Camera Triggering System
‘PhenoCam’

By
Edwin Ortega (eorteg27)
Daniel Salazar (dsalaz8)
Matt Pattermann (mpatte3)

ECE 445 Final Report, Senior Design, Spring 2022
TA: Qingyu Li

4 May 2022

Project No. 53

Abstract

For our project we aimed at creating an automated triggering system for taking camera images via a GoPro. This device works by taking input of the surrounding environment via an ambient light sensor and hot-wire anemometer. The ambient light sensor measures lighting intensity in order to prevent low-light images from being taken. The hot-wire anemometer indicates wind speeds in order to prevent blurry images from being taken. The sensor data is fed into the microcontroller which communicates with the GoPro via a Wi-Fi connection on when the most optimal images can be taken. The device is connected via a wall plug for constant power supply, but capable of running for about 40 hours on active use.

In this report we go over the design, requirements, verifications and results of our project. Aside from some minor challenges, a partially functional prototype was created in accordance with high level requirements.

Special thanks to:

John M. Hart for providing the initial concept and suggesting improvements to our design

Glen Hedin and Gregg Bennett from the ECE Machine Shop for having our idea turn into reality

Table of Contents

1 Introduction

1.1 Problem and Solution Overview

As time goes on, climate change keeps leaving its impact. Notably, it affects the development of plants (such as corn). To combat this the Institute for Genomic Biology at UIUC collects data in regards to the plants; such as, height, color, size and so forth. This is crop phenotyping and it is done manually by hand to be analyzed. The team led by John M. Hart has already developed an image processing tool, Computer Vision, to analyze pictures of these crops. The issue comes with the automation of taking the images. The GoPro cameras being used take an image per hour, take images when crops are moving due to wind, or under dark lighting settings. This results in not well taken images which the Computer Vision tool struggles to analyze. In turn, making analysis for the Agricultural Department difficult since the images taken need to be manually filtered out.

The solution to this would be a camera triggering system, called PhenoCam, that would be mounted on the rain intercept facility (RIF). This solution would be a GoPro enclosed system that will not be damaged by the weather elements. In order to minimize the number of pictures, it will be able to have variable time-lapse control. This is where sensors will come into play and detect wind and sunlight to adjust accordingly. In reference to wind, to reduce blurry images, the system will delay its imaging. Once the sun starts setting and it becomes dark outside, the camera will stop taking images. The procured images are then uploaded via Wi-Fi to then be analyzed by the Institute for Genomic Biology and Center for Digital Agriculture. An image of the final prototyped product is shown below in Figure 1.

Figure 1: Final Product Top View

1.2 High Level Requirements

In order for the device to work as a condition based camera trigger, it needs to adhere to applicable functionality within the rain intercept facility (RIF). Thus the following requirements are to be met:

1. Sensors detect the outside conditions and relay this information to the microcontroller. Hot-wire anemometer displays the wind speed in mph. Ambient light sensor displays the intensity of light in lux.
2. The entire system should be able to run for approximately 48 hours. Due to it being mounted on a 15 foot tall beam within the RIF, there can be inaccessibility to a direct wall plug. Since it is mounted, it also needs to be portable in order to be mounted onto other beams within the RIF to image other crops.
3. A user interface should be able to display the camera settings and allow for overrides to allow for imaging to be taken. Overrides are needed to negate any weather that falls under constant low lighting and wind conditions. In addition, a trigger amount is set to indicate the minimum amount of images desired by the user.

1.3 Subsystem Overview

The block diagram shown in Figure 2 includes the four subsystems utilized in our design. These include the power system, sensor system, control system, and the external system. The external system can be ignored since it includes the GoPro Hero 9 camera and battery provided by the Department of Agriculture.

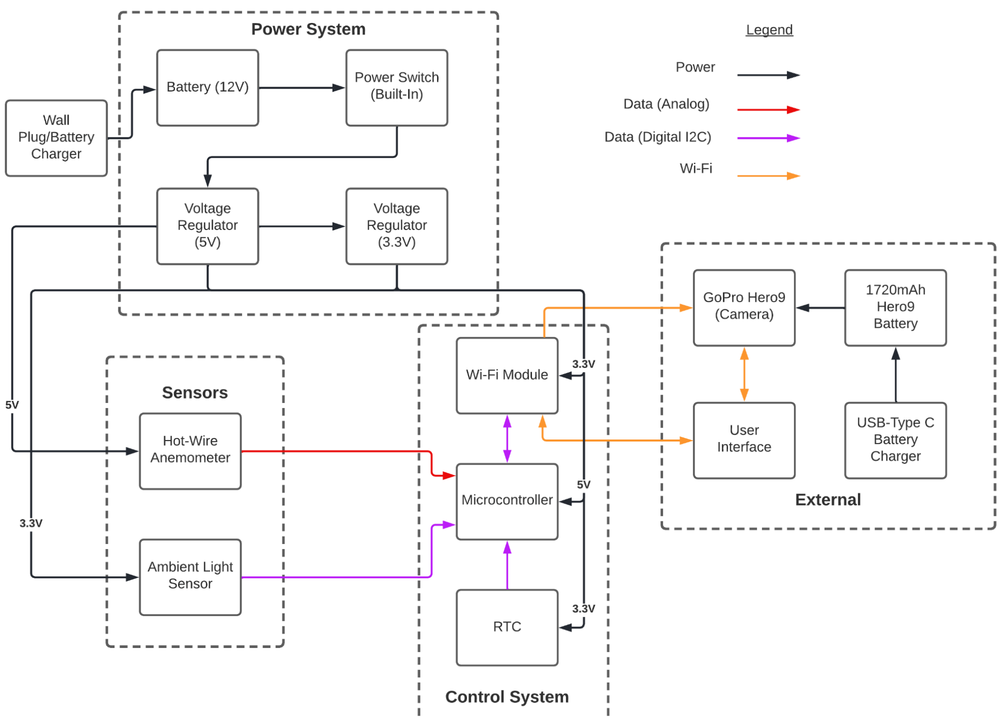

Figure 2: Top Level Block Diagram

1.3.1 Power Subsystem

The power subsystem is responsible for providing power to all other subsystems. It contains a 12V 6000mAh lithium-ion battery module which drops to 5V and 3.3V via voltage regulators. The 5V regulated voltage powers the entire sensor subsystem and the microcontroller within the control subsystem. The 3.3V regulated voltage powers the rest of the control subsystem: real time clock (RTC) and Wi-Fi module.

1.3.2 Sensor Subsystem

The sensor subsystem is responsible for determining the light level, in lumens, and the wind speed, in mph. The ambient light sensor (MAX44009) outputs a digital I^2C signal which is fed into the microcontroller from the control subsystem. The hot-wire anemometer (Wind Sensor Rev.C) outputs an analog signal, which is proportional to the wind speed detected, to the microcontroller. The outputs of the sensor subsystem dictate the optimal imaging conditions.

1.3.3 Control Subsystem

The control subsystem is responsible for polling the data from the sensor subsystem, communicating with the Wi-Fi module, and keeping track of time via the real time clock (RV-1805-C3). The microcontroller (ATtiny85) monitors the time being uploaded via the RTC through digital I^2C. If a trigger is set to occur, the microcontroller reads the sensor subsystem outputs to determine if they meet characteristized base values. If so, the microcontroller communicates via Wi-Fi thanks to the Wi-Fi module (ESP8266) to the user interface of the external subsystem. Note that in final implementation, RTC is excluded since device operating software the device pulls its own time.

1.3.4 External Subsystem

The external subsystem is responsible for the user interface input and actual GoPro Hero9 imaging. The user interface code waits for the control subsystem response from the microcontroller to be able to take an image. The user interface then communicates with the GoPro Hero9 via Wi-Fi to take a single image.

2 Design

The design consists of two parts: hardware and software. Hardware refers to the PCB design, soldering, part research, and connections made between modules. The PCB includes the voltage regulator, microcontroller, and molex connections to the ambient light sensor, Wi-Fi module, RTC, and hot-wire anemometer. The original design included a switching mechanism connected with the power subsystem which was excluded due to the availability of a direct wall plug connection. Software refers to multiple parts of the control subsystem. RTC, anemometer and Wi-Fi module code are adapted from associated GitHub files, this is outlined in detail within Section 2.6. The user interface code is based off of flowcharts in Appendix B and with reference to GoPro API libraries.

2.1 Power Subsystem

2.1.1 Lithium-Ion Rechargeable Battery

In determining the battery to be used, some characteristics were needed to be met as requested by the Department of Agriculture: rechargeability, wall-plug connectivity, run-time capacity (about 2 days worth), and compact. With these goals in mind, the YB1206000-USB TalentCell Lithium-Ion [5] battery was chosen. The battery is a 12V, max 3A output with 6000mAh of capacity. Under standby conditions, the device draws about 10mA, which results in a theoretical 600 hours of standby mode. Under normal operation conditions, the device draws about 160mA, which results in 37.5 hours of operation. The entire current draw never exceeded 210mA which corresponds to 29 hours of minimum expected operation. More details on battery life can be found in Section 3.1 and Appendix A.1. The battery module is also equipped with short circuit protection and low-voltage cut-off protection so it provides protection to the entire circuit and the GoPro Hero9 camera housed in the same case. Charging is done via a DC5521 wall-plug charger and LEDs on the battery module provide visual indicators of battery life. This provides convenience to users at the RIF. The battery module is 145mm by 85mm with a weight of 387g, thus making it quite large. This issue is solved with the provided device encasing from the Department of Agriculture; it is a 1170 Pelican Protector [6] case which boasts an interior size of 267mm by 152mm. This is evident in the physical design images.

