

Mobile Hive Checker

By

Fiona Cashin (fcashin2)

Olivia Guido (ojguido2)

Rawda Abdeltawab (rawdaka2)

Final Report for ECE 445, Senior Design, Spring 2026

TA: Hossein Atae

6 May 2026

Project No. 1

Abstract

The Mobile Hive Checker is a device that monitors the health of a beehive without disturbing the hive by measuring temperature, humidity, and carbon dioxide (CO₂) levels. The design is divided into power, sensor, and microcontroller subsystems. The 12 V input is stepped down to 5 V and 3.3 V, which are distributed to respective subsystems for proper data capturing. The microcontroller acts as the brain of the device managing communication and overall system operation. Through verification in the lab and in beehives, the device is able to take in environmental data while meeting our specified performance requirements. The design provides a non-invasive solution to a continual beekeeping issue, providing reliable real-time beehive monitoring.

Contents

| | |
|--|-----------|
| 1. Introduction..... | 1 |
| 1.1 Problem..... | 1 |
| 1.2 Solution..... | 1 |
| 1.3 High Level Requirements..... | 2 |
| 1.4 Block Diagram..... | 2 |
| 1.5 Subsystem Overview..... | 3 |
| 1.5.1 Power Subsystem Overview..... | 3 |
| 1.5.2 Sensor Subsystem Overview..... | 3 |
| 1.5.3 Microcontroller Subsystem Overview..... | 4 |
| 2. Design..... | 5 |
| 2.1 Hardware Design..... | 5 |
| 2.1.1 Power Feasibility Analysis..... | 5 |
| 2.1.2 Operating Voltage & Regulation..... | 6 |
| 2.1.3 Bidirectional Level Shifter for Display..... | 6 |
| 2.2 Software Design..... | 7 |
| 2.2.1 Finite State Machine (FSM) Logic..... | 7 |
| 2.2.2 Wi-Fi Cloud Integration & Remote Monitoring..... | 8 |
| 2.2.3 Temperature and Humidity Sensor Interaction..... | 8 |
| 2.2.4 MH-Z19C CO ₂ Sensor..... | 9 |
| 2.2.5 Threshold Comparison..... | 10 |
| 2.2.6 Display Control Logic..... | 11 |
| 3. Design Verification..... | 12 |
| 3.1 Power Subsystem Verification..... | 12 |
| 3.2 Sensor Subsystem Verification..... | 12 |
| 3.3 Microcontroller Subsystem Verification..... | 13 |
| 4. Cost..... | 14 |
| 4.1 Labor..... | 14 |
| 4.2 Parts..... | 14 |
| 4.3 Schedule..... | 16 |
| 5. Conclusion..... | 17 |
| 5.1 Accomplishments..... | 17 |
| 5.2 Uncertainties..... | 17 |
| 5.3 Ethical Considerations..... | 17 |
| 5.4 Future Work..... | 18 |
| References..... | 19 |
| Appendix A. Requirement and Verification Table..... | 20 |
| Appendix B. Additional Software Images..... | 22 |
| Appendix C. Schedule..... | 24 |

1. Introduction

1.1 Problem

Beekeepers must routinely monitor the hive conditions to maintain healthy colonies and prevent colony loss. Critical environmental parameters such as temperature, humidity, and CO₂ concentration directly impact brood development, honey production, and overall colony health. However, traditional monitoring requires physically opening the hive, which stresses the bees, disrupts internal thermal regulation, and can negatively affect productivity and behavior when the hive is opened frequently.

Conversely, insufficient monitoring increases the risk of swarming, starvation, disease, or freezing, any of which can result in the loss of an entire colony. Each lost hive can cost a beekeeper approximately \$100–\$200 in replacement costs alone, not including lost honey production or pollination value. On a broader scale, honeybee decline is associated with habitat loss, pesticide exposure, climate change, and improper apiary management [1]. Because pollination supports global agriculture and ecosystem stability, hive loss represents both an economic and environmental concern.

While commercial hive monitoring systems exist, many require permanent installation and cost approximately \$100 per hive, making them impractical for hobbyist and small-scale beekeepers [2]. Additionally, many products are manufactured overseas, limiting accessibility and domestic technical support. These factors motivate the need for a low-cost, portable, and minimally invasive hive health assessment device.

1.2 Solution

The proposed solution is a portable Mobile Hive Checker device that allows beekeepers to assess colony health without opening the hive. Two slim sensors, attached via cables, are inserted through the hive entrance, while the main electronics enclosure remains external. This minimizes disruption to the bees while enabling measurement of internal hive conditions.

The sensors measure temperature, relative humidity, and CO₂ concentration. An ESP32 microcontroller processes sensor data and displays readings on an integrated screen. The system compares measured values against predefined healthy operating ranges and provides a clear “PASSED” or “FAILED” result. This allows beekeepers to quickly determine whether intervention is necessary while avoiding unnecessary hive disturbance. The system is battery-powered and portable, allowing it to be used across multiple hives without permanent installation.

1.3 High Level Requirements

The following is a list of three quantitative characteristics this project should exhibit to solve the problem.

1. **Measurement Accuracy:** The device will measure hive temperature with an accuracy of ± 0.54 °F (± 0.3 °C), relative humidity with an accuracy of ± 2 % RH [3], and CO₂ concentration with an accuracy of $\pm(50$ ppm + 5 % of reading) [4].
2. **User Interaction:** The device will respond to a push button input. Pressing the ON/OFF button shall power the display and show a Start Screen, followed by a Testing Screen during data acquisition. Pressing the Reset button shall return the device to the Start Screen. A manual boot button shall allow firmware recovery if necessary.
3. **User Output:** The device will display measured temperature, humidity, and CO₂ values. The device will display “PASSED” if all measurements are within acceptable ranges (70–97 °F, 50–60 % RH, 400<CO₂< 5,000 ppm) along with ✘ or ✔ for each measurement. If any measurement falls outside these ranges, the device shall display “FAILED.”

1.4 Block Diagram

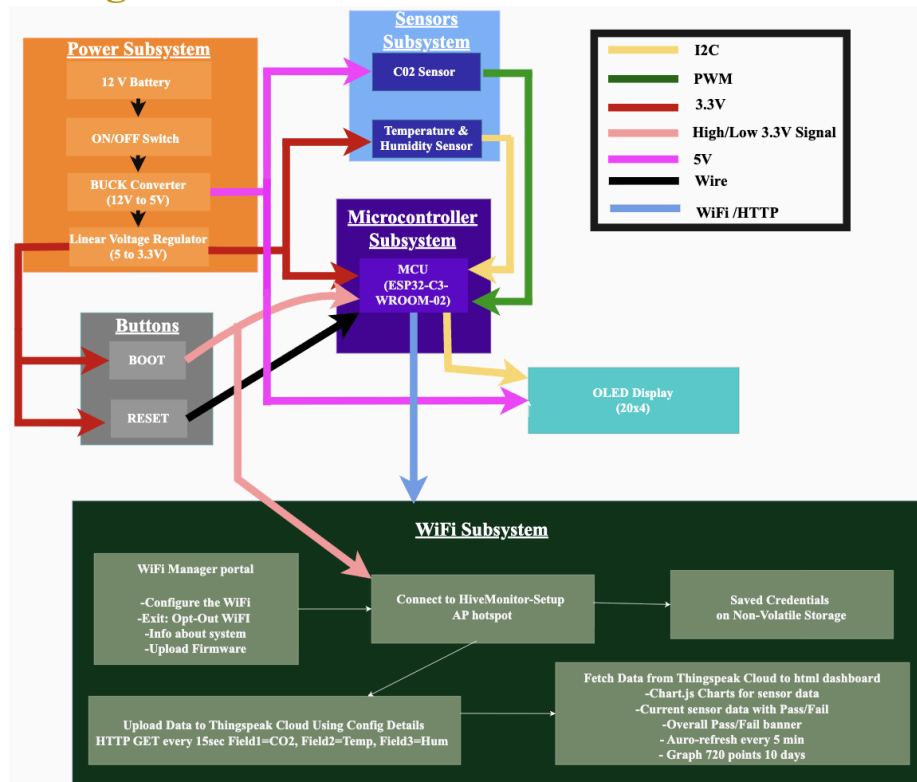


Figure 1. Block Diagram of Design

The main difference of this block diagram from our proposal is that we added Wi-Fi as presented in Figure 1. In the upcoming sections, we will describe the three subsystems shown in the Block Diagram along with the updated Wi-Fi section.

