Flare - A solar powered plane

On its construction and control

Casper Cromjongh
Jasper Rietveld

TU Delft

Flare - A solar powered plane

On its construction and control

by

Casper Cromjongh
Jasper Rietveld

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on Friday July 5, 2019 at 12:00 PM.

Student number:	4494156, 4581881	
Project duration:	April 23, 2019 – July 5, 2019	
Supervisor:	Dr. O. Isabella,	TU Delft
Thesis committee:	G. R. Chandra Mouli,	TU Delft
	M. Popov,	TU Delft
	J. Bastemeijer,	TU Delft

An electronic version of this thesis is available at `https://repository.tudelft.nl/`.

TU Delft

Abstract

This BSc Thesis is about the construction and control elements of the airplane group of the Solar Drone project. The project aims to improve the flight time of aerial vehicles, using solar energy. This document is delves into the possibility of applying solar cells to small model aircrafts. It outlines the challenges and solutions involved in building and controlling such a system. A ready-available airframe and fitting components will be used, in combination with a custom photo-voltaic and energy system, to create a prototype. The prototype takes the form of an unmanned aerial vehicle, making it capable of extended flights without requiring extensive monitoring. It will serve as a testing platform for the performance of the PV modifications done to the aircraft.
Included in this document are simulations for calculating optimal solar panel placement inside a transparent wing with opaque ribs, in addition to software and hardware modifications that were required to make the other systems work well in combination with the custom energy system.

Preface

This report is the thesis of the control group of the Bachelor Afstudeer Project called ”*Solar Drone*”, supervised by Dr. O. Isabella at the TU Delft. It provides a view into the research done, decisions taken, prototypes built and tests conducted during the project. We hope it will provide insight to anyone interested in this topic.

We would like to thank J. Koeners for his support in bureaucratic processes and practical help during the project, aiding the realisation of the prototype.

Casper Cromjongh
Jasper Rietveld
Delft, June 2019

Contents

1 Introduction

1.1. State-of-the-art analysis

Drones are becoming ever more popular due to reduced costs and improved technical systems. All quad-copters that can be bought today include some form of autonomy in control for e.g. stabilisation, attitude control or waypoint navigation. This shows that Autonomous Unmanned Vehicles (UAVs) are an important topic today.
In parallel, solar energy has also seen dramatic growth in the past decade. Global warming hazards, reduced cost and energy delivery to remote places have contributed to this growth. A great deal of research exists on the subject of solar panel types and efficiency.
The two fields of research have also been combined in the form of vehicles with solar panels that are able to charge the batteries on the go. Taking it a step further, solar technology has also been applied to airplanes. Solar powered and electric airplanes have a long track record of being an object of interest and study of the academical community. See for example [13]. Electrical planes spark the imagination, being hight-tech, silent and more environment friendly.

Planes capable of continuous, 24/7 flight have already been constructed. See for example the Sky Sailor [13] or the Helios, theoretically [12]. These planes have been custom designed and built from scratch for the soul purpose of achieving long flight duration or distance. While these efforts were in part successful of managing continuous flight, there is still a long way to go. Commercial products are not yet available, and manned electrical flight is even further away [11]. It is clear that to reach the point of electrical flight in the daily life requires a lot more research to be done.

1.2. Problem definition

The projects outlined in the previous section have all been custom designed and built from scratch. This is not feasible for many people who would wish to get into contributing to this topic. Another motivation is the short flight time electrical airborne vehicles typically have. Most electrical planes can stay in the air for half an hour at most. It would be a good step forward to use solar energy to increase that number.
The scope of this project is therefore to design a method to modify an existing plane or air-frame kit to enhance its endurance capabilities using solar cells and readily available components.

1.3. Thesis synopsis

As can be seen above, this document starts with a motivation to build a solar powered airplane based on readily available components. Thereafter, a definition of requirements is given that the design should follow. The main body consists of design considerations on the different aspects of the construction, electronic hardware, and software of the plane. The chapter starts by stating which components and systems are needed by looking at the requirements. It then gives a description of the design problems, possible answers and finally choices concerning these topics.
The thesis will then turn to the production of the prototype, based on the plans laid out in the design

chapter. It will explain the steps taken to materialise the design. Problems encountered are analysed, and if possible, overcome by adapting strategy. Finally, results of the prototype, tests and simulations will be discussed, followed by a conclusing chapter.

2

Programme of requirements

To be able to create a design an airplane equipped with solar cells, we need an extensive set of requirements. This section should provide a guide to the efforts spent designing. To achieve this, one should research the functionality and goals of the plane that are desired. Once that is clear, strict requirements can be formulated that are used in the rest of the project.

2.1. Mandatory requirements

Mandatory requirements are rules that the product design *must* adhere to. If not, the project is considered failed. Such requirements for the to-be-designed airplane are given below.

2.1.1. functional requirements

1. The UAV must be capable of powered flight
2. The plane must be capable of unmanned flight
3. It must be able to monitor the performance of the plane from a ground station.
4. The UAV must be powered through solar power
5. The flight time of the UAV must be greater than a comparable UAV without solar power.
6. The plane must be able to get to ground safely without motor power
7. The plane must have the option to be manually controlled
8. It must be possible to monitor the position of the plane at all times
9. The UAV must be suitable for flight within the legally allowed heights.
10. Transmissions made by the UAV must be within legal frequency bands.
11. The plane should be operable without specific safety gear

2.2. Trade-off requirements

Trade-off requirements are clear goals, but less strict. They form a combination of things that would benefit the plane design in its use and optimisation goals

2.2.1. functional requirements

12. Maximise flight time. **Functional**
13. The software should allow logging of flight data. **Functional**
14. The software should allow sending flight data to a ground control station. **Functional**

15. The plane should preferably be able to transmit live video to the ground station. **Functional**
16. The response time of the UAV should be as low as possible. **Functional**

2.2.2. non-functional requirements

17. The UAV should preferably be visible from the ground.
18. The UAV should preferably be made from sustainable materials (that is, durable, recyclable of renewable materials)
19. The plane should preferably not compromise the privacy of people in the surveyed area.

3

Design process and choice justification

3.1. System overview

3.1.1. Design approach

The goal of the project, as described in the Programme of Requirements, is to create a plane-type UAV that is capable of powered flight using power supplied by solar panels. A second requirement was that the plane should be built from parts that were commercially available. Two approaches were proposed to fulfil these requirements. The first approach is to buy a commercially available plane, and then make alterations to its structure and hardware so that the components needed for the use of solar power can be integrated into the plane. The second approach would be to build the plane from scratch, selecting commercially available parts to create a working plane, and then integrate the hardware for solar power within the design of the plane itself.

In order for the project to be successful, it was determined that the second approach should be taken. The first reason for this is that alterations to the hardware of commercially available plane-type UAVs may pose significant problems, as most of these planes use integrated circuitry without detailed documentation. In order to make alterations to such a system, either a large portion of the hardware of the plane has to be reverse engineered, or components will have to be replaced altogether. This problem becomes more prevalent considering that for most commercially available UAVs, no information about the implementation of the internal hardware can be accessed before the product is purchased. Only at this time could the plane be inspected to determine what changes have to be made to it in order to allow the plane to work on solar power. This may delay the design process considerably.

A second reason to build the plane from scratch is that it allows the hardware required to use the solar panels to be integrated into the plane from the design phase, which allows the plane to be more fit for its intended purpose. A downside of this approach is that more components have to be selected.

3.1.2. Required components

What components need to be included into the design becomes clear from the Requirements determined in section 2. This section will go over what components are required, the choices regarding the actual component selection are described in detail in Section 3.2 on the component choices.

The first component to be required is the plane itself. The plane required is an empty plane without electronics, with the only function to house all electronics and to generate lift if propelled forwards.

To move the plane forwards, some sort of propulsion has to be used (req. 1). Since it is required that the system is powered by electricity, a form of electric motor has to be chosen. This motor has to be controlled by means of an electronic speed controller, from here on referred to as the ESC.

The plane will have to be able to turn in the air to fly well. Most plane-type UAVs are controlled by moving a series of flaps to change the aerodynamics of the wing. Otherwise, the motor itself is moved to direct the plane. Usually, servo motors are used to control these flaps or move the motor. These servos will also have to be selected for their purpose.

In order to meet the requirement for unmanned flight (req. 2), a flight controller has to be implemented. This flight controller will be connected to the servos and the ESC of the motor to allow it to control the plane.

The flight controller will also require a number of sensors in order to operate. In order to navigate, a GPS and magnetometer have to be used. An accelerometer and a gyroscope also have to be included to measure the rotation and acceleration of the plane. These are, however, often already integrated within the integrated circuitry of the flight controller. Often, the flight controller also contains sensors to measure the battery level.

If a connection to a ground station is required, as is the case here (req. 14), either to relay sensor data or to receive commands such as way-points, a telemetry module should also be installed. Lastly, an airspeed sensor could be installed to measure the speed of the aircraft relative to the air it is passing through. This is important data, as this speed difference is what causes the aerodynamic shape of the plane to generate lift.

Another requirement of the plane is that it can be manually controlled (req. 7). Since the plane is unmanned, a type of remote control should then be implemented. This system has to consist of some form of transmitter and receiver where the transmitter sends control input from the user to the receiver which in turn controls the plane based on these inputs.

For safety requirements, it is required that it should be possible to monitor the performance and positions of the plane at all times (req. 8, 3). This requires a series of different components. Firstly, a set of sensors has to be included that can be used to monitor data regarding the performance of the plane. Sensors should be chosen to measure the power usage for different modules of the plane, as well as data from the power received from solar power. Data from the sensors used by the flight controller, such as the position, speed and the battery level also provides useful information on the performance of the plane. This data also has to be transmitted via a telemetry module.

Lastly, it may be useful to relay real time video from a camera on the plane (req. 15). This can be useful when controlling the plane manually, or in order to get a clear view on what the heading of the plane is. Care should be taken when this is included to make sure that the recorded data is used in accordance to the legislations regarding video surveillance in the areas of operation for the UAV in order to respect to the privacy of any people in said area.

3.1.3. Control overview

All the aforementioned hardware can be implemented by connecting a number of separate modules to a flight controller. The implementation that was decided on is displayed in Figure 3.1. The communication protocols used by the flight controller to interface with the different modules is displayed in the graph. The choice of these protocols depends mostly on the selected components, as described in Section 3.2 on the component selection. The only components that are kept separate from the flight controller are the FPV camera and transmitter. These modules can operate separate from the flight controller itself and also use their own communication channel, so that streaming the video through the flight controller is not necessary.

3.1.4. Communication overview

The communication was split into three different telemetry channels, namely the channel for manual control, the channel for use in autonomous flight and the FPV data. An overview of the communication network is seen in Figure 3.2.In this distribution of the channels the communication for the different protocols is kept separated on different hardware channels rather than being multiplexed through the same channel. While multiplexing this software on the same channel would reduce the hardware costs, it would also decrease the robustness on the system, as there would be no guarantee that all channels would have enough required bandwidth.

