Escola Universitària Politècnica de Mataró

Centre adscrit a:

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Mechatronics Engineering

DESIGN AND BUILD OF A CABLECAM

Memory

Eloi Vilalta
Tutor: Jordi Ayza

Spring 2016

TecnoCampus Mataró-Maresme

Abstract

The idea of this project came out from the need of a media company looking to get a cablecam in order to take aerial-like images without worrying about drone legislations. The company used a gimbal to control the camera movements, but they could not take a picture or start/stop the video.

The student that realized this project researched about different options of components to carry the camera and build a prototype for the company. Also, designed a small printed circuit board and its code to control the camera actions remotely.

During the design task, the main problem was the friction of the rope with the motor and the pulley positions. That is why the first design of the frame has multiple holes at different heights to attach the pulleys. It can create different forces to the motor.

After many hours of testing, the device works perfectly with all the company requirements.

Resumen

La idea de este proyecto ha salido de la necesidad de una empresa audiovisual para tener un cablecam para poder hacer imágenes aéreas sin la necesidad de un dron. La empresa usa un gimbal para controlar el movimiento de la cámara, pero no podían controlar remotamente la acción de disparador de foto y vídeo.

El estudiante que ha realizado este proyecto ha buscado diferentes opciones para los componentes con el fin de poder llevar la cámara y construir un prototipo para la empresa. También, ha diseñado un pequeño circuito impreso y lo ha programado para poder controlar las acciones de la cámara para tomar una foto o vídeo.

Durante el diseño, el primer problema encontrado ha sido sobre la fricción de la cuerda con el motor y las poleas. Así que en el primer diseño ha tenido que hacer diferentes posiciones para las poleas a diferentes alturas. Así pueden crear diferentes fuerzas hacia el motor.

Después de muchas horas de testeo, el dispositivo funciona perfectamente con todos los requerimientos propuestos por la empresa.

Resum

La idea d'aquest projecte ha sorgit de la necessitat d'una empresa audiovisual per a tenir un cablecam per poder fer imatges aèries sense un dron. L'empresa usa un gimbal per controlar els moviments de la càmera, però no pot controlar remotament l'acció de disparar per foto o vídeo.

L'estudiant que ha realitzat aquest projecte ha buscat diferents opcions per als components amb el fí de poder portar la càmera i construir un prototip per a l'empresa. També ha dissenyat un petit circuit imprès i l'ha programat per a poder controlar les accions de la càmera per disparar una foto o vídeo.

Durant el disseny, el primer problema trobat ha estat sobre la fricció de la corda amb el motor i les politges. Així que el primer disseny que ha realitzat ha fet diferents posicions per a les politges a diferents alçades. Així es poden crear diferents forces cap al motor.

Després de moltes hores de test, el dispositiu funciona perfectament complint amb tots els requeriments proposats per l'empresa.

Dedicatory

To my parents

Index

Image Index

All the images used in this project are taken from Google images or by the author.

Table Index

Glossary

C	Coulombs
D	Distance
dB	Decibels
DIY	Do It Yourself
ESUP	Escola Superior Universitaria Politècnica
ESC	Electronic Speed Controller
FDP	Final Degree Project
FPV	First Person View
DSLR	Digital Single Lens Reflex
Ghz	Giga Hertz
GUI	Grafic User Interface
H	High
L	Lenght
m	Mass
mW	Mili Watt
PnP	Plug and Play
PWM	Pulse Width Modulation
Kv	Revolutions per Volt
RC	Radio Control

Rpm	Revolutions per Minute
Rx	Radio Receiver
T	Tension
TFC	Treball Final de Carrera
Tx	Transmitter
VRx	Video Receiver
VTx	Video Transmitter
ω	Angular speed
Θ	Theta, angle

1. Objectives

1.1. Purpose

The purpose of this project is to design, build, test and improve a Cablecam for a media company. Is a device that hangs on a rope and has a camera with underneath gyro stabilization that provides unique camera angles tailored to client needs (as for example weddings, sport events, musical shows) where an aerial image is needed.

1.2. Finality

To provide the company a working device that can take pictures or shoot movies from a higher perspective than regular ground camera systems and where other devices are not allowed (like drones) due to safety measures, prohibition or any other inconvenience.

1.3. Objective

There are different objectives in this project:

- Design and build a fully working cablecam that supports a DYS 3 axis gimbal underneath with a medium size DSLR camera (Sony NEX-5R). It is relevant to underline that both together, the camera and the gimbal, can weigh up to 4 kilograms [1].
    - Gather all the information needed from the electromechanical components such as the motor, the ESC, the gimbal, the FPV, batteries and the RC system.
    - Explain the steps made to make everything work. From the design, the transmitter, gimbal and ESC configuration to the electronic PCB.
    - Calculate the appropriate motor rpm and size to get the desired speed and reaction.
    - Calculate the tension for the rope in different scenarios.
    - Build the device, test it and redesign if needed to improve the functionality.
- Design and test a small electronic circuit to send by IR signals to the camera the operations to take a picture and start/stop video.
    - Select the appropriate chip and program.
    - Create the circuit diagram.

- Write the code and test it.
- Send the PCB to produce and test it.
- Design a 3D support for the PCB to send the data to the camera regardless the direction of it.

1.4. Scope

This project is intended to build a fully functional remotely controlled cablecam system for a company gathering the information needed to understand how it is realized and why the components are selected. There is no need to be the lightest or the best solution, as the company wants a working device.

The different tools and materials used to build the cablecam must be cheap, like out of the shelve and close proximity. For example, one material that can be used to create the frame is methacrylate, and thanks to the ESUP, the laser cutter to cut any drawing. Also 3D printing technology as there is an access to a 3D printer.

The frame and 3d parts will be designed using Solidworks program, as the author of this project have experience with it. For the PCB Fritzing will be used, as again, the author of this project have experience with it.

Gather all the formulas needed to calculate the tension of the rope and be sure it does not break and can support the cablecam with no problems.

The camera and the gimbal are components chosen by the company who wants to implement the cablecam device using its own stock gimbal and camera, so this project is not about to choose the best appropriated gimbal and camera.

2. Technical Requirements

Once defined what this project is about, it must accomplish different objectives with their technical specifications, detailed here:

- CableCam system
    - Able to carry 5Kg of payload
    - Motor fast enough to run at more than 50Km/h
    - Have enough battery to run 5Km in total
    - Control the camera shutter and play/stop the video from the transmitter

Some of the requirements are already calculated in the draft project.

3. CableCam design and build

In order to make this project possible there is a study to do about the different components of the system and how do they interact between them.

3.1. Previous information

As explained in the draft project, the different components of the cablecam are chosen between different options (battery, motor, transmitter, ESC, etc). Also, the rope tension calculations are exposed.

The tool used most of the days by the company is an excel chart provided by a cablecam services website [2] that uses the next equation to calculate the tension:

$$Tension\ (N) = ((Weight\ (kg) * 9{,}81) * RAIZ((Height\ (m)^\wedge 2) + (Distance\ (m)/2)^\wedge 2))/(2 * Height\ (m)) \quad (3.1.1)$$

The cablecam runs over a rope that is hanging from two positions, usually at a distance between 10 and 150 meters.

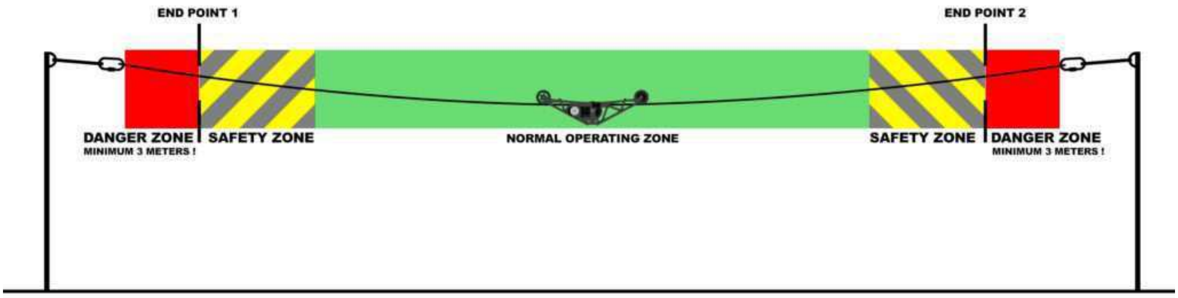


Image 1. Cable cam rope overview

The cable cam is composed of different parts as you can see below:

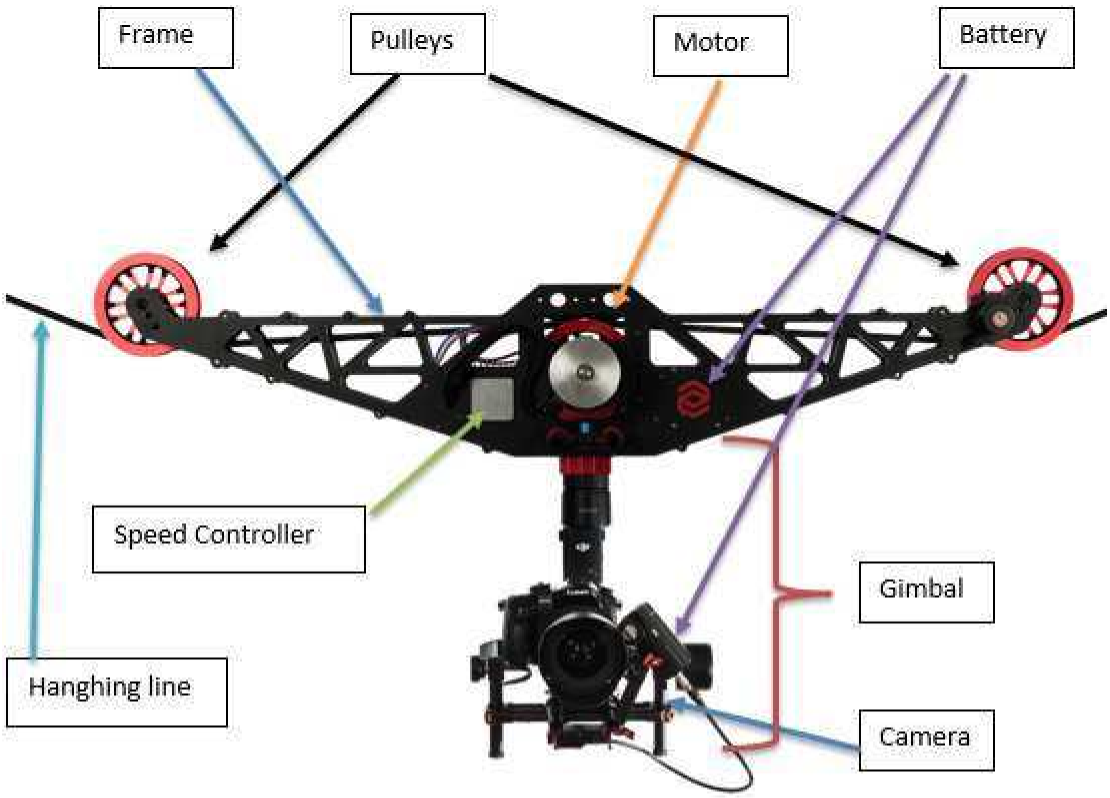


Image 2. Cablecam parts

Usually the image from the camera is transmitted via radio frequency to a receiver in the ground, where the person in charge of the movements is placed.

3.2. Motor speed and battery endurance

The motor chosen (in the draft project) for this project is the Multistart Elite 3510-350Kv with a 3D printed enclosure.

Running these motors at 22.2V gives a speed of:

$$\omega = Kv * NºCels * 3.7V = 350 * 6 * 3.7 = 7770rpm \quad (3.2.1)$$

The outside diameter where the center of the rope goes is 40 mm so:

$$linearvelocity = 2 * PI * 7770rpm * 26.24\, mmradius = 76.86\, Km/h \quad (3.2.2)$$

It can reach speeds of more than 75Km/h which is a 50% more than the 50Km/h needed.

Notice a fully charged 6S battery have a voltage of 25.2V, so these calculations are made with the nominal value.

Specs:
KV(RPM/V): **350KV**
Lipo cells: **4~6S**
Max current: **330W**
Max Amps: **15A**
No Load Current: **0.3A/10V**
Internal Resistance: **0.16ohm**
Number of Poles: **12N/14P**
Lamination thickness: **0.2mm**
Magnets: **N45SH**
Balancing spec: **0.005g or better**
Bolt hole spacing: **16 x 19mm/19 x 25mm**
Motor Shaft: **4mm**
Dimensions: **42 x 10mm**
Weight: **121g**

Image 3. Motor specifications

As the motor specification says, the maximum load is 15Ah.

The minimum battery duration is calculated as follows:

$$duration = \frac{capacity}{consumption} = \frac{1500mA}{15000mAh} = 0.1h = 6\ min \quad (3.2.3)$$

The travel it can make with one battery is calculated as follows:

$$travel = \frac{speed}{duration} = \frac{76.86Kmh}{0.1h} = 7.686Km \quad (3.2.4)$$

It means that with a battery of 1.5Ah it can travel at full speed 7.86Km, which is more than the 5Km needed for this project.

3.3. Transmitter and Gimbal configuration

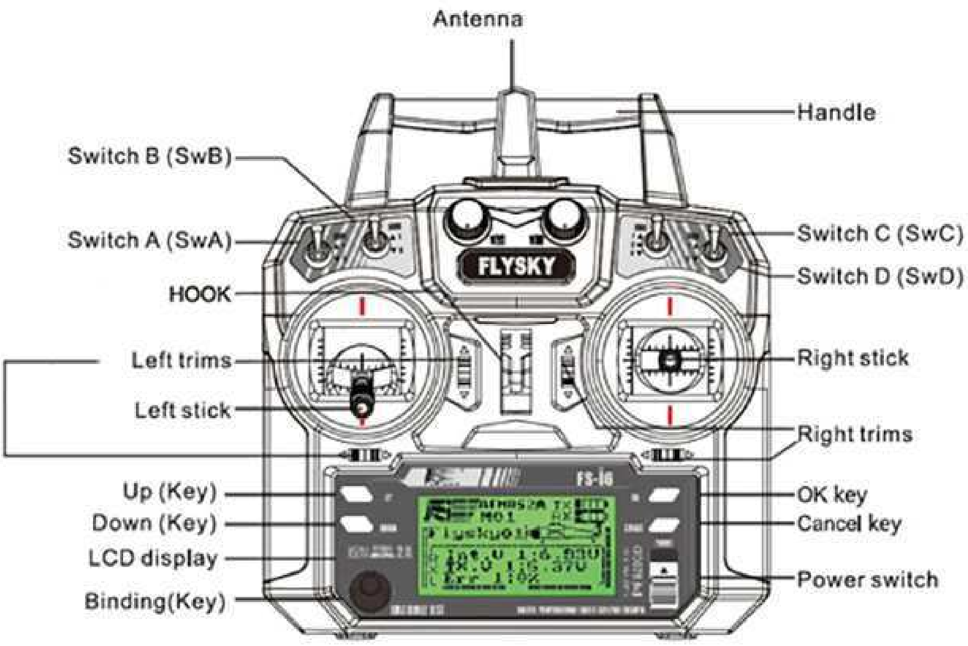


Image 4. Flysky Transmitter

In this project the FlySky i6 will be used, as offers the best configurability in order to achive the different movement.

3.3.1. Name configuration

This is the main screen where you can see whether or not there is any receiver working, the voltage it is getting and also other information like channel values:

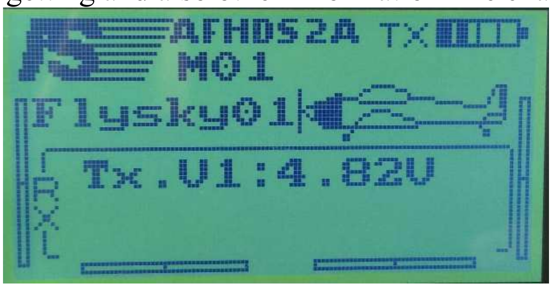


Image 5. Main screen of the transmitter

To get to the menu, the OK button must be pressed for a long time in order to get to it:

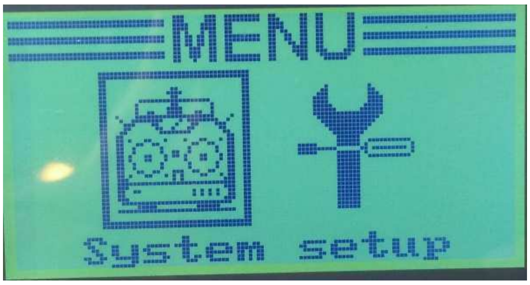


Image 6. Menu screen

Again, to enter the system setup and with the button "Down" till "Model name":

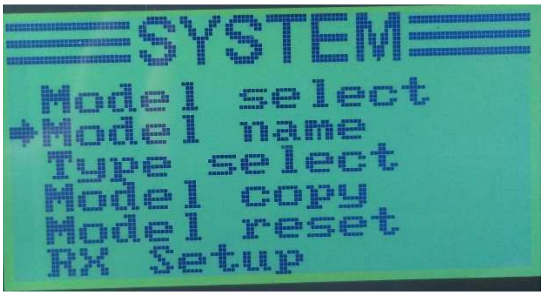


Image 7. System screen

Then, we type "Cablecam" by changing letter by letter using the "Up" and "Down" keys and pressing OK to change to the next character.

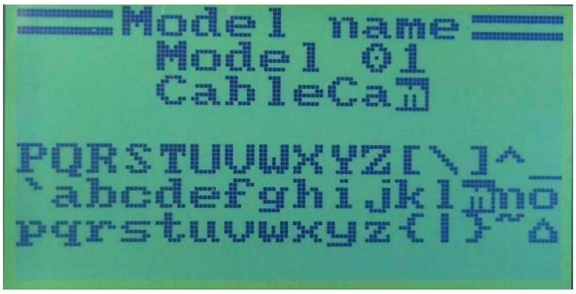


Image 8. Model name screen

Once we are done, you need to save it by holding "Cancel":

Image 9. New main screen

As you can see now the transmitter has a different name from the one it had on the beginning.

