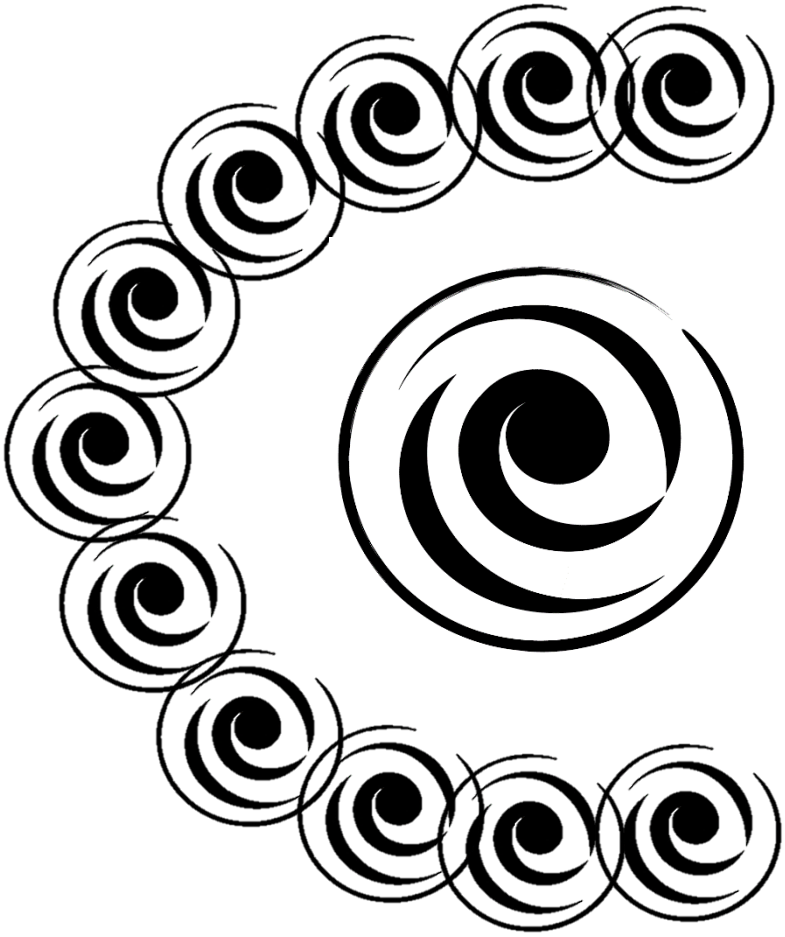


CYCLONE

Ashwin Ahuja, Benjamin Yass, Quentin Guéroult, William Eustace, Monty Evans, Daniel Halstead, James Crompton, Nicholas Palmer, Philip Fernandes

CANSATS IN EUROPE

SECOND INTERIM REPORT

Table of Contents

1 - Team Members

Ashwin Ahuja – Ashwin is one of the co-team leaders and also leading the efforts of the electronics and software parts of the team. He is also managing the team's finances.

Benjamin Yass – Ben is the other co-team leader and leading the efforts of the mechanical design team. He is also shouldering much of the team's organisation, ensuring the various parts of the team are working effectively.

Quentin Guéroult – Quentin is the head of the Outreach team, managing the efforts to inform the greater public about our project. He is also the lead publicist, designing the logos and website.

Monty Evans – Monty is a part of the Software and Electronics team, specifically looking at electronics design for the sensor system.

William Eustace – William is also a part of the Software and Electronics team, specifically leading the writing of the software, as a highly experienced coder in a number of languages (including C++ and C#). Additionally, he carries with him the experience of being the team leader of Team Impulse, and CanSat through this.

Daniel Halstead – Daniel is the Head of Flight Management, attempting to survey the specifications we have produced and determining whether the quadcopter will be flyable. He is very experienced in quadcopter design, having flown them for a number of years, and was particularly involved in the preliminary design of the product.

James Crompton – James is a member of the Mechanical Design team, in charge of the launch procedure, ensuring that the arms deploy as expected, and that the quadcopter begins its flight faultlessly.

Philip Fernandes – Phil is also a member of the Mechanical Design team, but is currently leading research into and creating the algorithms for finding Agricultural Viability.

Nicholas Palmer – Nick is also a member of the Mechanical Design team, with particular responsibilities for researching the most effective manufacturing choices and materials. Additionally, as an experienced mathematician, he will likely be running the data analysis effort, nearer to the launch date.

Figure 1.1: Team photo – from left to right: James Crompton, Nicholas Palmer, Ashwin Ahuja, Benjamin Yass, Daniel Halstead, Philip Fernandes, William Eustace, Monty Evans, Quentin Guéroult

2 - Brief Overview

2.1 - Proposal

The CanSat will fulfil the primary mission of measuring air temperature and barometric pressure and transmitting this data over a radio link, in fact using RF transmission, over the 434 MHz range. It will also contain a number of other sensors, which will also be reported to the base station, including relative humidity, location and acceleration. These, as well as a gyro, will be used for the main part of the secondary mission, that of producing a quadcopter that will open from the size of the CanSat, thus being used to investigate unknown landscapes. For this, both the array of sensors, and a live (likely using FPV 5.8GHz transmission) camera link will be used. The quadcopter will be designed to be able to autonomously move to a set of GPS coordinates, thus possibly being used to return to the launch-site. Finally, we also hope to find the relative agricultural viability of the area, using a predefined algorithm, which could be used on other planets, to find how likely the area could be cultivated for crops for human consumption.

2.2 - Progress Synopsis

Since the first interim report, much progress has occurred in electronics and hardware, with the goal of a full working prototype by Christmas appearing to be likely to be successful. In fact, we have systems flying perfectly, sensor systems transmitting data and camera systems all working, ready to be combined with the hardware, in which the integration has been well planned. Meanwhile, Outreach is also progressing steadily, with a constant stream of updates on multiple forms of communication, both the archaic paper one (through school publications) and on various social networks. The main area which is in need of work is the software section, where though much of the separate systems have been written, they are yet to be combined. While the ground system and autonomous motion algorithms are yet to be even considered, though these can be developed until launch, with the other systems testing happening simultaneously.

3 - Outreach

Cyclone's Outreach has already considerably developed in the planned outreach towards the local community, within the school and towards the wider public. All three strategies are underpinned by the team's website (http://teamcycl.one) that was developed from scratch by the software team alongside a recognisable and simple domain name being secured. The website contains a description about CanSat and Cyclone's entry, an overview of the team, a public folder with documents as well as videos and a blog that regularly discusses the progress of Cyclone. The website acts as the main platform through which people can learn about the team and follow its progress. It also contains links to all other platforms through which people can find out about the project. Additionally, all source code, designs and plans of the different departments of Cyclone are made available to the general public and community through GitHub by simply searching for Cyclone CanSat. A simple, easy-to-remember logo was also designed by the team, which unites all the team's efforts on all platforms.

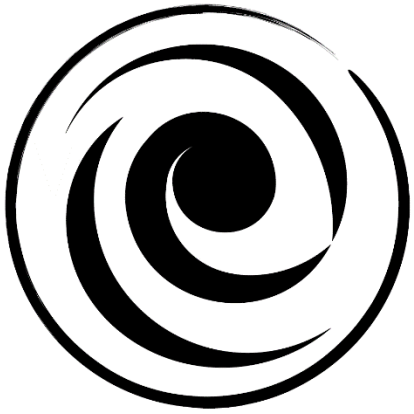


Figure 3.0.1: Cyclone Universal Logo

3.1 - Local Community

Cyclone's leaders and William Eustace (as the team-leader of Team Impulse) gave a talk on the 27th September to the Surrey Explorer's Club (http://www.surreyexplorers.org.uk/), a group of gifted primary school children where the CanSat competition was discussed amidst topics such as rockets, outer space and the uses of satellites as well as how do they function. Cyclone's entry and Team Impulse's European victory were also discussed. Finally, an interesting competition was organised which allowed the children to produce a paper aeroplane which would be released from a specially adapted paper-aeroplane-launching-remote-controlled-helicopter. The aeroplane with the longest flight time would win and this proved a fun activity for all as tactics were also discussed. The team are also planning to give further talks about CanSat to local schools in the area including a planned talk to a nearby preparatory school, Colet Court that has been now been organised for next term. We hope to manage to secure even more schools to talk to in the future, and thus further our purpose.

3.2 - School

Cyclone have already made popular talks at societies such as SPS Space outlining the CanSat competition and Cyclone's entry. Cyclone were also at the Societies Fair and the school's open day where we openly discussed CanSat and our project this year to both pupils and parents. Talks have now been planned for next term at both EnSoc and Physics Soc (the school's Engineering and Physics societies). These will take place during the second week of the Spring term. Additionally, promotion of CanSat has been carried out informally during a number of other societies, notably Robotics Society (run by the co-leaders) and HalleySoc, of which a couple of the members of the team are on the committee. An article on CanSat was published in the school's magazine Black and White that was distributed for free and that can be accessed electronically. The team

hoped that this would further increase publicity for CanSat and our entry as the school magazine is widely read by teachers and pupils alike. Meanwhile, the public school website and magazine reference CanSat in a number of places, talking about Impulse's famous European win last year as well as our continuing mission this year. The PDF version of the article is now also available on Team Cyclone's website to make it available to all and not just pupils of St Paul's School. Finally, plans to have a Cake Sale in school, acting as both fundraising and a piece of outreach, given that it would be accompanied with marketing of the cause of the sale, have progressed, with a tentative timetable now established, pending agreement from the higher management of the school.

3.3 - The Wider Public

Coupled with the website, the team has decided to be present on multiple platforms to further increase awareness of CanSat and Cyclone. Cyclone has a Facebook account (http://on.fb.me/1jTDXtu) as well as a Twitter account where a briefer, but more up-to-date account of the team's progress is available (@SPSCyclone). Android (http://bit.ly/20cqWM7) and Windows Phone Apps (http://bit.ly/1MW0Cfv) have also been made by the Software Team and can be downloaded. An iOS app is currently being developed and will hopefully be available soon on the App Store, however, though it has been developed, it was judged as Apple as targeting too 'niche' an audience, hence was rejected. Again, these apps are to further publicise CanSat. Several videos outlining the progress of the different software, electronic and mechanic teams have been uploaded. The team believes that these videos are extremely important when showing our progress. They are a crucial visual aid for the public when following the team in showing exactly what different members of Cyclone are designing or making, and we hope these continue up until the launch date in March.

4 - Funding

Firstly, Cyclone have approached a number of sources for sponsorship, and have been very successful to date. Newbury Electronics, the owners of PCBTrain have agreed to sponsor the team to the tune of £150, with free PCBs. Additionally, they have agreed to offer their expertise in checking the PCBs that we have sent. To date, only around £50 of the £150 has been used, with the first round of PCBs. Thus, two more revisions of PCBs would easily be possible, in order to make more improvements if necessary, or to fix any inevitable issues that may arise. Additionally, we have been sponsored by HobbyKing, a large online Remote Controlled parts producer and seller who are providing us with a limited amount of free parts. To date, this amount has been around £500, but there is a possibility of more parts if necessary. This amount has allowed us to easily get the best, rather than cheapest components, maximizing the chance of success of the project. Additionally, the chance of needing more products is also low, given that a number of spares of every necessary part has been obtained.

We are also planning a cake sale at school which as well as being a successful fundraising mechanism, will also act as good publicity. Last year, CanSat (under the guises of Team Colossus and Team Impulse) organised a cake sale, raising just under £300, showing the potential success of such an event. This year, we hope with similar preparation and organization, a similar sum could be generated.

Additionally, two members of the team (Ben and Daniel) are Arkwright Scholars, which contributes £200 per person annually to the school engineering department. This money could be used for CanSat if necessary. In fact, our school is also willing to sponsor the project (especially since we are this year's only team) to the tune of a few hundred pounds if necessary, with money coming from the Engineering budget. However, we hope to keep these costs to the minimum necessary, by continuing to seek corporate sponsorship and through a successful fundraising event in school. Though some costs have occurred since the last progress report, with the requirement of a couple of components which were otherwise omitted accidentally during previous purchases, as well as due to failures in other components such as the Control Board. From here on out, we envisage no further electronics or hardware costs, bar the nominal costs of 3D printing, due to the vast numbers of spares purchased.

Section	Expected Cost / Value	Costs to date
Outreach	£60	£50
Hardware	£26	£15
Electronics Components	£500	£500
PCB Manufacturing	£0	£0
TOTAL	£586	£565

Figure 4.1: Basic breakdown of costing – for more detailed breakdown, see Figure 4.3

However, despite the costs being over the acceptable limits of the CanSat cost per unit, it must be noted that this includes a number of spares for each component, in fact, we expect the cost to reproduce our CanSat (including the value of our sponsorship) to be much lower:

Item	Cost
Electronics Components	£290
PCB Manufacture	£50
Hardware	£30

TOTAL	£400

Figure 4.2: Breakdown of cost of a single Can.

Type	Name	Quantity	Cost	Sub-Total
Battery	Turnigy Nano-Tech 850mAh LiPo	6	£ 6.56	£ 39.36
Board	Hobbyking i8 Control Board	1	£ 10.98	£ 10.98
Motors	Turnigy Multistar Outrunner V2 Motors	8	£ 7.46	£ 59.68
Escs	Turnigy Multistar 20A Slim ESCs	8	£ 8.88	£ 71.04
Servos	Turnigy Analog Nano Servos	10	£ 2.55	£ 25.50
FPV TX	Hobbyking FPV Transmitter	3	£ 13.35	£ 40.05
Battery	Turnigy 2200mAh battery (for base station)	2	£ 6.39	£ 12.78
FPV Receiver	SkyZone FPV receiver	2	£ 12.77	£ 25.54
Antenna	Polarized SMA antenna	2	£ 3.19	£ 6.38
SMA Wire	SMA wire	5	£ 1.06	£ 5.30
OSD	Hobbyking OSD	2	£ 9.28	£ 18.56
FPV Camera	Mini FPV Camera	3	£ 18.84	£ 56.52
Bags	Lipol Bags	6	£ 1.34	£ 8.04
Radio TX	Orange Radio Transmitter	1	£ 41.59	£ 41.59
Radio RX	Orange Radio Receiver	3	£ 6.97	£ 20.91
LiPo Charger	Hobbyking LiPo charger	2	£ 7.98	£ 15.96
FPV RX	**Quanum Complete FPV Bundle Set**	**1**	**£ 46.01**	**£ 46.01**
Propellers	**Gemfan Multi-Rotor Prop Set 50mm**	**15**	**£ 0.80**	**£ 12.00**
Control Board	**OpenPilot CC3D**	**2**	**£ 11.18**	**£ 22.36**
Control Board	**OpenPilot CC3D Atom**	**3**	**£ 14.74**	**£ 44.22**
Wires	**Servo Wires**	**3**	**£ 1.00**	**£ 3.00**
Wires	**XT60 Wires**	**5**	**£ 3.00**	**£ 15.00**
Connectors	**3.5mm Connectors**	**5**	**£ 1.20**	**£ 6.00**
MAIN SENSOR BOARD				
Male Headers	Break Away Headers - Machine Pin	2	£ 2.95	£ 5.90
Resistor	Panasonic 75kOhm Resistor	100	£ 0.01	£ 0.90
Capacitor	Murata 100muF capacitor	10	£ 0.98	£ 9.80
Resistor	Bourns 10k SMD 0805 Resistor	50	£ 0.01	£ 0.44
Resistor	Bourns 10k SMD 0805 Resistor	50	£ 0.01	£ 0.44
Motor Driver	Texas Instruments DRV 8833	10	£ 1.78	£ 17.80
Capacitor	TDK 2.2muF	100	£ 0.05	£ 5.00
Capacitor	Kemet 0.01muF	10	£ 0.18	£ 1.84
MCU	Teensy 3.2	4	£ 13.02	£ 52.08
Humidity Sensor	HYT-271	5	£ 22.09	£ 110.45
Pressure Sensor	MS5637	10	£ 1.69	£ 16.90
IMU	SparkFun LSMDS1 Breakout	1	£ 16.28	£ 16.28
GPS:				
GPS Module	GP-2106	2	£ 32.59	£ 65.18
GPS Breakout	Sparkfun GPS Evaluation Board (GP-2106)	2	£ 6.49	£ 12.98
GPS Module Connector	Interface Cable GP-2106	5	£ 0.98	£ 4.90
RF	**Hope RFM98W**	**5**	**£ 5.99**	**£ 29.95**
Micro SD Breakout	Sparkfun OpenLog	1	£ 16.28	£ 16.28
POWER DISTRIBUTION BOARD				
5V Voltage Regulator		10	£ 2.00	£ 20.00
PCB Building Costs		3	£ 50.00	£ 150.00
Can Building Costs	3D Printing	4	£ 5.00	£ 20.00
	Grub Screws	2	£ 3.00	£ 6.00
Outreach Costs	Website	12	£ 5.00	£ 60.00
SUB-TOTAL	**£ 1,229.89**			
HobbyKing Sponsorship	£ 606.78			
PCBTrain Sponsorship	£ 150.00			
Cake Sale	£ 300.00			
TOTAL	£ 173.11			

Figure 4.3: Breakdown of costs (made using Microsoft Excel) – new purchases are highlighted

5 - Mechanics

5.1 - Aim

The goal of the mechanical design team is to design the physical structure of the Can such that it takes the shape of a Can of maximum size 66mm diameter, and 115mm height during launch, yet can in flight transform itself into a successful quadcopter.

