

DESIGN AND DEVELOPMENT OF AN INNOVATIVE DRAWING ROBOT FOR @BRISTOL SCIENCE MUSEUM

AUTHOR: GUILLEM CABO PITARCH

TUTOR: DR. D. VICTOR SANTIAGO PRADERAS

COTUTOR: D. MATTHEW VENN

SEPTEMBER 2016



ABSTRACT

Development and implementation of a vertical drawing robot or autonomous V-plotter for represent the energy generated by the roof top solar array of AT-Bristol Science museum with artistic patterns.

This project has been performed as part of internship at Matt Venn Engineering and with the support of RS Components.

RESUMEN

Desarrollo e implementación de un robot de dibujo vertical o V-plotter autónomo que representa la energía generada por las placas solares del museo de ciencias AT-Bristol en patrones artísticos.

Este proyecto se ha realizado bajo el convenio de prácticas con la empresa Matt Venn Engineering y con el patrocinio de RS Components.

RESUM

Desenvolupament i implementació de un robot de dibuix vertical o V-plotter autònom que representa l'energia generada per les plaques solars del museu de ciències At-Bristol amb patrons artístics.

Aquest projecte se ha realitzat baix el conveni de pràctiques amb l'empresa Matt Venn Engineering y el patrocini de RS Components.

ACKNOWLEDGEMENTS

Special appreciation to my family and Elena for believe in my and encourage me to do my best every day.

I want to this opportunity to gratefully acknowledge to Matt Venn, Angel Andres Palau and the other colleges of the co-working that help me day by day to grow as engineer.

Thanks to all the other people involved in this project here, in Valencia, and in Bristol.

INDEX

1. Introduction and objective	10
2. Description	11
2.a State of the art	12
2.b Customers specification	14
2.c Project agreement	15
2.d Sponsorship agreement	19
2.e Technical specifications	20
3. Project development	22
3.a Simulations	23
I. Tensions	23
II. Errors	25
3.b Software	27
I. LinuxCNC/Machinekit configuration	27
II. Communications	29
III. Pattern generation	30
3.c Mechanical design	37
I. Cabinet & frame	37
II. Gondola	38
III. Motors setup	43
IV. Paper roll	46
V. Guide pulleys	48
VI. Charging Station	50
VII. Calibration tool	51
VIII. Other parts	52
3.d Electronics	53
I. PSU	53
II. BeagleBoneBlack	53
III. BBB breakout board	55
IV. Gondola	57
V. Big digits	58
VI. Charging setups	58
4. Plans	62
5. Budget	73
6. Review and future improvements	76
7. Conclusions	78
8. Bibliography	79
APPENDIX I, QUICK SETUP USER MANUAL	83

APPENDIX II, MATLAB CODE	89
APPENDIX III, RESUME OF SIMULATIONS AND DESIGN PARAMETERS	113
APPENDIX IV, MACHINEKIT CODE MODIFICATIONS	117
APPENDIX V, PCB SCHEMATICS	133
APPENDIX VI, PLANS OF OTHER MECHANICAL COMPONENTS	143

INDEX OF FIGURES

Figure 1 - Energy MegaDrawz, At-Bristol museum September 2016.	10
Figure 2 – Hektor Draws a Landscape, Righthd by Alex Rich & Jürg Lehni, 2003	12
Figure 3 - RS Components	19
Figure 4 - Statics problem that define the position of the gondola (4).	23
Figure 5 – Output of function quadtpmap(500,200,240,180) on top side and zoom on the bottom side.	24
Figure 6 - Output of importingjpg for test pattern square_circle_triangle_100x100.png for homing x 2 units homing y-2 units error.	25
Figure 7 - RAM memory usage of BeagleBoneBlack running Machinekit with python interface instead of Axis GUI	27
Figure 8 - Axis GUI Linux CNC	29
Figure 9 - Robot dashboard on Cursivedata server (18).	30
Figure 10 – Code and output of Matlab patter generation a.m.	31
Figure 11 – Set 1 of patterns generated with Matlab	32
Figure 12 - Set 2 of patterns generated with Matlab	33
Figure 13 - testpat_cccc designed with Solidworks	34
Figure 14 - Input image for png2gcode_4levels_horizontalfill.m	35
Figure 15 -Trajectory generated with png2gcode_4levels_horizontalfill.m using figure 6 as input and web gcode simulator nraynaud.github.io/webgcode/	35
Figure 16 - Petals pattern generated by CursiveData sever from the solar panels energy generation of AT-Bristol museum from 22 to 28 of September.	36
Figure 17 - Render of the final design approved by both parts	37
Figure 18 UHMWPE Dyneema SK75 / SK60 properties. Graph provided by Mooring Solutions GmbH	38
Figure 19 - Attachment setups. Top side strings are directed to the motors and red strings are attached to a constant weight.	39
Figure 20 Gondola-Prototype 2 on the left and prototype 3 on the right	39
Figure 21 - Render of gondola prototype 4	40
Figure 22 - Gondola Prototypes 3 and 4 running a 500 liftings test with an Arduino UNO	40
Figure 23 Review of gondola prototype 4 with bearings and pen on intersection points drawing test patterns	41
Figure 24 - Render of the final design of the gondola	42
Figure 25 - Interface of Fusion360 at toolpath simulation before machining the plate of the company for the robot.	42
Figure 26 – Test setup with NEMA 17 stepper motor, optical sensor and threaded rod.	43
Figure 27 - Sliding mechanism driven with a NEMA 17 stepper motor with a string wrapped around a flat aluminium cylinder and optical sensor for identify home position.	44
Figure 28 Encoder used for measure the displacement of the string while motor setups where tested	44
Figure 29 - Render of the final motors setup	45
Figure 30 - Details of the motor setup, on the left hand side there is the reduction gear and on the right hand side the sliding mechanism that activate the optical sensor and the aluminium spool where the string is wrapped.	45
Figure 31 - Render of the final version of paper roll holder	46
Figure 32 - Paper roll mounting, open position at left hand side and lock position at right hand side.	47
Figure 33 - Paper roll assembly in At-Bristol at installation day.	47
Figure 34 -Delrin tension holder with adjustable friction	49
Figure 35 - Delrin enclosure for the pulleys used to redirect the string at the top part of the robot	50
Figure 36 - Gondola at charging position	51
Figure 37 - Render of the calibration tool for set the length of the pen	52
Figure 38 - Robot at installation day, BeagleBoneBlack and BBBB PBC on the left hand side, 24V PSU on the middle and 12V PSU on the right hand side.	53

Figure 39 - Picture of BeagleBoneBlack by Gareth Halfacree (22)	54
Figure 40 - Pins used at BeagleBone for the robot and functions.	54
Figure 41 - BeagleBone and breadboard configured for the V-Plotter.	55
Figure 42 - Front and rear view of the bbbbbb PCB.	56
Figure 43 - Final version of the BBBBBB PCB and BeagleBoneBlack	57
Figure 44 – Prototype and final Gondola PCBs	57
Figure 45 -Array of 4 Big 7 segment displays installed on the final robot.	58
Figure 46 - Wireless Rx Voltage/Charge Current vs Spacin of LTC4120 from datasheet specifications	59
Figure 47 - PCB inductive charge receiver before tests	60
Figure 48 - PCB inductive charge transmitter after soldering with IC oven	60
Figure 49 - Budget distribution by category	75
Figure 50 - Proposed budget for second unit of the robot	75

INDEX OF TABLES

Table 1 - Maximum position errors for 2400x1800 mm config. Data provided by Maximum_position_errors.m	26
Table 2 - Minimum requirements LinuxCNC (7) & Machinekit (8)	27
Table 3 - Budget components by date	74
Table 4 - Financial summary	74

1. Introduction and objective



Figure 1 - Energy MegaDrawz, At-Bristol museum September 2016.

At-Bristol is one of the most important interactive science centres in the UK. Their purpose is make science accessible to all and encourage a sense of adventure in their visitors. Annually, over 300,000 people are launched on a journey of discovery through engagement with their programs, including over 60,000 school children.

This project called Energy MegaDrawz born over the scope of the Robot Encounters. This event is designed as an evolving series of programming that explores questions and ideas about the second machine age, offering direct experiences between people, robots and researchers, enabling informed opinion and discussion, without dodging the emotions that arise around our robot infused futures.

Energy MegaDrawz represent a new permanent installation aiming for an eye catching 3m wide drawing robot that continuously draws a visualisation of energy generated by the museum's roof top solar array.

This project has been developed in collaboration with Matt Venn engineering, Bristol's green capital award in 2013, and with the Sponsorship of the leader in industrial components distribution RS.

2. Description

This section covers the elements that define the situation of the project and define the roadmap for the parts involved.

State of art helps to understand the situation of the technology, challenges, innovation. The customer specifications after the review of Matt Venn Engineering are consolidated in the project agreement.

Technical specifications define the main characteristics and development rules for meet the requirements of client and sponsor through the engineering team.

2.a State of the art

A V-plotter is the usual name for the vertical drawing robots driven by two strings, belts or chains, but other usual names are polargraph or bipod due to the characteristic shape and nature of the drawings.

The first bipod from which we have notice was built in 1998 with technic LEGO for the SIGGRAPH show, but the design details does not transcend. For most of the people Hektor (1) was the first real v-plotter, it was built by Jürg Lehni & Uli Franke in 2002 and catch the attention of artists and engineers. It uses a spray for draw and was driven by a script inside Adobe illustrator.

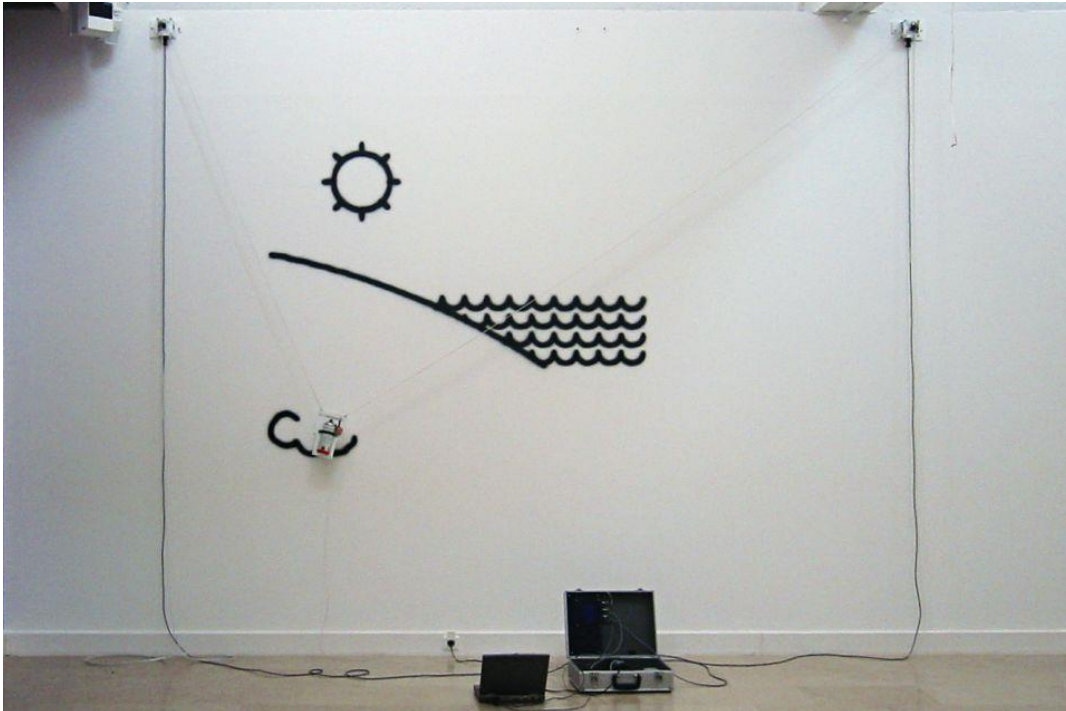


Figure 2 – Hektor Draws a Landscape, Righthd by Alex Rich & Jürg Lehni, 2003

In 2010 Harvey Moon, student at Art Institute of Chicago, raise 3,000 € at Kickstarter for finish the project, document it and provide open-source code, allowing other users to build their own machines (2). At the same time Sandy Noble another of the pioneers of this machines, was the first that use the name of polargraph (3). The name was used for describe their specific model of plotter but it become a usual name for this machines. The polargraph of Noble is an open source project with an active community of users that provide feedback and improvements.

V-plotters has been built in different ways. One of the elements that allow more variety of this devices is the actuator, also called gondola or penholder. This part hangs form the cords and can have lifting mechanism, spray actuator, different weight distribution, special string attachments and other kind of tricks for get the best results from the specifications and resources that each designer have. Another characteristic mechanical element of each machine is the use of strings, belts or ball chains.

The control system is a key point, from the early designs where patters where hardcoded, to the actual models were the designs are autogenerated. Hektor was intended as peripheral of a

computer and it get good results, but actually there are interesting versions running on Arduino and single-board computers.

2.b Customers specification

The customer, At-Bristol, located at Anchor Road, Harbourside, Bristol ,BS1 5DB. Specify the following requirements:

Delivery and installation:

- The robot shall be installed before 21 July.
- Foyer is open to public seven days a week and installation must be performed out of normal operating hours, and trying to avoid to use bulky equipment as scaffolding towers if it is possible.
- The museum is interested in use as drawing surface the black wall of the foyer and use chalk or removable ink.
- Warranty/technical support agreement for bigger issues and advanced maintenance shall be provided.

Design requirements:

- The robot will be a permanent installation that will be placed at the foyer of the museum, for that reason the machine shall be reliable and robust.
- The drawing area of the robot shall be around 200 cm². And the maximum dimensions of the robot shall be 300 cm wide by 200 cm height.
- Maintenance task as clean the wall and change the batteries shall be performed as maximum once a week.
- The drawings shall be done every two minutes.

Safety requirements:

- Even if is not allow to touch the machine the project shall consider the consequences for the machine and guaranty the safety of the public. Safety test will be defined and executed by the safety team of the museum after installation. This test can include:
 - Safe visitor interaction
 - Operation
 - At-Bristol Design risk assessment
 - At-Bristol In-situ risk assessment

At-Bristol resources available:

- Local maintenance team that can be trained for perform common trouble shooting and simple fixes.
- Graphical representation of the data can be performed by in-house graphic team if a specific input file type is provided.

2.c Project agreement

Parts of the agreement:

This agreement concern two parts, the purchaser represented by At-Bristol with physical address Anchor Road, Harbourside, Bristol ,BS1 5DB. And the supplier Matt Venn based at Carrer de la República Argentina, 22, 46021 Valencia.

Hereinafter together referred to as the Parties

1.Objective and scope

- At-Bristol will purchase a customized V-plotter from now on called Energy Drawing Robot or Energy MegaDrawz from Matt Venn.
- Chris Dunford, Sustainability Engagement manager will act as project manager for At-Bristol and Andy Cocco, Senior Design Engineer will have responsibility for signing off the final design of the exhibit.
- This agreement regulates the rights and obligations between parties, regarding design, purchasing and installing the exhibit in At-Bristol.

2.Scope of Delivery

- The Agreement includes the following Exhibits being supplied to At-Bristol
Exhibit name: X0339-1 Energy robot

3.Subcontractors

- The Supplier is entitled to use subcontractors been the supplier the responsible to fulfil the conditions established in this agreement.

4.Price and payments

- The amount destined for the project Energy MegaDrawz with reference name X0339-1 Energy robot is GBP £3.600 included development, delivery and installation, been this the total amount provided by the museum.
- Cost of on-going consumables is covered by the purchaser.
- The payment will be distributed as follow:
 - 50% on signing of this Agreement.
 - 40% after shipment
 - 10% when the acceptance form is signed by both Parties.
- Payment shall be made 60 days after receipt of invoice.
- The invoice shall be marked: PO#110284 X0339-1 Energy Robot and sent by email to: chris.dunford@at-bristol.org.uk & finance@at-bristol.org.uk

5.Packagin and liability

- The supplier is responsible of the damages produced on the Exhibit X0339-1 Energy robot before the delivery, installation and testing. This section does not exclude the supplier from the subjection of other sections.

6.Procedure

- The Supplier is obligated to comply with the following:
 - Comply the essential features of the final design specified at the end of this document.
 - Design, develop and produce the Exhibits, incl. documentation and O&M manual.
 - Report progress and quality to At-Bristol
 - Pack the Exhibit and prepare it for transport.
 - Transport the Exhibits to At-Bristol's building
 - During the warranty period (12 months), repair any defects to the Exhibit reported.
 - Provide training in the maintenance and operation of the Exhibits to At-Bristol's employees.

7.Reporting of faults and defects

- Any faults and/or defects of the Exhibits shall be reported to the Supplier within a reasonable time after the fault and/or defect has been discovered by At-Bristol.
- The Supplier shall repair the faults and/or defects within 10 days after the faults and/or defects have been reported.
- Modifications or maintenance procedures not specify by the supplier can avoid the warranty.

8.Customer rights

- On signing of this Agreement, At-Bristol shall have all rights to the maintenance and further development of the Exhibit.

9.Intellectual property rights

- At-Bristol have all rights and ownership to trademarks, copyright, design, graphics and expression of the 'Robot Encounters' programme.
- The Supplier shall guarantee that the Exhibit is brand new and manufactured for the purposes of At-Bristol. The Exhibit shall not be encumbered with any third-party rights, including any rights arising from copyright and related rights.
- - The Supplier will provide At-Bristol with a non-exclusive license, unlimited in time, for the Exhibit. The Supplier is entitled to use the Exhibit in other marketing.

10.Delivery

- The Supplier shall deliver the Exhibits to At-Bristol, Anchor Road, Bristol.
- All the cost derived from transportation shall be included in the contract costs.
- Supplier shall provide the exhibit following the deadlines established in section 11. The supplier will cover any financial loss that can derivate from the delays in the project. The maximum compensation that can be required by the purchaser will be of 0,1% of the price established for the project in section 4, for each day of delay.

11. Time schedule

- The supplier shall accomplish the goals stipulated in the following schedule:
- Order shall be confirmed before beginning of May 2016.
- The specifications must be fixed by the parts before May 15th 2016.
- Between 15th of May and 15th of July 2016 the progress of the project must be communicated to At-Bristol twice a month.
- The exhibits must be delivered, installed and tested before 21st of July 2016

12. Assisting workers

- The purchaser will provide personnel to oversee the power installation, perform the safety test and other tasks explicitly required in this agreement.

13. Safety regulations

- The Working Environment Act and Regulations for Systematic Health, Safety and Environmental Activities in Businesses (HSE Regulations) applies to the location of the Exhibit. The parts shall ensure that all activity complies with these regulations.

14. Transfer of possession

- The possession of the Exhibit shall be transferred with full functionalities to At-Bristol on the 22nd July 2016 in other case section 7 and 10 shall be applied and the procedure of transference will be delayed.

15. Duration

- The exhibit shall be designed to have a lifespan between 3 and 5 years and the warranty period shall be equal to 12 months from transfer of the possession.

Final specifications

The parts have agreed the specifications of the exhibit X0339-1 Energy robot also referred as Energy MegaDrawz as follows:

Delivery and installation:

- The robot shall be installed before 21 July.
- Foyer is open to public seven days a week and installation must be performed out of normal operating hours, and bulky equipment such as scaffolding towers are not allowed.
- The robot will be installed at the foyer, shall draw over a carpet of one metre width.
- The supplier does not guarantee full functionalities of the robot if a wired LAN connection is not provided, if this applies lack of functionalities will not be considered as breach of contract.

- The installation of a standard BS 546 socket connected to electric circuit and placed behind the robot cabinet will be provided by the purchaser.
- The drawing robot shall be floor standing but shall be securely attached to the wall. The supplier is allowed drill holes to the foyer wall when necessary.
- Warranty/technical support will be provided for the first twelve months after possession is transferred.
- Maintenance will be provided by the purchaser. The supplier will provide documentation for perform this tasks and at least one hour of training.

Design requirements:

- The robot will be a permanent installation that will be placed at the foyer of the museum, for that reason the machine shall be reliable and robust.
- The drawing area of the robot shall be equal or greater to 150cm². And the maximum dimensions of the robot shall be 300 cm wide by 200 cm height.
- Maintenance task as clean the wall and change the batteries shall be performed as maximum once a week.
- The drawings shall be done every two minutes.
- The public front part of the cabinet shall be made of transparent acrylic, and the different electric and mechanical parts that form the system shall be visible.

Safety requirements:

- Even if is not allow to touch the machine the project shall consider the consequences for the machine and guaranty the safety of the public. Safety test will be defined and executed by the safety team of the museum after installation. This test can include:
 - Safe visitor interaction
 - Operation
 - At-Bristol Design risk assessment
 - At-Bristol In-situ risk assessment
- The power supply shall be external, and a 30 mA Residual-current device shall be installed at the AC input by At-Bristol before final approval.
- The exhibit shall be attached to the wall with M8 screws.
- Aluminium extrusion for the frame shall be made with 40x40mm aluminium extrusion profiles or larger and connected to earth. This assembly shall be done with quick connectors.
- The weakest point of the structure shall be able to hold a load of 750N and the height of the cabinet shall be greater than 750 mm.
- Slots must be designed for avoid fingers to get stuck, this will be tested by the At-Bristol museum.

2.d Sponsorship agreement



Figure 3 - RS Components

This project caught the attention of RS Components in United Kingdom. The sponsorship agreement was established verbally over successive meetings. The conditions were established as follows:

RS agrees to provide 1500 pounds in material from their UK store and free shipping.

Matt Venn engineering will record a series of videos documenting the development process, these videos will contain the RS logo.

The head engineer of the project will write a blog post for RS that will be published after the presentation of the project.

The logo of RS will appear in the description board of the robot while it is installed in the Bristol Museum.

RS can record the robot and use this material without limitation.

2.e Technical specifications

For this project we establish some internal specifications in addition to the agreement specifications for reach the goals of the customer and guide the engineering team on the development.

