CYCLONE

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

CANSATS IN EUROPE

FIRST INTERIM REPORT

Table of Contents

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#).

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

Brief Overview

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.

Progress Synopsis

Since the summer, we have largely been in the planning stages of the product, producing designs for the electronics and mechanics of the product. However, we are now in the manufacturing stage, as PCBs have been ordered, and 3D printing is well underway. Thus, we are well on the way to producing a working prototype by the end of the year, our original objective. Outreach is also progressing very well, with a couple of sponsors having been obtained, and a number of events organised. Additionally, we have many plans for outreach events in the rest of this year, which would allow the CanSat project to gain even more exposure.

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 recognizable 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 2: Cyclone Universal Logo

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 organized 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.

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 will be given to more of the school’s societies such as to EnSoc (the schools engineering society). 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.

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. Again, these apps are to further publicize CanSat. Finally, we have a couple of videos available on Vimeo, and hope to continue making short interesting updates as the project continues.

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, maximising 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.

In the future, we also aim to complete a fundraising exercise, most likely a Cake Sale in 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.

Additionally, two member 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. Additionally, we anticipate few further costs, given that surplus of electronics and hardware were purchased, in case of failures of other components, so we anticipate not needing to buy more components. The main further costs will be PCB design (for which we are sponsored), and manufacturing costs. The manufacturing costs as below, are in fact very low, given that the material costs of ABS and PLA (the two materials used so far) are very low, and that we have access to a couple of 3D printers at school.

Section	Expected Cost / Value	Costs to date
Outreach	£30	£5
Hardware	£26	£10
Electronics Components	£400	£300
PCB Manufacturing	£0	£0
TOTAL	£456	£315

Figure 3: Basic breakdown of costing – for more detailed breakdown, see Figure 5

However, despite the costs being on the edge of the acceptable limits of the CanSat cost per unit, it must be acknowledged 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: 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
MAIN SENSOR BOARD				
Male Headers	Break Away Headers - Machine Pin	2	£ 2.95	£ 5.90
Resistor	Panasonic 75kOhm Resistor	20	£ 0.01	£ 0.24
Capacitor	Murata 100muF capacitor	20	£ 0.22	£ 4.40
Resistor	Bourns 10k SMD 0805 Resistor	20	£ 0.04	£ 0.78
Resistor	Bourns 10k SMD 0805 Resistor	20	£ 0.04	£ 0.78
Motor Driver	Texas Instruments DRV 8833	10	£ 2.50	£ 25.00
Capacitor	TDK 2.2muF	20	£ 0.19	£ 3.70
Capacitor	Kemet 0.01muF	50	£ 0.02	£ 0.78
MCU	Teensy 3.2	5	£ 13.02	£ 65.10
Humidity Sensor	HYT-271	3	£ 26.49	£ 79.47
Pressure Sensor	MS5637	10	£ 1.84	£ 18.40
IMU	SparkFun LSMDS1 Breakout	3	£ 16.28	£ 48.84
GPS:				
GPS Module	GP-2106	2	£ 32.59	£ 65.18
GPS Breakout	Sparkfun GPS Evaluation Board (GP-2106)	3	£ 6.49	£ 19.47
GPS Module Connector	Interface Cable GP-2106	5	£ 0.98	£ 4.90
Micro SD Breakout	Sparkfun OpenLog	5	£ 16.28	£ 81.40
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	6	£ 5.00	£ 30.00
SUB-TOTAL	**£ 1,108.54**			
HobbyKing Sponsorship	£ 458.19			
PCBTrain Sponsorship	£ 150.00			
Cake Sale	£ 300.00			
TOTAL	£ 200.35			

Figure 5: Breakdown of costs (made using Microsoft Excel)

Mechanics

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.

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.

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.

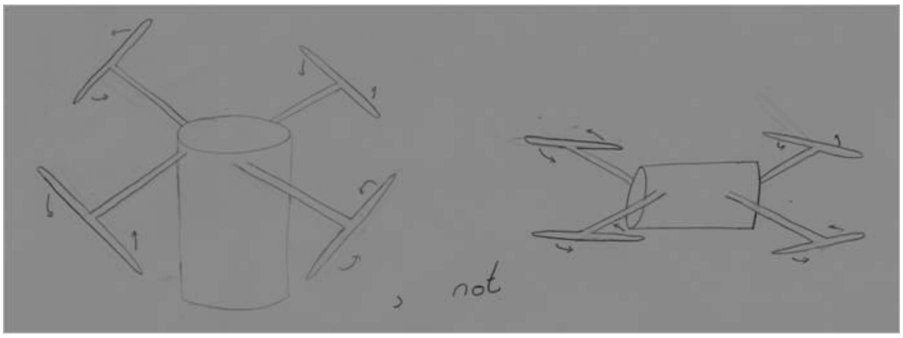


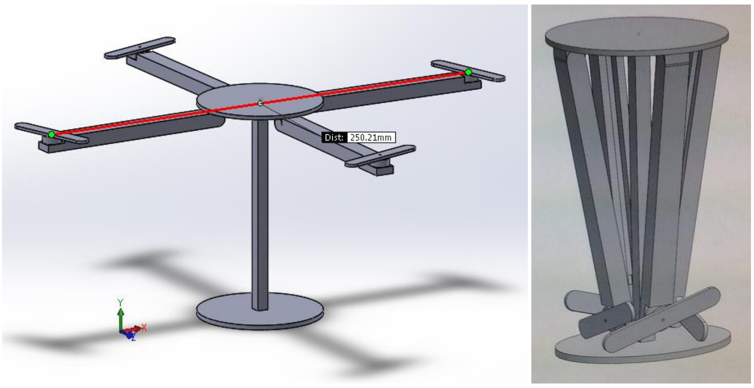
Figure 6 - Can Orientation Decision

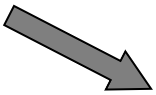
Throughout the design of the Can many decisions were strongly influenced by this space / stability trade off. For example, the exact length of the arms:

> This initial vertically orientated design (Figure 7), 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 8). Due to this change the diagonal distance decreased by about 35mm (see Figures 7 and 8).[1]

Figure 7





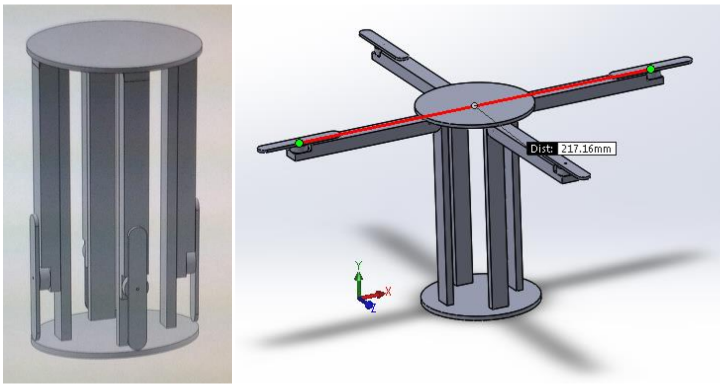


Figure 8

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 9 → Figure 10 (Figure 11 is the same as Figure 10 except with the arms unfolded to show the wall)):

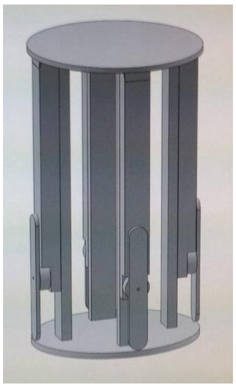


Figure 9

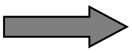


Figure 10

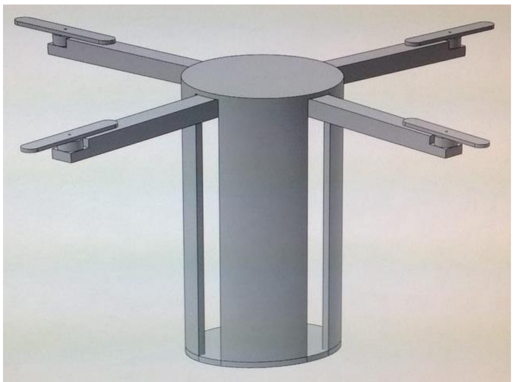


Figure 11

[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 12 shows a possible solution we came up with to increase arm stability

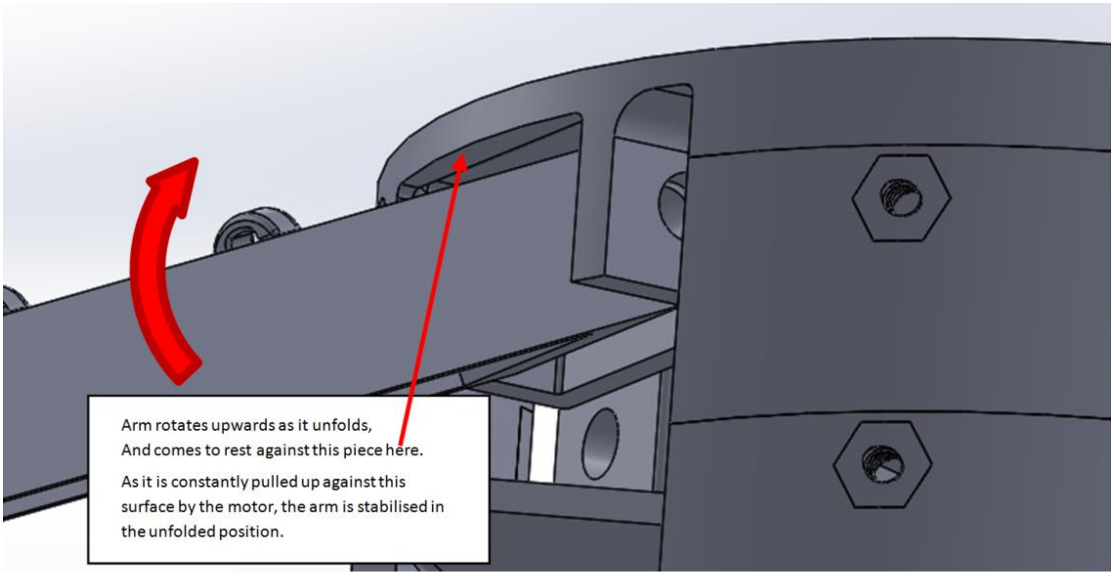


Figure 13 shows another possible solution we came up with to increase arm stability

We made a decision to use the solution shown in Figure 13 as we thought it would work sufficiently well, yet would be simpler and so more reliable than the supports shown in Figure 12.

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 14, and the consequent layer creation can be seen in Figure 15.