For most challenges involved in accomplishing the task above, any improvement in any aspect of the design will be beneficial to the overall performance of the Can, unless this is done to such an extent that we begin to suffer from lack of space within the Can. When creating the design, we found that this issue became most acute when trying to maximise the stability of the quadcopter, as this resulted in using up large amounts of space which is needed for other components.

5.2 - Parachute

After several brainstorming sessions of thinking about mid-air parachute release mechanisms and other techniques of using a parachute at first and then flying around after utilising it; we became confident that any benefit a parachute could bring to the launching of our quadcopter, would be hugely overwhelmed by the issues of the parachute becoming entangled in the rotors of the quadcopter. Additionally, we would be forced to develop a parachute release mechanism, adding more unnecessary complexity to both mechanics and electronics. Thus, we decided not include a parachute in our designs. However, if we were to get through to the European Competition, where the flight time and height would be much increased, we acknowledge that a parachute system may be necessary.

5.3 - General Design Creation

The trade-off between space and stability was the deciding factor for one of our high impact decisions on the overall setup of the quadcopter – the orientation that it would fly. The outcome of this decision was to go for the vertical design, as although the horizontal one would be more stable, it would result in a complete lack of space for many components.

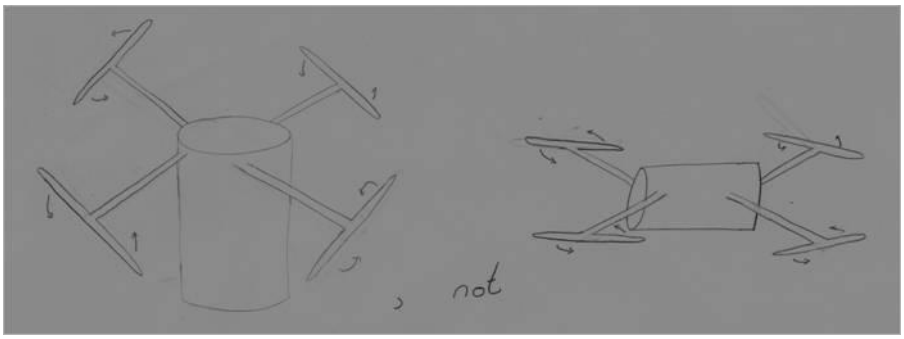


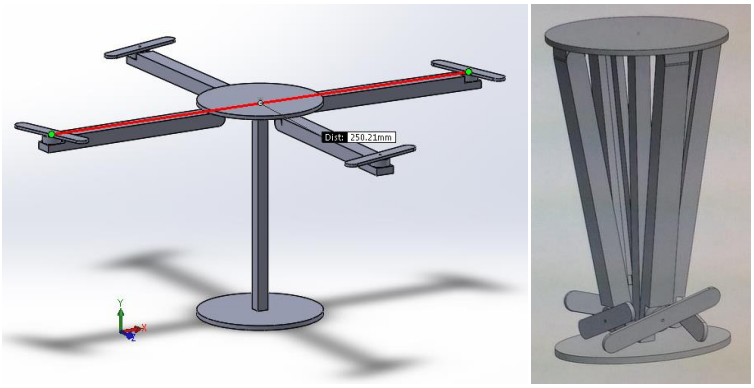
Figure 5.3.1 - Can Orientation Decision

Throughout the design of the Can many decisions were strongly influenced by this space / stability trade off. For example, the exact length of the arms:

This initial vertically orientated design (Figure 5.3.2), where arm length was maximised to increase arm stability (as the greater the distance between diagonally opposite rotors when unfolded, the greater the quadcopter's stability), had to be altered to ensure sufficient space was left inside the Can for other

components (Figure 5.3.3). Due to this change the diagonal distance decreased by about 35mm (see Figures 5.3.2 and 5.3.3).[1]

Figure 5.3.2



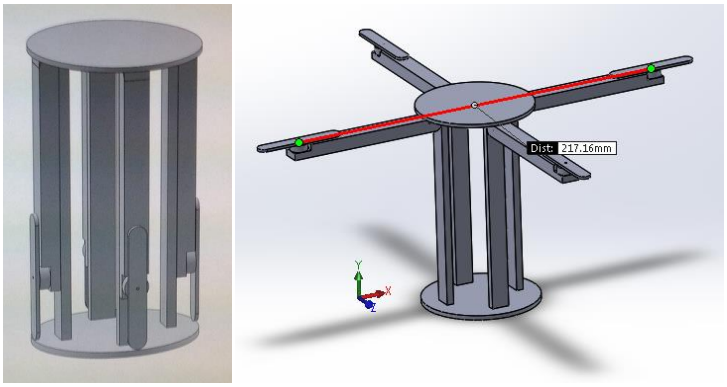


Figure 5.3.3

Other factors of the quadcopter's design were also strongly influenced by the minimal space available.
E.g. The Can's overall structures that hold it together had to be cleverly redesigned to maximise space for other components (From Figure 5.3.4 → Figure 5.3.5 (Figure 5.3.6 is the same as Figure 5.3.5 except with the arms unfolded to show the wall)):

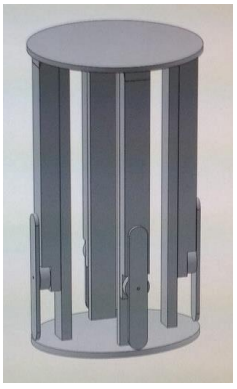


Figure 5.3.4

Figure 5.3.5

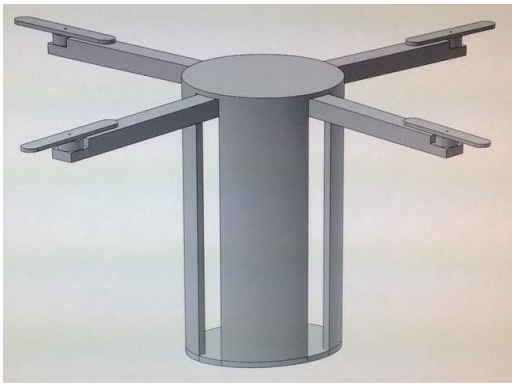


Figure 5.3.6

[1] All designs are made using SolidWorks 2015 – made by Dassault Systems – http://solidworks.com

Another decision we made with reference to the overall design of the Can was deciding how to stabilise the arms when unfolded – hence we experimented with various solutions.

Figures 5.3.7 shows a possible solution we came up with to increase arm stability

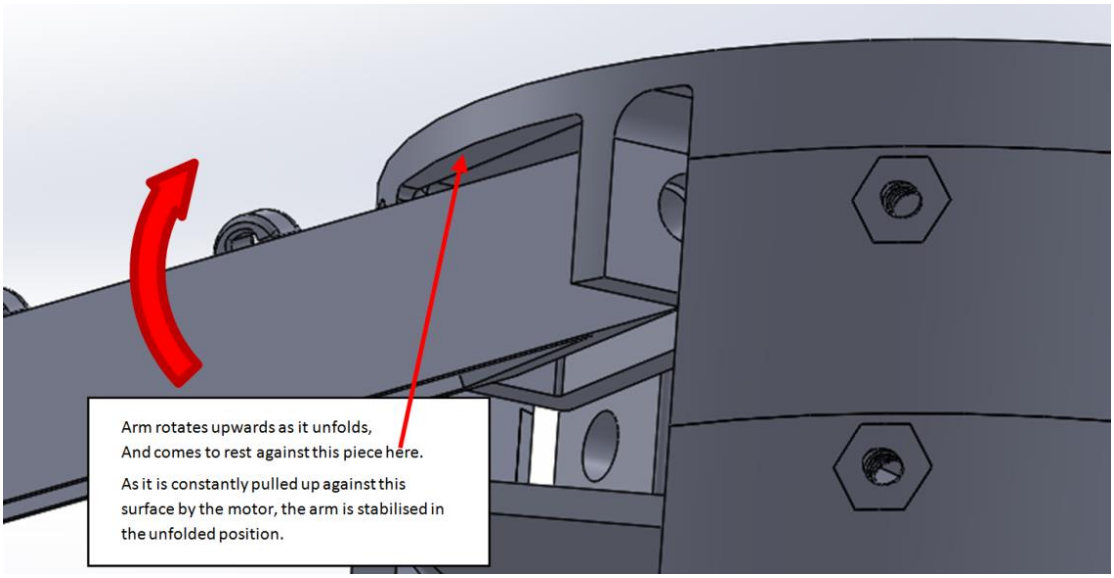


Figure 5.3.8 shows another possible solution we came up with to increase arm stability

We made a decision to use the solution shown in Figure 5.3.8 as we thought it would work sufficiently well, yet would be simpler and so more reliable than the supports shown in Figure 5.3.7.

The final major part of designing the general Can structure was working out layer spacing. This we did by modelling each component in the Can and working out the best way to fit them all in it. From this we then created layers at specific heights within the Can to facilitate easy mounting and organisation of components during assembly and integration of electronic components within the Can. The plan for this can be seen in Figure 5.3.9, and the consequent layer creation can be seen in Figure 5.3.10.

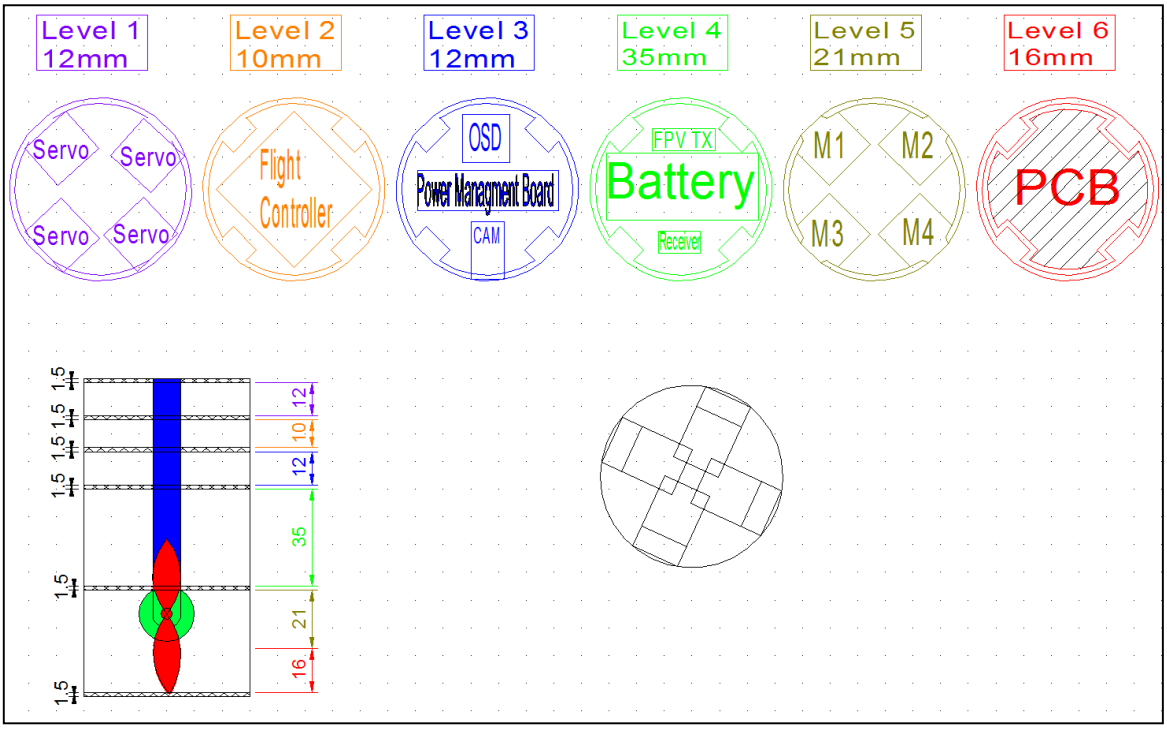


Figure 5.3.9 [2]

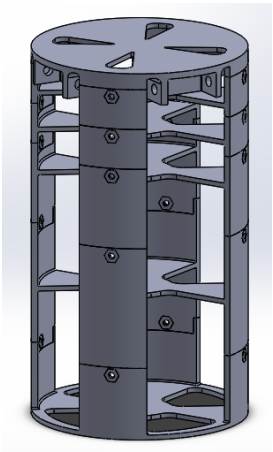


Figure 5.3.10

Before beginning the detailed refinement of the Can we decided to fillet (round) every edge that would be under stress (i.e. almost all of them) – shown in Figure 5.3.11. This strengthens the Can as it means any force acting to snap a corner is spread across an arc of material instead of being concentrated at a single line along an edge.

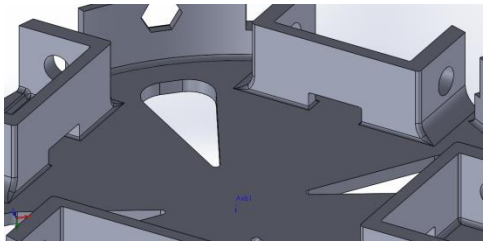


Figure 5.3.11

[2] 2D design made using 2D Design V2 - http://www.techsoft.co.uk/products/software/2D_Design_V2.asp

5.4 - Refinement of General Design

Now that the general design of the Can was complete, we begun work on the designing of the specific component parts of the Can.
This began with designing a way of being able to dismantle and reassemble the stack easily. We went for a modular system which utilised teeth like fittings to connect the layers securely together (as shown in Figure 5.4.1). Through the teeth can be seen holes which are for pins, so that we can secure the layers together.