General Specifications

- Final parts can be manufactured with 10mm acrylic, 6mm delrin, 3 and 5 mm aluminium, ABS or PLA 3D printed plastic.
- Drawing speed shall be configurable and greater than 10mm/s.
- Prototype of the robot shall be assembled locally for development and test.
- Mechanical design shall be made with fusion 360 or Solidworks.
- Pieces for laser cutter shall be exported to DXF and imported to CorelDRAW.
- Pieces machined with CNC milling machine shall be exported to Fusion360. A qualified operator will generate the best tool path and machine the piece.

Management

- Task will be scheduled in the Trello platform.
- On Mondays morning the schedule of the week will be discussed.
- State of the project will be reported to AT-Bristol and RS Components twice a month.

PCBs

- Two sided PCBs with all the possible 1206 SMD components considering technical limitations.
- Design with Ki-Cad or Eagle.
- Visual and electrical inspections shall be performed before connexion with power supply or other systems.
- M3 mounting holes shall be included.
- Include extra pin out if is possible and alternative functions to reuse the hardware for other projects.

Code

- Split the code in functions for improve teamwork, maintenance and reusability.
- When hardware is interfaced, create test files for check all the functions independently.
- Use GitHub repositories for log the modifications.

Frame

- Renders of the design shall be provided to AT-Bristol before order the components.
- The use of acrylic shall be reduced as much as possible.
- Use standard connectors and profiles with guides for screw in.

Gondola

- Gondola PCB shall be smaller than 100mmx40mm.
- Battery shall allow lift the actuator 303000 before discharge
- Pen shall be easy to adjust without modifications of the actuator by maintenance team.
- Gondola shall be tested for 200 hours before deliver.

Motors

- The motors can be servos or steppers.
- Simulation of the system shall be performed for determine their characteristics before buy the components.

Paper mount

- Design shall be optimized for reduce the amount of delrin necessary to build it. Acrylic cannot be used for structural parts.
- Design shall allow to fit rolls between 30 and 90 mm of diameter.
- -Replace the paper shall be a one-man task.

String guides

- Shall be designed thinking about appearance and reliability
- Component shall be tested for more than 120 hours.

Charging

- Determine best option between inductive charge or contact charge.
- Test for 200 hours

3. Project development

This project has been developed as result of the internship at Matt Venn Engineering. Work as part of a small team has allow me to participate in almost all the parts of the process. This memory covers the main aspects of design in a general way for keep the cohesion and pays more attention to the sections where more work load was assigned to me.

This section has four parts in function of their main subject.

3.a Simulations

In modern engineering simulation has become a crucial part of the design process because allow to extract non obvious information of a system through a virtual model. Once the environment is developed is possible to change the parameters for get information of situations that are unknown, detect relations and effect of the variables, generate useful information for take design decisions and detect problems in early stages. For that reasons our first task was design a set of functions in Matlab for verify the behaviour of the tensions, and the effect of different sources of errors.

The code has been designed to be modular, splitting the problem in small pieces and keep it reusable. This section will explain briefly the path followed in the development and the conclusions obtained, but for further details do not hesitate to review Appendix II, Matlab Code.

I. Tensions

The objectives of these simulations were generate a dynamic model for determine the maximum and minimum tensions and how they are distributed. One of the basic functions was *tcalc*, that determine the tensions in each wire for a gondola with two strings. This calculation is a straightforward statics problem that has as input the position of the anchor points, hanging weight and the actual position of the actuator. The weight of cord is omitted because have not a significant effect at the given configuration but shall be considered for larger setups with heavier strings.

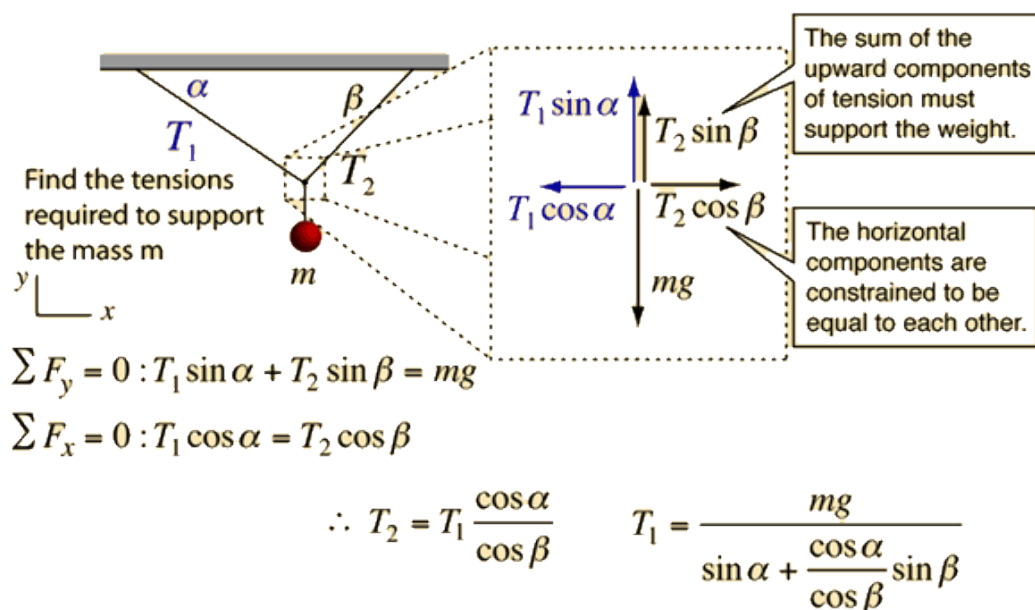


Figure 4 - Statics problem that define the position of the gondola (4).

Once that function was tested the next step was code the function *tmatr* that allow to introduce the weight of the gondola or actuator and the working area. The anchor points are intended to be in the top corners and samples are taken every unit increase in X and Y axes. That function returns one matrix for each string with the tension values.

Following the rules of modular function design, we decided of not include the graphical representation of the values inside the *tmatr* function for allow us to reuse this function. The

representation of the values is made with tpmmap, and the output format is a heat map, that allow to zoom in the desired areas and view the exact values of the tension.

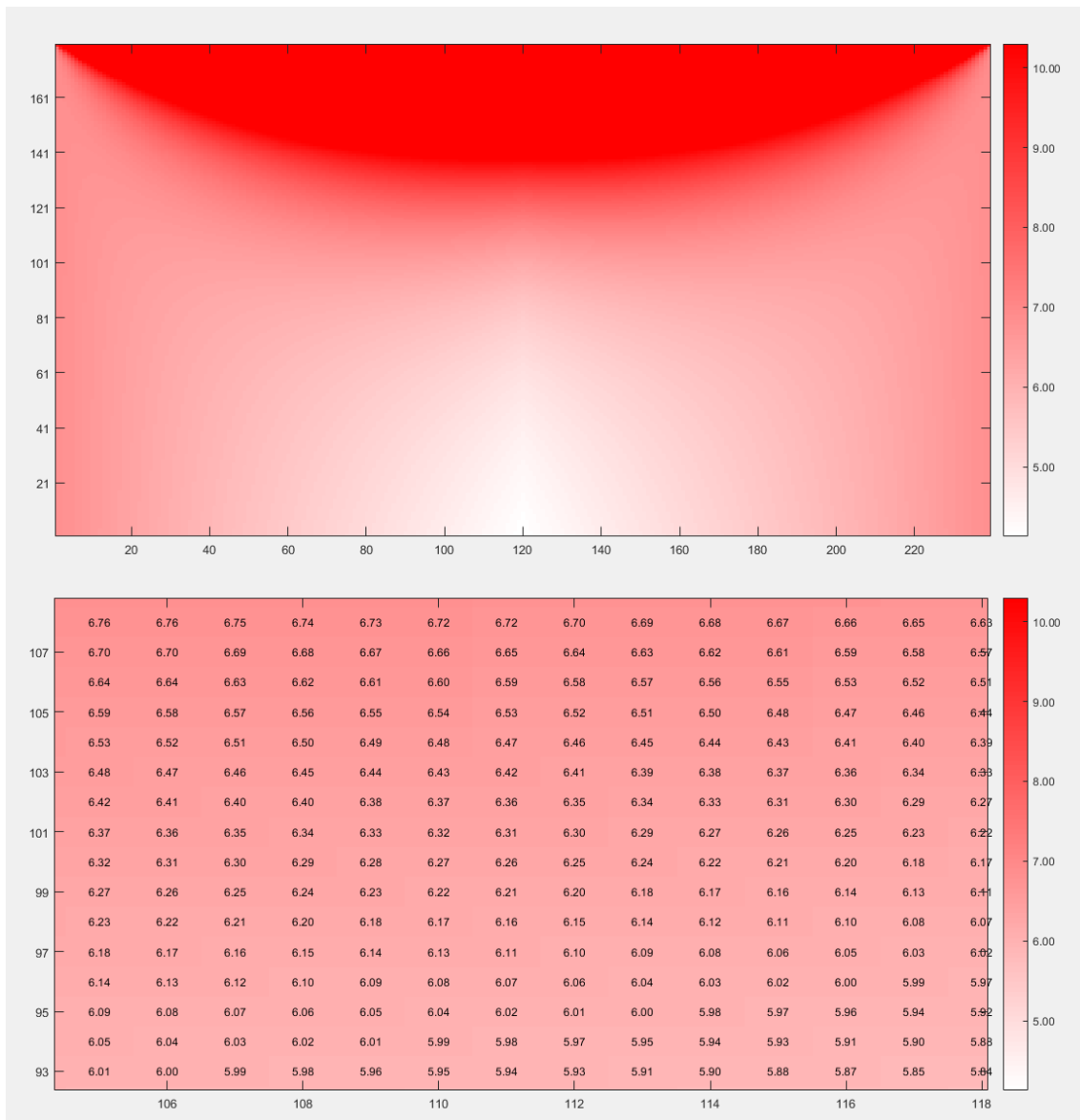


Figure 5 – Output of function `quadtpmap(500,200,240,180)` on top side and zoom on the bottom side.

From the graphical representation we can see that the maximum tensions drop significantly if we avoid the top part of the working area. For see how important the variations are we implemented the function `valuesarea` that with the size of the area, weight of the gondola and percentage of the working space determine the maximum, minimum and average tensions. With that procedure implemented we can get the values for different configurations and log them in an excel file where we will compute the final required torque considering other parameters as spool diameter, micro stepping and use of gears. Complete details are provided in Appendix III, Resume of simulations and design parameters.

For the four wires configuration we reused the bipod functions. In four and three wires configuration the extra weight is only handled by one string the modulus of tensions remains

constant and the final effect is always and increase on the average tension. Based on this *quadtcalc* and *tcalc* produce the same output if the input parameter *wstab* is zero.

From the results of the four wires configuration we decided that the possible advantages of the three and four string configurations where not related with the maximum or average tension at some point, and was more related with the centre of gravity and the moment.

II. Errors

In this section we were more interested with the motion of the robot, variation of the homing position and effect of the microstepping. A new set of functions was designed for simulate the movement in X and Y. Function *posangles* provide the direct kinematics given the lengths of the strings and distance between attachment points. The inverse kinematics are computed with *inverseposition* that given the attachment points and position of the actuator returns the required configuration. *param1calc* is used for compute several variables that are used frequently or are interesting for the mechanical design.

Once the behaviour of the robot can be obtained through the previous functions next step was generate a function for introduce and represent the errors. We focus on the quantization error produced by the number of steps of the motor and independent homing error in each axis. The function *possim1* is used for this purpose.

With the previous functions we are able to see the quantization errors and homing errors at a given point but for study how this errors are translated a long of the working area is required to test many points, for that reason we decide to draw several test patterns in a computer graphics tool and load it into Matlab for compute the values for each point. This function is called *importjpg* and make use of the image processing toolbox of Matlab for load the pictures, determine the base of the robot and generate a graphical output.

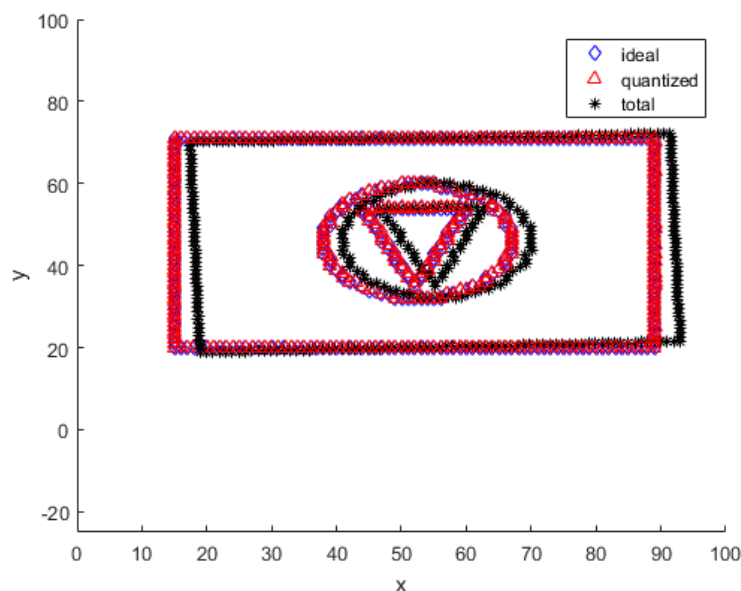


Figure 6 - Output of *importjpg* for test pattern *square_circle_triangle_100x100.png* for homing x 2 units homing y-2 units error.

In the previous figure can be seen the effects of different sign error in the homing procedure. The different combinations between sign errors produce characteristic distortions that can be used in calibration stages for troubleshooting.

The last function that we made was called Maximum_position_errors and as its name indicate provide the maximum position errors of the whole drawing surface as function of the spool diameter, working area, homing errors and robot dimensions.

Maximum errors for 2400x1800 mm config.						
Spool diameter (mm)	55					
Area (%)	0,75					
Homerror1	0,80	0,80	0,60	0,60	0,20	0,20
Homerror2	-0,80	0,80	-0,60	0,60	-0,20	0,20
Max x error (mm)	1,25	0,36	0,94	0,27	0,32	0,02
Max y error (mm)	0,33	2,30	0,25	1,73	0,08	0,59
Max distance error (mm)	1,29	2,33	0,97	1,75	0,33	0,59

Table 1 - Maximum position errors for 2400x1800 mm config. Data provided by Maximum_position_errors.m

In addition, this function provides also the maximum error due to quantization or number of steps. The data of several simulations with different amount of steps is plotted at table *Type of errors & maximum values for 2400x1800 mm* in Appendix III, Resume of simulations and design parameters. That values help us to identify that increase the number of steps per revolution for increase resolution will produce small improvement if the accuracy at homing is not improved.

3.b Software

In a project there is always more than one way to reach your goal, hardware decision use to be more constrained, due to materials, machinery or cost but software is more flexible. This capability to face a problem with several strategies shall be an advantage. Always is important keep in mind the time as a resource. With that in mind we cover this section reusing and adapting code of previous projects or open source repositories.

I. LinuxCNC/Machinekit configuration

There are several ways to control a V-plotter. People have done it with regular computers or embedded systems, examples of that are the project Hektor or Polargraph.uk respectively. We know about an open source projects that were used for control CNC machines and lathes with very good results and we thought that if we modify the kinematics files Machinekit, or LinuxCNC (5) will be capable to control our robot, allowing to include new functionalities easily.

Machinekit (6) is a fork of LinuxCNC focus on embedded devices support, It allows to run real-time extensions, supports stepper and servo drivers, G-code input and a graphical user interface that together with the hardware abstraction layer simplify configuration procedure and debugging. The use of this software instead of build a dedicated system allow to reduce the hardware and software development significantly.

We look for the minimum specifications for run them, and the feedback of the users.

	LinuxCNC	Machinekit
Processor	700MHz	1GHz
RAM	384 MB	1GB
Hard disk space	8 GB	8GB
Graphics card resolution	1024x768	1024x768

Table 2 - Minimum requirements LinuxCNC (7) & Machinekit (8)

Specifications of LinuxCNC are lower than for Machinekit, but there is a modified version of Machinekit for BeagleBoneBlack that can overcome the problem of RAM usage. In addition, most of the RAM is used for the GUI, but we will run a python interface.

```
top - 14:40:02 up 4 days, 20:29, 1 user, load average: 0.90, 0.87, 0.83
Tasks: 127 total, 5 running, 121 sleeping, 0 stopped, 1 zombie
%Cpu(s): 20.3 us, 14.3 sy, 0.0 ni, 65.0 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
KiB Mem: 507676 total, 463808 used, 43868 free, 81224 buffers
KiB Swap: 0 total, 0 used, 0 free, 161236 cached

  PID USER      PR  NI  VIRT  RES  SHR  S   %CPU  %MEM    TIME+  COMMAND
 7514 root        20   0 61812  58m 10m  S   0.0 11.7   1:44.57 rtapi:0
 667  root        20   0 35004  27m 2936  S   0.0  5.5   0:23.12 jekyll
 666  root        20   0 85140  25m 4676  S   0.0  5.1   0:06.73 node
 7686 machinek  20   0 21188  19m 14m  S   3.0  3.9 134:32.06 milltask
 1533 machinek  20   0 29184  16m 8268  S   0.0  3.2   0:01.36 wicd-client
 7672 machinek  20   0 13572  12m 9800  S   2.3  2.5 101:53.23 xbee.py
 1541 root        20   0 16036  9584 1372  S   0.0  1.9   7:17.13 supervisord
 1526 machinek  20   0 89504  9364 7076  S   0.0  1.8   0:18.28 pcmmanfm
```

Figure 7 - RAM memory usage of BeagleBoneBlack running Machinekit with python interface instead of Axis GUI

We can see from previous figure that the four heaviest task when Machinekit run with a python interface use around 25% of the total memory, that is equivalent to only 128MB.

From the webpage of elinux (9) can be obtained the ISOs for run Machinekit in our BeagleBone, once that is done can be burn in the microSD or eMMC. In our case we choose the microSD installation. The full setup process is well documented at blog of the project (10).

We choose the last version of Machinekit available on June. It was a Debian 7.1 distribution that use a Xenomai (11) *real-time kernel for make the best use of our hardware. Details of the board are explained at section BeagleBoneBlack on part 3.d.*

Once the system is loaded in the SD card we can login with the user "Machinekit" and password "makinekit" as well. You can login through SSH, USB and serial COM. The first that we made was provide WIFI connection with a USB dongle, this allow to access from several computers without physical connections with the board. We activate the X11 forwarding protocol that allow remote access to the BeagleBone display.

The SSH client used for Windows was Putty (12) and the x screen server was Xming (13). File management can be done through terminal but can be easier for some users use winSCP (14). For interface this applications with the BeagleBone is necessary to introduce the host name, by default beaglebone.local, or the IP address and the access port, by default port 22. In the configuration of putty x11 Forwarding shall be activated for run the GUI of the Machinekit. When the development environment is setup we can continue with the customization of the Machinekit. But we recommend to check that the clock and data are right and upload LinuxCNC to the last available version before continue.

Machinekit and LinuxCNC configuration is mainly defined in two files. In our case named bipod.ini and bipod.hal, source code of this and other configuration files referred in this section can be found in Appendix IV, Machinekit Code modifications. On the one hand, ".HAL" files configure the hardware abstraction layer. Here is where we choose which kinematics file shall be used, trajectory planner, threads and their functions, motors configuration and I/O devices with their respective pin location. On the other hand, ".ini" files define the software configuration as can be the type of display, debug level, home position, acceleration and speed limit, tool table, etcetera. Detailed notes of each file and configuration process can be found at <http://linuxcnc.org/docs/2.7/html/>.

For transform LinuxCNC from a cartesian CNC the default kinematics files shall be replaced. We used the file bipodkins.c where the *kinematicsForward* and *kinematicsInverse* functions have been modify to our setup. This functions are similar to *positionxy.m* and *inverseposition.m* used in the Matlab simulations and solve the same kinematic problem.

For the test we use the 'Axis', the standard GUI interface. It allows to home the axis, load and visualise trajectories, check the hardware signals, drive manually the machine, and more options that allow to verify that the machine will work as expect. That interface can be also used for use the machine as regular plotter where the files are loaded and positioned by the user.

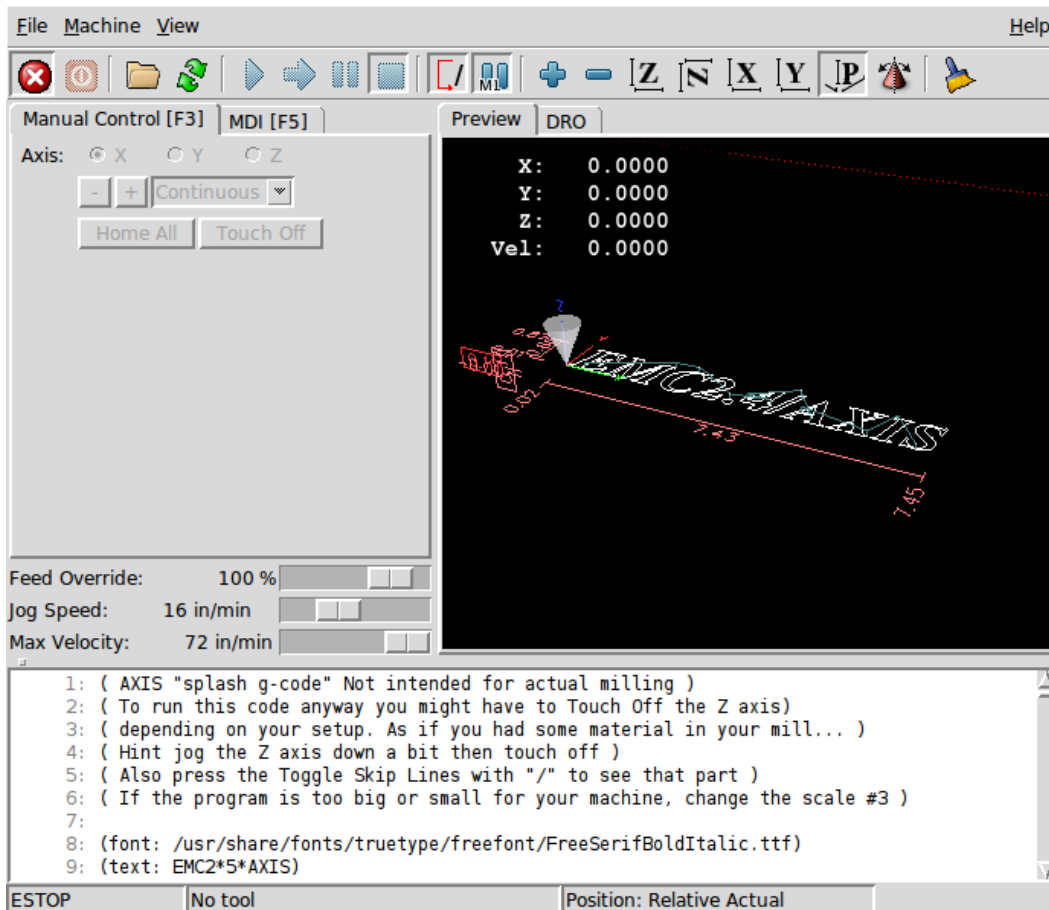


Figure 8 - Axis GUI Linux CNC

For add the autonomous function we add a python interface called *bipod.py*. That file allows to use the autonomous behaviour. This file integrates communications between parts as the gondola signal, interval between drawings, charging sequence, etcetera...

