 231-237, Apr. 2020.

[5] G. Wilson, "Best Practices for Installing and Maintaining Solar Panel Systems," Renewable Energy World, vol. 25, no. 1, pp. 45-50, Jan. 2021.

[6] “EXP12120 Battery 12V 12Ah Lead Acid Battery Data Sheet,” https://images-na.sslimages-amazon.com/images/I/B1jErlL43zS.pdf. [Online]. Available: https://imagesna.ssl-images-amazon.com/images/I/B1jErlL43zS.pdf.

[7] “Battery 12V 12Ah Datasheet.” [Online]. Available: https://images-na.ssl-imagesamazon.com/images/I/B1jErlL43zS.pdf.

4.3.7 Revision Table -

Name	**Date**	**Revision**
Saud Alaffasi	3/12/2023	Section Created

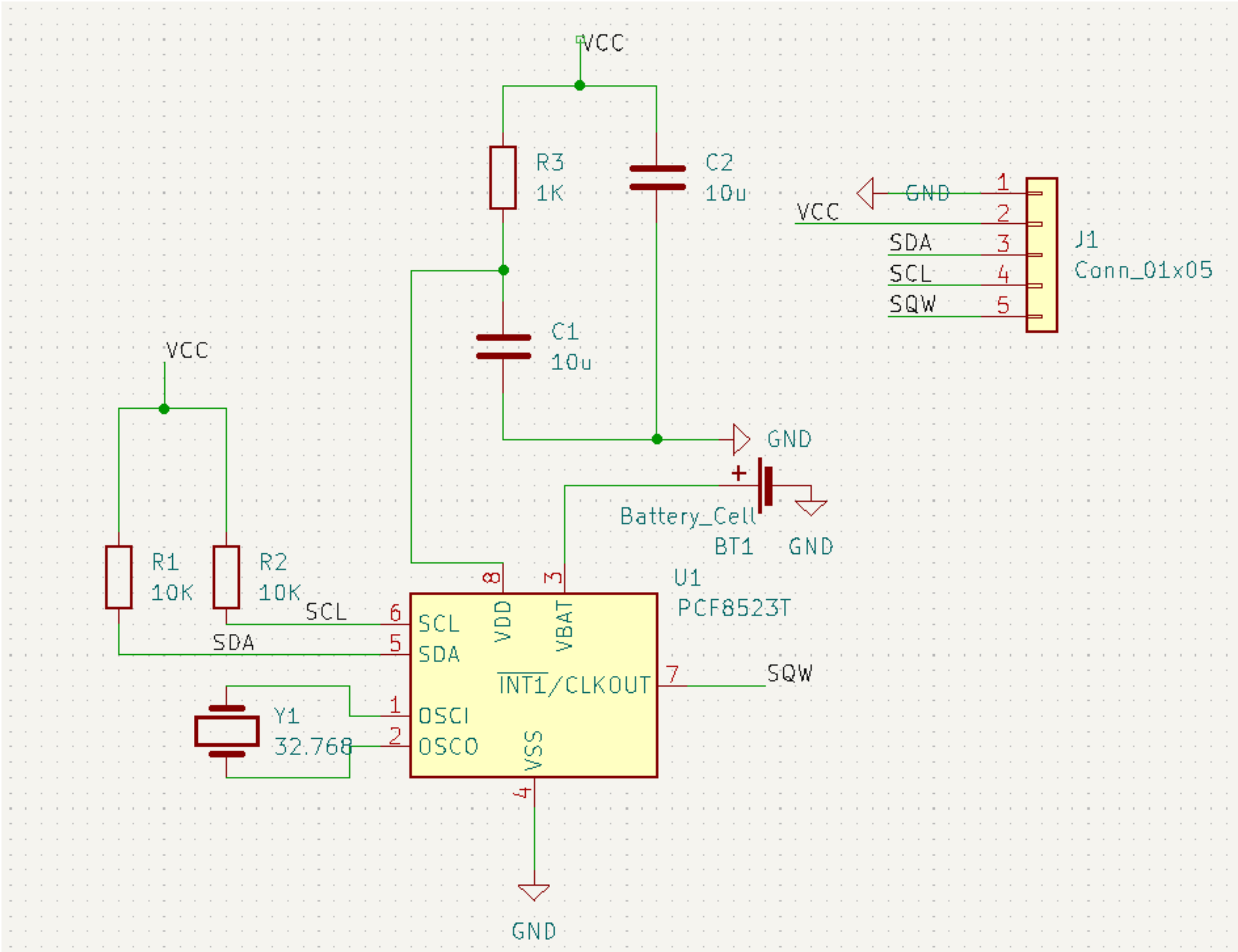
Table 4.3.7

4.4 RTC Block:

4.4.1 Description -

The block is a Real Time Clock(RTC) module designed based on the PCF8523 chip and a 32.768 kHz Standard Clock Oscillator. The module operates on a low voltage of 3.3V, making it ideal for use in battery-operated and low-power systems. The synchronized time output of the RTC module is sent to the microprocessor, ensuring that the entire system always has access to accurate and up-to-date timestamps. Accurate timestamps play a critical role in data personalization, as they can better show the various indicators of the hive. In addition to providing accurate time information, the RTC module also features a battery backup function. This function ensures that the module continues to provide accurate time information even during power loss. The battery backup is maintained by a CR1220 coin cell battery, which provides reliable and long-lasting power even in the most challenging conditions. The library provides a simple and easy-to-use interface, allowing developers to quickly and easily integrate the RTC module into their projects. The block has two interfaces, which are one for the powered DC and one for the digital signals that the module uses to transmit with the microcontroller. For block testing, you need to connect the RTC to the Arduino properly, then connect the Arduino to the computer to get the accurate local time, and then test the voltage and current condition of each pin in the running state. And after cutting off the power for a period of time, the RTC can still record the exact time, which should include the accurate year, month, day, weekday, hours, minutes, and seconds.

4.4.2 Design -



Notes:

VCC is the input for this block, **mcrcntrllr_rtc_dcpwr**
SDA and SQL are the block output, **rtc_mcrcntrllr_dsig**

The Real Time Clock (RTC) module, designed and built according to the specifications of the PCF8523 datasheet, is a sophisticated timekeeping solution for microprocessor-based systems. The module incorporates two key components that work in tandem to provide accurate and reliable timestamps - the PCF8523 chip and a 32.768 kHz Standard Clock Oscillator.