3.3.2. Dual Rate

The dual rate offers the possibility to reduce the sensibility of the sticks, by reducing the output depending on the stick position. To check which switch is assigned to the Dual Rate function the transmitter must enter now in the other setup:

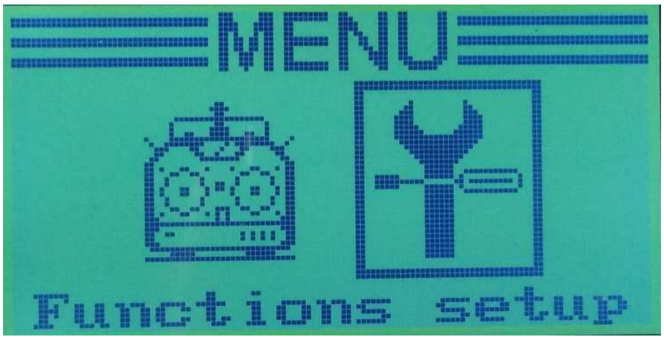


Image 10. Main screen functions option screen

Then “Down” till the “switches assign” option and check which switch is it related with the Fly mode:

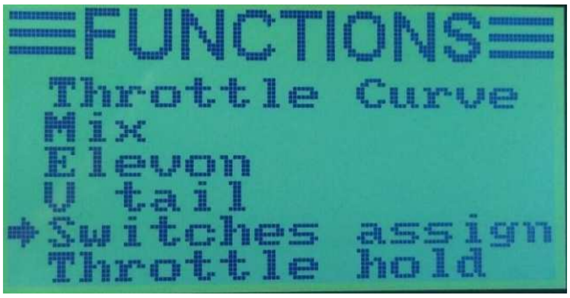


Image 11. Functions screen

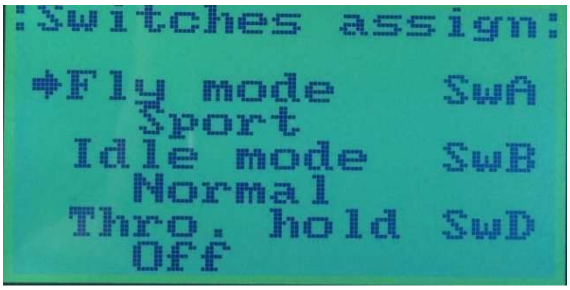


Image 12. Switches assign screen

After confirming it is the switch A, or if in any other switch, change it to switch A to make it more convenient.

The dual rate changes as said before the output signal. If we have a 100% curve in the mode sport and a 60% in the normal mod, when we use the sport mode if the stick is at 100% the output will be 100%, while in the normal mode, when the stick is at 100% the output will only be 60%, so there is more sensibility with the exponentials and less maximum and minimum endpoints.

Image 13. Dual Rate configuration

By changing the position of the switch “A” we can choose between two options.

If the switch A is on the top, the output of the receiver will be only a maximum of 60% of the signal, while if it is on the bottom, the output will reach the maximum value (100%).

3.3.3. Receiver to gimbal and ESC configuration

The transmitter has different switches that can be assigned as you want, but the sticks are configured as “Mode 2”, it means that the left vertical stick does not have any spring that centers the position of it. In “Mode 1” the non-spring stick is the vertical right one.

Image 14. Transmitter Stick Mode

The connection between the receiver, brushless gimbal controller, ESC and battery is as follows.

The receiver sends all the info via a PWM signal to the inputs of the different elements. The channel 2 (vertical right stick from the transmitter) connects with the ESC, if you increase the stick position the motor will run to one direction but if you lower the stick position the motor will run the opposite direction.

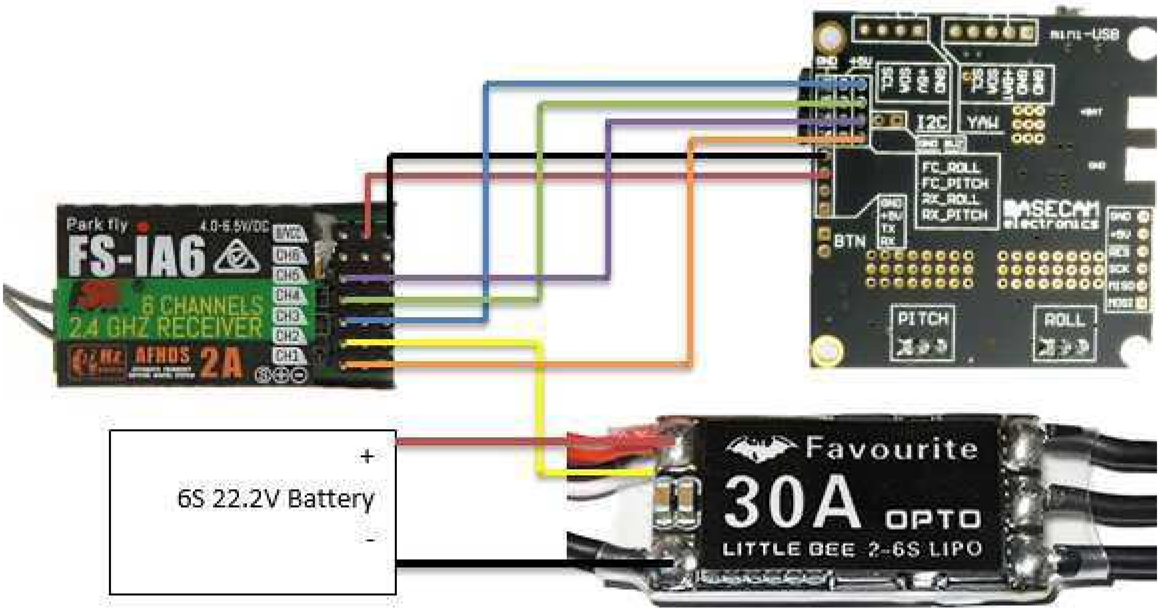


Image 15. Receiver wiring

As it can be seen in the image below, the Switch "C" is set as the auxiliary channel on the output 5, and the Switch "B" is on the channel 6. This Switch "C" has 3 different positions, so we can set different behaviors to the gimbal controller.

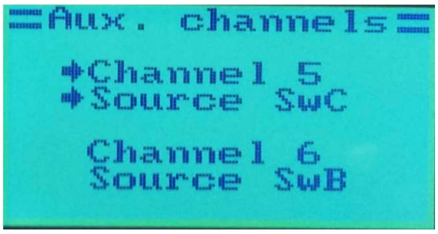


Image 16. Auxiliary channels

3.3.4. Brushless gimbal configuration

The 3rd axis and the sensor are connected as follows:

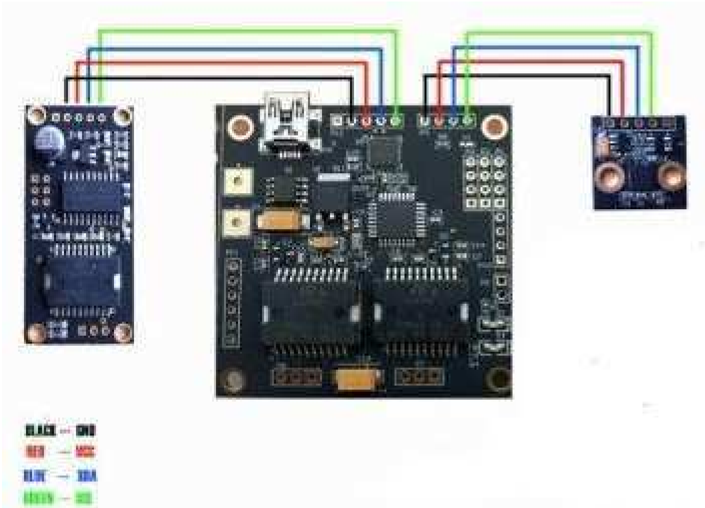


Image 17. BGC wiring

The sensor is placed just underneath the camera, pointing towards the front and upside down, as the IMU that is inside the board have the Z axis upside-down as you can see below.

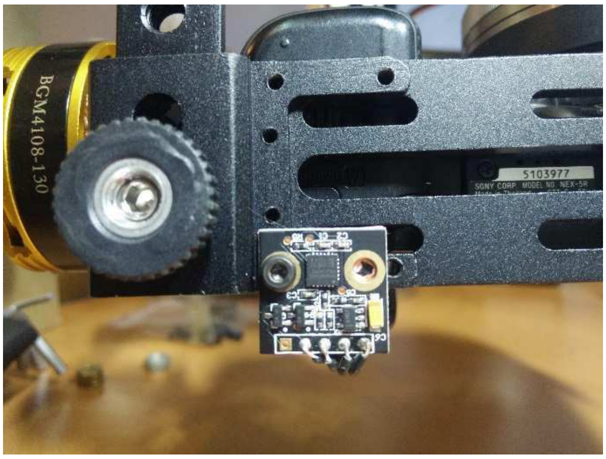


Image 18. Sensor placement

In the main program, there are different pages that are configurated as follows. The explanation of each option is not fully included in this project, as it depends on the owner of the gimbal. However, the most important values and the different options that have been changed or modified from the initial setup are shown below.

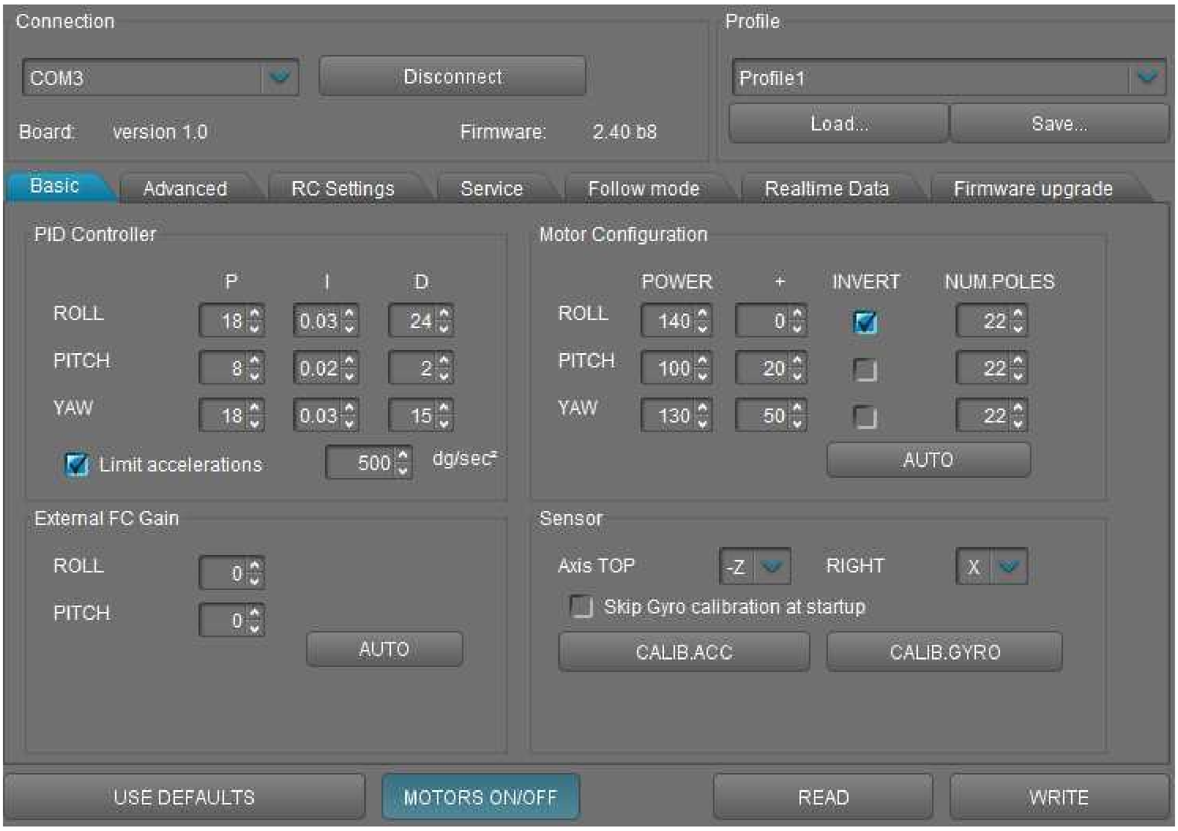
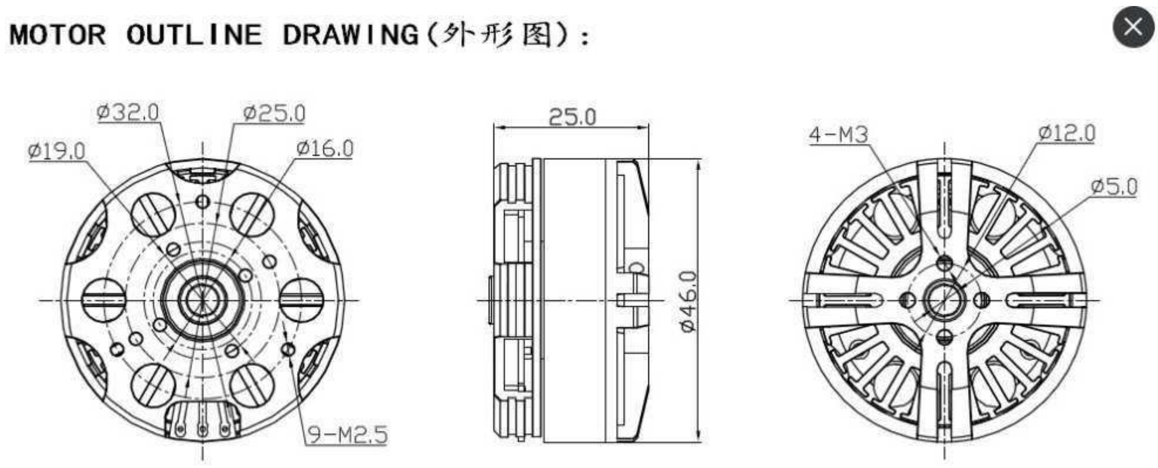


Image 19. BGC Basic

In the image on top it can be seen the different PID values for the Roll, Pitch and Yaw axis. The P is the proportional part of the signal to be implemented. The I value is the Integrative part, this is related to the acceleration of the movement and the D part, which is the Derivative part, is in charge of the break of the movement or slow down. Any of the above values set to a higher value will mean a higer response in that part.

On the other side, in the Motor configuration part, the power of each motor can be set, the minimum value is 0 (off) and the maximum is 255 (full power). The next column is the value that can be added or substracted to that power during the stabilization. As each motor is the same, the number of poles is also the same. This number is the same number of coils in the motor and can be seen also in the technical chart as below:



MOTOR PERFORMANCE DATA（性能参数）：

MODEL	Turns	Voltage (V)	Wire(mm)	Camera Range	Poles	Slots	Ri (Ω)	Lipo Cell	Weight (g)Approx
E4108	130	22.2	0.15	Nex6/Similar	22	24	17.0	3-6S	90

Image 20. Gimbal motor performance data

The next image shows the advanced tab:

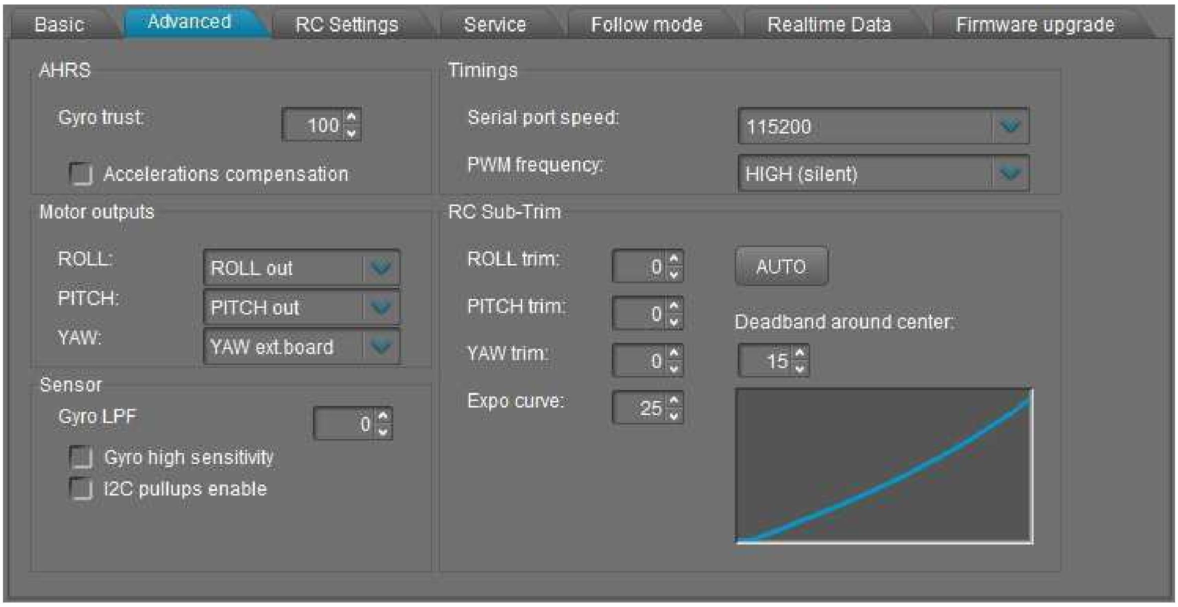


Image 21. BGC Advanced

In this tab the Motor output for YAW is setup for the external board, wired as before explained. The PWM frequency is set to HIGH, which makes the motor more silent but it needs more power and is less precise. If set to LOW the motor is more noisy, but drains less power and is more precise. In this case, the company chosed HIGH because they wanted to have the noise reduced.