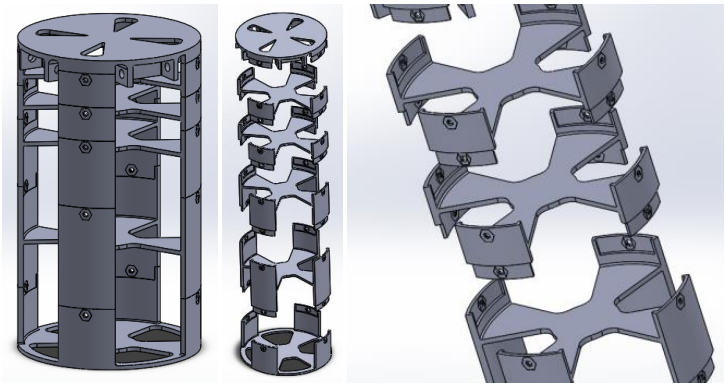


Figure 5.4.1

All pieces of this modular system are very similar apart from the base piece which simply does not have connections for a layer below it, and the top piece (Figure 5.4.2) which does not have connections for a piece above it, but does have hinges for the arms built into it, as well as the arm stabiliser as seen in Figure 5.3.8.

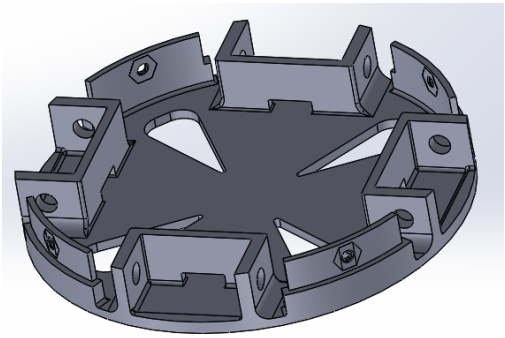


Figure 5.4.2

Furthermore, we designed the arms in such a way that they would be strong, hold the motors securely and also contain cavities in their underside for the ESCs so that they would be efficiently stored away (see Figure 5.4.3). Note that the bobbles on top of the arms are rings built into them for the wire (which will be used to unfold the arms) to run through. This will allow the wire to run along the top of the arm without it becoming a danger to the rotors.

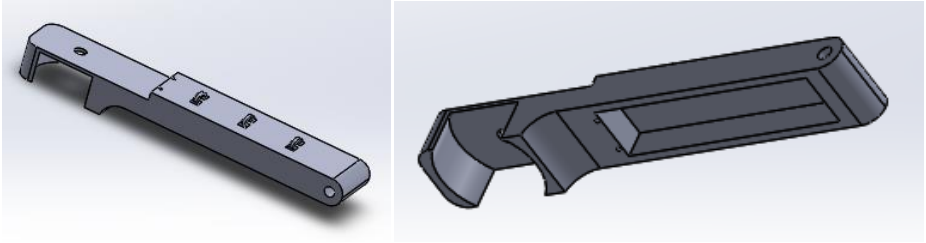


Figure 5.4.3

When constructed in SolidWorks the model looked as follows:

Figure 5.4.4

5.5 - Manufacture, Testing and Further Refinement

To test our CAD we 3D printed this latest design. While normally this would have been fairly easy, as our school has a 3D printer, it was unfortunately broken at the point we wanted to 3D print. Therefore, we decided to outsource the printing and contacted a local hobbyist group who printed the pieces at a low cost for us.

When these parts arrived we set about testing them and discovered some issues:

- The motor housing we had designed appeared to barely hold the motor;
- The layers did not slot together smoothly.

Hence we redesigned the motor housings to provide greater purchase on the motors, using secure screw fittings:

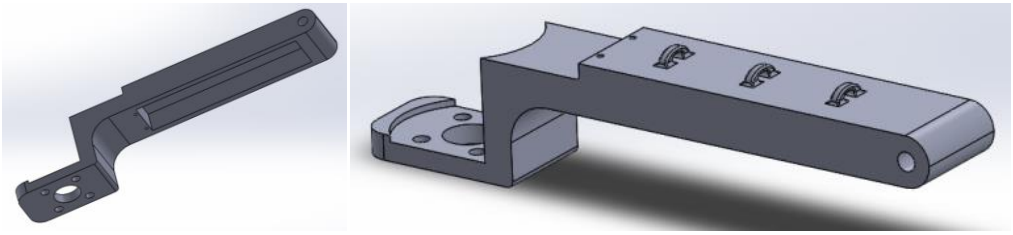


Figure 5.5.1

By this point the school's 3D printer had been fixed so we 3D printed an arm to see if the problem had been solved and we believe it has – the motor feels secure, however we will not know for sure until we carry out tests with the motor running at full speed (which we plan to do in the next couple of weeks). In addition, when we do this testing we also will be looking to test the hinge integrity.

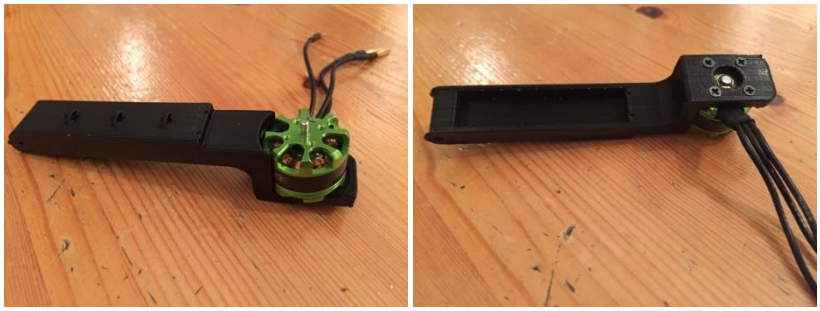


Figure 5.5.2 – motor screwed securely into arm

Furthermore, while the layers had fitted together perfectly in SolidWorks, due to the inaccuracies of the 3D printer we are using, they did not fit together once printed. The issue arose at the end of the thin tabs which slot into a matching section in the layer above (Figure 5.5.3). The problem was that when the 3D printer printed this tab, when the printer head rounded the corner, it deposited a blob of material on that corner as it could not make such a tight turn. The tight turn was due to the thinness of the tab. Hence, on every layer of material that the printer built up, additional material was deposited on this corner – and every other similar corner on the piece. Hence the width of this tab was increased meaning that it no longer slotted into its matching section on the next layer. Hence, to solve this, we thinned the tab and the matching section which it slots into by 0.4mm (see Figure 5.5.4 (before) and 5.5.5 (after)). Thus even when the pieces became wider due to the inaccuracies of the 3D printer the pieces still slotted together.

Figure 5.5.3

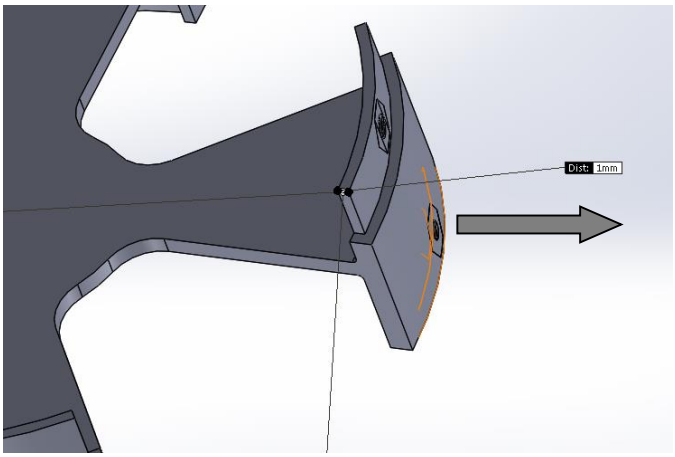


Figure 5.5.4

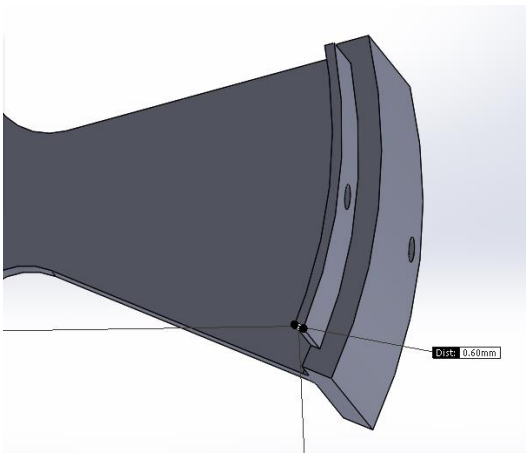


Figure 5.5.5

Hence, the layers can easily fit together, as shown with the top layer in Figure 5.5.6.

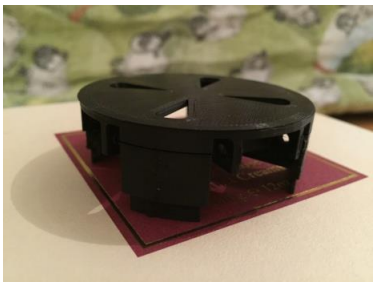


Figure 5.5.6

When constructed in SolidWorks including those changes the model now looks as follows:

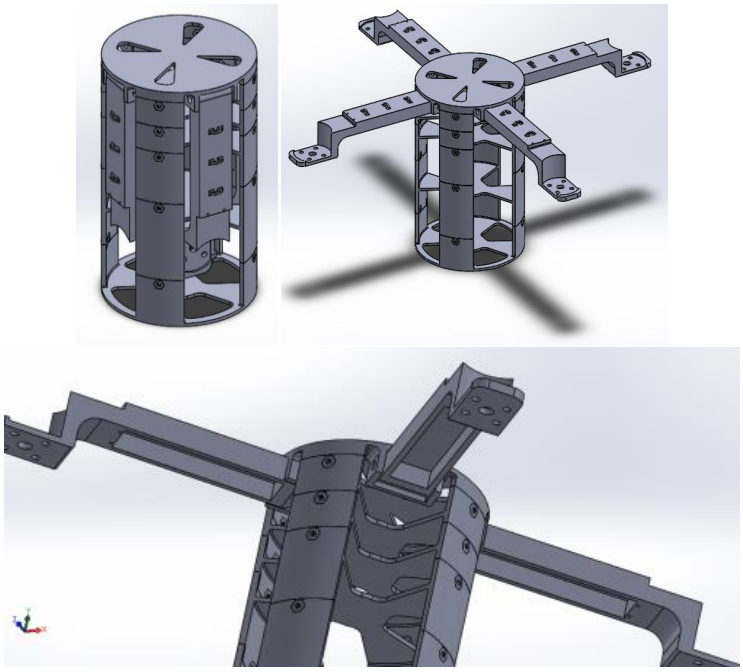


Figure 5.5.7

5.6 - Design Alteration

Although this fully functional design fulfilled all our requirements, we believed we could still improve it. This was because the tabs which connected the layers together were very fragile and the small knocks that the pieces were receiving in the day to day handling of them had already begun to show – one of the pieces had a broken tab and two others were clearly weakened. Hence, we decided to see if we could redesign this aspect of the Can, without affecting any other part of it.

The new design we came up with did just this. It consists of four vertical posts (the back and front of one of the posts can be seen in Figure 5.6.1) which replace the vertical extensions that protruded from the plates in the earlier design. These posts, situated at the circumference of the Can, fix onto plates at different heights, holding the plates securely in place at these heights. The plate spacing is the same as in the previous design, for the same reason – when we modelled each component in the Can this spacing optimised their packing and allowed all the components to be mounted securely and organised coherently. Each plate (Figure 5.6.2) has four teeth in place of their four protrusions which previously were used to link the layers together. Each tooth fits into a matching slot in one of the posts, and these teeth are fixed to the posts with M2 nuts and bolts which act as pins.

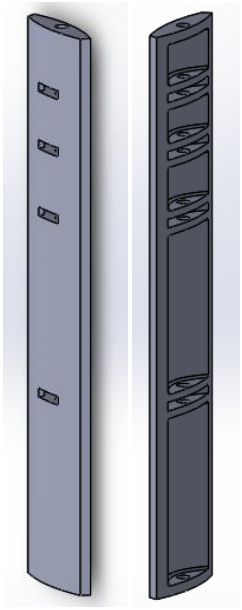


Figure 5.6.1

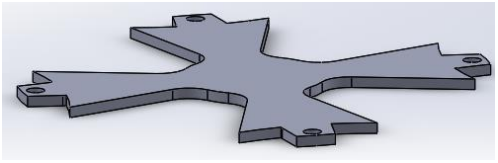


Figure 5.6.2

This new design has many advantages over the old one, and only a single disadvantage. The disadvantage is that on each layer there is less room than before. This is due to the fact that the horizontal cross-section of the posts is greater than that of the previous interlayer attachments. However, this reduction in space is so minimal (see Figure 5.6.3) due to the efficient design of the new system that this does not affect whether components fit into their allocated space on each layer, as previously planned (see Figure 5.3.9).

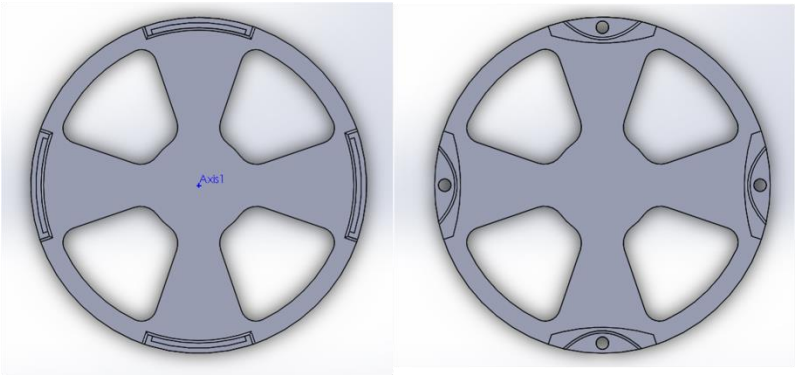


Figure 5.6.3 – Before vs. After

The benefits of the new design include:

- The interlayer connections are less fragile and stronger with the direct result that the Can is now too.
- The pieces are more 3D printer friendly. This is due to the extensive reduction in the number of thin protruding structures extending from the pieces – there is now nothing less than 1mm thick whereas before some sections were 0.5mm thick.

- An unforeseen benefit is that now there are only 4 designs of pieces – the top piece, bottom piece, plate and post; whereas before there were six – each of the six layers was slightly different.
- The new design still allows for easy access to components on each layer when the Can has been assembled. This is because all the fixings holding each post in place are mechanical, and the removal of one post allows access to all the layers.