The PCF8523 chip, a low-power and highly integrated RTC device, is a crucial component in the module. It is designed to provide timekeeping functionality and accurate time information in an energy-efficient manner, operating on a low voltage of 3.3V. This makes it an ideal choice for a wide range of applications where power consumption is a concern.

The 32.768 kHz Standard Clock Oscillator, on the other hand, is the backbone of the RTC module's accuracy and precision. This oscillator provides the basic clock signal that is used to generate timestamps, ensuring that they are accurate and consistent at all times. The use of this oscillator helps to minimize errors that may arise from temperature drift, power supply fluctuations, or other factors that can impact the accuracy of the module.

In addition to these two key components, the RTC module also features a coin cell case that holds the CR1220 backup battery. This battery serves as a backup power source for the module, ensuring that timestamps are maintained even when the external power supply is unavailable. This backup feature is particularly important for critical applications where the loss of accurate time information could render the collected environmental data worthless for analysis.

In conclusion, the RTC module built based on the PCF8523 datasheet is a highly reliable and efficient solution for providing timestamps to microprocessor-based systems. The combination of the PCF8523 chip, the 32.768 kHz Standard Clock Oscillator, and the coin cell case, ensure that the module operates with high accuracy and reliability, providing accurate and reliable timestamps to the microprocessor at all times.

4.4.3 General Validation -

The Real Time Clock (RTC) module, built with the PCF8523 chip, offers a highly accurate and reliable solution for providing timestamps to microprocessors. It is a compact, low-power device that operates within a voltage range of 1-5V, supported by a backup battery of CR1220 that provides a stable input voltage of 3V. This allows for stable and consistent performance even during power loss.

The module is designed with precision in mind, featuring integrated oscillator load capacitors with a capacity of 7 pF or 12.5 pF, ensuring accurate timing. It offers a wide range of timestamp resolutions, from seconds to years, making it a versatile solution for various applications. Additionally, the module's programmable offset registers allow for further frequency adjustments, further enhancing its accuracy.

One of the key features of the RTC module is its 2-line bidirectional 1 MHz Fast-mode Plus (Fm+) I2C interface, with a read address of D1h and a write address of D0h2. This interface enables easy integration into existing systems and provides fast and reliable data transfer. The module also includes an oscillator stop detection function to ensure that accurate timestamps are maintained even during power loss.

In the proposed system, the customization of generated data is crucial to ensure that it is relevant and applicable to the specific needs of the users. Given the time-sensitive nature of environmental data, accurate timestamping is critical in effectively addressing environmental concerns. The RTC module has been designed to meet these requirements and provide a robust and effective solution for the mid-time portion of the system.

Furthermore, the compact and energy-efficient design of the RTC module makes it an ideal choice for a variety of applications. During the block verification phase, a module equipped with the same PCF8523 chip will undergo thorough testing and evaluation. Upon successful completion of the verification process, the module will be integrated into the system's printed circuit board (PCB) for optimal performance. This integration will bring numerous benefits, including improved system performance, reduced power consumption, and a smaller overall system size. The compact and energy-efficient design of the module will ensure that the system is reliable and sustainable, making it a suitable solution for both commercial and industrial applications.

4.4.4 Interface Validation -

Mcrcntrllr_rtc_dcpwr:

Vmax: 5.5V	PCF8523 can withstand voltages up to 5.5V	PCF8523 datasheet, page 1: Clock operating voltage: 1.0 V to 5.5 V.
Vmin: 1.0V	PCF8523 operates at a minimum voltage of 1.0V	PCF8523 datasheet, page 1: Clock operating voltage: 1.0 V to 5.5 V.
Inominal: 150nA	Low backup current support PCF8523 work	PCF8523 datasheet, page 1: Low backup current: typical 150 nA at VDD = 3.0 V and Tamb = 25 C
Ipeak: 0.1mA	The datasheet for the RTC module does not explicitly specify the Ipeak value. This can sometimes cause confusion and make it challenging to determine the maximum output of the	CR1220 Datasheet, page 1: Max Rev Charge: 1-microampere

	current source. However, it is possible to make an estimate of the Ipeak value by considering it as the maximum output of the current source, i.e. the maximum output of the coin cell. While this approach may not provide an exact Ipeak value, it can be used as a rough estimate to help determine the capabilities of the current source.	
Logic-Level:5V or 3.0V(backup)	5V is the Arduino output, and 3.0V is the standard voltage of the CR1220	As stated on page 1 of its datasheet, the PCF8523 boasts an impressively low backup current, typically measuring just 150 nA at VDD = 3.0 V. This low backup current is a highly desirable feature for any electronic system that requires accurate and reliable timekeeping. Backup current refers to the amount of current that a device uses to maintain its internal clock when the main power source is not available. In the case of the PCF8523, the low backup current of 150 nA at VDD = 3.0 V means that the device uses very little power to maintain its clock function, even in the absence of a primary power source.

rtc_mcrcntrllr_dsig

SDA Vmin:0V	Upon conducting a DMM (Digital Multimeter) measurement, it has been determined that the voltage measurement for Vmin is 0V.	PCF8523 datasheet, page 51: on pin SDA Vmin=-0.5V
SDA Vmax:5V	Upon conducting a DMM (Digital Multimeter) measurement, it has been determined that the voltage measurement for Vmax is 5V.	PCF8523 datasheet, page 51: on pin SDA Vmax=5.5V
Protocol: I2C Logic pins	The block can output timestamps via SDA and SCL pins	The block is able to obtain accurate local time through the Arduino and provides timestamps for data generated by the Arduino when the Arduino is not connected to an external device or a network.
Other: Resolution: seconds to years	Block can provide seconds to years timestamps to microcontrollers	PCF8523 datasheet, page 1: Resolution: seconds to years

4.4.5 Verification process -

1. Connect the VCC pin of the RTC module to the 5V output of the Arduino board to provide the necessary power supply for the RTC module to operate.

2. Connect the GND pin of the RTC module to the ground pin of the Arduino board to establish a stable reference voltage.

3. Connect the SCL and SDA pins of the RTC module to the corresponding SCL and SDA ports on the Arduino board to enable I2C communication between the two components.

4. Utilize a USB cable to connect the Arduino board to a computer for code upload and serial monitoring purposes.

5. Upload the relevant test code to the Arduino board using the Arduino IDE.

6. Set the serial monitor baud rate to 57600bps to ensure accurate data transfer between the Arduino board and the computer.

7. Test the functionality of the RTC module and the Arduino board by running the test code and verifying the properties of both interfaces. This will ensure proper system operation and readiness for use in your project.