2.1.2 Voltage Regulator

In our design, two low drop positive voltage regulators were used: L7805 [1] and LD1117 [2]. The voltage regulators allow for the input battery voltage of 12V to be regulated to appropriate voltage inputs required by the hot-wire anemometer, ambient light sensor, RTC, Wi-Fi module and microcontroller. The voltage regulator system creates outputs of 5V (L7805) and 3.3V (LD1117). The criteria for choosing these regulators depends on the battery input voltage, system capability, and low power consumption. The L7805 regulators have a max input voltage of 40V and 15V for the LD1117 regulators; both of which are more than the 12V battery. In the implementation, the 5V output of the L7805 is fed into the LD1117 instead. The regulators provide a maximum current of 800mA (LD1117) and 1A (L7805), which exceeds the maximum current needed of 100mA for the entire device. The quiescent currents of the regulators are also small, which is ideal for low power consumption. It is 6mA (L7805) and 10mA (LD1117) for the quiescent currents. The last important factor of the regulators is the low noise factor of 0.003% for both. To improve on noise cancellation, decoupling capacitors are added at the input and output of each regulator as seen in Figure 3. LEDs are attached to indicate if power is being sent through as a visual indicator.

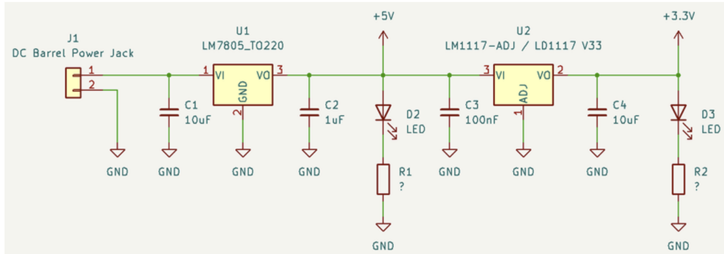

Figure 3: Voltage Regulator Schematic

Design Alternative:
In the original design, surface mounted low drop positive voltage regulators for 5V (RC1616M5T) [3] and 3.3V (AP7361-33E-13) [4] were considered. They each provided a maximum current of 1A, suitable for the expected 100mA current for the entire device. Additionally, the operating temperature maximums were 85℃ (AP7361-33E-13) and 120℃ (RC1616M5T) proving important for operation under the constant sun's heat. However, due to component shortage or long shipment times, the through-hole voltage regulators were purchased instead.

2.2 Sensor Subsystem

2.2.1 Hot-Wire Anemometer
The wind sensor or anemometer chosen was Wind Sensor Rev. C from Modern Devices. The measurement technique used is 'hot-wire', where the power is maintained in order to maintain temperature. Thus this creates a proportional relationship between wind speed and power applied to the sensor. The anemometer takes a 5V input and current between 20 and 40mA depending on the wind speed present. It can detect up to wind speeds of 60mph, which is relevant for wind speeds found in farmlands. The output is an analog signal which ranges from a minimum zero to maximum V_{CC} value. In order to calibrate the sensor, uploaded GitHub files [7] come with the sensor. Additionally, a pin layout is given to describe the importance of the TMP/RV pins which output the temperature and an Ardunio sketch file [23] for translating the TMP/RV pin voltage to temperature. An interior schematic of the sensor is in Appendix B.1 and Figure 4 displays the connections made on the PCB board.

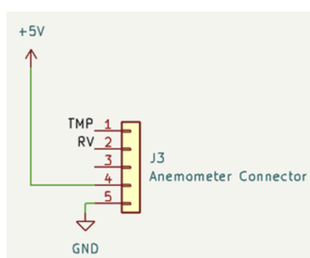

Figure 4: Hot-Wire Anemometer Schematic

Design Alternative:
The original idea for a wind sensor was a typical AcuRite cup [8] methodology where wind is used to physically spin cups. The problem with this method is the significant size of the cups, which would need to be mounted to the case. The other alternative was the Wind Sensor Rev. P [9] which needs a 9V input, draws minimum 40mA and detects up to 150mph for wind speeds. This sensor exceeds any wind speed that is expected to be measured and costs more in comparison to the Wind Sensor Rev. C chosen for the final design.

2.2.2 Ambient Light Sensor

The ambient light sensor chosen is the MAX44009 [10] which is a low-power consumption, digital I^2C output capable, and low-light operation. This component also comes with an 16-bit ADC (analog-to-digital converter) which allows for digital I^2C communication with the microcontroller. The input voltage ranges from 1.7V to 3.6V, which is within the regulated 3.3V from the LD1117 regulator. An operating current of 0.65µA which allows for extended device lifetime. The important feature is the range of lux detection of up to 188,000. Typical sunlight illumination on a regular day usually exceeds 100,000 lux, thus this sensor can distinguish between night and day once characterized. Another beneficial feature is the 800ms maximum integration time which allows for accurate light illumination. The schematic is attached in Section 2.3.1 for the microcontroller, as it is directly connected to it and power.

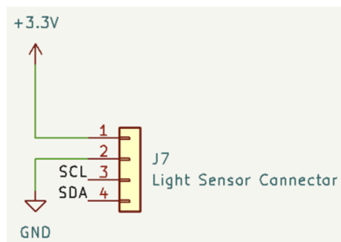

Figure 5: Ambient Light Sensor Schematic

Design Alternative:

The original design included the LTR-329ALS-01 [11] optical sensor which operated on input voltage 3.3V and operating current of 5µA. The lux range only reached 64,000 which is below a regular sunny day. This sensor would be intended for detecting small changes in illumination due to the function of a gain multiplier. Since the purpose of the sensor within the device is to distinguish night from day, this sensor is not used. Additionally, the sensor is 2mm by 2mm, with 0.4mm by 0.4mm soldering contacts making it difficult to solder onto the PCB.

2.3 Control Subsystem

2.3.1 Microcontroller

The microcontroller unit chosen is the ATtiny85 [12] for multiple reasons. The microcontroller needs to have a minimum of four I/O ports for the RTC, ambient light sensor, Wi-Fi module, and hot-wire anemometer. The ATtiny85 contains six general purpose lines, sufficient enough for the entire system. Using provided documentation, the ATtiny85 is configured to use I^2C protocol. Low power consumption to optimize device lifetime. The ATtiny85 draws 300µA under active mode and only 0.2µA under standby mode. Additionally, there are software programmable 'power saving' modes found within documentation associated with the ATtiny85. The operating voltage ranges from 2.7V to 5.5V, which sufficiently falls within the voltage regulated 5V. The operating speed reaches up to 16MHz when operating between 4.5V and 5.5V, making it optimal for device processing speed. Figure 6 shows the schematic connections of an AVR-ISP-6 used to program the ATtiny85 using respective pins. The ATtiny85 contains the I/O from each respective sensor: SDA and SCL for the ambient light sensor, Wi-Fi module and RTC. The anemometer relies on the RV and TMP pin connections outlined in Figure 6.

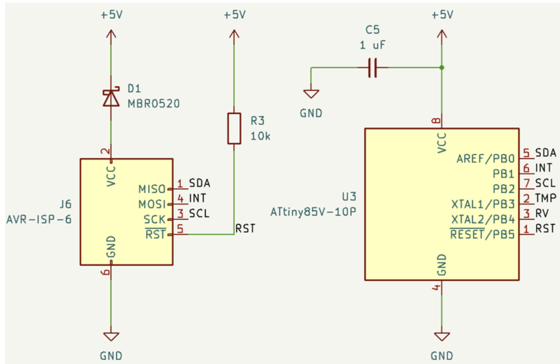

Figure 6: Microcontroller Schematic

Design Alternative:
The original design included the ATtiny441 [13] due to the included I^2C serial interface. As mentioned the ATtiny85 also can be programmed to allow this interaction. The other appeal was the 12 I/O ports for respective sensors. This proved to be much more than the needed four pins. The operating voltage of 4.5V to 5.5V and active mode current of 0.2mA is within the same scope as the ATtiny85.

2.3.2 Real Time Clock (RTC)

The real time clock (RTC) used is the RV-1805-C3 SparkFun Module [14]. The RTC consumes 80nA during active time and 22nA under the $RC_{AutoCal}$ operation mode (lowest frequency mode). The operating voltage is between 1.5V and 3.6V, meaning the 3.3V regulated voltage will work. The device comes with associated documentation libraries [15] (GitHub and Arduino) in order to display the time in USA standard (MM/DD/YYYY). Additionally, the time is displayed in 12-hr format and output via the SDA (serial data) pin on the RV-1805-C3. The RTC is nearly optimal for time tracking with only a 3ppm (three minute deviation after a year of time tracking). Figure 7 shows the connection to the INT (clock interrupt signal), SDA (serial data output), and SCL (serial clock input) for interrupting, capturing, and setting the time on the RTC module.

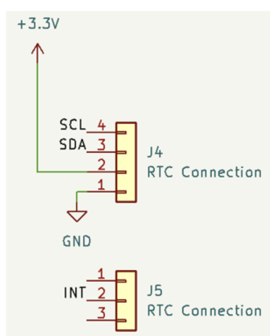

Figure 7: Real Time Clock Schematic

Design Alternative:
In the original design, the intended DS3231 [16] RTC module was considered. The main reason was the supplementary documentation for setting up and displaying time via the DS3231 on Arduino language. Additionally, it functions on the I^2C serial interface as the RV-1805-C3. The main downside is the active-time current consumption of 300µA and 110µA standby which diminishes overall device lifetime.

