MediDrone: Autonomous Medical Delivery Drone

Varadha Anandakumar, Lior Barak, Ryan Grant, Joey Hodson, Seth Horowitz, Seba Villalobos

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The goal of the autonomous drone system is to reduce delivery latency between hospitals and medical centers. In dire situations, hospitals may be limited in supply of a resource, such as a vaccine or an organ. The time required to deliver this sensitive resource between two medical facilities is critical. MediDrone aims to tackle the unique challenge of preserving medical items during the delivery process. The drone will integrate a double-insulated payload system while synonymously solving the common challenges of designing an efficient drone delivery system.**

***Index Terms* — Autonomous aerial vehicles, Communication protocols, Infrared sensors, PID control, Power distribution**

I. Introduction

MediDrone design aspects include interdisciplinary engineering including but not limited to electrical, mechanical, and aerospace engineering practices. The technical goals of MediDrone are to fly autonomously and deliver a payload containing objects of medical utility. These objects can be temperature sensitive medicines such as vaccines or sublinguals, small medical devices, or plant medicine. The MediDrone utilizes multiple double walled thermal insulation capsules, proprietarily named Chapie, to safely store the payload in physical and thermal safety. Utilizing both electrical and mechanical teams allows MediDrone to be very versatile technically, as the electrical team designed power delivery, control, and communication components where the mechanical team designed the drone's frame, motor mounts, and enclosures.

II. IEEE Standard for Drone Applications Framework

For safety, management, and application purposes, IEEE has set standards and specifications that provide a framework for drone application classes, scenarios, and execution environments. These standards are given in IEEE 1936.1-2021.

In Section 5.2, this standard sets the basic requirements for goods handling, transportation, and processing. In adherence to this, the time, scope, and mode of the operations are set, along with the types of goods and capacity that Medidrone can be delivering. Furthermore in application scenarios, the drone is to be intended to be used for applications relating to disaster relief as listed in Section 6.8.

As the applications are in accordance with the IEEE standard, the design, assembling, and testing of the drone shall stay in accordance with the requirements set forth by the standard.

III. Motives

Mainly, MediDrone aims to accelerate supply transport between hospitals and other medical centers. Without a binding operator to the aerial vehicle, other resources can be freed to support a more efficient medical network. As Floridian students, the team noticed the impact drones had in an instance like Hurricane Ian where individuals were left without power, food, and water. Some individuals required medical support urgently during this time of crisis. Drones have already been integrated in modern society in many different forms of support, but in reality, the future of drones has not even scratched the surface of their full potential.

IV. Testing & Safety Considerations

MediDrone's continual design iterations required many tests, much of this being flight. Aerial objects carrying high-density lithium batteries can be extremely hazardous. It was important for MediDrone to be designed under standard tests that followed all laid out safety regulations. Local regulations influenced MediDrone to be tested at Bill Frederick Park in Orlando, Florida. It was, and is, imperative to ensure the drone is not to fly over fire susceptible areas, namely dry brush and wildlife. It is also important to follow along under, or near, the drone with the necessary means to extinguish a flame if the battery were to explode under impact, preferably using a carbon dioxide fire extinguisher. CO2 extinguishers can ensure that the fire will be smothered and also allow a greater chance that any electronics may be salvaged from the damage. These extinguishers have proved effective at reducing oxygen in fires with corresponding minimal damage to electronic hardware.

V. Mechanical Design

The medical delivery drone was designed and implemented with various features to ensure safe and efficient delivery of medical supplies. This drone is equipped with obstacle avoidance sensors, a GPS system, and a camera to provide real-time monitoring of its location and surroundings.The payload capacity was optimized to ensure that it could carry enough supplies to meet the needs of the healthcare service.

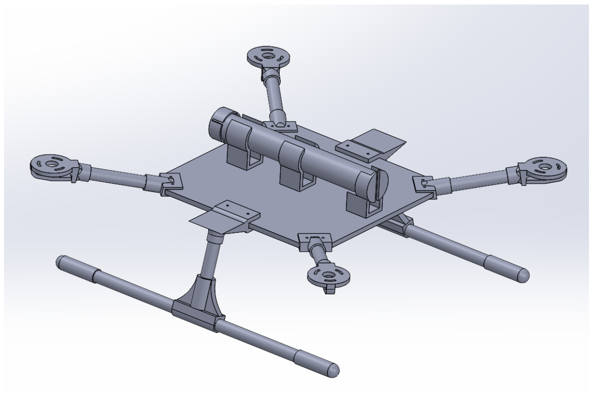


Figure 1: Drone Model

The present model demonstrates the framework of the drone with the added payload system. The team made a strategic decision to engineer the drone from scratch, with the goal of reducing costs and gaining greater autonomy over the drone's modifications.

