



MASTERS THESIS

IMPERIAL COLLEGE LONDON

DYSON SCHOOL OF DESIGN ENGINEERING

A Study on the Implementation of an
Autonomous UAV Indoor Sanitisation System

Author:
Colin Laganier

Supervisor:
Prof. Robert N. Shorten

Second Marker:
Prof. Samuel J. Cooper

A thesis presented to Imperial College London in partial fulfilment of the requirements for the degree of
MEng Design Engineering and the Diploma of Imperial College London

June 9, 2022

Abstract

The development of UAV technologies for practical civilian uses is growing rapidly and a variety of new applications are emerging. This thesis studies one of those potential applications: autonomous sanitisation of indoor public spaces. While autonomous indoor drone systems have been investigated in depth in literature, and operator-controlled disinfecting drone systems are available on the market, no research on a proposed combination of both exists. The project attempted to develop a cohesive hardware and software solution, designing a custom micro-drone and electronic platform weighing 397 g and with a focus on scalability. The proposed architecture relies on an UWB positioning system, a RGBD camera environment mapping method, a TCP based server-client communication framework and an onboard control system. Based on the results from the conducted experiments in a lecture hall environment, the developed positioning system proved very accurate and the hardware platform well suited for the proposed application. The open-source design, experimental results and system analysis will serve as a platform for further development of autonomous indoor micro-drone systems.

Keywords: Unmanned Aerial Vehicle (UAV) · Ultra-Wideband Positioning · Visual Mapping

Acknowledgement

I would like to express my gratitude and appreciation for my supervisor Bob for giving me the freedom of explore during the project and whose guidance, support and encouragement has been invaluable throughout this study. A special thanks to David, Tommaso, Roni and Martin for your assistance during the various stages of the project. I also wish to thank the entire design engineering cohort for motivating and pushing me during the final weeks of term, as well as Stack Overflow for always being there for me in desperate times.

Table of Contents

1	Introduction.....	8
1.1	Motivation	8
1.2	About the Project	8
2	Literature Review.....	9
2.1	Generalisation	9
2.2	Existing Research & Implementations	9
2.2.1	Drone Control	9
2.2.2	Indoor Positioning	11
3	Proposed Solution.....	15
4	Hardware Implementation.....	17
4.1	Drone Hardware	17
4.1.1	Overview & Requirements.....	17
4.1.2	Frame	18
4.1.3	Motors & Propellers.....	19
4.1.4	Spray System.....	20
4.2	Electronic Hardware	21
4.2.1	Flight Controller & Electronic Speed Controller	22
4.2.2	Custom Printed Circuit Board	22
4.2.3	Companion Computer	25
4.2.4	UWB Positioning Hardware	25
4.3	Microsoft Kinect Hardware	26
4.4	Hardware Comparison	27
4.4.1	Drone Hardware	27
4.4.2	Positioning Hardware.....	28
4.4.3	Mapping Hardware	28
5	Software Implementation.....	30
5.1	Positioning System	30
5.2	Room Mapping System	30

5.3	Communication Architecture	32
5.3.1	TCP Socket	32
5.3.2	SPI, UART, USB	33
5.3.3	MAVLink	34
5.4	Drone Firmware.....	35
5.5	Drone Client Software	36
5.5.1	Overview	36
5.5.2	Mission Control.....	37
5.6	Drone Server Software	38
6	Results & Discussion	40
6.1	Drone Evaluation.....	40
6.1.1	Drone Hardware	40
6.1.2	Drone UWB Flight	42
6.2	Room Mapping Evaluation	42
6.3	UWB Positioning Evaluation	44
6.3.1	UWB Position "Fast" Results	45
6.3.2	UWB Position "Slow" Results	46
6.4	Limitations	48
7	Conclusion	49
7.1	Summary.....	49
7.2	Future Work.....	49
8	Bibliography	51
9	Appendix	55

List of Figures

1	Diagram of Drone Body Frame and World Frame [1]	9
2	Diagram of a Four Anchor Two Tag Multi-agent UWB Positioning System	12
3	Asymmetric Two Way Ranging Time of Flight Communication Process	13
4	UAV Motion Planning and Positioning System Diagram	15
5	Render of the Assembled Drone Concept	17
6	Diatone Taycan C25MK2 Frame and Custom Ski-style Landing Gear Legs (Solidworks)	18
7	BetaFPV 2004 3000KV DC Motor and Drone Top View	19
8	Render of the Drone Concept's Pump and Water Tank	20
9	Drone 3D Printed Components	21
10	Onboard Electronic Stack-up	21
11	Assembly and Final Assembled PCB (one pound coin for scale)	23
12	PCB Components Diagram.....	23
13	PCB Design Overview (Altium Designer)	24
14	Raspberry Pi Zero 2 W and its Acrylic Support	25
15	Ultra-wideband Positioning System Hardware	26
16	Microsoft Kinect Camera System Hardware	26
17	Positioning Visualisation on DRTLS Management App (screenshot)	30
18	Screenshot of the RTAB-Map Development GUI	31
19	Communication Architecture of Proposed Solution	32
20	Drone Flight State Diagram	34
21	MAVLink Data Over-The-Wire Format	35
22	Drone Flight State Diagram	37
23	Server Input Command Example.....	38
24	Final Assembled Drone (front).....	40
25	Final Assembled Functioning Drone	41
26	Indoor Drone Test Flight	42
27	Occupancy Grid Map of the Lecture Hall and Associated Diagrams	43
28	DWM1001C UWB Positioning System Experimental Setup.....	44

29	UWB Position Results with Fast Pace	45
30	UWB Position Results with Slow Pace	46
31	Testing Experiment to Identify Volume Required.....	55
32	Static Finite Element Analysis of the Drone's Support Leg	56
33	Watertank 3D Printed Manufacturing Iterations	57
34	PCB Schematic (Altium Designer)	60
35	PCB Layers, (left is Top Layer, right is Bottom Layer)	61

List of Tables

1	Comparison of Considered Off-The-Shelf Hardware Solutions	27
2	Comparison of Considered RTLS Alternative	28
3	Comparison of Considered Mapping Solutions	29
4	MAVLink Data Types and Overview	35
5	Cartesian Position Error Metrics for Fast Pace	45
6	Cartesian Position Error Metrics for Slow Pace	46
7	Drone Bill of Materials (weight are estimates or as advertised)	58
8	PCB Bill of Materials	59

1 Introduction

1.1 Motivation

The recent COVID-19 pandemic has shed light on how vulnerable our society is to the wide spread of viruses. A sustained growth of the population and a continued densification of urban areas across the world will result in an increase in the spread of diseases and ultimately of pandemics [2,3]. Additional steps need to be taken in areas facilitating this spread. High-throughput indoor venues, such as cinemas and theatres, as well as offices and lecture halls are particularly high risk areas. One of the main reason for this is the close contact of the users in these venues and the direct contact with shared surfaces - such as tables and chairs - previously used by contaminated users. Antibacterial cleaning of the surfaces and equipment exposed to infectious diseases is the only way of effectively halting the spread of infections.

In the past decades, UAVs have grown from military tools and children's toys to crucial parts of security monitoring systems, industrial inspections, agriculture, payload delivery systems and many more [4]. Research and businesses are finding new ways to put to use UAVs to optimise workflows and automate tasks. One area which has started to grow in the past years is the cleaning services sector, with a growing interest to speed up the process and to tackle larger venues, and could play an important role in reducing transmission risks.

1.2 About the Project

The project aims to investigate the feasibility of an autonomous micro-drone system for disinfecting indoor spaces. Section 2 reviews published research on drone solutions and looks into drone control and indoor positioning methods. Section 3 gives an overview of the proposed solution. Section 4 illustrates the hardware implementation of the system, studying the drone hardware conception, electronic system and camera system, before doing a comparative analysis of the developed solution. Section 5 presents the underlying software implementation detailing the positioning and mapping system, followed by the communication architecture, drone onboard software, firmware and ground station system logic. Section 6 sets forth the experiments conducted and their results, and is followed by a discussion of the strength and limitations of the proposed solution as whole. Finally, section 7 gives a summary of the work, illustrates the potential impact of a drone base system and looks at future work to build upon this implementation.

2 Literature Review

2.1 Generalisation

While the use of drones as a sanitising device has only recently begun, research has been conducted on the effectiveness of such systems in both indoor and outdoor settings [5, 6]. Positive results have been obtained from these experiments as the spraying mechanism from a drone is very similar to the intended spraying patterns of antibacterial solutions. Additionally, indoor spraying removes any potential dynamic issues, such as weather changes and wind, which these experiments have highlighted as the main point of failure.

2.2 Existing Research & Implementations

2.2.1 Drone Control

As the proposed system relies on an autonomous implementation of drones, a rigorous control system is needed to ensure correct operation. This control system can be separated into the physical drone control as well as the communication process used to control the drone.

Quadcopters are particularly well suited for precise navigation within a three dimensional space due to their manoeuvrability and ability to hover. For this reason, applications drones and their dedicated control system have been use extensively studied in research. As these drones have 4 degrees of freedom (see Fig. 1) - thrust, pitch (θ), yaw (ψ), and roll (ϕ) - a multi-variable control system is required for them to navigate effectively.

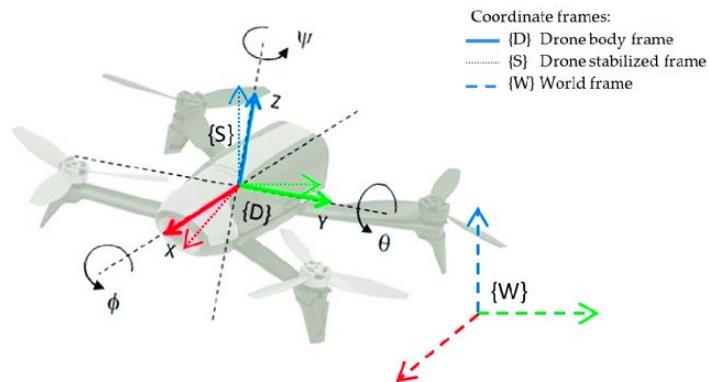


Fig. 1: Diagram of Drone Body Frame and World Frame [1]

The most commonly implemented control method is the PID controller, which computes the error value between a desired set point and a process variable using the current proportional error value, a sum of the error over time and the rate of change of the error. The overall control function can be expressed as:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de}{dt} \quad (1)$$

with K_p being the proportional gain, K_i the integral gain, K_d the derivative gain and $e(t)$ the error term.

Due to the number of degrees of freedom and control variables of the drone, the control architecture is structured as a series of PID controllers arranged in a cascade layout [7, 8]. This closed-loop control feedback system is typically implemented in drone autopilot systems such as Ardupilot, Betaflight and others. These run continuously to modulate user control and ensure the balance of the drone.

Communication is the second crucial component of a drone-based control solution, autonomous or not, as it tells the drone as to where it should be going. In the case of traditional drones, the vehicle receives the user input-controls from a handheld transmitter. Alternatively, autonomous drones or drone swarms typically connect to a central ground control station (GCS), which possesses multiple-input and multiple-output capabilities [9]. Among the communication networks which can be used, the most common are radio waves, ranging from a wide panel of frequencies and modulations. While older generations transceiver typically relied on 27 MHz and 35 MHz bands, interference with other channels proved particularly problematic in terms of reliability for deployment in multi-drone systems [10]. The industry standard for controllers has now shifted to 2.4 GHz band, providing lower interference and higher performance, and giving system developers a range of frequency modulations to tailor their product (FHSS and DSSS are industry standards) [11]. While less common in outdoor consumer devices, WiFi is also a common communication technology used. These systems are more easily accessible to consumers due to the prevalence of WiFi enabled electronics devices. Finally, while uncommon in larger systems requiring higher speeds and range, Bluetooth is commonly used in many toy drones and indoor solutions.

2.2.2 Indoor Positioning

To ensure that a motion plan is being followed and that the UAVs do not crash into the environment or themselves, a robust indoor positioning system is required. Indoor positioning is particularly complex as traditional positioning systems are heavily reliant on Global Navigation Satellite Systems (GNSSs) such as GPS. These are unusable indoors due to the low signal strength and accuracy because of the construction and therefore alternative methods are required to obtain an accurate position estimate. Available solutions can be separated into two categories, internal techniques, relying on onboard sensors and the interaction with the environment, and external techniques, relying on environment infrastructure. Optimal solutions in most cases rely on a combination of both, with a balance between accuracy and associated cost dictating the selection.

The most common technology implemented in indoor positioning research is computer vision based, with over half of published papers relying on it as its a main source of data [12]. This is mainly due to the prevalence of cameras in UAVs and the low cost nature of simple CMOS cameras, coupled with the popularisation of V-SLAM and visual odometry techniques thanks to increased on-board computation [13, 14]. While camera-based methods have shown positive results in the scope of indoor positioning, the associated computational cost and memory requirements of these methods is too great for our low-cost micro-UAV approach.

The second most common primary technology relies on an Inertial Navigation System (INS) approach, based on the data acquired from an Inertial Measurement Unit (IMU) - angular rate, specific force and orientation. From this, the position can be determined by knowing the initial pose, however, INS suffer from drift from the accumulation of position and angular velocity errors. While some very specific implementations have been researched to periodically rely only on IMU sensing [15], sensor fusion from a second source is most commonly used to correct the drift and to leverage the acquired data for more accurate localisation [16, 17].

The third most common technology used is radio frequency (RF) implementations for indoor positioning, and among those Ultra-wideband (UWB) and Wireless Local Area Network (WLAN) are two relevant technologies due to their support for multi-agent systems and their relative ease of imple-

mentation.