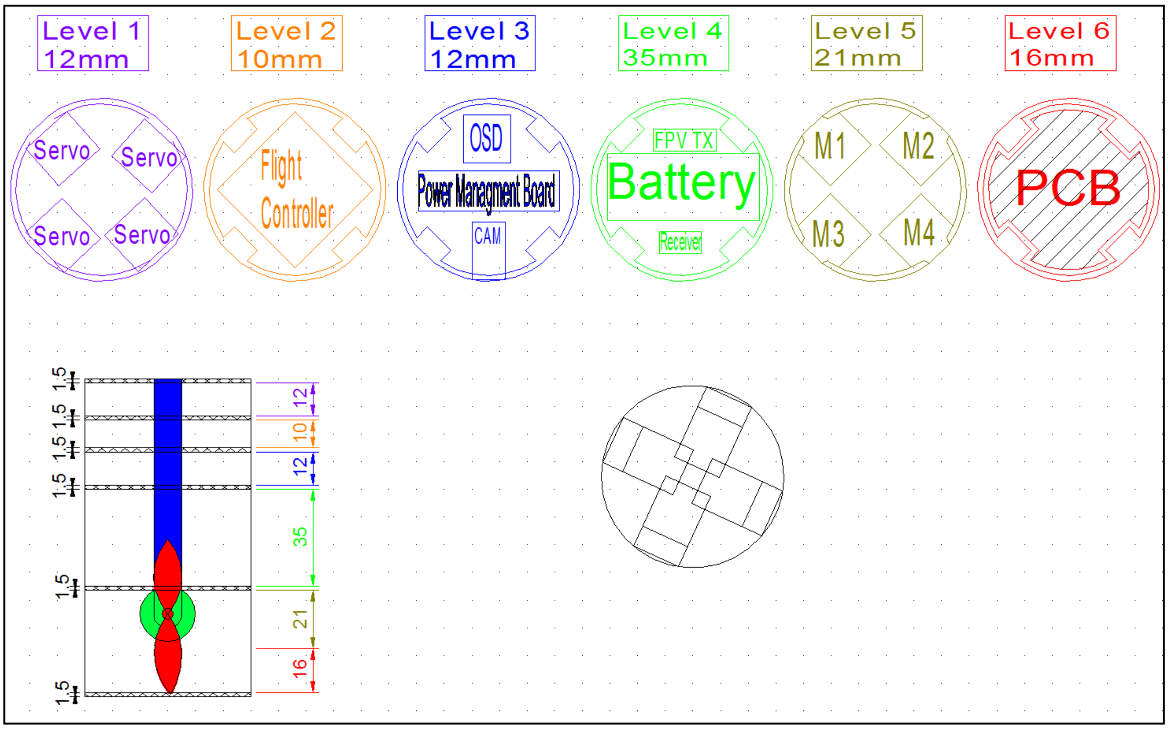


Figure 14[2]

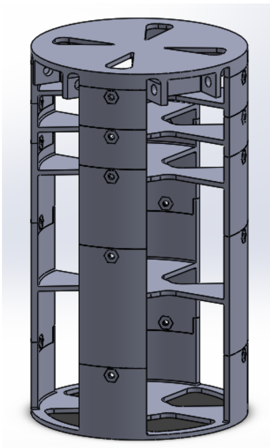


Figure 15

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 16. 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.

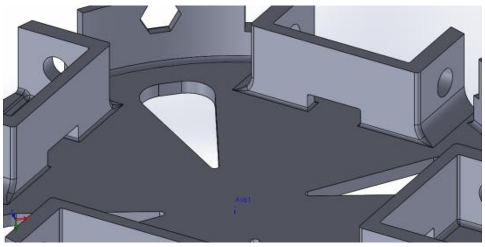


Figure 16

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

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 17). Through the teeth can be seen holes which are for pins, so that we can secure the layers together.

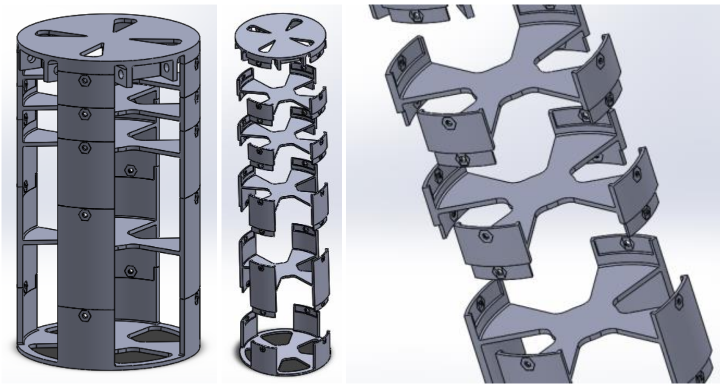


Figure 17

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 18) 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 13.

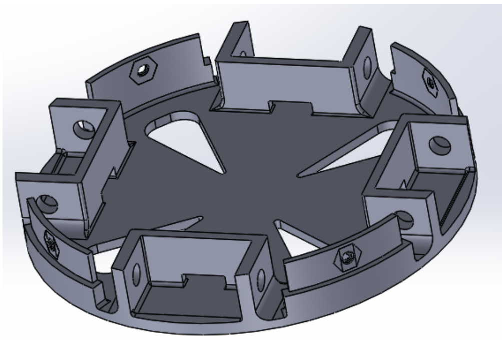


Figure 18

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 19). 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.

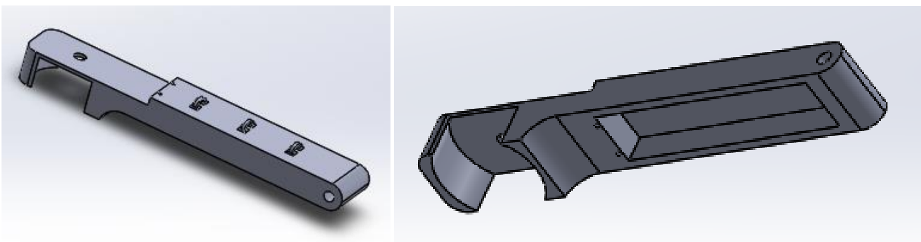


Figure 19

When constructed in SolidWorks the model looked as follows:

Figure 20

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:

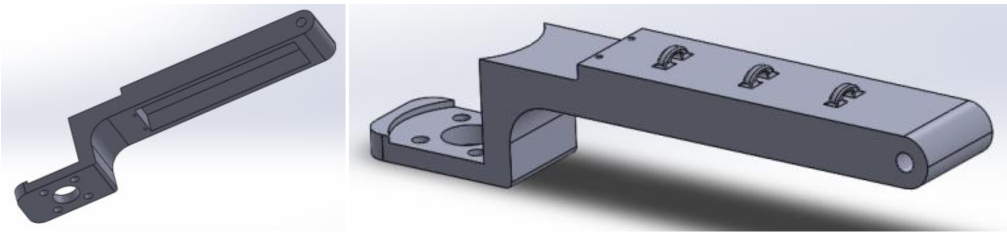


Figure 21

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.