To achieve the optimal balance between weight and practicality, many components such as the baseplate, battery container, payload container, side supports, and motor mounts have gone through several design iterations. The baseplate currently serves as the central support of the drone. It was optimized by cutting holes in sections that do not need to support equipment. The battery container has evolved from many wood-based designs to a tupperware lid connected to the base by screws. This allows for easy access to the battery, and has resulted in significant weight reduction. The payload container has also turned from a wooden box into a long plastic tube with a lid-attachment mechanism. The tube fits up to two Chapie stick containers of the large variety. The tube solution has also reduced the drone weight significantly, and rests securely at the top of the baseplate. The inside of the container tube can be padded with foam or similar to ensure the security of the payload by restricting movement, particularly if fragile objects are delivered. The side supports have experienced the opposite issue from the aforementioned components, as they were too thin and fragile to survive landings. The current side supports are thicker, and also serve as mounts for the obstacle avoidance boards. There are four of these supports attached to the baseplate, two of which are attached to the landing gear. The motor mounts have gone through several iterations to accommodate changes in motor choices. The original mount was designed for the sky-hero 2815-470Kv motor. Due to not being able to accumulate four of the originally desired motor, an alternative motor was pursued: Tarot TL68P07 6S 380Kv, which required a new motor mount design to fit.

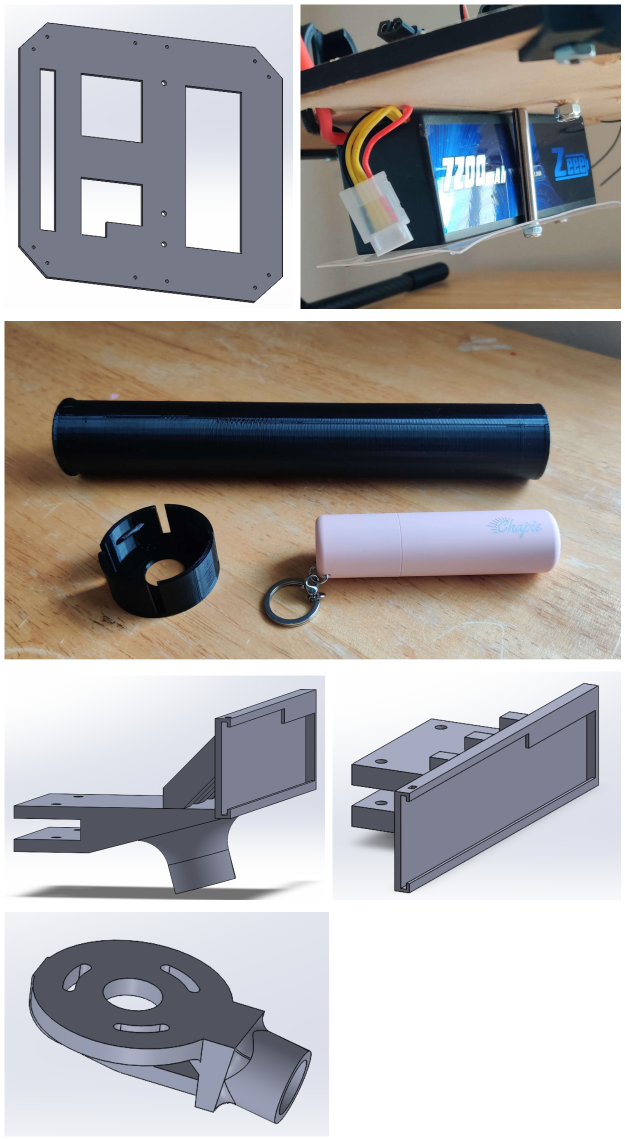


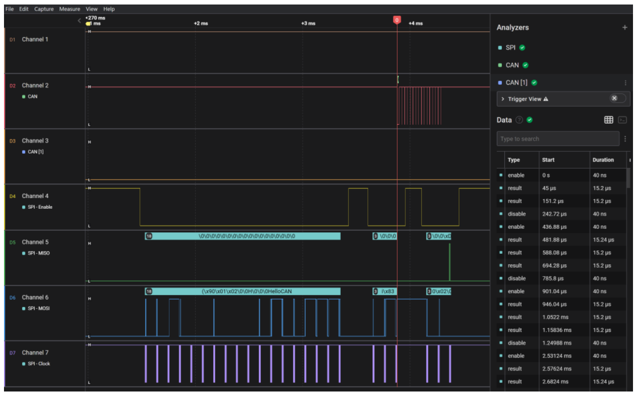
Figure 2: In order of row, then by column. Baseplate, battery container, payload container, side support (with landing gear connection), side support (w/o landing gear connection), motor mount