II. Communications

We can identify three communications systems in this robot. The first one is the communication between the BeagleBone and the developer, that as has been explained before can be done in several ways and is already implemented in the system. The second part is the communication between the gondola and the BeagleBone and the third one is between the server that receive the information obtained from the solar panels and the robot.

The communication with the gondola has been made with a wireless connection using two xbee2. From the gondola an ATmega328p provide the signal of a capacitive sensor for detect if a visitor has interfered with the robot, the battery voltage. And a checksum. At the other side Linux does the checksum and if both are the same register the information for modify its behaviour if somebody touch the gondola or the battery go to low. At the same time a package is send back to the gondola for provide the position of the servo that drives the Z axis.

The robot has communication with the data logger of the solar panels, a Sunny WebBox (15), through local network. The data of energy generation and status of the machine is send by Linux

to the CursiveData (16) server every two minutes. Energy data is transform to a graphical representation, exported to gcode and send it back to the specific folder that Machinekit.

All the values are recorded in the server and displayed in a dynamic webpage powered by freeboard (17) . This online dashboard allows to monitor the input data, actual drawing, energy generation and other advanced parameters of the robot, position, errors, exceptions, temperature, etc...



Figure 9 - Robot dashboard on Cursivedata server (18).

III. Pattern generation

LinuxCNC is designed for perform machining tasks for that reason the input files shall be written in g-code with the extensions .nc, .ngc or .gcode. The axis GUI have tools for transform pictures in JPG or PNG to this format nonetheless better and more predictable results are obtained with native gcode files.

We have used several ways for generate the trajectories that we wanted to draw in each situation. General steps are design a drawing in vector format, export it to DXF and translate it to gcode, sometimes this process is not straight forward and require adjust the size, relocate the origin of coordinates and configure the program for export the gcode with the configuration of the target machine.

We were interested to test the accuracy and reliability of our system for that reason we designed some complex geometrical shapes. The geometry allows to easily recognize mistakes and

distortions of the image and the intersection over the same points help us to check that the paper will not break due to the friction and excess of ink, moreover large files help to identify if the system have blockages or if mechanicals problems appear. We use Matlab, and several scripts where a polar function change over iterations and the output is plotted. This pattern generation files and some of the output results can be found in the virtual resources under the folder Pattern designs\Matlab.

```

theta = linspace(0,0.001);
rho1 = theta/10;
polarplot(theta,rho1)

rho2 = theta/12;
hold on
%polarplot(theta,rho2)
theta = 0:0.01:2*pi;
rho = sin(2*theta).*cos(2*theta)-1+sin(theta);
for i=1:20
polarplot(theta,rho,'b')
polarplot(theta+2*i,rho,'b')
end
hold off

```

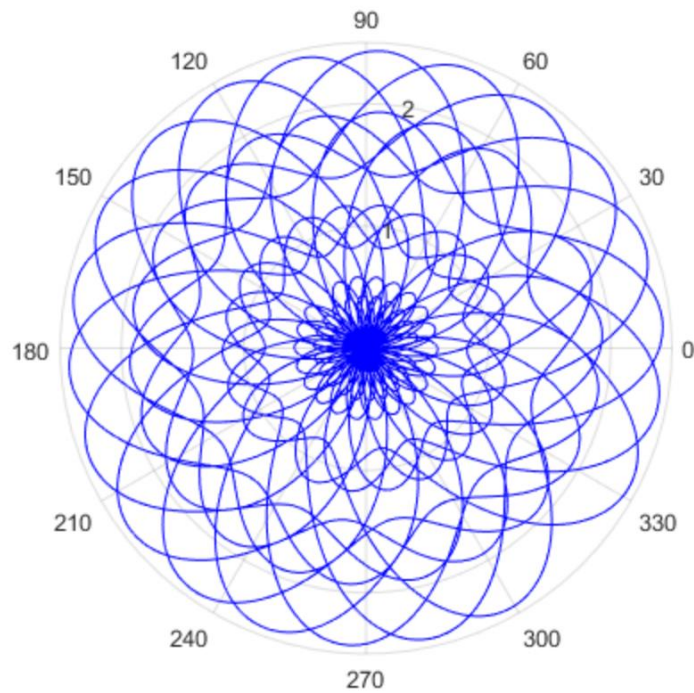


Figure 10 – Code and output of Matlab patter generation a.m.

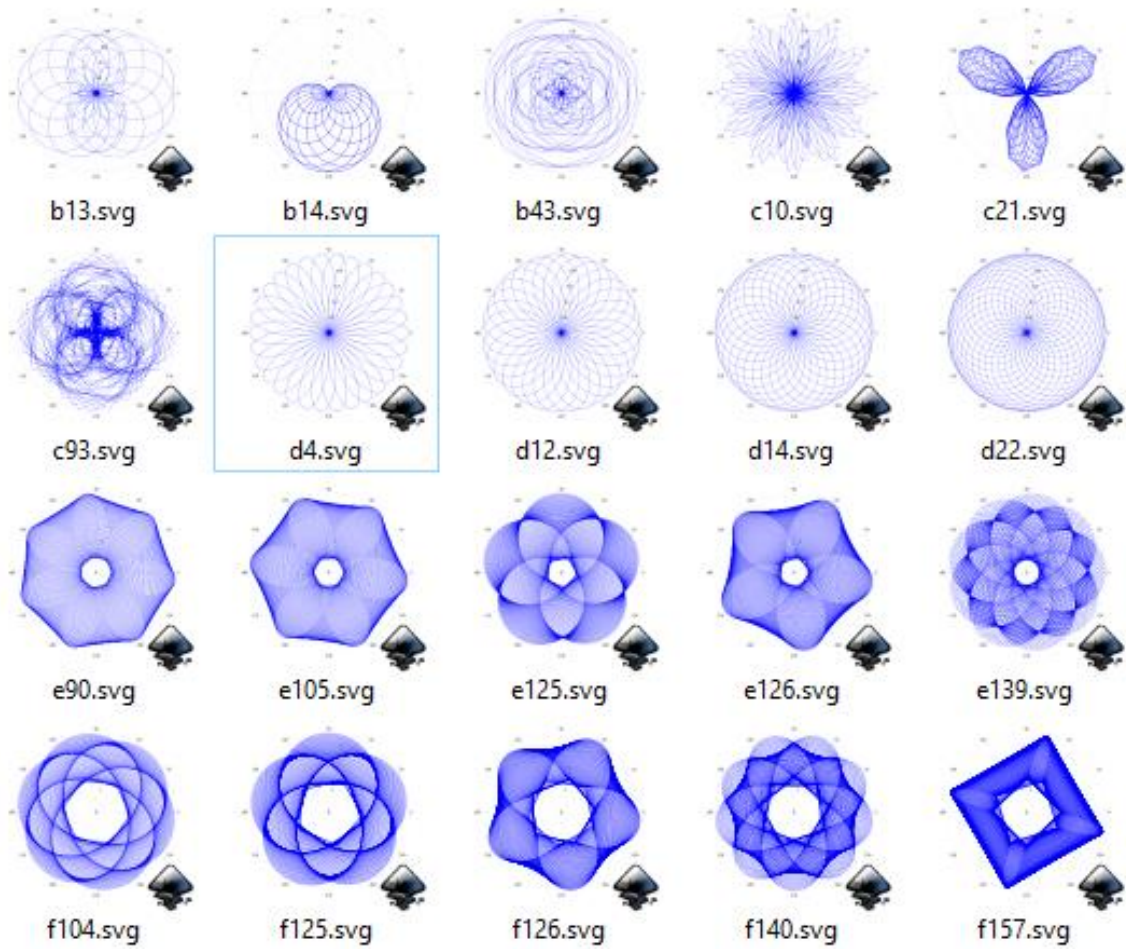


Figure 11 – Set 1 of patterns generated with Matlab

Once the files are saved in .SVG format they shall be scaled, reposition it according to the coordinates system that we are using in the robot and transformed to DXF. That has been made with the *open source* SVG graphics editor Inkscape (19). Once those tasks are performed the picture can be exported to gcode with several applications. In our case we use DXF2GCODE. Finally, this gcode can be loaded to the Machinekit. If the python interface is running the file shall be placed in the directory /tmp/gcodes, nevertheless if the axis GUI is enabled, we can search the file through any directory, preview the design, relocate it and start the drawing.

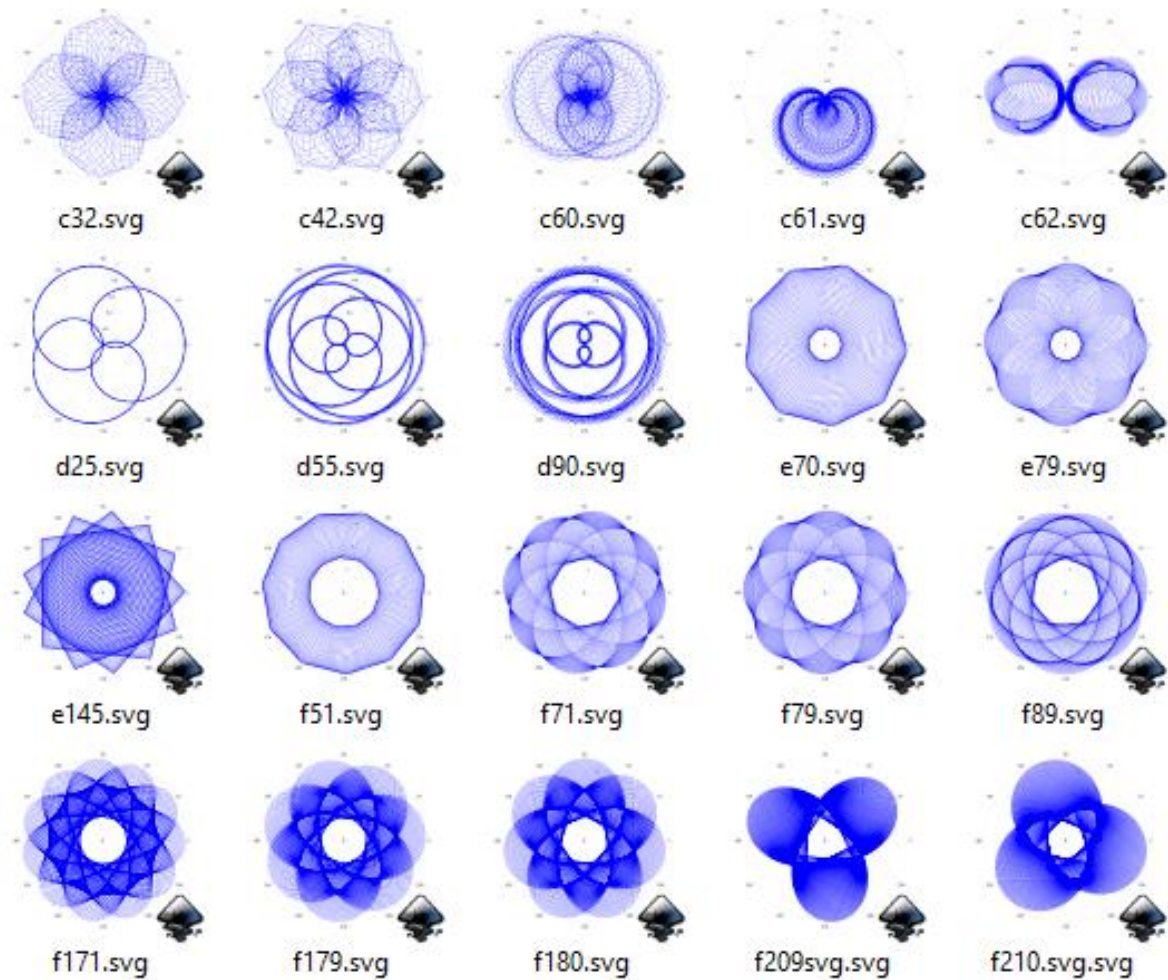


Figure 12 - Set 2 of patterns generated with Matlab

Another way that we use for design test patterns was use Solidworks. We design 2D drawings that where saved directly to DXF. Those patterns where thought for evaluate the lifting mechanism of the gondola, the accuracy for draw straight lines and circles and the capacity for manage large files and long printing times. Some of this designs produce problems for transform it to gcode due to the large amount of segments used for draw accurate splines. For that reason, is useful avoid Text and export the designs as polylines instead of splines.

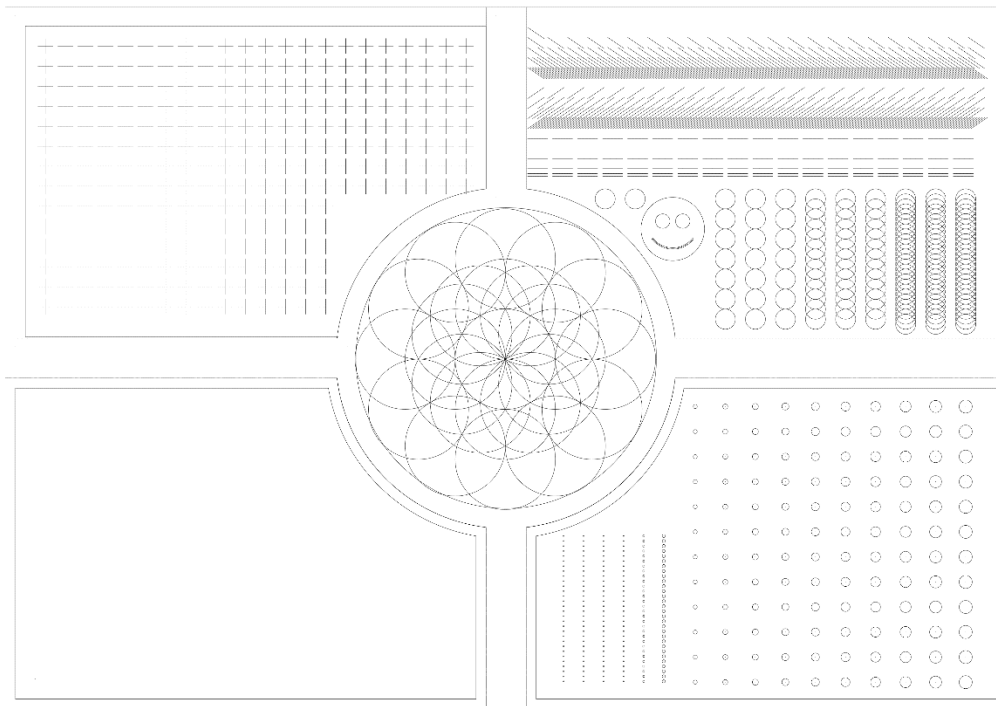


Figure 13 - testpat_cccc designed with Solidworks

Those patterns were useful for understanding the limitations of DXF2GCODE and spotting some communication errors with the gondola and the BeagleBone and fixing the movement in the Z axis.

As an experiment we developed a Matlab function that transforms a low-resolution grey-scale picture directly to gcode. The name of this function was [png2gcode_4levels_horizontalfill.m](#) and can be found in the virtual resources. Each pixel was transformed into a set of lines in the gcode. If the pixel is black, the maximum amount of lines is drawn, and if it is white, nothing is done.

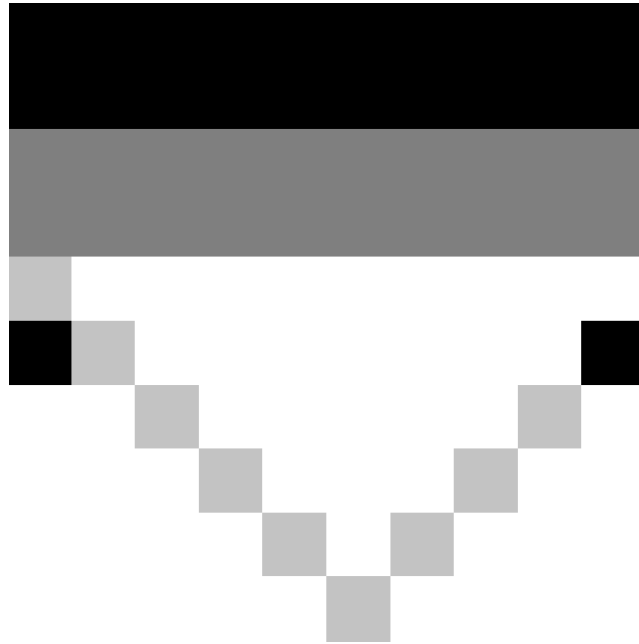


Figure 14 - Input image for png2gcode_4levels_horizontalfill.m

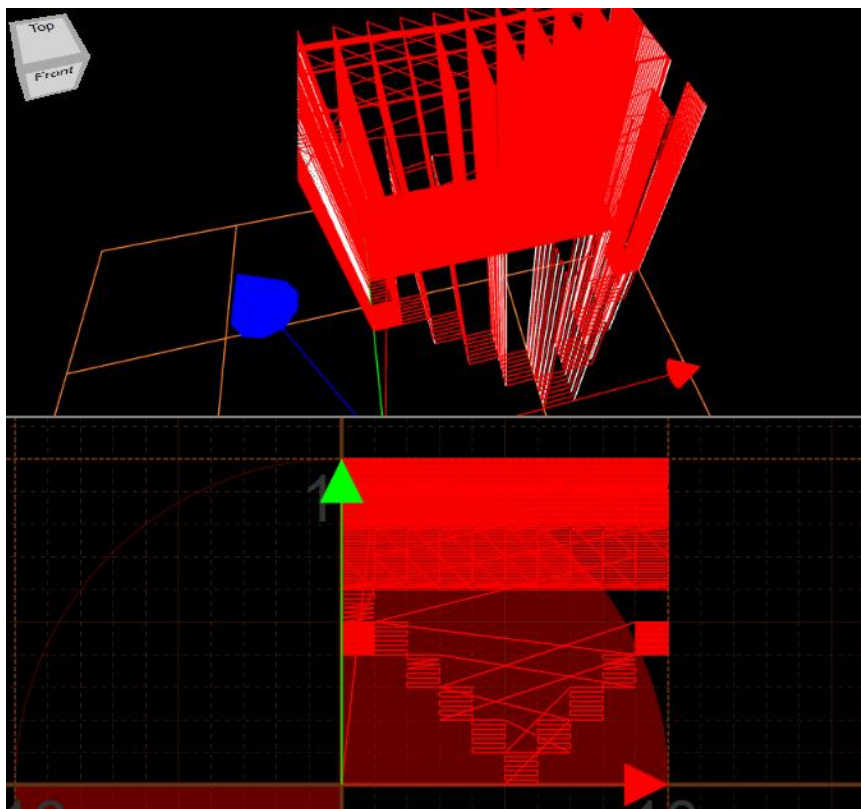


Figure 15 -Trajectory generated with png2gcode_4levels_horizontalfill.m using figure 6 as input and web gcode simulator nraynaud.github.io/webgcode/

Despite of the potential of this gcode generator we decided to stop the development for put more effort in other parts of the project and keep the source code for future projects.

The last way of generate the drawings is the option used in the usual operation mode of the V-plotter. We have a stream of data that is collected by the robot. That information is delivered to the server of CursiveData, the information is transformed in to a drawing, when more energy is

generated the shapes become larger and more intricate. The drawing is now exported to gcode and send it to the robot. Finally, the robot checks if there is an incoming file and runs the gcode. Every petal is generated with intervals of two minutes, and only the gcode corresponding to this petal is sent every time.