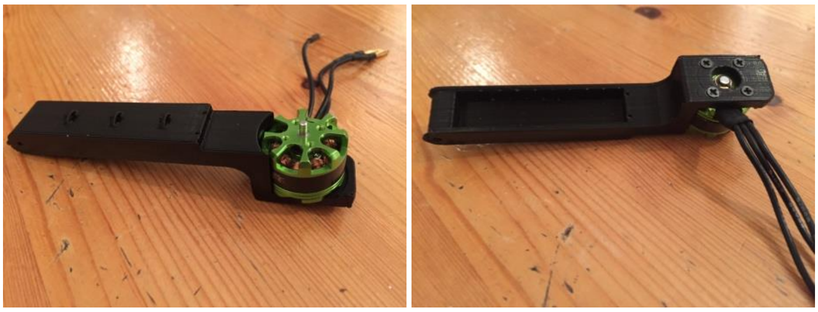


Figure 22 – motor screwed securely into arm

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 23). 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 24 (before) and 25(after)). Thus even when the pieces became wider due to the inaccuracies of the 3D printer the pieces still slotted together.

Figure 23

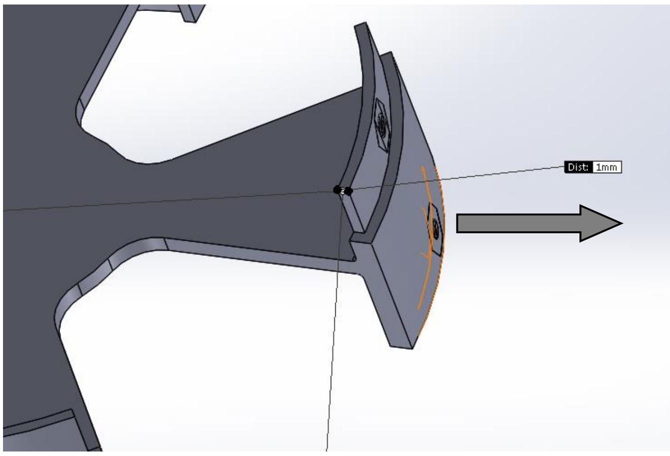


Figure 24

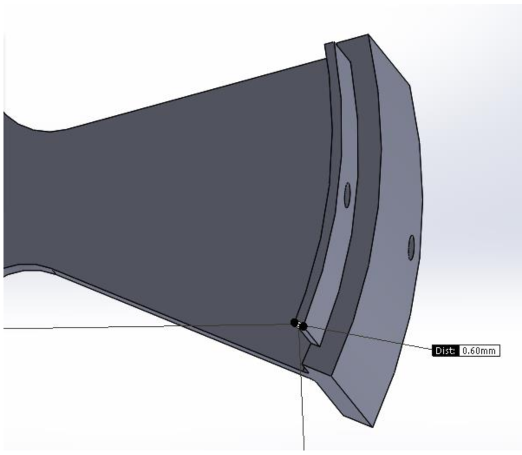


Figure 25

In the past few days we have 3D printed two layers of this improved design, which as can be seen in Figure 26, slot together well. In the next few days we hope to print the rest of the layers.

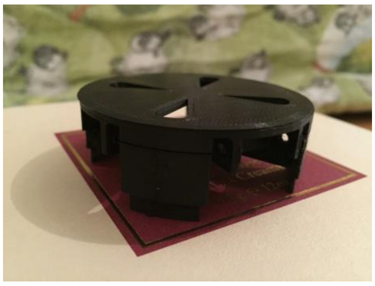


Figure 26

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

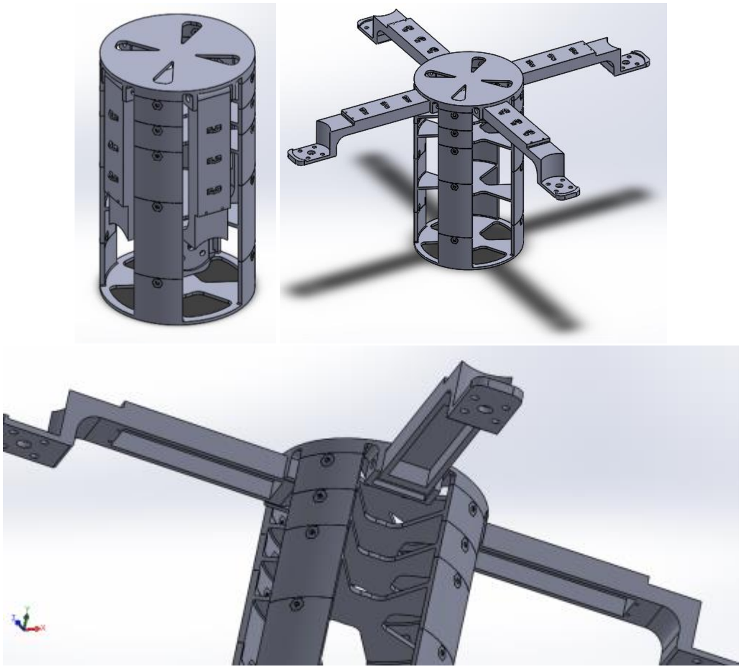


Figure 27

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.