1.5 Subsystem Overview

1.5.1 Power Subsystem Overview

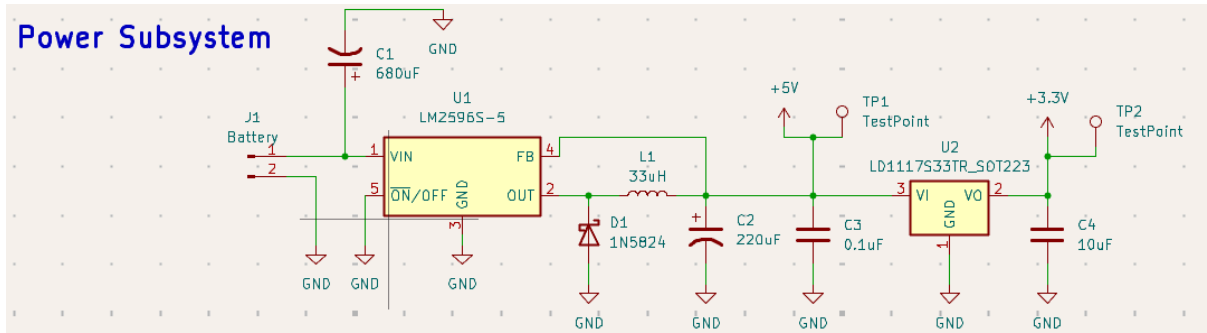


Figure 2. Power Subsystem Schematic

Figure 2 shows the power subsystem where it is powered by a 12 V battery. The battery voltage is stepped down to 5 V using a DC-DC buck converter to improve efficiency. The 5 V rail powers the CO₂ sensor and the display. A 3.3 V low-dropout (LDO) regulator steps the 5 V rail down to 3.3 V to power the ESP32 microcontroller and Temperature and Humidity Sensor. The voltage rails are labeled and distributed according to subsystem requirements.

1.5.2 Sensor Subsystem Overview

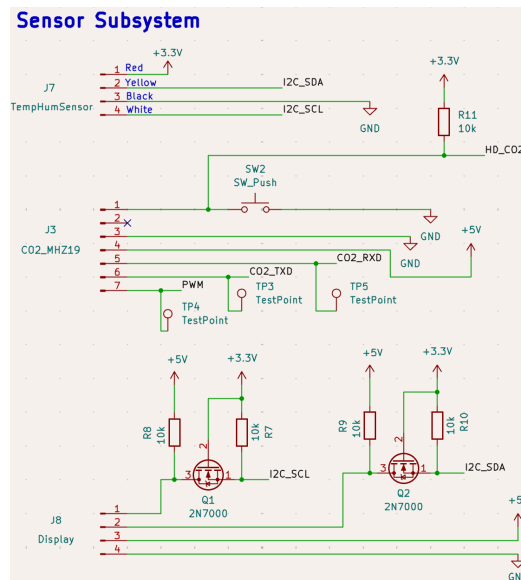


Figure 3. Sensor Subsystem Schematic

Figure 3 shows the sensor subsystem, where it measures internal hive temperature, humidity, and CO₂ concentration. The temperature/humidity sensor and CO₂ sensor are mounted on extended cables, allowing insertion into the hive while keeping the main enclosure external. Sensor outputs interface with the ESP32 microcontroller for processing.

Maintaining temperature between 70–97 °F is critical for brood survival. The selected temperature sensor provides ±0.54 °F accuracy, sufficient to detect meaningful deviations. The sensor’s range is -40 °F to 176 °F [3].

Humidity must remain between 50–60 % RH for proper nectar dehydration and brood health. The selected humidity sensor provides ±2 % RH accuracy and the sensor’s range is 0-100 % RH [3].

CO₂ concentrations above 0.5% (5,000 ppm) may indicate overcrowding or poor ventilation. The selected CO₂ sensor measures 400–5,000 ppm with ±(50 ppm + 5 % of reading) accuracy [4].

1.5.3 Microcontroller Subsystem Overview

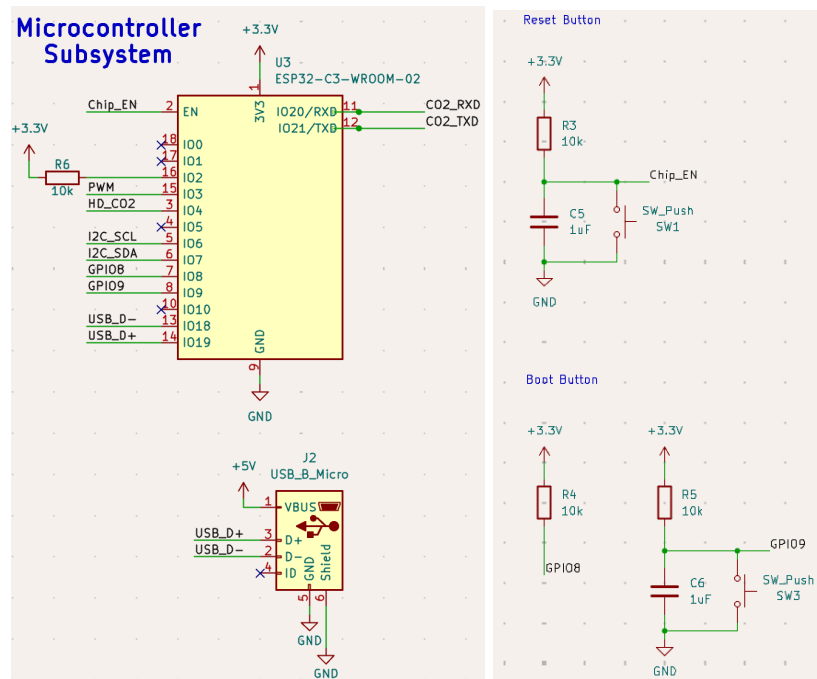


Figure 4. *Microcontroller Subsystem Schematic*

Figure 4 shows the MCU subsystem where the ESP32-C3-WROOM-02 MCU collects data from all sensors and processes the results. Temperature and humidity data are transmitted via I²C [3] while the CO₂ sensor provides an Pulse-Width Modulation(PWM) signal [4]. The ESP32 processes measurements and compares them against defined thresholds. Results are displayed via I²C to the LCD and the ESP32 operates at 3.3 V [5].

2. Design

2.1 Hardware Design

2.1.1 Power Feasibility Analysis

Another important consideration in our design is the ability of the battery to supply sufficient current to all components during operation. Because the device is intended to be portable and used across multiple hives, accurate estimation of current draw is necessary to ensure reliable operation and acceptable runtime.