2.3.3 Wi-Fi Module

The Wi-Fi module used is the ESP8266 SparkFun Module [17]. The purpose of this module is to allow remote connectivity between our microcontroller and user interface which interacts with the GoPro Hero9. The TCP/IP protocol stack will allow Wi-Fi communication to send an 'image trigger' to the user interface code prompting the GoPro Hero9 to take a picture. The ESP8266 operates at a 3.3V input voltage which works with our voltage regulator system. On average active mode operation, the module consumes 80mA of current. The supplemental documentation provides a GitHub wiki [18] for integrating 2.4GHz Wi-Fi with WPA/WPA2 security to prevent any intrusions from outside users. Figure 8 displays the connectivity to the GPIO0 and GPIO2 which are general purpose I/O ports to allow microcontroller Wi-Fi interaction.

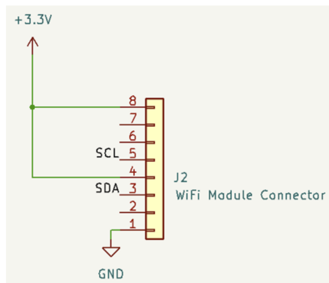

Figure 8: Wi-Fi Module Schematic

Design Alternative:

In the original design, the Department of Agriculture noted that a bluetooth connection would be necessary. Later into the design process, Wi-Fi connectivity was implemented in the RIF meaning the bluetooth module EMB1061 is not necessary. Additionally, the GoPro Hero9 contains API libraries for Wi-Fi integration.

2.4 External Subsystem

2.4.1 User Interface (UI)

The user interface will be the main focus of the software design for this device. It will be responsible for the following: setting the trigger amount, overriding the automatic triggering, and displaying the status of the sensors. The ATtiny85 will process the I/O of the anemometer and ambient light sensor, which will be relayed via the ESP8266 Wi-Fi module to the UI as a displayed time and lux reading. The UI will allow for user set triggering amounts and an interrupt to capture an image despite conditions. More detailed design explanation is in Section 2.6, Software Design.

2.4.2 GoPro Hero9

The camera provided from the Department of Agriculture is the GoPro Hero9. Battery life tends to last around 1hr 40min in operating mode. The implemented API libraries in the UI are in reference to this specific camera. The design would still be applicable with other cameras if they allow for Wi-Fi connectivity and API libraries exist for such interactions. This would require changes to code to satisfy any camera changes.

2.5 PCB Layout

The PCB hardware forms half of the basis for the end product created. Thus, design for the layout is made to be optimal, functional and compact. Other than the standard protocols for PCB layout and design outlined in the course provided documents, there are other considerations that were made.

The DC5221 Lithium-Ion Battery charging port was located in the upper left corner of the PCB in order to be isolated from the connector pins for each component in the control and sensor subsystems. This prevents any user made errors of connections to possibly short a component or exceed the voltage rating of a module. Additionally, the voltage regulator system is located on the top half of the PCB board to allow ease of access to replace any of the through-hole components. LEDs are implemented to prevent reversing the polarity.

The connector ports for each module are located on the lower half of the PCB board due to how the PCB will be mounted within the Pelican Protector case. Since the anemometer and ambient light sensor are mounted on the 'top' of the case, the molex connections will then connect from above the mounted PCB. Thus, the connectors are placed in respect to the physical placement of sensors in the end product. Appendix E contains physical images that represent this connection in the end product.

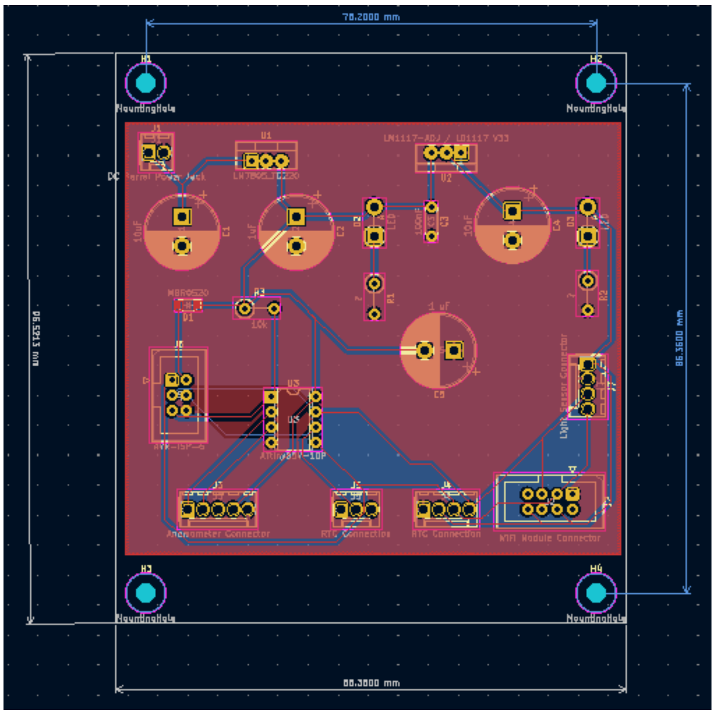

Figure 9: PCB Layout

Design Alternative:

The major change implemented between the original PCB design and finalized PCB design is the type of components that will be soldered. The surface mounted components often were out of stock or on back-order, thus through-hole components were ordered to finish within schedule. Thus the finalized PCB design included this adaptation to through-hole components.

2.6 Software Design

In order to read the sensor data output, relay it via Wi-Fi and then allow wireless GoPro control, software was created/adapted to accomplish the final goal. The general idea and software flow is explained for the ATtiny85 (microcontroller), ESP8266 (Wi-Fi Module) and User Interface (UI) since they are responsible for connecting all components together. To explore the detailed code the GitHub repository [23] contains all code implemented and relevant files.

2.6.1 ATtiny85

All communication between the microcontroller and wind sensor is done via the I^2C protocol. Often this protocol is used for communication between multiple integrated circuits (ICs). In the case of implementation used, the ATtiny85 communicates with the hot-wire anemometer IC via analog communication instead. This is because the Wind Sensor Rev. C communicates via analog signals only. Specifically, the RV and TMP pins are used in communication as outlined in the provided GitHub help repository that is found on the sensor website [7]. I^2C communication is used to request/send this data to the Wi-Fi module under a 'master' and 'slave' communication. This is incorporated using the TinyWire library which allows for I^2C 'slave' address to be associated with the microcontroller. In doing so, the ATtiny85 requests the wind sensor data and sends it to the connected 'master' which represents the ESP8266 Wi-Fi module.

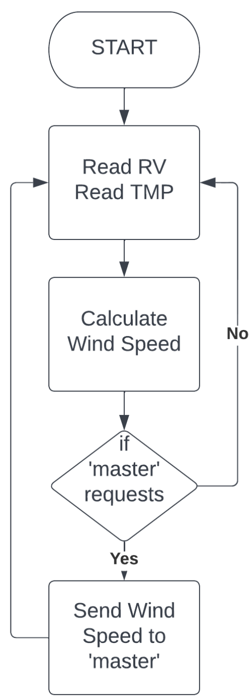

Figure 10: Flowchart of ATtiny85 Software

2.6.2 ESP8266

As mentioned in Section 2.6.1, the ESP8266 Wi-Fi module contains its own microcontroller. The entire module can only be operated as a 'master' thus it needs to request data from the corresponding 'slave' address. As seen in the GitHub [23] repository, the address is defined as 0x40 arbitrarily. It can not interfere with the predefined 0x4A MAX44009 ambient light sensor address or hot-wire anemometer 0x69 address. Initially, the ESP8266 connects to a specific Wi-Fi connection (SSID/Password via W2 security). Adapting existing ESP8266 Arduino IDE code, a request is sent to the 0x4A light sensor address to grab the light intensity (in lux). This is accomplished thanks to the MAX44009 libraries found that help characterize the SDA/SCL pins accordingly. This request is sent after connecting to the web server client established and eventually sends the data to the client. In addition, it requests the 0x40 ATtiny85 'slave' data which contains the already requested wind sensor data. Thus, both sets of sensory data are sent to the established web server on a 500ms delay. Note that the IP address of the web server client is found with Arduino serial monitor to ensure the correct configuration to the client.

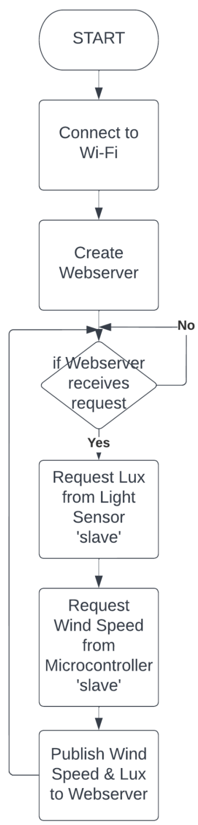

Figure 11: Flowchart of ESP8266 Software

2.6.3 User Interface (UI)

The code compared to the previous components is written in Python. The functionality of the user interface is split into multiple parts. GoPro operation is done via API libraries which allow for remote control of the GoPro via connection to GoPro created Wi-Fi. This allows for display of camera settings, zoom level adjustment, changing the mode (photo or video) and actually taking an image. In order for an image to be taken, the conditions need to be met. Thus thresholds for mph and lux are included which need to be compared to the sensory data transmitted to the web server client by the ESP8266. This is done by reconnecting the device to the web server client Wi-Fi and pinging for the data. This data is then compared to the set thresholds and check if image conditions are met. If so, it transitions to the created GoPro imaging functionality. Additionally, there is the code responsible for making an easy to use interface which is not the Python console. To see the UI, Appendix A.8 contains relevant screenshots.