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 single motor, 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).

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.

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 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

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

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 31), we have designed a power distribution PCB, (see Figure 28) 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.

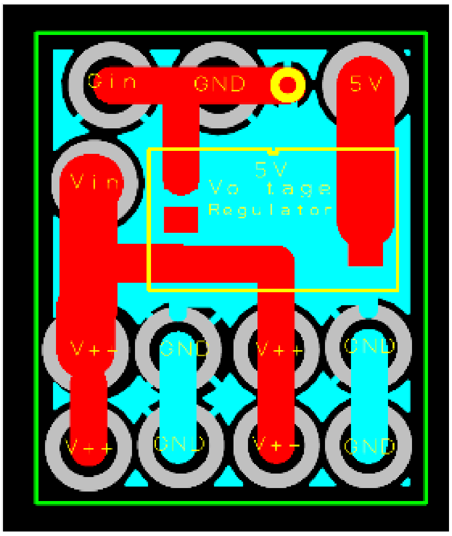


Figure 28: Power Distribution PCB (designed using DesignSpark PCB[4])

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. 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 would struggle with size. The Turnigy Outrunner v2 motors were selected because they were the smallest motors we

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

	could find but one of the most powerful. Though normally very expensive, the sponsorship by HobbyKing (from whom we purchased the motors) meant that they were the best choice.
Control Board – HobbyKing i86[6]	This control board was chosen as it is very simple, with little to go wrong, while having sufficient features, such as auto-stabilisation built in, using a barometer. Additionally, a member of our team (Daniel) had used the board before and had found it very easy to use. Furthermore, we could not purchase a more complex control board because of space concerns which meant the 40x40mm size of the board was very helpful.
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.
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 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

Sensor System

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

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

also chose to use the **Hope RF98W**[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 RF98W 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. Finally, we will be using the **DRV8833-PWR**[13] as a Motor Driver for the motor that opens the arms, since it was found to be simple and effective by Team Impulse. Though the even more simple L2D93D was being considered, it was discovered that we do not have the space on the PCB for this chip.

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**[14] 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[15] produced by Sparkfun, to help use the module, which has proprietary connectors, and makes use of 1.8V logic. Finally, we are going to use the **Sparkfun 9DOF Breakout**[16], as an IMU, containing a 3-axis accelerometer, 3-axis gyro and 3-axis magnetometer. This is 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.

PCBs

The main sensor system PCB is shown below:

It was designed in RS / Allied DesignSpark PCB 6[17]*. All PCB files are available on the GitHub repository*[18]*, under the Electronics Design section.*

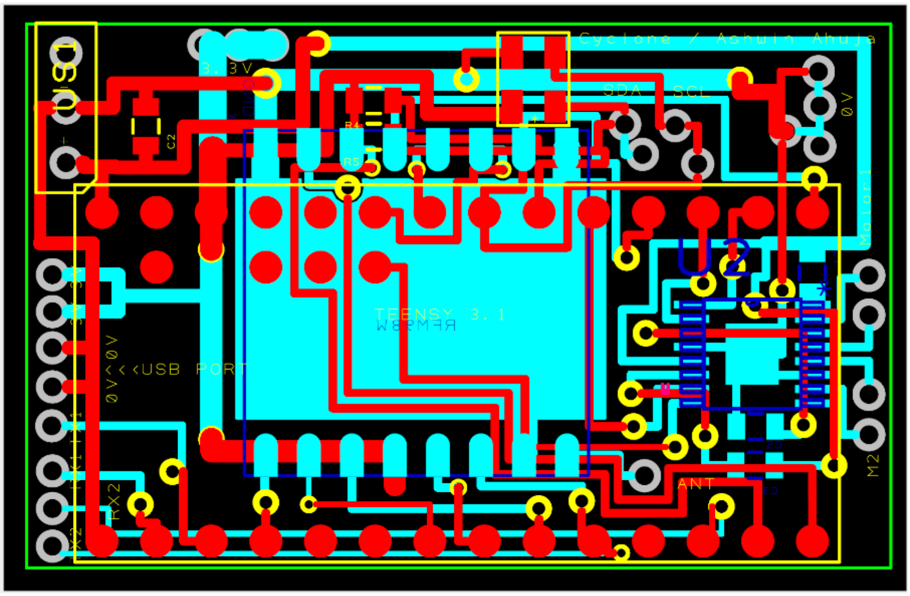


Figure 30: The design for the main sensor PCB

[12] http://www.hoperf.com/rf/lora/RFM98W.htm
[13] http://bit.ly/1M4X4Ln
[14] https://www.sparkfun.com/products/10890
[15] https://www.sparkfun.com/products/10995
[16] https://www.sparkfun.com/products/10736
[17] http://www.rs-online.com/designspark/electronics/eng/page/designspark-pcb-home-page
[18] http://www.github.com/CycloneCanSat

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 RF98W, 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. 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 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.

Camera System

For the camera system, we have chosen to use a pre-made FPV system, with the **Mini FPV Camera**[19] combined with the **HobbyKing FPV Transmitter**[20]. 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[21]. 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**[22]), we feel that it is actually inefficient, and in future revisions we hope to include a custom OSD in the power distribution board.

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 **HobbyKing Control Board**[23], 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.