A 12 V rechargeable Li-ion battery was selected instead of a 9V Duracell battery because of the current consumption of our components. Equation (1) shows the maximum current our battery will need to support.

$$345 \text{ mA} + 0.00098 \text{ mA} + 125 \text{ mA} + 40 \text{ mA} = 510.00098 \text{ mA} \quad (\text{Equation 1})$$

This results in an approximate total current draw of 510.0098 mA, which is close to the upper limit of a typical 9V battery's current capability. Because the estimated draw approaches the maximum supply of a 9V battery, we decided to use a 12 V battery.

Our 12 V battery has a rating of 1200mAh. Since the current draw should not go over about 511mA, we included a 1A fuse on the battery so the battery cannot provide more than 1A. This protects the internal components from breaking.

Table 1. *Voltage and Current Ratings of Components*

| Component | Operating Voltage | Maximum Current Draw |
|------------------------------------|---|---------------------------|
| MCU (Microcontroller) | 3.0 - 3.6 V | 345 mA (peak with Wi-Fi) |
| Temperature and Humidity Sensor | 3.3 - 5.2V (recommended supply is 5 V) | 0.00098 mA (measuring) |
| CO ₂ Sensor | 0 - 5 V (recommended supply is 5 V) | 125 mA (absolute maximum) |
| LCD Display | 2.8 - 5.5 V | 40 mA |

Since the battery is 1200mAh, Equation (2) calculation shows the battery should last about 2 hours and 21 minutes at full load.

$$\frac{1200 \text{ mAh}}{510.00098 \text{ mA}} = 2.353 \text{ (hours)} \quad (\text{Equation 2})$$

2.1.2 Operating Voltage & Regulation

The power subsystem powers the sensors, display, and MCU. To convert our battery's 12 V to 5 V, a Simple Switcher Step-Down Regulator was used.

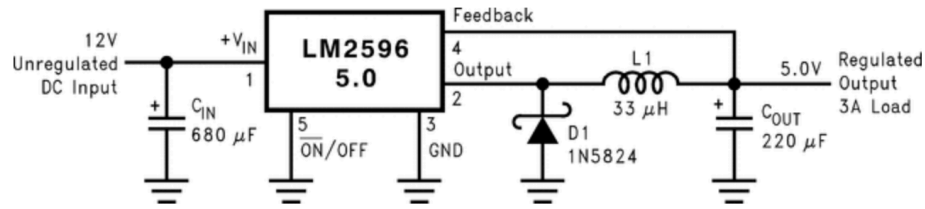


Figure 5. LM2596 Typical Application

As shown in Figure 5, a 12 V battery is typical to use for this component and the output current of 3A is more than enough as calculated in our Power Feasibility Analysis. The 5 V powers the display and the CO₂ sensor.

This circuit is then directly connected to the Low Drop Voltage Regulator to decrease the voltage even more for the MCU and Temperature and Humidity Sensor as shown in Figure 6.

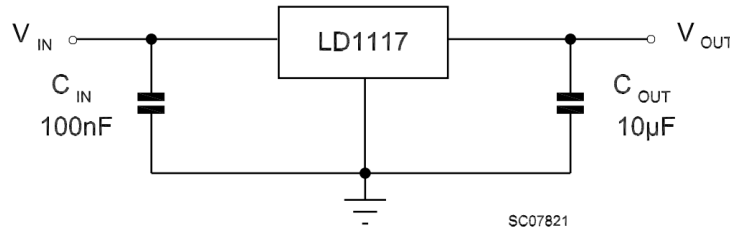


Figure 6. LD1117 Typical Application

2.1.3 Bidirectional Level Shifter for Display

A Bidirectional Level Shifter is a MOSFET that allows components operating at different voltages to communicate safely in both communication directions. The display SCL/SDA signal is 5V while the MCU can only tolerate 3.3 V +/- 0.3 V signal [5]. Therefore, we used a Bidirectional Level Shifter between the display and the MCU [6]. The level shifter source is connected to the lower-voltage side. When the 5 V (HV) side is pulled LOW (0 V), current flows through the body diode of the MOSFET [6]. This creates a voltage difference between the gate and source (VGS), which turns the MOSFET ON. When ON, it has low resistance, so the 3.3 V (LV) side is also pulled LOW.

When the 5 V (HV) side is pulled HIGH, the body diode is reverse-biased and does not conduct [6]. There is no voltage difference to turn the MOSFET on, so it stays OFF. In our case, we used a 10 kΩ pull-up resistor to pull the LV side HIGH to 3.3 V.

2.2 Software Design

2.2.1 Finite State Machine (FSM) Logic

The microcontroller's software is the central decision-making engine of our system. It is responsible for continuously acquiring data from all onboard sensors, processing the measurements, managing Wi-Fi connectivity, updating the display with the latest results and uploading data to the cloud. The system operates using a finite state machine (FSM) built around a simple polling-based control loop that achieves periodic sensor reads, evaluates system conditions, and transitions between states as needed to maintain a safe and reliable operation. The firmware is developed using Arduino IDE 2.0 and flashed on a microcontroller via USB. As shown in Figure 7, the main control flow states are:

- **START/IDLE:** system initialization where display shows a start screen with project name and information about buttons after the ON/OFF button is pressed. All sensors are powered but not actively measuring.
- **Wi-Fi:** the device broadcasts a "Hive-Monitor-Setup" hotspot as shown in Figure B1 where the user connects their phone to it and enters their network credentials via portal. The credentials are saved to the flash so it will auto-correct when the device is used again. To clear the saved Wi-Fi credentials and restart the setup process, the user must press and hold the BOOT button for 3 seconds during startup. If the portal times out in 3 minutes or the user chooses to opt out of Wi-Fi, the system will proceed in offline mode.
- **TESTING:** this is triggered once the system validates that sensors are ready for operation. It could be calibration wait or warming up phase for sensors. Then the MCU polls the temperature and humidity sensor over I²C and reads the analog output of the CO₂ sensor via its PWM pin. The display will show a "testing phase".
- **RESULT:** Display all three measured values (CO₂ in ppm, temperature in °F, and relative humidity in %RH) along with an PASS/FAIL status overall if one of measurement is over the threshold values.
- **RESET:** this is triggered by pressing the Reset button at any time where it returns the firmware to the START/IDLE state and clears all stored measurement values.