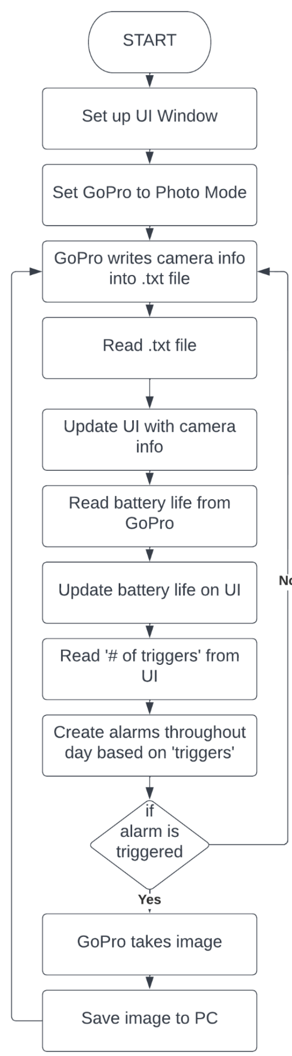

Figure 12: Flowchart of UI Software

3 Design Verification
Overall, our project met all of our high level requirements. The PhenoCam is capable of taking images at set intervals, under correct wind/light conditions, communicating via Wi-Fi, operating the camera wirelessly, display/change GoPro settings, view images captured and override the set wind/light threshold conditions.

In order to make a fully functional project the success of meeting all requirements for our subsystems is accomplished. To understand the depth of requirements and verification results for each subsystem, Appendix A explains all details.

3.1 High Level Requirements
The design verification is done in reference to the outlined requirements in Section 1.2 and is done according to the order outlined there. These requirements are a culmination of subsystems working in unison, thus Appendix A contains proof of verification.

3.1.1 Sensor Functionality
The two main sensors being used in the project are the hot-wire anemometer and ambient light sensor. These sensors are responsible for displaying wind speed in mph and light intensity in lux. Verifying this requirement is straightforward and can be seen within the video submission [22]. Appendix A.3, A.4 and A.7 show screenshot results of web server clients updating the sensory data in real-time.

3.1.2 Battery Life
To determine the expected battery life of powering the system, the following Equation (1) is used. Note that the capacity of the battery is approximately 6000mAh. The current drawn by the entire system during typical operating mode is about 150mA, assuming low-wind. Under high-wind, the operating mode reaches a 170mA current. This leads to a 40hr and 35.3hr battery lifetime. It is not within the desired 48hr, but it is directly wall plug connected. This is under the assumption that the system is constantly polling for sensory data and transmitting via Wi-Fi.

$$Hours = \frac{Battery\ Capacity\ (mAh)}{Current\ Drawn\ (mA)} \quad (1)$$

The battery life can experience diminished lifetime if the ESP8266 operates at 802.11b versus the implemented 802.11n. 802.11n refers to 72 to 600 Mbit/s rates which is sufficient for transferring the sensory data. The ESP8266 was implemented in modem-sleep, reducing the entire system current draw to only 65mA. This leads to an optimized 92.3hr lifetime, surpassing the desired 48hr lifetime.

3.1.3 User Interface
The user interface relates to the interface the user interacts with. The ability to see settings of the camera, battery life, images taken, zoom adjustment, conditions met for imaging and related features are the main focus. This verification is done by displaying the UI within the video submission [22]. Appendix A.8 contains UI screenshots showing relevant features implemented. In order to understand the implementation, Section 2.6 outlines the basic software infrastructure implemented and GitHub repository [23] contains code implemented in final design.

4 Costs

The cost of the entire device comes from team labor, electronic parts, machine shop work, and tools used for measuring. The machine shop work and tools used in the ECE 445 lab room are excluded from the grand total as they are provided. The labor cost is an estimation based on the following factors: hourly rate and number of hours. Based on the U.S. Bureau of Labor Statistics, the average hourly salary of an electrical engineer can be estimated as around $40 [20]. In addition to the hourly salary, a 2.5 multiplier is incorporated with scaling the project into company based work. Each member of the team worked about 7 hours per week on the project, if other courses permitted the workload. The project work spanned about 9 weeks of the course. This information will be used in Section 4.2 for calculating labor cost.

4.1 Parts

Since the project is in collaboration with the Department of Agriculture and John M. Hart, some components of the device were provided or obtained by them. These components will be labeled with an asterisk (*) in Table 1 to denote components not bought with our ECE 445 dedicated funding.

Table 1: Part Costs for 'PhenoCam'

Part	Manufacturer	Retail Cost ($)	Quantity	Actual Cost ($)
12V Battery (YB1206000)	TalentCell	39.99	1	39.99
Case (1170 Protector)*	Pelican	54.95	1	54.95
5V Regulator (L7805)	STMicroelectronics	0.47	20	9.45
3.3V Regulator (LD1117)	STMicroelectronics	0.78	12	9.40
Anemometer (Rev. C)	Modern Device	16.95	1	16.95
Light Sensor (MAX44009)	Noyito	7	2	14.00
Microcontroller (ATtiny85)	Microchip	5	1.49	7.45
Programmer (AVR-ISP-06)	Adafruit	0.95	1	0.95
RTC (RV-1805-C3)	Digi-Key	18.50	1	18.50
Wi-Fi Module (ESP8266)	Mouser	7.50	1	7.50
100nF Capacitor	Venkel	0.11	10	1.10
1µF Capacitor	Venkel	0.10	10	1.00
10µF Capacitor	S.E.M	0.10	10	1.00
10kΩ Resistor	Stackpole	0.10	10	1.00
GoPro Hero9*	GoPro	350	1	350
Total Excluding * Parts				**128.29**
Total of All Parts				**533.24**

4.2 Labor

$$C_{Labor} = People * \frac{\$}{Hr} * \frac{Hr}{Week} * Weeks * Multiplier \quad (2)$$

$$C_{Labor} = 3 * \frac{40\$}{Hr} * \frac{7\,Hr}{Week} * 9\,Weeks * 2.5 = \$18,900 \quad (3)$$

4.3 Grand Total Cost

In calculating the grand total cost, the total cost of all used parts in the end product are included despite some being funded by the Department of Agriculture as mentioned in Section 4.1.

$$C_{Grand} = C_{Parts} + C_{Labor} = \$533.24 + \$18,900 = \$19,433.24 \quad (4)$$

4.4 Schedule

The project is broken down into a 9 week span and the individual progress throughout this semester is shown in Table 2. Some shortcomings were run into around the spring break timeframe with an order of parts for testing the initial PCB being delayed for about two weeks. Week 1 denotes February 28th to March 6th. Week 9 denotes April 25th to May 1st. Note that the schedule excludes any course related assignments.

Table 2: Schedule for Designing ‘PhenoCam’

Week	Edwin	Daniel	Matt
1	PCB Design	PCB Design	PCB Design
2	Refine PCB Design	Power System Research	Sensor System Research
3	UI GitHub Research	Microcontroller Code Research	Sensor Arduino Code
4	Work on UI Code	Assemble PCB Version 1	Work on Sensor Code
5	Refine PCB Version 2	Test PCB	Test PCB
6	Refine UI Code	Refine Microcontroller Code	Refine Sensor Code
7	Test UI	Assemble PCB Version 2	Configure Sensors
8	Refine Entire System	Refine Entire System	Refine Entire System
9	Refine Entire System	Refine Entire System	Refine Entire System

5 Conclusion

5.1 Accomplishments

The entire project works as expected. Sensory data for wind and light is relayed via Wi-Fi to a web server created with any individual Wi-Fi hotspot. Then the user interface code pings for those values on a defined delay timer to indicate triggers and processes that data. If the sensor data is within threshold, an image can be taken. The override condition works, setting displays, adjustment of camera zoom and remote access to taken pictures are also features added. The main purpose described by Mr. Hart for this project was accomplished.

5.2 Uncertainties

There were two notable hiccups that occurred during the final iteration of the project. One of them was a fault in the voltage regulation system. This issue occurred twice in which either the L7805 or LD1117 would act as shorted components. In the first occurrence, it was poor soldering that shorted pins of the LD1117. In the second occurrence, the L7805 had a correct 12V input, but only a 2V output. This was solved by replacing the component. Since then, neither issue has occurred. The other hiccup was the GoPro saving images as .mp4 files. Typically they are saved as .jpg files, instead it would be saved as an audio .mp4 and not be able to be opened. Often this issue would be resolved by re-running the code. It is a repeatable error that occurs randomly and is difficult to pinpoint the cause. In the final version it became nearly impossible to occur after changing the order of sensor data acquisition and imaging code for the GoPro.

5.3 Ethical Considerations

The main ethical consideration is to maintain the safety surrounding the project. As stated in Section 1.1 of IEEE standards, “we must hold safety, health and the welfare of the public as our top priority, as well as strive for ethical and sustainable development practices, and disclose factors that might endanger the public or the environment” [21]. The two safety concerns relevant to the project are the interaction of water with electricity and mounting. Since the device is mounted in an open facility possibly exposed to rain, there is a possibility of shorting out the system. This is due to the power cables for the battery and GoPro sticking out the box. The other concern is the possibility of the entire system falling off of the 15 foot beam. If not mounted properly there is a chance that it can fall and possibly hurt someone below or damage the crops being analyzed.