UWB technology is very well suited for short-range high-bandwidth communication of up to 60m in line of sight and therefore indoor positioning systems. UWB is defined as a signal that has a 10 dB bandwidth greater than 500 MHz or a fractional bandwidth greater than 20% [18]. The positioning setup relies on a set of at least three fixed anchors (transceiver), spread apart in the target environment at known positions and set height (see Fig.2). Using these anchor nodes, the position of a mobile tag can be calculated using one of multiple methods, amongst the most common are Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Two Way Ranging (TWR) [19].

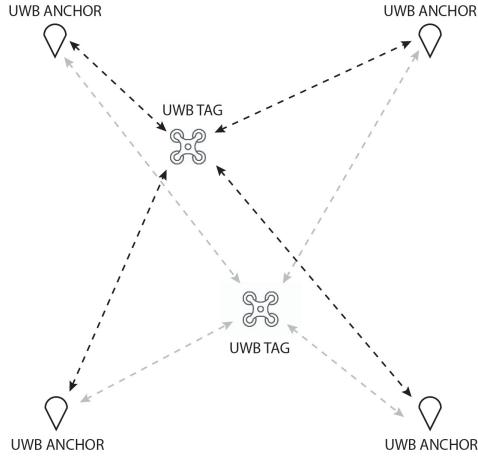


Fig. 2: Diagram of a Four Anchor Two Tag Multi-agent UWB Positioning System

ToA determines the position based on the propagation time of the wireless signal from the emitting anchors to the mobile tag, determining the distance between each and computing the coordinates using a series of geometric algorithms [20]. Alternatively, TDoA calculates the position by using the transmission time delay difference between the mobile tag and static anchors. While TDoA typically results in more accurate measurements [21], the requirement of synchronised anchors and a physical connection between them adds additional complexity and cost to the system. Finally, in TWR the radio transmissions between an anchor and tag are timed and mathematically combined to determine the distance between the devices. This method involves three transmissions between the devices and thus solves the issue of oscillator synchronisation while mitigating the error due to clock and frequency drift for accurate measurements.

The diagram¹ below shows the transmission process between the two nodes:

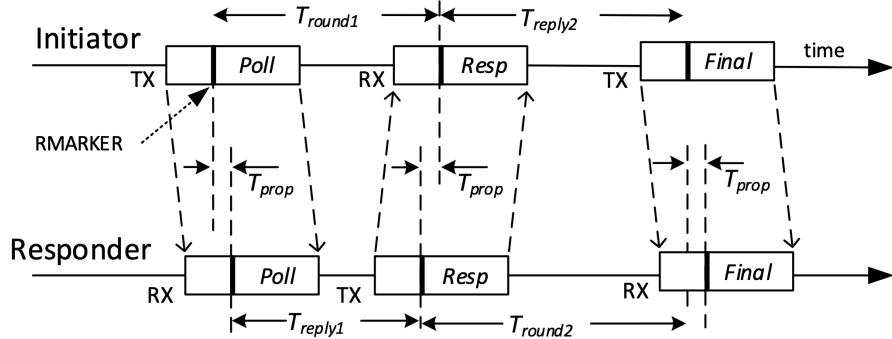


Fig. 3: Asymmetric Two Way Ranging Time of Flight Communication Process

The round-trip time of asymmetric TWR (see Fig. 3) can be formulated as [22]:

$$T_{round1} = 2T_{prop} + T_{reply1} \quad (2a)$$

$$T_{round2} = 2T_{prop} + T_{reply2} \quad (2b)$$

where T_{round1} and T_{round2} are the true round-trip times of a signal measured at Device 1 and 2 respectively. T_{reply1} and T_{reply2} are the true reply times measured at Device 1 and 2, respectively.

By multiplying Equations 2a and 2b, we obtain:

$$T_{round1} \cdot T_{round2} = (2T_{prop} + T_{reply1}) \cdot (2T_{prop} + T_{reply2}) \quad (3)$$

By simplifying the equation, the T_{prop} value is obtained as follows:

$$T_{prop} = \frac{T_{round1} \cdot T_{round2} - T_{reply1} \cdot T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} \quad (4)$$

¹Decawave TWR: <https://www.decawave.com/>

From this time delay between messages, the distance can be computed by assuming the propagation speed of the radio waves through air is the same as the speed of light. Depending on the implementation and filtering process, the expected mean error can typically be as low as 100mm to 200mm [12,23], thus creating an accurate indoor positioning system.

Due to the prevalence of WLAN networks in public spaces and mesh networks of access points, it has been used in research as a Signal of Opportunity (SoOP) with varying levels of success at a very low cost. Implementations vary, while a Received Signal Strength Indices (RSSI) fingerprinting method offers relatively high accuracy in order of 1m, the requirement for RSSI measured data point mapping of the environment makes this implementation poorly scalable [24,25]. Alternatively, a multilateration method can determine the location of the agent based on the RSSI and the position of the access points. This simplifies the implementation at the cost of lower accuracy, resulting in a mean accuracy of the system of ranging between 2 and 4 m [26,27].

3 Proposed Solution

To this day no end-to-end autonomous implementation has been developed for the purpose of sanitising either indoor or outdoor. Existing commercial solutions (such as EagleHawk² and Lucid Drone Technologies³), require UAVs to be tele-operated with a constant direct line of sight with its operator [28, 29]. This in itself is an imperfect system as it does not scale well due to the time-requirement of trained operators, and does not guarantee that the work is performed as a systematic procedure due to the human control and approximation of the spraying patterns.

The proposed solution is a system of micro-drones, carrying a spraying system and tank of antibacterial solution able to position themselves and identify surfaces that need that need cleaning through online identification and offline room mapping. In effect, the proposed system is threefold, the design of a bespoke drone for the intended use case, the development of a room mapping system and finally the creation of a control and positioning system.

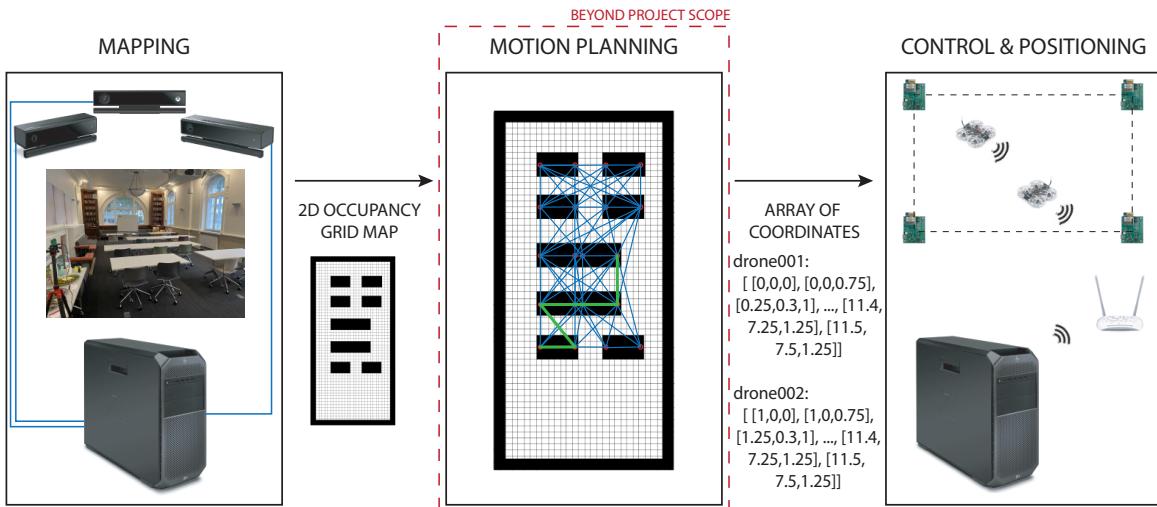


Fig. 4: UAV Motion Planning and Positioning System Diagram

First, based on the design and component of both racing *micro* drones (between 500 g and 2 kg) and that of agriculture and sanitising industrial *medium* drones (up to 150 kg⁴), a custom hardware platform is developed and manufactured. The focus of the produced hardware is scalability and therefore

²EagleHawk: <https://www.eaglehawk.io/drone-enabled-disinfectant-spraying>

³Lucid Drone Technologies: <https://www.luciddronetech.com/C1Info>

⁴Director General of Civil Aviation categories: <https://digitalsky.dgca.gov.in/faq>

with a focus on cost reduction and overall number of components.

Second, using an array of three depth sensing cameras connected to a central computer acting as a ground station, the furniture of the target room are identified and a two-dimensional occupancy grid map is produced. In this initial develop stage, a lecture hall is used a sample environment and the desks are the target furniture to be identified and sanitised.

Finally, relying on IMU, barometer, compass and UWB data, a real-time positioning system is created for the drone to navigate within the space. In parallel, an onboard control system is developed to communicate with the ground station computer, to control and handle the flight mode with live telemetry data and to interface the custom sensor electronic package with the flight controller.

4 Hardware Implementation

4.1 Drone Hardware



Fig. 5: Render of the Assembled Drone Concept

4.1.1 Overview & Requirements

The most basic quadcopter drone can be simplified into 6 different components: the frame, flight controller, electronic speed controllers, motors, propellers and battery. From these essential building blocks are a multitude of customisable components to tailor the function and use of the drone. For this specific application, the drone needed to be small for indoor use, safe for potential nearby users and be able to carry both a liquid payload and a spraying mechanism. For this reason the following constraints were established:

- Ability to hold 100 mL of antibacterial solution: this is based on early testing using a spray bottle (see Fig. 31 appendix) and represents one cleaning pass of two drones in a 24 desk lecture hall.
- Total footprint smaller than 200x200x100 mm: this is based on current market offerings from indoor solutions and the estimated smallest possible footprint to house all required equipment.

- Ability to fly for at least 5 minutes: this is based on sufficient time it is expected to take for a 30 m² room and based on expected power consumption of similar hardware carrying a similar payload.

For this reason, it was decided to create a fully custom micro-drone, ensuring each constraint could be satisfied from the design conception stage.

4.1.2 Frame

The frame was the starting point of the component selection as it would dictate the dimensions of the payload which could be carried. To satisfy the small footprint constraint and light weight, it was decided to choose a drone with 2.5 inch propellers. To ensure the drone would be safe for indoor operation, a quadcopter with propeller guards was chosen. The Diatone Taycan C25MK2⁵ frame kit satisfied the constraints and was therefore selected. The frame had a wheelbase of 110 mm and an overall footprint of 160x150 mm. Its ABS and carbon fibre construction weighs a total of 55 g and its design could easily accommodate for the addition of new components. While the frame could have been 3D printed and designed, the injection moulding construction would ensure the frame would not be a point of failure of the drone.



Fig. 6: Diatone Taycan C25MK2 Frame and Custom Ski-style Landing Gear Legs (Solidworks)

Additional landing gear support were designed, 3D printed and fitted to the frame for added height of the final construction. These were designed to be very lightweight with a PLA construction and weighed only 7 g. To ensure the design would sustain the forces expected when falling from heights, a finite element analysis study was conducted (see in Appendix Fig.32). While initial designs would

⁵Diatone Taycan C25MK2: <https://www.diatone.us/products/diatone-taycan-c25-mk2-cinewhoop-framekit>

break from falls greater than 2 meters high, the addition of a crossbar added rigidity and allowed the support legs to sustain falls from up to 7 meters at full weight (battery and filled solution tank).

4.1.3 Motors & Propellers

As dictated by the chosen frame, the propellers chosen for the solution were 2.5 inches in diameter. While larger propellers have greater surface area, hence producing greater lift while spinning at a lower rate and at a greater efficiency, the small footprint of the overall drone was deemed more important. In an attempt to negate this, 6-bladed propellers were chosen, therefore increasing the overall surface area. Pre-existing research showed promising results with higher blade counts in drones [30], and found that more thrust would be produced while creating less vibrations and air disturbances, both relevant in indoor spraying.

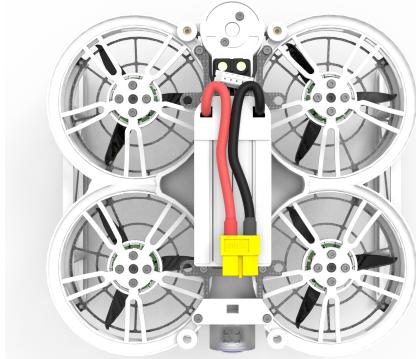


Fig. 7: BetaFPV 2004 3000KV DC Motor and Drone Top View

Based on the weight estimations of the assembled drone which was approximated at 490.3 g when filled up with the sanitising solution (see in Appendix Table.7), the thrust requirements were identified. For UAVs to be agile and nimble enough for flight, a 2:1 thrust to weight ratio is typically used, therefore allowing the drone to hover at half throttle. With this value in mind, many motors were studied and ultimately the BetaFPV 2004 3000KV brushless DC motors were chosen. Based on datasheet information and thrust comparisons with similar sized propellers, it was estimated that each motor would produce 350 g of thrust, for a total thrust of 1.4 kg. This safety factor of 1.43 would ensure the drone would fly and be nimble even if the weight was underestimated and if any advertised value was inaccurate.

4.1.4 Spray System

To effectively spray antibacterial solution on surfaces, a spraying solution was developed. A micro diaphragm water pump was coupled with a low-pressure atomising spray nozzle to create a mist of solution. The pump was operated using 9 Volts, and at maximum speed, a volumetric flow rate of 190 mL/min was measured. A pump holder was designed to be bolted to the frame using threaded inserts, and doubled as a holder for the XT60 connector connecting the battery to the flight controller.



Fig. 8: Render of the Drone Concept’s Pump and Water Tank

To provide the spraying pump with a constant flow of solution, a watertank was designed to be mounted on the underside of the drone frame. The tank was designed as to not impede the flow of air from the propellers in flight, hold 115 mL of solution without overflowing while being as light as possible, weighing 37.5 g. The tank construction is made of 3D printed transparent Nylon, therefore allowing for visual inspection of the water level, as well as benefiting from the shock absorbing properties of the material. Iterations using PLA and PETG filaments proved too brittle, and would therefore require thicker walls increasing the weight (see Fig. 33 in appendix). The tank was 3D printed using a dual extruder Ultimaker S3⁶ with water-soluble PVA support. The final printed tank was then coated with liquid silicone conformal coating, negating the water-absorbant properties of nylon and further ensuring the watertightness of the tank. Finally, a refill nozzle fitted with a check valve was installed, ensuring easy refill of the tank without the risk of water leaking out during operation.