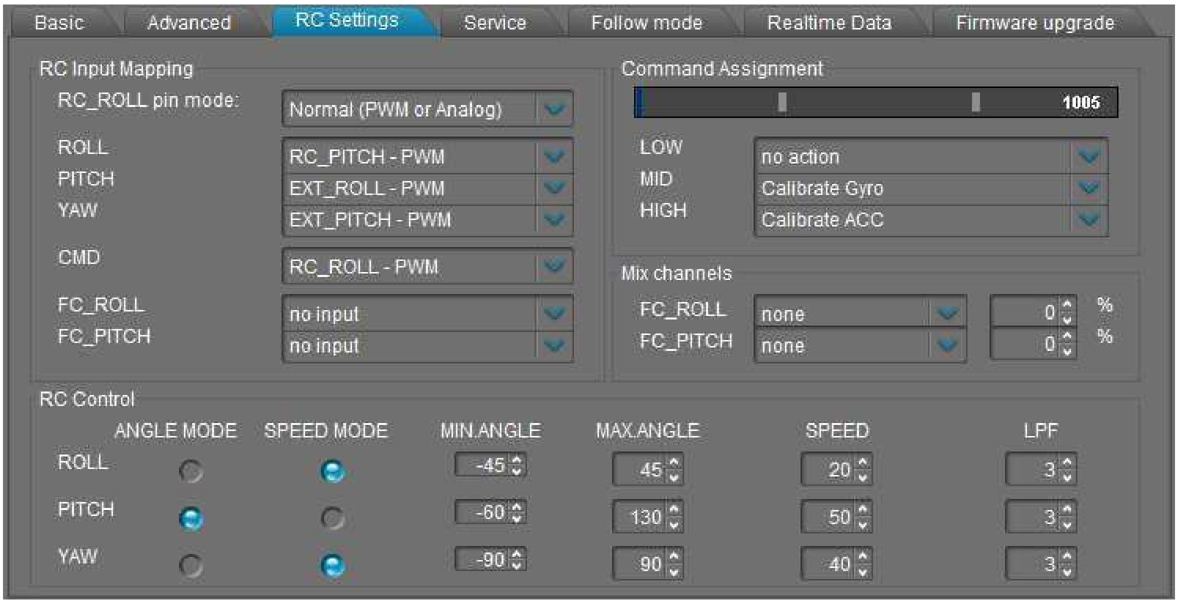


Image 22. BGC RC Settings

In the image above you can see how the different inputs are setup in order to read correctly all the PWM signals from the receiver. Roll is setup to the channel 1, this channel is from the horizontal movement from the right joystick of the transmitter, where the vertical one is setup as the ESC speed. It is usually not used, as the roll movement from the camera is just setup to correct the movement if the calibration is not done properly.

The pitch and yaw movement of the camera are setup for the left joystick of the transmitter, moving the stick forward it will make the camera look down, while backward it will make the camera look up. As it does not have the spring that centers the stick to the middle position, the RC Control mode is setup as angle mode. This mode is intended to just follow the stick position. If the stick is at the top, the camera will always look down.

However, if the yaw movement is setup as Speed Mode, which means that if the stick is in the center, the camera will not yaw to any direction, while if it is completely to the left, the camera will yaw left at maximum speed.

As you can see the min angle and max angle are also setup, so the camera can look more than vertically, but cannot look to the sky, as there is the cablecam in the view. The speed of the different movements is also setup in this tab. These values are in degrees per second.

The Command Assignment part is setup for the auxiliary “Switch C”, on the top position the gimbal will do nothing, on the middle position the gimbal will calibrate the gyro and on the bottom position the gimbal will calibrate the accelerometer.

3.3.5. ESC configuration

The Littlebee Pro 30A Electronic Speed Controller is setup with a program called BlHeliSuite.

Inside the program there are different options to change in order to make the motor run both ways, as it comes by default only in one direction:

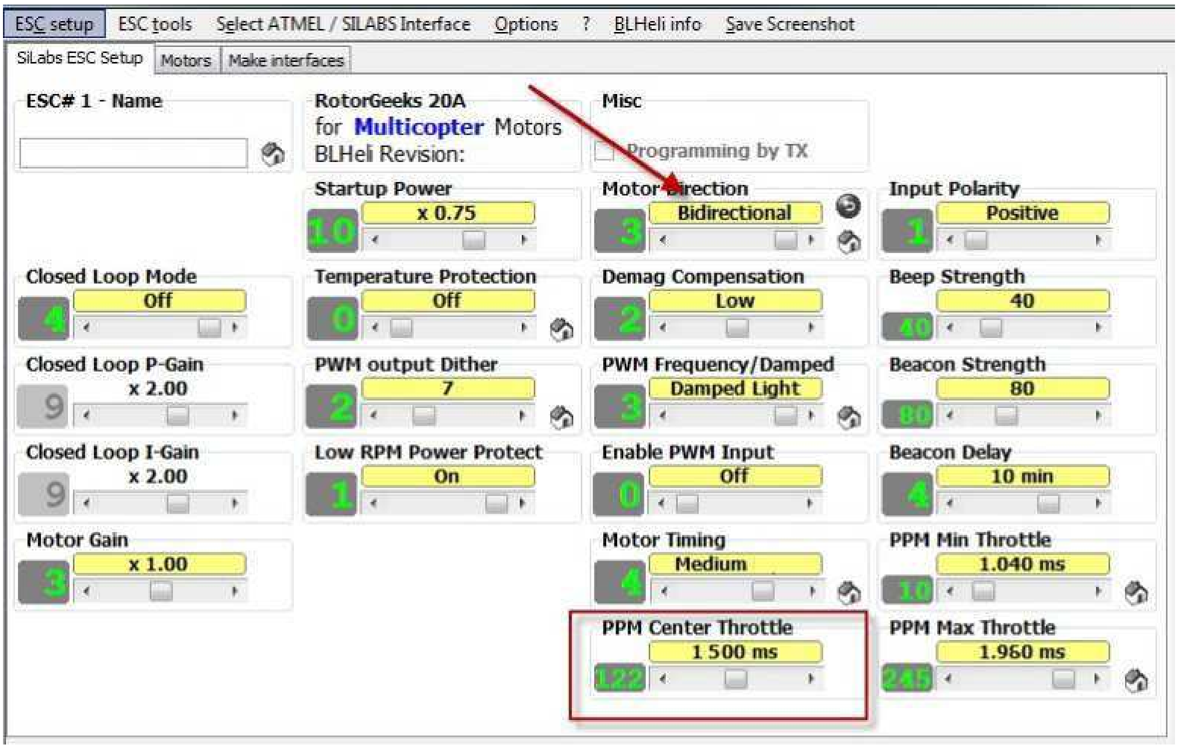


Image 23. BlHeliSuite program interface

In order to connect the ESC to the computer the Arduino board is used:

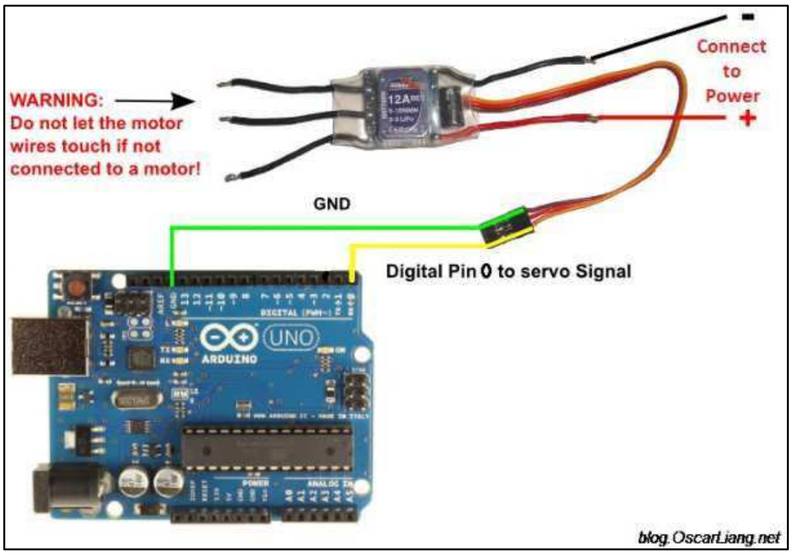


Image 24. Arduino to ESC connections

The receiver pin from the ESC connects to the digital pin 0 and the ESC is powered from the battery.

When choosing the “1-wire” option in the BlHeliSuite program, it also programs the Arduino board to do so. So any program is written in this project to program the ESC.

3.4. Rope

As explained before, the tension is calculated using the equation 3.1.1. But in the field, when it comes to the real tension of the rope, the company uses a device that actually shows the tension of it, the DYNAFOR Load Indicator:

Image 25. Dynafor Load Indicator

The load indicator is placed on any of the ends of the rope, just before the attachment with the ratchet.

As explained before in the draft project, the rope used is made from Dynema composites, which holds a very high tension force and for this project is more than enough to hold this cablecam weight without break.

3.5. Structural support

After the research is done, it is time to start the design of the cablecam. This cablecam is inspired by the commercial solutions that have already been made: two plates put together with spacers; two pulleys on the side and the motor in the middle with a direct drive solution, as there is no need to reduce the angular speed.

Down below you can see different commercial solutions gathered together:

Image 26. Commercial cablecam systems

3.5.1. Frame design

It consists of multiple 4mm clear methacrylate plates attached by screws and zip ties (for added safety in some holes). There are plates on top to hold the VTx and battery.

As the friction of the rope has to be tested, there are 4 holes on the sides at different heigths in order to change the friction of the motor to the rope. By this try and error method it can be clarified which high is best for the kind of pulleys and motor cover. As there are no experiments with PLA pulleys, flexible PLA motor cover and the Dynema rope together.

The distance from vertical lateral plates are 35mm because the motor fits in perfectly and can rest on the other side with a bearing, so 35mm spacers are placed in between to hold the structure in place.

The screws that hold the pulleys are M8 as calculated in the raft project, and the motor axis rest over ball bearings as shown below:

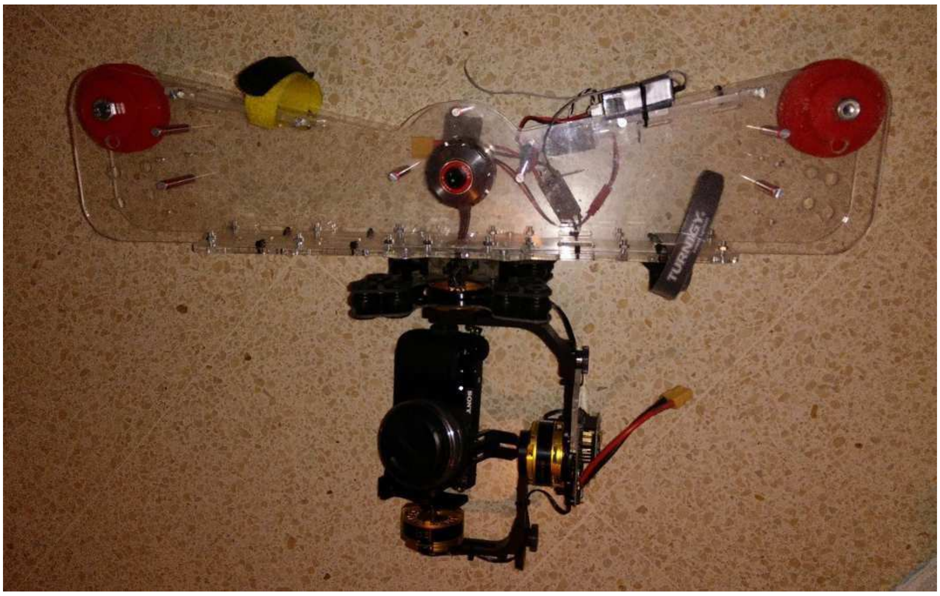


Image 27. Cablecam prototype build

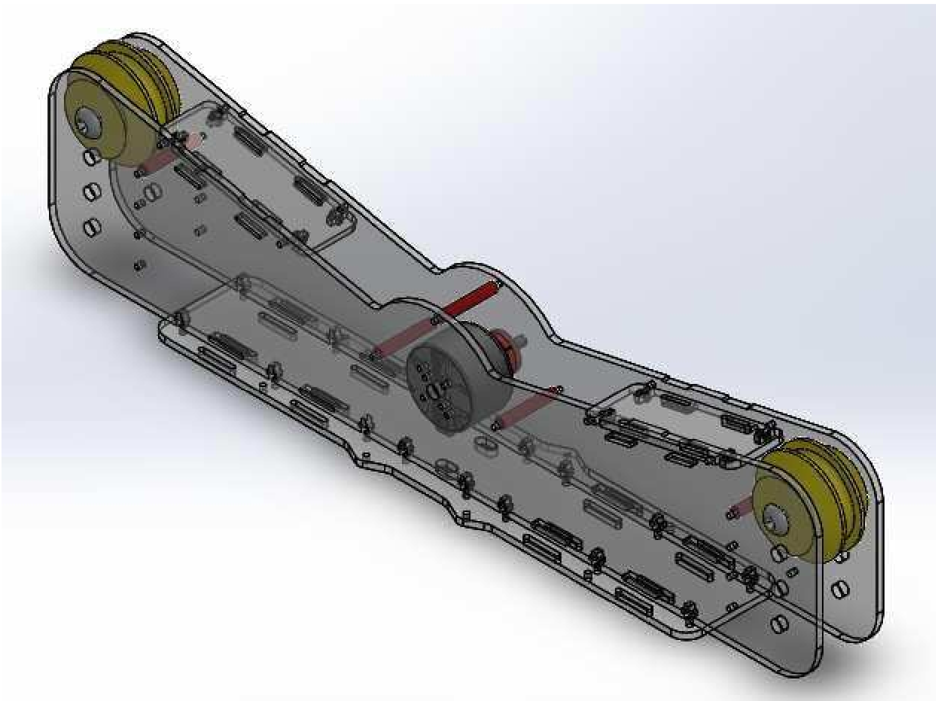


Image 28. Final SolidWorks assembly

The methacrylate plates (PMMA) have a traction modulus between 2400 and 3300 N/mm^2 (2K3 MN/m^2) [3]. In the first simulations the value does not get even to 1500N/mm^2:

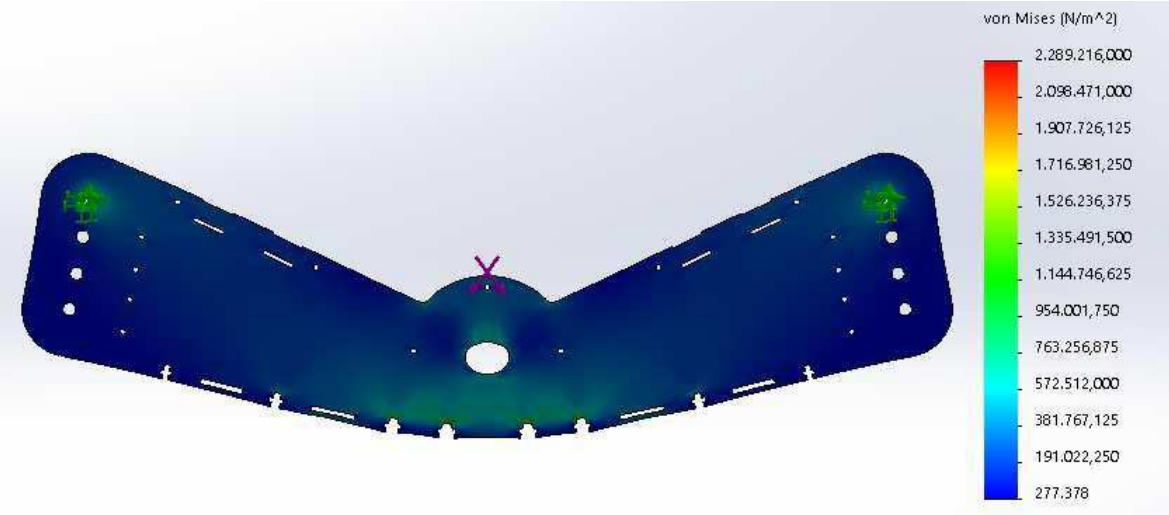


Image 29. Tension simulation

Only really close to the spots used to calculate the tensions there are some major tensions, but not close to the reality.

The force applied in the center hole is 50 Newtons, which approximately is 5Kg of force.

This first design will help to clarify which are the different failure points.

3.5.2. Side pulleys

As calculated in the draft, the screws are M8, which holds the cablecam over the rope.

In order to move freely over the rope with low friction, there are two ABEC7 deep groove ball bearings between the screw and the rope. ABEC11 ball bearings are much more expensive and not necessary for this project, as they are intended to run at a very high rpm. Normally the ball bearings are made out of ceramic materials.

The ABEC scale (Annular Bearing Engineering Committee) is an industry accepted standard for the tolerances of a ball beraing. This scale is designed to provide bearing dimensional specifications that meets the standards of precision bearing in a specified class.