[19] http://bit.ly/1PYEpTe
[20] http://bit.ly/1RhYGBA
[21] 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
[22] http://bit.ly/1XDTcoH
[23] http://bit.ly/1PUlKbW

Finally, for the sensor system, we are going with the tried-and-tested 434 MHz RF transmission frequency, given the high penetrating power 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[24], a safe bandwidth would be around 10kHz, thus the nearest setting, 7.8kHz will be used. Making use of the LoRa Modem Calculator Tool made by SemTech[25], this will allow a data rate of 417 bps, assuming the lowest possible Spreading Factor (6), with a link budget of 150 dB and a resistor sensitivity of -130 DBm.

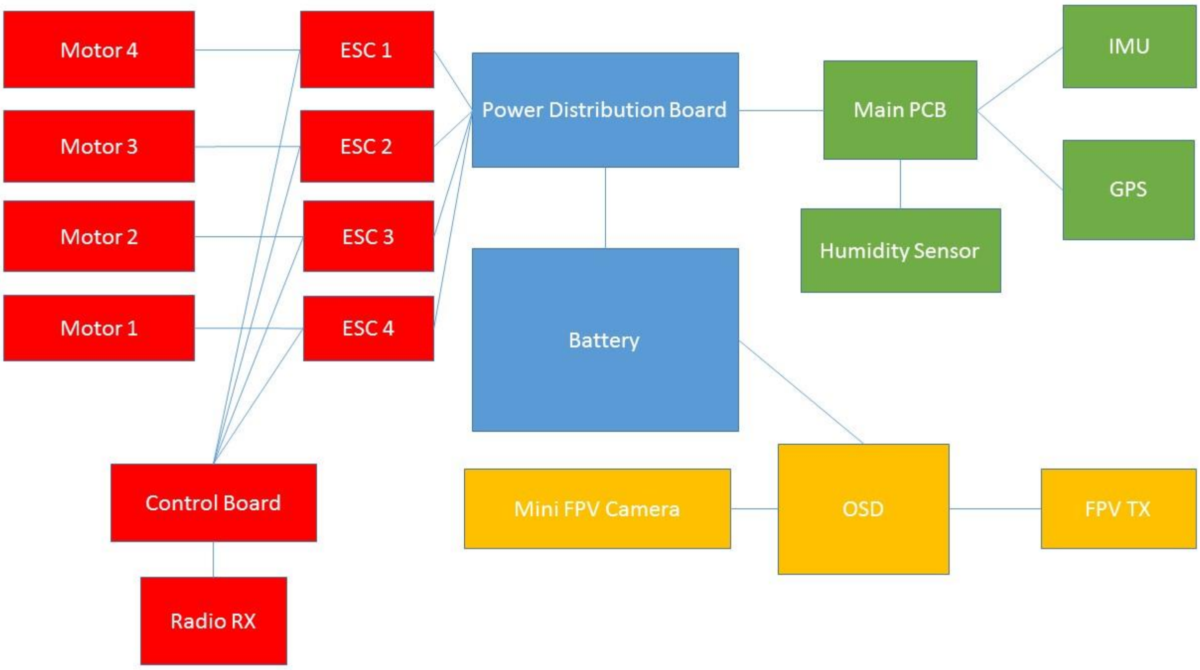


Figure 31: Diagram of system wiring (made using Microsoft Powerpoint[26])

[24] According to the spreadsheet of allocated frequencies provided during the 2015 CanSat Teacher's Workshop
[25] http://www.semtech.com/apps/filedown/down.php?file=SX1272LoRaCalculatorSetup1%271.zip
[26] https://products.office.com/en-gb/powerpoint