VI. CAN Design

To deal with an adequate amount of noise within the drone's electrical network, the team decided to utilize the Controller Area Network (CAN) protocol. CAN specializes in reducing communication faults and allows for distribution of computational power on the hardware and software side. This allows for parallelization of tasks where each node performs their specific duties. This serial, differential protocol was originally developed for, and is used frequently in, in-vehicle networks that are prone to high quantities of electrical disturbances. The drone can anticipate a decent amount of noise given the hardware is encompassed by four brushless, DC motors. This protocol utilizes a bus where every node on the network may read or write to. This allows for minimal latency and queued wait times and better direct communication between circuits when necessary between node-to-node communication. Every pcb, on the CAN bus, can therefore communicate with each other when needed. For this to happen, every node must have both a CAN controller, a CAN transceiver, and any other hardware necessary to be able to perform what its necessary task is. The CAN controller moderates the messages coming in and the messages going out. It queues messages in and out for the node to read and write. The controller additionally encodes and decodes messages for the rest of the node's circuit to read and do whatever it may with the data. The transceiver essentially acts as a gateway for incoming and outgoing packets. The transceiver will take in both signals from the bus, and relay only one signal back to the CAN control as both signals should be identical as CAN is based on a differential pair.

In MediDrone's case, each node is equipped with an MCP2517 as the CAN controller and a TJA1042 as the can transceiver. The MCP2517 comes from a family of CAN controllers and has a lot of online resources to integrate both hardware and software with hardware abstraction libraries. Currently, MediDrone has all four obstacle avoidance boards on CAN, the IR beacon receiver node, and the main flight controller board. A main benefit of utilizing a protocol with a bus is that it can be very scalable. If a new feature was to be implemented on the drone, requiring a new pcb, the MCP2517 and TJA1042 can be added to the pcb and then directly inserted in the bus. This allows for more mechanical flexibility too, allowing the creation of smaller, more independent circuits to be scattered throughout the drone with full communication with the rest of the hardware.

Interfacing with the MCP2517 required every node to be equipped with a SPI peripheral. In the obstacle avoidance nodes' case, the circuit centered around an MSP430FR5738, for reasons mentioned in the Obstacle Avoidance section. This microcontroller was equipped with two SPI channels and was therefore able to communicate with the CAN controller. Debugging CAN came with its own issues. With the help from a logic analyzer, the team was able to record stack traces looking at both the SPI message and related CAN packet. The figure below recounts the first message that was seen on the CAN bus while figuring out how to interface with the MCP2517 library used.



18 \0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0

3 \0\0\0

3 \0\0\xD

18 (\x90\x01\x02\0\0H\0\0\0HelloCAN

3 i\x83

3 0\x02\0

Type	Start	Duration
enable	0 s	40 ns
result	45 µs	15.2 µs
result	151.2 µs	15.2 µs
disable	242.72 µs	40 ns
enable	436.88 µs	40 ns
result	481.88 µs	15.24 µs
result	588.08 µs	15.2 µs
result	694.28 µs	15.2 µs
disable	785.8 µs	40 ns
enable	901.04 µs	40 ns
result	946.04 µs	15.2 µs
result	1.0522 ms	15.2 µs
result	1.15836 ms	15.2 µs
disable	1.24988 ms	40 ns
enable	2.53124 ms	40 ns
result	2.57624 ms	15.2 µs
result	2.6824 ms	15.24 µs