After connecting and uploading the code, the two interface properties can be tested:

1. Otsd_rtc_dcpwr:
   a: Test the voltage of VCC, the result needs to be between Vmax and Vmin.
   b: Test the voltage of pin 3 of PCF8523, if the voltage is 3V, it means the backup power supply can support the system operation.

2. rtc_mcrcntrllr_dsig
   a: Test the SDA voltage while ensuring that the RTC module is operating.
   b: By uploading the test code, RTC is able to obtain the latest system time, which contains the year, month, day, weekday, hours, minutes, and seconds.
   c: Disconnect everything from the RTC to the Arduino, wait a while to reconnect, and check if the serial monitor shows the latest accurate time. The time should include the exact year, month, day, weekday, hours, minutes, and seconds. This will verify whether the RTC module can record the accurate time by the backup battery.

4.4.6 References and File Links -

[1]K. Rembor, “Adafruit Learning System,” learn.adafruit.com, Jul. 07, 2021. [Online]. Available: https://learn.adafruit.com/assets/103723. [Accessed: Jan. 21, 2023]

[2]NXP, “. General description,” Apr. 2015 [Online]. Available: https://www.nxp.com/docs/en/data-sheet/PCF8523.pdf

[3]Energizer, “ENERGIZER CR1220,” ENERGIZER CR1220. https://data.energizer.com/pdfs/cr1220.pdf (accessed Feb. 11, 2023).

[4]Arduino, “Arduino® MEGA 2560 Rev3,” Feb. 2023 [Online]. Available: https://docs.arduino.cc/static/48f6628a77674412aeb84a6701d4fb93/A000067-datasheet.pdf. [Accessed: Feb. 12, 2023]

4.4.7 Revision Table -

Time	Revision made
1.20.2023	**Section created**
2.11.2023	**Adjusted the interface and its properties, update all Sections of the content according to the review**
2.17.2023	**Update the Interface Validation Value and Revision Statement**

4.4.8 Revision Statement

In the recent revision on February 11, significant changes were made to the interface and schematic of a specific block. The interface was adjusted from an external DC supply to a DC supply from the microcontroller, which is expected to provide a more reliable and efficient supply for the block. Additionally, the schematic of the circuit was updated to ensure that it was more concise and reliable, which should help to streamline the overall design process.

In order to provide a more comprehensive and detailed document, each section was expanded to include additional information. The content of each section was thoroughly examined and a summary of the main points covered in the Description section was provided. This summary will help the reader to gain a better understanding of the overall purpose and functionality of the block.

In the Design section, the important components of the block were introduced, as suggested. This will help the reader to better understand the design process and the various components that were used to create the block. Additionally, more emphasis was placed on the interface validation section, and it was made consistent with the content of the student portal. This ensures that students have access to accurate and up-to-date information.

Finally, the verification plan section was updated to include specific steps that will help the user to better perform technical verification of the module. By breaking down the verification process into specific steps, the user will be able to more easily and effectively ensure that the block is functioning correctly and reliably.

In conclusion, the recent revision on February 11 resulted in significant improvements to the block's design and documentation. By providing more detailed information, more emphasis on important components, and a better verification plan, the block is expected to be more reliable, efficient, and user-friendly.

Section 5: System Verification Evidence

5.1 Universal Constraints:

5.1.1 The System may not include a breadboard -

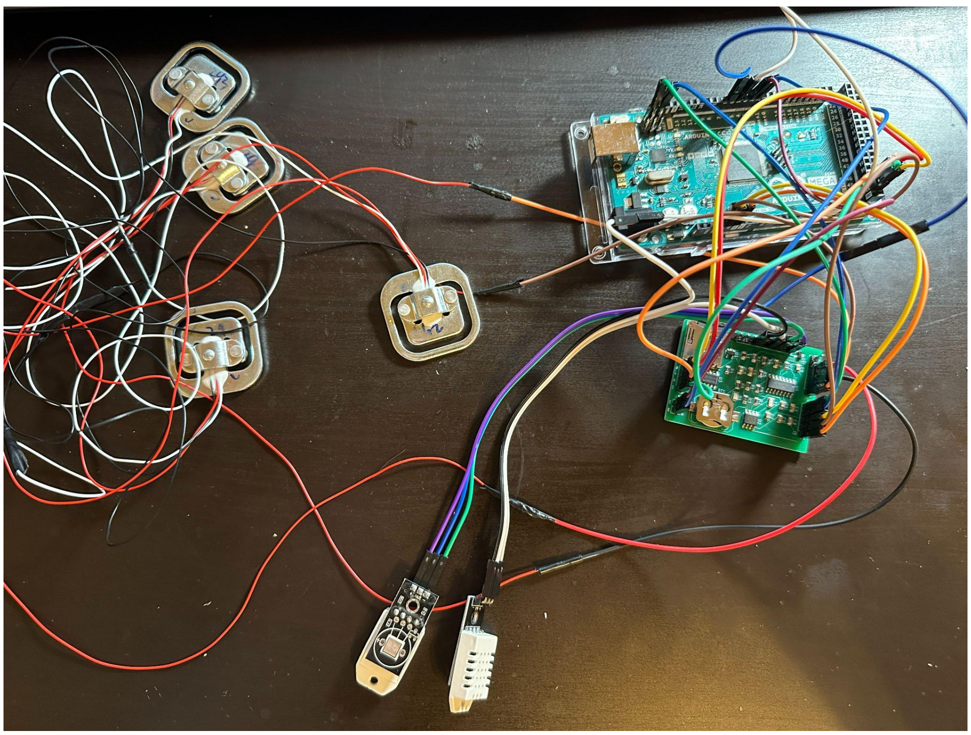


Figure 5.1.1

The picture above, figure 5.1.1 shows our current system and all of the componentes within, except for the battery which will be connected to the Vin and GND on the arduino. As seen in this picture our system is not using a breadboard and is all wired to the arduino through a PCB.