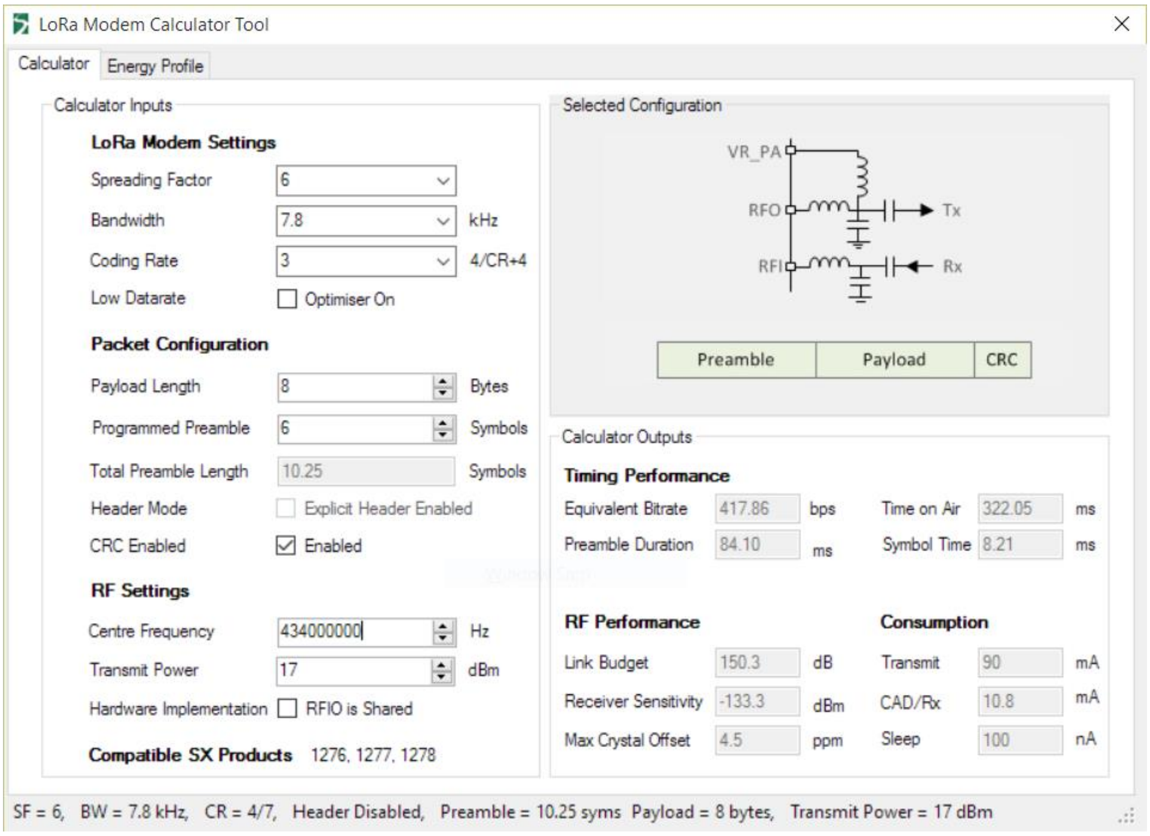


Figure 32: Calculations for Hope RF98W, made using the SemTech LoRa Modem Calculator Tool

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.

Algorithms

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.

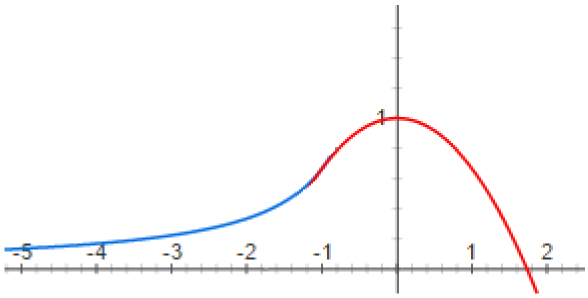


Figure 33: Graph used to find Agricultural Viability from temperature

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.

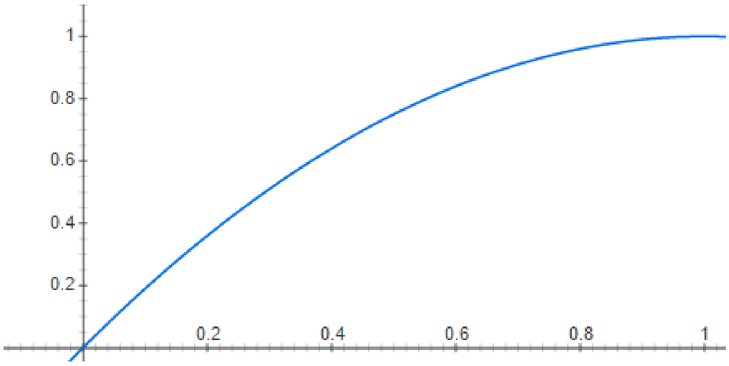


Figure 34: Graph to find Agricultural Viability using humidity

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.

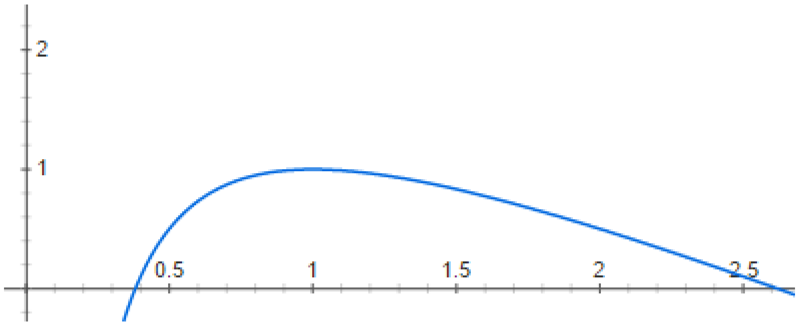


Figure 35: Graph to find Agricultural Viability using Barometric Pressure

Summary:

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

where;

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

Given the equation, we now hope to continue to research this and test it against existing yield data from online sources. Thus, we hope to make small adjustments to calibrate it and make it more accurate.

Website

Over the summer, the Software Team has worked hard to design a 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[27], 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[28]. 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[29], 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 it's market is too 'niche'[30]. However, we hope to make the app more general for engineering and especially St Paul's in order to counter this problem.

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 are going to 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. For the GPS sensor, we are going to use the TinyGPS++[31] (an open source GPS library) to parse the NMEA statements that the breakout produces. This library is very 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 RF98W in the actual program. However, for this, 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. To date, much of the MS5637 library has been written, but much work must still occur.

[27] Statistics found using Google Analytics
[28] Statistics found using Google Analytics
[29] Statistics found through analytics on the Google Play Developers' Console
[30] 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."
[31] http://arduiniana.org/libraries/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.

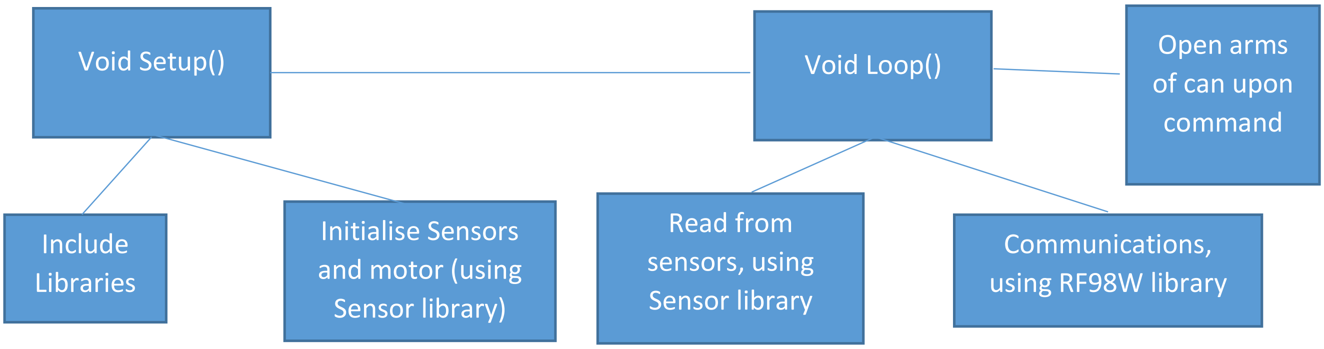


Figure 36: Design for Can code (made using Microsoft Powerpoint[32])

Though there is a lot of work to do, we believe that with much of the software team freed from electronics duty, this could happen fairly quickly, allowing us to have a working prototype of the completed code by Christmas, the original aim.