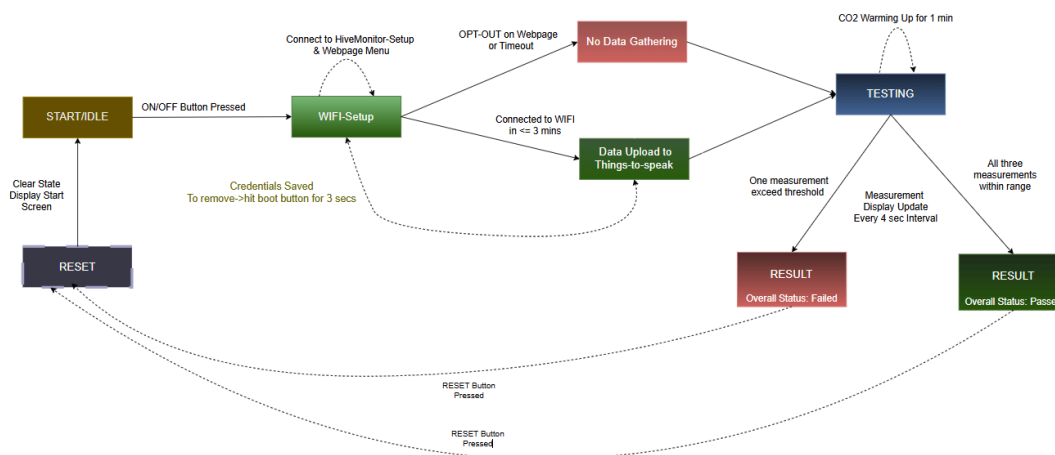


Figure 7. Finite State Machine (FSM) Logic

2.2.2 Wi-Fi Cloud Integration & Remote Monitoring

After completing all our requirements, we wanted to enhance user usability by uploading data to the cloud via ThingSpeak where it collects CO₂, temperature and humidity data every 15 seconds via HTTP GET requests. This integration allows automated data logging and time-series visualization to allow beekeepers to track the health of the hive over 30 days to form trends as shown in Figure 8. The ThingSpeak channel offers downloadable graphs of the measurements as CSV files for more analysis if needed as shown in Figure B2. To make user experience more enhanced, a custom HTML dashboard was developed to add measurements into a single interface. By polling the ThingSpeak API, the dashboard will show color-coded graphs for all three measurements, trend analysis and PASS/FAIL indication as shown in Figure 9.

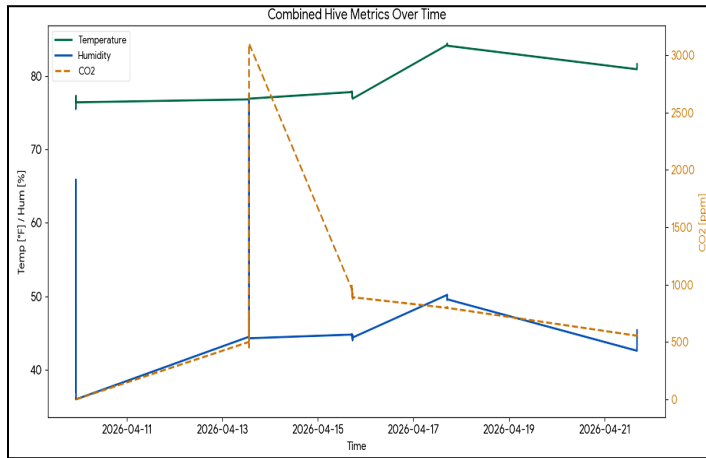


Figure 8. ThingSpeak Graph From CSV Files

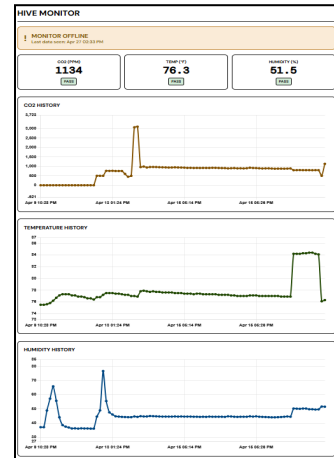


Figure 9. HTML Web Dashboard

2.2.3 Temperature and Humidity Sensor Interaction

The AM2301B is a sensor that measures humidity and temperature [3]. It communicates to the MCU using either standard or Fast I²C protocol mode at a 7-bit device address of 0x38 on its own bus compared to the display as shown in Figure B3. [3]. There are two processes for the I²C communication protocol for sensors: Initialization and Measurement Figure 10 & 11 [3]. We do not read CRC check data as there is no need for it.

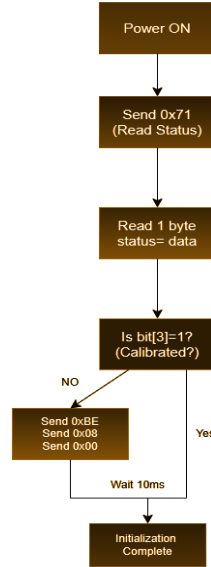


Figure 10. Initialization Sequence

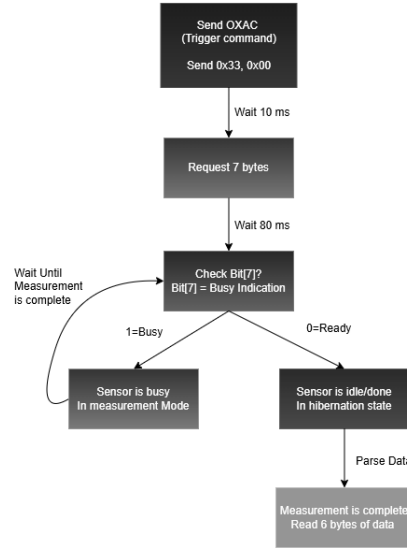


Figure 11. Measurement Sequence

SIGNAL CONVERSION FORMULAS : temperature and humidity are computed from the raw 20-bit values as shown in Figure B4 and using the Equations (3) & (4) defined in the datasheet after the measurement is complete [3]:

$$T[{}^{\circ}C] = (S_T/2^{20}) \times (200 - 50); S_T \text{ is raw temperature} \quad (\text{Equation 3})$$

$$RH[\%] = (S_{RH}/2^{20}) \times 100\%; S_{SH} \text{ is raw relative humidity} \quad (\text{Equation 4})$$

Also, temperature is displayed in ${}^{\circ}F$ which is derived by applying the standard Celsius to Fahrenheit conversion. According to the datasheet, it recommends sampling no more than once every 2 seconds to avoid self-heating of the sensor so the firmware enforces a 2-second minimum measurement interval [3].

2.2.4 MH-Z19C CO₂ Sensor

The MH-Z19C is a Non-Dispersive Infrared (NDIR) sensor designed to monitor CO₂ concentration within a range of 400-5000 ppm. In our design, any reading exceeding the 5000 ppm threshold is flagged as a critical hazard to the colony. The sensor requires a 5 V power supply [4]. We changed our SEN1059 sensor to MHZ-19C due to a better calibration process and communication protocol. While the sensor supports UART for bidirectional signal and checksum validated data, a Pulse Width Modulation (PWM) signal was chosen as the communication interface for our design to minimize resources and eliminate serial protocol overhead. The CO₂ concentration is determined by measuring the high (TH) and low (TL) durations of the signal cycle (which has a total time of 1004ms). This calculation utilizes Equation (5) and examples provided in Figure 12.

$$CO_2[ppm] = 5000 \times \frac{T_H - 2 \text{ ms}}{(T_H + T_L) - 4 \text{ ms}} \quad (\text{Equation 5})$$

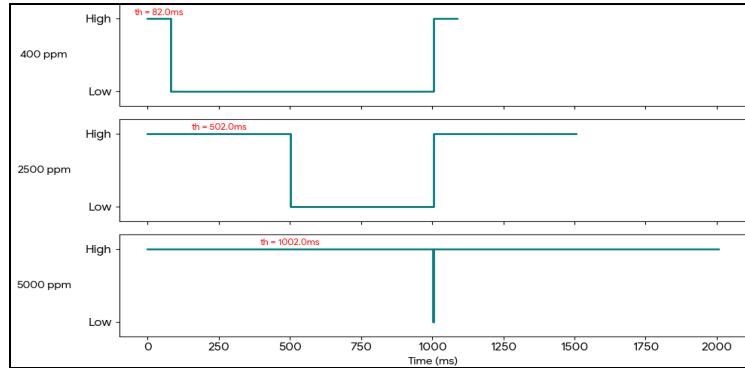


Figure 12. PWM Output Timing for 5,000 ppm Sensor

To ensure data integrity, we reject cycles outside the 800-1200 ms window and apply a 3-sample circular buffer to filter noise and fluctuating values. Since beehives maintain a high CO₂ concentration, which would confuse the Automatic Baseline Calibration (ABC) so we disabled this feature. This is why our design considered a manual calibration after a 20 mins in fresh-air exposure where the MCU pulls the HD pin Low for at least 7 seconds by pressing the button for 3 seconds with guided real-time display feedback.