The new design when assembled can be seen below:

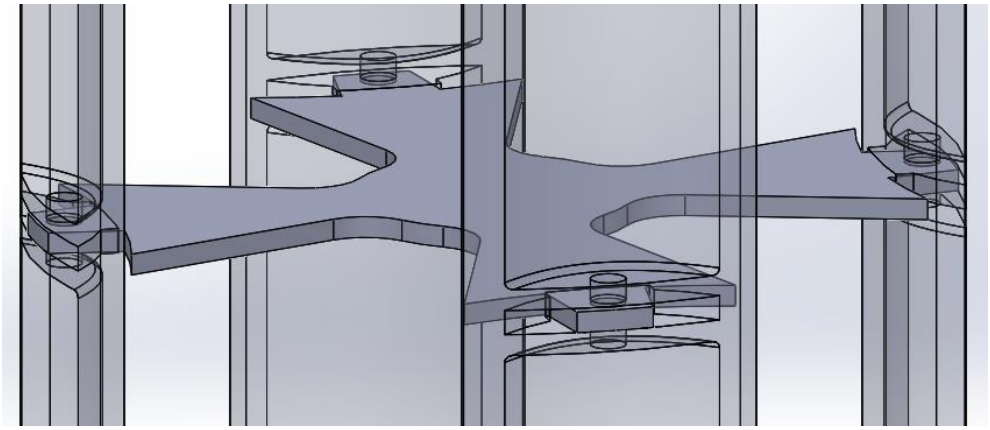


Figure 5.6.4

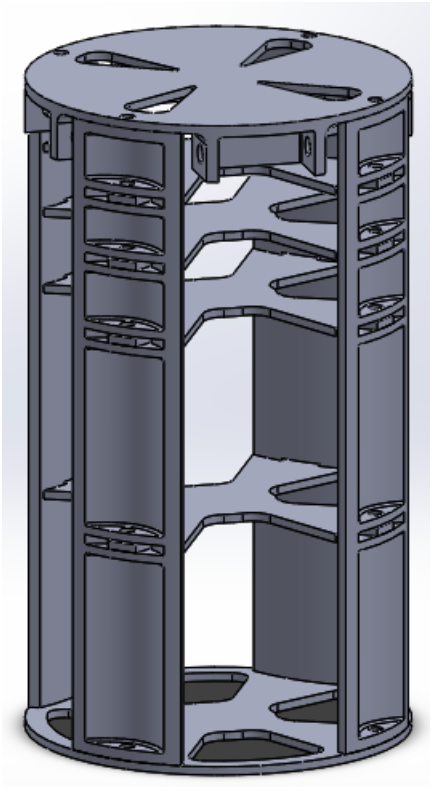


Figure 5.6.5

5.7 - Materials

For our testing so far we have been using some PLA and some ABS as these are cheap materials which work well in the school's 3D printer. For the testing of whether pieces are the correct size to fit components within

them, these materials are suitable. However, we are also conducting research into stronger materials, in case ABS and PLA fail in our upcoming testing.

5.8 - Arm Unfolding

Whilst originally we intended to use to raise each arm on a separate servo attached to the top plate, as seen in Figure 5.3.9, we quickly decided that this would not give a robust enough connection between the arms and the top plate. This is because the servos would need to be small. Therefore, the axel onto which the arm would be attached would be about 3mm in diameter and 4mm deep. This is too small to give a link that would endure the launch and then support the can's weight during flight.

Therefore we decided to redesign this system. During this process we realised that we would be able to attach the arms are more securely to the Can's body, and we would be able to free up more space within the Can if the arms were controlled by a motor which would wind up wires attached to and which ran along the loops on, the top of each arm. This principle we have tested with some of the 3D printed parts, with huge success (see Figure 5.8.1) – however for this test we were manually pulling on the wires to simulate what would, in the actual Can, be done using a motor to shorten the lengths of wire passing around the arms (as discussed in conjunction with Figure 5.4.3), thus lifting the arm structure up.

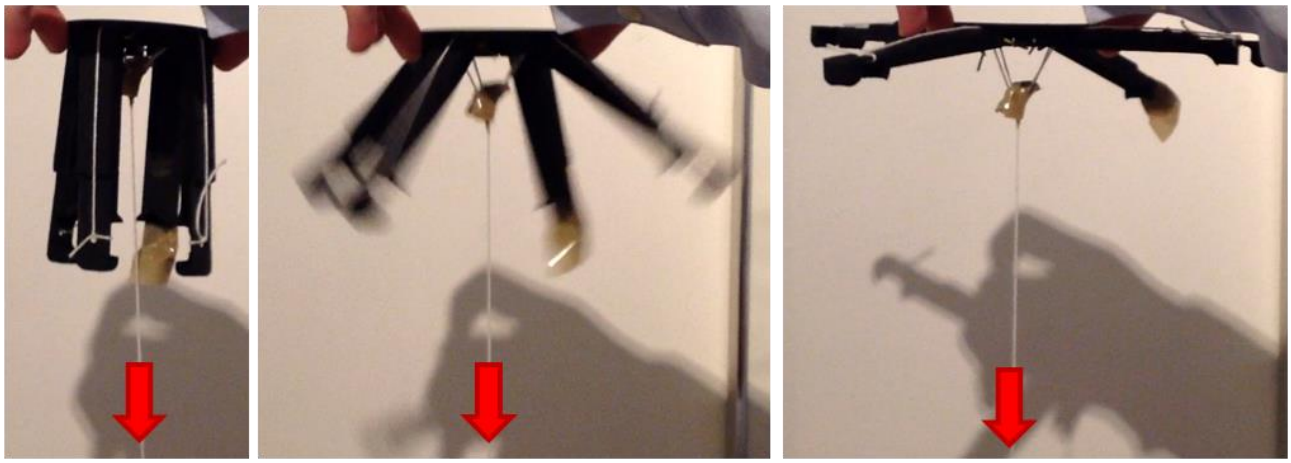


Figure 5.8.1

5.9 - Specific Details of Mechanism for Opening Arms

To raise the arms into their flight position we have decided to use a continuous piece of thin brass wire attached from one arm through the top plate to the opposite arm. In order to fully raise the arm piece, the wire must be shortened by 10mm to 16mm (measured from the 1st test assembly of the can). To do this we will wind the wire around a motorised spool.

Initially the idea was to wind up the wire on a spool attached to a dc motor. The smallest suitable motor was Como Drills Brushed DC Motor, which is 15mm long. The left picture in Figure 5.9.1 shows how this would not fit well into the design vertically, whereas if placed in horizontally one of the pieces of continuous wire would need to come on and off the spool in line with the axis of rotation, which would make it likely to jump and come off.

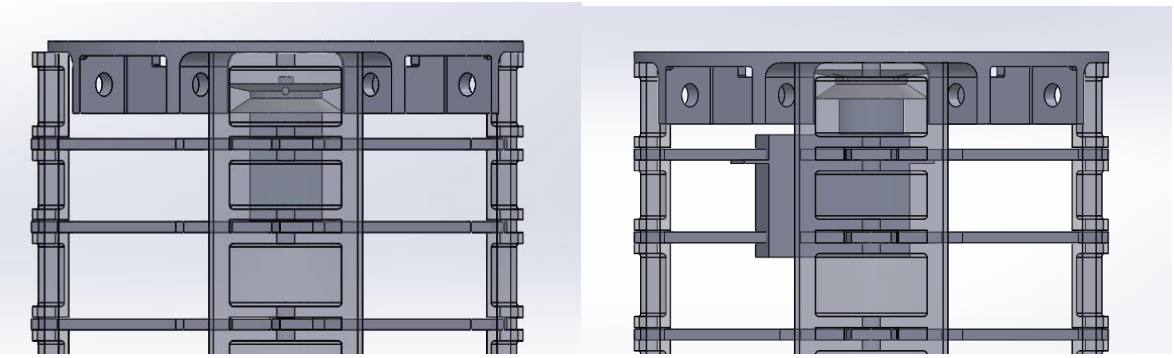


Figure 5.9.1

The right picture shows a similar arrangement for the Turnigy Nano Servo. This is much deeper and would pass through 2 layers, taking up part of 3 cavities, making this arrangement unworkable.

Therefore the best arrangement would be a horizontal servo. To get around the issue of one of the wires jumping two servos and spools were needed- one for each wire. This enabled us to utilise the ability of servos to lock with minimal impact on volume.

In the initial design we created structures (of the exact size of the servos) in the underside of the top plate to mount the servos. For this we designed pillar-like constructions for the servos to attach to and a raised platform for them to sit on so as to leave space underneath the servos for the wire to pass from one side of the top plate to the other. Figure 5.9.2 shows the design which we prototyped in the 3d printer.

Figure 5.9.2

The issues with this design was that there was no tolerance to insert the servos without breaking off two of the pillars first. Another problem was that the holes were too small to allow the antennae through the top plate.

The spools are also shown in Figure 5.9.2. They have a minimum diameter of 8mm giving a circumference of 25mm which will easily provide enough distance to raise the arms within the limit of 360° rotation for these servos. There are also 12mm diameter plates on either side of the smaller diameter part to ensure the wire does not jump off the spool.

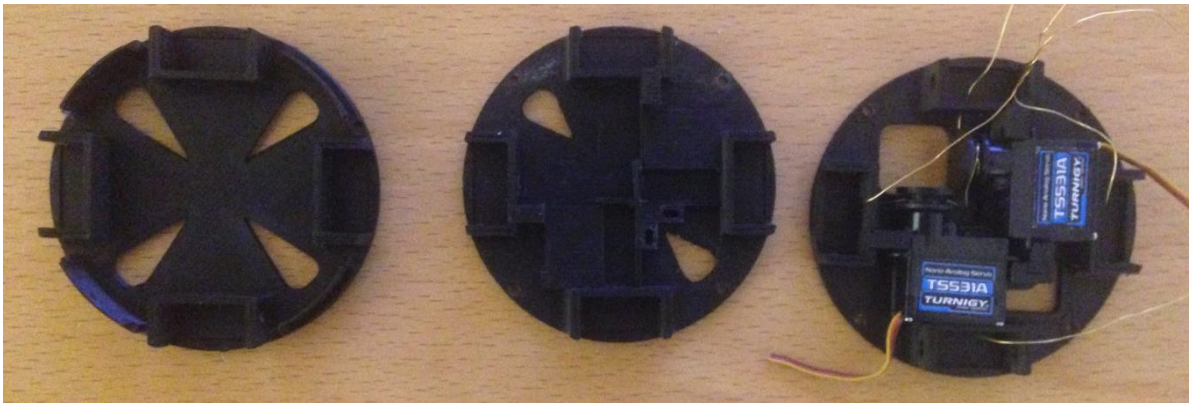


Figure 5.9.3

Figure 5.9.3 shows the evolution of the design of the top plate over the development period. The current design (Figure 5.9.4) has much larger holes to allow antennae through, while still giving a strong base for the servos to attach to. The pillars have been moved to allow a looser fit for the servo and the pillars that snapped last time have been joined together to make them stronger. The wire attaches to the spools through a small hole through their centre along a diameter. The spools then attach to the servos through friction fit joints, which when their design has been finalised, will be secured with adhesives.

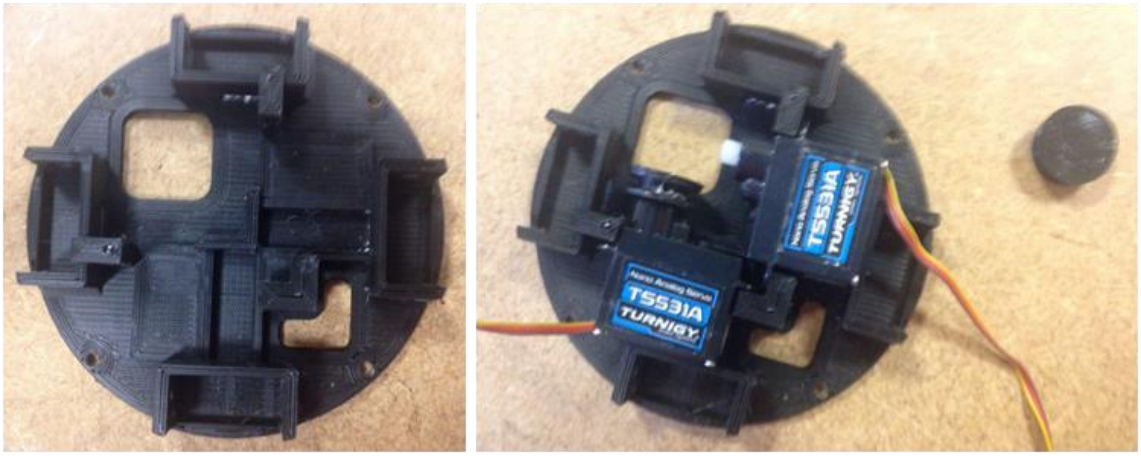


Figure 5.9.4

5.10 - Latest Design

Here (Figures 5.10.1 and 5.10.2) are our pictures of our latest design. It has been 3D printed on the school's 3D printer in PLA. In Figure 5.10.2 the servos are not in place to hold the arms open so the wire is manually being held taut.

Figure 5.10.1

Figure 5.10.2

5.11 - Propellers

Most of the information about propellers is decided by the existing mechanics of the product. It is clear that we want to maximise the size of the propellers, in order to ensure that the maximum air can be pushed downwards, thus increasing the lift of the can, maximising the efficiency of the system. However, there is inevitably a compromise between the length of the arms (which when increased, increases stability of the craft) and the length of the propellers, since the maximum of the arms + half the propeller diameter must equal 115mm or less, the maximum height we can use, as per the ESA regulations. To start off with, through carrying out research into various specifications, we determined that the minimum propeller size was 50mm, hence we started work with that. However, there is the possibility, that, if testing reveals that the lift is not great enough, we will reduce the length of the arms and increase the size of the propellers. The next decision was how to obtain propellers, with there being two main options, the first buying them, the second making them. If we were to make specific propellers for our purposes, it would be clear that we would need to make use of Injection Moulding as the only suitable mechanism for producing the precise shapes required for the propellers. However, the process is painful, requiring the creation of a mould, and very expensive. Unlike 3D printing, we do not have the facilities to carry out Injection Moulding at school, and outsourcing the production of the propellers to those who have the correct equipment would be very expensive, given the low volume required. Injection Moulding only becomes viable when the volumes increase, since the starting costs (that of the mould and equipment) which must be overcome, are very high, hence would not be possible for us. Additionally, we would not even have just one type of propeller, in fact requiring both CW and CCW propellers, since by the specifications of a X-type quad, two motors spin clockwise and two spin counter-clockwise (hence CW and CCW), effectively doubling the already high costs. Therefore, we decided to purchase ready-made propellers of 50mm size. Additionally, buying propellers means we could easily buy other sizes later if they are required. In fact, these would also be free, since we were sponsored by HobbyKing, the company from which we chose to obtain the propellers.

5.12 - Testing of the Mechanics

So far there has been rigorous testing of all individual components (i.e.: the 3D printed parts):

- We have tested and refined each piece such that now all of them fit snugly together.
- Each piece has been stress tested to ensure it can take reasonable stresses – this we did by bending, flexing and compressing each piece by hand until they began to show signs of snapping. Based on how hard it was to reach this point we decided whether the pieces would be strong enough. The only piece which we decided needed improving was the part of the arm which the motor attached to (Figure 5.12.1). This piece we have since increased the thickness of. Luckily increasing the thickness of this piece did not intrude on the space allocated to any other component.

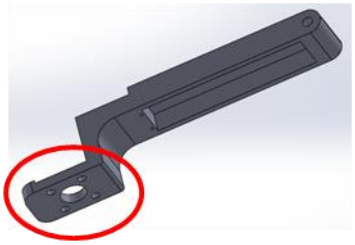


Figure 5.12.1

- We have also tested how well the housings for specific components hold the intended components. The description of the refinement of the arm design details how we encountered difficulties and resolved them for the motor housing, and the description of the specific details of the mechanism for opening arms details how we did the same for the mounting of the servos for the deployment of the arms.

In addition to this we have begun testing how well the different components perform when as part of a subsystem:

- We have fixed all the pieces together and performed stress tests as before, encountering no problems. We hope to improve these tests by fixing the actual motors in place and running them – further details of this can be seen in the integration of electronics and software within the mechanics section.
- We have also, as seen in Figure 5.8.1, tested the arms opening by the tightening of wire; we hope to test this system using the actual servos soon.

All further testing is non-inter-mechanical-component, and so is detailed in the integration of electronics and software within the mechanics section.