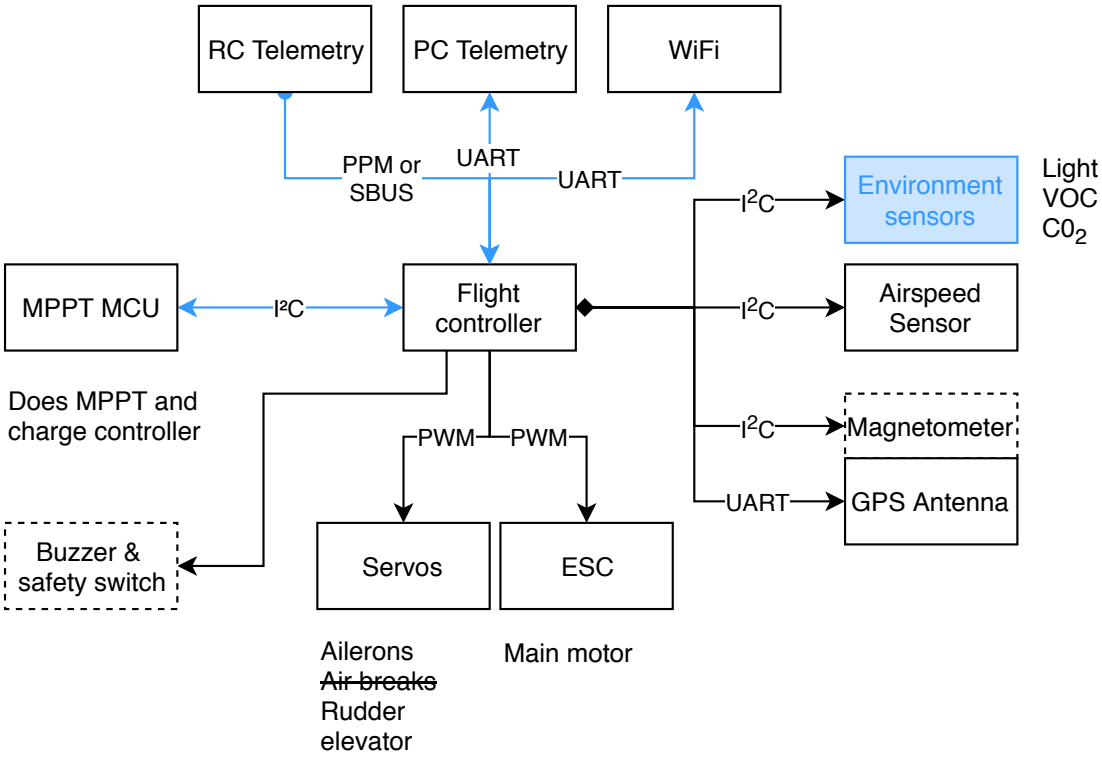


Figure 3.1: Flight controller wired connection diagram

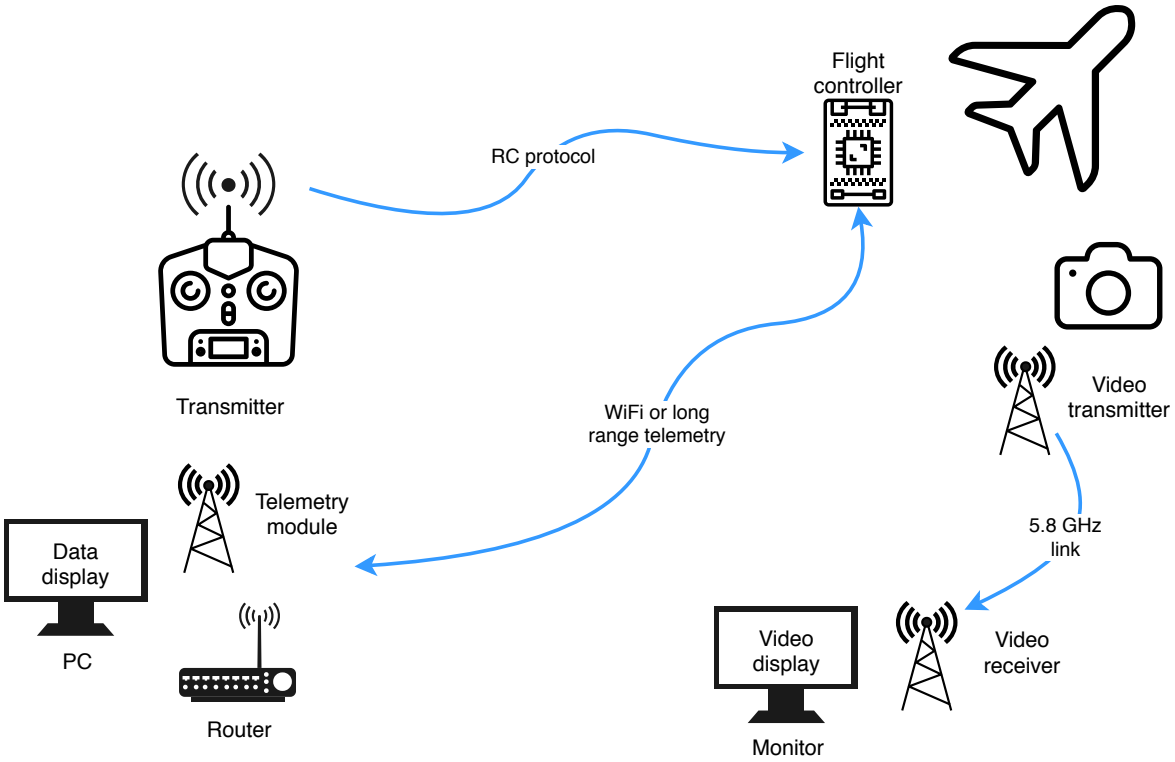


Figure 3.2: Flight controller wireless connection diagram

Figure 3.3: Top view of the airframe

3.2. Component choices

3.2.1. Airframe

The first component to be selected is the airframe. This is selected first, because all other electrical components have to be fitted to the airframe. To select a plane in accordance to the programme of requirements, the airframe is to be chosen based on its weight, wing area and aerodynamic properties.

Two quantities that are often used when describing properties relating to the weight and wing area of a plane are the aspect ratio of the plane and the wing loading. The aspect ratio is defined as the ratio between the area of the plane and the span of the wings. For rectangular wings shapes, this is equal to the ratio between the length and the width of the wing. The wing loading is defined as the ratio between the total weight of the aircraft and the area of its wings. In order to get an increased flight time, it is desirable to have a minimal wing loading with an aspect ratio as large as possible.
A wing with a larger aspect ratio has less induced drag than a wing with a lower aspect ratio of the same area. This is because the lift-to-drag ratio of airplanes scales with the aspect ratio of the plane [9]. The maximum lift-to-drag ratio is given by

$$\left(\frac{L}{D}\right)_{\max} = \frac{1}{2}\sqrt{\frac{\pi \epsilon A}{C_{d,0}}}$$

Where A is the aspect ratio, ϵ is the planes efficiency factor, and $C_{d,0}$ is the drag coefficient for the plane. A larger wing usually also decreases the manoeuvrability of the plane, but this is off less importance to the requirements as the benefit of the increase in efficiency. A smaller wing loading will mean that less energy is required to fly the plane, which will in turn increase the range. The downside of a reduced wing loading is that planes with a low wing area usually have a lower structural integrity. The plane type that usually has the highest aspect ratio as well as a low wing loading is the glider. For this reason, we considered a number of different gliders-type airplanes for the project.

In order to obtain the airframe, one of two options was considered. Most commercially available gliders are complete foam planes, with the electronics included within them. This type of plane could be bought, and then the electronics could be replaced. However, this approach was not chosen for the same reasons as given in Section 3.1.1. The other commonly bought commercially available airframes come in the form of balsa wooden kits. Examples of the two different plane types are shown in Figure 3.4. The balsa-wooden airframes usually are delivered as kits that have to be constructed manually

(a) A foam frame example

(b) A balsa kit example

Figure 3.4: Airframe options

Name	Wing span [mm]	Wing area [dm^2]	Weight [g]	Area/weight [cm^2/g]	Leveled flight factor	Airfoil
Scarlet Elektrosegler	3000	unknown	1450	3.9	unknown	MH-30, SD, RG7073 RG14.5
Neo Triple Thermic	2580	47,00	980	4,80	26,81704926	
Triple Thermic	2550	47,00	580	8,10	15,5952959	
Starfly E-segler	2500	46,95	1600	2,93	41,87366788	
Neomi Elektrosegler	2500	53,50	1200	4,46	22,65722951	MH-30, SD, RG7073 RG14.5
Xenon Elektrosegler	2500	53,50	1200-1500	3,96	25,4893832	
Galaxy Elektrosegler	2000	42,00	900-1150	4,00	25,97832027	
E-Flite Radian XL	2600	unknown	2254	unknown	unknown	unknown

Table 3.1: Different balsa kits in comparison

and then covered by a plastic foil to get the desired aerodynamic properties. The benefit of such an airframe is that it is highly customizable and relatively well priced. The main downside is that this type of airframes usually require a lot of work. It was decided that the benefit of the customizability outweighs the downsides. For this reason, a number of balsa wooden kits were considered. Some data on the kits that were considered is seen in Table 3.1.

To get a sense of the viability of using a given airframe for this project, a constant that will be called the *level flight constant*, k is introduced. This constant is used as a factor to compare of different airframes. As will become clear from the derivation, the exact value is not relevant, but rather the relative value compared to other airframes. The level flight constant is given by equation 3.2.

$$P_{\text{solar}} = I_r \cdot \sin(\theta_i) \cdot A \cdot \varepsilon \tag{3.1}$$

A can be roughly estimated by $l_{\text{wing}} \cdot l_{\text{chord}}$. To continue the derivation, the power for level flight is required. [13, p. 39] shows this quantity is proportional to $\sqrt{\frac{l_{\text{wing}}}{l_{\text{chord}}}} \cdot m^{2/3}$. By combining that with the estimation of A, it can be said that P_{level} is approximately proportional to $l_{\text{wing}} \cdot \frac{m^{3/2}}{A^{3/2}}$. If this proportional factor is divided by proportional relationship between P_{solar} and the area the factor F comes out as in equation 3.2.

$$F = l_{\text{wing}} \frac{m^{3/2}}{A_{\text{wing}}^{5/2}} \tag{3.2}$$

I_r is the irradiance in $\mathrm{W\,m^{-2}}$, A the wing area in $\mathrm{m^2}$, while l_{wing}, l_{chord} are the wingspan and wing chord respectively in m. m is the mass in kg, θ_i the angle of incidence of the sun and ε the solar cell efficiency.

The different kits in Table 3.2.1 have been compared. At first, the *Triple Thermic* kit seemed like the optimal choice. After a closer analysis, it seemed that that the fuselage of this model has very little space available in the width. To make sure that the airframe would have enough space to house all electronics, it was therefore decided that the newer version of this airframe, namely the *Neo Triple*

Thermic should be chosen. This plane has a large plastic fuselage that could house all electronics with more ease.

3.2.2. covering foil

The balsa wooden kit is to be covered by means of a type of foil that is ironed onto the airframe to enclose the wings and tail. This process will give the wings their shape and aerodynamic properties. The selection of this foil is also of importance for the performance of the plane. Firstly, different thicknesses of foil are available, each with different weights and strengths. Secondarily, different colours of the foil are available.

The choice of colour influences the functionality of the plane Since the solar panels are to be integrated within the wing, as will be reasoned in Section 3.3, the covering foil for the wings should be transparent so that light can reach the panels. For the other sections of the plane that have to be covered, a bright orange colour was chosen with a good contrast against the sky, so that the plane is easy to see when in flight. This is in line with the requirement that the plane should preferably be visible from the ground, as given in requirement 17.

The choice for the thickness was dependent mainly on the strength and weight of the foil. The company that manufactures the foils that we were considering to buy provides a useful table which lists the weight per area of all the different foils [7]. As mentioned, the wings were to be covered with transparent foil to let light through to the PV panels, and any other area should preferably be covered in the orange foil for visibility. The foils we considered to buy were of the Iron-on type, referred to as ”Bügelfolie” in the table. The company manufactures two different types of the foil, a normal foil type and a light foil type.

For transparent foil, the lightweight foil has a weight of 36 g/m^2 as opposed to 54 g/m^2. The lightweight orange foil has a weight of 44 g/m^2, whereas normal orange foil has a weight of 77 g/m^2. The area of the wings was stated in the data-sheet provided for the plane as 47 dm^2. Since both the top and bottom of the wing have to be covered with the transparent, at least 94 dm^2 of this foil was required. It was estimated that at most 50 dm^2 of the orange foil had to be used to cover the wing. Using these figures, the difference in weight between the two foil types could be calculated. For the light foil, the estimated weight would be

$$94 \cdot 36 \cdot 0.01 + 50 \cdot 44 \cdot 0.01 = 55.84\,\mathrm{g}$$

For the normal foil, the estimated weight would be

$$94 \cdot 54 \cdot 0.01 + 50 \cdot 77 \cdot 0.01 = 89.26\,\mathrm{g}$$

Resulting in a difference of $33.34\ g$. It then had to be decided whether this reduction in weight was worth the difference in strength of the foil. The product information as provided by the company specifies the tensile strength, the maximum force applied to the foil breaks as opposed to stretching, as $283N$ for the heavier foil and $150N$ for light foil, per $50mm$ [6]. Considering that the plane may be used outside, it was decided that this decrease in tensile strength was of higher importance than the reduction in weight, so the normal cover was selected.

3.2.3. Plane electronics

The plane is controlled by means of the *elevator*, *rudder*, and two *ailerons*. Each of these flaps is controlled by a servo, for a total of four servos. The choice of these servos is primarily constrained by the dimensions of the servo housings of the airframe. Other properties to consider are the torque the servo can provide, the speed at which the servo can move and their weight. The rotation speed of the servos will hardly contribute to the planes capability of flight, but faster servos will increase the responsiveness of the plane. It is, however, required that the servos are strong enough to hold their position and move the rudder at all times. If the servo would not be strong enough to move the flap while a great force is applied to it, such as through wind or a high velocity, the plane may lose control.

Because of the severity of a control failure, it was decided that an increase of the servo strength was preferred over a reduction in weight. This also adds some robustness against forces such as high wind to the plane. This choice was further reinforced by the metric that servos with a small increase in

Motor	kV rating	Max current [A]	Weight[g]
Roxxy C35-30-45	300	10	78
Roxxy C28-34-12	750	19	67
Roxxy C28-27-24	760	10	57
Turnigy D2836/11	750	20	67
Graupner Speed 400	400	?	72
Graupner Speed 500	500	?	72

weight usually have much greater strength, it was chosen that it was permissible to use stronger servos.

Propeller The propeller choice was also dependent on the airframe. The company manufacturing the airplane had specified a lower and upper bound on the propeller dimensions, based on the how well the plane can be controlled. It was chosen to order the largest available propeller within this specified range, as larger propellers are generally more efficient.

Motor The choice of the motor also depends heavily on the airframe. Because the plane will fly at a relatively low velocity with a large propeller, the torque provided by the motor is of higher importance than its rotational speed.

Battery The battery of the system is chosen in cooperation with the PE group. See their report for details.

Electronic Speed Controller Electronic Speed Controllers (ESC) are used to drive a brushless motor. They require power and a control signal from either an RC receiver or a flight controller. The power comes directly from the battery. This is because the motor can require great currents during acceleration, when the required force is great. Many ESCs include a Battery Eliminator Circuit (BEC) that can be used to power other electronics in the system. In this project, the BEC will not be used, as most sub-components are low-power and will be powered by the flight controller. The control signal is generally a PWM signal, which will be generated by the flight controller.