⁶Ultimaker S3: <https://ultimaker.com/3d-printers/ultimaker-s3>

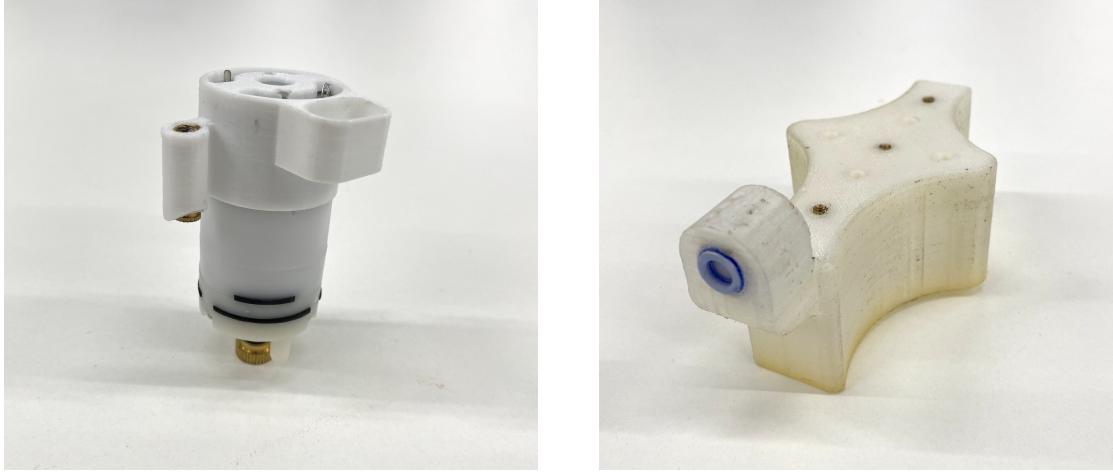


Fig. 9: Drone 3D Printed Components

4.2 Electronic Hardware

The onboard electronic package is composed of the flight controller (FC), sensor printed circuit board (PCB) and the companion computer. The drone was designed for these three components to be stacked on top of one another (see Fig.10) for compactness as well as to ensure the sensors would be as close as possible to the centre of gravity of the drone for accurate readings.



Fig. 10: Onboard Electronic Stack-up

4.2.1 Flight Controller & Electronic Speed Controller

At the heart of the drone is the flight controller, computing the control command to ensure the drone stays on track and balancing it in a PID control loop. Flight controllers work hand in hand with the electronic speed controller (ESC) which translates the thrust signals into the desired charge frequency of each of the brushless DC motor poles. While most drone systems separate these two components to reduce interference between the digital sensors of the flight controller and the analog power distribution board of the ESC, All-In-One (AIO) boards have grown in popularity in the recent years.

For space saving reasons and to have the ability to place a dedicated sensor board above the flight controller, the Flywoo Goku F7 AIO was chosen. This AIO has an 32-bit F7 processor from STM Electronics with 1 MB of flash, running at a frequency of 216 MHz, and is one of the fastest there is. The FC has an array of onboard sensors, namely a BMP280 barometric altimeter for altitude estimation and the MPU6000 three-axis accelerometer and three-axis gyroscope IMU. The ESC can output up to 40 Amps of current to the motors, fitting the required power output of the chosen motors.

At just 8.5 g, the Goku AIO was particularly well suited for the proposed solution, and its very small size of 33.5x33.5 mm is intended for small quadcopters while providing the necessary current output. Finally, the Ardupilot⁷ (Arducopter) firmware flashed on the flight controller and used to control the drone. Ardupilot is an open-source control system for drones, and offers multiple flight modes and tools for ease the development of autonomous drone systems (see Section 5.4)

While open-source solutions exist and are well documented, such as the Pixhawk, APM and KK2 for example, the dimensions of these were too large for the intended system. Additionally, these flight controllers all require the purchase of a separate ESC to control the motors, adding to the total cost. The chosen AIO did not restrict the development of the solution and offered a comparable number of input and output sources.

4.2.2 Custom Printed Circuit Board

As the proposed solution required several sensors and a motor driver, a PCB was designed, manufactured and assembled. A PCB was necessary for this application due to the vibrations of the drone

⁷Ardupilot: <https://ardupilot.org/copter/>

and as very little space was available on the drone for pre-built alternatives to fit. The four-layer circuit was designed using Altium Designer and matched the dimensions of the AIO with the same hole pattern (33.5x33.5 mm and 26.5x26.5 mm).



Fig. 11: Assembly and Final Assembled PCB (one pound coin for scale)

The PCB can be divided into 5 groups of components: DWM1001C, QMC5883L, DRV8871, Debugging and MPU-6050 (see Fig.12 for details).

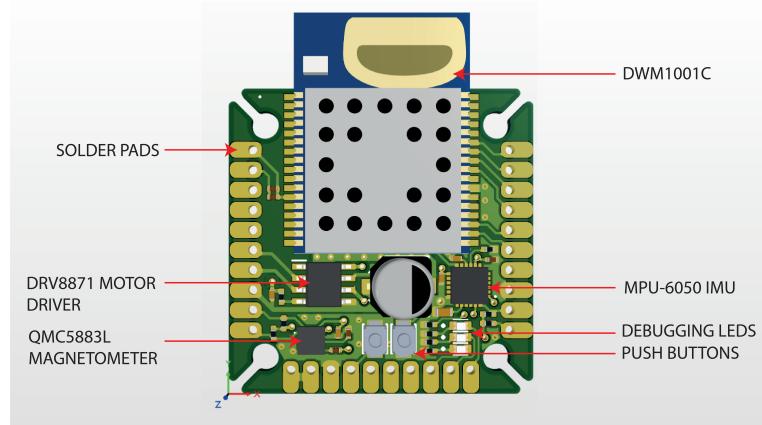


Fig. 12: PCB Components Diagram

The main component of the PCB is the DWM1001C chip from Qorvo⁸ (formerly Decawave). This chip is an ultra-wideband embedded system with a Nordic Semiconductor Micro-Controller Unit (MCU) and is the main component of the drone's positioning system. The PCB powers the chip and breaks out the SPI and UART communication pins. As detailed in Section 2.2.2, the chip serves as the positioning

⁸Qorvo DWM1001C: <https://www.qorvo.com/products/p/DWM1001C>

tag of the UWB system. Due to size constraints of the electronic package, pre-built boards like the DWM1001-DEV or third-party ones like Pozyx were not viable.

Another component of the PCB is the QMC5883L magnetometer. This chip is the compass of the drone and is directly connected to the flight controller using I2C. The use of a compass is crucial for non-GPS navigation to ensure the positioning system is aligned with the environment and in order to find magnetic north relative to where the aircraft is heading.

To control the speed of the water pump, the DRV8871 DC motor driver was added to the design. This gives greater granular control of the speed of the pump and therefore of the flow rate of sanitising liquid being sprayed and enabled the companion computer to activate the pump running at a higher voltage.

For debugging and development purposes, two push buttons and 3 LEDs were added to the PCB. This proved useful during the implementation of the sensor board and can serve as a mode button in future developments. Finally, the PCB was also designed to use an MPU-6050 IMU using the I2C communication protocol. The original positioning system intended on combining the data from the UWB with IMU data, however, this was not implemented in time and for this reason was not used in the final system.

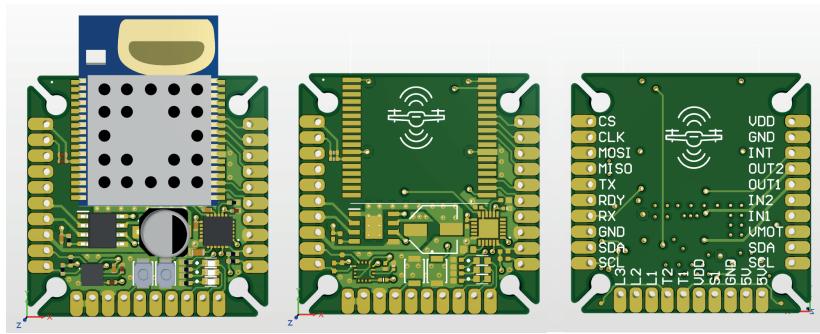


Fig. 13: PCB Design Overview (Altium Designer)

4.2.3 Companion Computer

The role of the companion computer is to interface and communicate with the flight controller and its firmware. In the developed solution, it serves as a WiFi bridge to the ground station, intermediate for the positioning system and control system of the drone. Raspberry Pi Zero 2 W was chosen as its Linux environment eased the development, its computational performance was sufficient and its onboard WiFi was functional, while being small enough for the drone to carry. Additionally, the ease of installation of a camera with the Pi Camera's dedicated port made it perfect for the proposed system. Finally, to hold the single board computer in place on the top side of the drone, a lightweight acrylic support was designed and laser cut.

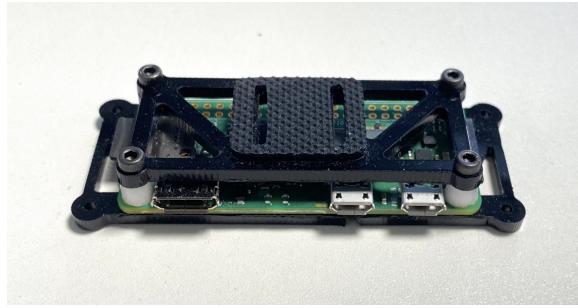


Fig. 14: Raspberry Pi Zero 2 W and its Acrylic Support

4.2.4 UWB Positioning Hardware

As explained in Section 2.2.2, for the UWB positioning system to be functional, an array of UWB anchors are required to be positioned within the environment. Following instructions from the manufacturer, 4 anchors were arranged following the diagram in Fig.28 with their device IDs and mounting height. As the altitude measurements were not necessary due to the onboard barometric altimeter and IMU measurements, and x-y accuracy was more crucial, the anchors were all placed at the same height. While this will give less accurate height measurement, it was found to produce more accurate results overall (see Section 6).

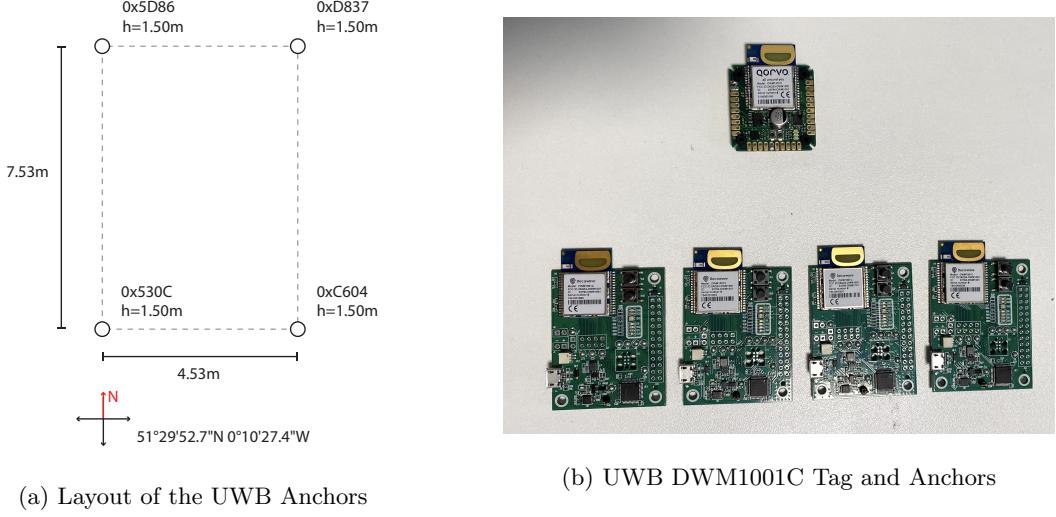


Fig. 15: Ultra-wideband Positioning System Hardware

4.3 Microsoft Kinect Hardware

To capture and map out the environment and the furniture in the space, a camera system was created using the Microsoft Kinect v2. These cameras are RGBD, therefore registering both depth content using an infra-red sensor, as well as colour using a traditional camera sensor. The depth camera maps out the environment between 0.5 and 4.5 meters, with a horizontal field of view of 70.6° . Among the RGBD cameras, the Kinect v2 is a very cost effective solution and for this reason is regularly used in research. The cameras were installed on tripods in three corners of the space to attempt to have as much of the environment within the camera's field of view.

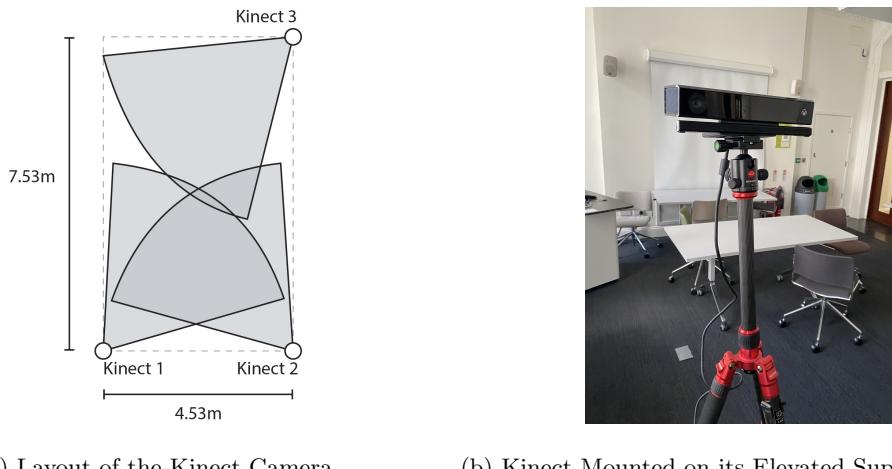


Fig. 16: Microsoft Kinect Camera System Hardware

4.4 Hardware Comparison

The objective of the hardware system is scalability and thus cost effectiveness. This will ensure the system can easily be implemented, and scaled to the needs of the target environment. All in all, the total associated cost of a 3 camera, 4 UWB anchors and 2 custom drone system with integrated UWB tag is £773.