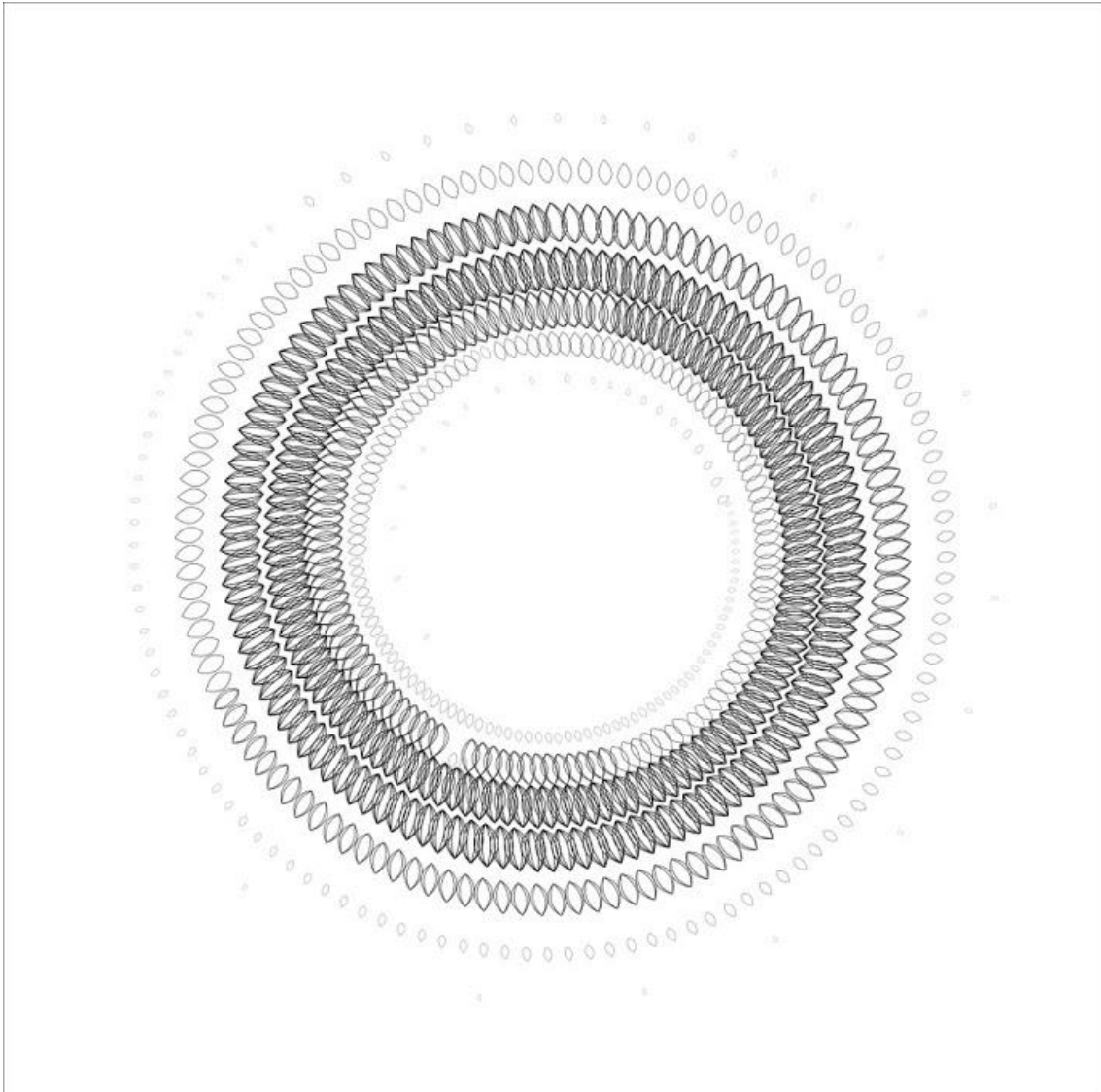


Figure 16 - Petals pattern generated by CursiveData sever from the solar panels energy generation of AT-Bristol museum from 22 to 28 of September.

3.c Mechanical design

These decisions have help to increase the performance of the robot and fulfil the safety specifications, furthermore the design required to be attractive and reliable and was bounded to the limitations of machinery, costs and time.

The prototypes described in this chapter have been designed for be able to manufacture them with a laser cutting machine, 3D printer and a lathe. At the same time the designs shall be doable with small changes on a CNC drilling machine for get a better finish.

I. Cabinet & frame

This is the largest structure of the project and was modified several times for meet the needs of supplier and provider. Cost, and appearance where fundamental but also the safety was a main topic specially with the anchor points.



Figure 17 - Render of the final design approved by both parts

The plans of this section are available on Appendix VI, Mechanical components. And are labelled with the initial letter F.

The cabinet has a structure of 40x40mm Rexroth aluminium profiles that allow easy assembly with T-nuts and quick connectors. The front part of the cabinet has two parts; the top part is covered with 3mm thickness transparent acrylic and contain the motors setup, 4 LED displays, the BeagleBoneBlack and circuit boards and the two power supplies.

The lower part is used for store the tools for maintenance as the length setter for the pen, the stick for replace the paper, grease, along with others. This part is covered with white veneered chipboard and secured with a key.

As was stated in the specifications the structure stands on the floor but was also secured by two M8 screws on the top part.

The paper is placed in the top side of the structure with a mechanism explained in subsection IV, Other elements, the paper is feed under an acrylic plate for keep it flat over the wood surface. After that it pass through a slot and is clipped to a beam in the lower part of the cabinet with two clamps.

As was stated in section 2.c Project agreement the cabinet must be able to hold 750 N on the weakest point. Those points are located at 1/4th and 3/4th of the beams on the top part of the cabinet. They were tested with a load of 80 Kg without damage.

II. Gondola

One of the main elements of a V plotter is the actuator or gondola, the design of this part have huge effect on the drawing quality. From Matlab simulations we saw that the number of strings does not affect the tension distribution for the motors and working area. Oscillations are strictly related with the centre of mass and moment. On the one hand, three and four strings configuration can reduce wobbling, in contrast the number of strings increase complexity and made the design less robust, allowing that somebody touch the strings while the robot is drawing.

The first question that we answer was which material was better for link the motors with the gondola. Other robots have used timing belts and sprockets but they are expensive and elasticity can introduce errors. Beaded cord is cheaper and still offer good accuracy but is not designed for engineering purposes and manufacturers does not provide strength values or lifespan. Our choice has been use a string made of Ultra High Molecular Weight Polyethylene or *UHMWPE* (20). This material is 15 times stronger than steel that allow us to use very thin light weight cords with really low elasticity. The string that we use was 1mm Dyneema with a nylon cover for protect the core from friction, breaking strength was greater than 550N and elongation at 1kg load, that was the expected weight of the gondola, tends to zero.

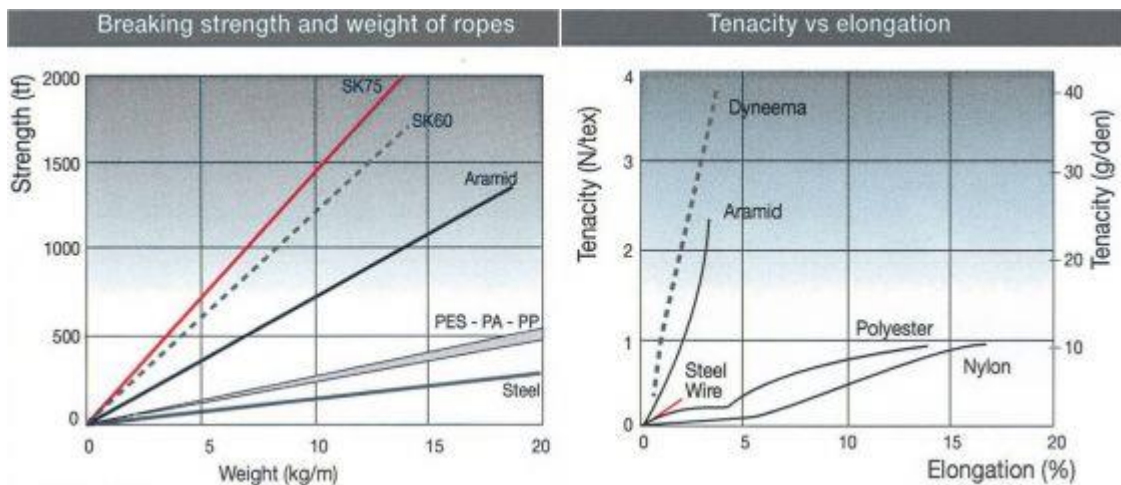


Figure 18 UHMWPE Dyneema SK75 / SK60 properties. Graph provided by Mooring Solutions GmbH

We build a simple gondola with three attachment point to check the differences between the setups of the picture above experimentally.

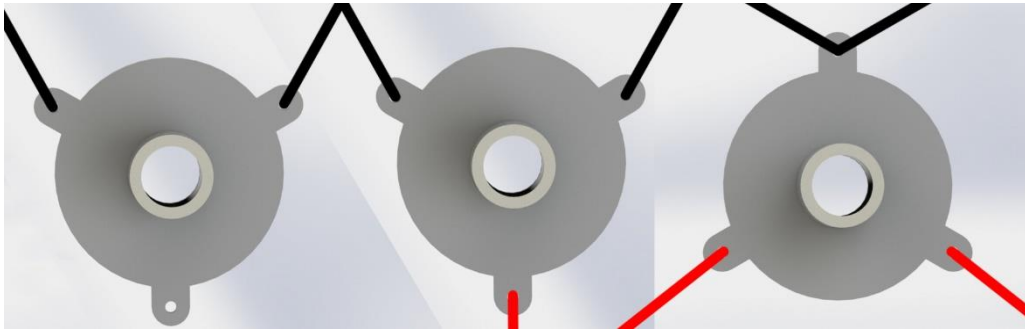


Figure 19 - Attachment setups. Top side strings are directed to the motors and red strings are attached to a constant weight.

With that setups we get that the performance of a gondola with two strings attached off-centre was similar to the four strings off-centre configuration. The second configuration was able to run faster and reduce the wobbling but require higher torque.

The second prototype was designed employing a linear lifting mechanism activated with a servo. The servo wraps a string and lift the pen that is attached inside a guide. At the same time, we build another prototype, it was made with a slider powered with a rack and a pinion. The rack was attached to the internal cylinder and pass through a slot of the external cylinder. A mount for the servo allow the pinion to engage and lift the pen when was needed. Later we run a quick test connecting both prototypes to an Arduino Leonardo board and code a short program for made the servos lift 500 times.

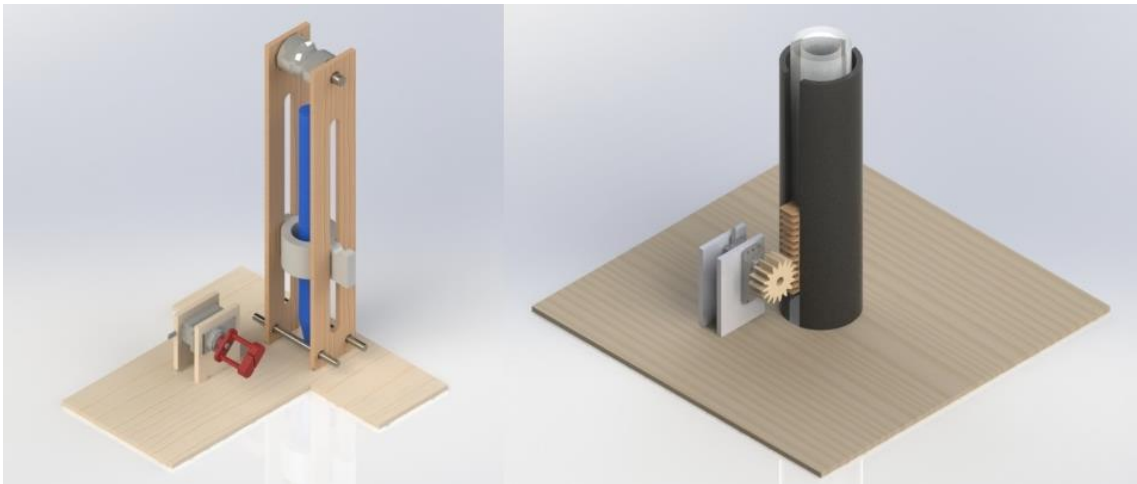


Figure 20 Gondola-Prototype 2 on the left and prototype 3 on the right

With the test we observe that the string prototype was more accurate, and allow to reach the same position between liftings, adding two springs at the sides of the mechanism the contact with the paper was more steady, but the slider was not smooth enough and get stuck during the test, moreover change the pen was not practical and require calibration on the robot.

From the early prototypes we decided that none of them fit our requirements of reliability, repeatability and accessibility, for that reason we made a new design with an improved lifting movement. The third prototype uses an aluminium profile specially designed for build slider mechanisms and four bearings for allow to have a sliding platform with the pen on the frontal

part. The platform for the pencil has 6mm thickness and is made with two layers of Medium-density fibreboard or MDF, that allow to build a socket where to fit a 3d printed part that will hold the pencil with a screw, both parts have magnets for keep them together but at the same time allow to remove the pen easily.

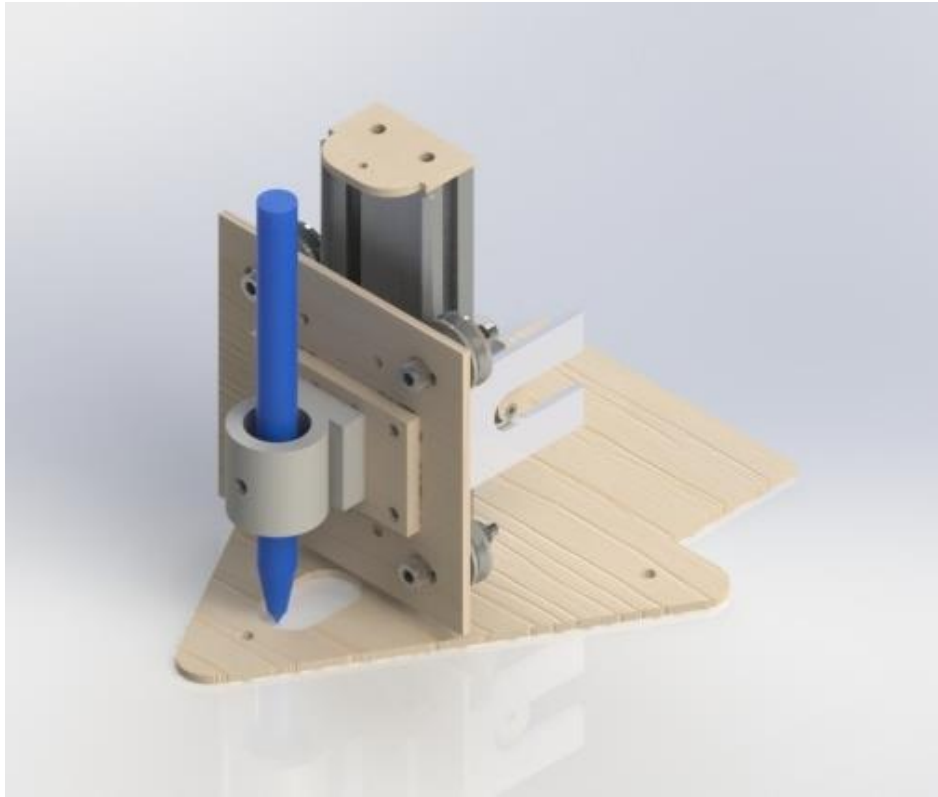


Figure 21 - Render of gondola prototype 4

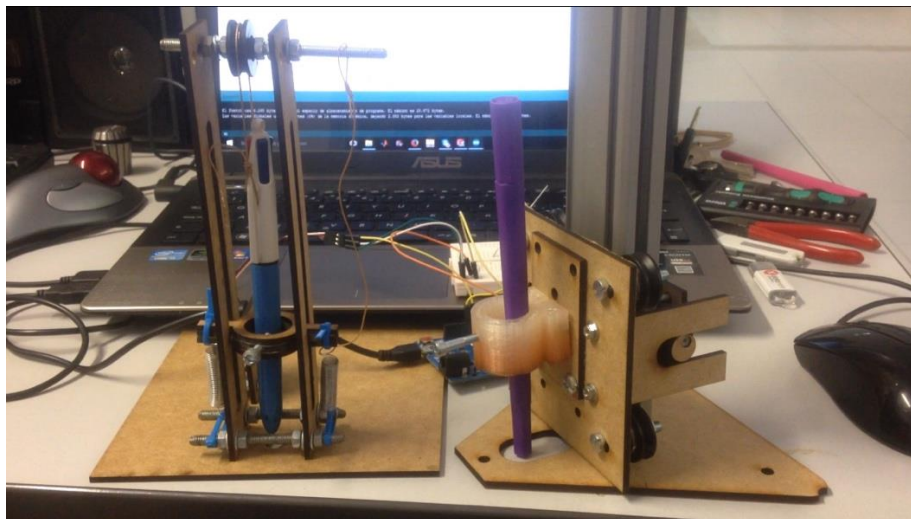


Figure 22 - Gondola Prototypes 3 and 4 running a 500 liftings test with an Arduino UNO

This prototype provide great improvements form the other models. The pen change is user friendly and the slider was more reliable. When de gondola was placed on the robot setup perform fine at low speeds but when it increases starts to oscillate. As in this design the pen was not placed over the attachment point of the strings it was also displaced, the results were good enough for draw abstract shapes and data representation that was the main use of our

project but we want to made a multipurpose robot that also allow to draw complex shapes and pictures.

The last prototype before made the final version was a revision of the previous one. In the ideal situation the strings are attached at the same point that the pen. If the strings are not link at the same point when the gondola reach the sides the position of the pencil is different than the intersection of the strings, and produce some error. When both strings are together but the pencil and the centre of gravity are at a different point we can get more or less displacement due to the oscillations induced by the inertia of the base. The closest way to the ideal condition was use two bearings, one for each string and place the pen at the centre. In this setting the gondola can oscillate but as the pen is at the centre of rotation it turns over itself without displacement.

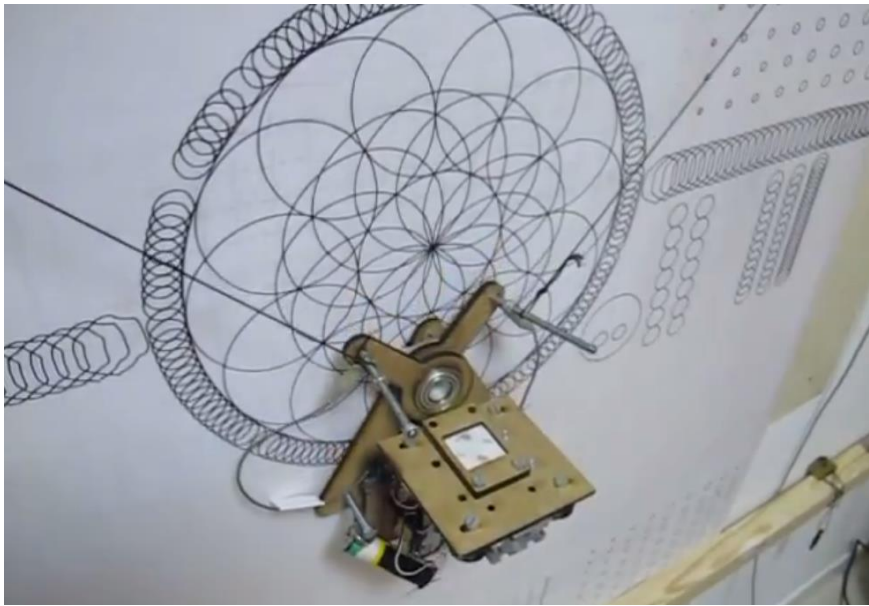


Figure 23 Review of gondola prototype 4 with bearings and pen on intersection points drawing test patterns

Until this point we used laser cut MDF of 3mm and 3D printed parts for all the components but the final version needs to have better finish. We choose machine all the parts that where possible with the CNC machine. The material that we use for this purpose was a layer of 6,7mm delrin, that is an acetal homopolymer easy to machine and with low friction and high wear resistance characteristics.



Figure 24 - Render of the final design of the gondola

The CAD models made with Solidworks need minor changes for adjust them to the CNC machining process as adjust the diameters from the tolerances of the laser cutter to the CNC drilling machine and modify corners and slots that require push fits as the servo lever.

Our CNC machine use LinuxCNC as well for the control and required a gcode input file. Those files have been generated with Fusion360. This tool allows to choose the way that the material is removed, simulate the trajectory and add final details as chamfers.

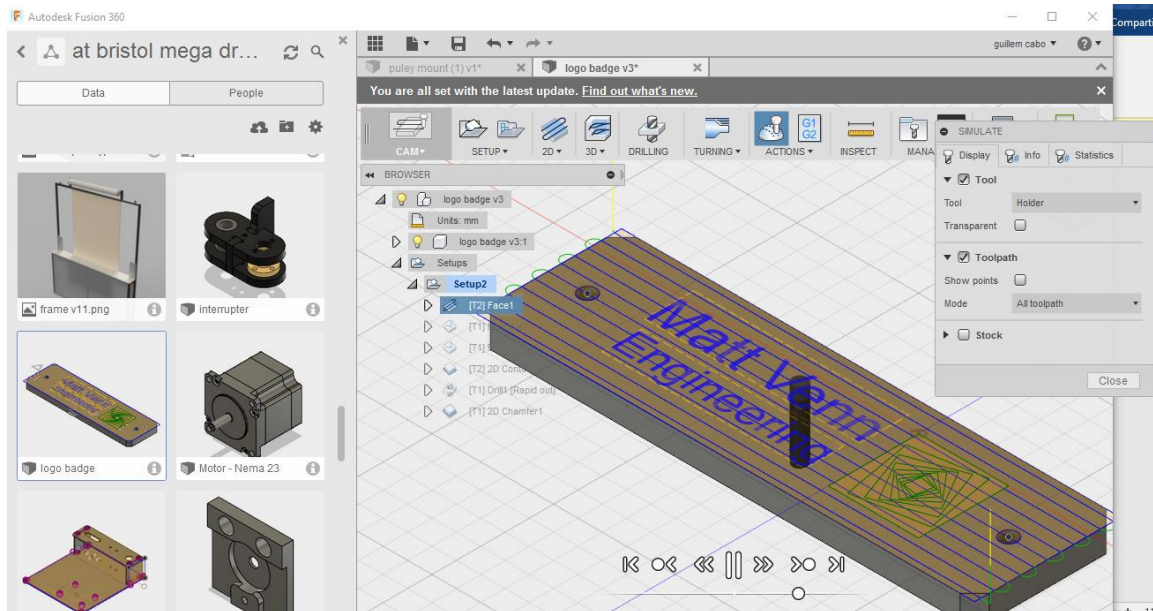


Figure 25 - Interface of Fusion360 at toolpath simulation before machining the plate of the company for the robot.

III. Motors setup

The setup for the motors has not been trivial in this project. We try to solve the problems of the homing and improve the drawing definition, motors selection and installation have a huge effect on this.