Figure 3: O.A. SPI/CAN Capture

VII. Obstacle Avoidance

The obstacle avoidance circuits equipped on MediDrone utilize two different proximity detection technologies to provide safe, reliable flight. Both technologies complement each other as well, both having deficiencies in certain scenarios that the other may be able to work better in. The first sensor is a time-of-flight sensor, specifically the TMF8805. The other is a simpler ultrasonic breakout board, the HC-SR04. The ultrasonic board solders directly into the obstacle avoidance node's pcb while the ToF chip is SMD. The TMF8805 can be communicated with via I2C and has many benefits over ultrasound. Beginning with distance alone, time-of-flight technology is superior over ultrasound as it emits a light rather than a soundwave. Light travels faster than sound while a light sensor has a greater amount of disturbance to filter out than sound. For example, on a very bright or hazy day, light sensors may have more noise present than an ultrasonic sensor. The light in this case is infrared and the TMF8805 can record distances greater than 3 meters in a common scenario. Typical distances versus ambient light, as noise, can be seen from the graph provided by the chip's datasheet below.

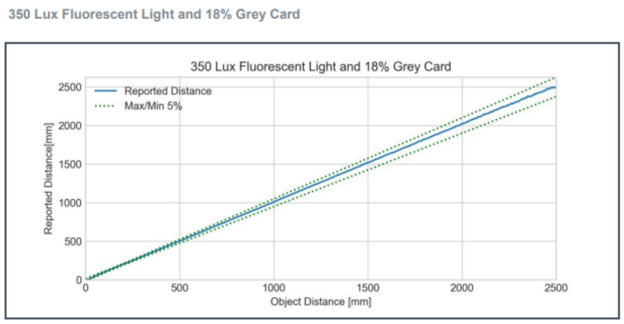


Figure 4: TMF8805 Low Lux Impact on TMF8805

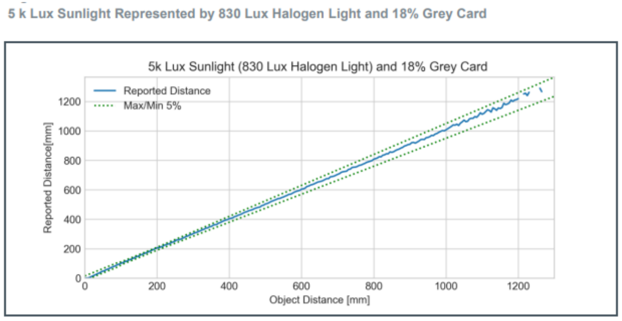


Figure 5: TMF8805 High Lux Impact on TMF8805

The manufacturer concluded through empirical analysis, given the same colored object (18% grey card), a greater presence of light will start to impact captures at roughly 2.5 meters. The sensor proved to be reliable up until that distance. 2.5 meters provides a lot of room for the drone to switch directions during flight given an obstacle being in its path. This data demonstrates ToF technology's largest limitation. This is why MediDrone's hardware is equipped with an additional ultrasonic sensor. The TMF8805 required a ram patch so additional firmware had to be downloaded to the sensor over I2C where the MSP430 acted as a middleman passage. This required the software flashed to contain a rather large quantity of I2C sequences that were converted from Intel Hex Records via a Python script. The next figure is of a snapshot taken from the communication between the MSP430 and the TMF8805 over I2C.

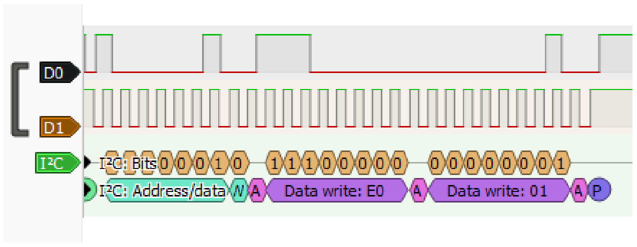


Figure 6: MSP430 I2C Write To TMF8805

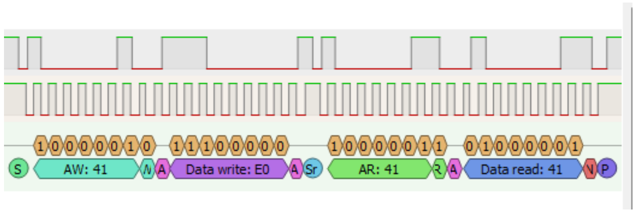


Figure 7: MSP430 I2C Read From TMF8805

The HC-SR04 provides an easy interface to a microcontroller with two digital I/O, a trigger and an echo. To operate, the trigger needs to be pulled high for somewhere in the range of 10 microseconds. The echo's signal is then high for a time that directly correlates to an object's distance. This time can be converted to distance with the formula below.

$$distance = \frac{time\ taken\ x\ speed\ of\ sound}{2}$$

This formula comes from the simple speed = distance / time. Utilizing the speed of sound we then can compute the distance to and from the detected object. The equation then divides by two to find just the distance *to* the object. The MSP430 on the obstacle avoidance board utilizes timer interrupts to capture the echo's rising and falling edges. Below is a capture provided by another logic analyzer debugging session, showing a continuous poll of the trigger going from low to high to low and the recorded input from echo.