4.4.1 Drone Hardware

While off-the-shelf drone solutions exist, such as DJI, Parrot and many others, these were not appropriate for our implementation for two reasons. First, the designed hardware was not designed to be modified, and for this reason, very few small-sized drones had the necessary torque to carry the water spraying hardware. Secondly, their closed software ecosystem would have it complicated for software personalisation.

Table 1: Comparison of Considered Off-The-Shelf Hardware Solutions

Name	Dimensions (mm)	Price (£)	Comment
Coex Clover ⁹	355x355x125	800	Dimensions too large, too expensive and requires additional UWB purchase
Ardubee ¹⁰	105x105x20	360	Failed Kickstarter campaign, underpowered for application
Bitcraze Crazyflie 2.1 ¹¹	92x92x29	250	Too small and underpowered for application
DJI Mini 2 ¹²	245×289×56	465	Dimensions slightly too large, requires additional UWB purchase difficulties with DJI proprietary firmware.
DJI Tello ¹³	98×92.5×40	100	Underpowered for application, requires additional UWB purchase difficulties with DJI proprietary firmware.
Parrot AR Drone 2.0 ¹⁴	517x517x110	50	Dimensions too large, requires additional UWB purchase, good Software Development Toolkit.

⁹Coex Clover: <https://coex.tech/clover>

¹⁰Ardubee: <https://www.kickstarter.com/projects/luminousbees/ardubee>

¹¹Bitcraze Crazyflie 2.1: <https://store.bitcraze.io/collections/kits/products/crazyflie-2-1>

By comparison, the designed drone cost a total of £276.15 and has an entirely open sourced hardware and software platform to build upon. This is cheaper than most solutions, and far better suited than what could have been obtained from retrofitting the required components.

4.4.2 Positioning Hardware

Decawave is the industry leader in ultra-wideband positioning systems and for this reason many of the available solutions rely on the DWM1000 chip inside their devices. Besides the DWM1001C development board used, a few alternatives are available on the market. It is important to note that some solutions are only available through the acquisition of a full dedicated system personalised for enterprises. Among these systems which could not be considered are Ubisense, Kinexon and IntraNav RTLS systems among others, offering UWB positioning in industrial settings. While the readily available Pozyx, Bitcraze or Sewio (see Table 2) systems may have more finely tuned onboard positioning software thus being reflected in the cost of the product, the price difference is quite substantial. From this overview of the alternatives on the market, it is evident that the proposed system implementation with 4 anchors and 2 tags worth £106.14 satisfies the goal of low-cost scalable system.

Table 2: Comparison of Considered RTLS Alternative

Name	Price (£)	Comment
Pozyx Kit ¹⁵	600	4 anchors, 2 tags
Bitcraze UWB Loco ¹⁶	690	4 anchors, 2 tags
Sewio Kit ¹⁷	3,000	5 anchors, 4 tags

4.4.3 Mapping Hardware

¹²DJI Mini 2: <https://store.dji.com/uk/product/mini-2>

¹³DJI Tello: <https://www.ryzerobotics.com/tello>

¹⁴Parrot AR Drone 2.0: https://www.parrot.com/assets/s3fs-public/2021-09/ar.drone2_user-guide_uk.pdf

¹⁵Pozyx Kit Lite: <https://store.pozyx.io/shop/creator-kit-lite-67>

¹⁶Bitcraze: <https://store.bitcraze.io/collections/positioning/products/loco-positioning-node>

¹⁷Sewio RTLS Kit: <https://www.sewio.net/product/indoor-tracking-rtls-uwb-wi-fi-kit>

A wide variety of RGBD cameras are available on the market, however the price varies greatly. While some alternatives have better resolutions, range of depth sensing or field of view, the Kinect v2 offers a solid performance in key metrics for a comparatively lower-cost. Ultimately, as the room mapping requires multiple cameras, the most cost effective solution was the purchase of the Kinect, totalling £114.72 for the 3 cameras and required hardware.

Table 3: Comparison of Considered Mapping Solutions

Name	Price (£)	Comment
Microsoft Azure Kinect ¹⁸	355	Higher RGB and Depth resolution with a smaller footprint
Intel RealSense Depth Camera D435 ¹⁹	240	Higher RGB and range with a smaller footprint
ASUS Xtion Pro Live	90	Discontinued, lower depth range.

¹⁸Microsoft Azure Kinect: <https://www.microsoft.com/en-gb/d/azure-kinect-dk>

¹⁹Intel RealSense D435: <https://www.intelrealsense.com/depth-camera-d435/>

5 Software Implementation

5.1 Positioning System

The positioning system relies on the Asymmetric Two-Way-Ranging Decawave Real-Time Location System (A-TWR-DRTLS) developed by Decawave. The system computes the distance between each anchor and a given tag, and estimates the exact position of the drone within the space based on a trilateration technique. The positioning software can use up to 16 anchors to give a position estimate and has a 10 Hz update rate. This firmware is not open-sourced, and therefore the actual implementation cannot be modified, but however provides an easy development platform accessible both through hardware communication (see Section 5.3.2) and through their Android app using Bluetooth (see Fig. 17).

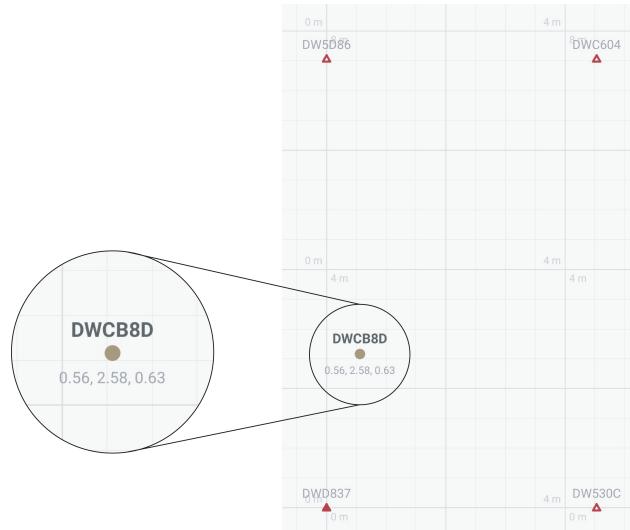


Fig. 17: Positioning Visualisation on DRTLS Management App (screenshot)

5.2 Room Mapping System

The developed room mapping system uses 3 Kinect v2 cameras connected to the USB 3.0 ports of a PC running Linux (Ubuntu 16.04). To produce the room mapping, Robot Operating System (ROS), a set of software libraries and tools that help you build robot applications²⁰ was used. The use of ROS proved very convenient as the raw data from the camera required to go through several steps to be used.

²⁰ROS: <https://www.ros.org/>

First the `libfreenect` driver was installed to provide access to the colour and depth image streams from the device, and fed into `iai_kinect2` to repackage the data into a camera feed usable by ROS in the form of topics. From this, the `depthimage_to_laserscan` package is used to converted the depth data from the RGBD camera into a laser scan therefore creating the planar cross section of the intersecting 3D objects. From this necessary step, the coordinate systems of the three cameras were determined and saved. The produced data is then fed into `rtabmap_ros`, a library using RGBD Simultaneous Localisation And Mapping (SLAM) algorithm to produce three-dimensional pointclouds and two-dimensional occupancy grid map. SLAM is a mapping technique relying on motion estimation and obstacle detection to gain cohesive understanding of the environment. As only one Kinect camera can be accessed at once, the algorithm loops through the three cameras, establishing the `kinect_bridge` with the camera and setting the coordinate system of said camera to the predetermined values. The data from the past mapping is stored in memory and the programmed believes the same camera has moved to a new location and has a new pose.

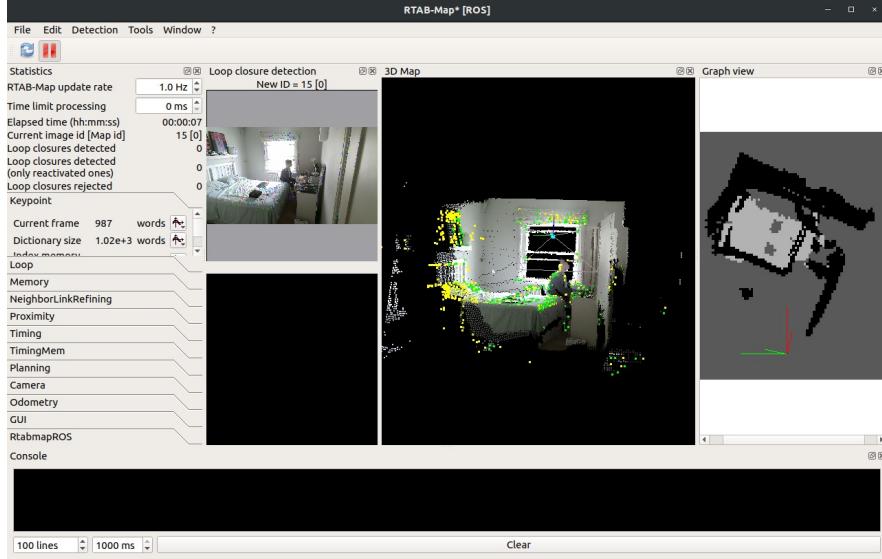


Fig. 18: Screenshot of the RTAB-Map Development GUI

From the above example taken during development (see Fig. 18), the three subsequent steps can be observed. First the RGBD data is fed to the system where a detection algorithm identifies markers. The flat horizontal surfaces, like the bed seen in the photo, are identified and colour coded in the two dimensional map in a light grey shade. This will identify the tables of the lecture hall needed to be sanitised. Furthermore, flat vertical surfaces, like walls the side profile of a shelf, are identified and

colour coded in black. This will be particularly useful to identify obstacles and the room constraints, crucial for any motion planning algorithm.

5.3 Communication Architecture

The proposed system has 4 MCUs running in parallel at all times, namely the ground station PC, Raspberry Pi companion computer, flight controller and DWM1001C embedded processor. A multi-modal communication architecture was developed to ensure the time-critical data was accessible to the relevant MCU at a given time. For instance, the FC's onboard Kalman Filtering algorithm requires the UWB ranging data for the real-time positioning and the ground station requires positioning coordinates for collision avoidance. The full communication architecture is visible in Fig. 19 below.

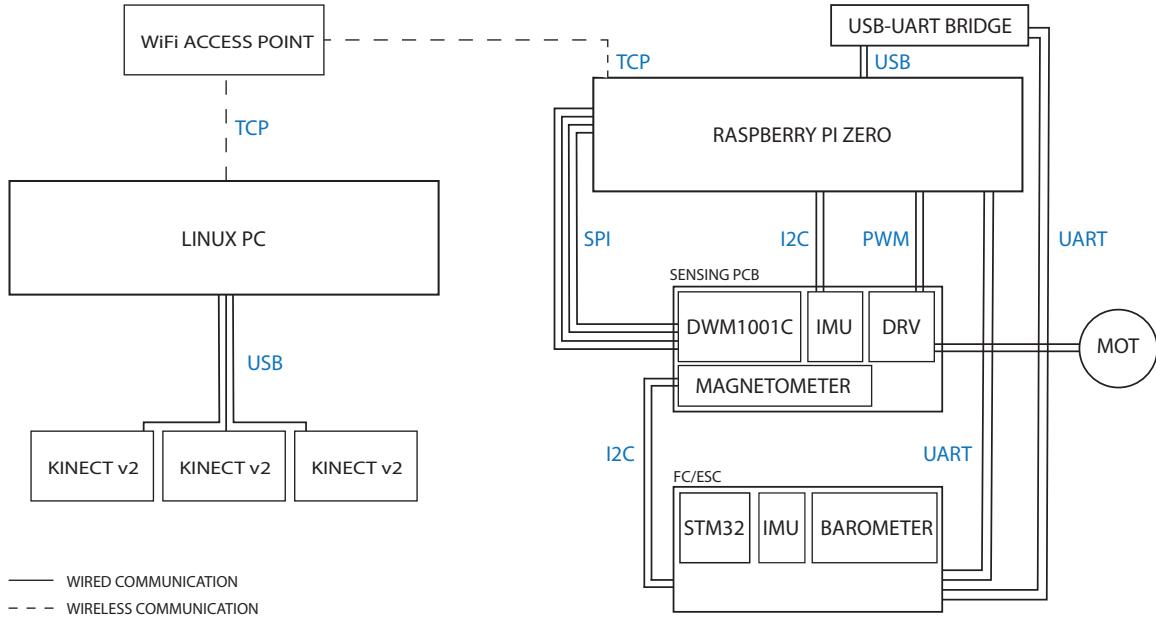


Fig. 19: Communication Architecture of Proposed Solution

5.3.1 TCP Socket

In order to communicate with each flying drone from the ground station, a Transmission Control Protocol (TCP) socket communication system was created. This communication process is very standard for multi-device communication and is very reliable, and only required the the ground station PC to become a communication server at a given static IP address on a local network for other devices to establish a communication with. The socket initialisation process can be simplified in two steps. The

socket is first created on the machine and then is bound to a server address. This opens the connection to any device, only requiring the IP address of the machine and the port number at which the socket used in the binding process. With the socket initialised and bound, the server now shifts to a passive state, where it waits and listens to connection requests from client sockets to accept and establish a connection to.