6 - Electronics

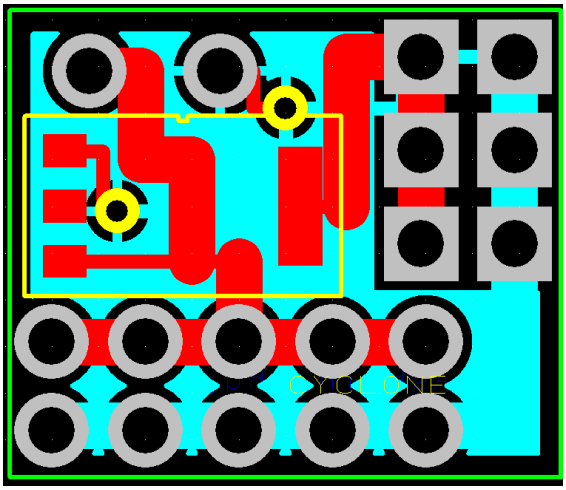
The software and electronics were largely split into three parts, which would not be interconnected except for sharing the same power source, as we wanted to ensure a fail-safe that if one system failed, the others would continue to work. Additionally, to provide more flexibility, the system that would be used in order to open the arms, with the two servos, will be connected to the sensor system. Additionally, this means we could trigger the system to work with changes in altitude if we wanted (though this may unreliable, and so too risky).

6.1 - Flight System

For the quadcopter system, the first thing we decided was the exact parts we would choose. The actual types of parts we required rather decided themselves, since we did not have the space to add any unnecessary parts. Thus only vital components were chosen. The first main decision that we made was that we would buy as many parts as possible for this system given the high complexity required to keep a quadcopter in flight, and the lack of space and time for parts that we might make ourselves. Additionally, parts such as ESCs and the Control Board, which we could theoretically replicate were very good, reliable and cheap, and since we were sponsored by HobbyKing, became free. Below, is the list of the exact parts we chose and why we decided that they were the right choices. Since the first progress report, there was one major change, the Control Board, which was required, simply due to the fact that the old control board was ineffective, not providing enough flexibility. Additionally, this was backlogged with no prospect of it returning to stock, thus if we broke the one board we had inherited from a team member, we would have no control board. Additionally, by looking outside of this board, we in fact managed to find an equally easy to use, but far more effective board.

Battery – Turnigy Nano-Tech 3s (11.1V) 850mAh[3]	The power supply system is vital to the entire project, since any failure in the system could prevent the operation of the entire CanSat. We need to thus ensure that all components of all systems receive a safe voltage. Additionally, the battery needs to be very small, given the lack of space that we have in the can. However, the smaller the battery the lower the capacity of the battery. In order to maximise the capacity per unit area, the use of a LiPo battery is the best choice. Though it is a very powerful battery, there are a few issues related to the safety of the battery, since LiPos are liable to explosions if they are overcharged, dis-charged or indeed short-circuited. However, certain batteries, such as the battery that we have chosen to use has specific protection built into the system to ensure that there is no excess in voltage or indeed current, which could be damaging. Though this could have been achieved using a Zener Diode and a collection of transistors, or indeed specific Integrated Circuits, it would have taken up some space and have probably been less effective than the system integrated in the battery. Given that the motors have to lift 370g of the can, the majority of motors we could use appeared to require the use of a 3s (11.1V) battery, so we chose to use this battery, since it was the one which best fit into the

[3] http://bit.ly/1GAFMWi

	existing mechanics of the can. Additionally, using our existing expertise and extensive online research we have found that this should be able to provide around 5 minutes of flight time, our aim, while also powering the sensor system. However, also through testing, we have found that the camera system takes a lot of power, so we hope to include a high-current logic-level MOSFET to control the camera remotely, so that we could keep the camera off when the can is idle. In order to distribute all the power according to the wiring chart (see Figure 6.4.1), we have designed a power distribution PCB, (see Figure 6.1.1) in order to allow power to go to the correct places. Additionally, on board we have a 5V voltage regulator, which will provide the correct voltage for the motor which opens the arms of the quadcopter. We have chosen to rely off the voltage regulator of our microcontroller (the Teensy 3.2) to supply the 3.3V required for the rest of the sensor system.  *Figure 6.1.1: Power Distribution PCB (designed using DesignSpark PCB[4])*
Motors – Turnigy Outrunner v2[5]	For the motor choices, we were forced to consider two important aspects, the size of the motors and their power. In order to provide a little leeway, we wanted to ensure that the motors we chose had at least a thrust of 400g, thus it could lift 400g. This meant that the four motors together had a thrust of over 1.6kg, more than enough to comfortably lift the 370g of the CanSat and accelerate it upwards. In order to calculate the thrust, we used online calculators. We soon found out that we would not struggle with thrust, with the majority of motors meeting our specification, but

[4] http://bit.ly/1AWvlGh
[5] http://bit.ly/1PUlKbW

	would struggle with size. The Turnigy Outrunner v2 motors were selected because they were the smallest motors (15.5g) we could find but one of the most powerful, with a KV specification of 1900RPM/V. Though normally very expensive, the sponsorship by HobbyKing (from whom we purchased the motors) meant that they were the best choice.
Control Board - OpenPilot CC3D Atom[6]	This specific control board was chosen due to its simplicity to use, as well as it having many people online who have used and support the CC3D, with tutorials on how to get started with the board. Additionally, it is tiny, with a size of 16x16mm, thus reducing the amount of space required for the board. Finally, it had sufficient specifications, including auto-stabilisation using a barometer to reduce the amount of coding required. Additionally, it was easy to program, with a very nice, cross-platform GUI called Ground Control Software, where the precise settings of the board, as well as calibration could be easily changed. As opposed to the previous choice, the Hobbyking i86, this is much smaller, as well as being better supported. Additionally, according to most, it is more effective, as, as opposed to the i86, it has the options on how to change the calibration settings of the ESCs and RX system.
ESCs – Turnigy Nano Tech 20A[7]	Given our motor choice, we knew we needed a 20A ESC. By working together with the Mechanics team, we chose this ESC, as it was long and thin, thus would fit in the arm as required, most easily. Additionally, the ESC was highly recommended with most users online having been able to use it with few issues. However, as we progressed through the project, we in fact had a number of issues with the ESCs. In fact, the first ESC we connected to power was faulty and burst into flames! Later, we also learned that the ESC did not have a BEC on board, thus meaning it could not be used to power the control board, thus explaining a number of issues with the control board not powering up. This was not in fact a large issue, since we already had a 5V output from the power distribution board that we could use if necessary, yet, this made a significant impact on the Wiring Diagram.
TX / RX system – Orange Nano[8]	This TX / RX system was chosen because it was very compact, thus most easily fitting in the little space we had in the Can. It was also very highly rated by many

[6] http://bit.ly/1PUlKbW
[7] http://bit.ly/1LFUOc8
[8] http://bit.ly/1M4X4Ln

	other users, who had found it easy to set up. Additionally, Orange, the manufacturer is a high end manufacturer, and normally produce reliable products.

Figure 29: Table showing the components of the flight system and why they were chosen

6.2 - Sensor System

6.2.1 - Components

The first choice we made was to use an Arduino microcontroller, an obvious one, given the team's familiarity with it, and the vast availability of parts and examples. We chose to use a microcontroller board rather than a microcontroller, given the significant amount of supporting circuitry required for regular operation and further circuitry required for reprogramming. We have chosen to use the '**Teensy 3.2**[9]' a very small board (as the name suggests), which includes all the equipment required to reprogram the Can. Additionally, it is very powerful, containing an ARM Cortex M4, far superior to the Atmel chips on other Arduino boards. It also contains a 3.3V voltage regulator, rated for 500mA which we could rely on. For many of the components, we chose to work off many of the choices made by Team Impulse, of whom a couple of members (including their Team Leader (and Head of Software and Electronics) – William Eustace) we had inherited. Thus we immediately chose to use the **MS5637**[10] and **HYT271**[11], the pressure sensor and relative humidity sensor that they had used, since they had been very effective. Though we had considered using the BME280, a Bosch sensor which included temperature, pressure and humidity sensing in one chip, we felt that it was too inaccurate, and too small to feasibly be soldered by hand. Additionally, we chose to harness both the MS5637 and HYT271's abilities to sense temperature to find a more accurate temperature of the surroundings by averaging their results. We also chose to use the **Hope RFM98W**[12], since by using Spread Spectrum Technology, the module is able to more accurately send all the information, over a longer distance. In fact, even without the use of a Yagi, the sensor has been found to be able to send data with little error over 3km, a larger distance than we would ever encounter over CanSat. Additionally, the RFM98W can be used to perform cyclic redundancy checks, ensuring the amount of data received is equal to the amount expected, which would allow us to ensure that errors in receipt of data can be ignored. Though originally we also planned to build in the DRV8833PWR to control the motor to open the arms, this is no longer required, since we are now moving towards two servos. However, this does mean that a second revision of PCBs would be required, since the old PCB is no longer correct.

However, from here the similarities with Team Impulse's electronics end, as we chose different parts. Firstly, we have chosen to use the **GP-2106**[13] GPS module as it has been very effective in testing, and has a very small footprint, much smaller than the GPS module used by either Team Colossus or Team Impulse (last year's teams). Additionally, we have chosen to use the Evaluation Breakout[14] produced by Sparkfun, to help use the module, which has proprietary connectors, and makes use of 1.8V logic. For the servo, we have chosen to use the **Turnigy Analog Nano Servo**, which allow 360 degree rotation, as well as having extremely high torque of 0.8kgcm, with the servo also locking in place when there is no signal transmitted to it, ensuring that the arms would not be able to close again once they have opened. Despite the very advanced specifications, the servo is also very tiny, ensuring two can easily fit inside the top layer piece. Finally, we are going to use the **Sparkfun 9DOF Breakout**[15], as an IMU, containing a 3-axis accelerometer, 3-axis gyro and 3-axis magnetometer. This is

[9] https://www.pjrc.com/teensy/
[10] http://www.meas-spec.com/product/pressure/MS5637-02BA03.aspx
[11] http://www.hygrochip.com/index.php?id=3854&L=1
[12] http://www.hoperf.com/rf/lora/RFM98W.htm
[13] https://www.sparkfun.com/products/10890
[14] https://www.sparkfun.com/products/10995
[15] https://www.sparkfun.com/products/10736

because it is very small, very accurate, and is well supported by Sparkfun, with them having produced a very good Arduino library for the device.

6.2.2 - PCBs

The main sensor system PCB (Revision 1) is shown below:

It was designed in RS / Allied DesignSpark PCB 6[16]. All PCB files are available on the GitHub repository[17], under the Electronics Design section.

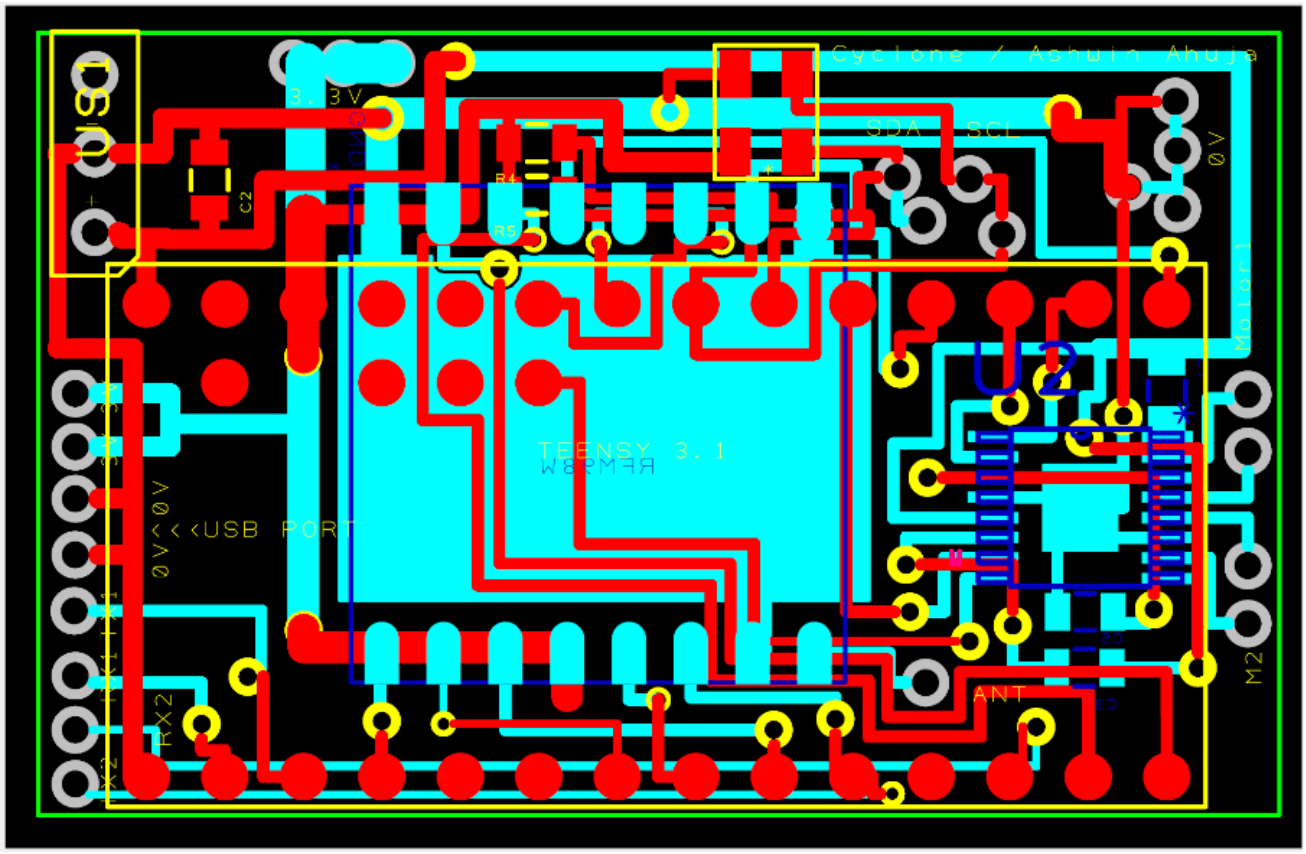


Figure 6.2.2.1: The design for the main sensor PCB

The board shows how the Teensy 3.2 is mounted on the top layer of the board, using Surface Mount Soldering. This was done so that we may conserve space, and could be achieved by ensuring that some insulation is placed between the board and the Teensy, and then soldering the pads of the Teensy directly to the pads on the board. Also on the top layer is the MS5637 – a 4 pin QFN package, as well as resistors required for the I2C line. On the bottom side, there is the Hope RFM98W, a breakout which will be surface mounted to the PCB. There is a ground plane under this, as recommended by the manufacturers, in order to reduce noise. Also, under the Teensy, (labelled U2) there is the Motor Driver (DRV8833-PWR), using the HTSSOP-16 package, and associated resistors and capacitors. It is clear however, that this board does not have the capability to manage the opening of the arms, since it was produced when the plan for arms opening was using a single motor. Hence, in the second revision, we plan to build in this ability as well as fixing a number of other irritations that have been found when the board was soldered. Finally, dotted around the board, there are a number of connectors for the many breakouts and sensors which must be on flying wires. Given this PCB is at the bottom of the Can, the Humidity Sensor will be just below, so that it may have exposure to the air. Additionally, the

[16] http://www.rs-online.com/designspark/electronics/eng/page/designspark-pcb-home-page
[17] http://www.github.com/CycloneCanSat

IMU breakout and GPS evaluation board will be on the same layer, attached with short wires. The GP-2106 module, however has to be at the very top of the Can, to ensure that it can get a fix, and find the Can's position.