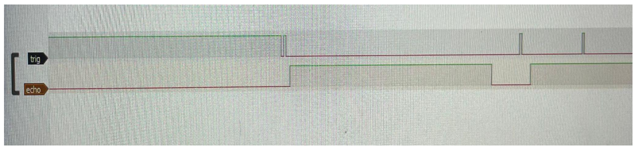


Figure 8: HCSR04 Trigger and Echo Capture

VIII. Autonomous Landing Beacon

In order to automate and optimize the flow of deliveries, the implementation of infrared (IR) beacons positioned at landing points will allow for a reliable method payload delivery and drone return to be guided to a designated and predetermined landing zone.

A. Landing Detection

The optimal method for autonomous landing was narrowed down to two methods. The first method is the use of machine learning. With machine learning, the drone can process a landing zone without the use of additional hardware needing to be placed. The addition of the camera needed to take the imaging would not add much weight to the drone's frame, but would require heavy software implementation. This method also poses issues in reliability with regards to imaging with environmental inconsistencies or obstructions, such as: shifting destinations, fog, smoke, etc. The second method is the use of infrared (IR) signal detection. This would involve the placing at landing points a beacon that transmits IR signals at a specific frequency, which then are detected and filtered through an IR receiver mounted on the drone. By opting to design and assemble an IR beacon for precision landing, it poses a feasible method to automate landing and have consistent reliable return landing for the drone.

B. Beacon Design

The IR beacon comprises two arrays of IR LEDs, totalling 24 bulbs, or emitters. The signal being transmitted from the beacon consists of pulses from the LEDs. In order for the IR beacon's transmissions to be detectable by the IR receiver, the signals will pulsing at a specific frequency so that the IR receiver can filter out unwanted, inconsistent interference from other sources or reflections of IR waves. This is achieved through the implementation of pulse-width-modulation (PWM).

Instead of incorporating a microcontroller to the LED array to attain PWM, a beacon board was designed to add a unique take on already existing IR LED beacon board designs as well as to showcase an original design for a senior project.

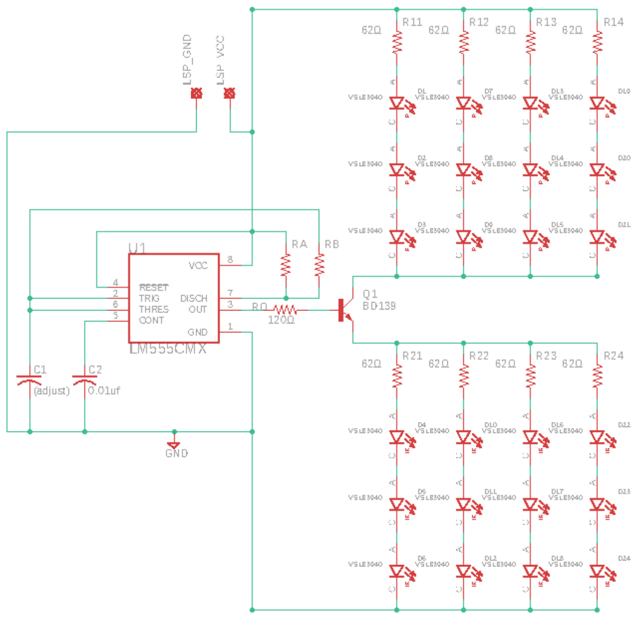


Figure 9: Schematic of IR LED Landing Beacon

This design implements the use of an LM555 timer to attain PWM. The duty cycles and frequency of the timer is adjustable through adjusting the values of the resistors and capacitors following the equations listed in the next section. Connected to the 6V source is the LM555 timer and the first array of LEDs. The current from the source goes down through the array and is joined into the collector of the transistor, which has its base connected to the output of the timer and its emitter going to the second array of LEDs. The resistor RA and RB are connected to both the discharge and the VCC and Trigger pins, respectively. Both values are used to determine high time and RB is used for low time for the timer. The capacitor C1 and C2 act as bypass capacitors to ground. C1 is used to determine frequency.