5.4 Future Work

The prototype of this device can be improved to meet other requested requirements by the Department of Agriculture. As previously mentioned, power switches were excluded in updated PCB design since maintaining power for long amounts of time was not necessary due to a direct wall plug connection available in the RIF. Though, for the ‘PhenoCam’ to be able to operate in isolated facilities, solar power would be the next step. This prevents the need to replace or recharge the battery in the device which is mounted 15 feet up. The other consideration would be the lack of internet connection in the isolated facilities. This would require some other form of communication, such as bluetooth, or a remote way of storing the photos immediately in a thumb-drive or SD card.

References

[1] “Positive voltage regulators - West Florida components.” [Online]. Available: https://www.westfloridacomponents.com/mm5/graphics/ds4/L78-series.pdf. [Accessed: 30-Apr-2022].

[2] “Adjustable and fixed low drop positive voltage regulator.” [Online]. Available: https://www.mouser.com/datasheet/2/389/cd00000544-1795431.pdf. [Accessed: 30-Apr-2022].

[3] “Rocelec.widen.net.” [Online]. Available: https://rocelec.widen.net/view/pdf/bskyyuabmn/RE_DSHEET_N80188_REI.pdf?t.download=true&u=5oefqw. [Accessed: 30-Apr-2022].

[4] "Média.digikey.com." [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Diodes%20PDFs/AP7361.pdf. [Accessed: 30-Apr-2022].

[5] “YB1206000-USB,” *12V Lithium ion battery - Talentcell Technology Co.,Ltd*. [Online]. Available: https://talentcell.com/lithium-ion-battery/12v/yb1206000-usb.html. [Accessed: 29-Apr-2022].

[6] “1170 protector case,” *Pelican*. [Online]. Available: https://www.pelican.com/us/en/product/cases/protector/1170. [Accessed: 29-Apr-2022].

[7] “Wind Sensor Rev. C,” *Modern Device*. [Online]. Available: https://moderndevice.com/products/Wind-Sensor. [Accessed: 29-Apr-2022].

[8] “AcuRite IRIS® (5-in-1) wind cups replacement,” *Welcome to AcuRite.com*. [Online]. Available: https://www.acurite.com/shop-all/weather-instruments/weather-sensors-and-parts/replacement-parts/replacement-5-in-1-sensor-wind-cups-5n1wc.html. [Accessed: 29-Apr-2022].

[9] “Wind Sensor Rev. P - low cost anemometer,” *Modern Device*. [Online]. Available: https://moderndevice.com/products/Wind-Sensor-Rev-P. [Accessed: 29-Apr-2022].

[10] “Industry’s lowest-power ambient light sensor with ADC.” [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX44009.pdf. [Accessed: 30-Apr-2022].

[11] “LTR-329ALS-01 product data sheet optical sensor - lite-on.” [Online]. Available: https://optoelectronics.liteon.com/upload/download/DS86-2014-0006/LTR-329ALS-01_DS_V1.pdf. [Accessed: 30-Apr-2022].

[12] “ATtiny25/45/85 automotive - microchip technology.” [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7598_Automotive-Microcontrollers-ATtiny25-45-85_Datasheet.pdf. [Accessed: 30-Apr-2022].

[13] “Atmel 8495 8-bit AVR microcontrollers - microchip technology.” [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8495-8-bit-AVR-Microcontrollers-ATtiny441-ATtiny841_Datasheet-Summary.pdf. [Accessed: 30-Apr-2022].

[14] “RV-1805-c3 - cdn.sparkfun.com.” [Online]. Available: https://cdn.sparkfun.com/assets/1/8/9/e/2/RV-1805-C3.pdf. [Accessed: 30-Apr-2022].

[15] Sparkfun, “Sparkfun/sparkfun_rv-1805_arduino_library: A SparkFun Arduino library for the extremely low power, very precise, and highly configurable RV-1805-C3 Real Time Clock from Micro Crystal.,” *GitHub*. [Online]. Available: https://github.com/sparkfun/SparkFun_RV-1805_Arduino_Library. [Accessed: 29-Apr-2022].

[16] “DS3231 Extremely accurate I2c integrated crystal.” [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf. [Accessed: 30-Apr-2022].

[17] “Esp8266 WIFI module (WRL-17146) - cdn.sparkfun.com.” [Online]. Available: https://cdn.sparkfun.com/assets/f/e/5/6/f/ESP8266ModuleV2.pdf. [Accessed: 30-Apr-2022].

[18] esp8266, “Home · ESP8266/ESP8266-Wiki Wiki,” *GitHub*. [Online]. Available: https://github.com/esp8266/esp8266-wiki/wiki. [Accessed: 29-Apr-2022].

[19] “ARM MCU abstract SPI EMB1061 Ble Module Hardware Block - Seeed Studio.” [Online]. Available: https://files.seeedstudio.com/products/113990637/res/DS0080EN_EMB1061_V1.3.pdf. [Accessed: 30-Apr-2022].

[20] “Electrical and electronics engineers : Occupational outlook handbook,” *U.S. Bureau of Labor Statistics*, 18-Apr-2022. [Online]. Available: https://www.bls.gov/ooh/architecture-and-engineering/electrical-and-electronics-engineers.htm. [Accessed: 29-Apr-2022].

[21] “IEEE code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 29-Apr-2022].

[22] https://drive.google.com/file/d/1KtXLSBGtJs5Ow0evqDHvU7r5aW3UOHr2/view?usp=sharing

[23] https://github.com/UIUCmpatte3/Camera-Triggering-System-PhenoCam

Appendix A Requirements and Verification Tables

As mentioned in Section 3 for Design Verification, the entire project works as intended. Thus below are the detailed requirement and verification tables for each separate subsystem. In addition it contains relevant gathered results to support the operational aspect.

A.1 Lithium-Ion Rechargeable Battery

Requirement	Verification	Requirement Met?
1. Battery must store 6000mAh, with a ~500mAh tolerance, of charge	1. Verification Process for Item 1 a. Attach 5.5Ω resistor as load b. Measure I and V for defined 3-minute intervals using oscilloscope c. Terminate test when $V_{battery}$ is less than 3.7 V d. Perform Riemann summation of collected data points e. Ensure at least 5500mAh extracted	1. Y
2. Sustain a peak current of 100mA ∓10% during normal use	2. Verification Process for Item 2 a. Attach a 120Ω resistor in series as load with $V_{battery}$ b. Measure I across load using oscilloscope c. Ensure $I_{recorded}$ is at least 90mA or at most 110mA	2. Y
3. Battery must be unable to discharge if left plugged into charger without power; tolerance of ∓15% of rated 12V	3. Verification Process for Item 3 a. Full charge battery using wall plug provided b. Unplug charger from the wall plug c. Allow the battery to sit for minimum 3 days d. Measure $V_{battery}$ using oscilloscope e. Ensure battery has not discharged beyond 15% tolerance	3. Y

Most requirements are battery specifications listed on the datasheet. The peak current is to test the voltage output being ~12V which is the case for this rated battery. The battery discharged about 20% of total capacity after not being used for 2 entire weeks. Thus a degradation rate of ~1.4% battery life per day.

A.2 Voltage Regulator

Requirement	Verification	Requirement Met?
1. 12V ∓ 0.2V input voltage regulator has output voltage of 5V ∓ 0.2V	1. Verification Process for Item 1 a. Attach oscilloscope across the load b. Supply regulated 12V ∓ 0.2V input voltage (DC) c. Ensure output voltage remains betweens 11.8V and 12.2V	1. Y
2. 5V ∓ 0.2V input voltage regulator has output voltage of 3.3V ∓ 0.2V	2. Verification Process for Item 2 a. Attach oscilloscope across the load b. Supply regulated 5V ∓ 0.2V input voltage (DC) c. Ensure output voltage remains between 3.1V and 3.5V	2. Y

The PCB board once soldered is probed in reference to appropriate pins of voltage regulators to determine the regulated voltages. The resulting voltages are well within the ∓ 0.2V of desired regulated voltage. To further understand the limits of voltage regulation, a power supply is used to vary the voltage supply input. It is seen that with an input voltage of around 6V, the voltage regulation diminishes and falls below the ∓ 0.2V interval set. This leads to detrimental voltage inputs for sensors which would lead to dysfunctionality. Since at 0% battery charge, the battery voltage does not fall below this 6V threshold. Thus the battery will always provide ample voltage to negate any insufficient voltage supply.

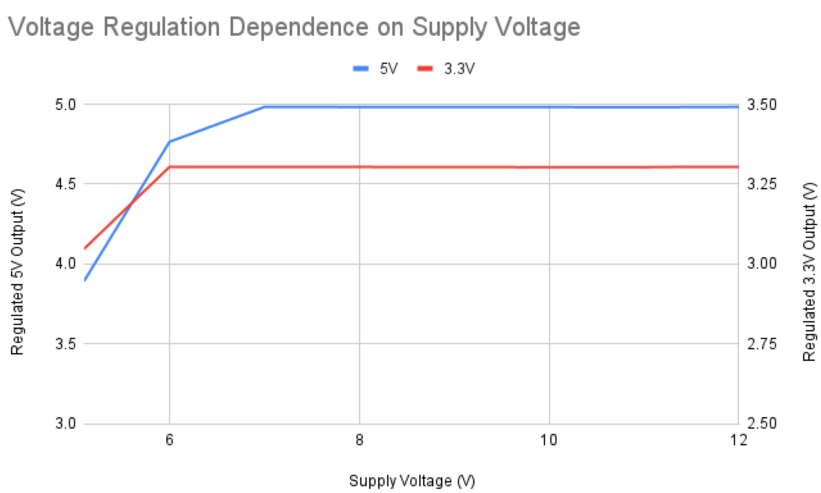

Figure #: Voltage Regulation Limit Exploration

A.3 Hot-Wire Anemometer

Requirement	Verification	Requirement Met?
1. Sensor must be capable of detecting up to (at minimum) 20mph ∓ 10% wind speeds when powered with 5V	1. Verification Process for Item 1 a. Power anemometer with 5V from voltage regulator b. Upload related Arduino IDE code to hot-wire anemometer c. Open serial monitor to display wind speeds in MPH d. Simulate the wind speeds to demonstrate functionality (use a fan to simulate wind)	1. Y

Since the voltage regulation system works as intended and PCB was designed properly, the header pins related to the V^+ and GND for the hot-wire anemometer provide the 5V∓ 0.2V. The related code can be found in the GitHub repository [23]. The resulting wind data is shown below.