2.2.5 Threshold Comparison

The microcontroller evaluated each sensor measurement against the predefined healthy beehive operating ranges after the sensor data had been collected.

Table 2. Threshold Comparison Logic

| Measurement | Healthy Range | Sensor Accuracy | Result |
|-------------------------------|---------------|-------------------------|-----------|
| CO ₂ Concentration | < 5,000 ppm | ±(50 ppm + 5 % reading) | PASS/FAIL |
| Temperature | 70-97 °F | ±0.54 °F (±0.3 °C) | PASS/FAIL |
| Relative Humidity | 50-60 % RH | ±2 % RH | PASS/FAIL |

An overall result of “PASSED” is displayed only when all three individual measurements are within their healthy ranges. If any measurement falls outside its range, it will display “FAILED” as the overall result. This ensures the beekeeper is promptly alerted.





2.2.6 Display Control Logic

The 20x4 DIYables display communicates to the microcontroller via I²C communication protocol and uses an edited version of LiquidCrystal_I2C library as the display is on its own bus compared to the temperature and humidity sensor. Since the display drives SDA/SCL at 5 V, we adjust it using a bidirectional level shifter so the MCU's input pins can tolerate the signal. The display has 0-19 cols and 0-3 rows to show the data. During the START screen, the display will show the name of the project and information about reset and power ON/OFF button. During the TESTING state, the display will show "SENSORS WARMING UP AND TESTING..." to activate processing. The display brightness can be reduced via the potentiometer on the display onboard to manage the current draw. During the RESULT state, the display will show four organized rows.

3. Design Verification





3.1 Power Subsystem Verification


Table 3. Power Subsystem Verification

| Requirement | Verification |
|---|---|
| Battery supplies 12 V DC |  |
| Battery supplies 1 A |  |
| Buck Converter converts 12 V to 5 V ±5% |  |
| Linear Voltage Regulator regulates 5 V to 3.3 V ±3% |  |

3.2 Sensor Subsystem Verification


Table 4. Sensor Subsystem Verification

| Requirement | Calculations | Verification |
|--|--|--|
| Temperature & Humidity Sensor operates within ±5% of its rated +3.3 V supply voltage | $3.3 - (3.3(0.05)) = 3.135 \text{ V}$ $3.3 + (3.3(0.05)) = 3.465 \text{ V}$ |   |
| CO ₂ Sensor operates within ±5% of its rated +5 V supply voltage. | $5 - (5(0.05)) = 4.75 \text{ V}$ $5 + (5(0.05)) = 5.25 \text{ V}$ |   |

| | | |
|---|--|--|
| <p>Temperature & Humidity Sensor provides accuracy of ± 0.54 °F and $\pm 2\%$ RH.</p> <p>CO₂ Sensor provides accuracy of $\pm(50$ ppm + 5% reading)</p> | $80 - 0.54 = 79.46$ °F $80 + 0.54 = 80.54$ °F |  <p>(National Oceanic & Atmospheric Administration data from 4/16/2026: 427.58 ppm). MSN Weather(80 °F, 50 % RH)</p> |
| | $58 - 2\% = 56$ RH $58 + 2\% = 60$ RH | |
| | $427.58 - (50 + ((410)(0.05))) = 357.08$ ppm | |
| | $427.58 + (50 + ((410)(0.05))) = 498.08$ ppm | |

3.3 Microcontroller Subsystem Verification

Table 5. Microcontroller Subsystem Verification

| Requirement | Verification |
|---|--|
| <p>MCU receives data from sensors when data is available.</p> <p>Data will be received from:</p> <ul style="list-style-type: none"> - Temperature & Humidity Sensor - CO₂ Sensor | <pre> 117 void lcd_print_labels() { 118 lcd.setCursor(0, 0); lcd.print("CO2: ppm"); 119 lcd.setCursor(0, 1); lcd.print("Temp: F"); </pre> <p>Output Serial Monitor X</p> <p>Message (Enter to send message to 'ESP32C3 Dev Module' on 'COM8')</p> <p>[DATA] CO2: 402 ppm (n=3) Temp: 57.5F Hum: 29.8% Pass/Fail/Fail</p> |
| <p>MCU delivers correct data to the display</p> <p>MCU boots system with the USB-B Micro mode</p> |  |

4. Cost

4.1 Labor

To date, our team (two EE and one CompE student) has averaged 15 hours per person per week on research and construction. While weekly commitments fluctuated between 10 and 30 hours, we believe 15 remains a fair estimate for future labor. Based on UIUC ECE 2023-2024 salary data, we have set labor rates at \$43/hr for EE and \$50/hr for CompE [7]. Our total labor cost is calculated over a 12-week development period, excluding breaks and final demonstrations.

Equation (6) shows cost to pay electrical engineering members:

$$\frac{\$43}{\text{hour}} \times 2.5 \times \frac{15 \text{ hours}}{\text{week}} \times 12 \text{ weeks} = \$19,350 \times 2 \text{ members} = \$38,700 \quad (\text{Equation 6})$$

Equation (7) shows cost to pay an computer engineering member:

$$\frac{\$50}{\text{hour}} \times 2.5 \times \frac{15 \text{ hours}}{\text{week}} \times 12 \text{ weeks} = \$22,500 \times 1 \text{ member} = \$22,500 \quad (\text{Equation 7})$$

We require the machine shop to fabricate two boxes. The first is a ventilated protective case for our CO₂ sensor. The second is a main housing for our monitoring components, which requires precise lid cutouts for the Boot and Reset buttons, as well as the LCD display. The estimated cost per hour for use of the machine shop services outside of this class is \$70 per hour.

Equation (8) shows cost to pay the machine shop:

$$\frac{\$70}{\text{hour}} \times \frac{3 \text{ hours}}{\text{week}} \times 2 \text{ weeks} = \$420 \quad (\text{Equation 8})$$

This brings our total for labor using starting salaries and estimated hours to be \$61,620.

4.2 Parts

Table 6. *Bill of Materials and Cost Breakdown*

| Description | Part # | Manufacturer | Vendor | Cost Per (\$) | # Ordered | Total Cost (\$) |
|-----------------|-------------------|-------------------|---------|---------------|-----------|-----------------|
| MCU | ESP32-C3-WROOM-02 | Espressif Systems | DigiKey | 3.28 | 5 | 16.4 |
| USB-C Connector | USB4085-GF-A GCT | GCT | DigiKey | 0.87 | 1 | 0.87 |