5.1.2 The final system must contain a student designed PCB -

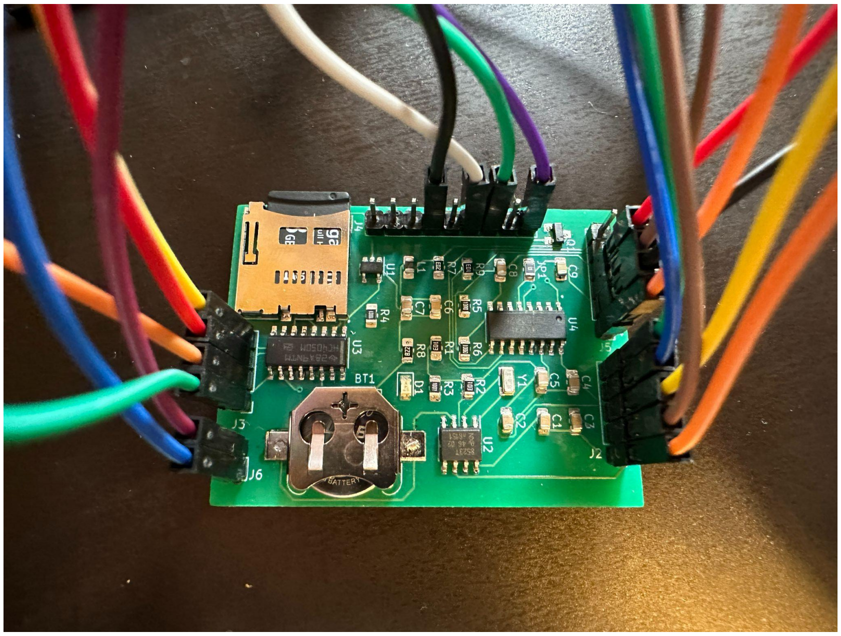


Figure 5.1.2

As seen in the picture above, figure 5.1.2 shows the PCB that our group has designed to work for our system.

5.1.3 All connections to PCB's must use connectors -

Also shown in the picture above, figure 5.1.2 shows that our system's PCB has all its connections using connectors.

5.1.4 All power supplies in the system must be at least 65% efficient -

Input Power (Pin) = 8V x 0.250A = 2W

Output Power (Pout) = 13.57V x 0.130A = 1.765W

Efficiency = Pout / Pin = 1.765W / 2W = 0.883

Efficiency as a percentage = 0.883 x 100% = 88.3%

These calculations show that the power supply's efficiency is 88.3%, which exceeds the minimum requirement of 65%. As a result, the power supply appears to satisfy the efficiency requirements for our project's beehive monitoring system.

Figure 5.1.4.1 Output Power (Pout)

Figure 5.1.4.2 Input Power (Pin)

5.1.5 The system may be no more than 50% built from purchased modules -

Blocks	Built or Purchased
Microcontroller	Built
SD Card	Built
Power	Built
Weight	Built
Temperature/Humidity	Purchased
RTC	Built
Transmitter/Receiver	Purchased
Enclosure	Built
Userguide	Built

Seven out of nine, about 78% of the modules were built by us.

5.2 Requirements:

5.2.1 Date Backup -

5.2.1.1 Project Partner Requirement -

Sensor data shouldn't be lost in the event of the transmitter going offline.

5.2.1.2 Engineering Requirement -

The system will assure delivery of sensor data even if the hub sub-system is disconnected for a period of at least 48 hours.

5.2.1.3 Verification Process -

We will run the system for 5 minutes, allowing it to store its sensor data onto the SD card. After 5 minutes of running the system we will verify that all data has been stored onto the SD card and prove that if the system were to continue writing data at the same rate for the next 48 hours, the card would be able to hold the amount of data that would be stored if disconnected for 48 hours.

5.2.1.4 Testing Evidence -

Below are three pictures that prove our system is able to store all of its data onto an SD card incase transmission is lost. The first, Figure 5.2.1.1 shows the amount of free space that is available on the SD card after formatting it, and Figure 5.2.1.2 shows the free space on the SD card after 5 minutes of running the system and storing all the data on the card. From these figures we can see that the data used up about 32 Kilobytes so if we were to run the system for 48 hours this would end up using about (32*48hrs*60min/5mins) = 18,432 Kilobytes. So after 48 hours of transmission being lost only 18.432 Megabytes of data will be stored on the SD card and there is 8 Gigabytes of data on the card. The third and last picture, Figure 5.2.1.3 shows the contents of the data that is stored onto the SD card showing that it is storing temperature, humidity, and weight.

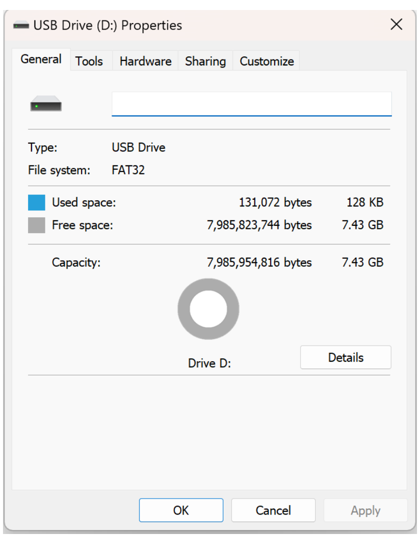


Figure 5.2.1.1

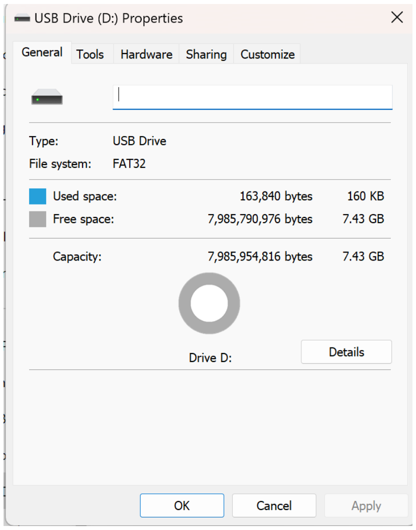


Figure 5.2.1.2

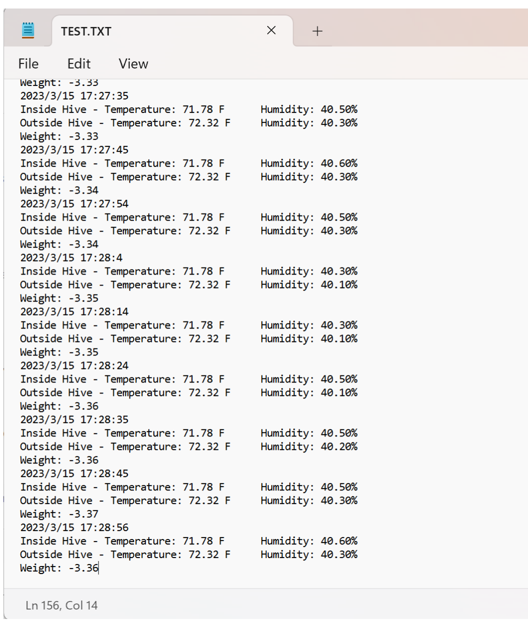


Figure 5.2.1.3

5.2.2 Data Processing -

5.2.2.1 Project Partner Requirement -

The system will generate three line graphs (one for temperatures, one for humidity, and one for weight) showing up to a week of collected data

5.2.2.2 Engineering Requirement -

The optional time must be accurate to each minute and show the overall optional time period. Users can choose different forms of data display, such as choosing to display several types of data on one diagram, or selecting different units to display the data.