PCB Assembly and Review

The manufactured PCB was as below:

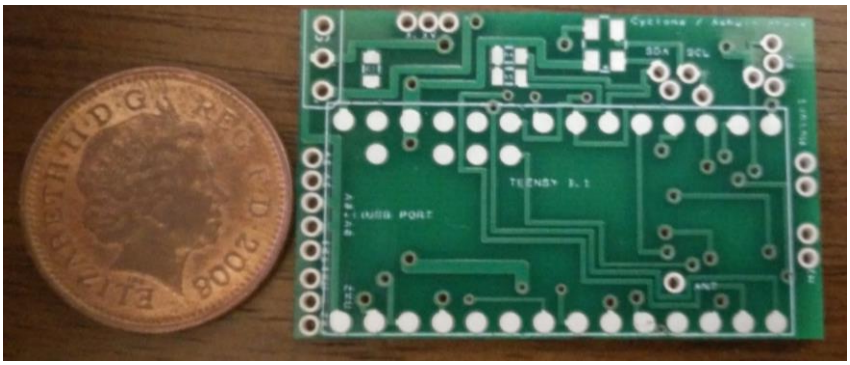


Figure 6.2.2.2 – Front view of manufactured PCB

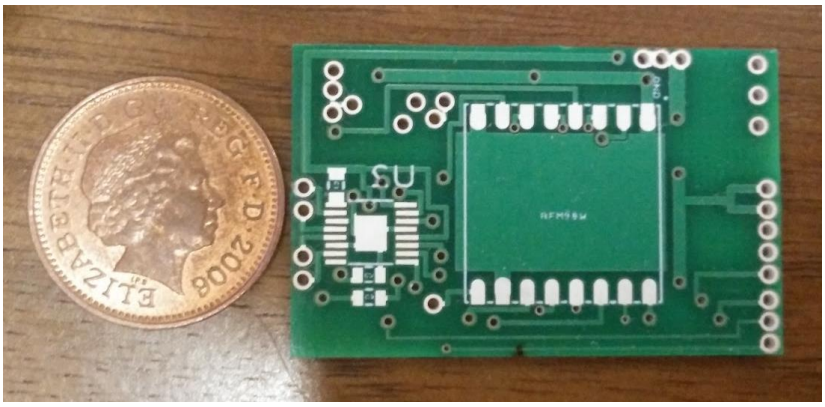


Figure 6.2.2.3 – Rear view of manufactured PCB

When all the parts had been soldered to the board (with the exception of the Hope RFM98W which had not arrived at the time of writing), the board, looked like this:

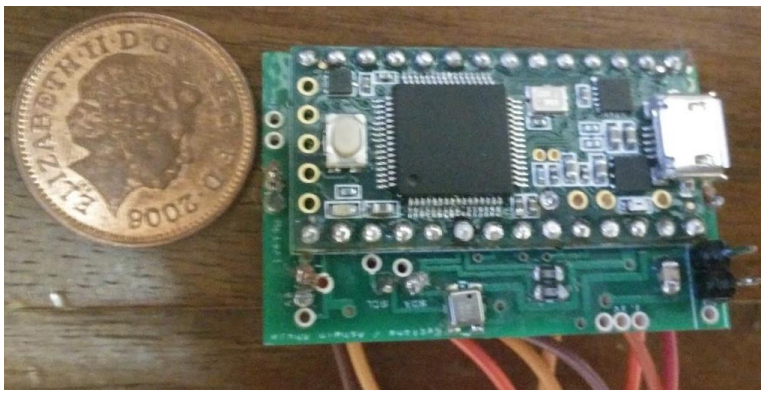


Figure 6.2.2.4 – Front view of assembled PCB

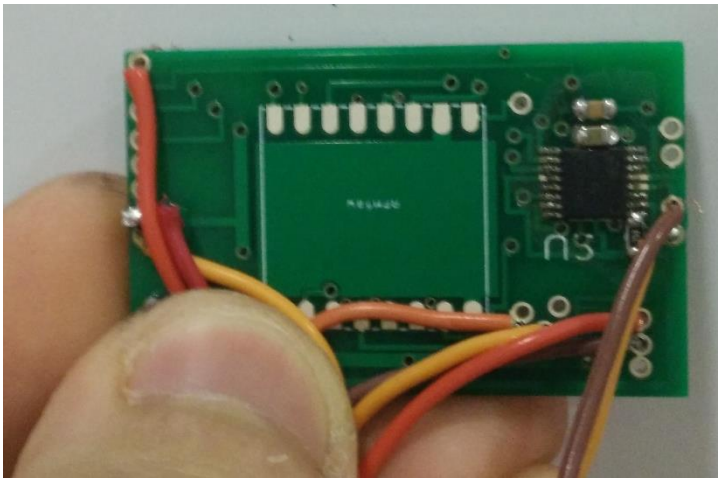


Figure 6.2.2.5 – Rear view of assembled PCB

From preliminary testing, all the components on the PCB work, including the ports which are made for Off-Board connectors, which have tested using easily removable Female Header Wires, connecting to Male Header Pins on the breakouts, to ensure no damage is done to the expensive breakout boards. This suggests that all the electronics design is entirely correct. However, there are a couple of minor issues. Firstly, the mechanism of soldering the Teensy to the PCB, by soldering through the holes in the Teensy straight to the PCB underneath despite working, was quite painful, requiring a lot of care. However, given the amount of space gained through this mechanism, it is likely that we will stick with this as a method of connecting the two. Additionally, a number of pins, placed directly underneath the Micro-USB connection of the Teensy were very hard to solder, and are always very close to touching the connector, sometimes shorting it out. Thus, in the next revision of the board, we hope to relocate them, even if this requires a slight increase in the size of the board.

6.3 - Camera System

For the camera system, we have chosen to use a pre-made FPV system, with the **Mini FPV Camera**[18] combined with the **HobbyKing FPV Transmitter**[19]. Though we had considered using an Intel Edison and Wi-Fi transmission with a NTSC webcam, it transpired that it would take up more space, as well as being a lower quality. Thus, we had the choice of 5.8GHz or 2.4GHz FPV. Though there were some problems regarding legality of using certain powers of 5.8GHz transmission, through more research, it is clear that one can use devices with a transmit power of under 25mW without the necessity of an Amateur Radio License[20]. Though this severely limits the quality of image we can use, it is still superior to the quality one receives when 2.4GHz is used. Additionally, there are similar issues regarding the maximum transmission power of 2.4GHz. Finally, we have also chosen to include an OSD (On screen display) to show the battery stats of the quadcopter, so the pilot would know when the quadcopter is about to run out of battery. Though we have tentatively ordered one from HobbyKing (in fact the **HobbyKing OSD**[21]), we feel that it is actually inefficient, and in future revisions we hope to include a custom OSD in the power distribution board.

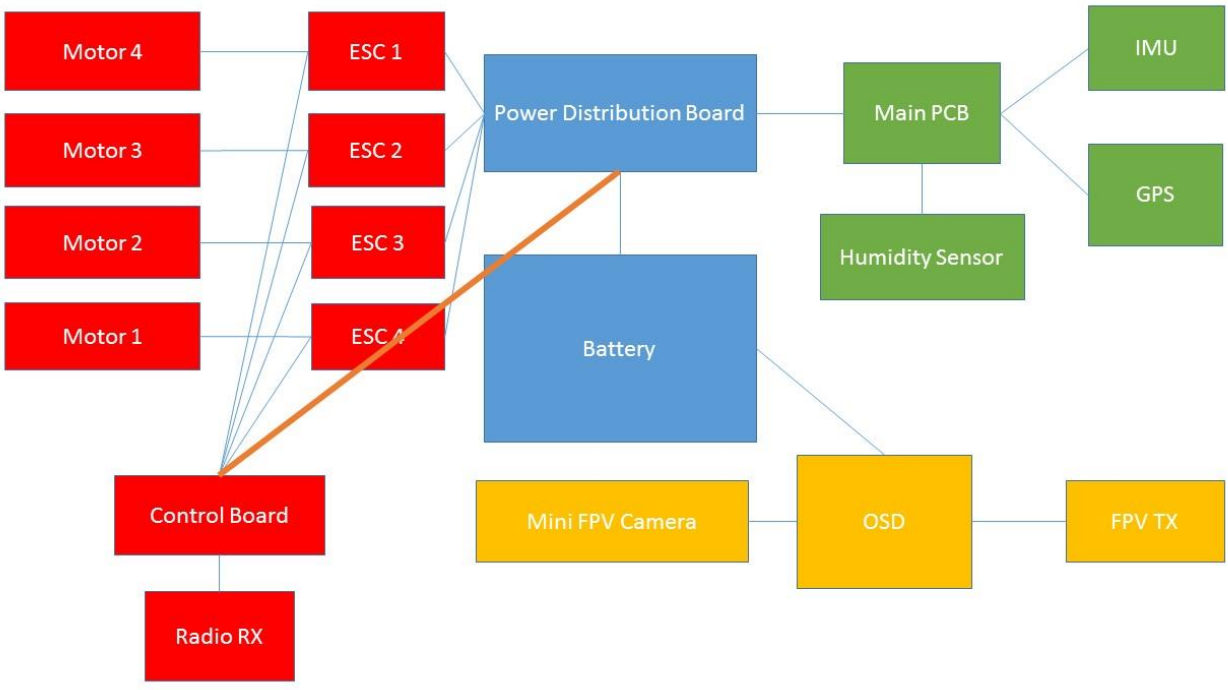
[18] http://bit.ly/1PYEpTe
[19] http://bit.ly/1RhYGBA
[20] Ofcom IR2030/27/3 – Available here (page 63): http://stakeholders.ofcom.org.uk/binaries/spectrum/spectrum-policy-area/spectrum-management/research-guidelines-tech-info/interface-requirements/IR_2030.pdf
[21] http://bit.ly/1XDTcoH

6.4 - Communications

Given the large amounts of data required, a number of different data transmission methods are being used to stream data to and from the Can. As most RC systems work off 2.4GHz, we are following this trend, as this allows to exploit the large number of small TX and RX units specifically built for RC helicopters and indeed quadcopters. Additionally, the use of this transmission method links up well with the **OpenPilot CC3D Control Board**[22], without much work, thus allowing us to simplify the process required to get the flight system working manually, thus allowing more time to produce autonomous movement.

For the camera system, as discussed above, we have chosen to make use of the 5.8 GHz transmission systems, though there are some problems with the low penetrating power of this high frequency. However, given the low distance that the CanSat will travel, we have concluded that this is not a large problem, especially given that we will always have line-of-sight with the Can.

Finally, for the sensor system, we are going with the tried-and-tested 434 MHz RF transmission frequency, given the lower path loss and thus large distances we could easily get using this protocol. This ensures a high reliability of data transfer, and though this means we could transfer less data, this is not a large problem, since the data, probably 32 bits in length, would only need to be transferred once a second. Given the specifics of the spectrum provided to us, we can also begin to determine the specifics we can use. We have been provided with a specific frequency of 434.07 MHz. Given the teams around us have 433.98MHz and 434.25MHz[23], a safe bandwidth would be around 100kHz, thus the nearest setting, 62.5kHz will be used. Making use of the LoRa Modem Calculator Tool made by SemTech[24], this will allow a data rate of 3348 bps, assuming the lowest possible Spreading Factor (6), with a link budget of 139 dB and a receiver sensitivity of -122 DBm.



[23] According to the spreadsheet of allocated frequencies provided during the 2015 CanSat Teacher's Workshop
[24] http://www.semtech.com/apps/filedown/down.php?file=SX1272LoRaCalculatorSetup1%271.zip

Figure 6.4.1: Diagram of system wiring (made using Microsoft Powerpoint[25]) – main change in red line

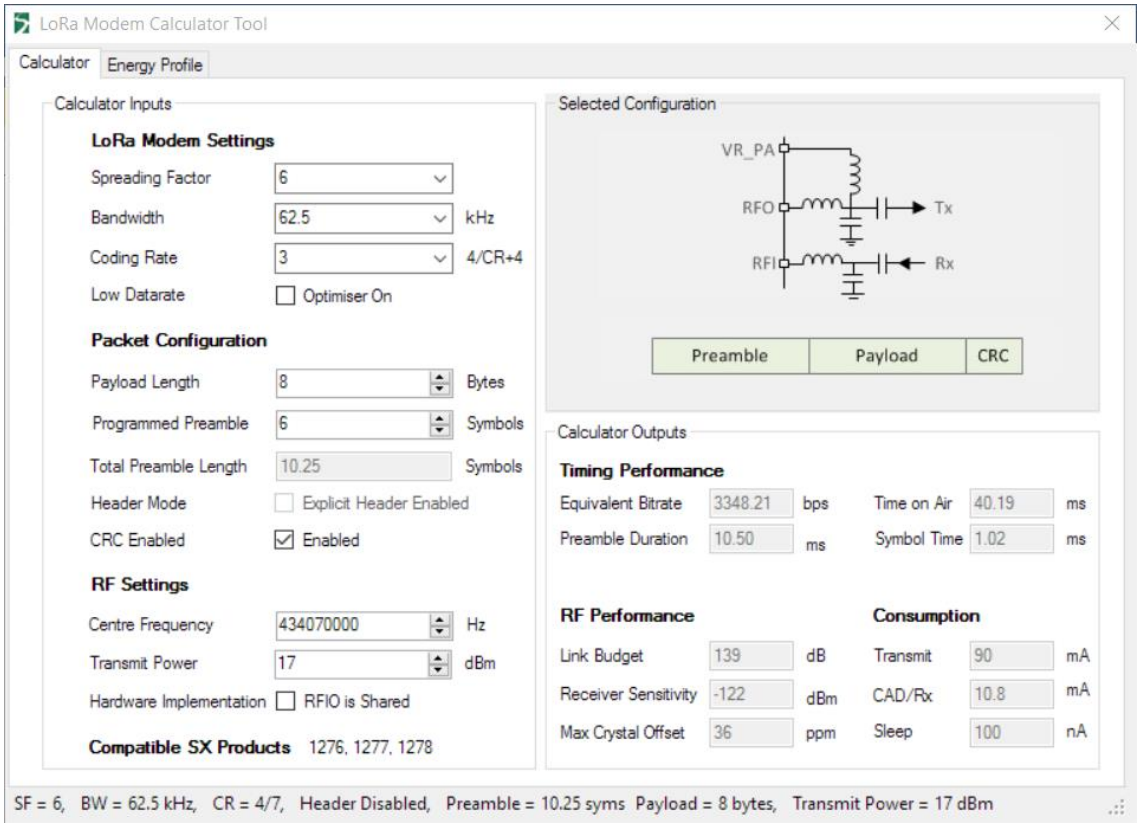


Figure 6.4.2: Calculations for Hope RFM98W, made using the SemTech LoRa Modem Calculator Tool

6.5 - Testing

6.5.1 - Sensor System

In order to test the sensor system prior to the PCBs' arrival, working alongside the software elements of the team, we tested the individual components, producing code individually for each one, before attempting to combine them. This tactic was however, ineffective for the pressure and temperature sensor, where the SMD package meant that it would impossible to test before receiving the PCBs, thus the previous working uses of the product as well as the datasheet was very closely investigated, in order to ensure the wiring (though in essence very simple) was not incorrect.