| | | | | | | |
|---------------------------------|---|---|-------------|-------|---------|-----------------|
| Temperature and Humidity Sensor | 5181 | Adafruit | Adafruit | 14.39 | 1 | 14.39 |
| CO ₂ Sensor | MH-Z19C | Winson | Amazon | 27.99 | 1 | 27.99 |
| LCD Display | I2C 20x4 with Backlight and ESP32 | DIYables | Amazon | 7.99 | 1 | 7.99 |
| 12 V Battery | 1200mAh Rechargeable Li-ion Battery | Shenzhen (KBT) Keep Better Tech Electronics Co. | Amazon | 15.99 | 1 | 15.99 |
| Buck Converter (9 - 5 V) | LM2596S-5.0/NOPB | Texas Instruments | DigiKey | 6.97 | 2 | 13.94 |
| Linear Voltage Regulator | LD1117S33 | STM | DigiKey | 0.32 | 3 | 0.96 |
| On/Off Button | RA1113112R | E-Switch | DigiKey | 0.64 | 2 | 1.28 |
| RST Button | PS1024AR | E-Switch | DigiKey | 1.87 | 2 | 2.74 |
| Boot Button | PS1024AB | E-Switch | DigiKey | 1.55 | 2 | 2.10 |
| Resistor | 4.7k, 10k, 5.1k Ω | Stackpole Electronics Inc | ECEB E-Shop | 0 | 4,12,4 | 0 |
| Capacitor | 680 μ F | Rubycon | ECEB E-Shop | 0 | 2 | 0 |
| Capacitor | 220 μ F, 0.1 μ F, 10 μ F, 1 μ F | Rubycon | ECEB E-Shop | 0 | 2,2,2,4 | 0 |
| Inductor | 33 μ H | EPCOS - TDK Electronics | DigiKey | 0.27 | 4 | 1.08 |
| Diode | 1N5824 | Vishay General Semiconductor | DigiKey | 0.62 | 4 | 2.48 |
| COMPOENTS | | | | | | \$108.21 |

| | | | | | | |
|------------------|--|--|--|--|--|--------------------|
| LABOR | | | | | | \$61,620 |
| TOTAL SUM | | | | | | \$61,728.21 |

4.3 Schedule

For a detailed design schedule, please see Figure 1 in Appendix C.

5. Conclusion

5.1 Accomplishments

Throughout the design process, we were able to produce a functional mobile hive monitoring system that detected temperature, humidity, and CO₂ levels while simultaneously uploading the recorded data via Wi-Fi. Beyond a working device, we were also able to present a polished physical device that is currently usable in beehives. As a team, we also developed skills individually. We all were able to learn more about sensor communication protocol, PCB design, surface mount soldering (tools and troubleshooting), and 3D printing. Furthermore, we also had success due to taking the importance of personal protective equipment seriously, especially when using the heat guns, soldering oven, wire cutters, and liquid flux in the lab.

5.2 Uncertainties

Despite our success throughout the semester, there were some uncertainties left in our design. This primarily includes the amount of resistance our sensor casings have to moisture within the hive. The casings we currently have on the sensors allow air to travel into them to be analyzed, but they are not fully weather resistant. When our team visited the beehives on the University of Illinois at Urbana Champaign Student Sustainability Farm, after inserting the sensors into the hives, there was no moisture or honey residue that was left on the sensor casings. However, if hives are not carefully cleaned and monitored, the inside of a hive could be more unpredictable. Additionally, the team was only able to validate the sensors using local weather conditions and air quality reports. Using a controlled environment would enable a more precise assessment of all sensor readings in the design. By having an environment with regulated temperature, humidity, and CO₂, the reliability of the device could be quantified in the future.

5.3 Ethical Considerations

A key aspect to the motivation behind our design was to improve the beekeeping experience. Humans need honeybees. The pollination helps increase the type and amount of plants and crops. This product will help society as it will allow hobby beekeepers continual success for keeping bees alive all year round. Increasing the presence of bees will increase pollination, which will thus increase the prosperity of plants and crops. This is especially important in current times as the number of honeybees is decreasing at a rapid rate due to “habitat loss, improper apiary management, pesticide usage, [and] climate change” [1]. While we cannot control many of these environmental factors, our design aims to improve apiary care through a monitored hive [8]. This project was evaluated for ethical and safety concerns during development and potential misuse. The design aligns with the IEEE and ACM Codes of Ethics by prioritizing public welfare, honesty, competence, and harm prevention [8], [9].

However, there were a few key ethical considerations that we needed to consider before creating our final product. Ethically, the device aims to reduce stress on bee colonies by minimizing hive

intrusion, benefiting the environment through improved pollination and honey production. The system does not collect personal data and presents minimal risk of misuse. Ethical risks are further reduced through accurate performance claims, clear documentation, and disclosure of system limitations [8].

Safety is a major focus of the project. All electronic components are properly insulated and enclosed to prevent electrical hazards, and materials are chosen for safe outdoor and agricultural use. Additionally, our design includes appropriate covers to further protect all components against harsher weather conditions. While the components are not claimed to be waterproof or water-resistant, there should be ample protection against damages from the hive's internal environment. Key safety practices include avoiding battery short circuits, keeping liquids away from electronics, preventing battery overload, using correctly sized wiring, monitoring heat from voltage regulators, ensuring correct battery polarity, and regularly inspecting for exposed wires. All components are housed in a protective enclosure to reduce shock risk. Additionally, all elements of the PCB in this design have been further inspected by a member in the University of Illinois's ECE445 Staff. All team members completed required laboratory safety training, including electrical safety. Overall, the project follows the IEEE Code of Ethics by prioritizing safety, accurately representing performance, working within the team's expertise, and supporting respectful collaboration [8].

5.4 Future Work

While we were able to create a polished, functional product, there are several areas for future improvements. This includes adding a cable management system that allows the user to extend or retract the cables. This both increases convenience and wiring safety into our design. Additionally, we could add a battery management feature that reminds the user to charge the battery on a regular schedule in order to improve reliability.

Beyond technological improvements, we are also interested in producing more Mobile Hive Checkers in order to find further advancements and potentially create a scalable product. As discussed previously, the world needs to promote bee health and local beekeeping, and our design provides a more cost-effective way to reach that goal. With the right tools and management, our design has the potential to create a new niche within the market.

References

- [1] A. Mull, J. Gunnell, S. Hansen, R. Ramirez, A. Walker, C. Zesiger, and L. Spears, "Factors contributing to bee decline," Utah State University Extension, Feb. 2022. [Online]. Available: <https://extension.usu.edu/planthealth/research/factors-contributing-to-bee-decline>
- [2] Beehive Monitoring, "Beehive Monitoring," 2016. [Online]. Available: <https://beehivemonitoring.com/>
- [3] Adafruit Industries, "AM2301B - wired enclosed AHT20 - temperature and humidity sensor," [Online]. Available: https://cdn-shop.adafruit.com/product-files/5181/5181_AM2301B.pdf
- [4] Neumuller Elektr, "Infrared CO2 Sensor Module MH-Z19C Datasheet," 2021. [Online]. Available: <https://www.neumueller.com/datenblatt/neumueller/MH-Z19C.pdf>
- [5] Espressif Systems, *ESP32-C3-WROOM-02 & ESP32-C3-WROOM-02U Datasheet*, [Online]. Available: https://documentation.espressif.com/esp32-c3-wroom-02_datasheet_en.html
- [6] "Level shifter circuit: Theory, types, and applications," *Analog Circuit Design*, Apr. 2, 2024. [Online]. Available: <https://analogcircuitdesign.com/level-shifter-circuit/>
- [7] University of Illinois Urbana-Champaign, "Salary averages," [Online]. Available: <https://ece.illinois.edu/admissions/why-ece/salary-averages>
- [8] Institute of Electrical and Electronics Engineers, *IEEE Code of Ethics*, 2020. [Online]. Available: <http://www.ieee.org/about/corporate/governance/p7-8.html>
- [9] Association for Computing Machinery, *ACM Code of Ethics and Professional Conduct*, 2018. [Online]. Available: <https://www.acm.org/code-of-ethics>