We test several string wrapping mechanisms and sensor distribution for the homing of the robot. We have to take in account that in contrast to a cartesian CNC machine a V-plotter shall be able to identify which axis to home first because if the machine is in a configuration where the length of Y axis is too short and the robot start to home X the motor will put all the effort to wrap the string of this axis but Y will not allow to reach the desired position, tension will increase significantly and as consequence motor will lose steps or break components.

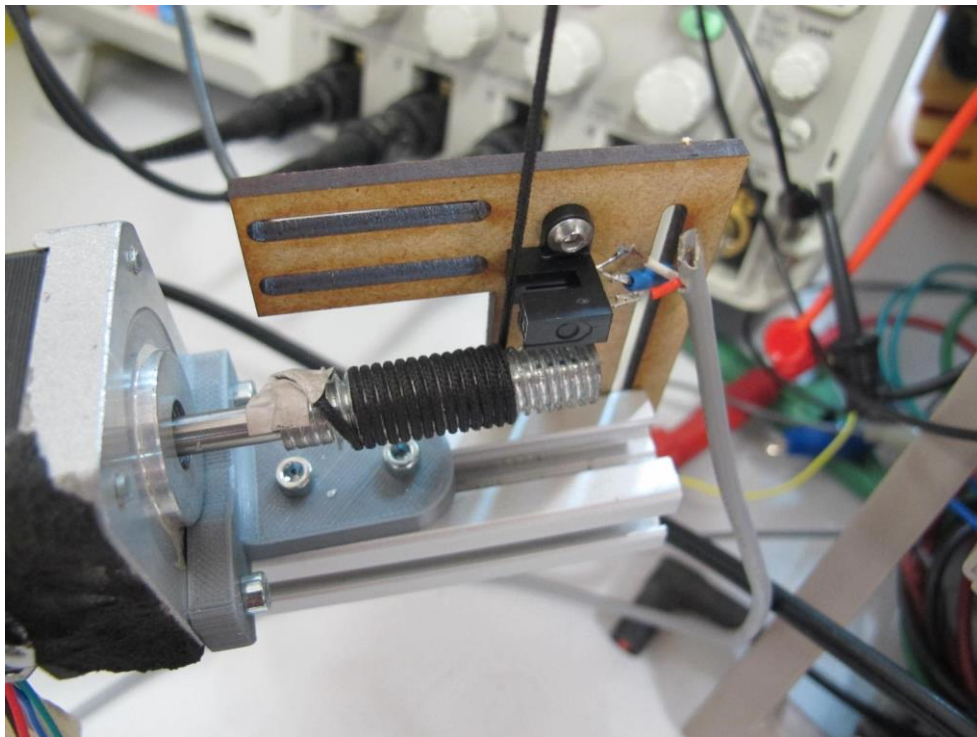


Figure 26 – Test setup with NEMA 17 stepper motor, optical sensor and threaded rod.

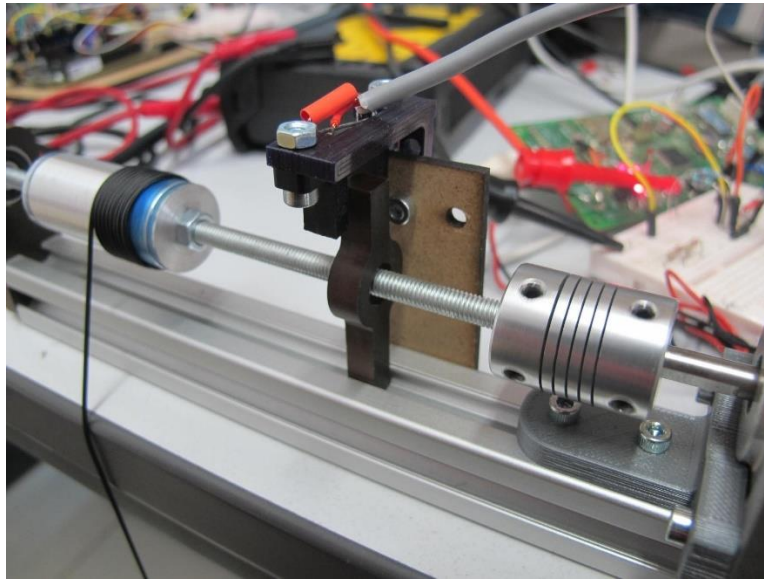


Figure 27 - Sliding mechanism driven with a NEMA 17 stepper motor with a string wrapped around a flat aluminium cylinder and optical sensor for identify home position.

The accuracy of the different setups was measured with an encoder. With that data we were able to identify problems with the setup as elongation of the string and misalignment between the motor and the threaded rod.

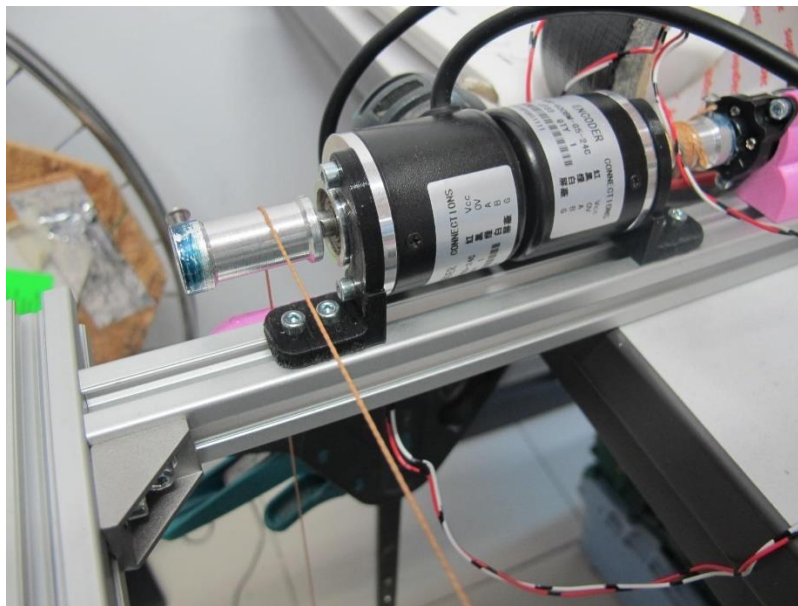


Figure 28 Encoder used for measure the displacement of the string while motor setups where tested

From the Matlab simulations and the Excel calculations of Appendix III we obtain that if we use a gondola of one kilo, 75% of the working surface and a 35mm of diameter rod for wrap the string the minimum required holding torque was of 47,60Ncm but that value increase up to 120,38 Ncm when the motor is driven with a 4 microsteps per step. For that reason we decide to include a reduction gear with 0,56 ratio that will reduce the maximum speed but increase the holding torque. With the gear configuration the minimum holding torque was 69,10 Ncm. Which let us to choose between a large range of motors in the range of prices between 50 and 100 pounds.

Adding a reduction gear and use a motor with high holding torque help use to solve also another problem. When the power of the motor is disconnected as the stepper has permanent magnets we get a detent (21) torque around of the holding torque that allow a smooth descend of the gondola to a safe position instead of fall freely and crash over the cabinet.

Consequently we designed the final version of our motors setup with the experience obtained from the experimental setups and the simulations for obtain a reliable mechanism that increase the holding and detent torque of the motor, allow to wrap the same amount of string with each step and provide a one-bit encoder, crucial for determine the homing sequence required in each case.

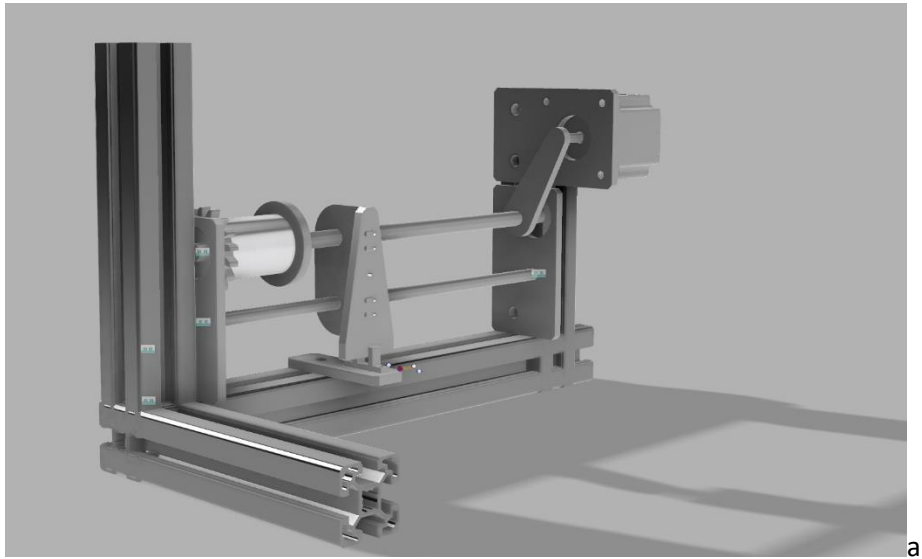


Figure 29 - Render of the final motors setup

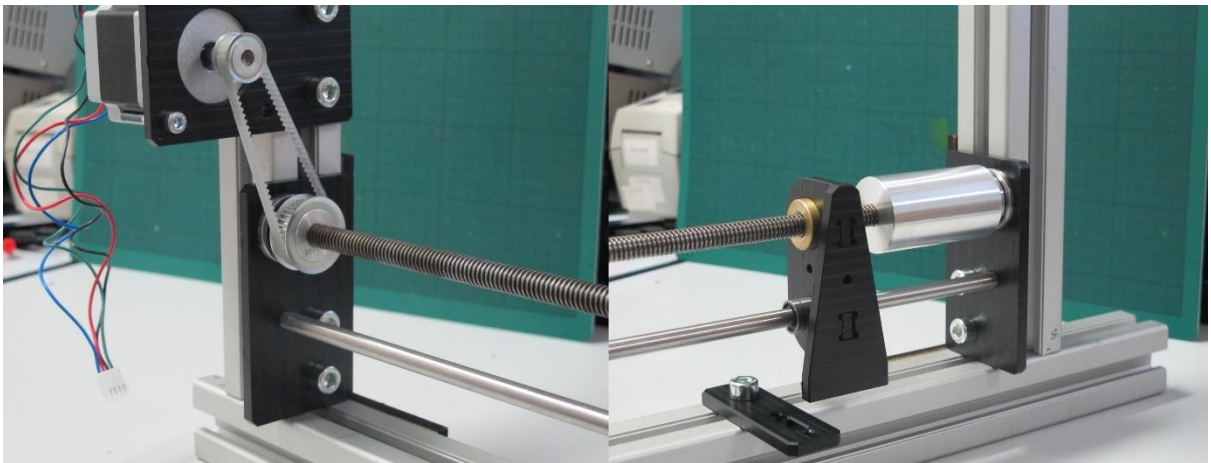


Figure 30 - Details of the motor setup, on the left hand side there is the reduction gear and on the right hand side the sliding mechanism that activate the optical sensor and the aluminium spool where the string is wrapped.

The final motor setup that appear in the pictures above has a plate where the motor is mounted. This piece can slide up and down of the aluminium extrusion for adjust the tension on the timing belt that connect the motor and the belt pulley. At the lower part there are two delrin parts that hold straight a rod and a spindle. On the right side of the picture over this lines there is the

component that interrupt the optical sensor, when the motor is on the spindle force our piece to follow a linear displacement and that is used for identify if the string is coiled or not.

IV. Paper roll

The robot draws on a paper that shall be advanced weekly. For make this process easier we provide a structure on the top of the robot where a roll of paper can be attached, in that way the operator can unclip the ends of the paper located on the lower part of the cabinet and pull until the paper is replaced without the need of use a ladder to reach the top of the tree meters tall robot.

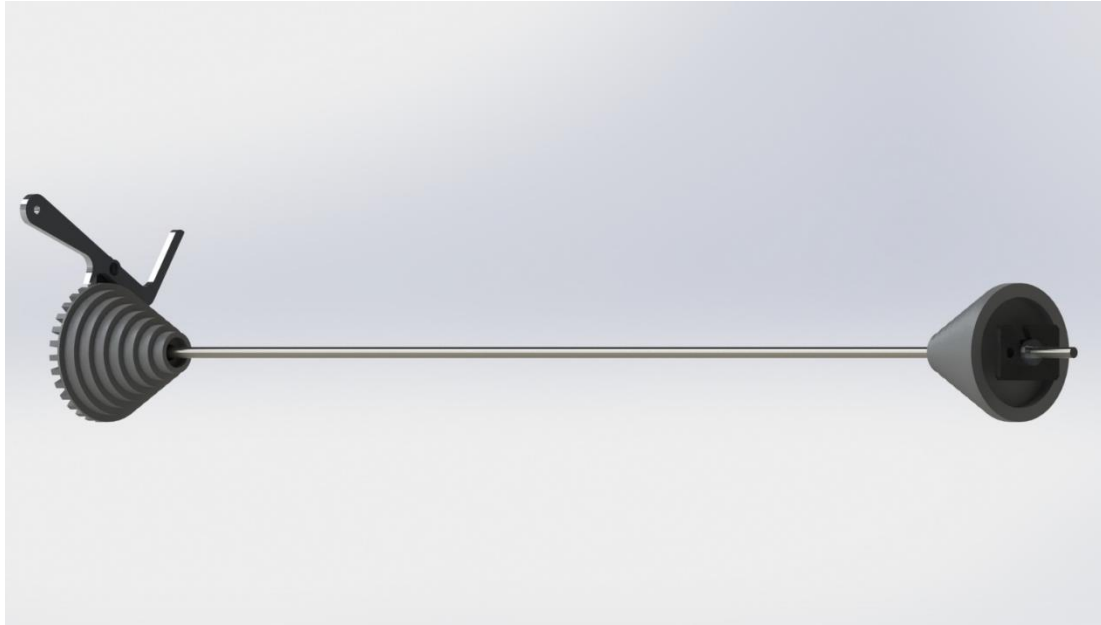


Figure 31 - Render of the final version of paper roll holder

This part requires to be locked automatically for avoid that the visitors pull down the paper. We use a ratchet mechanism powered with the gravity. The lever has an asymmetrical design, the centre of mass is different to the rotation axis, this produce a rotational force and make the lever engage with the cone limiting the movement. When the operator wants to pull down the paper he only needs to hang a stick that is hidden on the lower part of the cabinet to the external part of the lever, then the cone can turn freely.

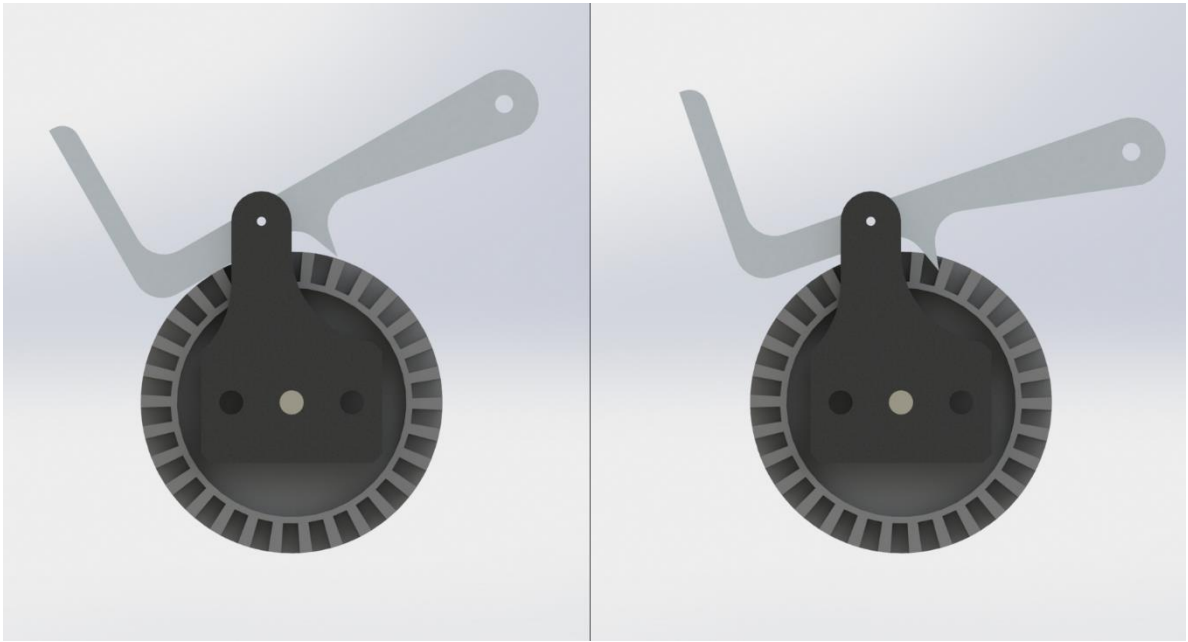


Figure 32 - Paper roll mounting, open position at left hand side and lock position at right hand side.

The cone design allows to use paper rolls with internal diameters between 30 and 90mm, giving to the customer the flexibility of choose between different brands of paper rolls.

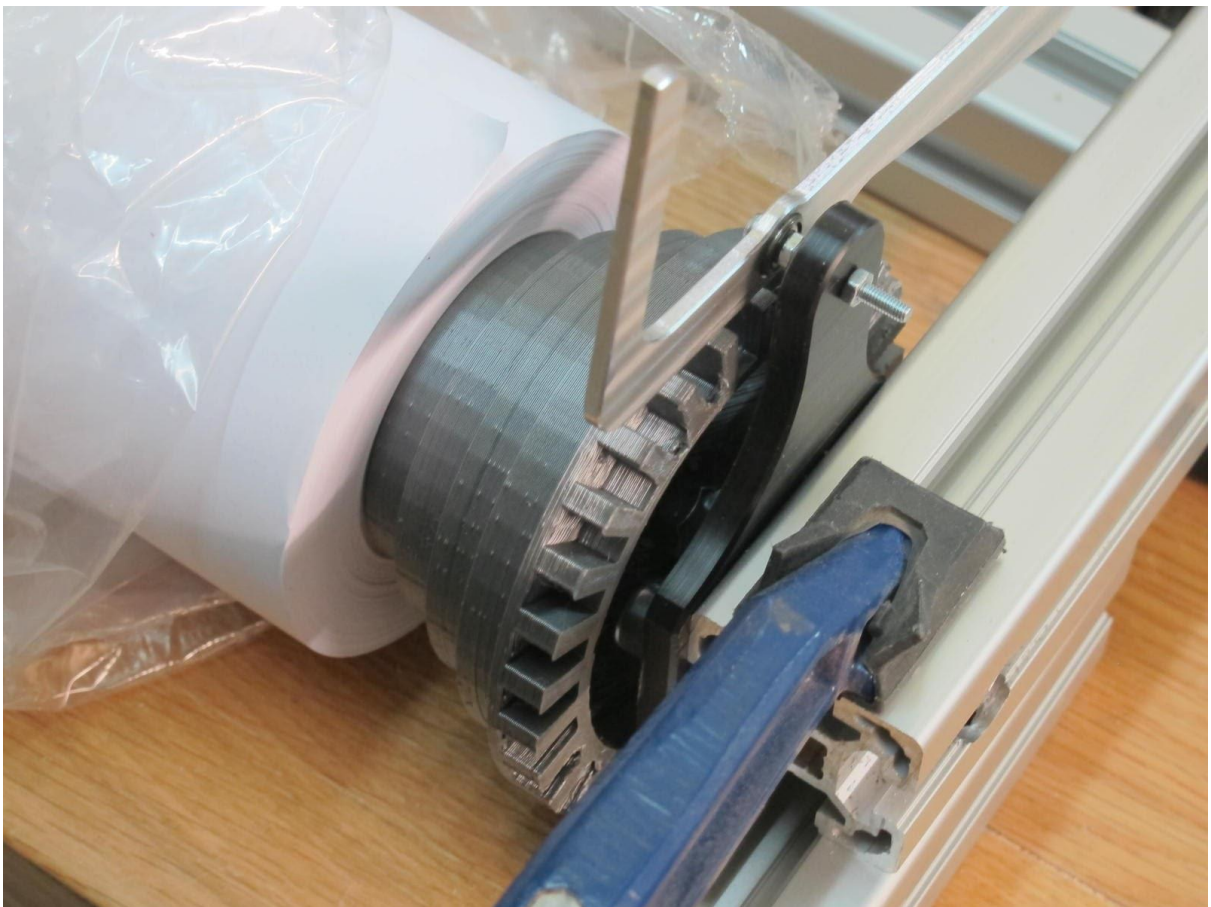


Figure 33 - Paper roll assembly in At-Bristol at installation day.

V. Guide pulleys

As the motors were located at the lower part of the robot inside the top part of the cabinet, we needed to send the strings to the top part of the robot and after that redirect them to the gondola. We must design a part that guarantee that if somebody lift the gondola the string will continue in the right place without getting stuck or even worst jumping out making the gondola fall.

In addition, we were interested to keep the tension in the part of the string that come from the motors to the pulley even if the other side, from pulley to the gondola, was loose. That will make easier coil the string around the spool without mistakes.

In the initial setups we use two acetal Pulley of 20 mm that guide the strings, they work smoothly and does not produce distortion of the drawings but where not able to keep the tension on the motor side. That make us think about implement a piece that provide enough friction for avoid that the string become lose on the motor side if somebody lift the gondola but still let us to draw when the gondola is hanging.

We started designing a MDF plate where the pulley was attached. At the side of the motor we put two M3 screws that had a flat surface at the contact point with the string. We thought that changing the angle and number of turns that the string made around the screws will increase the friction as in the climbing devices used for rappelling but the results were not clear.

We continue with this problem and we try to use a Perfluoroalkoxy tube, better known as PFA, inside a MDF sandwich instead of screws. This design was more aesthetic and guaranty that the string does not jump out of the place but the results continue without be good enough and for adjust the friction was necessary to make a new piece.

The last experiment that we made for design a piece that hold the tension of the string was a CNC machined piece of delrin with a zig-zag slot with several inputs and outputs, in that way we were able to adjust the friction easily choosing the number of turns made at the interior of the piece.