Once the PCB had arrived, we attempted to replicate the same process, using the PCB, and this was relatively successful, though revealed the problem that some of the pins were not connected to the Teensy upon the board first being soldered, due to the rather tricky mechanism of connecting the two, though the mechanism was clearly worth the pain. Additionally, we are yet to test the Hope RFM98W on the PCBs, since the parts were delayed due to them being backordered. However, given William and other members of the team's relative familiarity with the board, we expect few problems with the electronics.

6.5.2 - Camera System

The camera system was initially tested easily, with the setup being well documented. Though the quality of the images produced was not very high resolution, it was at a very fast frame rate, which ensured there was little to no lag, a very important and necessary part if the quadcopter was needed to be piloted manually using the camera. Additionally, we found the field of vision rather low, with it being around 90 degrees. Hence, we

[25] https://products.office.com/en-gb/powerpoint

are now beginning to look for possible wide angle lenses which would increase the field of vision, making it easier to fly using the camera if required, while anyway generating better and less constricted images. Additionally, it is possible that when we start to connect the separate systems, there might be a need to voltage regulator of some kind, in order to reduce the noise produced by the ESCs which would affect the camera image, especially given that both of them are connected to the same 3 cell power supply.

Figure 6.5.2.1 – Camera system sample frame

6.5.3 - Flight System

The first part of the flight testing consisted of using a frame equivalent to our CanSat and mounting all the parts on it, in order to test it would be able to fly. However, due to the lack of our power distribution board, we were also forced to use an external BEC and a bulky bought Power Distribution Board, thus increasing the size of the setup. However, this (as shown in the image) was able to be assembled, and then flown (with a mass of 370g), easily being capable of lifting its weight with less than 30% throttle required, however with larger propellers than the ones which we plan to use. Hence the next step is to use the correct propellers, attempt to do the same, before moving on and testing the system inside the actual can, with the other systems, which we hope to accomplish before the end of the year. From here, any problems could be fixed, with any required redesigns to the electronics or mechanics.

Figure 5.5.3.1 – Frame for Flight System testing

6.5.4 - Battery Management

A central concern around the electronics system is the battery size and whether the system will fly for a sufficient period of time. This was attempted to be found during testing. The power was left on for the motors for in excess of 15 minutes, depleting the entire battery in this time. Given the power requirements of the sensor system and camera systems are minimal in comparison to the flight system, this is largely promising. In comparison to this, the camera system remained on for in excess of five hours in testing, in fact starting to heat up the FPV Transmitter before anywhere near running out of power. It is important to note, while the motors will begin to turn off when the voltage drops below around 11V, the sensor system would continue to work until around 7V (the limit of the voltage regulator), while the camera system would work until around 6V! Hence, if this were to be flown to an unknown area, the can has the benefit of continuing to transmit positional data to help locate it if required, even if the battery is too low for the quadcopter to fly anymore.

Current Draws from Sensor system components (used to help determine battery life)

Component	**Current Draw (milliamps)**	**Conditions**
Teensy 3.2	*38*	*Flashing an LED*
GP-2106	*65*	*Reporting GPS Data at 1Hz*
HYT-271	*4*	*Reporting Humidity and Temperature Data at 1Hz*
MS5637	*2*	*Reporting Pressure and Temperature Data at 1Hz*
LSM9DS1	*10*	*Reporting all possible Data at 1Hz*

Figure 6.5.4.1 – Current Draw of Sensor System

7 - Software

From the fore, the team knew that there was a number of different strands of software that were needed in order to control the can well, allowing it to become an autonomous drone. We decided, alongside the Electronics team, to keep the flying systems (through the main control board), the sensor systems and camera system separate, thus allowing us to ensure that in case of failure of one system, the others would continue to work. We also divided the team into a number of different strands, with some working on the website, alongside the outreach team and others with the electronics team. We also aided in the outreach, putting all of our code on GitHub (http://github.com/cyclonecansat) under the MIT license, thus allowing anyone else to use our code if they wanted to. We are also attempting to produce documentation to better explain how everything works, for this to be even easier. In many ways, this process also works the other way, as we rely off some work others have completed, especially in the sensor system, learning from their examples. The flying system of the Can required little to no software, given that we were using an advanced control board which includes auto stabilisation. Additionally, we decided that the autonomous aspect of movement would be achieved by altering the signals sent to the flying system, using a computer to calculate the signals to send to the RX of the Can. However, we were still required to write a very complex base station software to allow us to send the correct commands to the Can, so it could monitor the place of the Can at all times and issue commands to move it to a user-defined location. Since the first progress report much of the sensor library has been written / adapted from other sources, producing a system which has been used to individually test specific components. From here, the aim is to put the sensor system together, forming one complete program which will control all sensors as required.

7.1 – Algorithms

7.1.1 – Designing the Algorithm

To design an equation for estimating agricultural viability of a particular site, we thought about how the different inputs (temperature, humidity and pressure) would affect the growth of crops. We came up with several mathematical equations to illustrate these relationships, plotting several graphs where the y-axis is a measure of agricultural viability with a maximum value of 1 and the x-axis is the input.

Temperature would control a plant's enzymatic activity, meaning that temperatures near the optimal temperature for a given enzyme would favour plant growth. To show this representation graphically, the graphs of $y = \frac{-2}{3x}$ and $y = 1 - \frac{x^2}{3}$ were combined to produce the graph shown here. The lines cross at x=-1 where the two gradients are the same, ensuring the curve is continuous. When x is less than -1, $y = \frac{-2}{3x}$ is used, showing the increasing enzymatic activity as temperature increases and hence the increasing agricultural viability. For values of x greater than -1, $y = 1 - \frac{x^2}{3}$ is used, with the steep drop representing the effect of enzyme denaturing at high temperatures. For appropriate use by the can, the measured value for the temperature would be altered such that the optimal temperature (e.g. 27°C for rice) returns a value of 1 for agricultural viability and normalised such that a range of suitable temperatures still return relatively high values for agricultural viability. The graphs could also then be stretched to allow a sensible range of temperatures to yield high viability. For rice we will replace x with $\frac{T-27}{9}$ which creates a maximum temperature of T=42.5.

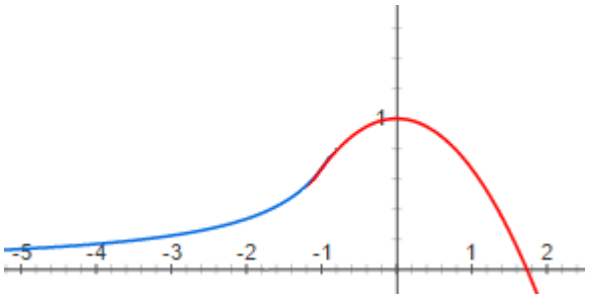


Figure 7.1.1.1: Graph used to find Agricultural Viability from temperature

We decided that humidity could be a proxy for soil water content, with higher humidity resulting in more water available for crops and so a higher agricultural viability. Therefore, we decided that $y = -x^2 + 2x$ could be used. The humidity measured by the can would be divided by 100 so that 100% humidity returns a value of 1 for agricultural viability.

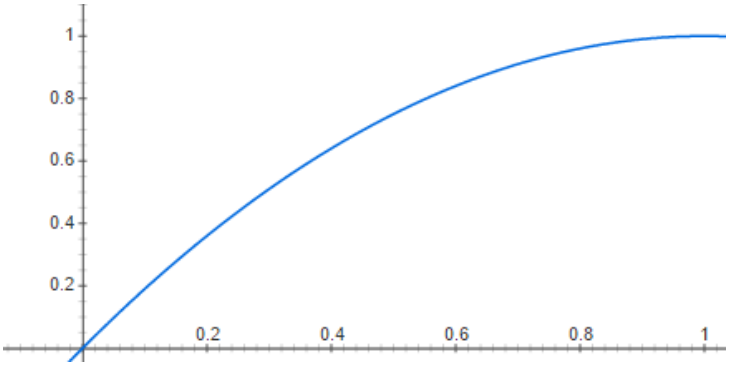


Figure 7.1.1.2: Graph to find Agricultural Viability using humidity

We assumed that for Earth-originating crops, a pressure of 1atm would be ideal for a plant, with a vacuum being more detrimental than higher pressures. Taking this into account, the graph of $\frac{-x^2+3x-1}{x}$ was chosen as it displayed the features we wanted. As it peaks when x=1, this graph would not require any further calibrating provided the pressure is in atmospheres. The three values for agricultural viability multiply together to give an overall value between 0 and 1.

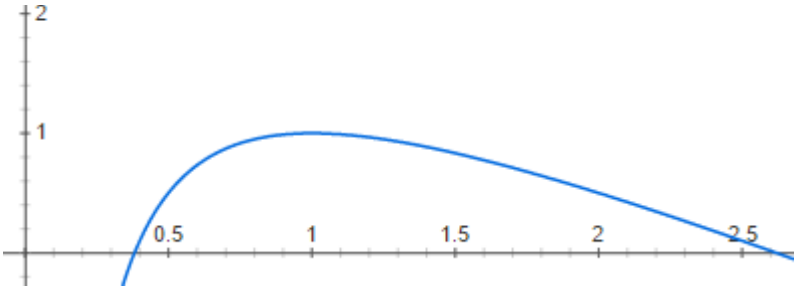


Figure 7.1.1.3: Graph to find Agricultural Viability using Barometric Pressure

Summary:

$$Agricultural\ Viability = V_T \times V_H \times V_P$$

where;

	Calculation	Condition	Notes
V_T = Viability of Temperature	$V_T = \frac{-2}{3(\frac{T-27}{9})}$	For $T < 18$	Where T = temperature in degrees Celsius.
	$V_T = 1 - \frac{-(\frac{T-27}{9})^2}{3}$	For $18 \leq T \leq 42.5$	
V_H = Viability of Humidity	$V_H = -\frac{H}{100}^2 + 2\frac{H}{100}$	For $0 \leq H \leq 100$	Where H is relative humidity in %
V_P= Viability of Pressure	$V_P = \frac{-P^2 + 3P - 1}{P}$	For $0.4 \leq P \leq 2.6$	Where P = pressure in atmospheres

Figure 7.1.1.4 – Table of equations for calculating Vt, Vh and Vp

7.1.2 - Assessing and improving our algorithm

	Nebraska	Minnesota	Illinois	Nevada	Texas	Virginia	Alaska	Copenhagen	Timbuktu
Temperature/C	10	5.8	11.4	11.4	21.1	14.2	5.6	11	29.1
Pressure/atm	1.002714	1.002714	0.9967925	1.006662	1.002714	0.9987664	1.00074	0.990781	1.004688
Humidity/%	45	55	50	20	55	60	90	85	5
V(temperature)	0.545454546	0.394736842	0.625	0.625	1.000041152	0.882352941	0.38961039	0.6	1.27
V(pressure)	0.999992654	0.999992654	0.999989679	0.999955912	0.999992654	0.999998476	0.999999453	0.999914219	0.999978125
V(humidity)	0.6975	0.7975	0.75	0.36	0.7975	0.84	0.99	0.9775	0.0975
Agricultural Viability	0.380451751	0.314800319	0.468745162	0.22499008	0.79752696	0.741175341	0.385714075	0.58644969	0.123822291
Actual Agricultural Output	90	50	75	10	90	50	10		
High									
Higher Middle									
Lower Middle									
Low									

Figure 7.1.2.1 – Table comparing predicted agricultural viability versus actual agricultural output

To test out our algorithm for agricultural viability, we input existing data for temperature, humidity and pressure for a range of locations. Because of the abundance of available data and the diversity in climate across the region, we decided to use data from across the USA. The algorithm was calibrated for growing wheat, the most widespread crop in the area, by setting the optimal condition for temperature to 21C. In addition to the data from the USA, we also tested the algorithm for two more locations that we expected to be at opposite ends of the spectrum for growing wheat: Copenhagen, Denmark, one of the world's largest producer of wheat per hectare, and Timbuktu, Mali, a city in the Sahara Desert.

To measure how well our algorithm worked, we compared the algorithm outputs with data for the percentage of land that is farmland in each region, assuming that areas which have a higher natural agricultural viability would contain a higher percentage of farmland.

Overall the algorithm outputs match up relatively well to our expected values. The greatest dissimilitude between an output and the real world situation was the value for Nebraska, with the algorithm returning a relatively low value for agricultural viability despite it being one of the most agriculturally productive regions that we looked at. We believe this mismatch is largely down to other factors that our algorithm does not take into account, such as soil quality and precipitation, areas in which Nebraska performs well.

There are two main ways that we seek to improve the algorithm. Firstly, when calculating the viability of the temperature, our algorithm returned values that were greater than 1 for Texas and Timbuktu. This should be impossible as 1 should be the maximum possible value, indicating that the temperature is at the optimal temperature. This is an issue we will look to fix. Also, while researching the agricultural output of certain regions, we noticed that often areas that produce a lot of a certain crop have conditions that are not necessarily close to the optimal conditions. This is most evident in the data for Nebraska and Copenhagen where the average temperature is around 10C below the optimal temperature, yet the crop production is still very high. As it stands this difference causes the output value for agricultural viability to be significantly affected. To address this, we will improve the equation for the viability of temperature to allow a wider range of inputs and still return a high value for overall agricultural viability, more accurately reflecting real life.

7.2 - Website

The Software Team has worked hard to design (and continue to develop) responsive website, now available at: http://teamcycl.one which is both very attractive aesthetically, as well as containing a lot of information, allowing visitors to easily find out more about the project and indeed us. The website even has an embedded blog where each team has so far written an article, including the Software and Electronics team. We hope to continue doing this and even make them more frequent as the launch draws nearer. This website has been extensively promoted using our social networks, and so far has encountered over 2000 page views[26], thus allowing this many people to learn more about our project. The website proved an interesting challenge for us, as we attempted to use all aspects of web development to our advantage, using JavaScript to allow the website to be responsive, CSS for stylistic aspects, and HTML(5) for the core programming. Additionally, a mobile website was designed, as this responds to the idea that more and more of the visits we would receive would be from mobile devices. The mobile website was designed to be as simple as possible, while still very usable. This has proved very successful, as the average time spent on our website (approximately 2 minutes on other devices) is closer to 4 for mobile[27]. Finally, we decided to produce mobile apps for iOS, Android and Windows Phone, in order to reach a greater audience even more easily. To do this, we employed the abilities of the Software Teams, producing apps in Swift, Java and C# respectively. To date, Android and Windows Phone apps have been launched to their respective App Stores, and been successful, with the Android app having more than 400 downloads[28], however, we are having a few problems with the iOS application. Though it has been designed, the app is currently being rejected by Apple as its market is too 'niche'[29]. However, we hope to make the app more general for engineering and especially St Paul's in order to counter this problem. This however, is taking a lower priority, while the software team begins to concentrate on the code required for the project itself, hence the iOS app falling behind schedule.