In the proposed system, a handshake procedure takes place, where the drone identifies itself to the server and establishes a connection, which in turn registers the id of the drone and its current position to local storage. When this handshake process is complete, the server triggers the setup of the on-board device, i.e. the connection to the flight controller and setup of the UWB positioning system. With the connection established, both the client and server can send and receive messages. Finally, to ensure both parties are connected and accessible at all times, the client periodically pings the server, effectively implementing a heartbeat message to detect dead connections.

A generic 2.4 GHz TP-Link router was used as an access-point for the system as the Raspberry Pi only supports the 2.4 GHz band and as the required bandwidth was quite low considering the data transmissions were restricted to 256 characters (bytes).

5.3.2 SPI, UART, USB

The DWM1001C communicates to the onboard system using the Serial Peripheral Interface (SPI) using a Type-Length-Value (TLV) encoding scheme in a request-response pattern triggered by the host device. Through the SPI connection, the Raspberry Pi can initialise and configure the DWM1001C module as well as reads the data transmission from the chip when the positioning data is ready. This step is required to repackage the data from the chip into a readable structure format for the FC's firmware to understand and process.

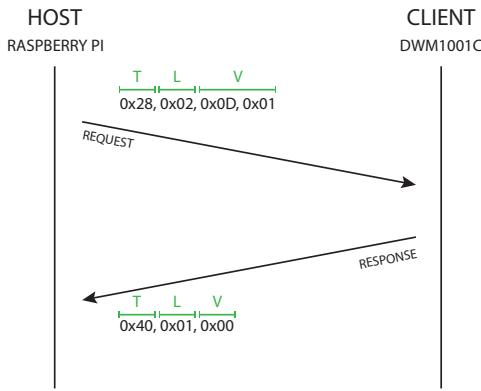


Fig. 20: Drone Flight State Diagram

The flight controller telemetry and control communication relies on Universal Asynchronous Receiver/Transmitter (UART) communication and the underlying MAVLink communication protocol (see Section 5.3.3). Due to hardware limitations of the Raspberry Pi Zero 2 W, a CP2102 USB to UART adapter board was soldered to the data transmission pads of the onboard USB of the board. Indeed, the FC would only accept the positioning data from the DWM1001C through a serial communication, however the only high-speed hardware serial port of the Raspberry Pi was already in use. Through this conversion layer the data could be repackaged in the correct protocol, encoding and baud rate.

5.3.3 MAVLink

The most crucial communication in the system is the serial link between the flight controller and the companion computer. The main role of this connection is reading telemetry data from the flight controller and sending flight control commands using the Micro Air Vehicle Link (MAVLink) communication protocol, and more specifically MAVLink v2. This lightweight messaging protocol is an industry standard for communication between onboard drone components and follows a publish-subscribe and point-to-point design pattern. In this sense, the serialised message format of MAVLink packets is structured as visible in Fig. 21 below²¹.

²¹MAVLink Protocol: <https://mavlink.io/en/>



Fig. 21: MAVLink Data Over-The-Wire Format

Table 4: MAVLink Data Types and Overview

Byte Index	C version	Content
0	uint8_t magic	Packet start marker
1	uint8_t len	Payload length
2	uint8_t incompat_flags	Incompatibility Flags
3	uint8_t compat_flags	Compatibility Flags
4	uint8_t seq	Packet sequence number
5	uint8_t sysid	System ID (sender)
6	uint8_t compid	Component ID (sender)
7 to 9	uint32_t msgid:24	Message ID (low, middle, high bytes)
n-byte payload	uint8_t payload[max 255]	Payload
(n+10) to (n+11)	uint16_t checksum	Checksum (low byte, high byte)
(n+12) to (n+25)	uint8_t signature[13]	Signature

A MAVLink message handler was created to serialise (pack) and deserialise (unpack) the messages between the flight controller and computer, as well as call upon the predefined MAVLink messages. These are defined in an XML file and are dependant on flight controller firmware used as well as the hardware. This also allowed for the creation of custom message to be sent to the controller for new functionalities.

5.4 Drone Firmware

The proposed system relies on the Ardupilot firmware to control the autonomous drone during flight. Among the features from the open-source firmware, Ardupilot was chosen due to its pre-defined flight

modes which suited the proposed solution, integrated PID calibration tool, UWB beacon integration as well as ease of customisation for non-traditional drone formats.

Among the flight modes, the Stabilise, Loiter and Auto mode were used in the implementation. The Stabilise mode is a fully IMU and barometer based PID control loop and is used to keep the drone level during takeoff. The Loiter mode takes the UWB positioning input to keep the drone in the same position within the space. Finally the Auto mode is used for the autonomous navigation using of the waypoint mission navigation.

The mission planning feature of Ardupilot was particularly relevant for the proposed solution. This functionality allows for a series of conditional commands to be given to the flight controller, in the form of waypoints for the drone to navigate to and events (such as takeoff and landing). Additionally, the speed and altitude can be specified between each target coordinate. Finally, the mission can be temporarily interrupted before being resumed, a particularly useful feature for the proposed solution in case of collision detection.

5.5 Drone Client Software

5.5.1 Overview

The drone client software was programmed in C++ as the on-board logic required fast-processing and an object-oriented software design made sense. The onboard software was sub-divided into two sections running in parallel on two separate threads. The first is the flight focused control loop, designed in a conditional control structure with a shared timer variable for time-critical and dependant processes to be triggered. Indeed within the loop the following steps are periodically called upon:

- Reads telemetry data from the flight controller.
- Reads the positional data from the DWM1001C.
- Sends a periodic heartbeat message to the flight controller.
- Sends the reformatted positional data to the flight controller.
- Sends the flight mode changes to the flight controller.
- Processes the received control command from the ground station.
- Prepares the TCP message (current position or mode change) to send to the ground station.

The second thread of the software is focused on the TCP connection. As this process required constant listening of the port for messages and response, a dedicated thread handles this. If a message is received, the message is placed in a shared buffer accessible to both threads, with a thread-safe mutex boolean flag to ensure data is not modified by both thread at once. The same process is followed when a message from the control thread is to be sent to the ground station over TCP.

5.5.2 Mission Control

When in operation, the companion computer is responsible for changing of flight modes and controls the current operation of the drone. The computer can either serve as an interface if ground station triggers a unexpected mode change, or a decision maker when actions are completed (see Fig. 22 below).

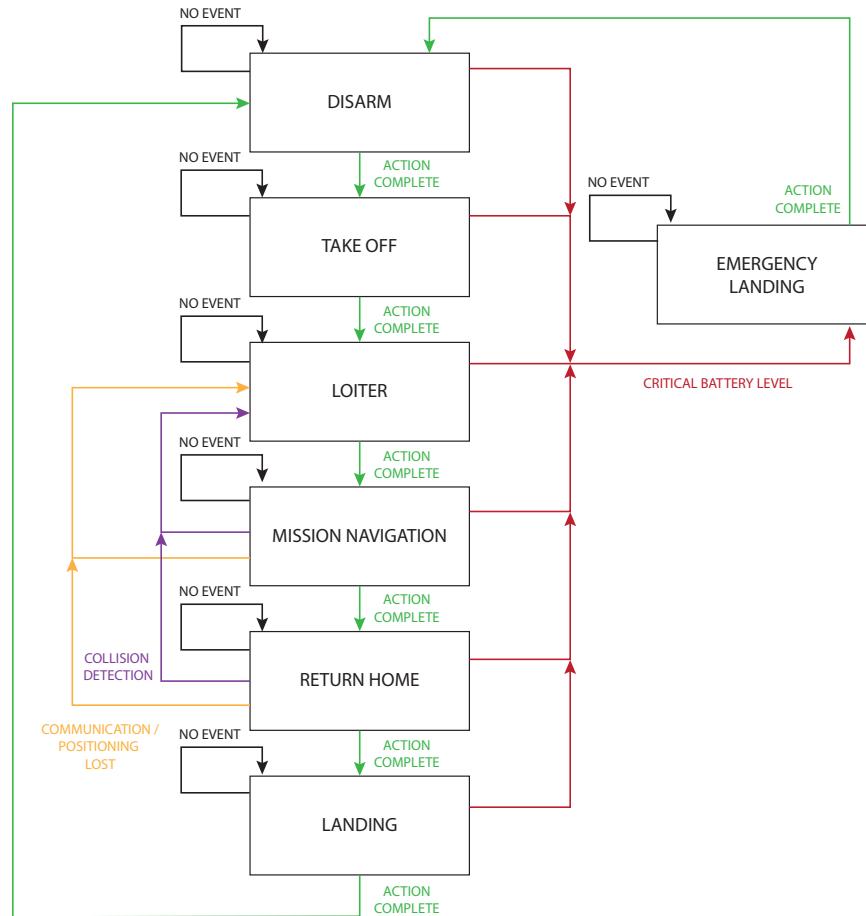


Fig. 22: Drone Flight State Diagram

Within the onboard software, the drone computer has its own states which modify what happens in the control loop. The states are defined as:

```
DRONE_STATE = { SETUP, DISARMED, ARMED, TAKE_OFF, STATIONARY, WAYPOINT_NAV, LANDING }
```

For instance, during the setup phase, the computer initialises the different sensors and ensures the different subsystems like the positioning system are functioning as intended. This ensures only the computation and communication necessary for the current state are being triggered, thus reducing power consumption and increasing the theoretical maximum flight time.

Before the mission start, the ground station computer sends the waypoints the drone need to follow during flight. These waypoints are communicated through a series of TCP messages, stored in the onboard computer's memory, repackaged into MAVLink mission command format and sent to the flight controller.

5.6 Drone Server Software

The ground station server operates as the initiator and overseer of the operation of the system. It is designed as a C++ command line script, in a threaded parallelised manner similarly to the onboard software. The TCP server is initialised on one thread and listens to the TCP messages from up to 5 drone clients. This theoretically could be expanded to a larger number of clients, however the communication architecture will require modifications to ensure there are no delays between each listening operation. The second thread runs the control loop parsing and handling the function calls.

The user can input messages to call functions and to change the flight modes of each of the connected drones. The text-based commands are sliced based on colons following a standardised pattern, as visible in Fig. 23 below:



Fig. 23: Server Input Command Example

While the motion planning algorithm was not developed as part of the proposed system, server software was designed to be able to launch the ROS room mapping script, which would output the two-

dimensional occupancy grid map, which would then be used to compute the most optimal path for each drone and thus initialising the waypoint navigation of the drones automatically.

6 Results & Discussion

6.1 Drone Evaluation

For a comprehensive evaluation of the drone hardware and software performance, a series of experiments were conducted. These attempted to study how well the initial design constraints were satisfied and how close to an autonomous drone system the produced work is.



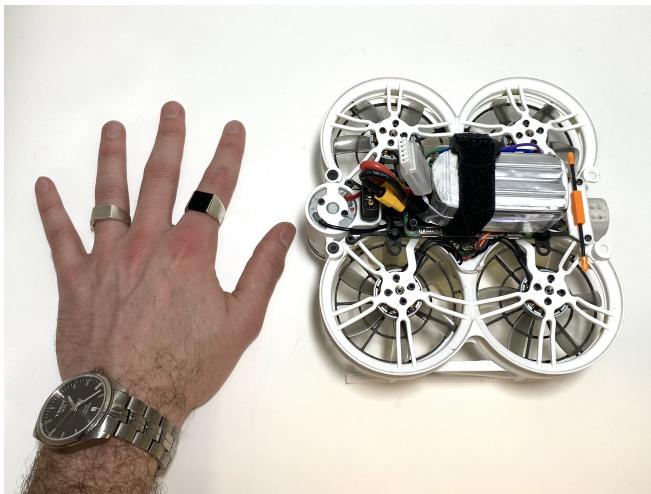
Fig. 24: Final Assembled Drone (front)

Initial drone flights proved very unstable due to the improper tuning of the drone. This would cause the drone to sway from side to side as well as oscillate. This was not surprising as the default PID values used were only based on the drone's wingspan and propeller size and the designed solution is indeed far heavier than typical drone systems of the same size due to the spraying mechanism and large battery. To mitigate this, multiple rounds of autotuning were conducted, with each iteration improving the stability of flight and ultimately reaching acceptable stability for use.

6.1.1 Drone Hardware

The development of the hardware platform of the drone for indoor sanitation is a major aspect of the produced work. The developed hardware construction weighs a total of 397 g fully assembled

with the battery (empty solution tank) and measures 150x165x80 mm; this is within the footprint bounds set during conception. The side and bottom propeller guards work effectively to both protect the propellers from the environment and vice-versa. Indeed, over the duration of indoor testing, no damages have been observed to either the room or frame in itself. While no exact thrust measurements were conducted as no experimental equipment could be constructed within the time frame, the total lift was instead measured. It was found that the drone would only just be able to take off with an additional 500 g weight added to the drone frame. From this we can estimate the total produced thrust of the system to be between 900 g and 1 kg. While this is lower than the anticipated value estimated from the datasheet values, the thrust to weight ratio is very close to 2:1 and therefore shouldn't require the motors to run at more than 60% throttle to hover, and therefore should not lead to excessive strain. Finally to test out the drone's power consumption and autonomy, the drone was flown indoors until the battery cell voltage dropped to 3.5 V. The drone managed to stay airborne for 6 minutes and 33 second. While the drone was not following a set motion plan and instead was flown using a transmitter, it can be assumed with a high degree of confidence that it would have managed to stay airborne for a similar amount of time in waypoint navigation mode.