Sensor choices For sensors such as the GPS and airspeed sensor, the main properties of interest of the plane are the accuracy, weight and dimensions of the sensor. Ideally, one would make a trade-off between these three properties. However, upon evaluation of the market it became clear that the availability of the components was limited when also requiring timely delivery. Therefore, the price and delivery time for these components were of increased influence for the sensor selection.
The flight controller (section 3.2.5) includes a lot of sensors by itself. On-board are:

- Barometer (MS5611)
- Accelerometer (ICM-20608, MPU9250)
- Gyroscope (ICM-20608, MPU9250)
- Magnetometer (MPU9250, HMC5983)

It is desirable to measure the light intensity the solar cells could receive. Not all light sensors are suitable for this purpose. The mission will be carried out in direct sunlight conditions. Therefore, the light sensor should be able to handle great luminosity without saturating. On a bright day, light intensity can get as high as 100.000 lux .No sensors were found capable of measuring such a high irradiance. The best sensor that was found (and chosen) is the TSL2591, capable of reading up to 88 klux. Thus, the irradiance won't prove too much of a problem, since the sun will not always be directly above the plane and the margin is not too large. This sensor has the added benefit of being able to measure the light intensity of both the visible spectrum and IR light, giving more information about the environment.

GPS module Most GPS (Global Positioning System) modules communicate over UART using the NMEA protocol, which is generically explained in [1]. In contrast to the paper, GPS modules today generally use a baud-rate of (at least) 9600 symbols/s. The precision and sensitivity of GPS modules varies. Typically GPS modules have a imprecision of a few meters (< 10) in radius. Other options for position determination exist, such as RTK-GPS[1] which use a ground station communication system that offers very high precision (centimetre level). These systems are very expensive, however. For the purposes of the airplane that is the goal of this project, a normal GPS module will suffice.

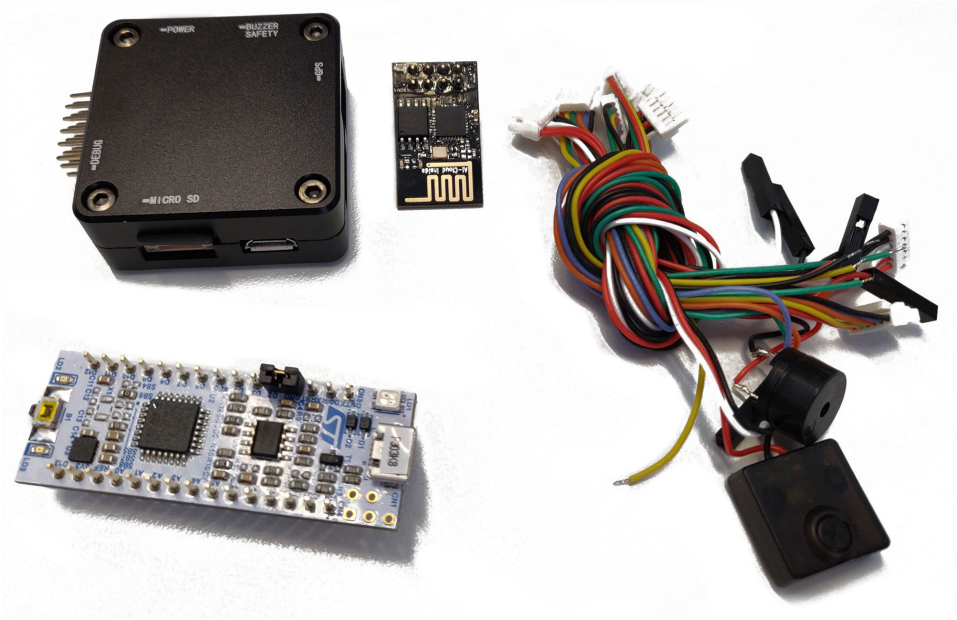


Figure 3.5: Some electronics used in the airplane

3.2.4. Flight software

It has been decided that the plane should be able to fly autonomously (requirement 2). Developing a software package capable of doing so is a daunting task and unfeasible in the given time-frame. Methods for creating control software for UAV systems are outlined in [2]. Luckily, a multitude of open-source software projects are available to facilitate that need. Below, a list of established projects are listed.

- Ardupilot
- PX4 (Dronecode)
- Paparazzi
- Librepilot
- MultiWii

The different options have been closely examined. From this analysis it followed that only a subset was suitable for our goals. MultiWii for example is an well settled project, but has an old code-base which is not very flexible. In addition, it has limited airplane support (as it was developed for quad copters).
…
Eventually PX4 was decided on as the best choice. Main selling points are a good software structure which is very modular and flexible with advanced structural features. Two of those are an internal messaging system between software components and hardware abstraction. This messaging system will serve as the basis for the software modifications explained in section 4.2. As explained in [2], the most common method for controlling the UAV vehicle is through PID. The PID control implementation of

[1] Real-Time Kinematic GPS

Board	Weight [g]	Size [mm]	Connections
Pixhawk 1	38	$50.0 \times 81.5 \times 15.5$	+
mRo-X2.1	10.9	36×50	~+
HKPilot32	33.1	$44.0 \times 81.0 \times 15.0$	+
Pixfalcon	unknown	unknown	~-
DroPix	15	$50 \times 67 \times 6$	++
Pixracer		36×36	~
Cube (Pixhawk 2)		90×40	+
Pixhawk 3 Pro	83	$50 \times 72 \times 23$	+++
Pixhack V3	63	$44 \times 68 \times 17$	+
Pixhawk 4	15.8	$44 \times 84 \times 12$	++
Pixhawk 4 Mini	unknown	$38 \times 55 \times 15.5$	+
Pixhack v5	90	$44 \times 84 \times 12$[3]	++
Omnibus F4 SD		36×36	unknown

Table 3.2: Comparison table of different flight controllers

PX4 can be found on [5]. It contains derivations and explanations based on the different forces acting on the UAV.

3.2.5. Flight controller

The choice of a flight controller is (thankfully) limited by the flight software choice. The officially supported list of flight controllers can be found on the PX4 documentation website[2]. It is advantageous to take an officially supported board because of the following reasons:

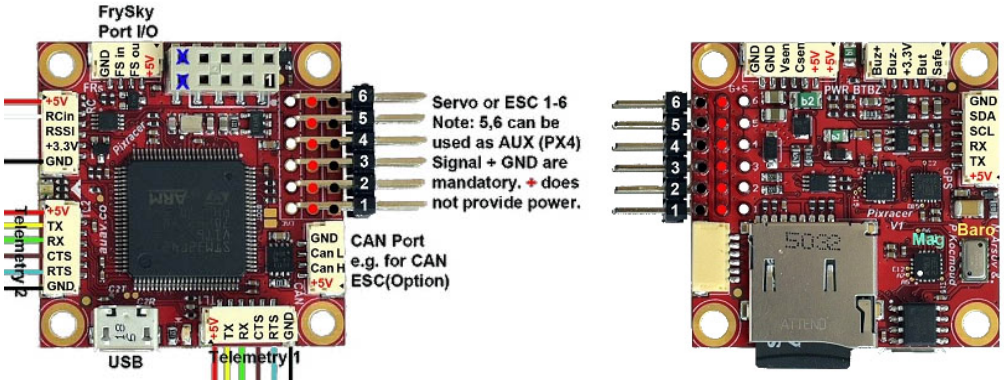
- Firmware should work out of the box
- More support tickets from other users are available
- The boards are high quality
- Extensive documentation

A comparison of the different options can be found in Table 3.2. In reality, more options are available with official support than shown in the table. It was decided, however, to only look at configurations that were not full computers running Linux. Using such configurations would add unnecessary complexity and power consumption. For this project, the following connections are needed: 6 PWM output ports, a GPS (UART) port, RC input (preferably PPM or S.BUS, see [8]) and telecommunication connection (UART). See Figure 3.1 for all connections to and from the flight controller. Almost all flight controllers in Table 3.2 have these abilities. Thus, the decision of the model is base on form-factor, weight, availability and cost. Availability of the boards has proved to be a problem. Many boards are deliverable in the United States only.
Seeking an option that was small in form factor, not too heavy or expensive, and available, the module that was eventually chosen is the flight controller board called the **Pixracer**[4]. It has fits into the fuselage easily, has a low weight and enough connection abilities to suit our needs. On top of that it is relatively cheap and deliverable to the Netherlands with a good build quality as icing on the cake. An image of it is displayed in Figure 3.7.

3.2.6. Hardware modifications

To allow communication between the flight controller and the air speed, I2C has to be used. As the Pixracer has no separate I2C pins exposed (see Figure 3.6), the connector for the GPS module has to be leveraged. The pin-out of the GPS module is given in Table 3.3. The connector on the Pixracer side can be seen in Figure 3.6a on the right. This connection can be modified to get access to the pins required for I2C (GND, SDA, SCL). This is done by taking the wires out of the JST-GH connector,

[2] `https://docs.px4.io/en/flight_controller/`
[3] Height is excluding cube.



(a) Pixracer top view (b) Pixracer bottom view

Figure 3.6: Pixracer pin descriptions
Courtesy of [4]

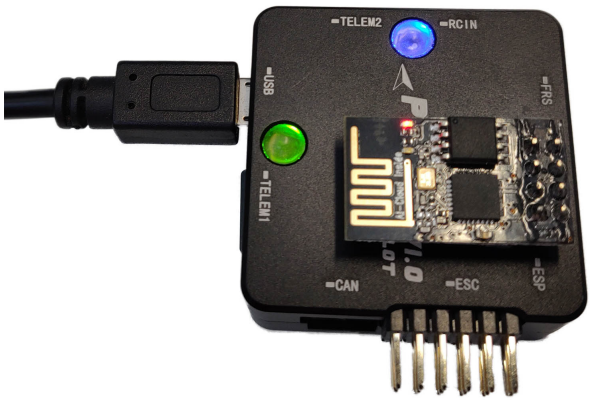
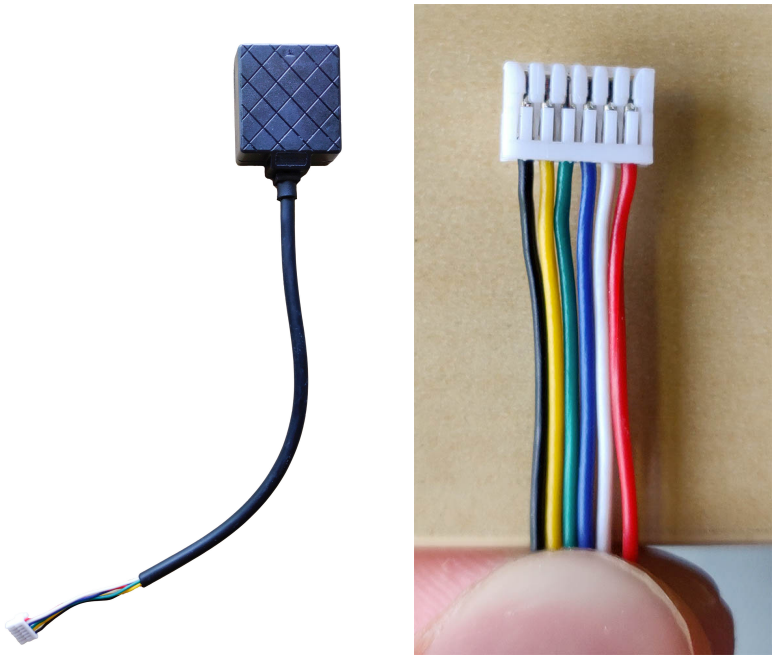


Figure 3.7: Pixracer connected to a PC with the ESP plugged in



(a) GPS module (b) GPS module wires

Figure 3.8: GPS module depictions

stripping the wire in the middle using a stripping tool and connecting another wire to it. This should create a Y-shape. The modification is finished by insulating the wires with heat-shrink.

Colour	Signal
Black	Ground
Yellow	SDA
Green	SCL
Blue	Tx
White	Rx
Red	V_{cc}

Table 3.3: Pin-out of the GPS module connected to the Pixracer

3.3. PV panel placement

An important design choice for the plane construction is choosing a method of mounting the solar cells onto the wing. While a different subgroup of the team is was responsible for the exact placement and grouping of the PV panels on the wing, the method of placement was to be determined by the group responsible for the construction of the plane.

Ideally, the solar panels should not increase the drag of the solar plane. For this reason, it was decided that the panels should be integrated within the wing. Because of the balsa wooden kit that was chosen, this approach would be relatively practical. The wings of the kit consist of a wooden frame that leaves a lot of empty space. To get the actual aerodynamic properties, the airframe is covered with a transparent foil after the frame has been constructed. The panels can then be mounted onto the airframe or within the empty space within the wing before the wings are covered with the foil. The foil then serves as protection. Two methods of placement have therefore been considered that fit the criterion of the panels not interfering with the airfoil of the wing.

The first method considered was to place the panels within the airframe, in the open spaces between the wings. The panels would then in no way interfere with the airfoil, leaving the aerodynamic properties of the wing unchanged. This method has the advantage that the panels can be rigid, as opposed to

flexible. However, the panels would be less exposed to the sun, given that if the plane is headed such that the sun is to the side of the plane, the panels would be partially overshadowed by the ribs. Ideally, the panels would lie along the surface of the wing. If that is the case, the largest area possible would be exposed to the sun regardless of the angle from which the sunlight comes.