Base Station

Little work has been completed on the Base Station software, beyond deciding what the base station software entailed. However, we have now decided to write the software in C#, probably using the very popular GTK+ framework[33], due to the familiarity of the main coder on the project with this, with in fact, Team Impulse's Base Station Software having been written using this. However, beyond receiving the data, the base station has many complex objectives.

The specification is as follows:

1. The software should parse and store sensor data from the Can (in a sensible file format)
2. The software should show the live camera feed from the Can
3. The software should display the current Can location, preferably on a map (making use of Google Maps[34] or OpenStreetMaps[35])
4. The software should use the incoming data to calculate altitude (using barometric pressure) and the agricultural viability (using the predefined algorithm)
5. The software should allow you to open the arms of the can, by sending a command over the RF98W
6. The software should be able to send commands to the control board of the quadcopter
7. The software should be able to autonomously monitor the position of the quadcopter and move it to a user-defined location, using (6)

[32] https://products.office.com/en-gb/powerpoint
[33] http://www.gtk.org/
[34] http://maps.google.com
[35] http://www.openstreetmap.org/

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 de-risked 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.

Gantt Chart

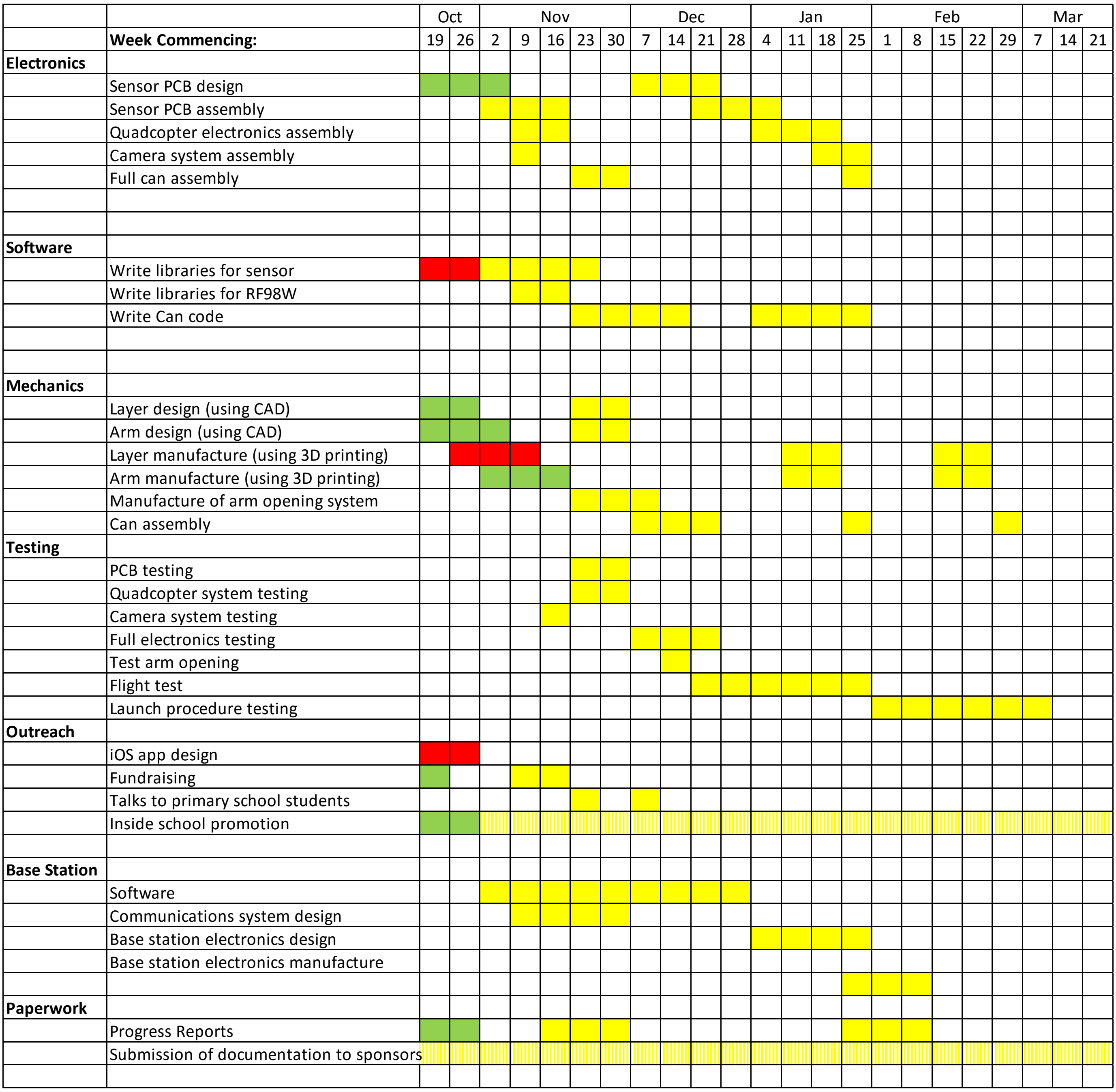


Figure 37: Gantt Chart, outlining future plans (made with Microsoft Excel[36])

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

Mission Criteria

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.

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.