5.2.2.3 Verification Process -

After creating a file that contains at least a day’s worth of data (288 points), we will run our script and use that data to generate graphs. Making three different graphs that show how the temperature, humidity, and weight change.

5.2.2.4 Testing Evidence -

Below is an example of what the generated plots would look like. The top being the humidity, the middle being the weight, and the bottom showing both the internal and external temperatures.

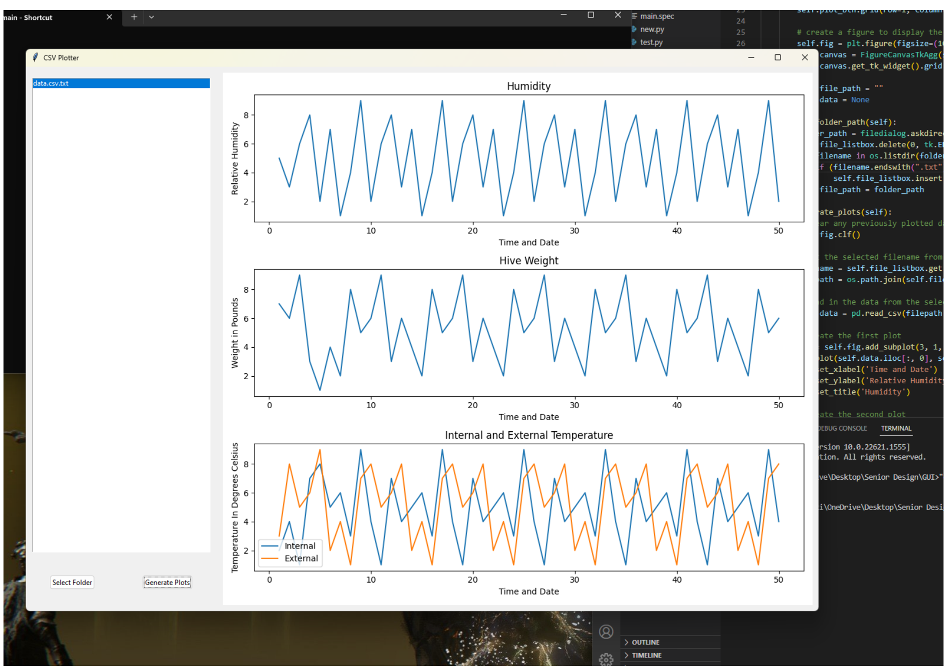


Fig 5.2.2.4: Data Processing

5.2.3 Humidity -

5.2.3.1 Project Partner Requirement -

The system must be able to measure the humidity inside of the hive.

5.2.3.2 Engineering Requirement -

The System will measure humidity inside the hive at least every 5 minutes and within 5% of absolute humidity.

5.2.3.3 Verification Process -

We will run code on our system that will read the humidity from the sensor on the inside of the hive, this code will read the humidity every 10 seconds proving that it can measure the humidity at least every 5 minutes. We will also use a separate humidimeter from our system to compare the values and show that the readings from our system's sensors are within 5% of relative humidity.

5.2.3.4 Testing Evidence -

Below are three pictures proving that our system meets this requirement. The first picture, Figure 5.2.3.1 shows the humidity measured in the room by a separate humidimeter from our system. The second photo, Figure 5.2.3.2 shows the values that are measured from sensors in our system. Comparing the values from these two pictures proves our systems sensor is within 5% of the actual relative humidity. And the last picture, Figure 5.2.3.3 shows another measurement of humidity being taken about twenty seconds after the first measurement shown in Figure 5.2.3.2, this proves that our system is able to take measurements of the humidity at least every five minutes.

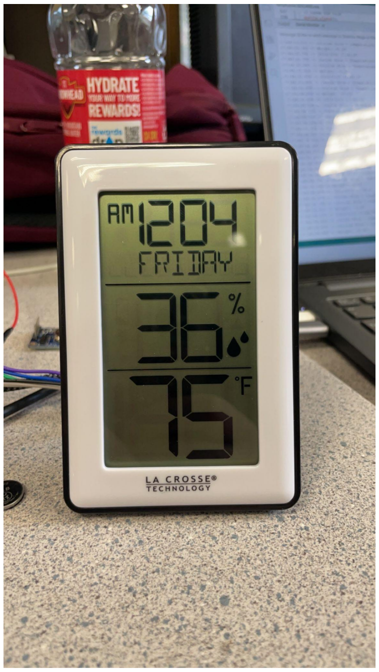


Figure 5.2.3.1

```
17:39:14.590 -> 2023/3/15 17:39:12
17:39:14.623 -> Inside Hive - Temperature: 74.84 F      Humidity: 39.20%
17:39:14.656 -> Outside Hive - Temperature: 74.48 F     Humidity: 39.40%
17:39:14.722 -> Weight: -0.03
```

Figure 5.2.3.2

```
17:47:22.810 -> 2023/3/15 17:39:33
17:47:22.842 -> Inside Hive - Temperature: 74.66 F      Humidity: 39.00%
17:47:22.908 -> Outside Hive - Temperature: 74.30 F     Humidity: 39.40%
17:47:22.941 -> Weight: -0.02
```

Figure 5.2.3.3

5.2.4 Temperature -

5.2.4.1 Project Partner Requirement -

The system must be able to measure the temperature both inside and outside of the hive

5.2.4.2 Engineering Requirement -

The System will measure the temperature inside and outside the hive at least every 5 minutes and within 1° Celsius of the actual temperature.

5.2.4.3 Verification Process -

We will run code on our system that will read the temperature from both the inside and outside temperature sensors, this code will read the temperature every 10 seconds proving that it can measure the temperature at least every 5 minutes. We will also use a separate thermometer from our system to compare the values and show that the readings from our system's sensors are within 1° Celsius of the actual temperature.

5.2.4.4 Testing Evidence -

Below are three pictures proving that our system meets this requirement. The first picture, Figure 5.2.4.1 shows the temperature measured in the room by a separate thermometer from our system. The second photo, Figure 5.2.4.2 shows the values that are measured from sensors in our system. Comparing the values from these two pictures proves our system's sensor is within 1° Celsius(1.8°F) of the actual temperature. And the last picture, Figure 5.2.4.3 shows another measurement of temperature being taken about twenty seconds after the first measurement shown in Figure 5.2.4.2, this proves that our system is able to take measurements of the temperature at least every five minutes.