In order to quantize the effect of the ribs overshadowing the panels on the power generated by the wings, a MATLAB script has been written that would calculate for a given rib shape, panel placement and direction of sunlight the effective area of the panel that is exposed to sunlight. The exposed area is calculated by projecting the points forming the top of the rib onto the plane in which the panel lies along the direction from which the sunlight comes. Two metrics of efficiency are used within these scripts. Firstly, the area efficiency is considered. This is the ratio between the surface of the panel that is exposed to direct sunlight and the total area of the panel. Secondarily, the power efficiency is considered. This is the area efficiency multiplied by an efficiency factor based on the angle between the direction from which the sunlight comes and the normal of the solar cell. The MATLAB code used to calculate these efficiency values for a single sun direction is found in Appendix A.2.5. The script used to calculate the value for different sun directions can be found in Appendix A.2.4. Using this data it would be possible to draw conclusions on the feasibility of placing the panels between the ribs.

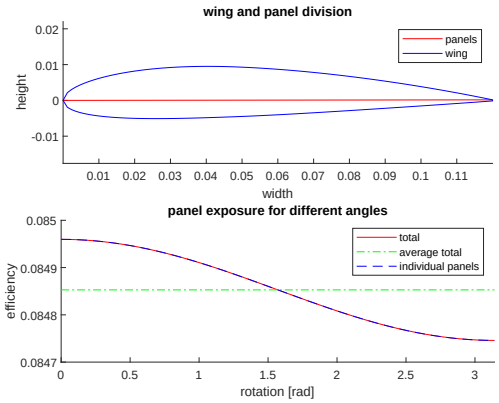
Since the plane will likely fly in any direction with respect to the sun, the angle from which the sun comes has to be taken into account. To this end, the efficiency for the panels is calculated for a range of angles from which the sunlight can come. this series of positions starts with the sun in front of the wing, and ends with the sun behind the wing, moving in a counter clockwise direction when viewed from the top. In order to determine the influence of panels being placed between the ribs rather than being placed along the surface of the wing a number of scenarios was compared. Firstly, the difference effect of the shadow of the ribs is investigated. In each scenario, a number of 5 cm wide panels are modelled along the surface of a 12 cm long rib corresponding to a NACA 2412 airfoil. As a reference, the shadow of the rib is not cast in the first test and the area efficiency of panels is then set to one.For comparison, the scenario is modelled with one, two, four and eight panels. With a larger number of panels, the panels lie closer to the surface of the wing. Therefore, the data obtained from these models can be used to determine the effect on the area of the panels if the panels are placed inside the rib rather than along its surface. The sun is rotated from the front to back of the wing, at an azimuth of $.5\pi$ radians from straight overhead. In order to compare the different efficiency values between different simulations, the efficiency is multiplied with the length of the cells to obtain the total efficiency. The results of this simulation can be seen in Figure 3.9.

It can be seen that the average efficiency increased when the number of panels increases. This is, of course, as expected, as the combined length of the panels is increased. More area is then exposed at the different angles. There is a slight variation in efficiency even without the shadow of the rib. This change in efficiency is caused by the angle at which the panels are positioned. This changes the power efficiency of the panels depending on the rotation of the sun. One may notice that there is also a slight variation in the scenario with one panel, even though this panel appears to be flat. The reason for this is that the first panel is also at a slight angle, because the generated airfoil does not end in a point at height 0, but slightly above it.

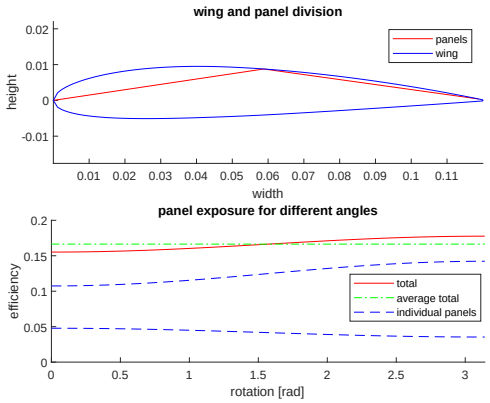
To determine the effect of the shadow cast by the ribs of the panel, the same scenario is simulated again. However, this time the shadow of the panels is cast onto the planes and the decrease in area efficiency is modelled. The resulting efficiencies are seen in Figure 3.10. It can be seen that the efficiency of the panels is decreased by the overshadowing ribs. As expected, the decrease of the sun is larger for angles where the sun is more to the side of the wing. It can also be seen that the decrease in efficiency is less pronounced for higher number of panels, as the shadow cast in these scenarios is smaller. A decrease in average efficiency is also present. However, this decrease is much less significant than the decrease in average efficiency caused by the decrease in total area of placing a smaller amount of panels in the wing.

To investigate the effect of the overshadowing ribs further, the scenario was modelled again. All parameters remained the same as they were for the generation of Figure 3.10, but the width of the panels has been decreased to 1 cm. This will make the effect of the shadow more pronounced, as the shadow will cover a larger fraction of the panel. The results of this simulation are seen in figure 3.11. It can indeed be seen that the efficiency is reduced more than for the case with 5 cm wide panels. however, the decrease in efficiency caused by the number of small panels is still significantly greater.

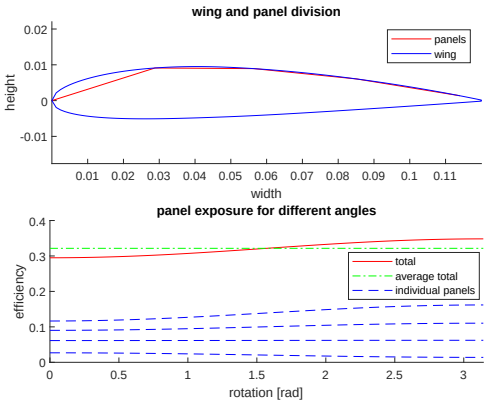
It is then determined that placing the panels closer to the wing surface is much preferred, mostly



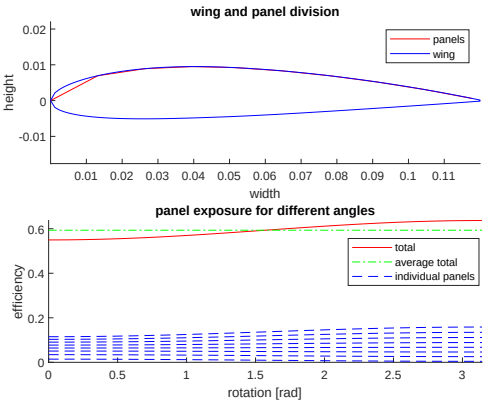
(a) 1 panel, mean efficiency 0.0849



(b) 2 panels, mean efficiency 0.1664



(c) 4 panels, mean efficiency 0.3216

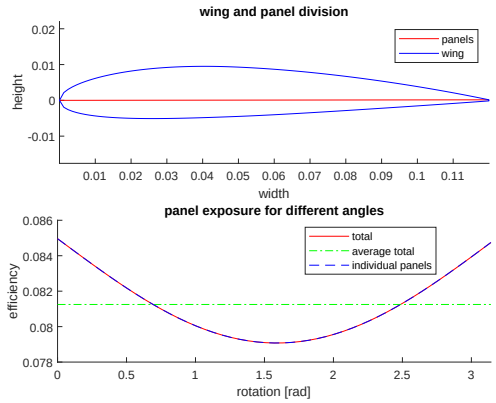


(d) 8 panels, mean efficiency mean 0.5933

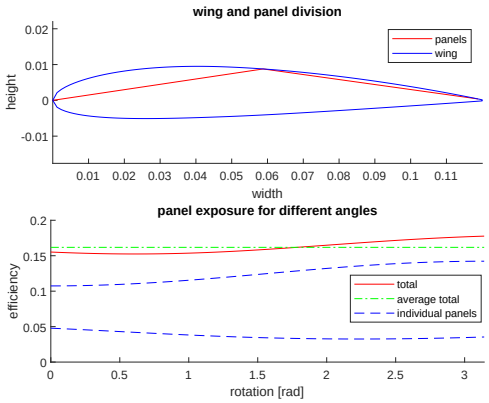
Figure 3.9: Efficiency of 5 cm wide cells next to 12 cm without being overshadowed

because of the increase in area. A second method where the panels are placed on the surface of the wing would be preferred. the first method considered to achieve this is to bend flexible panels across the surface of the wing. This does, however, require flexible panels to be used. These panels are usually less available and tend to have a lower efficiency. Because of this, it was proposed that normal PV panels would have to be used. usually, these panels are laminated with a relatively strong, transparent polymer to increase the strength of the panels. If the panels are not laminated, these panels can be bent slightly. However, of the unlaminated panels are bent too far, they will crack, causing the panels to be reduced in efficiency drastically. Because of this, extreme care has to be taken if this method is chosen. To attach the panels to the wing, they are glued in place carefully and held in place while asserting as little pressure as possible. After the panels have been placed, the wing can be covered with the transparent foil as usual.

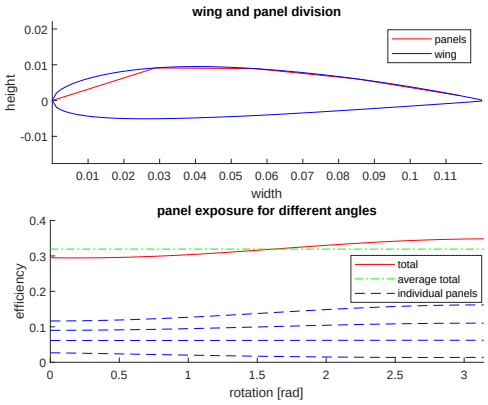
Although initially it was decided to use the first method, the second was chosen later on. This choice was made both because it would improve the efficiency of the panels and this method would require considerably less time.



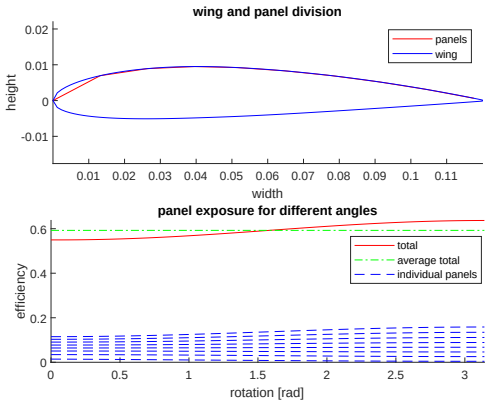
(a) 1 panel, mean efficiency 0.08125



(b) 2 panels, mean efficiency 0.1618

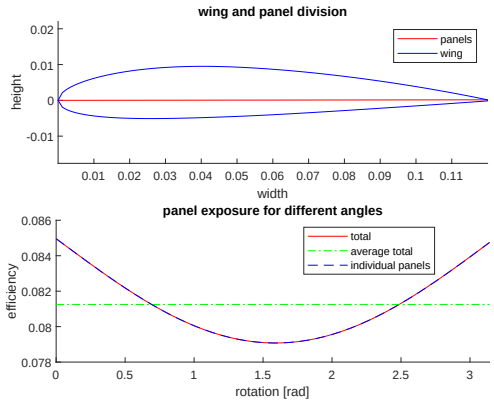


(c) 4 panels, mean efficiency 0.3193

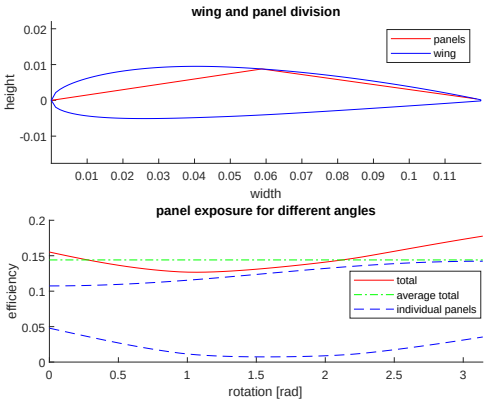


(d) 8 panels, mean efficiency mean 0.5925

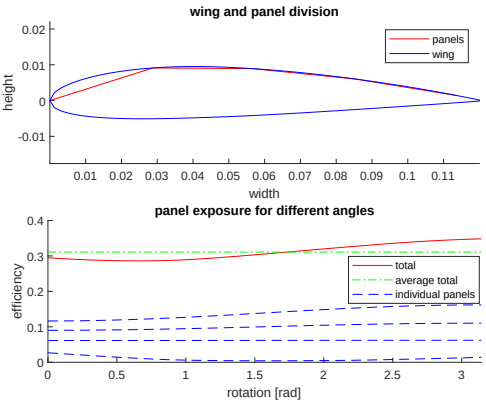
Figure 3.10: Efficiency of 5 cm wide cells next to 12 cm long ribs



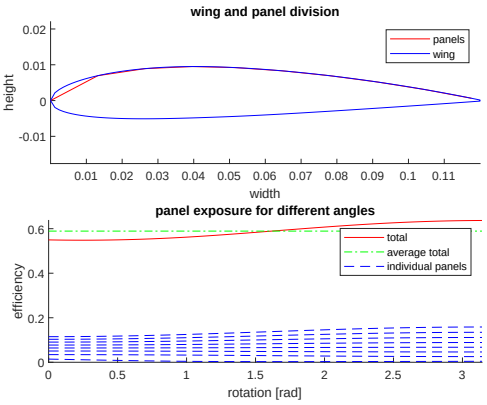
(a) 1 panel, mean efficiency 06684



(b) 2 panels, mean efficiency 0.1441



(c) 4 panels, mean efficiency 0.3108



(d) 8 panels, mean efficiency mean 0.5892

Figure 3.11: Efficiency of 1 cm wide cells next to 12 cm long ribs

4

Prototype implementation and validation

4.1. Component testing

The servos can be tested together with the Radio transmitter and receiver. First the servos to the RC receiver. The RC receiver is powered via a power supply. Once all systems have been powered, the receiver can be paired to the RC transmitter. If this process is succesful, the servos can be controlled via the RC transmitter.