(a) Drone Top View (hand for scale)



(b) Drone Spraying System (50% PWM duty cycle)

Fig. 25: Final Assembled Functioning Drone

6.1.2 Drone UWB Flight

The integration of UWB positioning within the Ardupilot flight control attempted to bridge the gap between sensing and actuation. While the UWB positioning could be implemented, some underlying issues and software bugs were identified during testing. The indoor loitering method could be tested with good success, enabling the drone to stay at a steady position, correcting its offset. The drone stayed within a 0.5 meter radius of the intended position during the 30 second test. However, during development an issue persisted with UWB based waypoint navigation, which caused the drone to lose the waypoints as the mission started. While in depth investigation and testing helped identify the root cause of this software issue, it was not solved in time before the end of the project. With more development time, the UWB navigation problem will be fixed and fully autonomous flight will be tested.



Fig. 26: Indoor Drone Test Flight

6.2 Room Mapping Evaluation

To evaluate the room mapping system developed, the camera system was fitted inside the Dyson Lecture Hall with eight tables. Based on the maximum range of the camera, it was found that if placed in the corners of the room, the central part would not be mapped out. To be able to map out as much of the space as possible, the tables were placed closer to the centre of the room and the cameras tripods were placed in the corners of this theoretical smaller room. The developed ROS program was then launched and the occupancy grid map of the lecture hall was obtained (see Fig. 27 left).

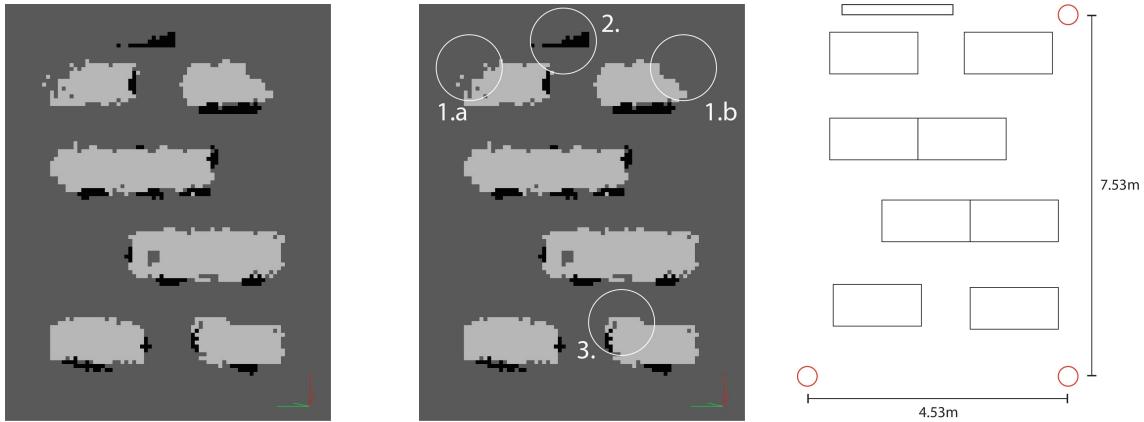


Fig. 27: Occupancy Grid Map of the Lecture Hall and Associated Diagrams

From the produced map, the eight tables can clearly be identified in grey and their associated dimensions seem to match that of the tables in the real world. In black are the vertical obstacles identified by the system, namely the chairs and legs of the table. However, a few elements are interesting to note. First, while most of each table was captured by the cameras, some sections are missing from the final map. Indeed, as annotated 1.a and 1.b in the middle subfigure of Fig. 27, the far reaching regions as well as close proximity were cut off by the Kinect in the top right corner. This illustrates the requirement for fine tuning of the final positions of each camera if the entire environment needs to be mapped out. Secondly, as annotated 2. in the same diagram, for three-dimensional obstacles to be correctly mapped out, these need to be directly facing the camera. Indeed, the annotated obstacle is a whiteboard, however, its shape and overall dimensions were lost in this process and could therefore have caused a collision if the motion plan was based solely on this produced map. This highlights the requirement of multiple angles of each object to truly reconstruct the three-dimensional mapping of the space. Finally, as annotated 3 in the subfigure, some issues can be seen from the overlay of the mapping from two different cameras. Indeed, in the annotated area, both cameras successfully mapped out the table, however, due to some calibration offset, the estimated position of the object from both cameras do not match. This could be simply due to a calibration error which could be solved in a tuning process, or could be a limitation of the cameras being unable to correctly map out far reaching objects.

6.3 UWB Positioning Evaluation

To evaluate the effectiveness of the proposed UWB positioning, experiments were conducted to measure the accuracy of the system. For the experiments, an assembled drone mounted with the positioning circuit board was carried along the target perimeter demarcation lines. The target path was a square of 2m side, positioned in the middle of the room (see Fig. 28b). Two different paces were used, first a *faster* pace, at an approximate speed of 19 m/min (1.14 km/h), followed by the measurement at a *slower* pace of about 14 m/min (0.84 km/h). The sensor was positioned at an altitude of 2 m, simulating the intended cruising of the drone during operation. The experiment started from the bottom left corner of the square and the path was followed in a clockwise direction. The positioning data was collected for a duration of 1 minute for each trial and was processed and saved directly on the onboard Raspberry Pi.

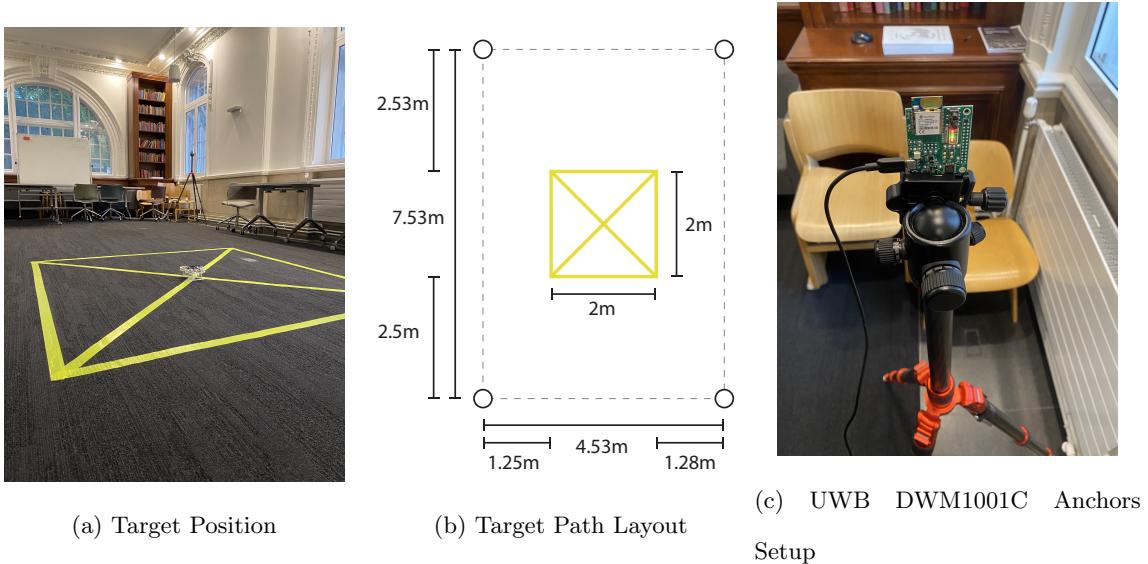


Fig. 28: DWM1001C UWB Positioning System Experimental Setup

From the estimated position obtained, the coordinates were plotted overlaid on top the expected coordinates and the values were studied. The colours of the points are based on the timestamp value on a gradient from red to blue. As no exact real position from a calibrated external positioning system could be used, only the horizontal error values were computed on the vertical displacement and the vertical error on the horizontal ones.

6.3.1 UWB Position "Fast" Results

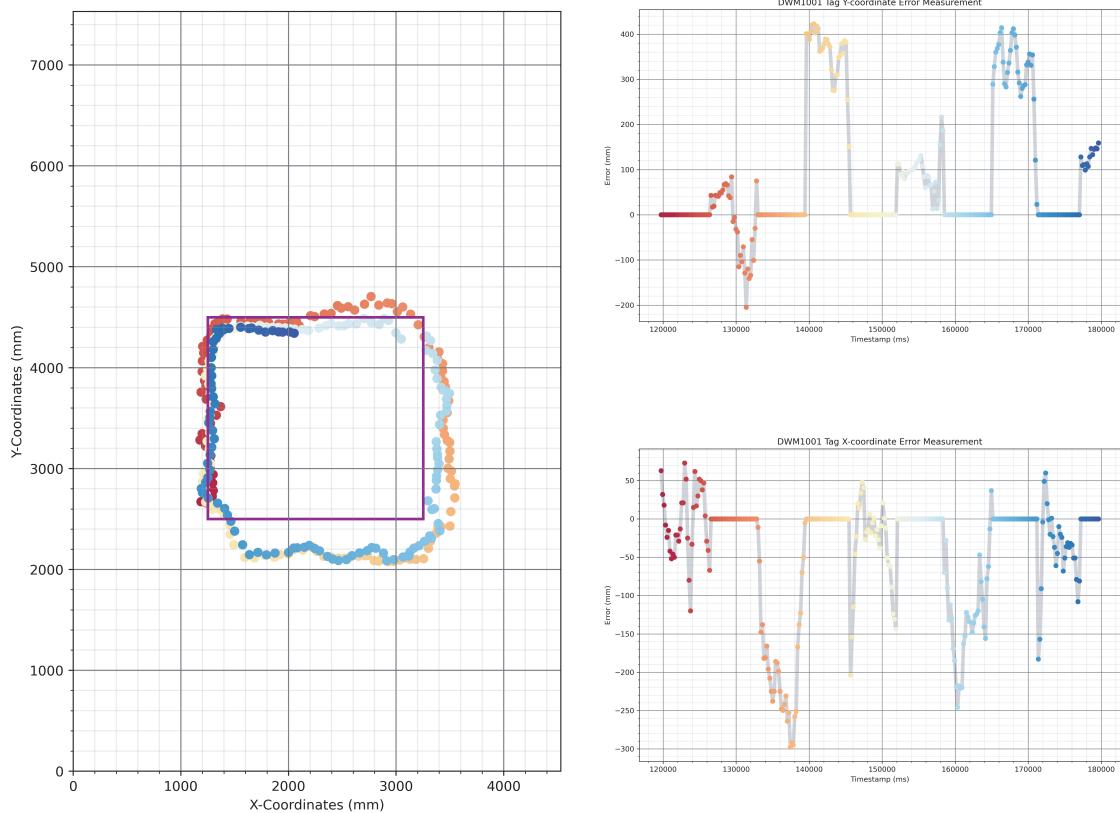


Fig. 29: UWB Position Results with Fast Pace

Table 5: Cartesian Position Error Metrics for Fast Pace

	X (mm)	Y (mm)
Average	-80.3	179.6
Median	-51.2	129.5
Standard Deviation	91.6	163.0
RMSE	121.6	248.1

6.3.2 UWB Position "Slow" Results

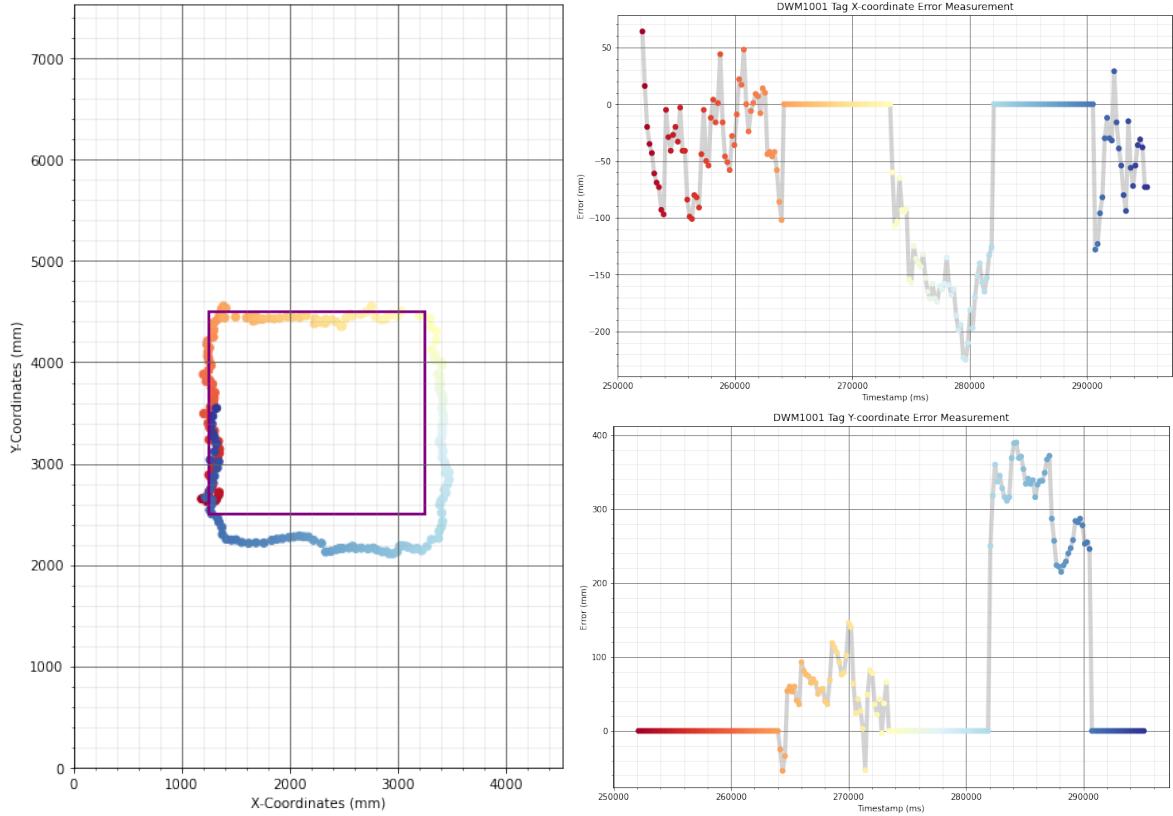


Fig. 30: UWB Position Results with Slow Pace

Table 6: Cartesian Position Error Metrics for Slow Pace

	X (mm)	Y (mm)
Average	-81.6	121.4
Median	-73.1	69.5
Standard Deviation	70.5	128.9
RMSE	107.6	176.3

The standard deviation of the position values, as shown in Table 5 & 6 is contained below 100mm in the x-axis and 170mm in the y-axis. This is close to the expected ideal positioning accuracy advertised by UWB and observed in existing research implementing UWB positioning systems [31,32].