The different tolerances are illustrated in the next image:

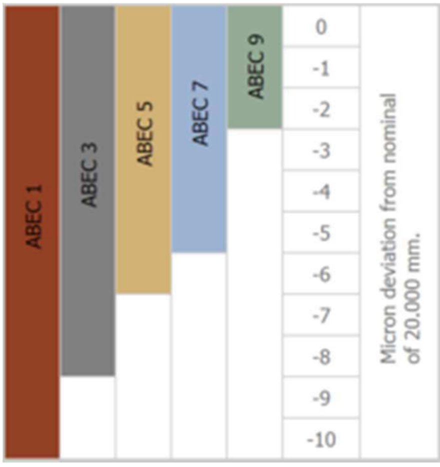


Image 30. ABEC tolerances

And the relationship with the ISO and DIN standards are shown below:

ABMA/ANSI Estd. 20	**ISO 492**	**DIN 620**	**JIS B 1514**
ABEC 1	Normal	P0	0 class 6Xth class
ABEC 3	6th class	P6	6th class
ABEC 5	5th class	P5	5th class
ABEC 7	4th class	P4	4th class
ABEC 9	2nd class	P2	2nd class
ABEC 11	1st class	P1	1st class

Table 1. ABEC standards

There are two different types of option to hold the cablecam over the rope, the first one, which is the one used in this project is by passing the screw through both plates, so any axial and bending forces are applied, only cutting force. The disadvantage is that the rope must be placed inside the cablecam prior to the tension of it, because it cannot be removed easily. The only possibility to place the cablecam over the rope when its tight is to disassemble the screw and assemble it again. Here is an example of this system:

Image 31. Both side pulley

The other option is to hold the screw just on one side of the frame, but causes bending forces and are harder to design due to the external forces, the frame must be harder out of aluminum or any other metal, this design is really expensive and the company does not like this extra cost. Down below you can see this kind of structure:

Image 32. One side pulley

The solution for this part is to 3d print a couple of spacer on both sides of the bearings, holding them in place. The bearings (8*22*7mm) have a spacer in between used in most skates, as the axis is also 8mm. This way we can have two bearing to increase the stability of the device. After that, another 3D print is designed to run the rope underneath:

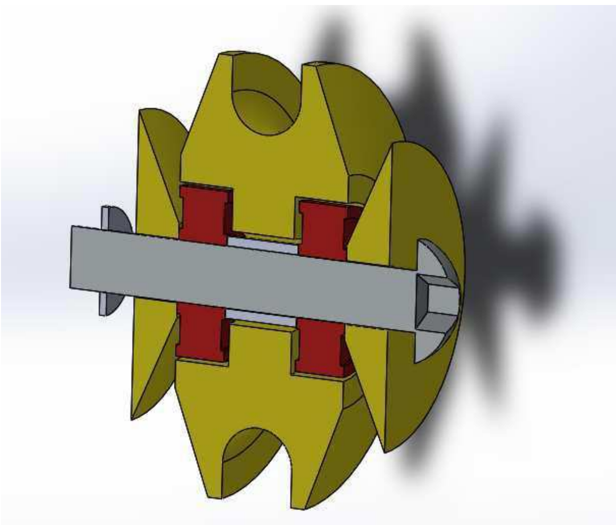


Image 33. Pulley design

The dimensions of the pulleys can be seen in the image below:

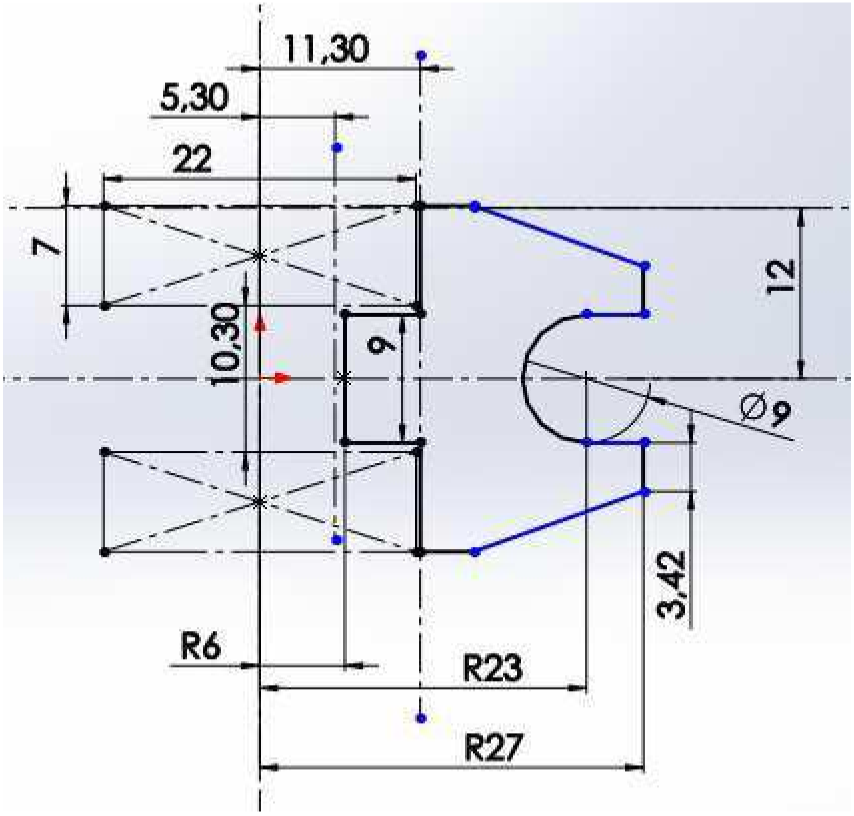


Image 34. First central part design

Here is how it looks like in the structure:

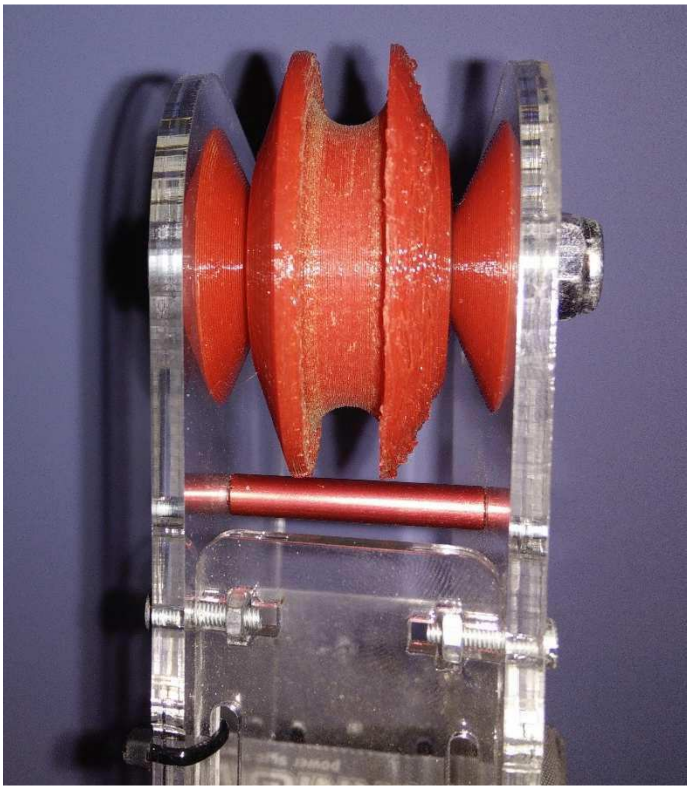


Image 35. Top view

Also in this picture below the different parts can be seen in an exploded view:

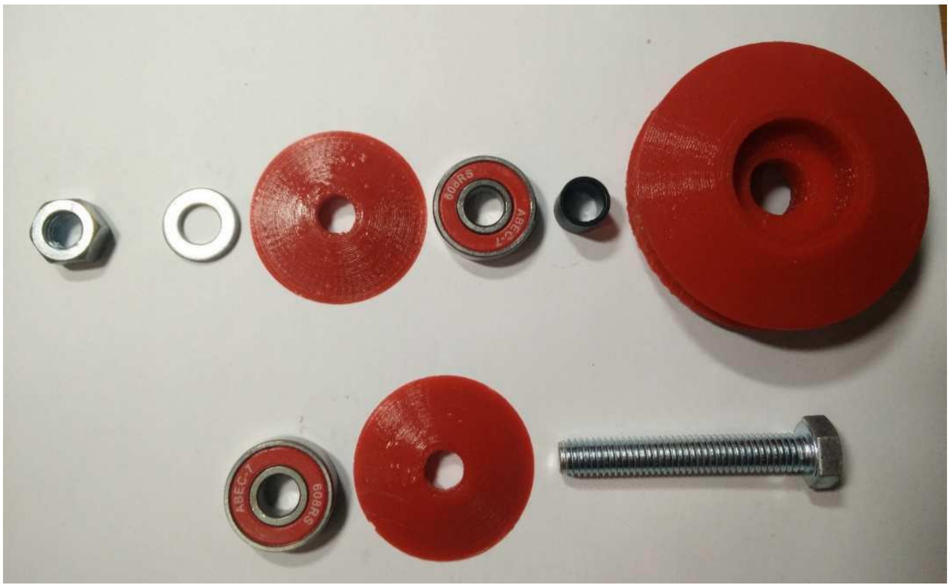


Image 36. Parts of the side pulleys

After many hours testing the pulleys, the rope was having a little bit of friction on the sides, so a new design is 3D printed with more open sides:

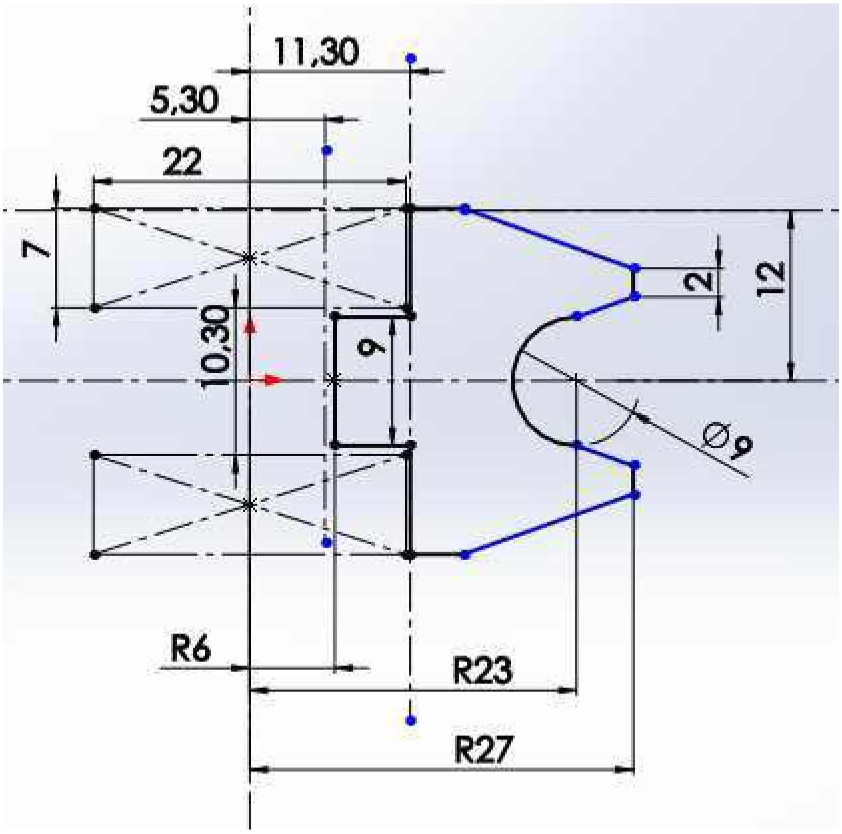


Image 37. 2nd design of the pulley central part

This new pulley lets the rope move easier without generating friction on the sides.

Here is a comparison between the old (upper part) and the new central part (bottom):

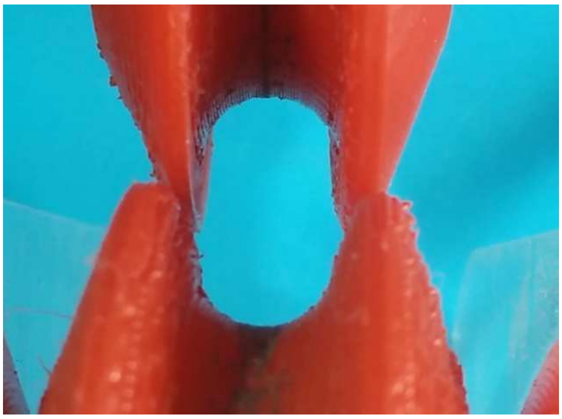


Image 38. Central part comparison

3.5.3. Motor enclousure

The motor is placed in between the two plates, attached from behind with 4 metric 3 screws and resting on the other side with a ball bearing to reduce the bending forces on the back plate.

In the next tree images you can see the design, the final placement of the motor and the rear screw holes to hold the motor in place:

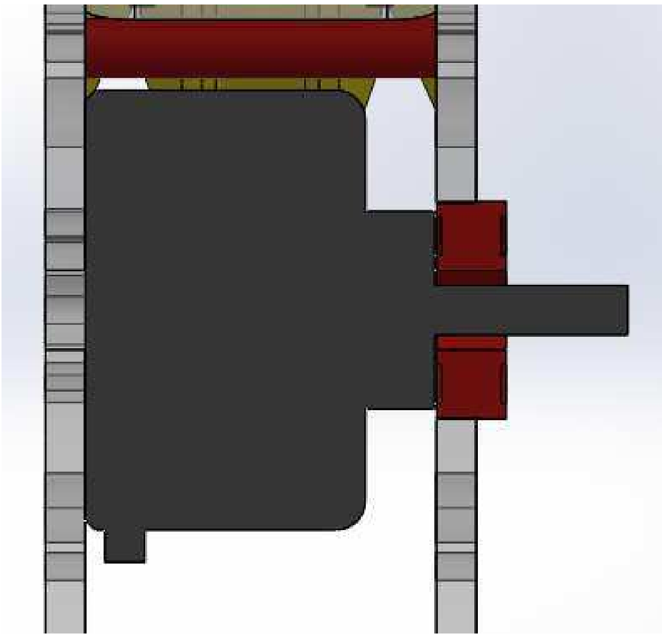


Image 39. Side cut view of the motor

Image 40. Motor placement

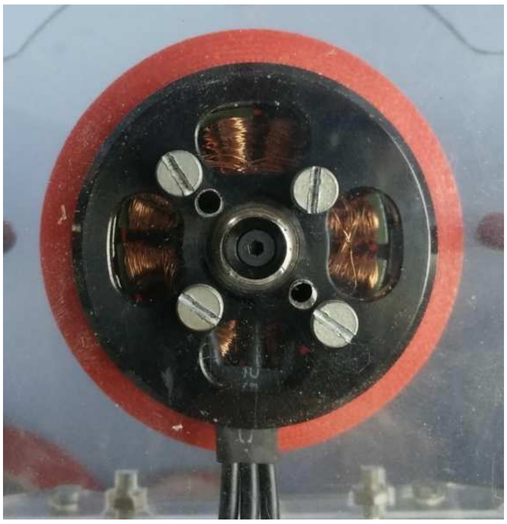


Image 41. Back holes for the motor

In order to create more friction, different motor covers are designed and tested:

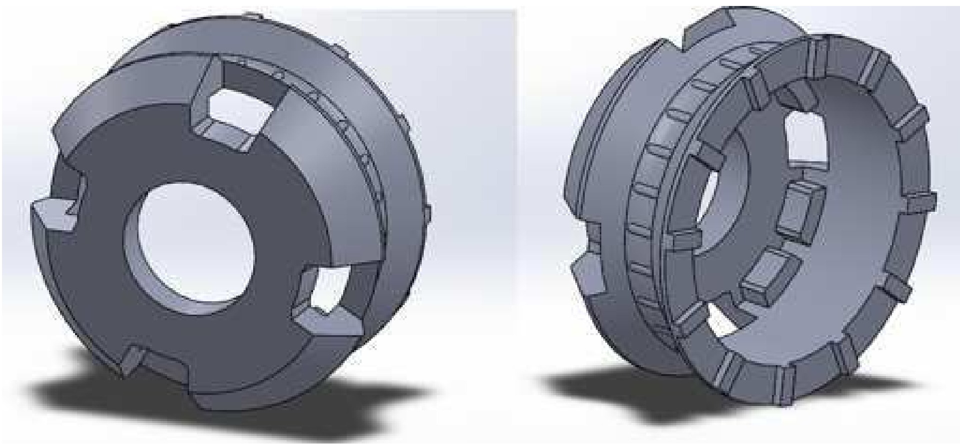


Image 42. SolidWorks designs for the motor cover

The motor has different dents that push the inner side of the cover, transmitting the rotation to the cover and pushing the rope to one side or the other.

After many hours testing different designs, we found out that it had to have no dents on the outer side of the cover. With that, we achieved full contact which generated complete friction with the rope. We printed it with a flexible PLA compound that made it more elastic to the pressure of the rope.

Image 43. Final motor cover

In the image below, you can see different views of the part and the revolution sketch used to design it:

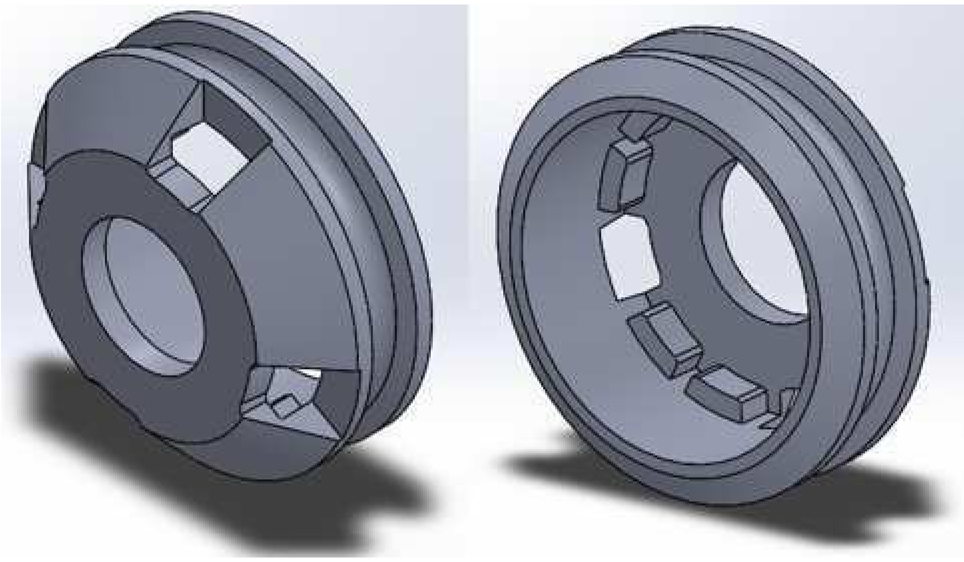


Image 44. Dimetric front view and back view

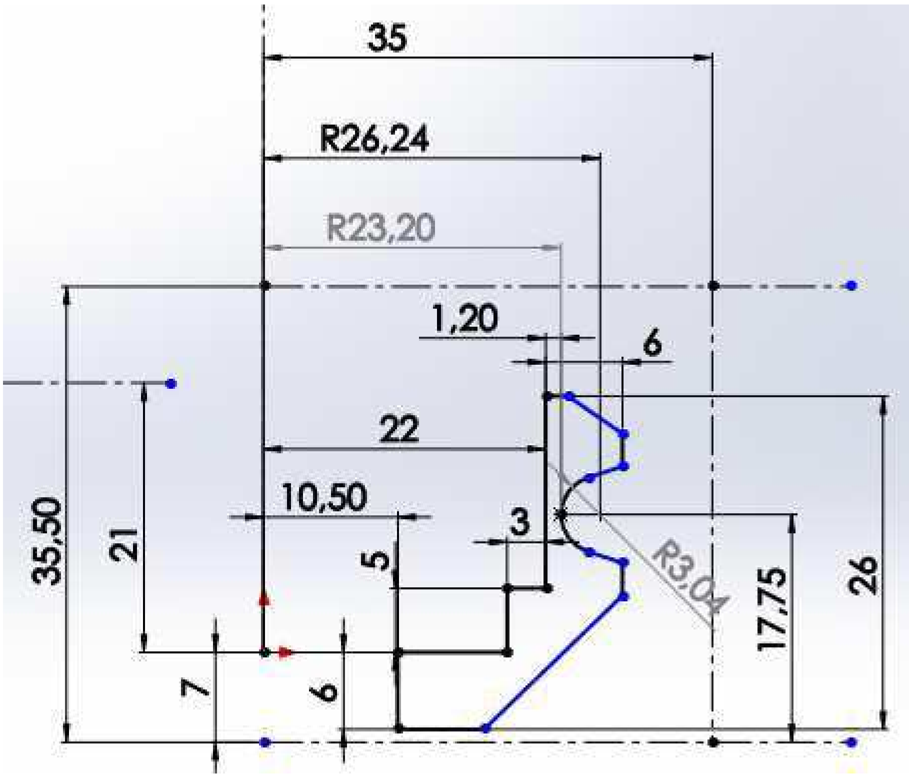


Image 45. Revolution sketch of the motor cover

As it can be seen the little wings on the back side are no longer printed, because at this speeds there is no enough air flow to generate a turbine air flow movement to the side part of the motor.