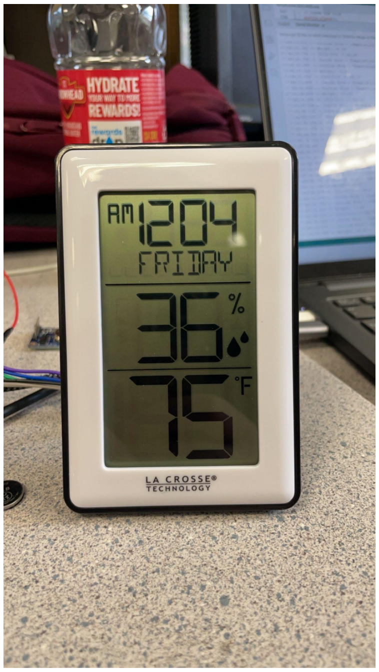


Figure 5.2.4.1

```
17:39:14.590 -> 2023/3/15 17:39:12
17:39:14.623 -> Inside Hive - Temperature: 74.84 F      Humidity: 39.20%
17:39:14.656 -> Outside Hive - Temperature: 74.48 F     Humidity: 39.40%
17:39:14.722 -> Weight: -0.03
```

Figure 5.2.4.2

```
17:47:22.810 -> 2023/3/15 17:39:33
17:47:22.842 -> Inside Hive - Temperature: 74.66 F      Humidity: 39.00%
17:47:22.908 -> Outside Hive - Temperature: 74.30 F     Humidity: 39.40%
17:47:22.941 -> Weight: -0.02
```

Figure 5.2.4.3

5.2.5 Power -

5.2.5.1 Project Partner Requirement -

The system must run through the night.

5.2.5.2 Engineering Requirement -

The system will run for 8 hours without an external power source (from the wall outlet, sun, or otherwise).

5.2.5.3 Verification Process -

We will connect all the components and try it with a 12V 10ah Battery that connects to the MPPT charger controller that controls the power and gives us the load and the battery voltages. After that, we will connect a DMM in case to measure the voltage battery and see how the project consumes power. Letting it for 8 hours working without external power as we use solar power as an external source for our system to recharge the battery.

5.2.5.4 Testing Evidence -

https://drive.google.com/file/d/1_cLPNmvo8NvuwMz6LrJGI_cvvXwLpKTU/view?usp=sharing

5.2.6 RTC -

5.2.6.1 Project Partner Requirement -

Selectable timeframe for data to be displayed

5.2.6.2 Engineering Requirement -

The system will display measurement data with a timestamp that is accurate to within ten seconds per day.

5.2.6.3 Verification Process -

Upload the code, the system will synchronize the PC time. By relying on the coin cell battery on the PCB, the system can keep accurate time without a power supply. Open the Arduino IDE serial monitor, each piece of data collected by the sensors will be given an accurate time stamp.

5.2.6.4 Testing Evidence -

```
13:53:34.107 -> 70.52,47.80,70.34,0.01
13:53:39.219 -> Writing to test.txt...Weight: -0.00 lbs
13:53:39.255 -> done.
13:53:39.255 -> 5/9/2023 13:53:39
13:53:39.255 -> 70.52,47.90,70.34,-0.00
13:53:44.366 -> Writing to test.txt...Weight: 0.00 lbs
13:53:44.366 -> done.
13:53:44.366 -> 5/9/2023 13:53:44
13:53:44.366 -> 70.52,47.80,70.34,0.00
13:53:49.498 -> Writing to test.txt...Weight: -0.00 lbs
13:53:49.541 -> done.
13:53:49.541 -> 5/9/2023 13:53:49
13:53:49.541 -> 70.52,47.90,70.34,-0.00
13:53:54.645 -> Writing to test.txt...Weight: -0.00 lbs
13:53:54.690 -> done.
13:53:54.690 -> 5/9/2023 13:53:54
13:53:54.690 -> 70.52,47.80,70.34,-0.00
13:53:59.797 -> Writing to test.txt...Weight: -0.01 lbs
13:53:59.844 -> done.
13:53:59.844 -> 5/9/2023 13:53:59
13:53:59.844 -> 70.52,47.80,70.34,-0.01
```

5.2.7 Transmit -

5.2.7.1 Project Partner Requirement -

The System must be able to transmit data at least 450ft away

5.2.7.2 Engineering Requirement -

The system will transmit data at a distance of at least 450ft and with a transmission efficiency greater than 99%.

5.2.7.3 Verification Process -

Upload separate codes for the transmitter and receiver, and then adjust the transmitter and receiver antennas so that they are aimed at each other. Keep the receiver 450 feet away from the transmitter and make sure that there are no large objects blocking the view in the middle of the journey. The receiver side will receive data from the system every ten seconds, which includes accurate time stamps, temperature, humidity as well as weight information.

5.2.7.4 Testing Evidence -

https://drive.google.com/file/d/1bT7JTag8frWwv1kPw8GiOWocz9LmLImj/view?usp=share_link

5.2.8 Weight -

5.2.8.1 Project Partner Requirement -

The system must be able to track the weight of the beehive

5.2.8.2 Engineering Requirement -

System must measure the weight of the beehive within 0.5lbs of actual weight

5.2.8.3 Verification Process -

An object of known weight will be placed on the scale and the weights output by the scale over the next 10 seconds will be recorded. The average of those weights will be compared to the known weight of the object. If the average weight is within 1lb of the known weight, this verification is passed.

5.2.8.4 Testing Evidence -

From the two pictures shown below, it can be seen that the weight measured using our project's weight block is very close to what a bathroom scale reads. The two values only being slightly different, our project measuring the weight as 0.09lbs higher, but that is well within the tolerance that we had set of 0.5lbs.