Through visual inspection as well as looking at the statistical analysis of the results, it is evident that the positioning accuracy is far better in the x-axis than the y-axis. This can be seen in both the drifting of the data on the bottom side of the square in Fig 29 & 30 as well as in the 44.9% greater standard deviation of the positioning values in y over those in x as shown in Table 5 & 6. This discrepancy can be attributed to the overall distance between anchors in the y-axis. Indeed, while the horizontal anchors are 4530 mm apart, the vertical anchors are 7530 mm. While the chosen layout is the one recommended one for a 4 anchor indoor setup, the data indicates that the system would benefit from the anchors being more equally spaced apart. A simple mitigation method would be installing anchors splitting the difference between the vertical anchors.

As observed in the Tables 5 & 6, the accuracy of the positioning increases with a reduced speed. Indeed, with a 26.3% velocity decrease, the standard deviation decreased by 23.0% and 20.9% in x and y respectively. While this result was expected, it is interesting to note and should be a critical factor when determining the speed of the drone during operation in a working system. Additionally, this could also be relevant when implementing such a system in a particularly large environment, where the inherent distance between the drone and the 4 nearest anchor may be greater.

The obtained results both validate the expected accuracy of the UWB system and help identify flaws within the proposed solution for potential improvements. By increasing the number of tags to 6 in the tested environment, the standard deviation would be expected to be decreased to below 100mm in both the x and y axis. One potential downfall of this implementation would be the 10 Hz refresh rate of the TWR positioning. While this was not tested in enough depth, it would be interesting to study TOF of TOA positioning methods with lower expected accuracy but with a higher refresh rates and compare the different implementations.

6.4 Limitations

While the designed hardware solution worked well during testing and experiments, and satisfied the constraints established during the conception phase, one issue which was not considered ahead of time was identified: the sound of the quadcopter. While this did not hinder testing or development of the solution, extended testing attracted a lot of attention and even complaints from neighbouring students. Further testing of the system with a set of propellers of different design may identify a more adequate solution.

While the PID control loop was sufficiently well tuned for testing, it is expected that further parameter tuning would be beneficial for the overall system. The main focus of the next phase of tuning would be on tailoring the drone flight to indoor navigation. This will entail reducing the maximum allowed yaw and tilt angles to smaller values, as well as reducing the maximum allowed travel speed. The maximum travel speed will be set to 12 m/min (0.72 km/h), which would be even lower than the tested positioning speed in Section 6.3.2, and thus which should give even more accurate position readings. While this would decrease the agility of the drone, this would prove useful for smaller confined spaces where a more controlled navigation approach would be more fitting.

While the developed camera system was functional for the current application of the sanitisation system, the narrow field of view of the Kinects proved very restrictive and poorly scalable. A future iteration of the system would have to improve upon this. One of three solution could be investigated. First, a rotating base for the Kinect mounted on a support structure with a rotational encoder could be created. The rotational position would then be fed back into the ROS mapping algorithm during processing of the RGBD data. This solution would be quite cost effective and would only require a minimal number of components. An alternative solution would be purchasing a new array of RGBD cameras with a wider field of view, such as the new Microsoft Azure Kinect or Mynt Eye D1000 offering a horizontal field of view of up to 120°. Finally, a system using projection mapping with custom light bulbs fitted with a central camera could be investigated. While this could be an added constraint of the system, it would make the system less intrusive with the imposing depth sensing camera around the room.

7 Conclusion

7.1 Summary

The deployment of an autonomous drone system for sanitisation of indoor public spaces is a promising automated solution for a growing real world problem.

The hardware platform, including frame and custom electronics, was designed, tested and satisfied the constraints established during the initial conception phase. The micro-quadcopter concept was able to efficiently vaporise antibacterial solution within a space and had a battery life of over six minutes within the desired footprint. The control software system was developed and tested successfully, enabling a ground station to initialise the autonomous drone flight and creating a dedicated communication link between them. The developed UWB drone positioning system created an accurate localisation estimate within the room, enabling for sub-13 cm horizontal and vertical standard deviation error respectively. The room mapping algorithm developed worked successfully within its constraints and created a solid data acquisition method for an environment aware system. While only subsystem validation could be obtained as issues with UWB waypoint navigation could not be resolved in time, these have confirmed the potential of the proposed solution and highlighted its potential.

7.2 Future Work

When the development of the UWB-positioning based waypoint navigation will be finished, the system could be tested further in real-world situations. This will also open the door to testing with multiple UAVs collaboratively, and allow for evaluation of the multi-agent system that was hoped to be created.

The hardware platform and software system developed during this project are a solid foundation for an autonomous drone system, however, additional developments have the potential of creating a system ready for deployment. For a fully autonomous system to be developed upon the proposed solution, a charging and refill base system should be designed. The base in itself should be fitted with charging contact pads to deliver the current to the drone, as well as a telescopic nozzle mechanism to refill the tank. On the drone side, this would require an additional battery charging and balancing circuit for the LiPo battery. Building upon the current system, a downward facing camera and computer vision algorithm would be able to provide the centimetre level accuracy required for the drone to land

accurately on a base. Similar implementations have been studied independently in research [33, 34] and in industry from companies like Hextronics and Heisha Tech among others with positive results.

8 Bibliography

- [1] Elena López, Sergio García, Rafael Barea, Luis M. Bergasa, Eduardo J. Molinos, Roberto Arroyo, Eduardo Romera, and Samuel Pardo. A multi-sensorial simultaneous localization and mapping (SLAM) system for low-cost micro aerial vehicles in GPS-denied environments. *Sensors (Switzerland)*, 17(4), 4 2017.
- [2] Ruiqi Li, Peter Richmond, and Bertrand M. Roehner. Effect of population density on epidemics. *Physica A: Statistical Mechanics and its Applications*, 510:713–724, 11 2018.
- [3] Geoffrey P. Garnett and James J. C. Lewis. The Impact of Population Growth on the Epidemiology and Evolution of Infectious Diseases. *HIV, Resurgent Infections and Population Change in Africa*, pages 27–40, 2007.
- [4] Rico Merkert and James Bushell. Managing the drone revolution: A systematic literature review into the current use of airborne drones and future strategic directions for their effective control. *Journal of Air Transport Management*, 89:101929, 10 2020.
- [5] Higinio González Jorge, Luis Miguel González de Santos, Noelia Fariñas Álvarez, Joaquin Martínez Sánchez, and Fermin Navarro Medina. Operational Study of Drone Spraying Application for the Disinfection of Surfaces against the COVID-19 Pandemic. *Drones 2021, Vol. 5, Page 18*, 5(1):18, 3 2021.
- [6] K. Ramesh, B. Priya Dharshini, K. Haridass, S. Deepak Kumar, R. Gokul Raj, and V. Hariprasad. Sanitization using Hexacopter Autonomous Drone. *IOP Conference Series: Materials Science and Engineering*, 1059(1):012043, 2 2021.
- [7] Ernesto A. Paiva, Juan C. Soto, Julio A. Salinas, and William Ipanaque. Modeling and PID cascade control of a Quadcopter for trajectory tracking. *CHILECON 2015 - 2015 IEEE Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies, Proceedings of IEEE Chilecon 2015*, pages 809–815, 2 2016.
- [8] PX4 Autopilot. Controller Diagrams — PX4 User Guide.
- [9] Abhishek Sharma, Pankhuri Vanjani, Nikhil Paliwal, Chathuranga M.Wijerathna Basnayaka, Dushantha Nalin K. Jayakody, Hwang Cheng Wang, and P. Muthuchidambaranathan. Communication and networking technologies for UAVs: A survey. *Journal of Network and Computer Applications*, 168, 9 2020.
- [10] Dan Mototolea. A study on the actual and upcoming drone communication systems. *ISSCS 2019 - International Symposium on Signals, Circuits and Systems*, 7 2019.

- [11] Gerton de Goeij, Eildert H. van Dijken, and Frank Brouwer. Research into the Radio Interference Risks of Drones. 2016.
- [12] M. C. Pérez, D. Gualda, Jorge De Vicente-Ranera, J. M. Villadangos, and J. Ureña. Review of UAV positioning in indoor environments and new proposal based on US measurements. *IPIN*, 2019.
- [13] Nils Gageik, Michael Strohmeier, and Sergio Montenegro. An autonomous UAV with an optical flow sensor for positioning and navigation. *International Journal of Advanced Robotic Systems*, 10, 10 2013.
- [14] S. Chen, H. Chen, W. Zhou, C. Y. Wen, and B. Li. End-to-End UAV Simulation for Visual SLAM and Navigation. 12 2020.
- [15] Ahmed Abdulmajuid, Osama Mohamady, Mohannad Draz, and Gamal El-Bayoumi. GPS-Denied Navigation Using Low-Cost Inertial Sensors and Recurrent Neural Networks.
- [16] Davide Scaramuzza, Michael C Achtelik, Lefteris Doitsidis, Fraundorfer Friedrich, Elias Kosmatopoulos, Agostino Martinelli, Markus W Achtelik, Margarita Chli, Savvas Chatzichristofis, Laurent Kneip, Daniel Gurdan, Lionel Heng, Gim Hee Lee, Simon Lynen, Marc Pollefeys, Alessandro Renzaglia, Roland Siegwart, Jan Carsten Stumpf, Petri Tanskanen, Chiara Troiani, Stephan Weiss, and Lorenz Meier. Vision-Controlled Micro Flying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments. *IEEE Robotics Automation Magazine*, 21(3):26–40, 2014.
- [17] Francisco Javier Gonzalez-Castano, Felipe Gil-Castineira, David Rodriguez-Pereira, Jose Angel Regueiro-Janeiro, Silvia Garcia-Mendez, and David Candal-Ventureira. Self-Corrective Sensor Fusion for Drone Positioning in Indoor Facilities. *IEEE Access*, 9:2415–2427, 2021.
- [18] I. A. Glover and R. Atkinson. Overview of wireless techniques. *Wireless MEMS Networks and Applications*, pages 1–33, 1 2017.
- [19] Wang Shule, Carmen Martínez Almansa, Jorge Peña Queralta, Zhuo Zou, and Tomi Westerlund. UWB-Based Localization for Multi-UAV Systems and Collaborative Heterogeneous Multi-Robot Systems. *Procedia Computer Science*, 175:357–364, 1 2020.
- [20] Ang Liu, Shiwei Lin, Xiaoying Kong, Jack Wang, Gengfa Fang, and Yunlong Han. Development of Low-Cost Indoor Positioning Using Mobile-UWB-Anchor-Configuration Approach. *Communications in Computer and Information Science*, 1362:34–46, 12 2020.
- [21] Janis Tiemann and Christian Wietfeld. Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization. *undefined*, 2017-January:1–7, 11 2017.

- [22] Cung Lian Sang, Michael Adams, Timm Hörmann, Marc Hesse, Mario Porrmann, and Ulrich Rückert. Numerical and experimental evaluation of error estimation for two-way ranging methods. *Sensors (Switzerland)*, 19(3), 2 2019.
- [23] Wang Guosheng, Qin Shuqi, Lv Qiang, Wei Heng, Lin Huican, and Liang Bing. UWB and IMU System Fusion for Indoor Navigation. *Chinese Control Conference, CCC, 2018-July*:4946–4950, 10 2018.
- [24] Widyawan, Martin Klepal, and Dirk Pesch. Influence of predicted and measured fingerprint on the accuracy of RSSI-based indoor location systems. *4th Workshop on Positioning, Navigation and Communication 2007, WPNC'07 - Workshop Proceedings*, pages 145–151, 2007.
- [25] Jaegeol Yim. Comparison between RSSI-based and TOF-based Indoor Positioning Methods. *International Journal of Multimedia and Ubiquitous Engineering*, 7(2), 4 2012.
- [26] Pratik Palaskar, Rajesh Palkar, and Mayur Tawari. Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points –A Tri-lateration Approach. *International Journal of Scientific & Engineering Research*, 5(4), 2014.
- [27] Veli Ilci, V. Engin Gulal, Reha Metin Alkan, and Huseyin Cizmeci. Trilateration Technique for WiFi-Based Indoor Localization. pages 25–28, 10 2015.
- [28] Nuno Gracias, Rafael Garcia, A S M Shihavuddin, Ricard Campos, Natalia Hurtos, Ricard Prados, Tudor Nicosevici Nicosevici, Armagan Elibol, Laszlo Neumann, and Javier Escartin. Computer Vision in Vehicle Technology: Land, Sea & Air. *Computer Vision in Vehicle Technology: Land, Sea, and Air*, pages 133–160, 2017.
- [29] Vadim Kramar. UAS (drone) in Response to Coronavirus. In *2020 27th Conference of Open Innovations Association (FRUCT)*, pages 90–100, 2020.
- [30] Yuvaraj.S, C.J. Thomas Renald, Artūras Jukna, J. David Rathnaraj, and M. Nallamani. Design and Numerical Analysis of Five Blade Propeller for a Drone. *International Journal of Engineering and Advanced Technology (IJEAT)*, 8(2S2), 1 2019.
- [31] Kevin Minne, Nicola Macoir, Jen Rossey, Quinten Van Den Brande, Sam Lemey, Jeroen Hoebeke, and Eli De Poorter. Experimental Evaluation of UWB Indoor Positioning for Indoor Track Cycling. *Sensors (Basel, Switzerland)*, 19(9), 5 2019.
- [32] Dadiq Tian and Qiang Xiang. Research on Indoor Positioning System Based on UWB Technology. *Proceedings of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020*, pages 662–665, 6 2020.