To test the flight controller and its sensors, the sensors are connected to the flight controller and the sensors are connected to the flight controller. The flight controller is connected to a laptop for power. The working of the flight controller and its sensors can then be tested by calibrating via the flight control software on the laptop.

In order to test the motor and the ESC, the ESC is connected to the power supply. The propeller is connected to the motor and the motor is subsequently connected to the ESC. To allow the motor to be controlled easily, the ESC is connected to the RC receiver which is paired to the RC transmitter. The speed of the motor can then be controlled by use of the RC transmitter to test its working.

To test the working of the FPV transmitter, camera and receiver. To each component its required supply voltage was applied using a power supply. The FPV receiver was connected to a laptop via a capture card. The camera connected to the FPV transmitter. Once power was supplied to the components, the FPV receiver can be paired to the FPV transmitter. Once this has been completed video from the camera can be received on the computer.

4.2. Altering flight controller communication

Requirement 14 states that data from the airplane should be monitorable from a ground station. The software that has been chosen (section 3.2.4) has this built in for all but the hardware elements that were custom added. Those elements mainly include the PV system. This system lies at the heart of the power delivery to the rest of the plane, so it is important to be able to be able to observe the state. Thanks to the modularity of the PX4 software, the firmware running on the Pixracer can be modified to incorporate additional functionality such as this.
Figure 4.1, taken from the documentation website in the caption, shows the software architecture of the firmware. It is built up from modules (in blue), long-term data (gray) and a messaging system (green). To allow PV data to be displayed on a ground station, custom messages and modules are needed. Requirement 13 will also require incorporating changes to the logging system (*Database* in the figure). The flight controller communicates wirelessly with the ground station using a telemetry module, as stated in Figure 3.2. The communication protocol that is used for this is the Micro Air Vehicle link (MAVLink) protocol. It is explained and documented in [10]. To send the PV data over this connection, the flight controller should first acquire it from the MPPT controller (connection shown in Figure 3.1). There are two main ways to do this:

1. Use the I2C bus from section 3.2.6 with the Pixracer as master and the MPPT MCU as slave
2. Use the Telemetry 2 UART interface of the Pixracer (Figure 3.6b), and communicate with the MPPT MCU using the MAVLink protocol

Method 1 has the advantage of being natively spoken by almost all micro-controllers, including this one. It requires no third party libraries or headers. Cons are the fact that I2C is a shared bus, with a default low speed and using it can prove inflexible in development. The speed is no issue in this situation as the polling rate won't be that high (1-5 Hz)[1].
Method 2 has the advantage of having an individual connection to the MPPT MCU. In addition, using the MAVLink protocol instead of I2C offers a lot of flexibility in development and scalability. However, the initial investment is higher: the exact functioning of the MAVLink protocol, its usage in C++ in combination with the framework that is used on the MPPT controller and generation of code and headers has to be investigated.
It was therefore decided to go for I2C communication until the MAVLink system was better understood. The software being modular, switching communication method later is a possibility.

To be able to use the acquired data in the PX4 firmware, a custom module has to be created. How this can be done is documented in [3]. This module will acquire the data from the MPPT MCU using the aforementioned method, process it and publish it on the MAVLink to the ground control station. It is also used in development to print additional information. The code of the module as it stands is given in appendix A.1.1.
Additionally, uORB message types have to be generated to send data messages between different code modules. The message definitions are given in appendix A.1.3. These message definitions are used in the module code.

4.3. Plane construction

The most work in building the plane prototype involves assembling the air-frame. As mentioned in Section 3.2.1 on the component selection, a balsa wooden construction kit for a glider type plane was chosen. This kit was to be assembled and then customised to house the extra electronics as well as the PV cells. The steps to constructing this kit will be discussed briefly to give an overview of the work that has gone into it.

The construction of the plane can be divided into four steps. Firstly, the wooden wings are to be assembled. This is divided into two steps. The first step involves glueing the laser cut balsa wooden parts together to form the air-frame of the wing, as well as installing the servos to control the ailerons. The second step consists of mounting the PV cells onto the wings and finalizing the wings. This includes connecting the PV cells, integrating their wiring into the wings, and covering the wings along with the PV cells with transparent foil to obtain the desired aerodynamic properties of the wing. In the third step, the tail section is constructed and subsequently covered by foil as well. Connecting the tail section to the fuselage is also included in this step. The last step involves integrating all components. This involves preparing the fuselage to house the electrical components, and finally mounting all electronics for the flight control as well as the power electronics into the fuselage. Each of the steps will now be described in more detail, to document the complete process that was carried out.

4.3.1. assembling the wing air-frame

The chosen kit delivers the components of the wing as a large number of laser cut balsa wooden parts, as well as several beams of hardwood to give the construction a higher structural integrity. A manual is included, describing in detail the process required to assemble the wooden parts to create the wings. The wing was at first assembled according to the manual and any alterations were done afterwards, to make the process easier to replicate. The wing is divided into four sections, two per side. Each of these sections is to be assembled separately. The wing sections can be connected and separated after the plane is built, which allows for easier transport and maintenance of the plane.
The section that is to be connected to the fuselage is straight (that is, the airfoil is constant over its length) and houses the airbrakes. The outer section of the wing is longer and narrows towards the end to form the wing tips. This section includes the ailerons and houses the servos for the ailerons within it. Although the shape of these sections is different, their construction process is roughly the same. The construction process of the wing sections is documented in detail in the manual included in the kit, but a short overview is given here.

[1] The variations in light intensity are generally slow.

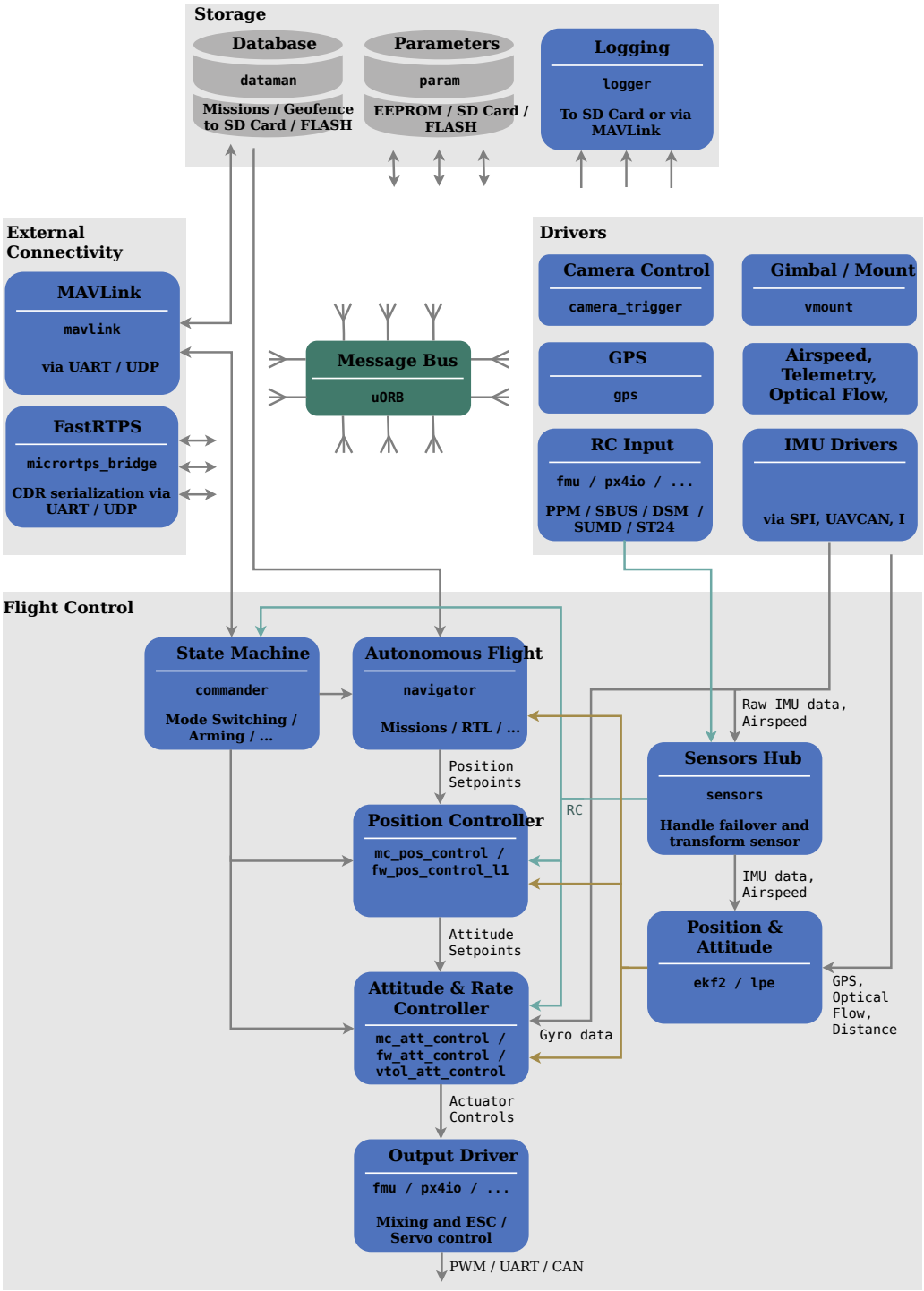


Figure 4.1: PX4 software architecture
source: `https://dev.px4.io/en/concept/architecture.html`

The first step of constructing the wing sections is to glue a balsa wooden 'comb' onto a hardwood beam, so that a spine is formed pointing upwards with slots for the ribs. Next, the ribs of the wings are glued into this comb to form the shape of the air-frame. Flat, balsa wooden panels are glued onto the bottom and the top of the front part of the ribs, to give the wing a better airfoil and to increase the rigidity of the wing. Next, any servo housings are constructed in the appropriate place. The servos are placed within them and connected to the aileron they control. Then, wooden beams are glued along the front of the wing section and sanded round to form the round profile at the front of plane. Lastly, two panels

are glued on the top and bottom of the ribs along the back of the wing section. These sections are sanded to create a sharp edge at the back of the wing, so that the airfoil ends in a point everywhere along the wing.

4.3.2. Installing the PV cells and finalising the wings

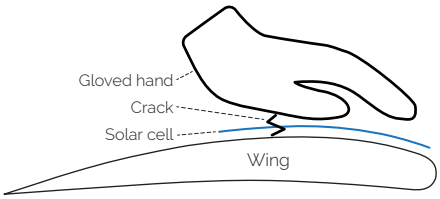
In order to maximize the amount of area used for PV cells, it was decided that PV cells would be cut into shapes to fit the wing. This was a task of a different subgroup of the team, and is described in detail their report. Once the panels have been cut, they can be integrated into the wings. This is done alongside finalising the wing sections. The wing sections have to be covered with a specific type of foil to get the aerodynamic properties. This foil sticks to the wood and shrinks under application of heat, which is supplied by the use of a small heated iron. The chosen panels are slightly flexible when not laminated. It was therefor chosen to mount the panels onto the wing unlaminated, covering the panels along with the wing in a transparent variant of the covering foil afterwards. Because the panels were then not laminated, their weight was less specific. The transmittance of the covering foil was tested to study the effects on the efficiency of using covering foil instead of laminating the panels. This study was carried out by the subgroup responsible for choices regarding the PV cells, and the results are stated in their report.

The plan The process of covering the wings with the foil starts by placing the PV cells onto the wing. The panels are held in place with a bit of epoxy type glue. After the panels have been placed, their connections are soldered. Because some connection points are on top of the wooden panels of the wing, slots have to be cut into the panels to allow the wires to be connected. The connection points of the panels are soldered together so that the panel groups are connected. Small connectors are soldered onto the panels at the ends of the wing sections, so that the wing sections can still be disconnected when needed.
After the panels have been placed and the connections were made, the wings can be covered by the transparent foil. This is done by cutting parts of the foil to shape, and then ironing them onto the wings. The wings are covered from bottom to top. At places with sharp corners, a small piece is first ironed into the corner, so that when the larger surfaces next to these parts are covered, a tight corner is left as a result.
When the wing covering process is done, the wings construction is completed, and construction on the next section can begin.

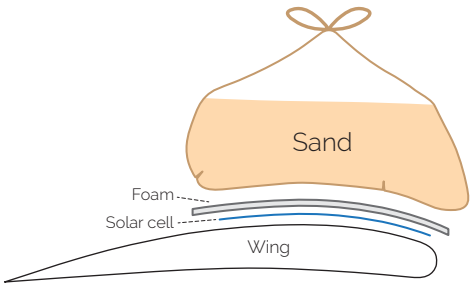
The execution Making a plan is one thing, but executing it correctly can prove to be a problem. The solar cells that were chosen, however bendable, are very brittle due to their thinness. This makes them very light which is advantageous, but makes the attachment to the wing difficult.
Initial efforts on putting the solar cells on the wing have failed. A diagram of the technique can be seen in Figure 4.2a. Epoxy glue was applied to two of the soldering strips at the bottom of the cell (see Figure 4.3 for a diagram of a cell). That side was then held against the wing and pressure applied to bend it. The pressure hotspots created by this technique proved too much for the cells, however, and they broke. Figure 4.5 shows the situation of the wing from the side. From it one can clearly see how far the panel should bend. Having all 3 panels break in the first experiment revealed that another take on the problem was required. The second plan that was formed incorporates using a foam layer and a bag of sand to distribute pressure evenly over the surface of the cell. A diagram demonstrating this idea is shown in Figure 4.2b. To test this theory, an a malfunctioning but physically intact cell was bent but not glued. This process caused the panel to break where a scratch was already present. To improve the method, the foam was indented on the location of the soldering strips. Another test was then successfully carried out. A picture of the resulting set-up and shape of the solar cell can be found in Figure 4.4.