At the image below you can see the pulley without the central 3D printed part:

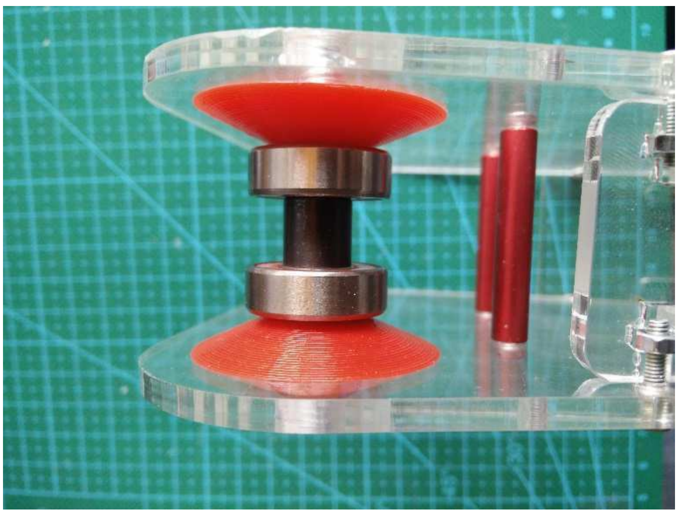


Image 46. Pulley without the central part

3.6. Build and test

To test the different positions of the pulley the screw must be disassembled and assembled again in another hole. After the 4 positions are tested it is clearly working with the top position of the pulleys, while at the bottom the motor does not have enough torque to push the rope due the increase of the friction.

The methacrylate frame is hard as glass and does not bend or break thanks to the different spacers. The vertical screw cracks a bit the support and this part have to be redesign.

The device holds the camera no matter how fast it goes without vibrations thanks to the width of the frame.

The device is working correctly and the transmission of the image (not included in this project) is working properly. So it can be used from the distance.

During the build and test it is included the above improvements explained.

3.7. Further improvements

In this part of the project are explained the different improvements that have to be done after the end of this project in order to have a better cablecam device.

3.7.1. Motor cover

We had to modify the motor cover in order to have more contact points with the motor, so it has more points of contact which reduces the pressure on each of them.

Modifying the refrigerations holes in order to suck more air and generate more airflow inside the motor.

3.7.2. Pulleys

The pulley positions must be redesigned to let an easy placement of the rope, while the frame must be out of the aluminum cnc machine in order to hold the different forces applied.

3.7.3. Structure

The points of anchorage of the screws are weak and do not support the pressure when screwing them in place, breaking the methacrylate plates if the load is high enough. Here is an example:

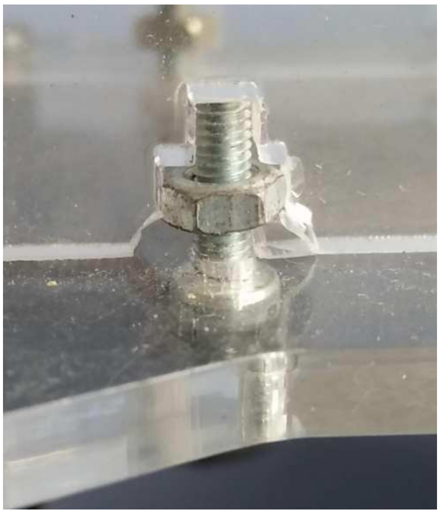


Image 47. Screw problem

The screws must be longer in order to have more methacrylate to hold the pressure, this 2mm are not enough. That is why zip ties are placed all around, to add more safety if the support breaks. They are not really necessary as while one screws is tighten more than enough, there are in total 16 M3 screws holding the bottom plate in place.

Increasing the high to at least 5 mm will help to increase the holding force of the methacrylate during the build of the bottom plate. Also little dents from the horizontal plates to get in the plates side instead of screws holding a shear stress must be redesigned.

Remove the diagonal cut of the side panels to reduce the stress.

3.7.4. Standard Rails

Using small screws to hold the gimbal is a good short term idea to try if the cablecam works or not. But not the best long term idea, as other devices can be placed underneath like a GoPro gimbal instead of this huge gimbal with the Sony camera. The frame must be redesigned in order to place underneath the common used rails as other camera rigs uses:

Image 48. Example of rails

4. Electronic PCB

There are multiple steps to end with a good PCB to place in front of the camera. This section will cover from the first steps of the circuit design and the final enclosure. There are commercial applications like "#map2" but at much higher cost, and the company wanted to have a board of its own, to be able to modify and adapt to any case, while open source.

4.1. Hardware

In order to create the PCB which can control the camera, it takes different elements, from the first breadboard that can program the attiny85, the software files that allows to upload the code to the microcontroller, different leds and resistors, if the led is a high power it needs a transistor to control it, as the microcontroller outputs a maximum of 50mA and it needs round 90mA as specified in their datasheets. [4][5]

- Arduino Tinny85
- Arduino UNO + bootloader
- Attiny programing breadboard
- Capacitor
- ATTINY85
- Resistance
- IR leds
- Servo Futaba J female connector

4.1.1. Programmer

In order to program the attiny85 it must be done through another Arduino UNO as a programmer.

We need these items:

- Arduino
- Breadboard
- Straight pins
- ATtiny85
- 8dip chip support

- wires
- 10uF 16V electrolytic capacitor, between power and ground from the Arduino UNO

Connecting the Arduino to the ATtiny as follows:

- Arduino +5V ---> ATtiny Pin 8
- Arduino Ground ---> ATtiny Pin 4
- Arduino Pin 10 ---> ATtiny Pin 1
- Arduino Pin 11 ---> ATtiny Pin 5
- Arduino Pin 12 ---> ATtiny Pin 6
- Arduino Pin 13 ---> ATtiny Pin 7

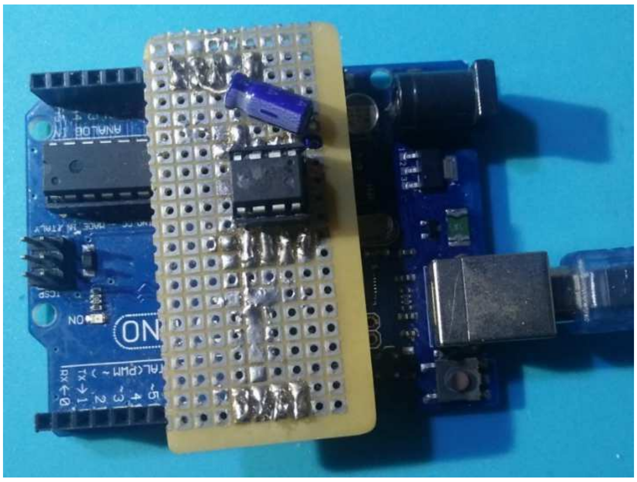


Image 49. Arduino with breadboard on top

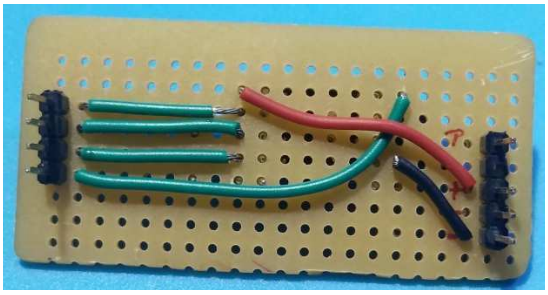


Image 50. Breadboard bottom side

Then, select the "ArduinoISP" sketch from the "Examples" menu. Upload the sketch to the Arduino as any other sketch, pressing the upload button. The Arduino is now configured as a serial programmer that can program other chips.

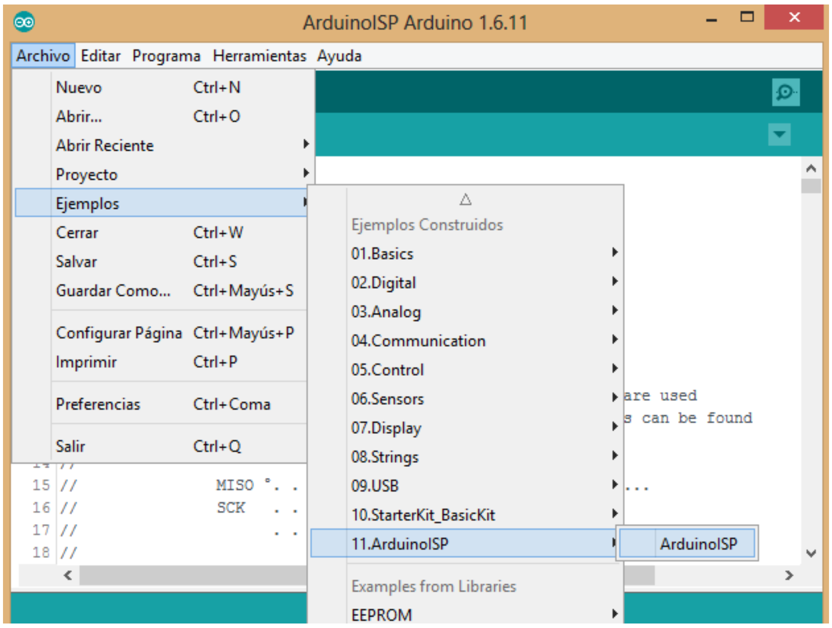


Image 51. Programming interface

From now on, this Arduino can upload any sketch to the attiny85 or attiny45.

4.1.2. Attiny85 Cores

Arduino environment can program different microcontrollers, but third party microcontrollers created the software files to aloud Arduino to program them. In the link from the code.google.com site [6] there are the files to download. Once are downloaded, they must be placed inside the hardware folder, and then the option to choose the Attiny85 appears:

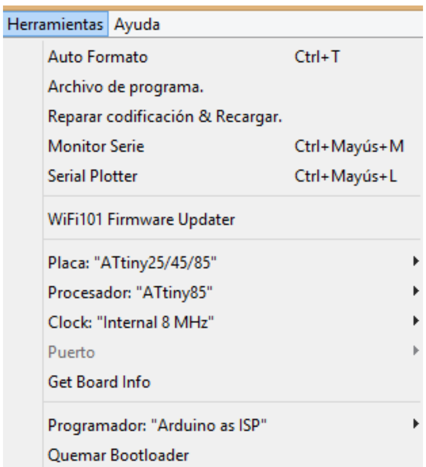


Image 52. Board options in IDE

4.1.3. Circuit

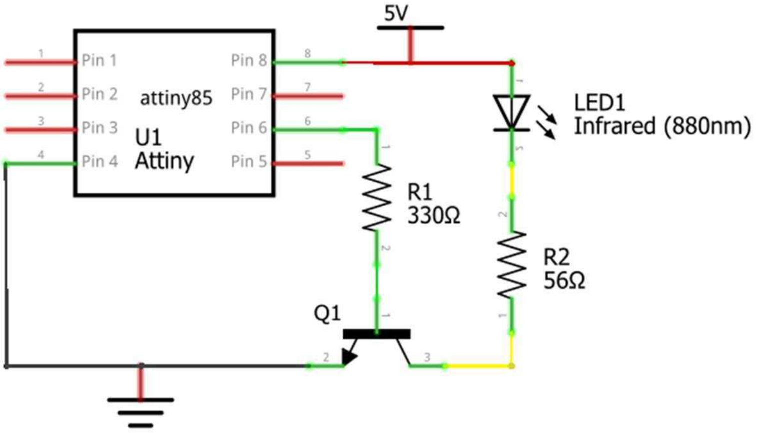


Image 53. Proposed circuit

The circuit to send the IR commands is pretty simple, there is a servo cable connected to the 5V, GND and the input data pit from the Attiny, which is the pin 5. The pin 6 activates the transistor that will let the led light up with more than the only 50mA provided by the output pin, instead it can take 90mA with the 56 ohm resistor.

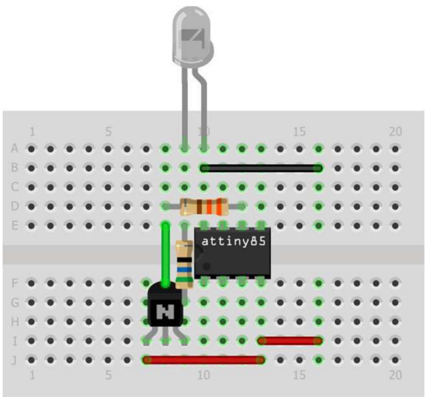


Image 54. Protoboard test

This is the basic drawing that can be designed in the Fritzing program, just by dragging the components and dropping them in the screen.

Down below you can see the first build to check if everything is correct:

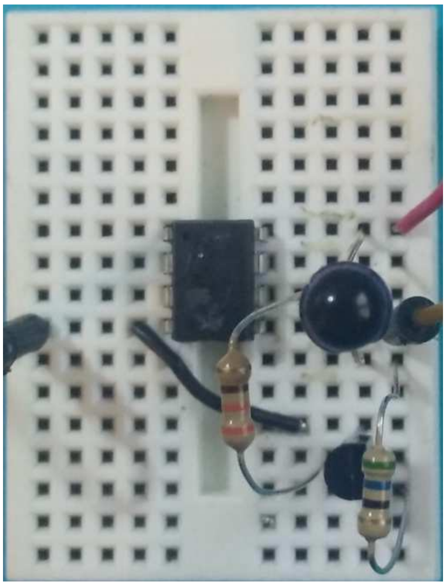


Image 55. Protoboard first prototype

Once the circuit is checked is time to solder the components on a micro perforated board, to check the components placement as you can see below:

As you will see later on, both resistors will be placed parallel and the path will cross from one side to the other to make a much cleaner image.

Image 56. Breadboard prototypes

Two different IR led are tested, the one that had more range is the purple one (usually used in TV remotes), looks like the high brigthness one (transparent body) does not work the same way, giving bad results during the tests.

Down below you can see one of the circuit breadboards tested:

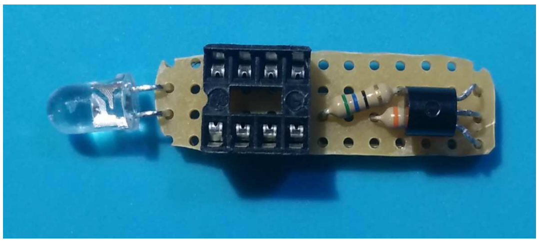


Image 57. Breadboard with IR led

4.2. Software

The program used to write the code is Arduino, thanks to Ken Shiriff to produce the fantastic library; this will help a lot to finish the project, as the libraries come almost done.

Image 58. Arduino IDE interface

4.2.1. The code

The code consists of the following:

```
#include <multiCameraIrControl.h>
```

First of all the library is called, so at any moment of the cicle it can be used.

```
int pwm_in=0;
#define PWM_PIN 0;
```

In order to read the receiver, the code will save the value into the "int" data and will read the signal from the pin PWM_PIN.

`Sony NEX5R(LED_PIN);` At this moment the function NEX5R is called and the program knows which kind of code have to use.

```
void setup() {
  // put your setup code here, to run once:
pinMode(LED_PIN, OUTPUT);
}
```

At this point the LED_PIN is configured as an OUTPUT, so, the Arduino prepares to don't read and only write by changing the internal pull up resistor.

`void loop() {` This statement starts the program that will run all the time, normally this part is needed if you want to run the code for more than once.

`pwm_in = pulseIn(PWM_PIN, HIGH, 20000);` This sentence stores the PWM value read in the pin PWM_PIN, which can be as long as 20000 micro seconds.

```
if (pwm_in>1700){
  NEX5R.shutterNow();
  delay (5000);
}
```

If the signal is received and the value is less than 1700 micro seconds it sends the code to send a picture. The camera have a little delay between shoots, it is calculated as 4.7 seconds, so this way you can set it up to shoot continuously.

```
if (1300>pwm_in){
  NEX5R.toggleVideo();
  delay(3500);
}
```

Instead, it takes 1.11 seconds to start recording and 3.4 seconds to stop recording and be able to start again, so this simple part of the program switches the video mode and wait to be able to modify it again.

```
  delay(10);
}
```

This little time adds program stabilization to the device, as it waits a little between the PWM reads through the PWM_PIN. If the program does not wait 10 micro seconds it is probably that the pwm_in data get corrupted by really fast reading.

4.2.2. Checking the reading

In order to check if the code is working properly, there is one Arduino with a simple IR reader that will record all the bytes sent by the Attiny85.