Figure 34 -Delrin tension holder with adjustable friction

With that design we saw how part of the tension remain at the motor side without become undone but we also detect a distortion in the circular shapes. As we were changed several components and the position of the motors the same day we thought that the problem was with the calibration, but we ignore the problem for the moment. After few days of intensive testing we detect damage on the cover of the string. We replaced again this component with the initial pulleys and the distortion disappeared.

Knowing about this vulnerability we decided to include a capacitive sensor in the gondola for stop the motion if somebody touch it and set the rest position at the top of the frame. The pulleys where enclosed inside a Delrin case that allow to attach them to the Aluminium extrusion and guaranties that the wire does not get stuck at the joints.



Figure 35 - Delrin enclosure for the pulleys used to redirect the string at the top part of the robot

VI. Charging Station

As will be explained in section 3.d the charging setup changed from an inductive charging to a regular charging, where the connections were physical.

This system has a dock on the top of the robot with two arms with a limited range of movement, the robot is preconfigured to rest at the charging station making contact with the charging pads on the gondola.

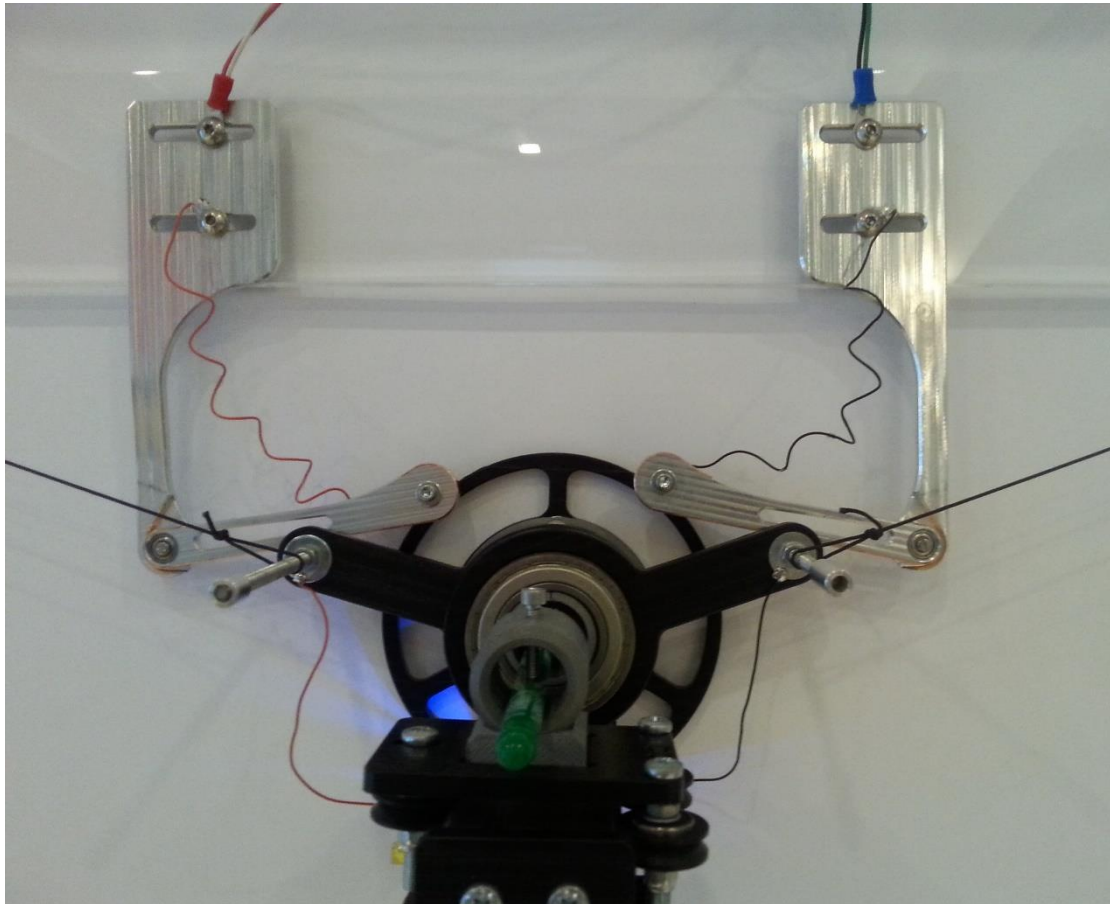


Figure 36 - Gondola at charging position

The arms of the charging station can be adjusted and the design of the gondola guaranty that the battery cannot be short-circuited.

For guaranty the reliability of this mechanism we bypass the bearings with wires and Silver Conductive Grease was applied to contact surface of the charger and the pads of the gondola.

VII. Calibration tool

The design of the pen holder was made thinking about the maintenance. The pen change must be done once a week. Our objective was make that procedure as easy as possible and avoid changes in the robot, for that reason the pen can be removed of the structure but an extra calibration tool was required for adjust the length. If the pen is not placed properly the quality of the drawings can decrease or the tip can be always in contact regardless of the motion of the Z axis.

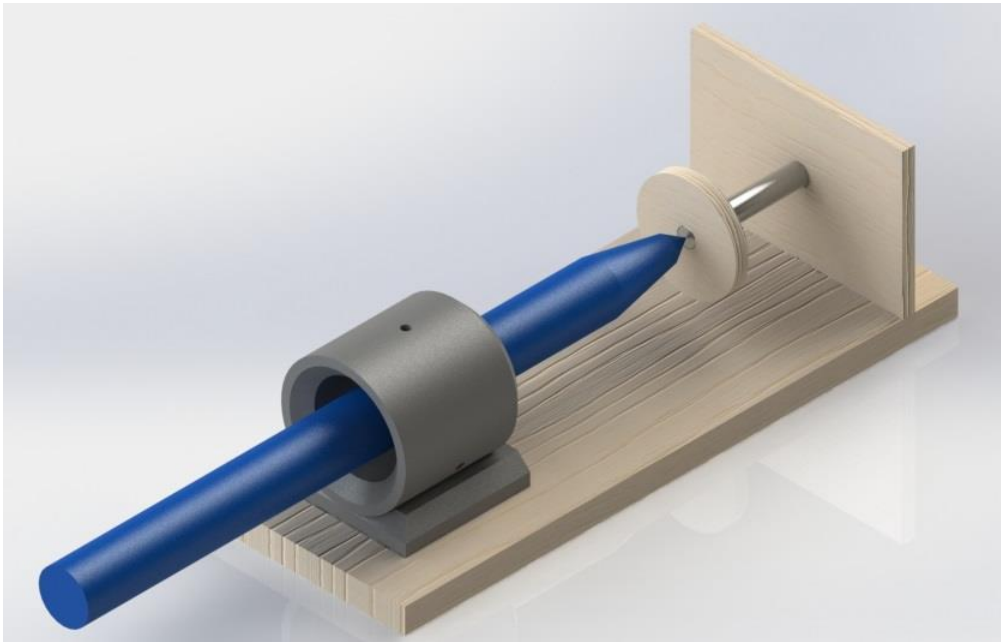


Figure 37 - Render of the calibration tool for set the length of the pen

VIII. Other parts

In addition to the main functional components of the robot we designed other parts that we considered not relevant enough for require a complete section on this memory and have been included on the Appendix VI or virtual resources. Some of those components were only used for the prototypes or experimental setups.

3.d Electronics

I. PSU

For the robot we used two RS pro power supplies. The first one is a 156W 1 Output, Embedded Switch Mode Power Supply (SMPS), 24V dc, 6.5A and was used for drive the motors. The second one is a 54W power supply of the same type that the first one but with to outputs of 5V,2A and 12V,6A dc.

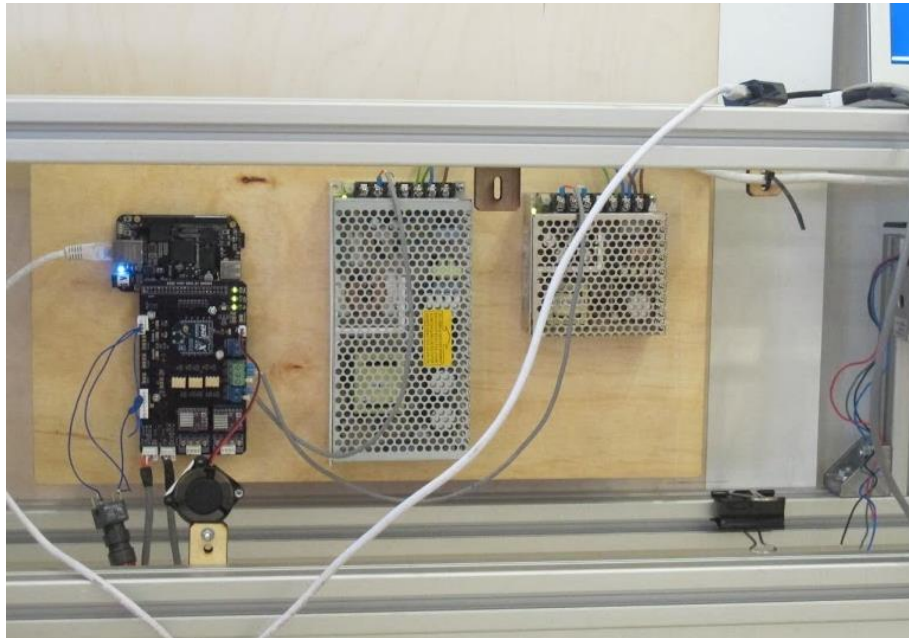


Figure 38 - Robot at installation day, BeagleBoneBlack and BBBB PBC on the left hand side, 24V PSU on the middle and 12V PSU on the right hand side.

II. BeagleBoneBlack

This is a low cost, open-hardware and open-software single-board computer with an active community of users and compatible with Ubuntu. From the hardware point of view, we have an AM335x 1GHz ARM® Cortex-A8 processor, 512MB DDR3 RAM, USB client, USB host, Ethernet, HDMI, micro-SD adapter, 92 I/O pins and two Programmable real time units (PRU).

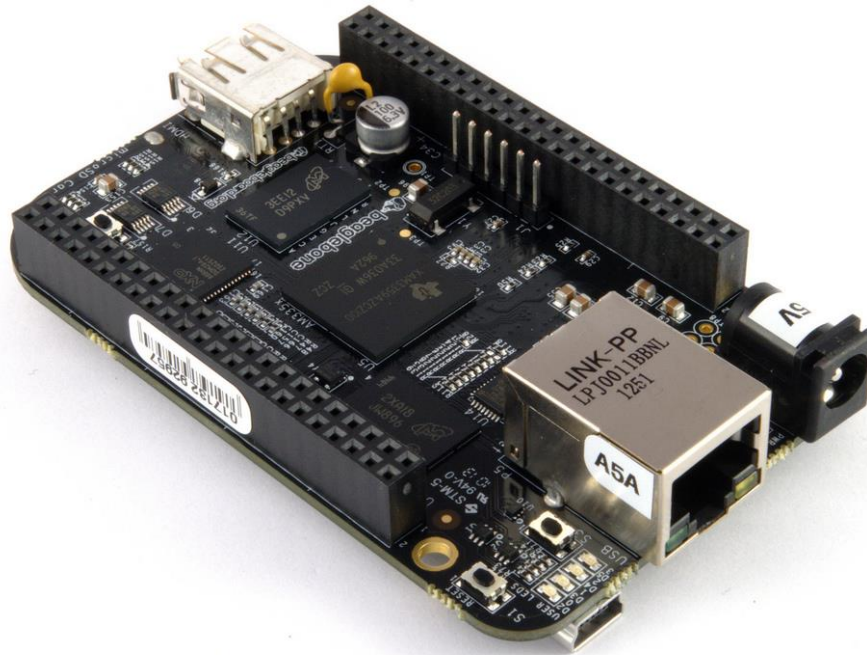


Figure 39 - Picture of BeagleBoneBlack by Gareth Halfacree (22)

From the 92 I/O available we only use port P9.

1st func	Pin number	1st func	2nd func
gnd	1	2	gnd
3.3	3	4	3.3 led:power
5v	5	6	5v
butt:power	9	10	
butt:stop	11	12	butt:reset
gondola charge	13	14	enable pulled high/low & with led
ystep	15	16	ydir
xstep	17	18	xdir
n/a	19	20	n/a
homex	21	22	homey
gpio1	23	24	xbee: rx
gpio2	25	26	xbee: tx
gpio3	27	28	7seg: le
gpio4	29	30	7seg: sdo
gpio5	31	32	
7seg: clk	41	42	7seg:not_oe servo:control
gnd	43	44	gnd
gnd	45	46	gnd

Figure 40 - Pins used at BeagleBone for the robot and functions.

III. BBB breakout board

This is the largest PCB that we used in this project. At the beginning we test the functionalities with the breadboard. The design includes two drv2250 stepper driver, a xbee2, emergency stop button and two inputs for the home sensor on axis X and Y.

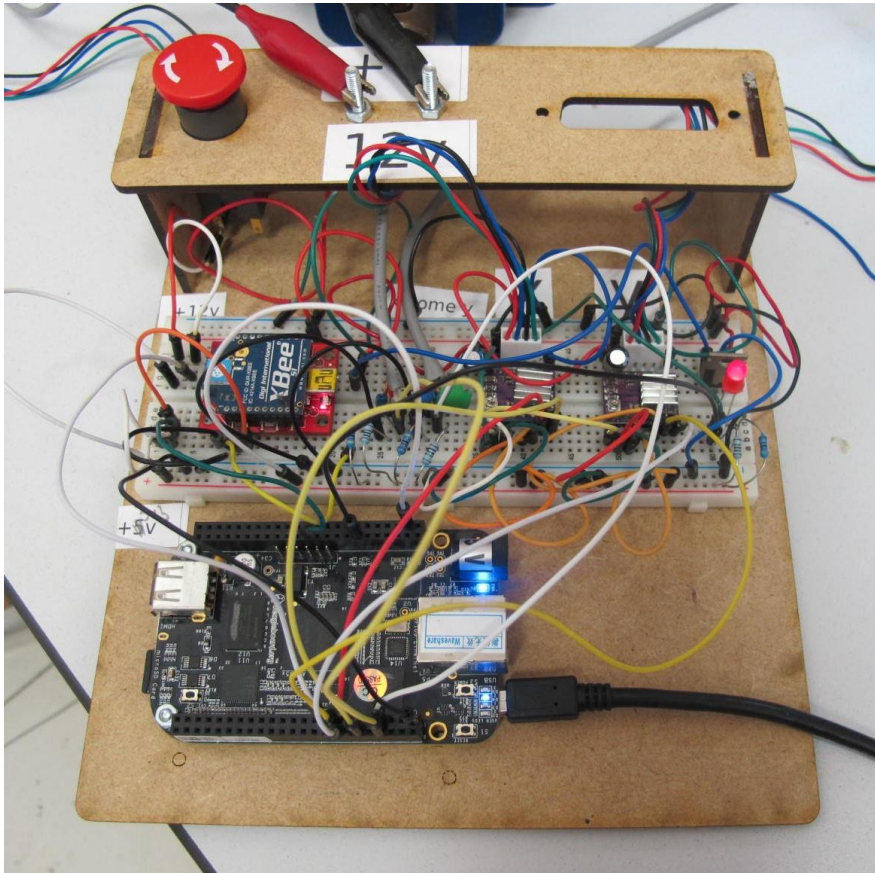


Figure 41 - BeagleBone and breadboard configured for the V-Plotter.

The breadboard work fine and we transferred the design to a PCB. We wanted to keep the design flexible for allow us to use the same boards for next robots even if the specifications where slightly different. We add a I2C temperature sensor connected to pins 19 and 20, extra pinout for two solenoids in case that we need to break for the motors mechanically after a power loss, pins for control a servo, three optocouplers for isolate the BeagleBone from the drivers, pinout for add a seven segment display and some LEDs for show voltage status.

The PCB is a 2 sided board with mostly STM components, SMT components were soldered with Infrared Reflow oven and through hole components soldered by hand with soldering iron. Before testing an intermediate inspection consistant of two parts.

Visual inspection at 4x magnification

- Solder joints shall exhibit complete solder melting and good wetting between parts.
- Verify orientation of all the IC, diodes and connectors.

Electrical test

- No visible cuts in the traces.

- Check short circuits between power supply.
- No solder bridging.
- No opens
- No shifted components
- Check short circuits between close pads.
- Check opens.

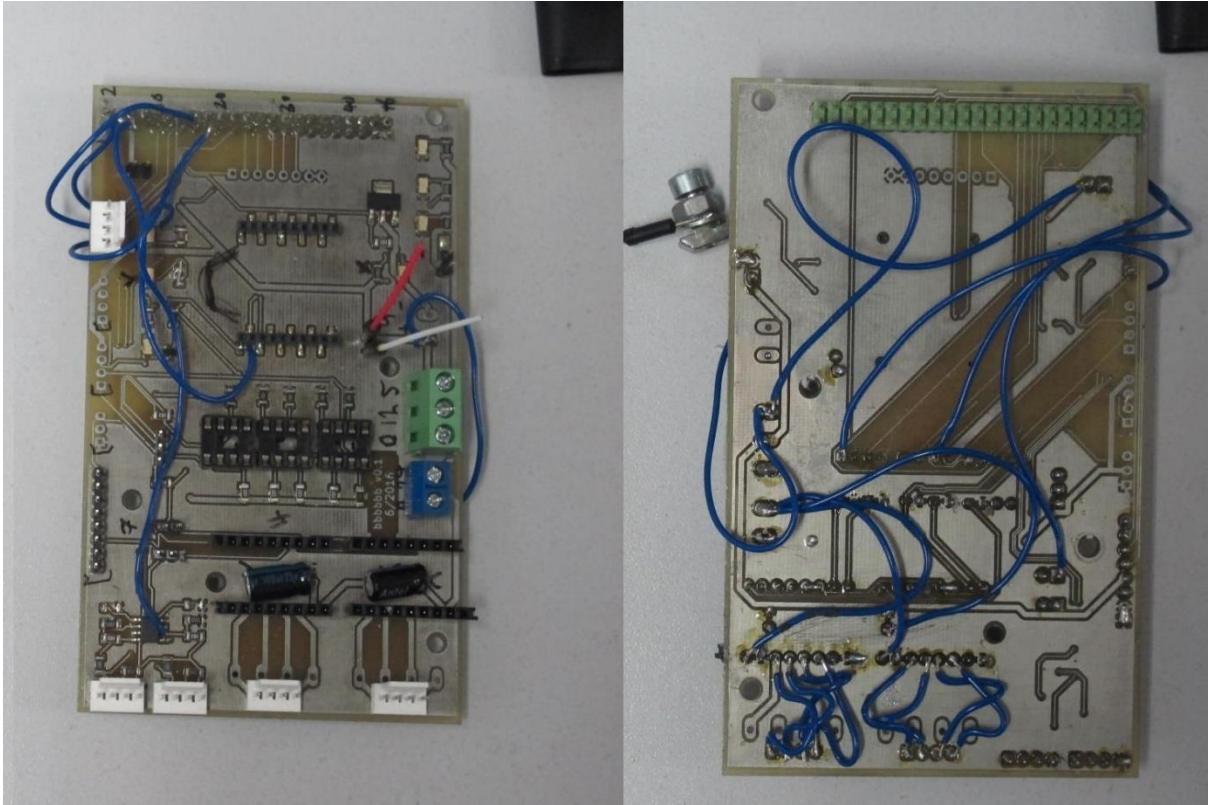


Figure 42 - Front and rear view of the bbbbb PCB.

We ordered the PCB to a local shop but was supplied with defective vias impedance and traces needed to be bypassed, after some effort we were able to make it work and verify that the design does not present problems and order the final version to OSH park.

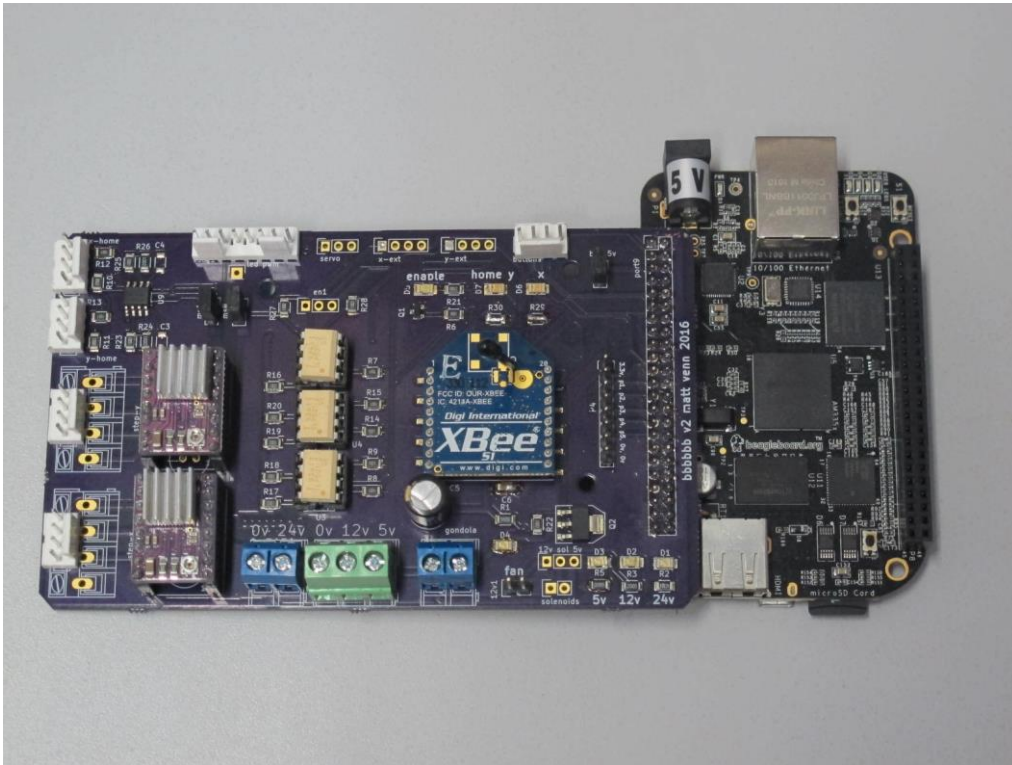


Figure 43 - Final version of the BBBBBB PCB and BeagleBoneBlack

IV. Gondola

At the beginning of the project we use a stripe board for weld the minimum amount of components required for the tests.