Fig 5.2.8.4.1: Store Bought Scale Measurement

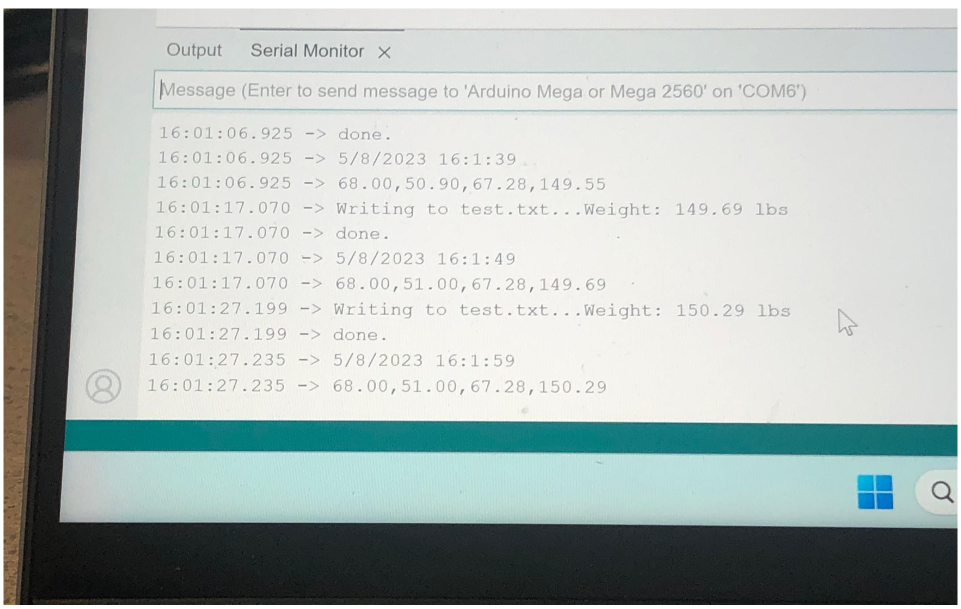


Fig 5.2.8.4.2: System Output

5.3 References and File Links:

System Code:
https://drive.google.com/drive/folders/1V3kDijpoxm9FOcn0hkhg_OhEVMc0LwHs?usp=share_link

Receiver Code:
https://drive.google.com/drive/folders/1n1m4wjCXEvIKB32M2FdIuOoDvYUD13oz?usp=share_link

Cool Term: https://freeware.the-meiers.org/

5.4 Revision Table:

Name	**Date**	**Revision**
-	3/9/23	Section created
Porter	3/15/23	Added content to sections, 5.2.1.3, 5.2.1.4, 5.2.3.3, 5.2.3.4, 5.2.4.3, 5.2.4.4
Saud	5/7/23	Added content to sections, 5.2.5, 5.2.5.3, 5.2.54, 5.2.6.4, 5.1.4, 5.1.4.1, 5.1.4.2, 5.2.6.4
Justin	5/8/23	Added content to sections, 5.2.2.3, 5.2.2.4, 5.2.7.4, 5.2.8.3, 5.2.8.4
Tieying	5/8/23	Added content to sections, 5.2.7.3, 5.2.6.3, 5.3

Table 5.4

Section 6: Project Closing

6.1 Future recommendations:

6.1.1 Technical Recommendations:

1. Develop a mobile application:

   While our current project provides access to the data through a graphical interface, a future team could consider developing a mobile application to provide a more convenient and user-friendly interface for beekeepers. The application could include features such as push notifications for critical alerts, real-time data visualization, and the ability to customize the monitoring settings. “Beyond beekeeping, knowledge cities require sustainability assessment tools that can help decision-making through environmental monitoring. The Summit highlighted that the mobile phone offers the affordances of cameras, sensors, GIS capabilities, screens for analyzing maps or videos and networking ability.”[1]

2. Add a camera to the system:

   While our current project focuses on monitoring temperature, humidity, and weight, a future team could consider adding a camera to the system. This would provide valuable visual information about the state of the hive, such as the number of bees present or the presence of pests. Additionally, it could be used to monitor hive behavior and detect potential issues.“In the past few years, the number of electronic devices for beekeepers has mushroomed, Burlew says. I’ve frequently been asked to test these new devices, so I have quite a bit of experience with them. But bottom line, the only one I use on a regular basis is an infrared camera.”[2]

3. Expand the range of data collected:

   While our current project focuses on monitoring the temperature, humidity, and weight of the beehive, there are other parameters that could be tracked to gain a more complete picture of the hive's health. For example, tracking the pH level of the hive could help to detect issues with the bees food supply, or tracking the sound level could help to detect potential problems with the bees behavior.

4. Assemble the RF system on the PCB.

   While our current project uses a transmitter and receiver module. We could design our own RF transmitter and receiver on the PCB. Designing an RF system from scratch can be challenging due to the specialized knowledge required in RF engineering, including frequency, power, noise, modulation, and demodulation. Antennas are also a critical component and designing an effective one requires consideration of factors such as size, gain, and radiation pattern. These complexities can make it difficult to design a perfect RF system within a short timeframe.

6.1.2 Global Impact Recommendations:

1. Addition of a shielding on the enclosure:

   This is to limit the amount of radio waves being transmitted towards the hive. Some studies have shown that prolonged exposure to RF radiation can have negative effects on overall hive health. While it would be difficult to eliminate the risk of radio waves interfering with the bees, it can be reduced as much as possible. Even with the use of a directional antenna, there is still a small amount of RF radiation being released in every direction.

2. Implement sustainable and environmentally friendly practices:

   In addition to designing a beehive monitoring system that is beneficial for beekeepers, a future team could also consider the global impact of their project. One recommendation would be to implement sustainable and environmentally friendly practices in the design and manufacturing of the system. This could include using biodegradable materials for the enclosure, reducing the amount of packaging waste, and implementing energy-efficient practices in the operation of the system [3].

6.1.3 Teamwork Recommendations:

1. Set for the team specific goals and targets:

   To guarantee that everyone is working toward a shared goal and to give the project a clear direction, the team must establish specific goals and objectives. Coming up with acceptable schedules and deadlines. "Teamwork is crucial for maintaining a successful business model. While working collaboratively, individuals contribute their diverse perspectives, skills, and experience to accomplish their goals, making it easier to overcome challenges and achieve objectives" [4].

2. Being sure that everyone is aware of their responsibilities:

   To avoid misunderstandings, unnecessary work, and missed deadlines, it is important to make sure that everyone in the team is aware of their tasks. This entails defining each person's specific roles and responsibilities and making sure that everyone gets the resources and support that they need to carry out their tasks.

3. Establish effective communication channels:

   Effective communication is crucial for successful team collaboration. Make sure to establish communication channels that facilitate quick and efficient communication, such as a group chat platform, weekly team meetings, and regular progress updates. The team should also ensure that everyone is aware of the communication channels being used and is comfortable using them.

4. Create a positive team environment:

   Team morale and motivation can have a significant impact on the success of a project. To create a positive team environment, a future team could prioritize team building activities and create a supportive atmosphere where everyone feels valued and appreciated. This could include celebrating team accomplishments, acknowledging individual contributions, and providing opportunities for growth and development.