7.3 - Can Code

The majority of progress in the can code has been on the libraries that govern how we read data from the various sensors. This has largely been delayed as the majority of the coders, such as Ashwin and William, have been highly involved in the Electronics Design and choice of components, where software actually played a large part in why specific components were chosen. Ultimately, most of the components chosen use the relatively simple I2C protocol, apart from the GPS sensor which will make use of TTL Serial, another well documented protocol. Unfortunately, many of the components do not have recognised libraries, so we are producing our own libraries. We will produce two main libraries, one for the sensors, and one for communications. The Sensor library will manage the reading of data from all sensors which will in fact make use of other libraries for each sensor. While the MS5637 and HYT271 libraries are being made in-house, the library for the IMU has already been produced by Sparkfun, so we are making use of this. Additionally, for the MS5637 and HYT271, we are basing much of our work upon the Team Impulse libraries written by William for the CanSat competition last year, under the MIT licence. However, we are hoping to make improvements, especially exploiting the greater accuracy that the MS5637 provides (with the sensor claiming an accuracy of 10m for altitude) than used by Impulse. To date, much of this sensor library has been written, with it streaming live data from the sensors. However, there is still much improvement to come to further simplify the code required for the main Arduino C code of the product. For the GPS sensor, we are going to use the TinyGPS++[30] (an open source GPS library) to parse the NMEA statements that the breakout produces. This library is very

[26] Statistics found using Google Analytics
[27] Statistics found using Google Analytics
[28] Statistics found through analytics on the Google Play Developers' Console
[29] Rejection message from Apple was the following: "We found that the usefulness of your app is limited because it seems to be intended for a small, or niche, set of users."
[30] http://arduiniana.org/libraries/tinygps/

well acknowledged and the team has lots of experience having used it in the past (though with other GPS units). Additionally, we are also planning to produce a communications library, to simplify the use of the Hope RFM98W in the actual program. For this, again, we are working off the base of the Team Impulse Library, which was produced by William as part of last year's competition (under the MIT licence), which makes the job much simpler. However, it still must be adapted in order for it to work most efficiently for us. While little progress has been made on this particular library, largely waiting for the availability of Hope RFM98W boards to be tested with, the code from Impulse's library has been tested and found to be largely meeting most of our specifications. Hence, our library will inherit significantly from theirs, though we are also hoping to implement parts of the specific transmission protocols that we will use into the libraries to further simplify the main code. Hence the list of libraries we will use is as follows:

1. Sensor Library
2. RFM98W Library
3. LSM9DS1 Library
4. TinyGPS++

Additionally, a basic plan, illustrated by the flowchart below, for the main loop (since we are using Arduino, a language which has a cyclic program structure), where the data is collected and then transmitted, has been created.

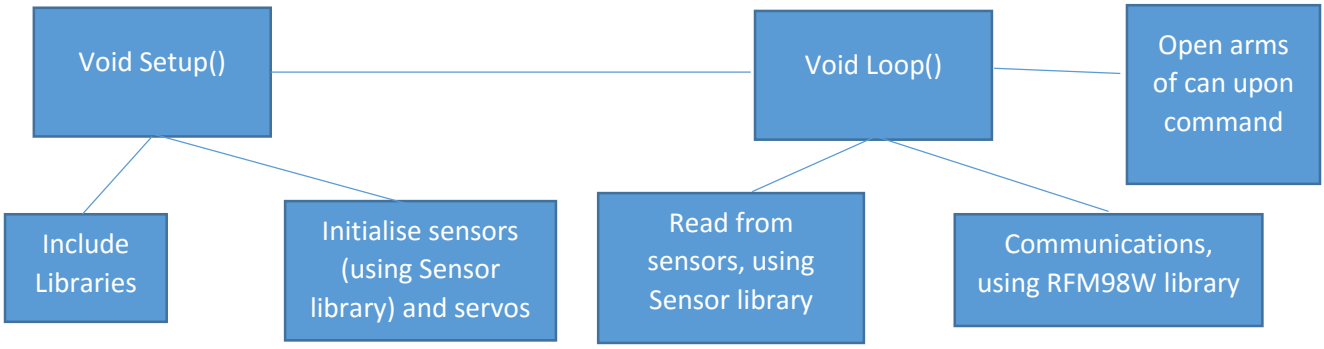


Figure 7.3.1: Design for Can code (made using Microsoft Powerpoint[31])

Though not yet pieced together, code has in fact been written for reading from each of the sensors, thus, we will be quickly able to put it together and create one coherent program. From this testing our observations were as follows: We found the vast majority of sensors largely unchallenging to read from given the documentation provided and in fact sample code from Impulse which we could adapt. In fact, the only system that has so far posed any challenge is the communications using the RFM98W, given the complex protocols that it makes use of; the SPI protocol. This has been hugely assisted by William's personal experience using the module, since he has used it on a number of occasions for a number of different projects. However, despite managing to read from the LSM9DS1 relatively easily, we have found a problem with the sensor, notably that the inbuilt sensor calibration, which does not appear to show the correct heading, a very important part of the system, since the autonomous motion relies upon this ability. Hence, we are currently experimenting with other physical manners of calibrating the magnetometer, in order to prevent this from occurring. Additionally, though concerned that the proximity of the motors to the magnetometer in the can (directly above it) could have a large impact on the results, by testing the sensor nearby motors, (especially given that when the motors begin to spin, the arms will open and the motors will no longer be in proximity to the LSM9DS1), we found

[31] https://products.office.com/en-gb/powerpoint

little impact to the results. Additionally, we hope that the software would be able to compensate for any inaccuracies in the magnetometer by making use of the GPS position, and thus extrapolating the heading of the can.

7.4 - Base Station

Since the last progress report, the exact requirements of the base station have been developed, with a greater emphasis on parsing and storing data with a lower emphasis on live graphing of data, though this might still be carried out. We have decided to make use of C# with the WPF framework for this base station software, given the main coders familiarity with this. Hence, our updated specification is below:

1. The software should parse and store sensor data from the Can (in a '.CSV' – Comma Separated Values file. The data transmitted will be exclusively in bytes, as required by the RFM98W, thus there would likely be a requirement of adapting the exact input in order to arrive at the correct data received from the can.
2. The software should display the temperature, pressure, relative humidity and IMU data in real time from the can.
3. The software should be able to calculate the altitude, also making use of a barometer on the ground station, hence calculating relative height, and displaying this to ensure the person flying the quadcopter has sufficient data to fly comfortably.
4. The software should be able to display information about the exact information about the quality of the link to the Can, providing the signal strength, and information about the received packages.
5. The live camera feed should be displayed on a separate monitor, ensuring that the systems are separated, even in the base station. This will make use of a separate battery supply, with FPV RX and antenna, as has been tested.
6. The software should display the current Can location, preferably on a map (making use of Google Maps[32] or OpenStreetMaps[33] to help display the location of the can.
7. The software should use the incoming data to calculate altitude (using barometric pressure) and the agricultural viability (using the predefined algorithm)
8. The software should allow you to open the arms of the can, by sending a command over the RFM98W
9. The software should be able to autonomously monitor the position of the quadcopter and move it to a user-defined location, using (6), taking input from the user.

[32] http://maps.google.com
[33] http://www.openstreetmap.org/

8 - Integration of Electronics and Software within the Mechanics

9 - Data Processing

Given the possibility of launch delays, there is a likelihood that the team will have very little time to analyse any data. In order to reduce the time taken to analyse all the data that we input, we considered two main manners. One involved the use of Microsoft Excel to plot a series of graphs, and then analysing them, finding the correlation factors and suchlike individually for each graph. Though very simple to use, this method is time inefficient and in fact could produce imperfect results, with a lack of customisability in Excel. The other method is making use of MathWorks MATLAB, a high level language, used to model data and visualise it. Though this is in many ways much more of a challenge, since it is a new programming language for much of the team, it is much more efficient, and produces better results. Additionally, it would require far less effort from an already tired team, at the launch. Hence, we chose to make use of MATLAB, and determined the specification for the code that we would use for Data Processing:

1. Read in a CSV file, individually finding and separating the readings of different sensors.
2. Graph all possible pairs of data (i.e. temperature vs relative humidity, pressure vs temperature, altitude vs time) using a number of different graphs, and calculate relevant data for each graph, including the correlation factors and best fit lines and curves. From here, we are likely to select the most interesting graphs to incorporate into our presentations to the judges.
3. Produce an interesting GUI, making use of MATLAB's built in publishing functions, containing all data and graphs which we could put on our website, to allow all users to easily and more comprehensively read about the results of our project.

10 - Ground Support

During launch, we need to ensure that we are able to easily monitor and control all the different systems as required. As with most of the other electronics, we chose to keep them largely separate, in order to ensure a failsafe mechanism. The only separation from this is the connection from the flight system, with the TX controller also connected to the computer, so that it could be connected automatically by the computer base station software, which will be running on the computer. This will allow the system to autonomously control the quadcopter, sending signals through the Endurance R/C PCTx, which connects from a computer to the 2.4GHz transmitter. In addition to this, the controller itself will also be ready to be used if necessary, if autonomous travel is not working. In fact, this might be particularly necessary towards the start of launch in order to stabilise the quadcopter, and may also be necessary at the end to land the quadcopter, though we are hopeful that through testing, the landing system could be built into the autonomous software. In order to run all the computer functions, there will be at least one computer running the C# program. Since this program would likely make use of the .NET framework, this would be a Windows computer. In addition to this one computer, there would likely be at least one more computer in case the first does not work, or runs out of battery (as happened last year during the launch procedure). Entirely separate to the computer, there would also be at least one FPV RX setup, using a 3000mAh 3S LiPo battery connected to a FPV RX (bought as part of a set from HobbyKing), and a small 7" monitor. We are also currently looking into acquiring a set of FPV goggles, in order to further enhance the flying experience of the pilot. This FPV RX will also be connected to a computer which will act as a second monitor. Combining this with free screen capture technology, this would allow us to both easily see the video in a larger screen if necessary and store the video to be viewed later. Finally, the sensor system ground system will be using the same sensor board as the can, since it would have all the required components, as well as always allowing us to compare data in the sky with data from the ground, allowing us to assess changes with altitude. Additionally, the use of a barometer on the ground station would allow me to use a more accurate ground pressure to find the height at which the can is flying. Also, the GPS on board the ground station would enable functions such as 'Return to Home' where the quadcopter would return to where it started, landing at the ground station. In fact, since the can also must have a capability of receiving data (for when to open the arms) the code for the base station electronics can also be very similar, hence reducing the amount of work required by the software team. The only electrical change will be in connecting the RFM98W to a Yagi Directional Antenna rather than a wire antenna. This is simply to ensure that the data transmission will be as successful as possible. In fact, this is not really necessary, with distances of over 3km across London being achieved without the use of the Yagi. However, the Yagi requires little work, since our school already owns one, and in fact would increase our range dramatically. In fact, in a recent High Altitude Balloon flight, the link between base station (with directional Yagi) and RFM98W was maintained at even 110km, a rather ridiculous and unnecessary distance. In fact, for this line of sight was the limiting factor as opposed to the range of the transmission method. Additionally, the choice of using a premade Yagi was made, given that (using experience gained during last year's CanSat missions) the bought one was far more effective than the other Yagi that was built by Team Colossus. In addition, with no other team in the competition from St Paul's this year, the necessity of making one is nil, since the bought one would not be used, if we did not.

11 - Risk Mitigation

Risk	Mitigation
Team Members unable to work due to illness or other reasons	Each team member has a particular role and specialism; hence illness of one member could have a significant impact during the current design and prototyping phase. Our regular team meetings, including conference calls on Skype, help to mitigate absences and the risk of illness by ensuring that everyone knows the current priorities and overall schedule of work, and can contribute even if they cannot be present. Thus, although there have been occasions on which some team members have been unavailable, this has not had any serious impact to date. As we progress through design, manufacture, testing and competition, the risk of absence will remain and we will respond by reallocating roles if and when required.
Delays in construction	Our project plan allows ample time for manufacturing, and is further mitigated by our prototyping approach and use of CAD/CAM techniques. This provides a feedback loop to allow us to refine the design to simplify final manufacture.
Malfunction of tools or equipment	This risk materialised during our prototyping phase. We mitigated the impact by outsourcing the 3D printing of our prototype Can parts to a local company. These parts took longer to be delivered than had been promised and turned out to be of poorer quality than we could have manufactured ourselves. Whilst we were able to use the parts to prototype our design, we have reverted to in-house manufacture (on repaired equipment) for our final parts.
Receipt of late, poor quality or damaged components	Our primary mitigation against receiving unusable components is to plan ahead and order parts early with float in the schedule, allowing for some re-work or re-ordering if necessary. This requires good team co-operation in order to be able to order long-lead items ahead of schedule.
Problems with software production and testing	The key to successful software development is to have clear and unambiguous requirements. And by working across disciplines, we are able to review and confirm the developer's understanding of these requirements. Our plan includes both software testing and integrated testing with the Can components, as well as an allowance for some software re-work should this be required.
Malfunction of electronic components	We will review the performance of our electronics during testing and consider whether this risk warrants production of spares or any revision to the design.
Overheating of electronics once in the CanSat	Having considered this risk, we are confident that overheating is unlikely to be an issue. The heating effect of our rotor motors will not be an issue as they will be at the ends of the arms and away from the electronics within the Can. Additionally, air flow through the Can will cool the electronics.
Injury during manufacture	Relevant team members are experienced in the use of school DT equipment and will nonetheless review the proposed manufacturing techniques (laser cutting, 3D printing, etc.) with the teacher in charge and agree safety procedures, levels of supervision and conditions for operating the equipment.

	School safety procedures will apply in the unlikely event of any accident.
Injury during testing and operation e.g. lacerations from spinning rotors	We have consulted with our teacher in charge and have agreed that initial testing will take place with the Can tethered to a table and behind a Perspex screen. This is due to the inclusion of high speed rotors in our quadcopter design and the risk of parts not being safely attached to the Can.
LiPo explosions in case of malfunctions	The batteries we have chosen to use mitigate this problem by monitoring the usage of the battery and turning the system off, if the battery has too high or too low a voltage, or if there is an over-current.

12 - Gantt Chart

		Oct		Nov					Dec				Jan				Feb					Mar		
	Week Commencing:	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	1	8	15	22	29	7	14	21
Electronics																								
	Sensor PCB design																							
	Sensor PCB assembly																							
	Quadcopter electronics assembly																							
	Camera system assembly																							
	Full can assembly																							
Software																								
	Write libraries for sensor																							
	Write libraries for RF98W																							
	Write Can code																							
Mechanics																								
	Layer design (using CAD)																							
	Arm design (using CAD)																							
	Layer manufacture (using 3D printing)																							
	Arm manufacture (using 3D printing)																							
	Manufacture of arm opening system																							
	Can assembly																							
Testing																								
	PCB testing																							
	Quadcopter system testing																							
	Camera system testing																							
	Full electronics testing																							
	Test arm opening																							
	Flight test																							
	Launch procedure testing																							
Outreach																								
	Fundraising																							
	Talks to primary school students																							
	Inside school promotion																							
Base Station																								
	Software																							
	Communications system design																							
	Base station electronics design																							
	Base station electronics manufacture																							
Paperwork																								
	Progress Reports																							
	Submission of documentation to sponsors																							

Figure 37: Gantt Chart, outlining future plans (made with Microsoft Excel[34])

[34] https://products.office.com/en-gb/excel

13 - Mission Criteria

13.1 - Primary Mission

1. The CanSat should transmit air temperature and barometric pressure to the ground at least once per second
2. The CanSat should comply with all of the CanSat guidelines, notably:
    a. It should weigh 370g
    b. It should have a maximum diameter of 66mm
    c. It should have a maximum height of 115mm, bar antennae
3. The CanSat should log all data both on the ground and in the Can.

13.2 - Secondary Mission

1. The CanSat should be able to fly comfortably, being relatively stable in the air and low winds
2. The CanSat should be capable of at least 5 minutes of flight.
3. The CanSat should be able to transmit further data, including relative humidity, thus allowing a computer to calculate the agricultural viability of the area.
4. The CanSat should be able to navigate to a set of co-ordinates autonomously.
5. The CanSat should also be able to controlled manually, if necessary.
6. The CanSat should be able to live stream video to the ground.