4.4. Assembling the tail section

The tail assembly consists of four smaller sections. Two sections form the vertical stabilizer and the rudder, the other two form the horizontal stabilizer and the elevator. The sections are made by gluing two thin laser cut panels onto either side of a cutout of a frame for the wing. The two wings are glued together onto a connector, and that connector is then glued to the central shaft of the plane, a hollow



(a) Cell placement strategy using hand



(b) Cell placement strategy using sand for pressure distribution

Figure 4.2: Cell placement strategies that were tested

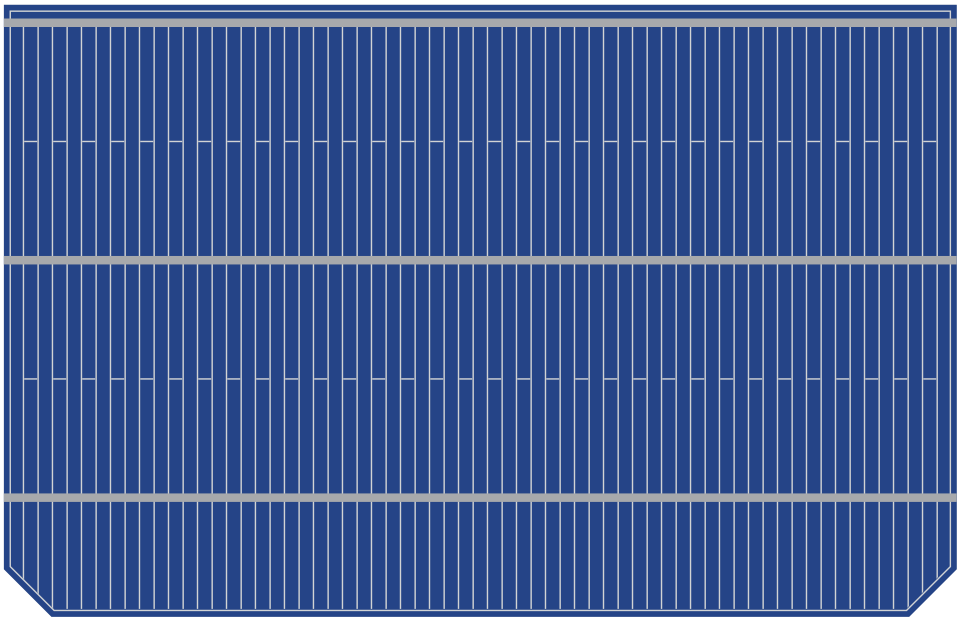


Figure 4.3: Schematic of the top of the solar cell

carbon fiber tube. The horizontal and vertical stabilizer can then also covered in foil to improve its aerodynamic properties.
To connect the central shaft to the fuselage, a small wooden panel with a hole is glued into place within the fuselage with two-component glue. See figure 4.6b. The same glue is applied to the shaft, after which the shaft is pushed into a hole in the back of the fuselage until it passes through the hole in the wooden panel. This ensures that the tail is straight behind the fuselage.
The servos that control the tail rudders are also located in the fuselage. They are placed into a wooden panel with cutouts that is glued into the fuselage. To control the rudders, two metal wires from the servos to the rudder pass through a tube within the central shaft.

4.4.1. integrating the electronics

In order to install the motor and the propeller, a part of the front of the fuselage is cut off. The motor is then attached to a part of wood that is glued into the front of the fuselage such that the axle of the motor exits the front of the fuselage. The propeller, along with the nose cone, can then be mounted onto the axle. All other electronics are attached to the inside of the fuselage, either by use of glue or with the aid of two sided tape and Velcro, to ensure that the components can still be removed for maintenance. The weight distribution of the heavier components such as the battery are also used to balance the plane, making sure that the center of gravity is at the desired position.

Figure 4.4: Application of the PV panel using the sand method

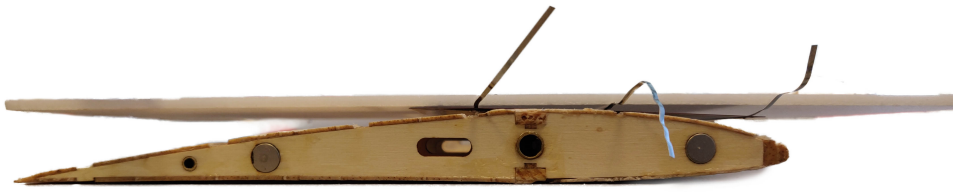


Figure 4.5: Side view of to-be-applied PV cell showing the bending angles required.

(a) A look at the wing mounting

(b) A look at the tail mounting

Figure 4.6: Mounting systems used in the air-frame

Once all components have been placed, the subsystems can be connected and prepared for flight.

4.5. Sensor integration

Since the airspeed sensor has to pass through the moving air to conduct measurements, it can not be mounted inside the fuselage. It is instead installed in the wing, some distance from the fuselage. This ensures that its measurements are not influenced by the airflow generated by the propeller. The sensor itself is mounted fully inside the wing, with the pitot tube exiting out of a hole in the front of the wing section. The wires from the wing are passed through the wing to the flight controller in the fuselage.
The light sensors also have to be mounted outside the fuselage, as they would otherwise not be exposed to direct sunlight. The sensors would be mounted within the wing with one sensor facing upwards and one sensor facing downwards. Mounting the sensors within the wing does mean that they will be covered by the transparent foil. This will attenuate the measured light levels slightly. However, the fraction between the direct light measured on the top side of the wing and the diffuse light measured on the bottom should not be altered by this change. The decrease in received light also allows the sensor to measure stronger light levels without the sensor reaching its maximum output, which might otherwise occur when measuring direct sunlight.

4.6. Flight testing procedures

After all subcomponents have been tested in a full systems test, and have subsequently been integrated into the plane, the plane is prepared for a test flight. Three test flights are used to verify the working of the system. The first flight is to verify the basic working of the system, such as how well the plane is controlled and the working of the manual radio control. The second test is to verify the working of the flight controller. The third flight will be an endurance test to determine the flight time of the plane. The procedures of the three test flights will be described in more detail in the following paragraphs.

During all flights, all sensors will be connected, and all data received from the sensors will be logged to allow the data to be processed afterwards. A stopwatch will be used to measure the times at which tasks are carried out. The data that is logged during each flight includes the location found by the GPS, the rotation measured by the accelerometer and magnetometer, the airspeed measured by the airspeed sensor,light received by the light sensors, and lastly the power consumed by the components of the plane and the power received from the different solar groups.

4.6.1. Maiden flight, radio control testing

The first flight of the plane is meant to verify the working of the different components, and therefor a number of different manoeuvres will be carried out, each to verify a different part of the system. These manoeuvres will be carried out only a small distance from the ground, with a margin to allow the plane to lose some altitude during its manoeuvres. This is to reduce damage to the plane in case of a failure. All tasks in this test flight will be carried out via manual radio control. The following list specifies these tasks, and what part of the system they test.

1. After liftoff, the plane will be instructed to fly in a large circle around the pilot to verify the planes stability.

2. The plane will be instructed to fly in a straight line, so that it can be trimmed to flight straight when no input is given.

3. The plane will be instructed to fly in a figure-eight in order to test the steering of the plane.

4. The plane will be instructed to fly in a circle, to measure the effect of different angles with respect to the sun on the obtained power.

5. The plane will be instructed to move up and down in a wave like pattern, to verify how well the plane can change its pitch.

6. The plane will be instructed to wave its wings by changing it's roll back and forth, in order to test the ability to control the roll of the plane.

7. The plane will be instructed to fly in a a straight line attempting to achieve top speed, to test the efficiency of the plane at different speeds and to verify that the plane construction can handle the forces at higher velocities.

8. The plane will be instructed to first fly over a surface known to have a high albedo, and then over a surface known to have a low albedo. The measurements of the light sensors during this time can be used to verify their working.

9. The plane will be flown to a higher altitude, and the pitch will be increased until the plane stalls. The throttle will then be lowered, and the plane is stabilized again while gliding down. This is to test at which pitch the plane stalls and how well it can be recovered in case of an emergency. Depending on observations in previous tests, it may be chosen to leave this test out to avoid the risk of damaging the plane prototype.

10. The plane will be decreased in altitude and eventually brought to the ground.

4.6.2. Test flight two, control testing

The second test is to verify the working of the flight controller. Commands will be issued to the plane by setting waypoints in the ground control software. The tasks will be carried out at a higher altitude, to allow more time to respond in case that the flight control fails. The following tasks will be carried out:

11. The plane will be instructed to fly in a large circle, to test how well the plane maintains altitude.

12. The plane will be instructed to fly through a series of waypoints spiraling inwards, to test how tight turns can be in order to be followed by the flight control.

13. The plane will be controlled to follow four waypoints placed in a square, to verify how the plane tracks sharp turns.

14. The plane will be instructed to follow a series of waypoints alternating between two different heights. The waypoints are placed closer and closer together. This is to test how high of a slope the flight control can track.

15. The plane will be instructed to follow a series of waypoints alternating between two different roll levels, the distance between the waypoints is incrementally smaller, to test how fast of a roll the flight controller can respond to.

16. The plane will be instructed to increase its pitch until it would stall, to test how well the flight controller recovers when stalling or whether the controller prevents stalling in the first place.

4.6.3. Test flight three, endurance test

In the third flight, the plane will be instructed via the flight controller to continuously fly in a large circle. The battery level will be closely monitored. If the battery level has decreased to nearly the desired DOD, the plane will be instructed to land again. From this test, we can derive the flight time of the plane under solar conditions. Furthermore, we can measure the received solar power at different rotations with respect to the sun, as well as the received light by the light sensor. We can also make an estimate of the power required for level flight.

5

Discussion of results

5.1. Flight rest results

although most work required to prepare for the flight test has been completed, it has not yet been possible to execute the tests. This is the result of a number of setbacks, some of which happened at later stages within the design process

The first setback is that the solar panel placement took longer than anticipated. It was decided to change the method of solar panel placement quite late into the design process. The new solar panel placement method proved difficult because of the fragility of the PV cells. Before a panel could be placed, the panels had to be laser-cut to shape and their contacts points had to be soldered on. Only after this was achieved could the panels be placed on the wing. It happened multiple times that the cells cracked during placement. Because this happened at the point where the cells were already attached to the wing with glue, removing the cracked panels without damaging the airframe required significant work. The unlaminated panels easily chipped, meaning that removing the cells required carefully scraping off the glued on fragments. After the broken panels have been removed, the section of the airframe also has to be thoroughly cleaned up again to allow for new panels to be placed. This entire process proved to be time consuming.

The second setback was in ordering the components. In order to ensure that the components were ordered in time, they were ordered well in advance of their scheduled testing and integration. On top of that, distributors of the components were chosen that promised to ship the components within short time frame. Distributors that were located close to the place of assembly were preferred, in the hopes that this would decrease the time of delivery. However, because of the limited availability of some of the parts, is was not always feasible to ensure timely delivery. In particular, the airspeed sensor has not yet been received, although it has been ordered well in advance from a distributor in the United Kingdom. This causes significant problems, as this sensor is required to arm the flight controller for flight testing.

5.2. Component test results

While it has not yet been possible to conduct the flight tests, all control systems that have arrived have been tested. The working of all received components has been verified by the procedures as described in Section on component testing. The flight controller and its sensors were calibrated and verified to work as expected. Using the FPV modules, it was possible to transmit video from the camera to the laptop without any problems. There was, however, a noticeable latency of at most two seconds. By connecting the servos to the RC modules, it was possible to move the servos by using the RC transmitter, which verifies the working of both the RC transmitter and receiver, as well as the servos themselves.

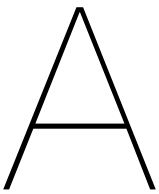
6

Conclusions, recommendation and future work

6.1. Conclusions

Because the flight testing has not yet been finished, no significant conclusions can be drawn on whether the plane meets the requirements. However, a number of recommendations can be made based on the construction process that has been completed so far.

Firstly, from the research on the solar panel placement in Section 3.3, it can be concluded that it is preferred to place the panels as close to the wing surface as possible. This, however, requires flexible solar panels to be applied. Although the availability of these panels is limited, their implementation is recommended over the application of unlaminated solar cells, because of the low structural integrity of the latter.