6.1.4 References and File Links:

[1] L. Neville, "CCTP-820: Leading by design – principles of technical and Social Systems," CCTP820 Leading by Design Principles of Technical and Social Systems, https://blogs.commons.georgetown.edu/cctp-820-fall2015/a-case-design-for-a-beekeeping-mobile-app-networking-of-knowledge-cities/ (accessed May 14, 2023).

[2] "Saving beehives with Flir Thermal cameras," Saving Beehives Using the FLIR ONE | Teledyne FLIR, https://www.flir.com/discover/professional-tools/saving-beehives-with-flir-thermal-cameras/ (accessed May 14, 2023).

[3] "Cleaning Up Electronic Waste," EPA, https://www.epa.gov/international-cooperation/cleaning-electronic-waste-e-waste (accessed May 14, 2023).

[4] Indeed. (n.d.). 14 reasons why teamwork is important in the workplace, https://au.indeed.com/career-advice/career-development/why-teamwork-is-important (Accessed: Apr. 28, 2023).

6.2 Project Artifact Summary With Links:

PCB Layout:
https://drive.google.com/file/d/10uMd8zLBrYsxZxBPEz2cLkWBYuZdagZL/view?usp=sharing

RTC Module Test Code:
https://drive.google.com/file/d/1ct8Rk1BAgR186M8djN78_6MfuC2ouXRU/view?usp=sharing

Transmitter and Receiver Test Code:
https://drive.google.com/file/d/1s448O5jrvPfh5vNQ3Bqhx5n7wGVy66Ob/view?usp=sharing

SD Card Module Code:
https://drive.google.com/file/d/15oV7ydnvpj2DCuBAkxJ3e5lWadj6_a-9/view?usp=sharing

System Code:

https://drive.google.com/drive/folders/1V3kDijpoxm9FOcn0hkhg_OhEVMc0LwHs?usp=sharing

Weight Scale code:

https://drive.google.com/drive/folders/1A8VhNO2rkATm_dsWgvvk_R-g1chii9bw?usp=share_link

User Guide:

https://drive.google.com/file/d/1_BeK-B4jPjoNYrAJDfIJ8NnTQP-_EQze/view?usp=sharing

Project BOM:

https://docs.google.com/spreadsheets/d/1OqriDH6jEnlc650N6ULkfFzwr-6_MbBJepRaQF-P_p4/edit?usp=sharing

Enclosure:

https://drive.google.com/file/d/1sXgM2b9HWsQLPHyfGh8-GMs_b9C7APYM/view?usp=share_link

https://drive.google.com/file/d/1sXgM2b9HWsQLPHyfGh8-GMs_b9C7APYM/view?usp=share_link

6.3 Presentation Material:

Project Showcase Link:

https://eecs.engineering.oregonstate.edu/project-showcase/projects/edit?id=mH73tv4NUnGaVBBl

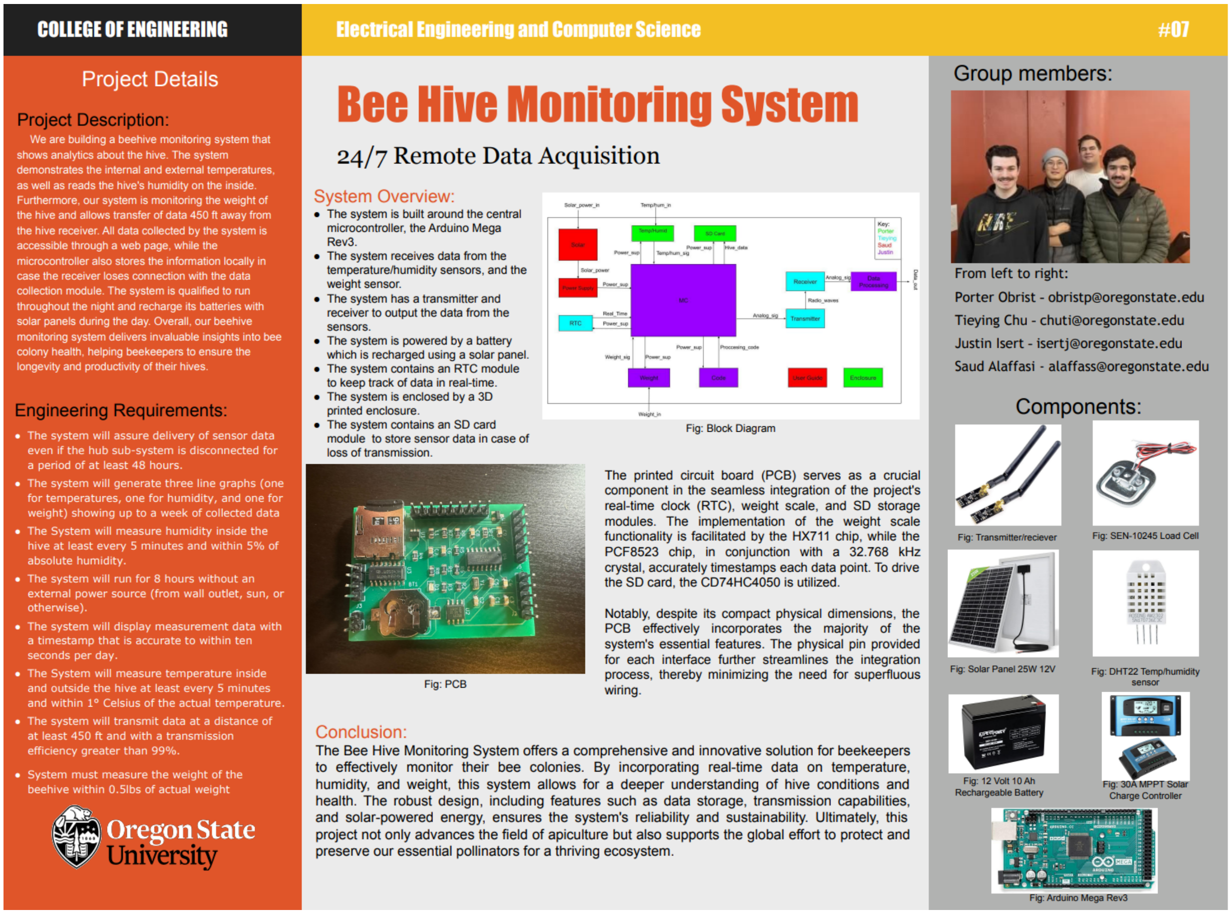


Figure 6.3.1: Project Poster