Appendix A. Requirement and Verification Table

Table A1. *Power Subsystem Requirements and Verification*

| Requirements | Verification | Verification status (Y or N) |
|--|--|------------------------------|
| The battery shall supply 12 V DC. | Measure output voltages under load conditions using a multimeter | Y |
| Buck Converter shall regulate 12 V to 5 V $\pm 5\%$ under full load. | Measure output voltages under load conditions using a multimeter | Y |
| The Linear Voltage Regulator shall regulate 5 V to 3.3 V $\pm 3\%$. | Measure output voltages under load conditions using a multimeter | Y |
| The power subsystem shall supply at least 1A peak current without voltage droop exceeding 5 %. | Apply maximum expected load and verify voltage stability. | Y |

Table A2. *Sensor Subsystem Requirements and Verification*

| Requirements | Verification | Verification status (Y or N) |
|---|--|------------------------------|
| The Temperature and Humidity Sensor shall operate within $\pm 5\%$ of its rated +3.3 V supply voltage. | Measure Temperature and Humidity Sensor +3.3 V supply voltage using a multimeter. | Y |
| The CO ₂ Sensor shall operate within $\pm 5\%$ of its rated +5 V supply voltage. | Measure CO ₂ Sensor +5 V supply voltage using a multimeter. | Y |
| The Temperature and Humidity Sensor shall provide temperature accuracy of $\pm 0.54\text{ }^\circ\text{F}$ and humidity accuracy of $\pm 2\%$ RH. | Compare sensor readings to known environmental references or calibrated instruments. | Y |
| The CO ₂ sensor shall provide CO ₂ accuracy of $\pm(50\text{ ppm} + 5\% \text{ reading})$ | Compare sensor readings to known environmental references or calibrated instruments. | Y |

Table A3. *Microcontroller Requirements and Verification*

| Requirements | Verification | Verification status |
|--------------|--------------|---------------------|
|--------------|--------------|---------------------|

| | | (Y or N) |
|---|---|----------|
| <p>The microcontroller successfully receives data from peripheral sensors when data is available.</p> <p>Data will be received from:</p> <ul style="list-style-type: none"> - Temperature and Humidity Sensor - CO₂ Sensor | <ul style="list-style-type: none"> - Observe output data on serial monitor to confirm successful communication - Ensure data is received after push button is pressed | Y |
| The microcontroller successfully boots the system with the USB-C mode | Observe the booting sequence on serial monitor | Y |
| The microcontroller successfully delivers information to the display | Ensure correct output is displayed by comparing output data from the serial monitor to the display | Y |
| The display successfully shows the correct data from the microcontroller conveniently to the user | Ensure correct output is displayed by comparing output data from the serial monitor to the display | Y |

Appendix B. Additional Software Images

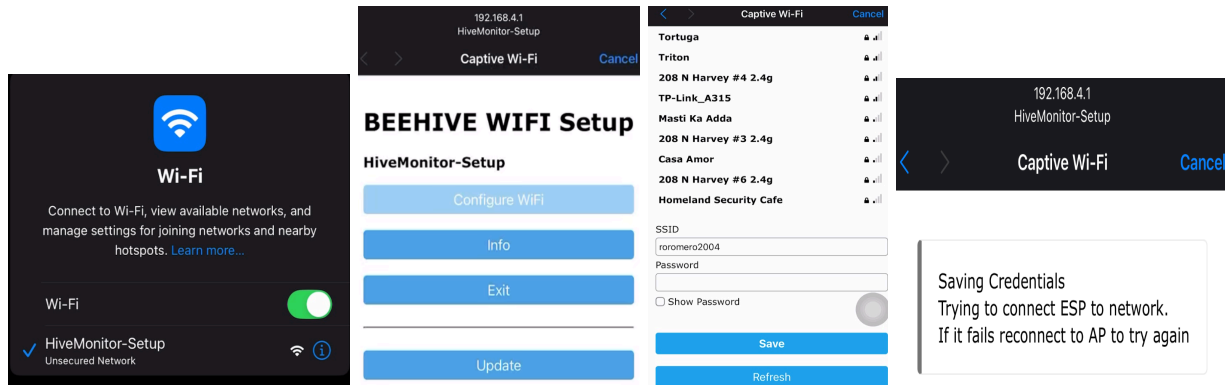


Figure B1. *Hive-Monitor-Setup Portal*

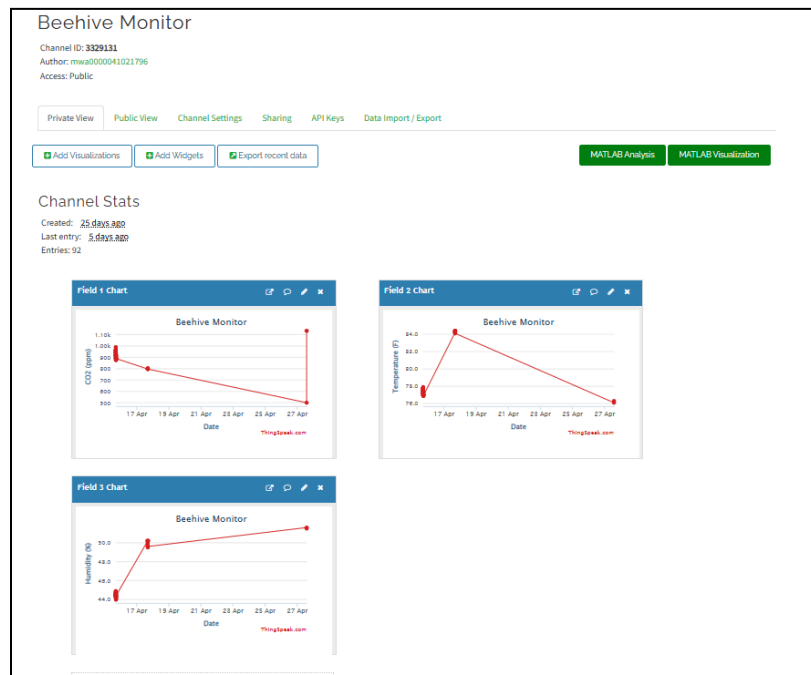
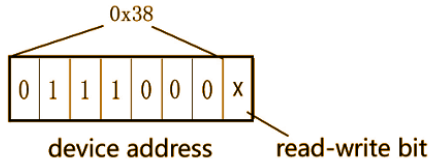


Figure B2. *ThingSpeak Channel Beehive Monitor - ThingSpeak IoT*



| Bits | Significance | Description |
|----------|-----------------|--|
| Bit[7] | Busy indication | 1-Equipment is busy, in measurement mode 0- Equipment is idle, in hibernation state |
| Bit[6:5] | Retain | Retain |
| Bit[4] | Retain | Retain |
| Bit[3] | CAL Enable | 1 - Calibrated 0 - Uncalibrated |
| Bit[2:0] | Retain | Retain |