Appendices

A.1. Software code

A.1.1. MPPT communication module code

```
/****************************************************************************
 *
 *   Copyright (c) 2012-2016 PX4 Development Team. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 * 3. Neither the name PX4 nor the names of its contributors may be
 *    used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
 * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
 * COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
 * OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
 * AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
 * ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 *
 ****************************************************************************/

/**
 * @file mppt_communicator.c
 * airplane mppt subsystem communication module
 *
 * @author Example User <mail@example.com>
 */

#include <px4_config.h>
#include <drivers/device/i2c.h>
#include <px4_tasks.h>
#include <px4_posix.h>
#include <unistd.h>
#include <stdio.h>
```

```
#include <poll.h>
#include <string.h>
#include <math.h>

#include <uORB/uORB.h>
#include <uORB/topics/sensor_combined.h>
#include <uORB/topics/charge_controller.h>

__EXPORT int mppt_communicator_main(int argc, char *argv[]);

int mppt_communicator_main(int argc, char *argv[])
{
    PX4_INFO("Hello from the mppt_communicator");

    /* subscribe to sensor_combined topic */
    int sensor_sub_fd = orb_subscribe(ORB_ID(sensor_combined));
    /* limit the update rate to 5 Hz */
    orb_set_interval(sensor_sub_fd, 200);

    /* advertise charge controller topic */
    struct charge_controller_s at2;
    memset(&at2, 0, sizeof(at2));
    orb_advert_t at2_pub = orb_advertise(ORB_ID(charge_controller), &at2);

    /* one could wait for multiple topics with this technique, just using one here */
    px4_pollfd_struct_t fds[] = {
        { .fd = sensor_sub_fd,   .events = POLLIN },
        /* there could be more file descriptors here, in the form like:
         * { .fd = other_sub_fd,   .events = POLLIN },
         */
    };

    int error_counter = 0;

    for (int i = 0; i < 5; i++) {
        /* wait for sensor update of 1 file descriptor for 1000 ms (1 second) */
        int poll_ret = px4_poll(fds, 1, 1000);

        /* handle the poll result */
        if (poll_ret == 0) {
            /* this means none of our providers is giving us data */
            PX4_ERR("Got no data within a second");

        } else if (poll_ret < 0) {
            /* this is seriously bad - should be an emergency */
            if (error_counter < 10 || error_counter % 50 == 0) {
                /* use a counter to prevent flooding (and slowing us down) */
                PX4_ERR("ERROR return value from poll(): %d", poll_ret);
            }

            error_counter++;

        } else {

            if (fds[0].revents & POLLIN) {
                /* obtained data for the first file descriptor */
                struct sensor_combined_s raw;
                /* copy sensors raw data into local buffer */
                orb_copy(ORB_ID(sensor_combined), sensor_sub_fd, &raw);

                /* set at2 and publish this information for other apps */
                at2.battery_enabled = true;
                at2.current_a = (float)3.46 + raw.accelerometer_m_s2[0]*1;
                at2.voltage_v = (float)4.62 + raw.accelerometer_m_s2[1]*1;
                at2.power_w = at2.current_a * at2.voltage_v;
                at2.pv_enabled = true;


                PX4_INFO("Charge controller:\t%u\t%u\t%8.4f\t%8.4f\t%8.4f",
                            (unsigned int)at2.battery_enabled,
                            (unsigned int)at2.pv_enabled,
```

```
                                (double)at2.current_a,
                                (double)at2.voltage_v,
                                (double)at2.power_w);

                        orb_publish(ORB_ID(charge_controller), at2_pub, &at2);
                    }

                    /* there could be more file descriptors here, in the form like:
                     * if (fds[1..n].revents & POLLIN) {}
                     */
                }
            }

            PX4_INFO("exiting");

            return 0;
}
```

A.1.2. MPPT communication module CMakeList

```
############################################################################
#
#   Copyright (c) 2015 PX4 Development Team. All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
#
# 1. Redistributions of source code must retain the above copyright
#    notice, this list of conditions and the following disclaimer.
# 2. Redistributions in binary form must reproduce the above copyright
#    notice, this list of conditions and the following disclaimer in
#    the documentation and/or other materials provided with the
#    distribution.
# 3. Neither the name PX4 nor the names of its contributors may be
#    used to endorse or promote products derived from this software
#    without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
# BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
# OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
# AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
# LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
# ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
# POSSIBILITY OF SUCH DAMAGE.
#
############################################################################
px4_add_module(
        MODULE drivers__mppt_communicator
        MAIN mppt_communicator
        STACK_MAIN 2000
        SRCS
                mppt_communicator.c
        DEPENDS
        )
```

A.1.3. uORB message definitions

A.1.4. PV system status uORB message definition

```
float32 voltage_v               # PV voltage in volts
float32 current_a               # PV current in amperes
float32 power_w                 # PV generated power in watts
float32 duty_cycle              # DC/DC converter duty-cycle
```

A.1.5. PV charge controller status uORB message definition

```
float32 voltage_v               # Battery voltage in volts
float32 current_a               # Battery outflowing current in amperes
float32 power_w                 # Battery outflowing power in watts

bool pv_enabled                 # Whether or not the PV panels deliver power
bool battery_enabled    # Whether or not the battery delivers power
```

A.2. MATLAB simulation code

A.2.1. PV panel placement: wing evaluation demo

```
% Wing evaluation demo

% author          : Jasper Rietveld
% student number  : 4581881

% This program generates an airfoil with NACA number ”2412” with a chord
% length of 12 cm. The efficiency of 5 cm wide panels placed next to the
% rib is evaluated for different directions of sunlight.

% Generate the crosssection of the airfoil
wing = NACA_airfoil(”2412”, .12, 100);

% Evaluate different panel placements
analyze_airfoil(wing, 1, .05, .25*pi, 1, 0);
analyze_airfoil(wing, 2, .05, .25*pi, 1, 0);
analyze_airfoil(wing, 4, .05, .25*pi, 1, 0);
analyze_airfoil(wing, 8, .05, .25*pi, 1, 0);

% Evaluate the placements without overshadowing ribs
analyze_airfoil(wing, 1, .05, .25*pi, 1, 1);
analyze_airfoil(wing, 2, .05, .25*pi, 1, 1);
analyze_airfoil(wing, 4, .05, .25*pi, 1, 1);
analyze_airfoil(wing, 8, .05, .25*pi, 1, 1);

% Evaluate the placements with thin panels
analyze_airfoil(wing, 1, .01, .25*pi, 1, 0);
analyze_airfoil(wing, 2, .01, .25*pi, 1, 0);
analyze_airfoil(wing, 4, .01, .25*pi, 1, 0);
analyze_airfoil(wing, 8, .01, .25*pi, 1, 0);
```

A.2.2. PV panel placement: Airfoil evaluation

```
% Analyze airfoil

% author          : Jasper Rietveld
% student number  : 4581881

% calculates the efficiency of a number of cells placed within a
% transparent wing, overshadowed by a rib. The area efficiency is the
% fraction of the cell that is exposed to sunlight. The power efficiency
% is the area efficiency multiplied by the decrease in power based on the
% angle of the sun with respect to the normal of the cell. The efficiency
% is calculated for a number of angles as the sun moves from the front of
% the rib to the back of the rib, and the results are plotted.

% wing:             A row matrix containing the two points between which
%                   the cell is located, in coordinates within the wing
%                   cross-section.
% wing:             Matrix with the points that form the rib outline as
%                   its rows.
% rib_distance:     Distance between the ribs, or the cell depth.
% altitude:         Height of the sun in radians.
% rotate_quality:   The number of positions for which the efficiency
%                   should be calculated.
% plot_graph:       Logical set if the function should plot the efficiency
%                   of the panel for the different rotations,
% no rib:           Logical set if the function should ignore the shadow
%                   cast by the rib.

function analyze_airfoil(wing, panel_num, rib_distance, altitude,...
```

```
    plot_graph, no_rib)
    % Place panels at regular intervals to
    data_num = ceil(length(wing(:,1))/2);
    upper_wing = wing(1:ceil(data_num),:);
    panels = upper_wing(1:floor(data_num/panel_num)-1:end,:);
    if (isempty(panels))
        return
    end

    wing = flipud(wing);
    rotate_quality = 50;

    total_area_panels = 0;
    rotate_quality = 50;
    eff_power_list = zeros(panel_num, rotate_quality+1);
    eff_power_mean_list = zeros(panel_num, 1);
    eff_power_total = zeros(1, rotate_quality+1);
    eff_power_total_mean = 0;

    for n = 1:panel_num
        points = panels(n:n+1,:);
        panel_size = sum(sqrt(sum(points.*points,2)));
        [eff_power_mean, eff_power] = get_average_efficiency(points,...
            rib_distance, wing, altitude, rotate_quality, 0, 0, no_rib);
        eff_power_list(n, :) = eff_power*panel_size;
        eff_power_mean_list(n) = eff_power_mean*panel_size;
        total_area_panels = total_area_panels + panel_size;
    end

    % Calculate the total and average efficiency of the panels
    eff_power_total = sum(eff_power_list,1);
    eff_power_total_mean = mean(eff_power_total)

    if(plot_graph)
        % Plot the wing and the panels
        figure;
        subplot(211)
        hold on;
        plot(panels(:,1), panels(:,2), "r");
        plot(wing(:,1), wing(:,2), "b");
        title("wing and panel division")
        xlabel("width");
        ylabel("height");
        legend("panels", "wing")
        axis equal

        % Plot the different panels and their efficiencies
        subplot(212)
        hold on;
        plot([0:pi/rotate_quality:pi], [eff_power_total], "r")
        plot([0, pi], [eff_power_total_mean, eff_power_total_mean], "g-.")
        plot([0:pi/rotate_quality:pi], [eff_power_list], "b--")
        xlim([0, pi]);
        title("panel exposure for different angles")
        xlabel("rotation [rad]");
        ylabel("efficiency");
        legend("total", "average total", "individual panels")
    end

end
```

A.2.3. PV panel placement: NACA airfoil generation

```
% NACA airfoil generation

% author          : Jasper Rietveld
% student number  : 4581881

% This script generates a point list of a crosssection of a NACA type
% airfoil.
% For more detail on the formulas relating to NACA airfoil, one can
```

```
% visit for example https://en.wikipedia.org/wiki/NACA_airfoil

function [wing] = NACA_airfoil(NACA, scale, detail)
    NACA = num2str(NACA);
    c = scale;                          % chord length
    m = 0.01*str2double(NACA(1));       % digit 0
    p = 0.1*str2double(NACA(2));        % digit 1
    t = 0.01*str2double(NACA(3:4));     % thickness, digits 2, 3

    % Define the axis for the piecewise functions.
    dx = c/detail;
    x1 = 0:dx:p*c;
    x2 = p*c:dx:c;

    % Calculate the chamber line.
    y_c1 = m/p^2.*(2.*p.*(x1./c)-(x1./c).^2);
    y_c2 = m/(1-p)^2.*((1-2*p)+2.*p.*(x2./c)-(x2./c).^2);

    % Generate the airfoil.
    x = [x1, x2];
    y_c = [y_c1, y_c2];
    y = 5*t*(0.2969*sqrt(x/c)-0.1260*(x/c)-0.3516*(x/c).^2+0.2843*(x/c).^3-0.1015*(x/c).^4);
    y_1 = y_c + y;
    y_2 = y_c - y;

    % Scale the airfoil and add it to a point list.
    wing_x = [x, fliplr(x)];
    wing_y = [y_1, fliplr(y_2)].*scale;
    wing = [wing_x(1:end); wing_y(1:end)]';
end
```

A.2.4. PV panel placement: Cell efficiency with sun rotation

```
% get_average_efficiency

% author          : Jasper Rietveld
% student number  : 4581881

% calculates the efficiency of a cell within a transparent wing,
% overshadowed by a rib. The area efficiency is the fraction of the
% cell that is exposed to sunlight. The power efficiency is the area
% efficiency multiplied by the decrease in power based on the angle
% of the sun with respect to the normal of the cell. The efficiency is
% calculated for a number of angles as the sun moves from the front of the
% rib to the back of the rib, and the results are plotted.

% cell_points:      A row matrix containing the two points between which
%                   the cell is located, in coordinates within the wing
%                   cross-section.
% rib_distance:     Distance between the ribs, or the cell depth.
% rib:              Matrix with the points that form the rib outline as
%                   its rows.
% altitude:         Height of the sun in radians.
% rotate_quality:   The number of positions for which the efficiency
%                   should be calculated.
% plot_problem:     Logical set if the function should plot the scenario
%                   cross section and 3D scenario for one angle.
% plot_graph:       Logical set if the function should plot the efficiency
%                   of the panel for the different rotations,
% no rib:           Logical set if the function should ignore the shadow
%                   cast by the rib.


function [eff_power_avg, eff_power_rotate] = get_average_efficiency(...
    cell_points, rib_distance, rib, altitude, rotate_quality,...
    plot_problem, plot_graph, no_rib)

    % Calculate the efficiency for the panel for a number of directions.
    eff_area_rotate = zeros(rotate_quality+1,1);
    eff_power_rotate = zeros(rotate_quality+1,1);
    index = 1;
```

```
    for azimuth = -.5*pi:pi/rotate_quality:.5*pi
        [eff_area, eff_power] = get_efficiency(cell_points,...
            rib_distance, rib, altitude, azimuth, 0, 0, 0, no_rib);
        eff_area_rotate(index) = eff_area;
        eff_power_rotate(index) = eff_power;
        index = index + 1;
    end
    eff_area_avg = mean(eff_area_rotate);
    eff_power_avg = mean(eff_power_rotate);

    % Plot the scenario if required.
    if(plot_problem)
        get_efficiency(cell_points, rib_distance, rib, altitude, 0,...
            1, 1, 0, no_rib);
    end

    % Plot a graph of the final efficiencies for the different rotations
    % of the panel
    if(plot_graph)
        figure;
        hold on;
        plot(-.5*pi:pi/rotate_quality:.5*pi, eff_area_rotate, ”b”)
        plot([0, 2*pi], [eff_area_avg, eff_area_avg], ”r--”)

        plot(-.5*pi:pi/rotate_quality:.5*pi, eff_power_rotate, ”c”)
        plot([0, 2*pi], [eff_power_avg, eff_power_avg], ”m--”)
        axis([0, 2*pi, 0, 1.25]);
        title(”panel exposure for differnt angles”)
        xlabel(”rotation [rad]”);
        ylabel(”efficiency”);
        legend(”area efficiency”, ”mean area efficiency”, ”power efficiency”, ”mean power
            efficiency”)
    end
end
```