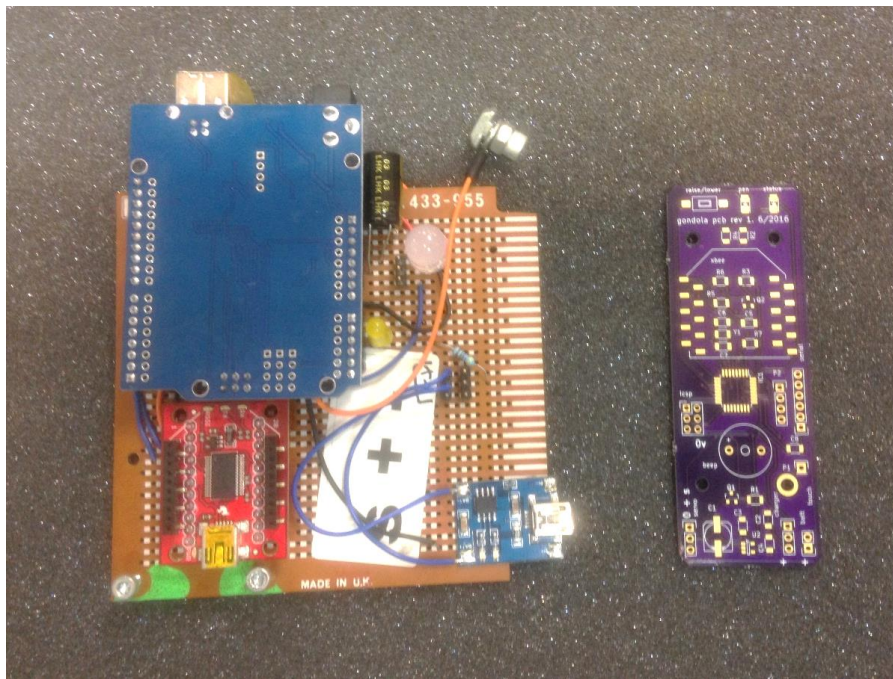


Figure 44 – Prototype and final Gondola PCBs

This board has a socket with a serial to USB connector for an xbee, servo output, a voltage regulator and a primitive capacitive sensor, in addition the board required an Arduino UNO that control all the outputs. That board was unpractical but allows us to continue with the development of the code while the PCB arrives.

The final PCB integrate all the components of the gondola in the board. It has a voltage regulator for charge lithium polymer , now on LIPO, batteries, a buzzer, state LEDs, a xbee socket, *In-Circuit Serial Programming* o ICSP port, an ATmega328p and other discreet components. It is a 2 sided board with mainly SMD components. Assembly procedure was the same that for previous PCBs, we use a reflow IC oven with the curve of temperature according to the specifications of the soldering past. Before connect it to the power the same intermediate inspection as in section V was performed.

This board use a MikroElektronika 3.7V Rechargeable Lithium Battery of 2000mAh that provide autonomy for more than seven days.

V. Big digits

For keep the attention of the visitors at the museum decide to place four seven segment displays for show a count down between drawings. This displays (23) were developed by as open source hardware project in 2014 by Matthew Venn. This boards use a TLC5916 constant current LED driver and have serial data in and out pins that allow to concatenate the displays without problems.



Figure 45 -Array of 4 Big 7 segment displays installed on the final robot.

VI. Charging setups

This were one of the most challenging parts of this project. Charge is a crucial element of the robot, by customer specifications we were allow to change batteries as part of a maintenance tasks and build an external charger, but for obtain a real autonomous robot we thought that the charging process shall be integrated with the routines of the robot.

The alternatives to an external charger where an inductive charger or a charging dock with mechanical connection with the robot. The inductive charge avoids the problems of oxidation, weak contact and shortcuts but increasing the complexity of the electronic design and the cost. The key point for choose this solution was the reliability and transference to future projects.

For the inductive charging was used the integrated LTC4120-4.2 (24) that is a constant-current / constant-voltage wireless receiver and battery charger LIPO with a charging current up to 400mA.

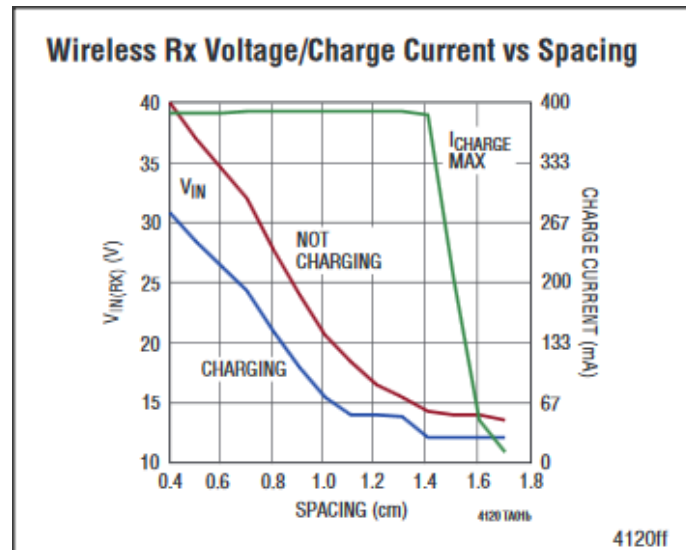


Figure 46 - Wireless Rx Voltage/Charge Current vs Spacin of LTC4120 from datasheet specifications

The specifications of the PCBs were a two sided board with mostly SMD components, using the reference design of the LTC4120-4.2. The recommended manufacturer for the components required was also specified. For avoid problems we same part numbers shall be used. The design of the layout was outsourced and the manufacturing was provided by OSH.

Once the PCB arrive we assembly the components in place and soldered in an infrared IC reflow oven. Initially the boards were tested following an intermediate Inspection. No failures where found after the following tests:

Visual inspection at 4x magnification

- Solder joints shall exhibit complete solder melting and good wetting between parts.
- Verify orientation of all the IC, diodes and connectors.

Electrical test

- No visible cuts in the traces.
- Check short circuits between power supply.
- No solder bridging.
- No opens
- No shifted components
- Check short circuits between close pads.
- Check opens.

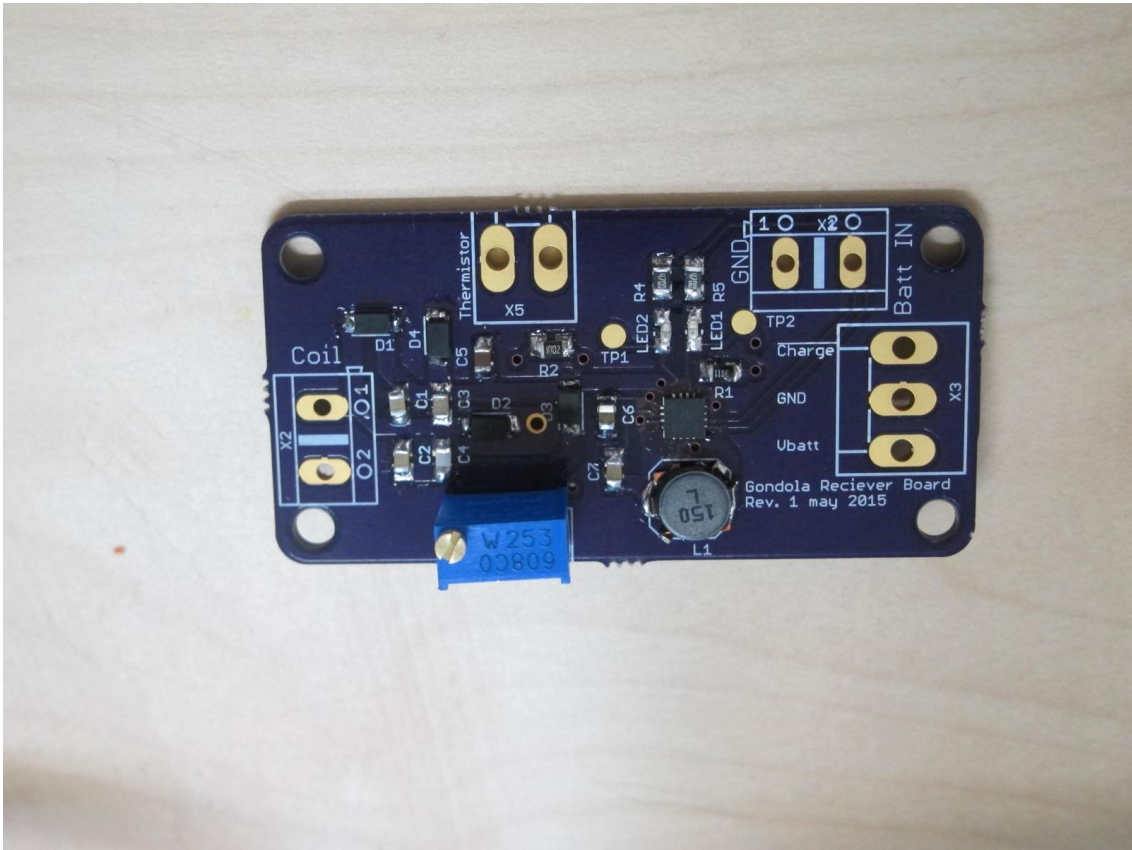


Figure 47 - PCB inductive charge receiver before tests

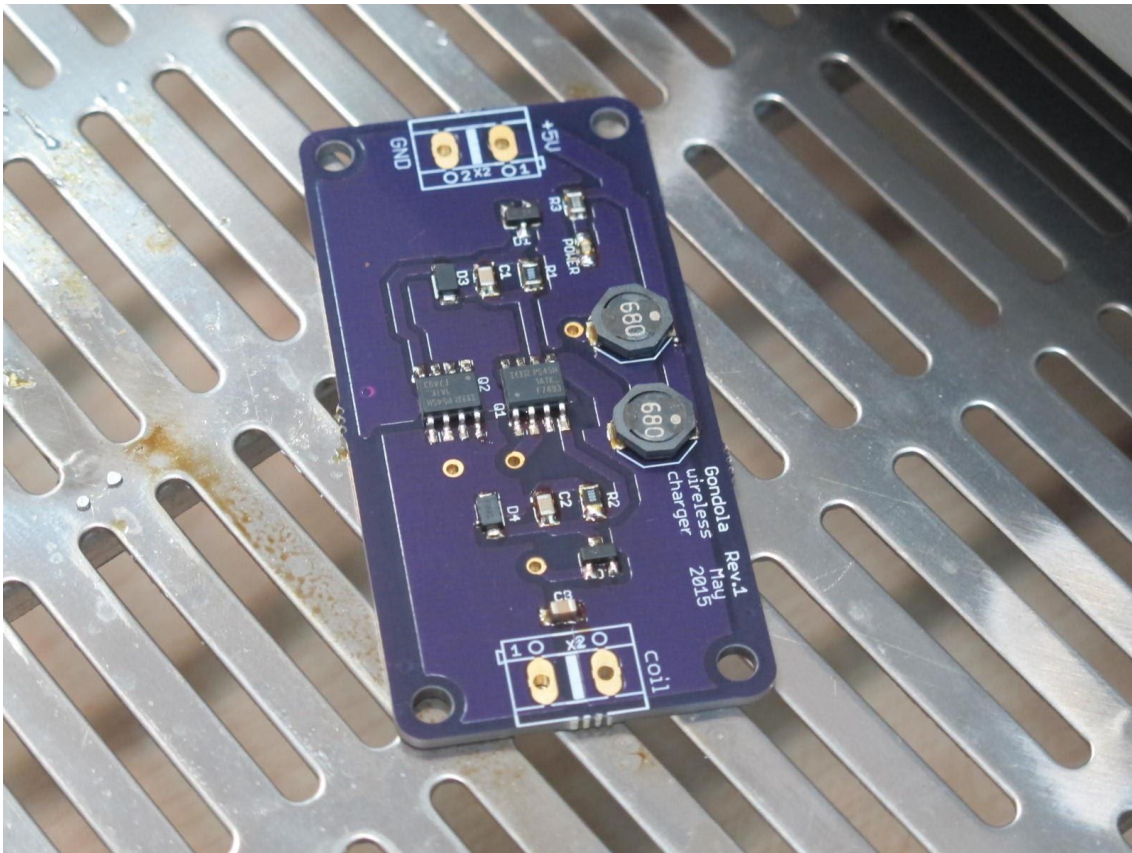


Figure 48 - PCB inductive charge transmitter after soldering with IC oven

The board pass the inspection but neither receiver or emitter provide the expected output. In the emitter wave forms shown on the oscilloscope had frequency of 15KHz instead of the 750KHz required. After debugging of the design we cannot identify the failure of the resonant circuit.

The resonator of the transmitter was bypassed by an Arduino Leonardo generating an output signal of 750KHz for activate the MOSFETs and test the receiver board.

The receiver passes the same electrical and visual quality test than the transmitter, but when the emitter was connected the behaviour was also different from what we expect.

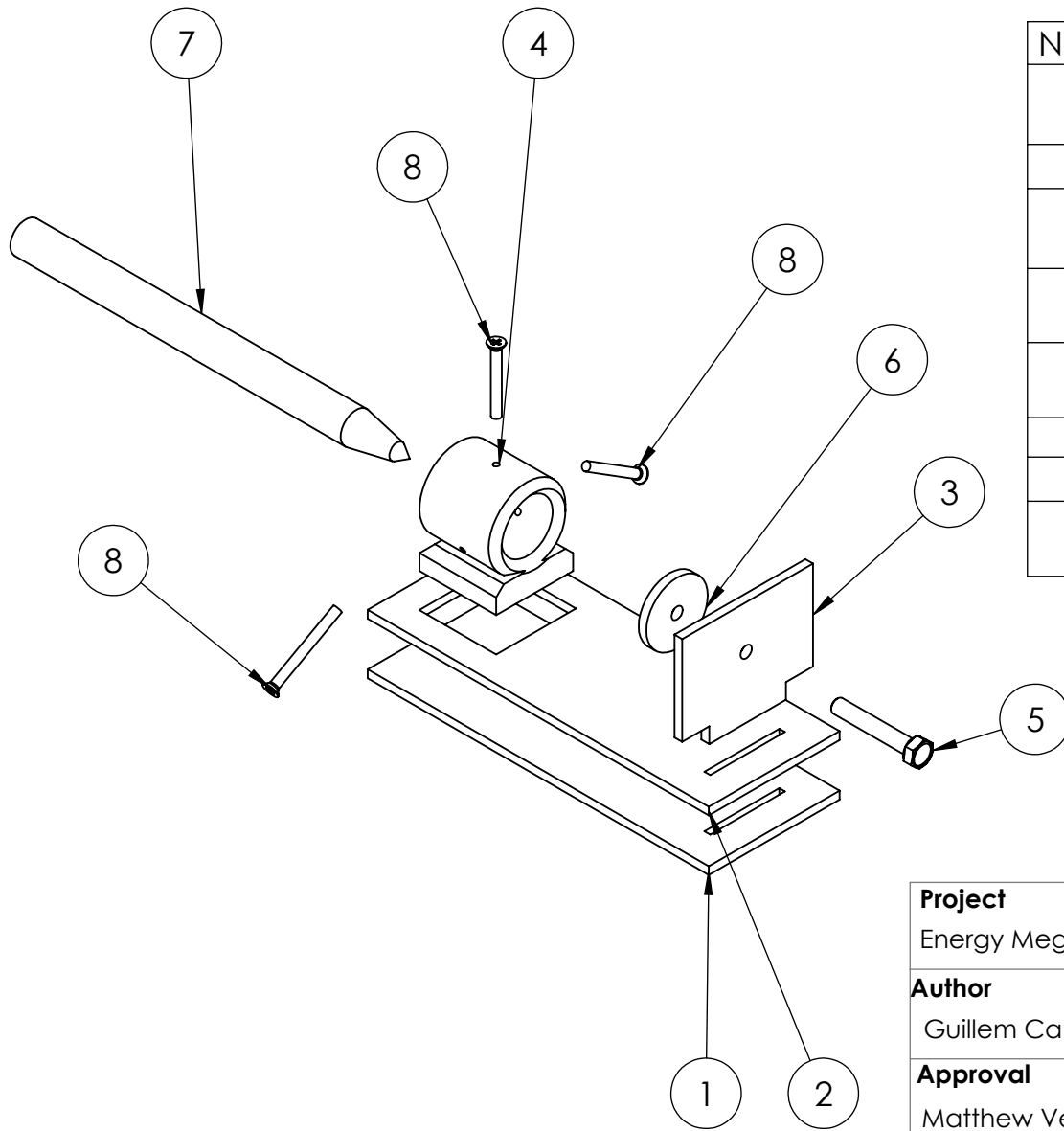
We run a more detailed inspection. Values and orientation of all the components was review, simulations were made with LTspiceIV and the demo circuit provided by LTC (25). The simulations help us for identify the signals and debug our circuit but this effort was unsuccessful. And we switch to a mechanical solution. Weeks after installation of the robot we identify a design mistake at the coil input of the receiver and further development is scheduled.

The mechanical design was simpler from the electric point of view but harder to reach the same reliability. The dock station shall be machined from a conductive material and the design shall allow lateral adjustment and had mobile parts for contact with the gondola without force the motors if there is a problem during the homing. Our solution was build a dock with 10mm acrylic and place two aluminium arms on the top, this material was selected because was easy to machine but we concern about the quick oxidation that decrease the conductivity. We test the charging station with aluminium arms for a week with good results. Further test reveal that after longer run the pads can deteriorate reducing drastically the performance. For avoid that problem conductive silver grease (26) was placed at contact surfaces.


The mechanical system has two electrodes that come from the BBBBBB PCB and are connected to the arms, the bearing that is used for joint the mobile parts is bypassed with a wire for reduce conductivity issues. Charging pads where located on the wings of the gondola. Those pins are link to the charging circuit through a crocodile connector.

4. Plans

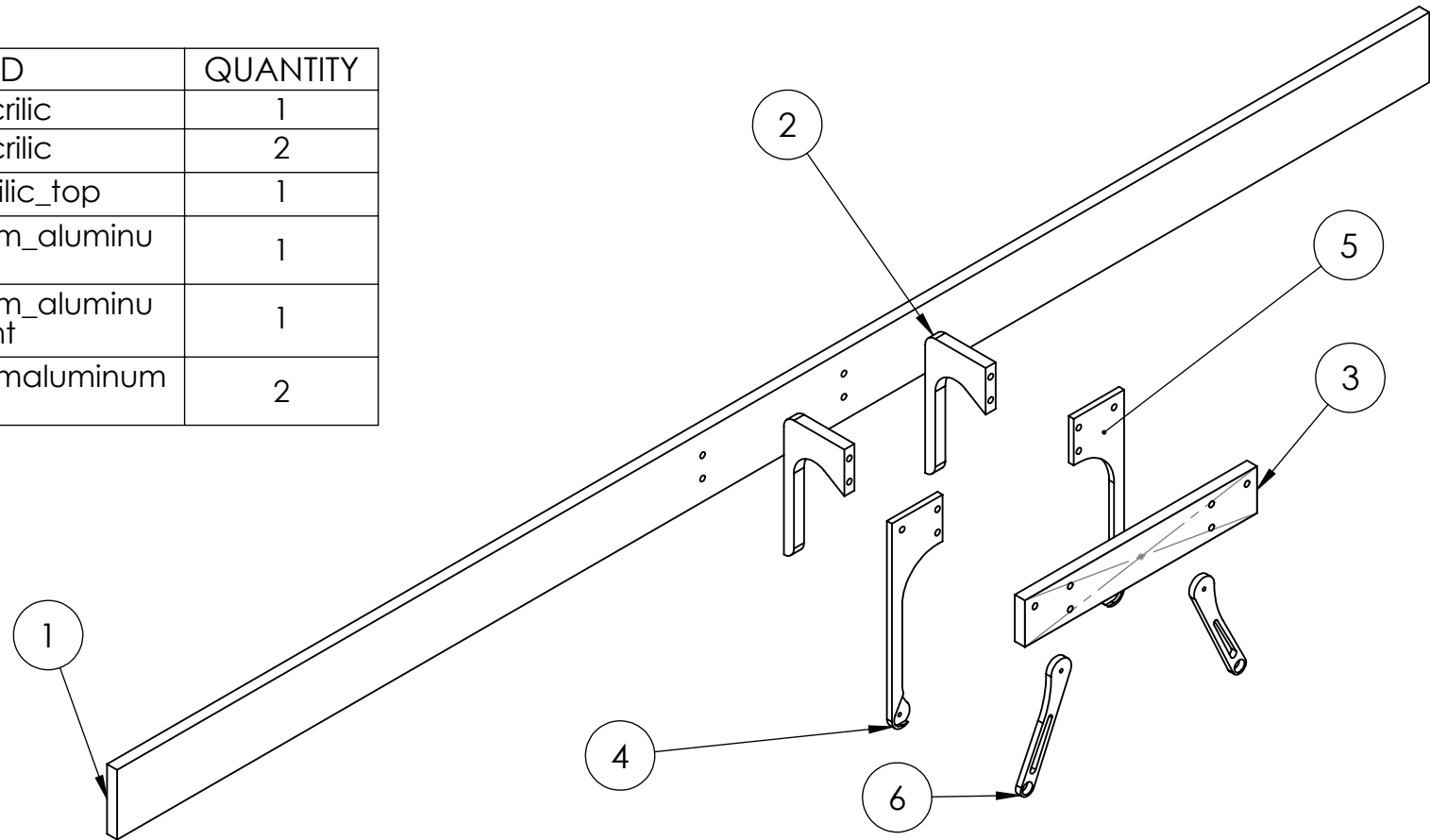
This section contains some of the more representative plans of this project and allow to have a good idea of the number of elements and distribution. In appendix VI, and virtual resources can be found the plans of other components, subassemblies and failed prototypes..