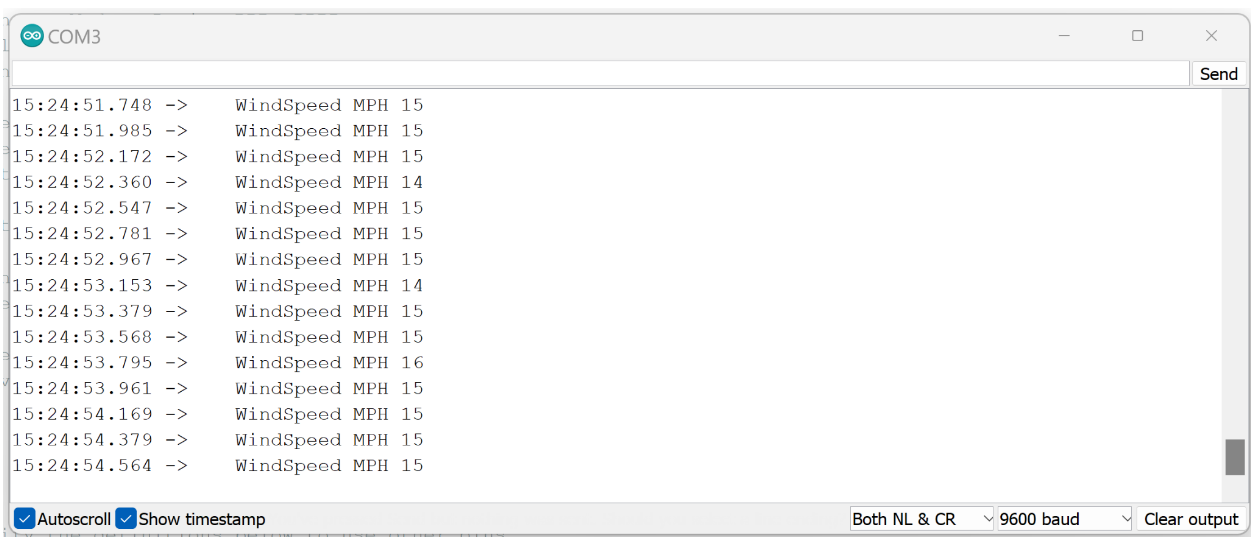

Figure 13: Serial Monitor of Wind Speed Readings

A.4 Ambient Light Sensor

Requirement	Verification	Requirement Met?
1. Sensor must be capable of detecting ambient light intensity in units of lux when powered with 3.3V	1. Verification Process for Item 1 a. Power light sensor with 3.3V from voltage regulator b. Upload related Arduino IDE code to ambient light sensor c. Open serial monitor to display light intensity in lux d. Simulate the light intensity to demonstrate functionality (cover sensor to show no light)	1. Y

Since the voltage regulation system works as intended and PCB was designed properly, the header pins related to the V^+ and GND for the ambient light sensor provide the $3.3V \mp 0.2V$. The related code can be found in the GitHub repository [23]. The resulting light intensity data is shown below.

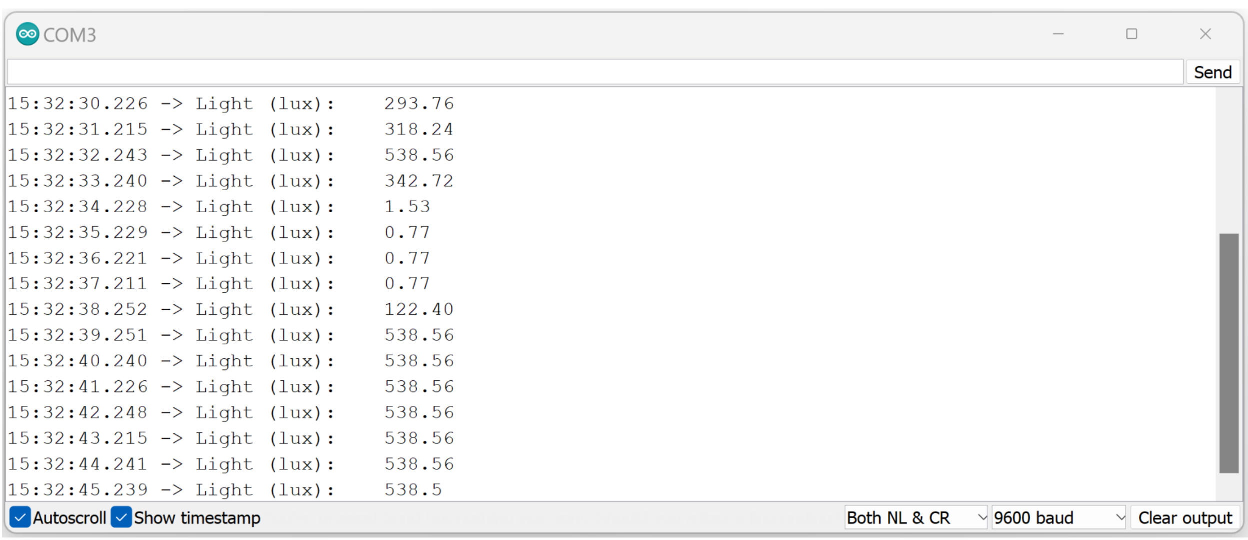

Figure 14: Serial Monitor of Light Intensity (Lux) Readings

A.5 Microcontroller

Requirement	Verification	Requirement Met?
Microcontroller behaves like a ‘slave’ to relay information to the Wi-Fi module when powered with 5V	1. Verification Process for Item 1 a. Upload related Arduino IDE code b. Make sure microcontroller operates as ‘slave’ by requesting information c. Allows to relay wind sensor value	1. Y

Within the final design, the functionality of the microcontroller is represented as the functionality of the ambient light sensor. By requesting the wind sensor data and relaying it to the Wi-Fi module ‘master’ web server client, it represents functionality. Thus the individual functionality of both the ambient light sensor and Wi-Fi module verify the microcontroller’s operation. Appendix A.7 contains the web server client screenshot demonstrating the functionality.

A.6 Real Time Clock (RTC)

Requirement	Verification	Requirement Met?
1. Must be able to display time in 12-hour format (HH:MM:SS) when provided with 3.3V ∓ 0.1V	1. Verification Process for Item 1 a. Upload related Arduino IDE code b. Request serial monitor prompt c. Arduino will probe SDA and SCL pins d. Time should be displayed in monitor and updated every 1000ms	1. Y

In the final design, the RTC is not included. The reason for this is the software running on a user's device allowing for time to be probed there. Despite this, the RTC module works as shown below. This is with relevant code uploaded on GitHub repository [23].

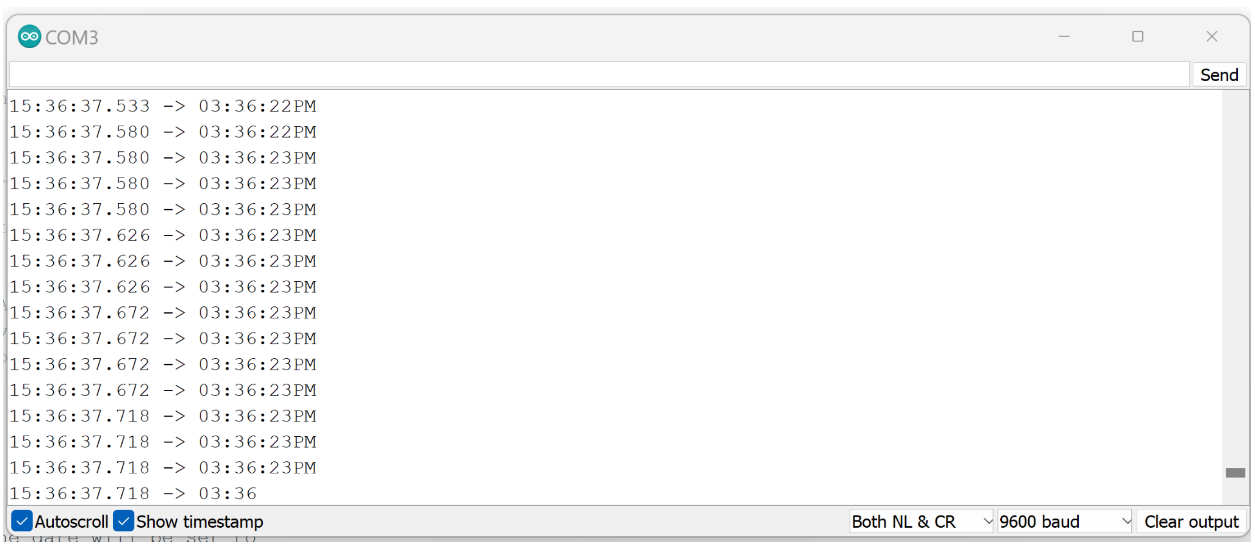

Figure 15: Serial Monitor of RTC Readings (Device Time→ RTC Time)

A.7 Wi-Fi Module

Requirement	Verification	Requirement Met?
1. Current during operation does not exceed 200mA when powered with 3.3V input	1. Verification Process for Item 1 a. Connect the Wi-Fi module with appropriate Arduino UNO setup to upload code b. RESET the Wi-Fi module to change mode of operation c. Probe the V^+ and GND pins to get current readings	1. Y
2. Communicating with ambient light sensor to read value & contact microcontroller for wind sensor value	2. Verification Process for Item 2 a. Upload created Arduino IDE code b. Request the Wi-Fi module to read the value of the ambient light sensor and display it c. Contact the 'slave' microcontroller to relay wind sensor value	2. Y

When the Wi-Fi module has code being uploaded to it, the current draw reaches nearly 248mA. This is not with use of the battery/PCB setup but rather the Arduino UNO. Likewise, when attempting to connect to Wi-Fi it reaches nearly 250mA. Though once the code is uploaded and connection is established, the module operates at around 65mA as outlined in Section 3.1.2. Progressively over longer durations of operation, it reaches 79mA of operational current. Thus this falls below the maximum 200mA of current draw during operation by nearly three-fold.