C. Equations

$$t_h = 0.693\,(R_A + R_B)\,C_1 \qquad eq1$$

$$t_l = 0.693\,(R_B)\,C_1 \qquad eq2$$

$$f = \frac{1.44}{(R_A + 2R_B)\,C} \qquad eq3$$

IX. Power Source & Distribution

Optimizing for weight and range is a critical consideration when designing an aerial vehicle like a drone. Given the current state of battery technologies, MediDrone is equipped with Lithium-Polymer (LiPo) cells. Currently, the drone has utilized 4-series, 14.8v batteries of two different charges, 7.2Ah and 1.5Ah. The different charges are due to the consideration above, optimizing for range and weight. The 7.2Ah battery proved to be potentially too heavy, causing the initial lift of the drone to be difficult against the motors' thrust. The 1.5Ah cells together weighed roughly 6 ounces, a little more than one-quarter of the weight of the larger battery. With this battery, the drone responded much better to initial liftoff and got off the ground with ease, much like a drone should. The lesser weight on the drone also appeared to allow greater flight stability.

Each motor is controlled by a 40-amp ESC. Therefore, the power supply needed to accommodate this. Not only the power supply, but the max-continuous battery current discharge as well. The 1.5Ah battery was rated for 120C as its charging/discharging rate. 1.5 x 120 provided a continuous discharge of about 180 amps. This results in a leftover 20 amps for the rest of the electronics on the drone, this being more than sufficient. The 7.2Ah battery had a similar rating and obviously could discharge the current that the motors sank.

The power distribution board allowed for 4 channels, each with 40 amp traces. This was ideal. It is also worth noting that every power connection in the system is done through XT60 connectors. The 60 implies the max amperage, this being 60 amps. This amount of continuous current is sufficient for every connection. The 5v regulator onboard the PDB could only source 2.5 amps. This was concerning early on as, after some calculations, it appeared the total electronics would sink roughly 3 amps total. After some findings, each of the four obstacle avoidance boards, on average, pulled under 100 milliamps continuously. The flight controller and beacon receiver did not pull near 2 amps combined; thus, the regulator was sufficient. These findings imply that, if more hardware were to be added, like another CAN node, then it may be best to either swap the regulator on the distribution board, design a new distribution board, or to find a more suitable replacement.

X. Software Design

The software on MediDrone can be split into two parts. The high level control code runs on the Raspberry Pi, and the rest of the code runs on various MCUs on the drone. Most of the components are on a PCB shield for the Raspberry Pi, communicating over SPI. The overall software block diagram is shown in Figure W.

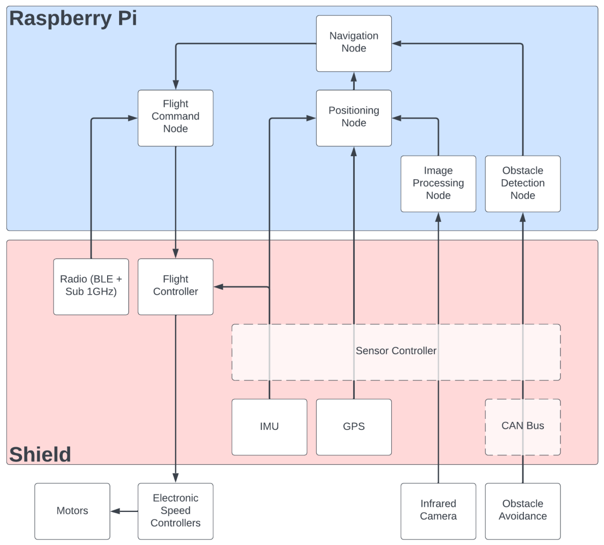


Figure 10: Software block diagram

A. ROS

The high level software that controls the MediDrone's navigation runs on the Raspberry Pi. MediDrone uses the Robot Operating System (ROS) to manage the multiple software components. ROS uses a publisher-subscriber model for communication between nodes, so each node can be viewed as an independent program that communicates via the shared topics.

B. Flight Controller

The flight controller is the component that reads in the direction commands and the IMU readings to provide output to the ESCs. The flight controller is responsible for stabilizing the flight of the drone. At the highest level, this can be viewed as a PID controller, pictured in Figure X. The IMU readings are subtracted from the raw inputs from the remote controller or the Raspberry Pi (depending on flight mode). These are fed into the PID controller, and the outputs are sent to the ESCs. These are mapped to the ESCs according to the equations in Figure V.