- [33] María Álvarez Custodio. Autonomous Recharging System for Drones: Detection and Landing on the Charging Platform, 2019.
- [34] Aqeel Mahmood Jawad, Haider Mahmood Jawad, Rosdiadee Nordin, Sadik Kamel Gharghan, Nor Fadzilah Abdullah, and Mahmood Jawad Abu-Alshaeer. Wireless power transfer with magnetic resonator coupling and sleep/active strategy for a drone charging station in smart agriculture. *IEEE Access*, 7:139839–139851, 2019.

9 Appendix

9.1 Source Code

Please find the source code and additional resources regarding this project on the Github repository:

<https://github.com/colinlaganier/AutonomousDrone>

9.2 Spraying Solution Testing



Fig. 31: Testing Experiment to Identify Volume Required

To evaluate the volume of water needed to be carried for the sanitation to be effective, the volume of solution needed to clean the 24 tables of the Dyson Level 0 Lecture Hall was calculated. The final volume of water used to clean the tables was found to be **128 mL**.

9.3 Finite Element Analysis

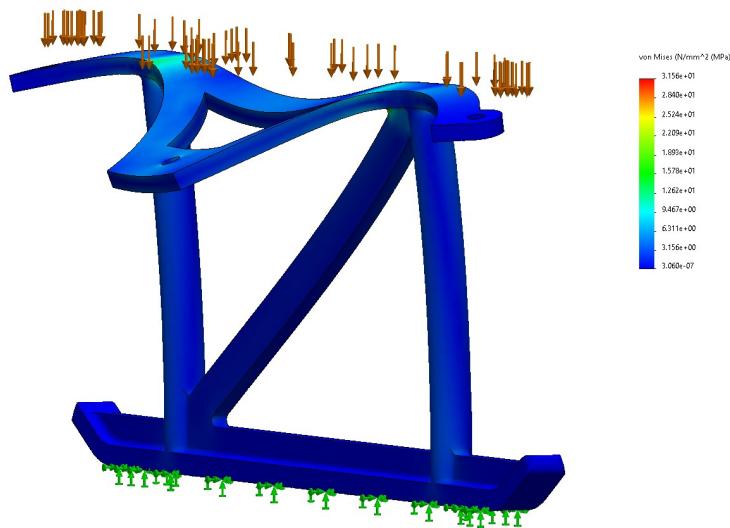


Fig. 32: Static Finite Element Analysis of the Drone's Support Leg

For a 6 meter high fall the maximum stress observed in the component is of **31.7 MPa**, less than the expected tensile strength of 3D printed PLA filament which is of about 37 MPa²².

²²BCN3D, PVA: <https://www.bcn3d.com/pla-filament-stands-for-strength-temp/>

9.4 Water Tank Manufacturing Iterations

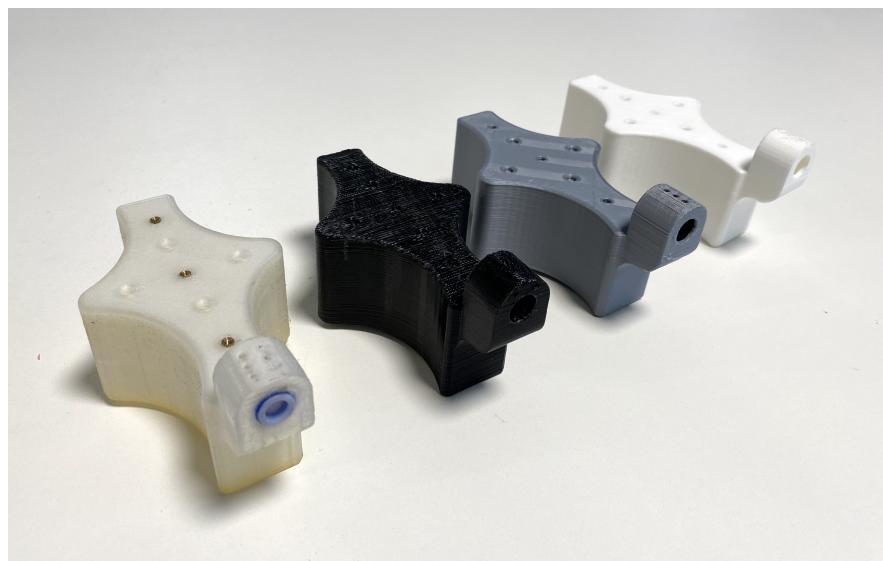


Fig. 33: Watertank 3D Printed Manufacturing Iterations

9.5 Bill of Materials

Table 7: Drone Bill of Materials (weight are estimates or as advertised)

Component	Name	Number	Weight/unit (g)	Price/unit (£)
Frame	Diatone Taycan C25 MK2	1	55	24.95
Motor	BetaFPV 2004 3000kv	4	16.6	12.92
Propellers	HQprop T63mmx6 63mm 6-blade Poly Carbonate Propeller	4	1.85	0.49
/FC/ESC	Flywoo Goku GN 745 40A V1.2	1	8.5	93.87
PCB	Custom PCB	1	20	48.05
Battery	Tattu 850mAh 14.8V 75C 4S1P Lipo Battery Pack	1	109	15.44
Companion Computer	Raspberry Pi Zero 2 W	1	11	24
Pump	RF310 Micro Diaphragm Pump	1	40	3.2
Camera	Raspberry Pi Camera	1	3	10
Water Tank	Custom Water Tank	1	37.5	3
Nuts & Bolts	-	50	0.25	0
3D Printed Parts	-	-	20	0
Total	-	-	390.3	276.15

9.6 PCB Bill of Materials

Table 8: PCB Bill of Materials

Comment	Designator	Footprint	JLCPCB #	Description
CL05B104KO5NNNC	C1, C3, C5, C9, C11	0402	C1525	16V 100nF X7R ±10% 0402 Multilayer Ceramic Capacitors - SMD ROHS
CL05A106MQ5NUNC	C10	0402	C15525	6.3V 10uF X5R ±20% 0402 Multilayer Ceramic Capacitors - SMD ROHS
EEHZAJ220XV	C2	CAP-SMD	C454361	63V 22uF ±20% 8x6.3x8mm Solid Polymer Electrolytic Capacitor ROHS
CL05B103KB5NNNC	C4	0402	C15195	50V 10nF X7R ±10% 0402 Multilayer Ceramic Capacitors - SMD ROHS
0402B222K500NT	C6	0402	C1531	50V 2.2nF X7R ±10% 0402 Multilayer Ceramic Capacitors - SMD ROHS
CL05B224KO5NNNC	C8, C12	0402	C16772	16V 220nF X7R ±10% 0402 Multilayer Ceramic Capacitors - SMD ROHS
KT-0603W	D1, D2, D3	LED_0603	C2290	White 0603 Light Emitting Diodes (LED) RoHS
DRV8871DDAR	IC001	SOIC-8_3.9x4.9x1.27P	C75864	SO-8 Motor Drive ICs ROHS
MPU-6050	IC002	QFN-24_EP_4.0x4.0x0.5	C24112	QFN-24_4x4x0.5P Attitude Sensors ROHS
QMC5883L	IC003	LGA-16(3x3x0.9mm)	C976032	LGA-16(3x3x0.9mm) Magnetic Sensors ROHS
0402WGF3002TCE	R1	0402	C25776	±1% 1/16W Thick Film Resistors 50V 30kΩ 0402 Chip Resistor - SMD ROHS
0402WGF1002TCE	R2	0402	C25744	±1% 1/16W Thick Film Resistors 50V 30.1Ω 0402 Chip Resistor - SMD ROHS
0402WGF4701TCE	R3, R4	0402	C25900	±1% 1/16W Thick Film Resistors 50V 4.7kΩ 0402 Chip Resistor - SMD ROHS
0402WGF0000TCE	R5	0402	C17168	±1% 1/16W Thick Film Resistors 50V 0Ω 0402 Chip Resistor - SMD ROHS
0402WGF2201TCE	R6, R7	0402	C25879	±1% 1/16W Thick Film Resistors 50V 2.2kΩ 0402 Chip Resistor - SMD ROHS
0402WGF301JTCE	R8, R9, R10	0402	C67234	±1% 1/16W Thick Film Resistors 50V 30.1Ω 0402 Chip Resistor - SMD ROHS
B3U-1000P	SW1, SW2	SW-SMD-2_2.5x3.0x1.6	C231329	2.5mm 1.6mm Button 50mA 12V SMD, 2.5x3x1.6mm Tactile Switches ROHS

9.7 PCB Schematic

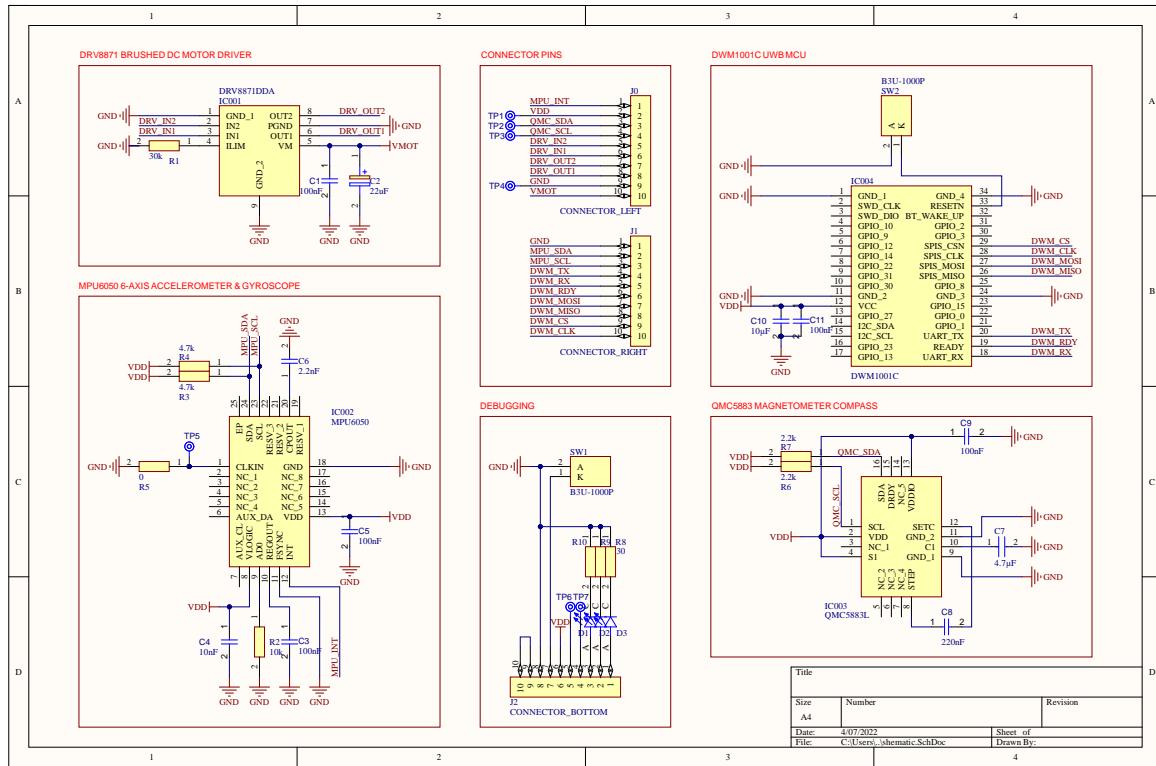


Fig. 34: PCB Schematic (Altium Designer)

9.8 PCB Layers

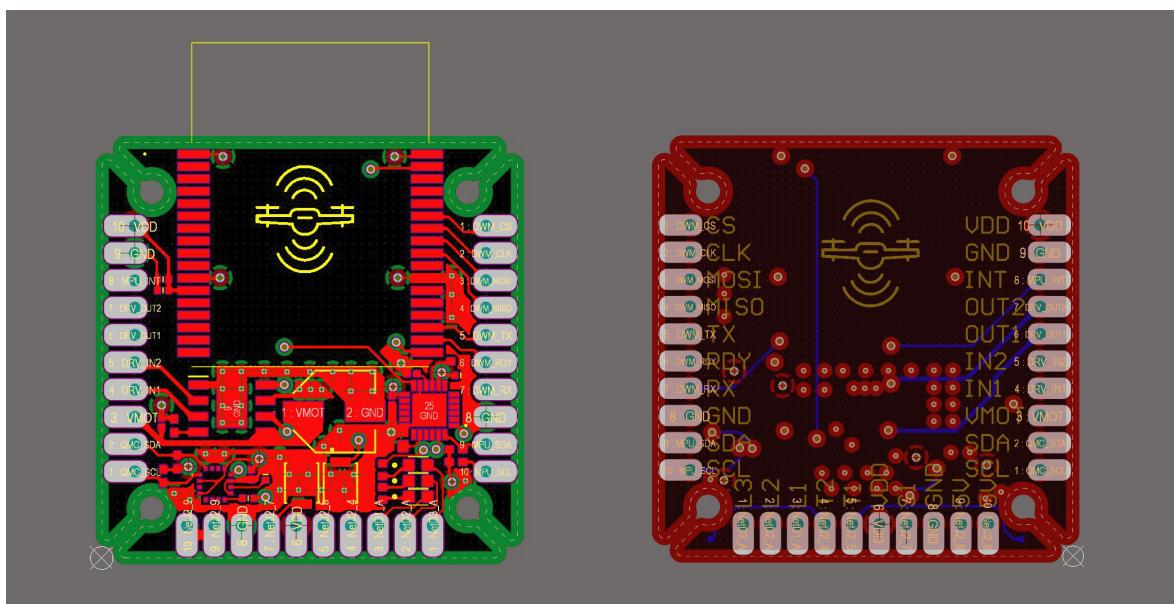


Fig. 35: PCB Layers, (left is Top Layer, right is Bottom Layer)