The sensor used is a TSOP48 infrared reader:

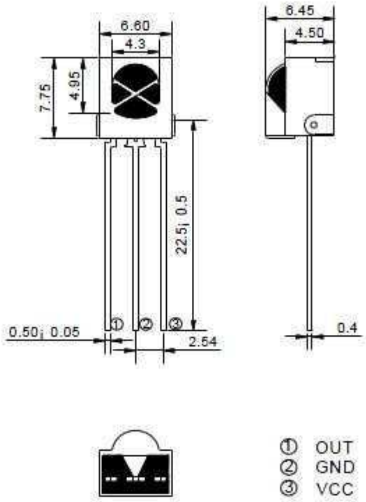


Image 59. TSOP48 IR Reader

The connections are:

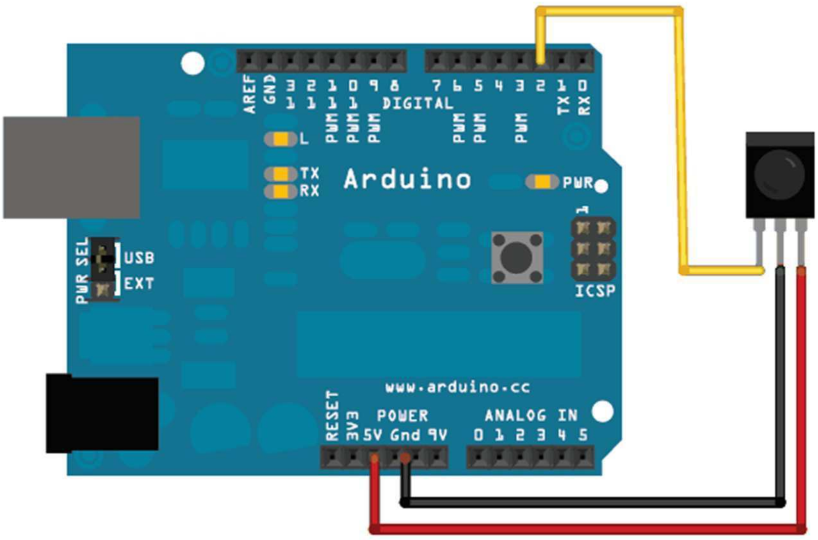


Image 60. Arduino UNO IR Reader

This receiver has one of the easiest connections of all sensors, as you don't need the protoboard to place it. Just a small wire attached to the bended left leg of the receiver to any free digital input.

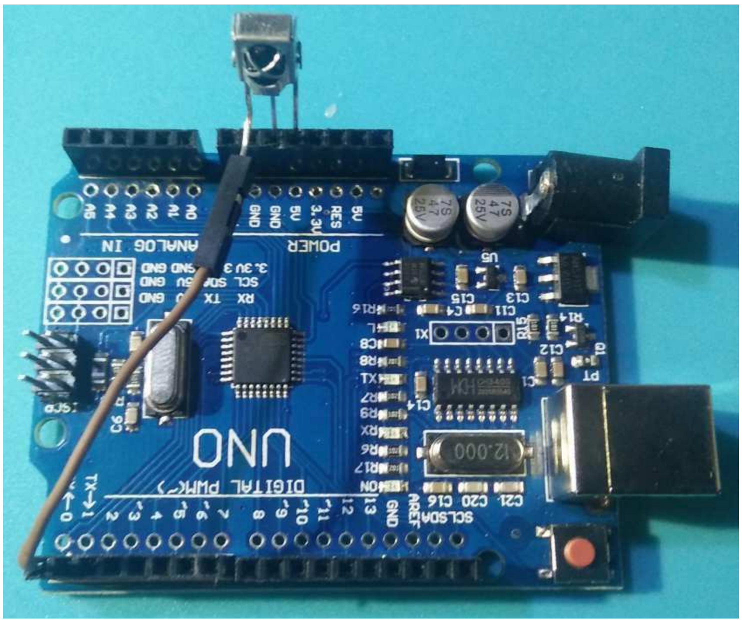


Image 61. Arduino IR Reader

In this case the receiver is Connected to the Input 0, so in the program this have to be checked.