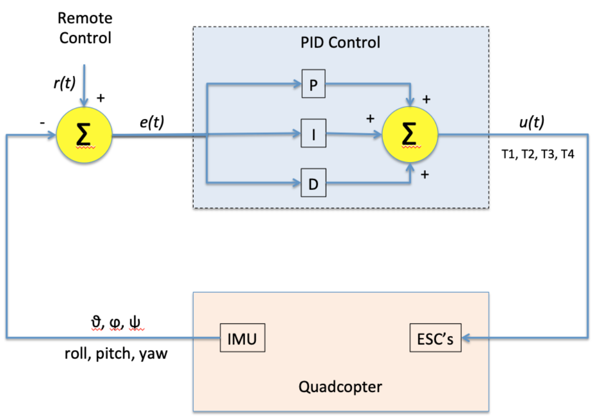
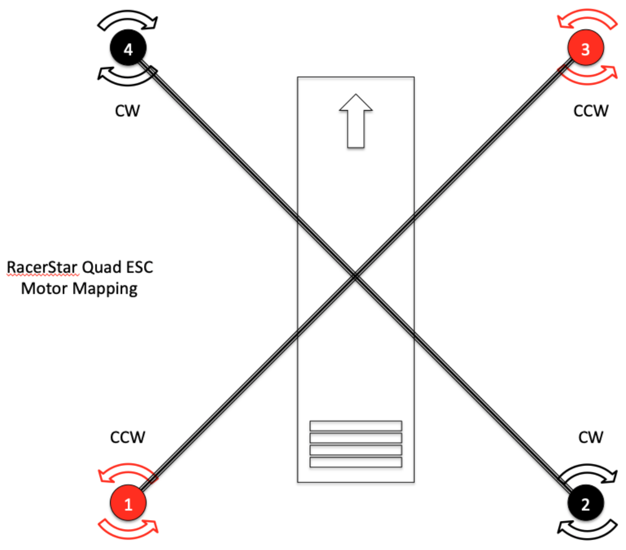


Figure 11: Block diagram for the PID controller that drives the flight controller



Motor 1 (CCW - back left) = Thrust + Yaw – Pitch – Roll

Motor 2 (CW – back right) = Thrust - Yaw – Pitch + Roll

Motor 3 (CCW – front right) = Thrust + Yaw + Pitch + Roll

Motor 4 (CW – front left) = Thrust – Yaw + Pitch - Roll

Figure 12: Mapping of flight controller outputs to ESC powers

C. IR Beacon Detection

A compilation in detecting the IR beacon arises from the fact that sunlight has a large infrared component (Figure Y). The IR component of sunlight can be reflected by roads and sidewalks, so searching for the brightest IR point in the image will not be sufficient to accurately locate the position of the IR beacon in an image.