A.2.5. PV panel placement: Cell efficiency

```
% get_efficiency

% author          : Jasper Rietveld
% student number  : 4581881

% calculates the efficiency of a cell within a transparent wing,
% overshadowed by a rib. The area efficiency is the fraction of the
% cell that is exposed to sunlight. The power efficiency is the area
% efficiency multiplied by the decrease in power based on the angle
% of the sun with respect to the normal of the cell

% cell_points:  A row matrix containing the two points between which
%               the cell is located, in coordinates within the wing
%               cross-section.
% rib_distance: Distance between the ribs, or the cell depth.
% rib:          Matrix with the points that form the rib outline as
%               its rows.
% altitude:     Height of the sun in radians.
% azimuth:      Rotation of the sun in radians.
% plot_cross:   Logical set if the function should plot the
%               cross-section of the wing
% plot_3d:      Logical set if the function should plot the 3d result
% plot_2d:      Logical set if the function should plot the cell along
%               with the shadow of the rib that covers it.
% no rib:       Logical set if the function should ignore the shadow
%               cast by the rib.

function [area_efficiency, power_efficiency] = get_efficiency(...
    cell_points, rib_distance, rib, altitude, azimuth, plot_cross,...
    plot_3d, plot_2d, no_rib)

    if(no_rib == 0)
        % Determine the wing section above the cell
        cell_mid = cell_points(1,:)+diff(cell_points)./2;
```

```
    cell_plane = (cell_points-cell_mid)*5+cell_mid;
    [xi, yi, ii] = polyxpoly(rib(:,1), rib(:,2),...
        cell_plane(:,1), cell_plane(:,2));
    rib_upper = rib(ii(1)+1:ii(2),:);
    rib_upper = [xi(1), yi(1); rib_upper; xi(2), yi(2); xi(1), yi(1);];

    % Obtain the shadow of the wing section in the cell plane
    cell_normal = diff(cell_points)*[0, 1; -1, 0];
    cell_normal = cell_normal/norm(cell_normal);
    proj = shadow([rib_upper, zeros(size(rib_upper,1),1)], azimuth,...
        altitude, [cell_mid, 0], [cell_normal, 0]);

    % Find coordinates for the shadow within the plane of the cell
    base_1 = [cell_points(2,:) - cell_points(1,:), 0];
    base_1 = base_1.*1/norm(base_1);
    base_2 = [0, 0, 1];
    base_2 = base_2.*1/norm(base_2);

    proj_flattened = proj - [cell_mid'; .5*rib_distance];
    base_1_proj = repmat(base_1', 1, size(proj_flattened,2));
    proj_x = dot(base_1_proj, proj_flattened);
    base_2_proj = repmat(base_2', 1, size(proj_flattened,2));
    proj_y = dot(base_2_proj, proj_flattened);

    cell_3d = [cell_points(1,:), 0;
            cell_points(1,:), rib_distance;
            cell_points(2,:), rib_distance;
            cell_points(2,:), 0;
            cell_points(1,:), 0];
    cell_3d_flattened = cell_3d'- [cell_mid'; .5*rib_distance];
    base_1_panel = repmat(base_1', 1, size(cell_3d_flattened,2));
    panel_x = dot(base_1_panel, cell_3d_flattened);
    base_2_panel = repmat(base_2', 1, size(cell_3d_flattened,2));
    panel_y = dot(base_2_panel, cell_3d_flattened);

    % Determine the fraction of the cell that is not covered by the shadow.
    projection_shape = polyshape(proj_x, proj_y);
    panel_shape = polyshape(panel_x, panel_y);
    exposed = subtract(panel_shape, projection_shape);

    area_efficiency = area(exposed)/area(panel_shape);

    sun_vec = -[sin(azimuth)*sin(altitude);
            cos(altitude);
            cos(azimuth)*sin(altitude)];
    angle_decrease = dot([cell_normal, 0], -sun_vec);
    angle_decrease(angle_decrease<0) = 0;
    power_efficiency = area_efficiency*angle_decrease;
else
    cell_normal = diff(cell_points)*[0, 1; -1, 0];
    cell_normal = cell_normal/norm(cell_normal);

    area_efficiency = 1;

    sun_vec = -[sin(azimuth)*sin(altitude);
        cos(altitude);
        cos(azimuth)*sin(altitude)];

    angle_decrease = dot([cell_normal, 0], -sun_vec);
    angle_decrease(angle_decrease<0) = 0;
    power_efficiency = area_efficiency*angle_decrease;

    return
end

% plot cross section
if(plot_cross)
    figure;
    hold on;
    plot(rib(:,1), rib(:,2));
    plot(cell_plane(:,1), cell_plane(:,2));
```

```
        plot(rib_upper(:,1), rib_upper(:,2));
        plot(xi, yi, "*");
        axis([-2 2 -2 2]);
        pbaspect([1 1 1])

        xlabel("x");
        ylabel("y");
        title("wing cross-section")
        legend("airfoil", "cell plane",...
            "overshadowing airfoil", "intersection");
    end

    % Plot the cell, the direction along which the sun rays lie,
    % the points of the rib and its shadow,
    if(plot_3d)
        % Add a small offset to avoid Z-fighting.
        tolerance = 1e-3;
        cell_3d_plot = cell_3d - [0, tolerance, 0];

        % Plot the cell, rib and rib shadow.
        figure;
        hold on;
        grid minor;
        fill3(cell_3d_plot(:,1),cell_3d_plot(:,3),cell_3d_plot(:,2),"y");
        alpha(0.3);
        fill3(rib_upper(:,1),0.*rib_upper(:,2),rib_upper(:,2),"b");
        alpha(0.3);
        fill3(proj(1,:),proj(3,:),proj(2,:),'r');
        alpha(0.3);

        % Plot the sun direction
        sun_vec = -[sin(azimuth)*sin(altitude);
            cos(altitude);
            cos(azimuth)*sin(altitude)];
        sun_ind_scale = .25;
        sun_ind = [cell_mid'; .5*rib_distance];
        sun_ind = [sun_ind, sun_ind_scale*sun_ind+sun_vec];
        plot3(sun_ind(1,:),sun_ind(3,:),sun_ind(2,:),"g");

        xlabel("x");
        ylabel("y");
        title("3D plot of the cell and rib");
        legend("cell", "rib", "shadow", "sun direction");
    end

    % Plot the cell along with the shadow of the rib that covers it.
    if(plot_2d)
        window_shape = polyshape(1.5*panel_x(1:end), 1.5*panel_y(1:end));
        projection_window = intersect(projection_shape, window_shape);

        figure;

        hold on;
        plot(projection_window,'FaceColor','red','FaceAlpha',0.3)
        plot(panel_shape,'FaceColor','yellow','FaceAlpha',0.3)
        plot(exposed,'FaceColor','green','FaceAlpha',0.3)
        axis([1.5*panel_x(1), 1.5*panel_x(3),...
            1.5*panel_y(1), 1.5*panel_y(2)]);

        xlabel("x");
        ylabel("y");
        title("Exposure of cell to sun");
        legend("shadow", "covered cell", "exposed cell");
    end
end

% Function used to obtain the shadow of a series of 3d points onto a plane
% The plane is defined by a point and a normal. The points are projected
% onto this plane along the direction from where the sun is positioned.
% points:       Points to be projected onto the plane.
% altitude:     Height of the sun in radians.
```

```
% azimuth:      Rotation of the sun in radians.
% plane_point:  A point within the plane.
% plane_normal: The normal of the plane

function proj = shadow(points, azimuth, altitude,...
    plane_point, plane_normal)

    % Determine the vector along which the points are projected
    sun_vec = -[sin(azimuth)*sin(altitude);
            cos(altitude);
            cos(azimuth)*sin(altitude)];

    % Determine the location of the projected points
    lambda = (dot((points' - plane_point'), repmat(plane_normal',...
        1, size(points',2)))./dot(sun_vec, plane_normal'));
    points_projected = points' - lambda.*sun_vec;

    proj = points_projected;
end
```

A.3. Order list

Retailer	items	amount	total price	of which VAT	total shipping	used
hobbyking.com	Turnigy D2836/11 motor	1	€ 13,44	€ 2,82		+
	Turnigy Accucel-6 charger	1	€ 31,12	€ 6,54		+
	Turnigy 180A Watt Meter	1	€ 19,17	€ 4,03		+
	Turnigy TGY-i6 AFHDS Transmitter + Receiver	1	€ 47,61	€ 10,00		+
	Controller PPM/SBUS receiver	1	€ 12,31	€ 2,59		+
	FPV receiver	1	€ 34,14	€ 7,17		+
	FPV transmitter	1	€ 16,37	€ 3,44		+
	FPV camera	1	€ 25,00	€ 5,25		+
	Total		€ 216,90	€ 45,55	€ 0,00	
hoelleinshop.com	Aero-Naut Triple Neo Thermic	1	€ 189,00	€ 39,69		+
	CAMcarbon 12x7 klapluftschraube	1	€ 7,95	€ 1,67		+
	Turbo-Leichtspinnerkappe ø35mm	1	€ 5,00	€ 1,05		+
	Prop-Gummi	1	€ 2,00	€ 0,42		+
	RS-Mittelteil 35/4,0/8	1	€ 12,00	€ 2,52		+
	ORACOVER transparent	2	€ 10,80	€ 2,27		+
	ORACOVER Orange	1	€ 11,20	€ 2,35		+
	Total		€ 248,75	€ 47,26	€ 0,00	
conrad.com	Bluebird Micro-servo BMS-115HV	4	€ 14,87	€ 3,12		+
	Master mini-servo	2	€ 8,22	€ 1,73		
	Conrad ENergy LiPo acuu 2400 mAh	1	€ 16,11	€ 3,38		+
	Reely Sky-Series 20A	1	€ 16,99	€ 3,57		+
	TSL 2591	2	€ 11,56	€ 2,43		+
	Total		€ 132,14	€ 25,11	€ 0,00	
Farnell	Nucleo-32 microcontroller	2	€ 10,09	€ 2,12		+
	Total		€ 20,18	€ 4,24	€ 0,00	
www.flyingtech.co.uk	mRo PixRacer R15	1	€ 87,09	€ 20,33		+
	Total		€ 87,09		€ 0,00	
ftec-shop.nl	Radiolink TS100 Mini M8N	1	€ 24,00			+
	V2 Single 3DR telemetry kit	1	€ 28,50			+
	Total		€ 52,50		€ 0,00	
unmannedtechshop.co.uk	MPXV7002DP	1	€ 24,36			[1]
	Total		€ 24,36		€ 0,00	
	Grand total		€ 781,92			

Table A.1: Order list

[1] The sensor will be used but did not arrive yet.

Bibliography

[1] Klaus Betke. The nmea 0183 protocol, May 2000.

[2] HaiYang Chao, YongCan Cao, and YangQuan Chen. Autopilots for small unmanned aerial vehicles: A survey. *International Journal of Control, Automation, and Systems*, 8(1):36–44, 2010. doi: 10.1007/s12555-010-0105-z. URL http://www.springer.com/12555.

[3] PX 4 community. Px 4 module creation documentation, 2019. URL https://dev.px4.io/en/apps/hello_sky.html.

[4] PX4 community. Pixracer documentation, 2019. URL https://docs.px4.io/en/flight_controller/pixracer.html.

[5] PX4 community. Controller diagrams, 2019. URL https://dev.px4.io/en/flight_stack/controller_diagrams.html.

[6] OLANITZ-PRENA FOLIEN FACTORY GmbH. product information english, June 2010. URL https://www.oracover.de/downloads/viewcat/23/productinformation-all-products-english.

[7] OLANITZ-PRENA FOLIEN FACTORY GmbH. gewichtstabelle, August 2017. URL https://www.oracover.de/downloads/viewcat/30/weight-table.

[8] Oscar Liang. Rc tx rx protocols explained, February 2018. URL https://oscarliang.com/pwm-ppm-sbus-dsm2-dsmx-sumd-difference/.

[9] Jr. Loftin, LK. Quest for performance: The evolution of modern aircraft. nasa sp-468, 1985. URL https://www.hq.nasa.gov/pao/History/SP-468/app-c.htm.

[10] MAVLink. Mavlink website, 2019. URL https://mavlink.io/en/.

[11] Brian McManus. Are electric planes possible, 2019. URL https://www.youtube.com/watch?v=VNvzZfsC13o.

[12] Thomas E. Noll, John M. Brown, Marla E. Perez-Davis, Stephen D. Ishmael, Geary C. Tiffany, and Matthew Gaier. Helios prototype aircraft mishap. 1, Januari 2004.

[13] A. Noth. *Design of Solar Powered Airplanes for Continuous Flight*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, September 2008.