The code is used to check the values sent:

```
#include <IRremote.h>

const int RECV_PIN = 6;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
  irrecv.blink13(true);
}

void loop() {
  if (irrecv.decode(&results)) {
    if (results.decode_type == NEC) {
      Serial.print("NEC: ");
    } else if (results.decode_type == SONY) {
```

```
      Serial.print("SONY: ");
    } else if (results.decode_type == RC5) {
      Serial.print("RC5: ");
    } else if (results.decode_type == RC6) {
      Serial.print("RC6: ");
    } else if (results.decode_type == UNKNOWN) {
      Serial.print("UNKNOWN: ");
    }
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

And this is the kind of serial output we expect:

Image 62. IR read example

4.3. Printed Circuit Board

In order to create the board, Fritzing program is used, it is fast and really easy to use.

4.3.1. Design the PCB

The way the components are placed is also by drag and drop, and it accomplishes the standards in electronic boards design. Sizes of the holes, width of the path, silk screen… Everything is controlled by the program.

Here you can see a little bit of the process to create these boards:

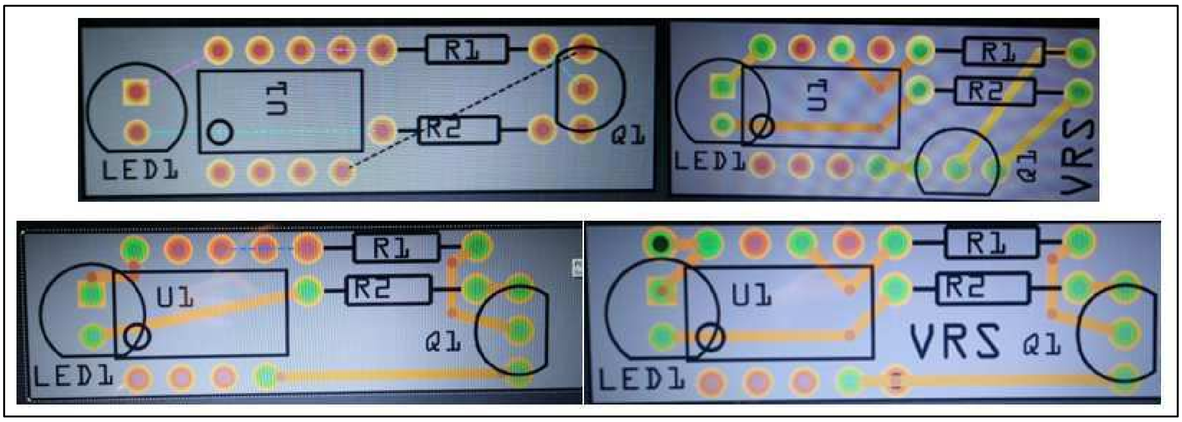


Image 63. PCB process examples

After many iterations and checks with the company the board design is finished:

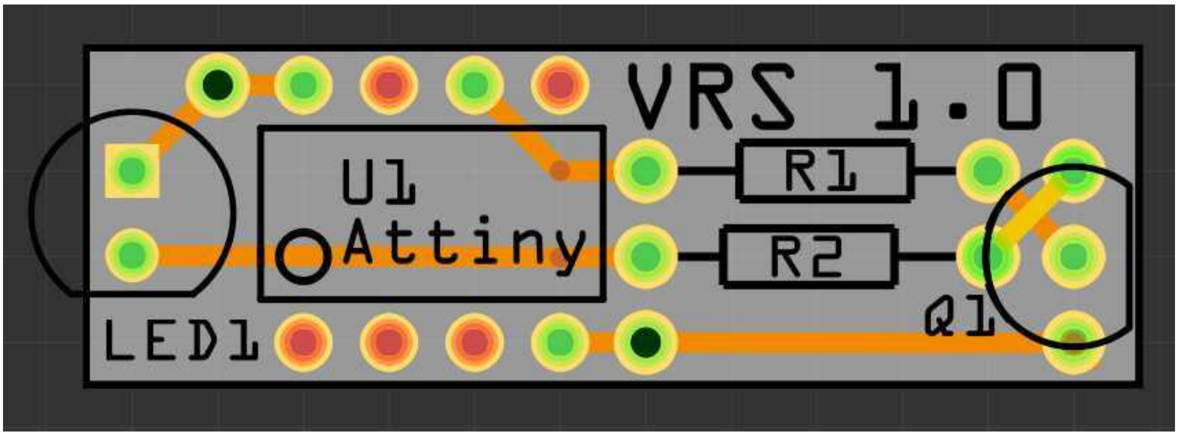


Image 64. VRS PCB

Notice there is no 90 degree angle corner, it uses two sided board, in order to make the connections from the transistor easy, so there is no crossed resistors like in the breadboard design.

All gerber and related files are attached in the CD.

4.3.2. Soldering the components

As you can see below, the board is covered with a purple color and the holes are standard sizes in order to fit all the components legs without any problem:

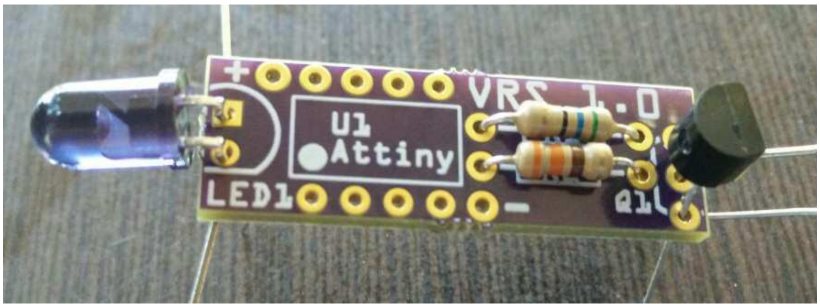


Image 65. Testing the components position

In the image above, notice the transistor is in the opposite way it should, because in this occasion when the transistor is bended it gets to a higher position, as it cannot turn as much as a flat face. The high is annotated for further work with the 3D cover.

The way the components are soldered comes from the lowest to the highest one, starting with the resistors, then the microcontroller support, transistor and finally the IR led.

A servo connector is added under the board to connect it via a servo connector extension to the receiver, via the pin 5 of the attiny, as you can see in the side view below:

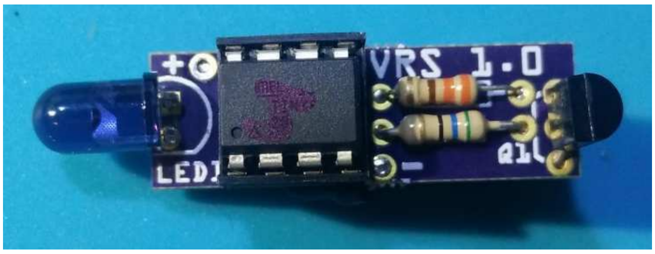


Image 66. First PCB finished top view

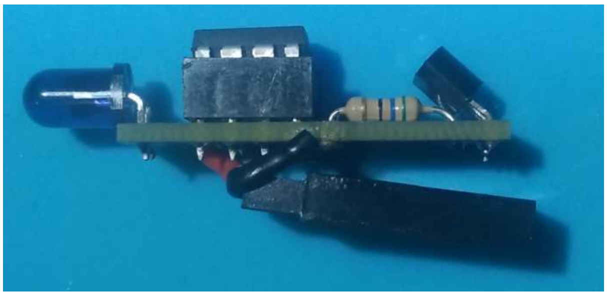


Image 67. First PCB finished side view

As the first board is finished it was covered with a thin orange shrinkable tube, to cover the components from the moisture.

Image 68. PCB with cover

As the company that produced this PCB sells the boards by sizes, like A5 board, the company wanted the full area for further development, so a total of 45 PCB are made:

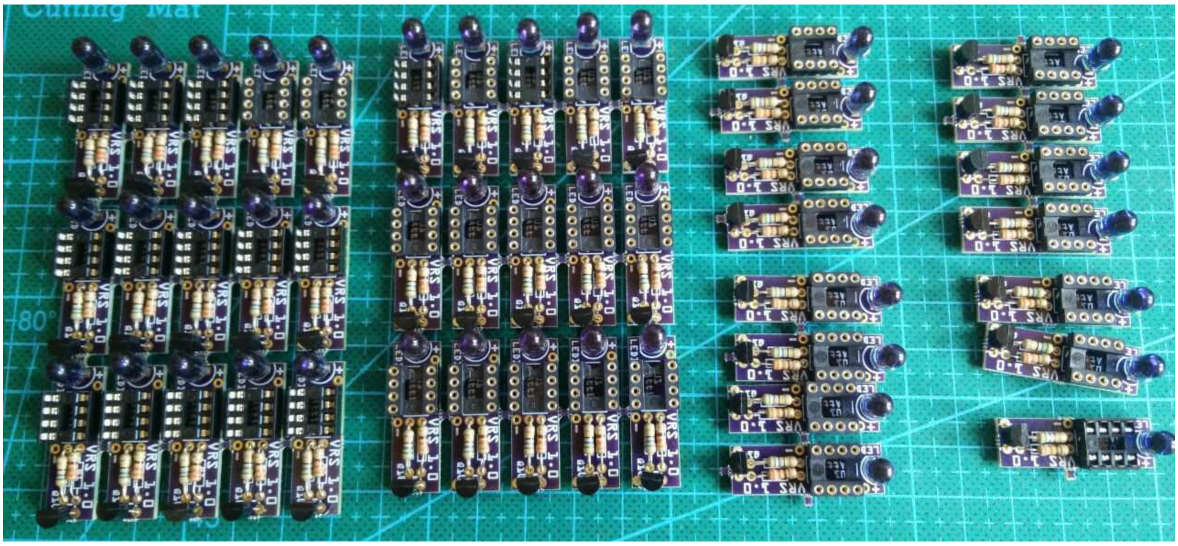


Image 69. 45 PCB finished

The company wants to sell this PCB to any other company or use them in different platforms. The good aspect about this board is that the Attiny85 can be programmed with any of the 40 KHz protocols for cameras (Sony, RC5, RC5 and raw codes protocols). Also, this little microcontroller have 5 more free pins to read another channel from the receiver or control directly the ESC or avoid collision with different sensors.

4.4. 3D support

In order to protect and be able to attach the VRS to almost any place, a 3D printed cover is designed to fit inside the PCB.

4.4.1. Initial cover

Down below you can see a transversal cut and a back view of the final cover:

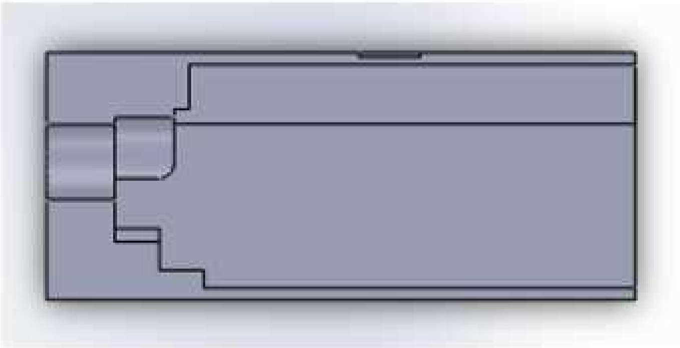
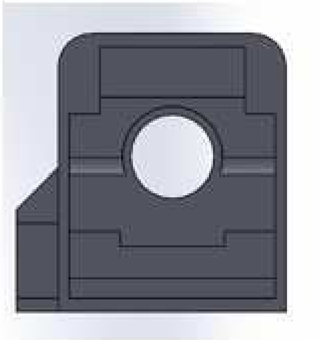



Image 70. Side and back view

This is achived by lot of try and error prints, because the 3D printer technology is not 100% perfect and you have to play with tolerances from the machine:

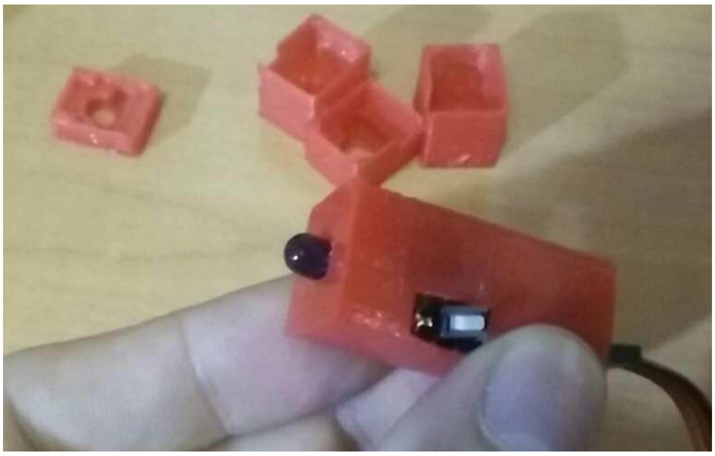


Image 71. PCB in the 3D printed cover

4.4.2. Side button cover

The steps to get the perfect placement took over many iterations, down below you can see the different steps used to create it with a dimetric view:

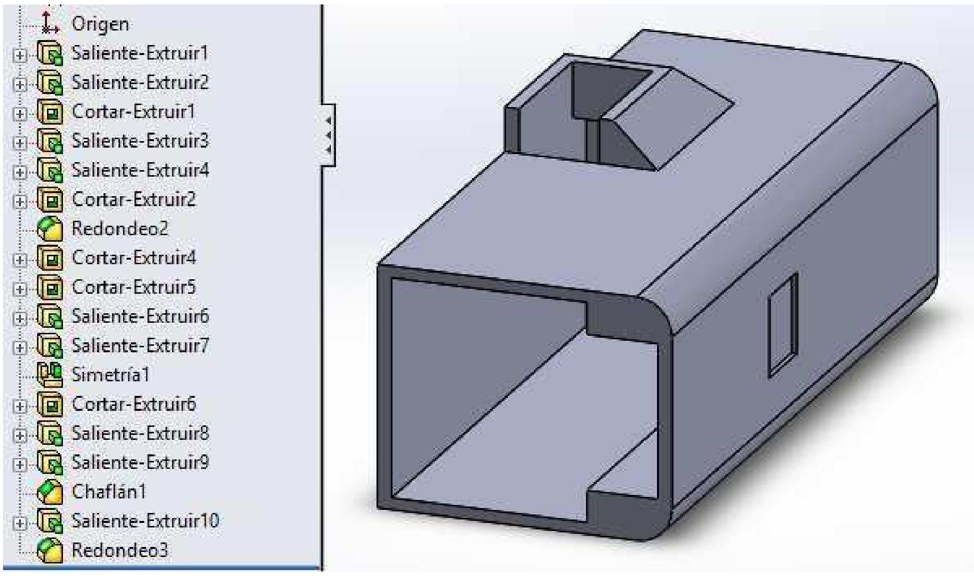


Image 72. Dimetric view and steps

Notice the side button, it is placed as a step up input, so when the button is pressed it reads a 5V input, and then, depending on the time it is pressed it takes a picture or start/stops the video. This way the control of the camera can be done manually with a power bank.

4.4.3. Camera support

The camera have a metallic support already on top, hold by the objective, this is used to create a kind of "L" shape to place the VRS directly in front of the IR receiver of the camera.

Image 73. Upper part of the camera in detail

The VRS size is 13.6mm wide and 16.6mm high, this is because it have to take in account the nozzle diameter, which is 0.4mm and have to work with multiples. The VRS support is exactly the same size, as the printer prints a little bit wider (0.1mm) it will be locked by itself to the camera and can be removed if needed.

This support is hold by two screws over the objective of the camera.

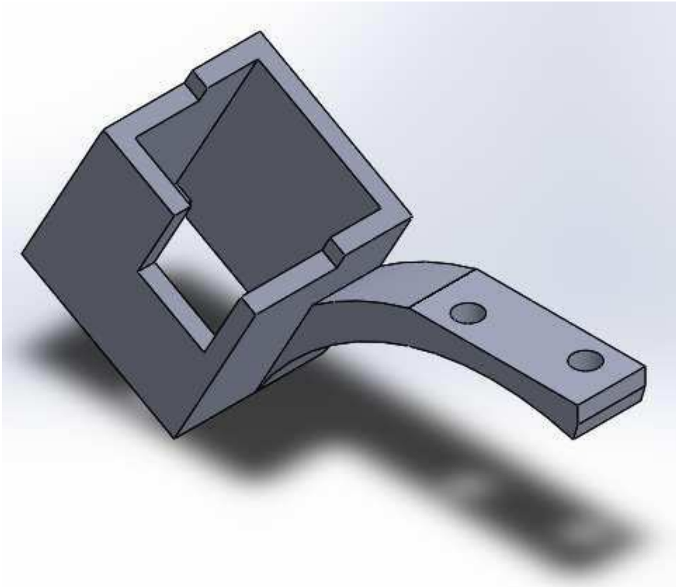


Image 74. VRS final support

After many iterations and different field tests, it got 3 different designs, at first the VRS was too low, then it was at a good high, but not pointing to the sensor (it worked as well), the last one have a 10 degree input to the sensor, which helps to improve the reception during sunny days.

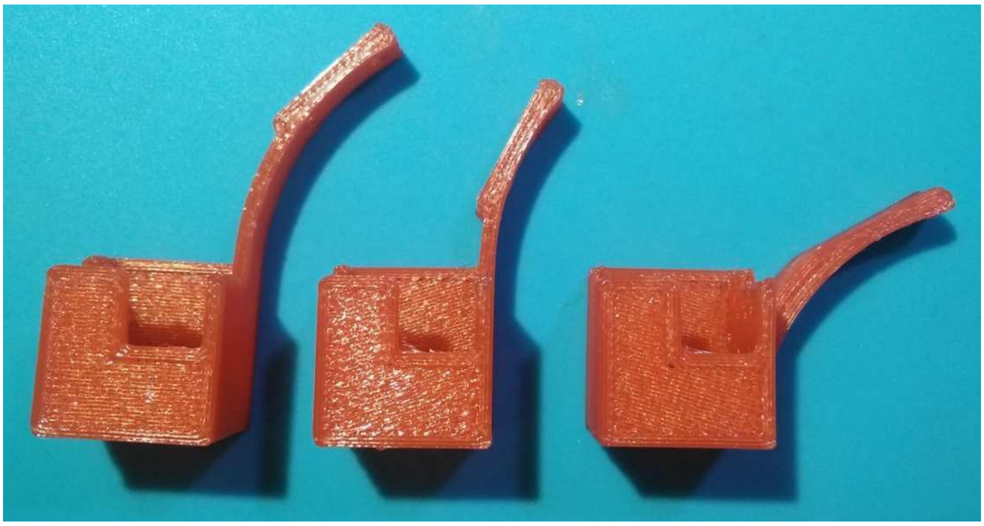


Image 75. Different iterations of the VRS support

At the end the support points directly to the sensor as you can see below, it also produces some shade which helps in the reception of the code:

Image 76. VRS support from the top

Image 77. VRS support detail

5. Plan Update

It should be mentioned that due to the extension of the detailed project planning made in the draft is not valid and therefore another plan is realized to achieve the necessary goals.

5.1. Draft plan

As it can be seen in the draft project, the plan was to have the cablecam completed by the end of January 2017:

	Nombre de tarea	Duración	Comienzo	Fin	Prede
2	**Gather all the information**	**156 horas**	**lun 25/07/16**	**mar 06/09/16**	
3	Observe allready made SliderCam systems	60 horas	lun 25/07/16	mar 09/08/16	
4	Choose frame design	40 horas	mar 09/08/16	lun 22/08/16	3
5	Choose all hardware components	40 horas	lun 22/08/16	mié 31/08/16	4
6	Buy hardware and electronic parts	8 horas	mié 31/08/16	lun 05/09/16	5
7	Design the circuit	16 horas	mié 31/08/1(	mar 06/09/1(	5
8	Deliver PreProject	0 horas	mar 06/09/16	mar 06/09/16	7

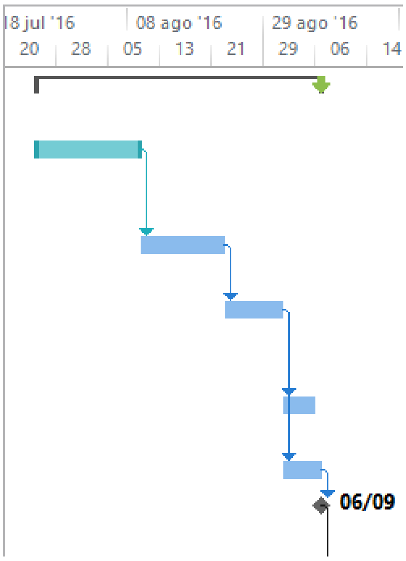


Table 2. 1rst part plan

	Nombre de tarea	Duración	Comienzo	Fin	Prede
9	**SliderCam Structure**	**184 horas**	**mar 06/09/1(**	**lun 31/10/16**	
10	Design the frame	40 horas	mar 06/09/1(	lun 19/09/16	8
11	Cut the frame	8 horas	lun 19/09/16	mar 20/09/1(	10
12	Build the frame	16 horas	mar 20/09/1(	lun 26/09/16	11
13	Test the cablecam	8 horas	lun 26/09/16	mar 27/09/1(	12
14	Design next generation	80 horas	mar 27/09/16	mié 19/10/16	13
15	Cut the frame	8 horas	mié 19/10/16	lun 24/10/16	14
16	Build the new frame	16 horas	lun 24/10/16	mié 26/10/16	15
17	Test the cablecam again	8 horas	mié 26/10/16	lun 31/10/16	16
18	Frame Completed	0 días	lun 31/10/16	lun 31/10/16	17

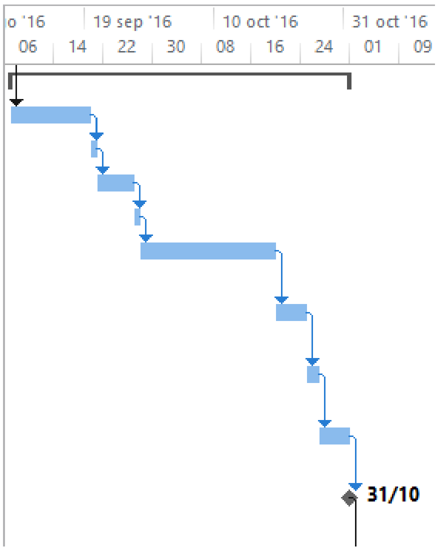


Table 3. Cablecam structure plan

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
19	**Wireless Camera Control**	**216 horas**	**lun 31/10/16**	**lun 02/01/17**	
20	Write test code	80 horas	lun 31/10/16	mar 22/11/16	18
21	Test on protoboard	32 horas	mar 22/11/16	mié 30/11/16	20
22	Test on Perforated Prototype Board	8 horas	mié 30/11/16	lun 05/12/16	21
23	Design the PCB	40 horas	lun 05/12/16	mié 14/12/16	22
24	Solder components	8 horas	mié 14/12/16	lun 19/12/16	23
25	Test VRS	16 horas	lun 19/12/16	mié 21/12/16	24
26	Design 3D case support	16 horas	mié 21/12/16	mar 27/12/16	25
27	Print the 3D case	8 horas	mar 27/12/16	mié 28/12/16	26
28	Test the 3D case	8 horas	mié 28/12/16	lun 02/01/17	27
29	VRS finished	0 horas	lun 02/01/17	lun 02/01/17	28

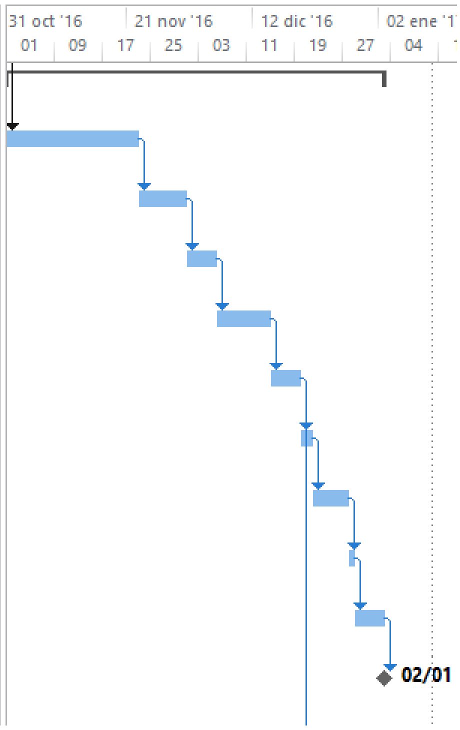


Table 4. PDB part plan

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
30	**Write the Project**	**74 horas?**	**mar 20/12/16**	**mar 10/01/17**	
31	Write the project	74 horas	mar 20/12/16	mar 10/01/17	24
32	Deliver Project	0 horas	mar 10/01/1	mar 10/01/1	31

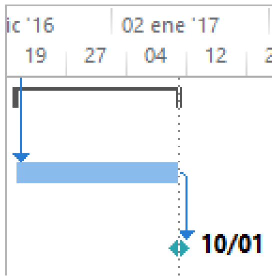


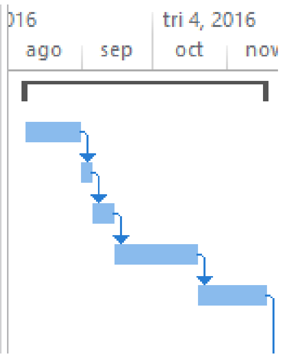
Table 5. Writing Project plan

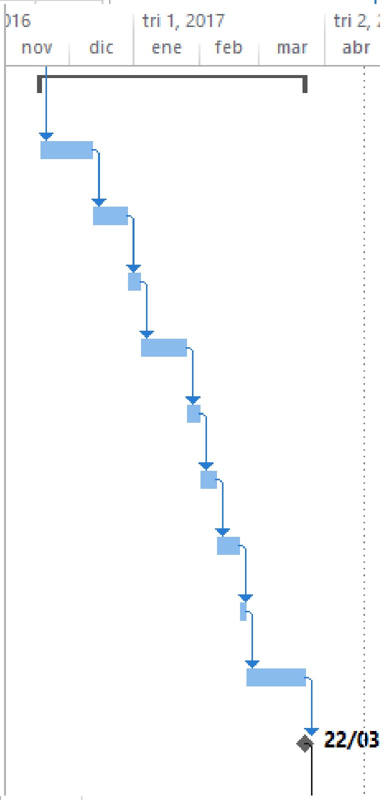
5.2. Plan Execution

The plan explained before is no longer valid due to the extension of the project. This is why we made a second plan updated.

The first part, until delivering the draft project is the same, as it was done in the plan time. Furthermore, the amount of hours printing, binding and taking the project to the university are included in the task 31.

	Nombre de tarea	Duración	Comienzo	Fin	Prede
9	**SliderCam Structure**	**176 horas**	**lun 08/08/16**	**mié 16/11/1**	
10	Design the frame	40 horas	lun 08/08/16	mié 31/08/1	
11	Cut the frame	8 horas	mié 31/08/1	lun 05/09/16	10
12	Build the frame	16 horas	lun 05/09/16	mié 14/09/1	11
13	Test the cablecam	62 horas	mié 14/09/1	mié 19/10/1	12
14	ReDesign next improvements	50 horas	mié 19/10/16	mié 16/11/16	13
15	**Wireless Camera Control**	**216 horas**	**mié 16/11/16**	**mié 22/03/17**	
16	Write test code	40 horas	mié 16/11/16	lun 12/12/16	14
17	Test on protoboard	32 horas	lun 12/12/16	mié 28/12/16	16
18	Test on Perforated Prototype Board	8 horas	mié 28/12/16	mié 04/01/17	17
19	Design the PCB	40 horas	mié 04/01/17	mié 25/01/17	18
20	Solder components	8 horas	mié 25/01/17	mié 01/02/17	19
21	Test VRS	16 horas	mié 01/02/17	mié 08/02/17	20
22	Design 3D case support	16 horas	mié 08/02/17	lun 20/02/17	21
23	Print the 3D case	8 horas	lun 20/02/17	mié 22/02/17	22
24	Test the 3D case on CableCam	48 horas	mié 22/02/17	mié 22/03/17	23
25	VRS finished	0 horas	mié 22/03/17	mié 22/03/17	24
26	**Write the Project**	**74 horas**	**mié 22/03/17**	**mié 03/05/17**	
27	Write the project	74 horas	mié 22/03/17	mié 03/05/17	25
28	Deliver Project	0 horas	mié 03/05/1	mié 03/05/1	27





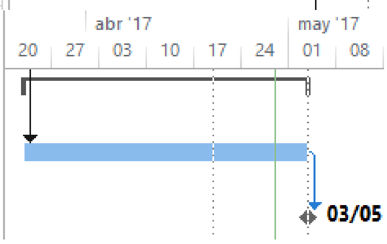


Table 6. New planning

Some activities have been removed, and the time applied to them has been reduced in order to stick to the plan presented. Also, the testing part during field operations was what took more time than expected: to change, redesign, build again, etc. Finally, the last design of the main frame is left for further build. The blue line indicates the critical path.

6. Conclusions

This project ended up with a fully working cablecam from which the author learned a lot.

Design and build a cablecam system it is not as easy as it was planned. However, different problems occurred during the process of testing and different redesigns have been made to accomplish the proper work of the cablecam.

The different delays in this project made it quite annoying, but with further preparation and knowledge, the timing and planning will improve.

In sum, the company is happy with the final version of the cablecam and is looking forward to see the upcoming improvements for it.

7. References

[1] Gimbal weight + camera weight at the description: http://www.hobbyreal.com/es-moy-steadg-s-32bit-brushless-handheld-3-axis-camera-gimbal-mount-for-5d3-gh4-a7s-dslr-p241681.htm?

[2] Calculation Chart of the required tension for the rope: http://videoworx.se/Streamline-Cable-Cam/streamline/Rigging.html

[3] Methacrylate properties:
https://www.construnario.com/bc3/3647/FT_METACRILATO.pdf

[4] datasheet arduino attiny85: http://www.atmel.com/images/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf

[5] datasheet high brightness IR led: https://cdn-shop.adafruit.com/datasheets/IR333_A_datasheet.pdf

[6] Attiny85 cores: https://code.google.com/archive/p/arduino-tiny/

Escola Universitària Politècnica de Mataró

Centre adscrit a:

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Mechatronics Engineering

DESIGN AND BUILD OF A CABLECAM

Economic Study

Eloi Vilalta
Tutor: Jordi Ayza

Spring 2016

TecnoCampus Mataró-Maresme

Table Index

Index

1. Cablecam budget

1.1. Components

Code	Item	Unit price	Quantity	Price
1.1	Methacrylate	20€	1	20€
1.2	Laser Cut	10€/h	1	10€
1.3	Brushless Motor	33.58€	1	33.58€
1.4	ESC	13€	1	13€
1.5	PLA	19€/kg	1 kg	19€
1.6	Screws	7€	1	7€
1.7	Spacers	0.20€	7	1.40€
1.8	Rope	1.39€/m	60 m	83.40€
1.9	Tie Ratchet	16.80€	1	16.80€
1.10	Battery	35€	2	70€
1.11	Transmitter	55€	1	55€
1.12	VRx+goggles	285€	1	285€
1.13	Antennas	7€	2	14€
1.14	Cables and connectors	15€	1	15€

Table 1. Cablecam components budget

643.18€ is the total cost of the components.

1.2. PCB

In this part, the PCB is analyzed per single board:

Code	Item	Unit price	Quantity	Price
2.1	Attiny85	1.4€	1	1.4€
2.2	Board	1.5€	1	1.5€
2.3	330 Ohm resistor	0.01€	1	0.01€
2.4	56 ohm resistor	0.01€	1	0.01€
2.5	5mm IR led	0.1€	1	0.1€
2.6	Transistor	0.02€	1	0.02€
2.7	Soldering	25€/h	0.33h	8.25€

Table 2. PCB single board budget

The total cost of a single PCB is 11.29€.

1.3. Human resourses

In this case, all the work was done by one single engineer which worked for a total of 630h to make this project possible.

Code	Item	Unit price	Quantity	Price
3.1	Single engineer	25€/h	630h	15750€

Table 3. Human resourses budget

The cost of the design and functioning of the cablecam is 15750€.

1.4. Amortization of equipment, instruments and software

Code	Item	Unit price	Quantity	Price
4.1	Computer	0.5€	400	200€
4.2	Software SolidWorks	1€	300	300€
4.3	Microsoft Office	1€	200	200€
4.4	Multimeter	0.5€	20	10€
4.5	Solder equip	1€	40	40€

Table 4. Amortization budget

The total amount of amortization of equipment is 750€.

1.5. Budget summary

Code	Item	Unit price	Quantity	Price
5.1	Components	643.18€	1	643.18€
5.2	PCB	11.29€	45	508.05€
5.3	Human resources	15750€	1	15750€
5.4	Amortization	750€	1	750€
5.5	Overhead	+20%	1	3530.25€

Table 5. Objective budget summary

In total the project costs 21181.48€.

Escola Universitària Politècnica de Mataró

Centre adscrit a:

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Mechatronics Engineering

DESIGN AND BUILD OF A CABLECAM

Annex

Eloi Vilalta
Tutor: Jordi Ayza

Spring 2016

TecnoCampus Mataró-Maresme

Image Index

Index

Annex 1. Arduino Code

```
/******************************************

 *

 * Name.......:  Versatil Remote for Sony

 * Description:  Read a PWM signal from the receiver and takes a
picture, does nothing or start/stop a video

 * Author.....:  Eloi Vilalta Revenaque

 * Version....:  1.3

 * Date.......:  24-12-2016

 * Project....:  DESIGN AND BUILD OF A REMOTE CONTROLLED CABLECAM

 * Contact....:  elrroi.v@gmail.com

 * Library....:  http://sebastian.setz.name/arduino/my-libraries/multi-
Camera-IR-Control/

 * License....:  This work is licensed under the Creative Commons
Attribution-NonCommercial-ShareAlike 3.0 Unported License.