Figure B3. Status Bit Description

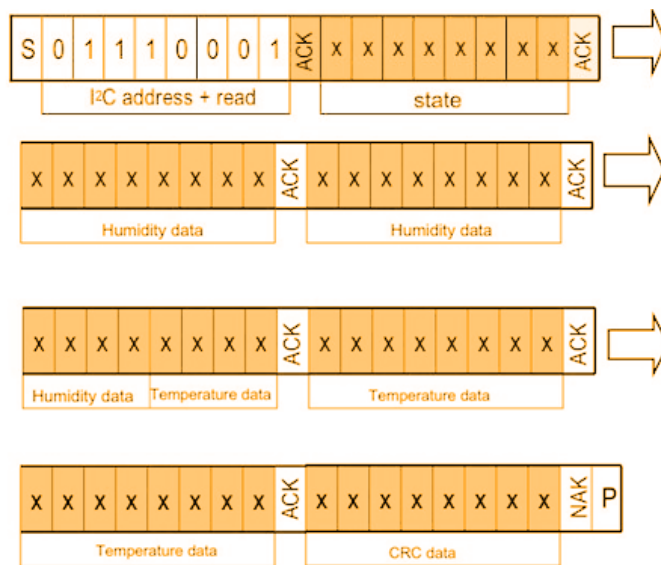


Figure B4. Read Temperature & Humidity Data after Trigger Command

Appendix C. Schedule

Table C1. *Schedule*

| Week | Team's Goals | Person | Deadlines |
|--------------------|--|--|---|
| Jan 19 - Jan 25 | Getting organized in the class and communicating as a group. | Whole group | Initial Web Board Post (Jan 22) |
| Jan 26 - Feb 1 | Each member is in charge of creating 2-3 well thought out project ideas <hr/> Attend a group meeting to discuss pros and cons to each design <hr/> Keep track of ideas on the Web Board <hr/> Begin working on the RFA document | Whole group <hr/> Whole group <hr/> Whole group <hr/> Whole group | Early Project Approval (Jan 29) |
| Feb 2 - Feb 8 | Begin project proposal <hr/> Gather all old lab materials to have a sense of what the group has <hr/> Members start sourcing possible sensors, displays, and MCUs to use | Whole group <hr/> Whole group <hr/> Whole group | Project Approval (Feb 5) |
| Feb 9 - Feb 15 | MCU Pin, Sensors and USB Micro B layout <hr/> PCB Design KiCad <hr/> Finish Project Proposal | Rawda Abdeltawab <hr/> Fiona Cashin <hr/> Whole group | First Meeting with TA (Feb 10th) <hr/> Project Proposal (Feb 13th) |
| Feb 16 - Feb 22 | Finish PCB Design <hr/> Test Functionality of MCU with blinking LED <hr/> Test MCU with display <hr/> Finish team contract | Whole group <hr/> Rawda Abdeltawab <hr/> Rawda Abdeltawab <hr/> Whole group | PCB review (Feb 20th) <hr/> Team Contract (Feb 20th) |
| Feb 23 - Mar 1 | Test temperature and humidity sensor with and without display. <hr/> Test Temperature and humidity sensor with Separate I2C Bus. | Rawda Abdeltawab Olivia Guido <hr/> Rawda Abdeltawab | Design Review Sign-up (Feb 27th) Design Document |

| | | | |
|--------------------|--|---|---|
| | <hr/> Test design ideas for 5 V to 3.3 V SDA/SCL From display to MCU <hr/> Assemble MCU+display + temperature and humidity sensor on breadboard. <hr/> Finish Design Document | <hr/> Rawda Abdeltawab <hr/> Rawda Abdeltawab Olivia Guido <hr/> Whole group | (Feb 27th) |
| Mar 2 - Mar 8 | Test the power across connected components. Use both the power supply from the ECE445 lab and a 12 V battery. <hr/> CO ₂ Sensor Code Development <hr/> If there are any issues with the PCB, a new design is required by March 5th, 4:45pm | Olivia Guido <hr/> Rawda Abdeltawab <hr/> Fiona Cashin Rawda Abdeltawab | Second Round PCBway Orders (Mar 5th) |
| Mar 9 - Mar 15 | CO ₂ & temperature and humidity testing <hr/> Compare first round PCB to design (look for any imperfections) | Olivia Guido Rawda Abdeltawab <hr/> Fiona Cashin | Third Round PCBway Orders (Mar 8th) Last day for revisions to the machine shop (Mar 9th) |
| SPRING BREAK | <i>Enjoy Spring Break while keeping track of any components that are being delivered during this time!</i> | Whole group | |
| Mar 23 - Mar 29 | Continue with testing <hr/> Adjust the code development as needed <hr/> Soldering PCB <hr/> Complete PCB Power Subsystem Testing & Validation | Whole group <hr/> Rawda Abdeltawab <hr/> Olivia Guido <hr/> Olivia Guido Fiona Cashin | Fourth Round PCBway Orders (Mar 26th) |
| Mar 30 - Apr 5 | Everyone should finalize their own individual progress reports in addition to other group duties <hr/> | Whole group | Individual progress reports (April 1st) |

| | | | |
|------------------------|---|--|--|
| | <p>Schedule Demo with UIUC Bee Club</p> <hr/> <p>Continue working on PCB functionality / working out any issues with design</p> <hr/> <p>Finalize extra WIFI/software features</p> | <p>Olivia Guido</p> <hr/> <p>Whole group</p> <hr/> <p>Rawda Abdeltawab</p> | |
| <p>Apr 6 - Apr 12</p> | <p>Everyone should individually complete Team Contract assessment in Canvas</p> <hr/> <p>Continue working on PCB functionality / working out any issues with design</p> <hr/> <p>Add WiFi capability and data collection</p> <hr/> <p>Solder updated round of PCB completely (power subsystem)</p> <hr/> <p>Complete validation and testing on new soldered PCB</p> <hr/> <p>Solder complete PCB (all subsystems) and perform comprehensive testing</p> | <p>Whole group</p> <hr/> <p>Whole group</p> <hr/> <p>Rawda Abdeltawab</p> <hr/> <p>Olivia Guido</p> <hr/> <p>Olivia Guido Rawda Abdeltawab</p> <hr/> <p>Olivia Guido</p> <hr/> <p>Rawda Abdeltawab</p> | <p>Team Contract Assessment (April 10th)</p> |
| <p>Apr 13 - Apr 19</p> | <p>Heatwrap cables to extend and secure sensor cable length</p> <hr/> <p>Continue comprehensive PCB testing (with the whole system). PCB, buttons, sensors, display, battery</p> <hr/> <p>Code cleaned up and comment</p> | <p>Fiona Cashin Olivia Guido</p> <hr/> <p>Olivia Guido Rawda Abdeltawab</p> <hr/> <p>Rawda Abdeltawab</p> | <p>.....</p> |
| <p>Apr 20 - Apr 26</p> | <p>Film demo with UIUC Bee Club</p> <hr/> <p>Solder second Round 4 PCB (solder buttons and sensors directly to this round) to finalize PCB</p> <hr/> <p>Before mock demo practice as a team</p> <hr/> <p>Continue to work on Final Report</p> | <p>Whole group</p> <hr/> <p>Olivia Guido</p> <hr/> <p>Whole group</p> <hr/> <p>Whole group</p> | <p>Mock Demo (April 22nd, during weekly TA meeting)</p> <hr/> <p>Mock Presentation (April 23th-April 24st)</p> |
| <p>Apr 27 - May 3</p> | <p>Continue to work on Final Report</p> <hr/> <p>Prepare for the Final Demo</p> | <p>Whole group</p> <hr/> <p>Whole group</p> | <p>Final Demos (April 27th - 29th)</p> <hr/> |

| | | | |
|------------------|---|-------------------------------------|---|
| | | | Final Presentation (April 30th-May 1st) |
| May 4 - May 7 | Finalize and turn in Final Report ----- Return LCD display to the lab, turn in lab notebooks | Whole group ----- Whole group | Final Report Due (May 6th) ----- Lab Checkout (May 7th) ----- Lab Notebook Due (May 7th) |