To demonstrate the functionality of 'master' to 'slave' communication, the web server client screenshot is attached below. This shows the IP address connection and related sensory data that was uploaded by the ESP8266 to the established client.

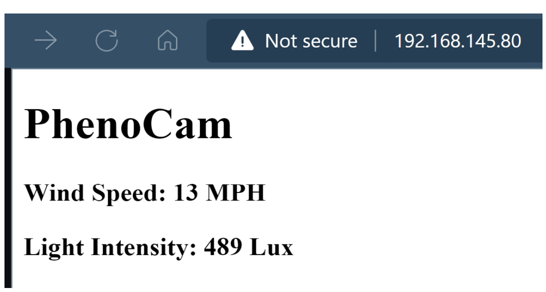

Figure 16: Web Server Client Established by Operational Wi-Fi Module

A.8 User Interface

Requirement	Verification	Requirement Met?
1. Allow for a user to specify the desired amount of camera triggers per day	1. Verification Process for Item 1 a. Input user desired triggers b. Check the images taken c. Confirm set number of images were taken	1. Y
2. Allow for a user to specify the desired camera settings (zoom)	2. Verification Process for Item 2 a. User checks the current operating settings of the GoPro b. Inputs zoom setting c. Checks operating settings again to confirm	2. Y
3. Display the images taken by the GoPro	3. Verification Process for Item 3 a. Access the device's folder to check images b. Access the GoPro remote app to check images	3. Y

To accurately demonstrate this functionality the video link [22] can be viewed to see all working aspects. Screenshots are provided below of the UI to demonstrate the same functionality. To understand the operational code the GitHub repository [23] can be referenced.

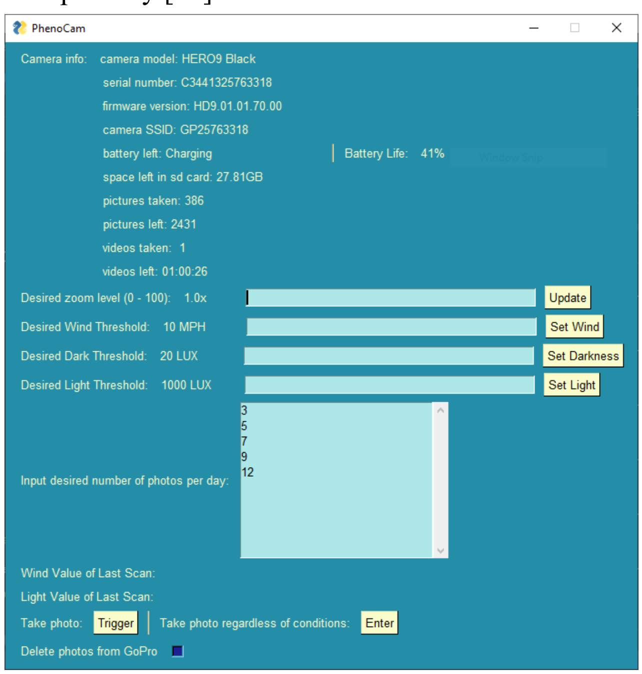

Figure 17: User Interface Showing Functionality Options

Appendix B Schematics

Below are schematics of components used within the project that were used for reference when creating connections within the final PCB design. It additionally includes the entire schematic of the full project.

B.1 Hot-Wire Anemometer

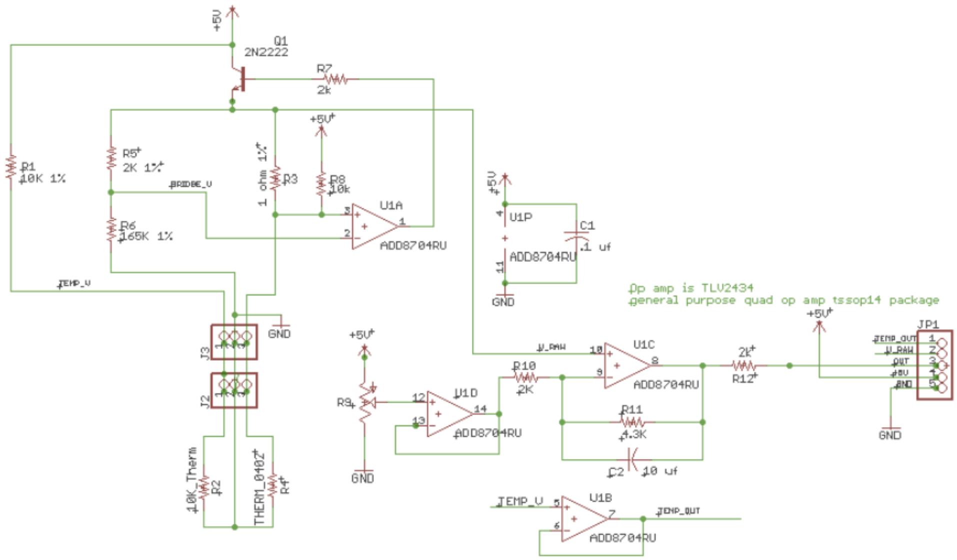

Figure 18: Wind Sensor Rev. C Schematic [7]

B.2 Real Time Clock (RTC)

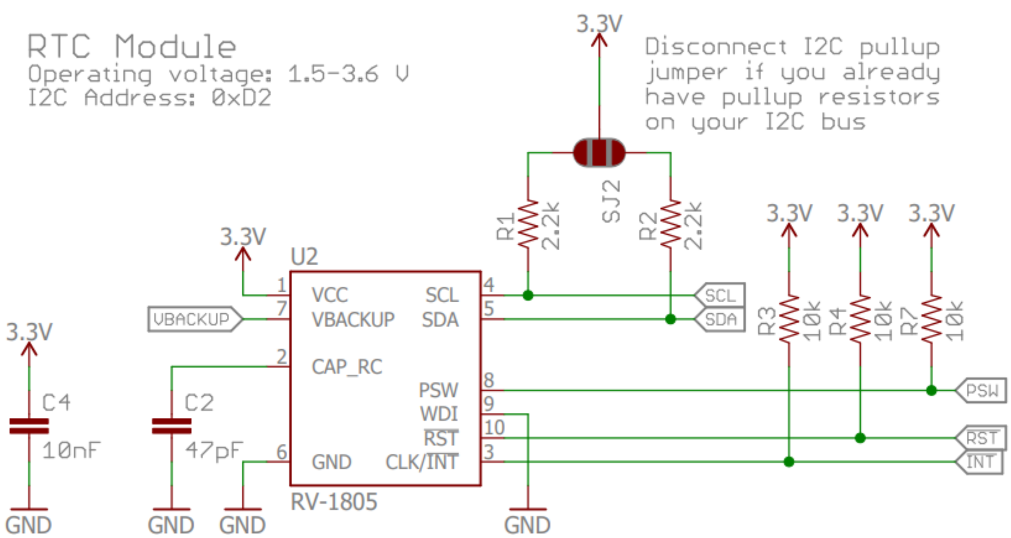

Figure 19: RV-1805-C3 SparkFun RTC Module [14]

B.3 Wi-Fi Module

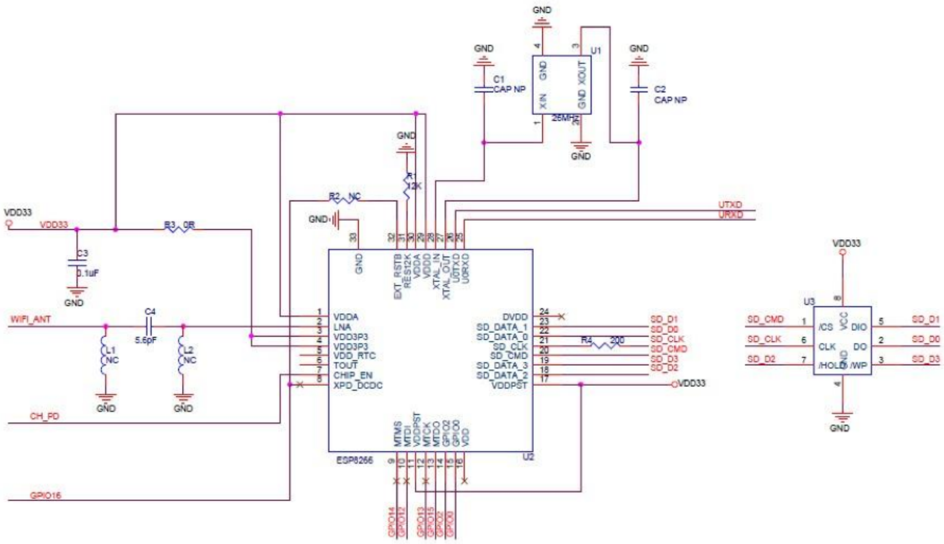

Figure 20: ESP8266 SparkFun Module [17]