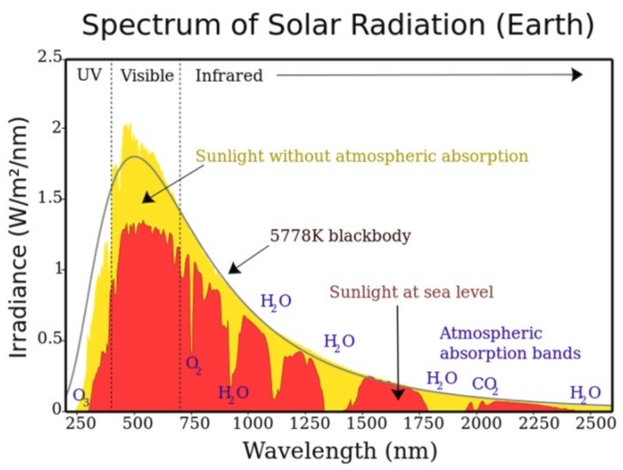


Figure 13: Irradiance vs. wavelength for sunlight

We approach this problem by flashing the IR beacon at a fixed frequency, and searching for a point that is changing at the same frequency. This allows us to filter out points that are always on (reflecting sunlight). Our algorithm is shown below in Figure Z. The idea is to maintain a buffer of images, all taken within one period of the beacon's flashing. This allows us to have an image for both states of the beacon, and search for points that are different in these two images. Points that are different for enough iterations to cross the threshold are then output as candidate points for the beacon.

```
procedure ProcessFrame($f, frames, states, counts, threshhold, T$)
    Enqueue($frames, f$)
    while Time(Front($frames$)) $+\frac{T}{2} <$ Time($f$) do
        Dequeue($frames$)
    end while
    $f' \leftarrow$ Front($frames$)                      ▷ Frame with opposite beacon state
    $positions \leftarrow \emptyset$
    for each position $(x, y)$ do
        if $f[x, y] \neq f'[x, y]$ then
            $s \leftarrow 1$
        else
            $s \leftarrow 0$
        end if
        $counts[x, y] \leftarrow counts[x, y] + s-$ Front($states[x, y]$)
        Dequeue($states[x, y]$)                     ▷ Update buffer of previous states
        Enqueue($states[x, y], s$)
        if $counts[x, y] \geq threshhold$ then
            Insert($positions, (x, y)$)
        end if
    end for
    return Average($positions$)
```

Figure 14: IR beacon detection algorithm

XI. Conclusion

The reader will find that the technical detail of the use case of the MediDrone carrying out medical deliveries meets and goes beyond the threshold for student derived, designed, and prototyped technology in a financially constrained environment. Medidrone's sponsorship from

Chapie, a product of UCF entrepreneurial alumni, plays a major role in the integration of recreational technology for critical and live saving use cases. MediDrone retains character from its interdisciplinary team including electrical, mechanical, and aerospace majoring students as well as in depth software talent. The MediDrone tackles design challenges in each of these fields and strives to expand the current knowledge base on utilizing quadcopters in medical delivery. Building further upon quadcopter technologies, payload configurations, and autonomous flight protocols is a success whether large or small, to develop technologies to be expanded upon in such an open source manner and throughout academia is simply futurism and bringing dreamworthy technology closer to the human realm.

Biography

Varadha Anandakumar is currently a fifth-year senior at the University of Central Florida studying Mechanical engineering. He plans to graduate in May of 2023. He is currently working at Lockheed Martin as a CWEP Intern. Post-graduation Varadha intends to move to Seattle Washington to work as a process engineer.

Lior Barak is a fifth-year senior at the University of Central Florida studying Aerospace Engineering. He plans to graduate in May of 2023, and has started taking graduate level computer science courses at UCF in pursuit of a Master in Computer Science. Lior has accepted a summer internship offer at Statsig, a software company based in Seattle. Following the internship, he will return to UCF to complete his Master in Computer Science.

Ryan Grant is a fifth-year senior at the University of Central Florida. He plans to graduate in May of 2023 with a Bachelor of Science in Electrical Engineering. He is currently researching at Exolith Lab, a space research organization funded by The Center for Lunar & Asteroid Surface Science at UCF. His primary interests are in urban planning and electrical design in architecture.

Joey Hodson is currently a fifth-year undergraduate at the University of Central Florida. He intends to graduate in May of 2023 with a Bachelor's in Electrical Engineering while also receiving a minor in Computer Science. Post-graduation Joey intends to move to Seattle Washington to work as a software engineer. His specialty lies in embedded system development and software development.

Seth Horowitz is a fifth-year senior at the University of Central Florida studying Electrical Engineering. He intends to graduate in May of 2023. Seth has accepted a full time position at Blue Origin as a GNC Hardware Engineer I working with the Electrical Ground Systems Equipment Team located at Cape Canaveral Space Force Station.

Seba Villalobos is currently a fifth-year senior at the University of Central Florida studying Computer Engineering and Mathematics. They plan to graduate in May of 2023. Seba has accepted an offer to pursue a Ph.D. in Computer Science at the University of Wisconsin-Madison. Their interests are in efficient algorithms, architecture for High-Performance Computing, databases, and programming languages.

Acknowledgement

The authors wish to acknowledge the assistance and support from Dr. Samuel Richie, Dr. Lei Wei, Justin Phelps, Kyle Dencker, and Dr. Mike Borowczak.

References

[1] "IEEE Standard for Drone Applications Framework," in IEEE Std 1936.1-2021 , vol., no., pp.1-28, 17 Dec. 2021, doi: 10.1109/IEEESTD.2021.9652498.

[2] "LM555-mil Timer Datasheet - Texas Instruments," Jan-2015.[Online].Available:https://www.ti.com/lit/ds/symlink/lm555-mil.pdf.

[3] "TMF8805 Datasheet - Ams," https://ams.com/documents/20143/4372629/TMF8805_DS000692_5-00.pdf