 *               To view a copy of this license, visit
http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to

 *               Creative Commons, 171 Second Street, Suite 300, San
Francisco, California, 94105, USA.

 * Keywords...:  arduino, library, camera, ir, control, sony, interval

 * History....:  2016-11-24 V1.0 - release

 *               2016-12-15 V1.1

 *               2017-01-06 V1.2 - changing pin number, because the
wiring at the uP was easier

 *

 ******************************************/

#include <multiCameralrControl.h>


int pwm_in=0;
```

```
#define PWM_PIN 0;


Sony NEX5R(LED_PIN);


void setup() {
  // put your setup code here, to run once:
pinMode(LED_PIN, OUTPUT);
}


void loop() {
  pwm_in = pulseIn(PWM_PIN, HIGH, 20000); //Y
  if (pwm_in>1700){
    NEX5R.shutterNow();
    delay (5000);                  //takes 4.7s to take a picture, save
it ad be able to take another one
  }
  if (1300>pwm_in){
    NEX5R.toggleVideo();
    delay(3500);                //takes 1.11s to start recording and
3.4s to stop recording and be able to start again
  }
   delay(10); //added for program stabilization
}
```

Annex 2. SolidWorks Drawings

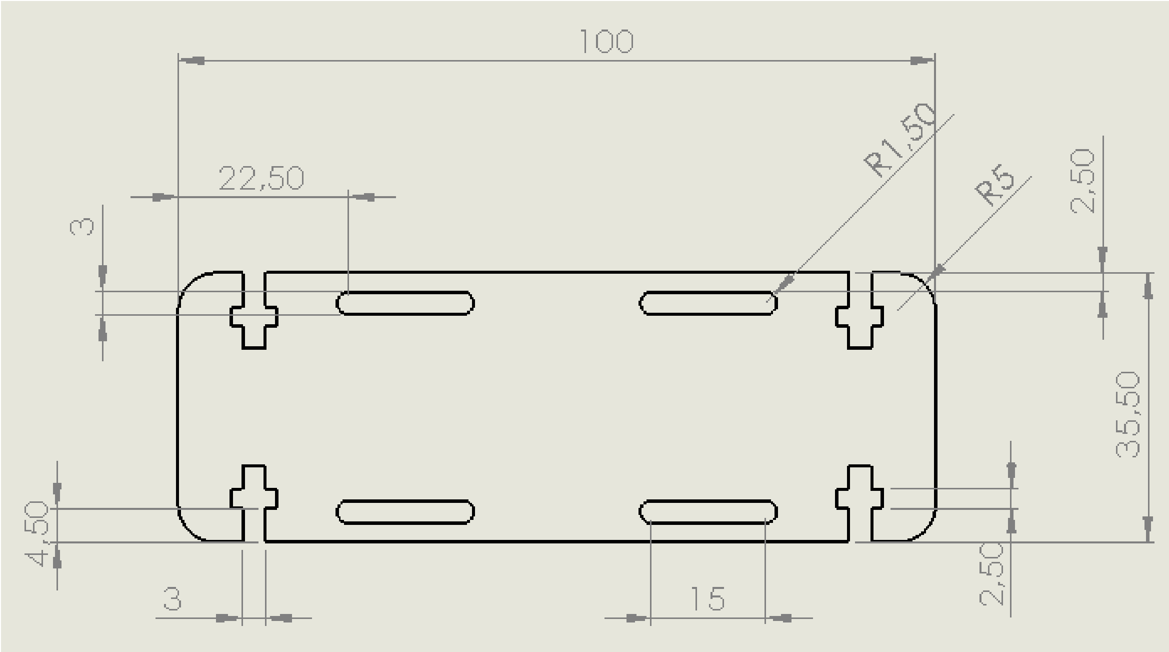


Image 1. Superior plates

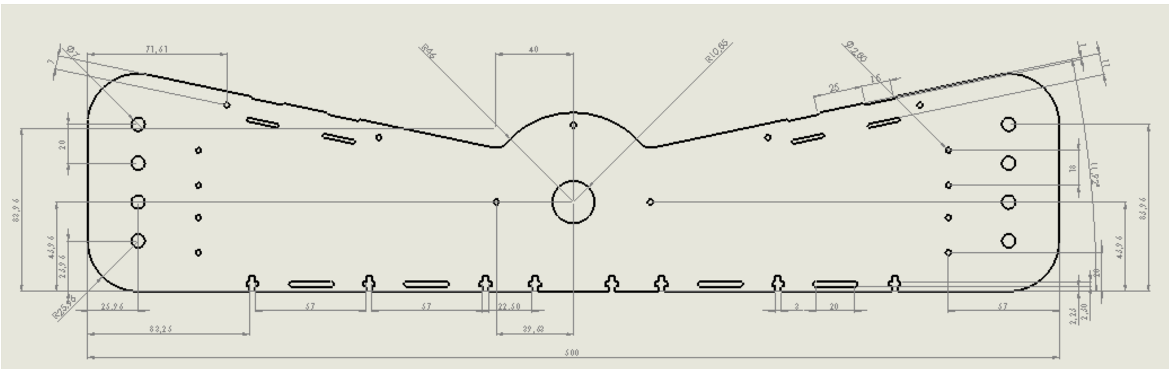


Image 2. Left side

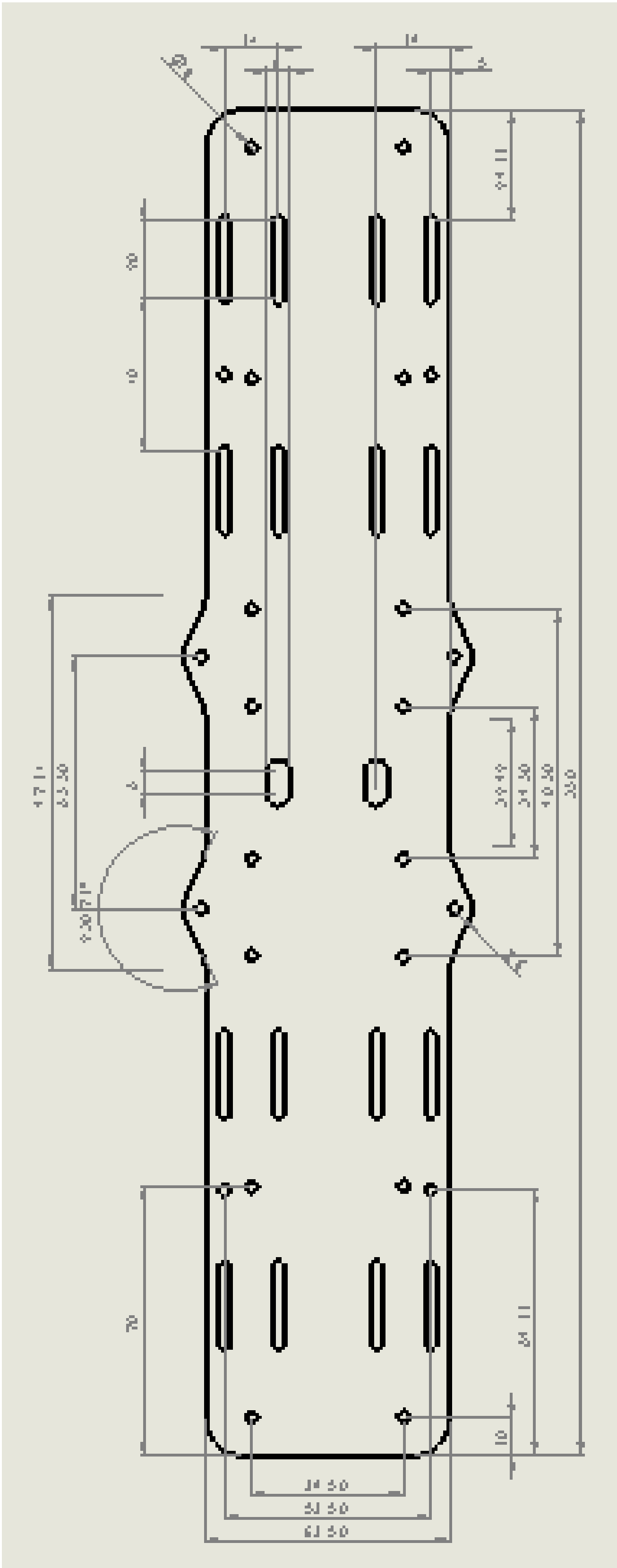


Image 3. Bottom part

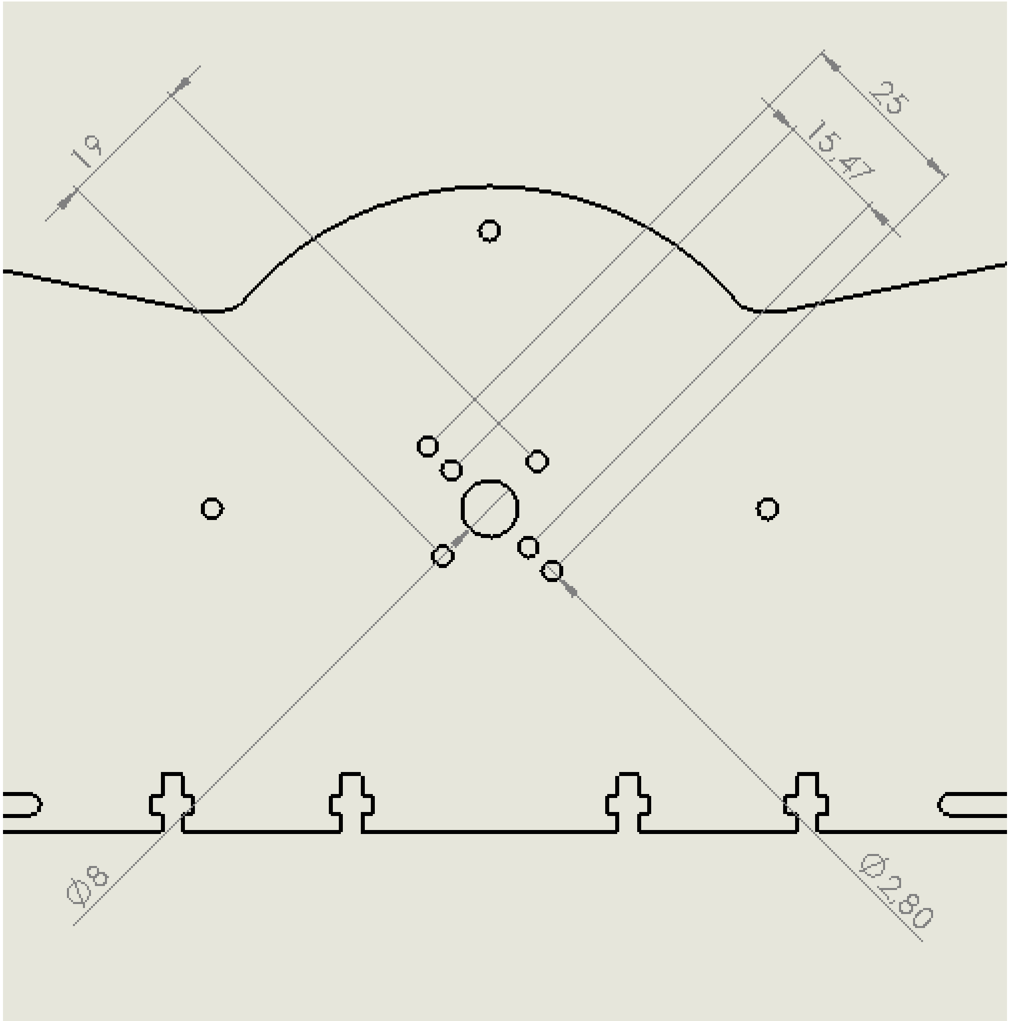


Image 4. Detail of right side

Annex 3. Cable cam components and BGC connections

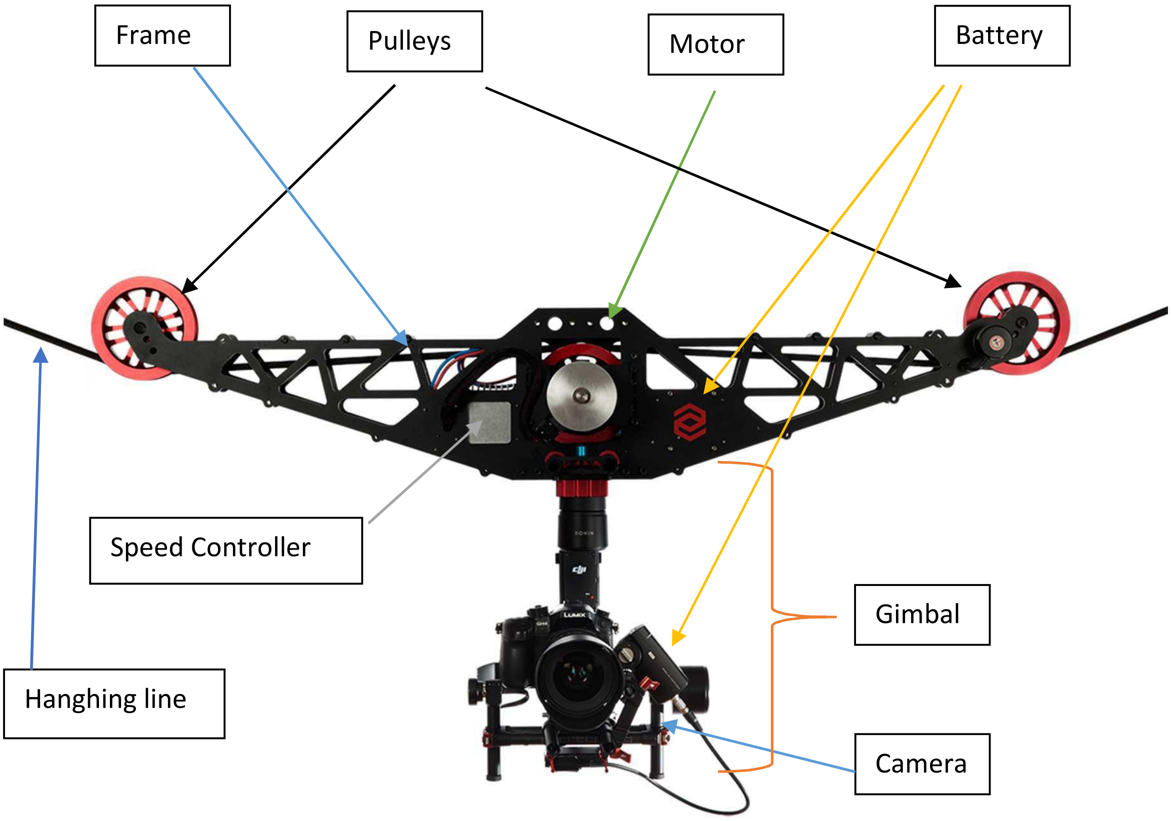


Image 5. Cable cam components

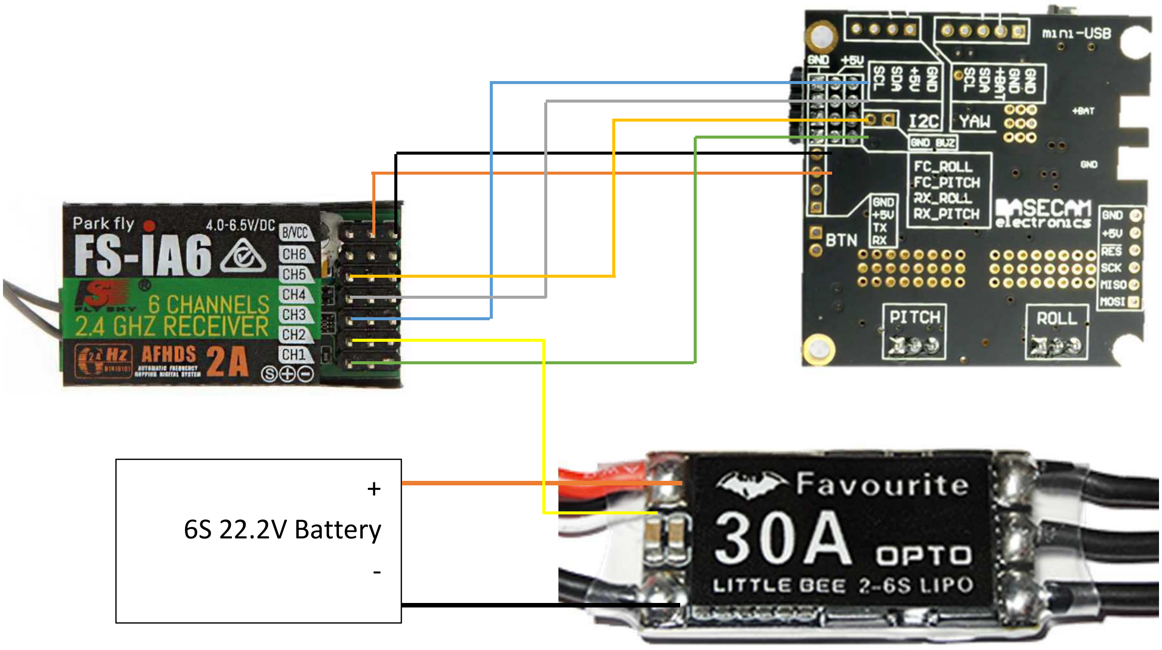


Image 6. Wiring

Annex 4. CD content

- Documentation (draft project, memory, annex)
- Gerber files for the PCB
- Arduino Codes and program
- SolidWorks files
- Excel for tension calculation
- Fritzing files
- Project planning file