B.3 Complete Circuit Schematic

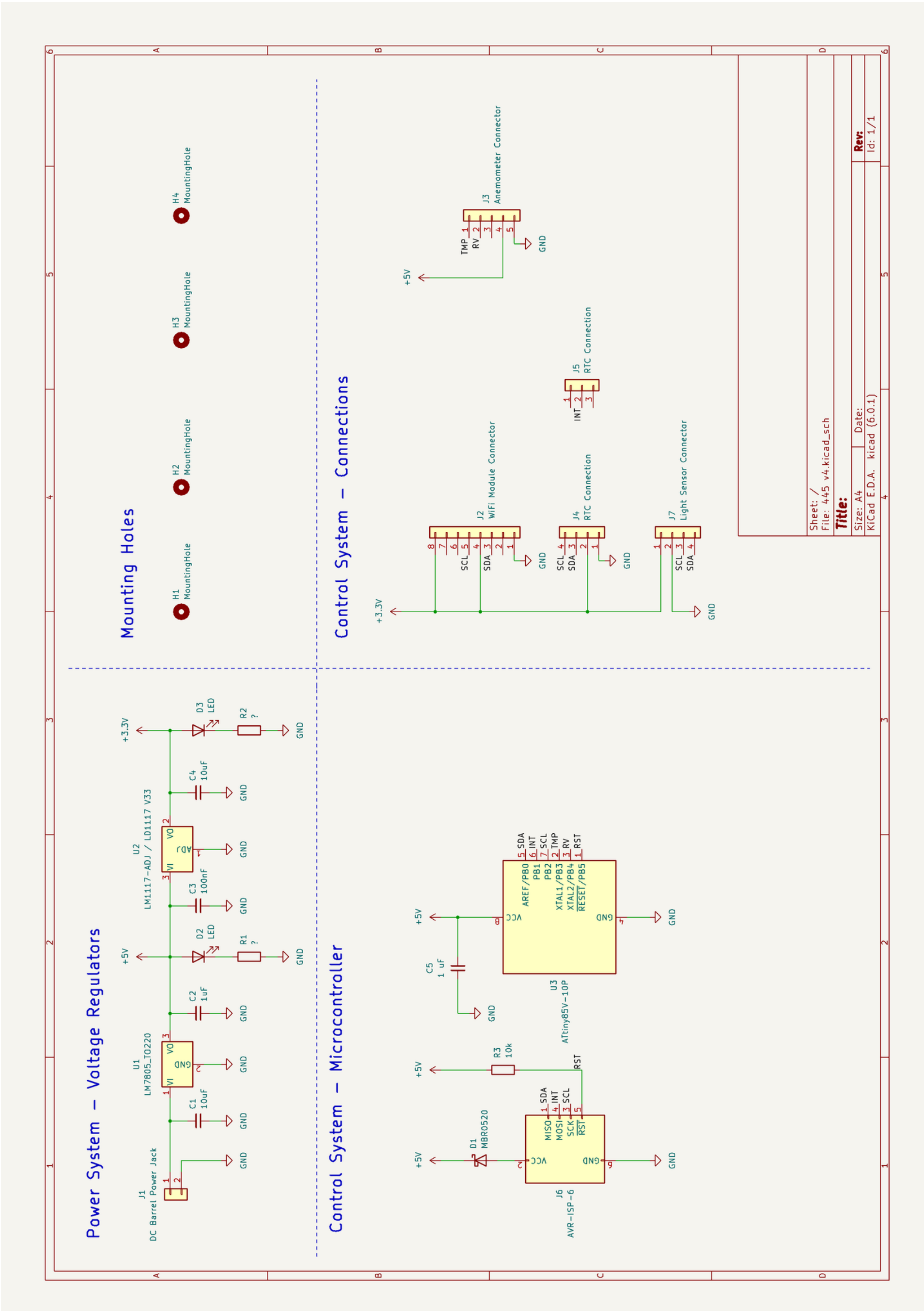

Figure 21: Complete Schematic of Device

Appendix C PCB Board

Below is the physical empty PCB board and PCB board with soldered on components.

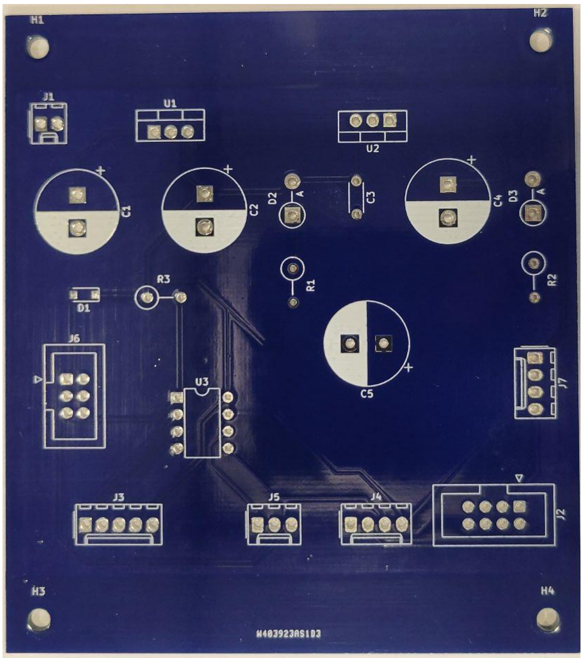

Figure 22: Empty PCB Board

Figure 23: Soldered Complete PCB Board

Appendix D End Product Images

Below are different profile views of the final prototyped product. This allows for a visualization of the product within the Pelican Case with a substitute 2” square beam as the RIF beam.

Figure 24: Top View

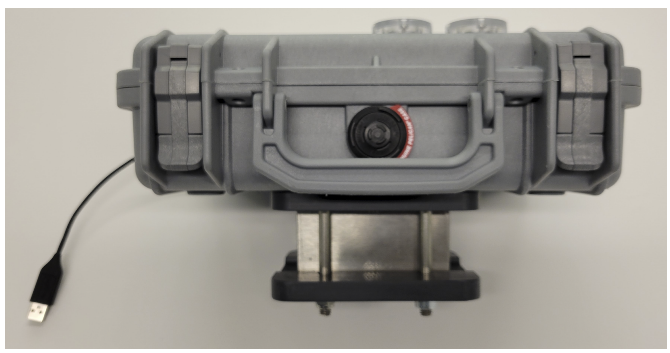

Figure 25: Beam Attachment View

Figure 26: Charging Port Side View

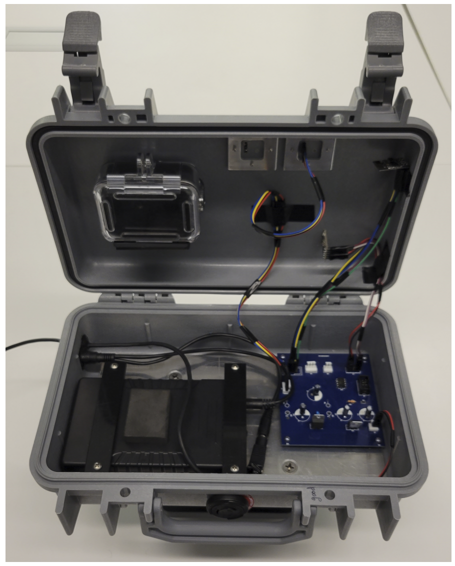
Figure 27: Interior View of PhenoCam

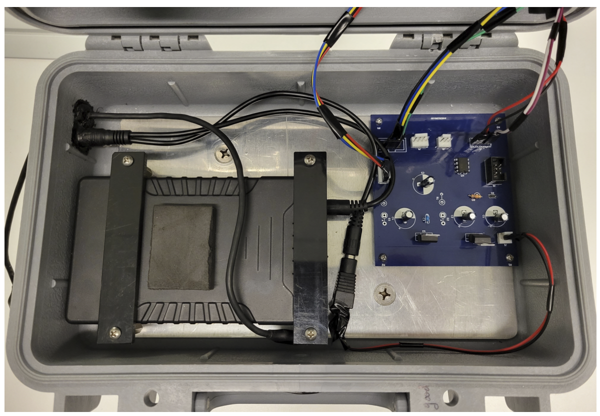
Figure 28: Interior View of Battery/PCB

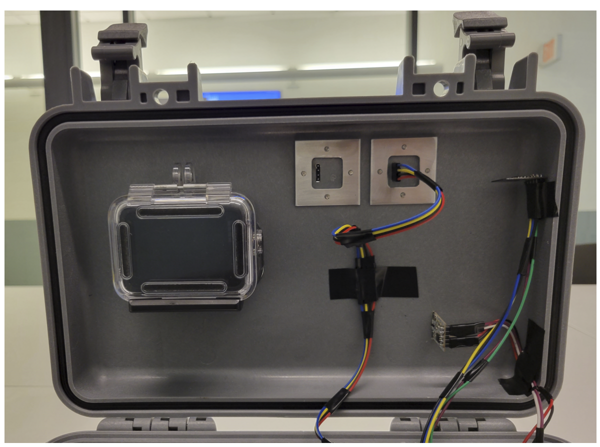
Figure 29: Interior View of GoPro, Light Sensor, Anemometer and Wi-Fi Module