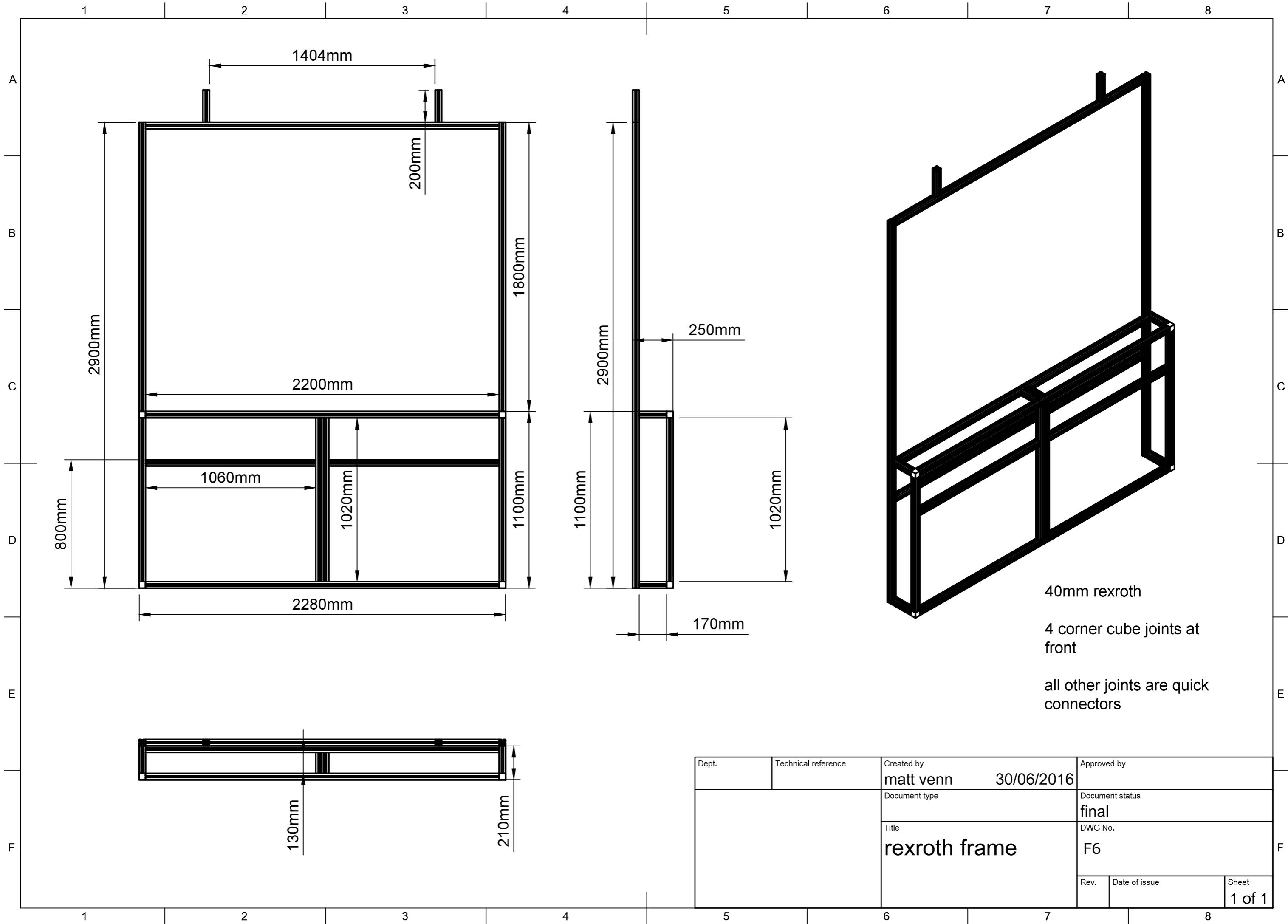
NUMBER	ID	DESCRIPTION	QUANTITY
1	layer0	Glue with layer1	1
2	layer1		1
3	screw_adj	Glue with layer layer1	1
4	bolisoportemagnetico_v2.1		1
5	formed hex screw_am		1
6	screw_limit		1
7	pen1		1
8	ISO 7046-1 - M3 x 25 - Z --- 25N		2

Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number C.a
Description C.a. e_pen_length_setter: Tap holes on 3 and 4 for correspondent screw size. Glue 1,2 and 3.		Matt Venn Engineering 

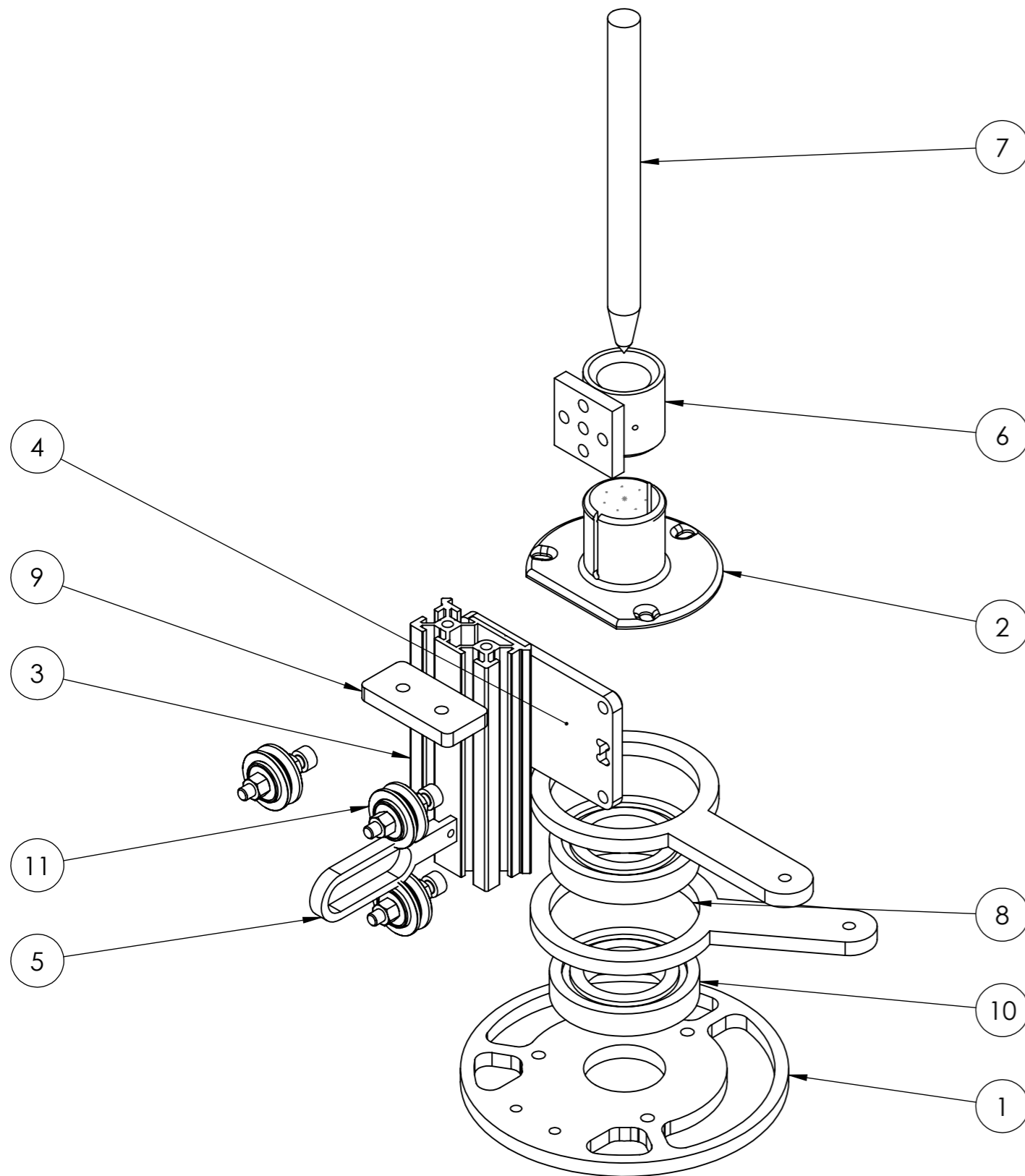
NUMBER	ID	QUANTITY
1	1300x60_acrylic	1
2	supports acrylic	2
3	230x40_acrylic_top	1
4	180x60_5mm_aluminum_arm_left	1
5	180x60_5mm_aluminum_arm_right	1
6	110x30_5mmaluminum_door	2




Project Energy MegaDrawz		Date 06/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:5
Approval Matthew Venn	Signature	Plan number CS1.a
Description E_charging: Assembly on ground before installation. Tap hole for correspondents screws. Use a 10mm bearing between 5,6 and 4,6.		Matt Venn Engineering 

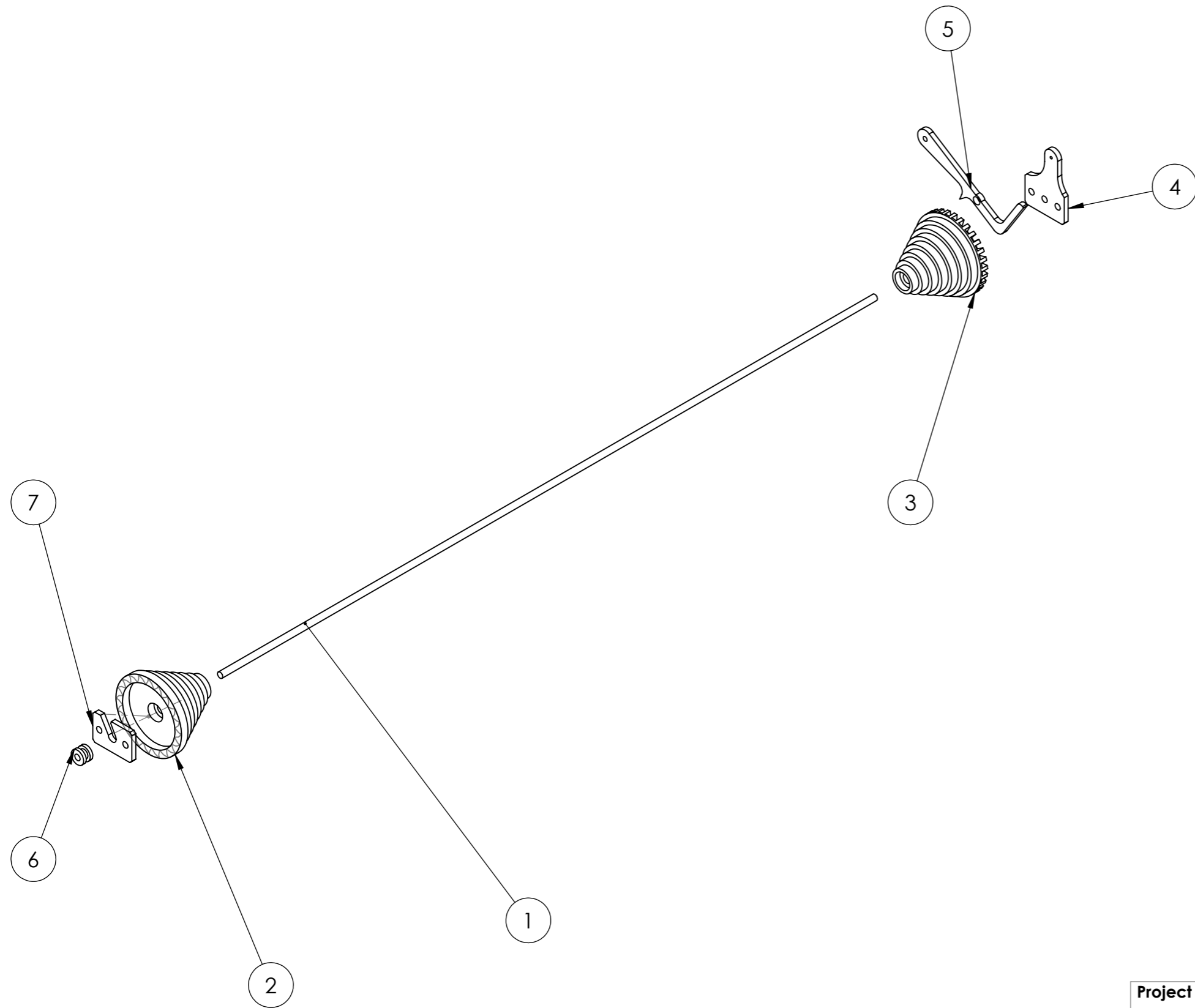


Dept.	Technical reference	Created by matt venn	30/06/2016	Approved by
		Document type	Document status final	
		Title rexroth frame	DWG No. F6	
Rev.	Date of issue	Sheet 1 of 1		




NUMBER	ID	QUANTITY
1	base_v2	1
2	bearing_connector_30mm_v2	1
3	maker slide 100mm	1
4	soporterodamientos_v2	1
5	guialeva_6mm	1
6	bolisoportemagnetico_v2	1
7	pen1	1
8	wings_bearing_55mm	2
9	slider_top_cover	1
10	bearing30_55	2
11	G5.sa	3

Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G5.a
Description E_gondola_v3: Extra screws are required for piece 2. Threaded rod of 5cm shall be screwed to both of 8 components. Verify pushfits between 8 and 10. If it is required screw piece 5 to 4. Glue magnets on socket of piece 4. Tap holes when required.		Matt Venn Engineering 



NUMBER	ID	QUANTITY
1	m8_varilla_1000mm	1
2	soporte_liso_v2	1
3	soporte_ranuras	1
4	lateral_v4	1
5	palanca_freno	1
6	h_beam_holder	1
7	lateral_v3_izq	1

Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:5
Approval Matthew Venn	Signature	Plan number P.a
Description Epaper_roll_v3: secure components with nuts when required. Use 10mm bearing at attachment point between 5 and 4.		Matt Venn Engineering 

5. Budget

AT-Bristol science museum initially approached us with a budget of £2000. We did some projections and responded with a request for £3000 combined with a provision for external sponsorship. The project would be undertaken if an industrial partner could be found to contribute an additional £1000. RS components agreed to sponsor the project with materials from their catalogue.

It was understood that the project was underfunded, but the opportunity to have a large project in the foyer of one of the UK's largest science museums was deemed to be worth the lost money in payment for labour. For this reason, Matt Venn engineering accepted around a 3x reduction in payment.

In mid July, we became concerned with the projected cost of the frame. Andy (senior engineer) at the museum had specified a minimum requirement for the strength of the frame which meant the total cost of materials (including labour) increased to £1400. The museum agreed to increase their payment to £3600, and RS agreed to increase their material budget to £1500.

The project was completed on time with a small surplus of £170, which will be used for ongoing maintenance costs.

Description	Date	Category	Cost (£)	Cost (€)
beaglebone black	12/05/2016	Non RS materials	60,00£	69,00 €
stepper drivers drv8825	12/05/2016	Non RS materials	20,00£	23,00 €
pulleys, tap & die, optos	12/05/2016	Non RS materials	100,00£	115,00 €
dyneema string	18/05/2016	Non RS materials	20,00£	23,00 €
Arturo - pcb design	18/05/2016	Time	150,00£	172,50 €
crimp tool, optos, loctite, grub screws, emerg stop	19/05/2016	RS materials	100,00£	115,00 €
wood for full size prototype	22/05/2016	Non RS materials	50,00£	57,50 €
charger/receiver prototype pcbs	25/05/2016	Non RS materials	18,00£	20,70 €
components for charger, strip tools, end mills, ally	01/06/2016	RS materials	220,00£	253,00 €
psu, motors, bbbbbb, delrin	16/06/2016	RS materials	330,00£	379,50 €
motedis order for leadscrews, nuts & bearings	22/06/2016	Non RS materials	63,48£	73,00 €
nuts & bolts	22/06/2016	Non RS materials	73,00£	83,95 €
pcb gondola	22/06/2016	Non RS materials	21,00£	24,15 €
bbb pcb	25/06/2016	Non RS materials	60,00£	69,00 €
paper & pens	01/07/2016	Non RS materials	48,54£	55,82 €

aluminium saw blade	01/07/2016	Non RS materials	28,54£	32,82 €
return flights to UK	01/07/2016	Expenses	230,00£	264,50 €
lathams order wood for frame	04/07/2016	Non RS materials	213,60£	245,64 €
plastock order plastic for frame	05/07/2016	Non RS materials	266,64£	306,64 €
gondola, wire and PCB parts	06/07/2016	RS materials	140,00£	161,00 €
Nic - frame build & installation 2 days	18/07/2016	Time	300,00£	345,00 €
Jon - installation & maintenance 1 day	18/07/2016	Time	150,00£	172,50 €
frame order	20/07/2016	RS materials	618,00£	710,70 €
Guillem - design, prototyping, build - 30 days	01/08/2016	Time	304,35£	350,00 €
Matt - project lead - 30 days	01/08/2016	Time	1.200,00£	1.380,00 €
Matt's UK expenses	01/08/2016	Expenses	150,00£	172,50 €

Table 3 - Budget components by date

Financial summary	Cost (£)	Cost (€)
Non RS materials	942,80£	1.084,22 €
RS materials	1.508,00£	1.734,20 €
Time	2.104,35£	2.420,00 €
Expenses	380,00£	437,00 €
Total without RS support	4.935,15£	5.675,42 €
Total including RS support	3.427,15£	3.941,22 €
@bristol payment	3.600,00£	4.140,00 €
Project surplus	172,85£	198,78 €

Table 4 - Financial summary

Budget distribution (%) of total

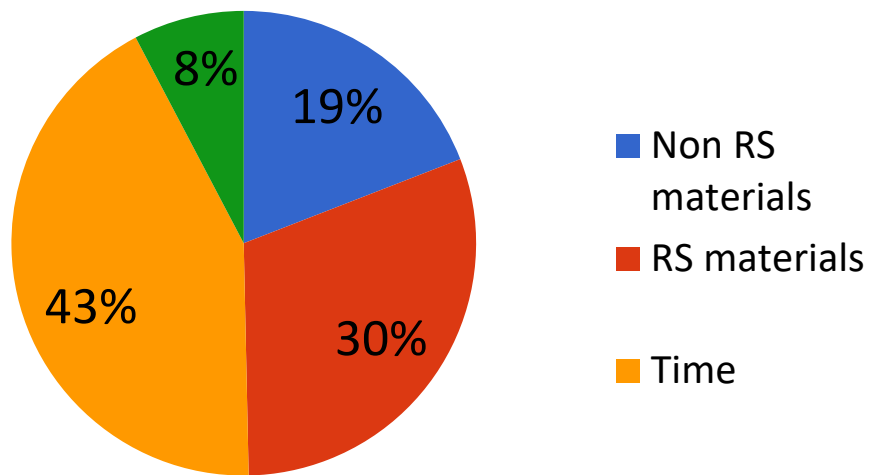


Figure 49 - Budget distribution by category

Estimated budget for second unit		
Materials	2.450,80£	2.818,42 €
Time	1.354,35£	1.557,50 €
Expenses	380,00£	437,00 €
Total	4.185,15£	4.812,92 €

Figure 50 - Proposed budget for second unit of the robot

6. Review and future improvements

Some aspects of the robot can be improved. It is also useful to log the problems found in the development process to avoid them in the future.

From the technical point of view:

PCB

- Places without silkscreen for polarity, labelling, sockets have been found on several boards over the assembly process. They shall be fixed in next revision.
- Late identification of covered pads due to faulty gerbers produced by kicad. Power connectors of BBBBPCB shall be reviewed.
- Mounting holes were asymmetrical requiring generation of a CAM model for drilling the supports properly.
- SMD component missing in the order and require adjustment of values at assembly time.
- Charger and receiver PCBs use non 1206 SMD components and present design fault of the traces. These boards shall be redesigned.

Electronics

- Charger circuit is a critical element of the system, shall be considered use a higher quality part.
- The gondola shall be able to disable the radio for following revisions of the robot.
- Increase the charge rate of the battery.
- Include protection measures for interface big LED 7 segment displays with BeagleBoneBlack and never connect or disconnect with the system running.

Mechanic design

- The mechanics of the charging dock require too much maintenance. More time shall be invested with the inductive charging or redesign the mechanical system for allow reliable contact without conductive grease.

From the management point of view:

Timing

- PCBs delivery time was underestimated and delay the testing.
- The decision between use motors or servos was taken too late and reduce the design and test time. Numerical valuation methods shall be used in situations where it is difficult to identify a clear solution.
- Outsource CNC machining or form more employees to avoid bottleneck between design and test.

Communication

- Specifications for the BBBBBB PCB that was manufactured locally were not strictly defined and only visual inspection was performed after payment. Now on specifications will be provided by mail to put on the record the details and a better inspection procedure shall be designed.
- Plans of the acrylic parts shall be better defined for next version.

7. Conclusions

This project has been a challenge for the capability of integrate different branches of engineering, different tools and pass over all the stages from the idea to the final product with a small team involved in each one of the stages.

Robot was finish on time and fulfils our internal specifications and the specifications of the client, at the same time open a new way of control V-plotters. The project is open source, and as we made with Machinekit developers can continue including functionalities to their V-plotter using our code.

The sponsorship was also a success, RS Components increase the initial amount from 1000 to 1500 pounds during the project and are starting other collaborations in the following months.

Actually we are building a new prototype for continue with the development, optimizing the functionality and including new features as a spray actuator, better trajectory planning, and calibration through image processing.

Even if this project aims to show the artistic uses of the robot we hope that in the future they can be adapted to clean large surfaces of difficult access as windows of skyscrapers with lower implementation cost that any other solution in the actual market.

8. Bibliography

1. Franke, Jürg Lehni & Uli. *juerglehni.com*. [Online] 2002. <http://juerglehni.com/works/hektor>.
2. Kickstarter. [Online] 2 2010. <https://www.kickstarter.com/projects/notever/the-drawing-machine>.
3. Noble, Sandy. *Polargraph.co.uk*. [Online] 2010. <http://www.polargraph.co.uk/>.
4. Nave, R. *hyperphysics.phy*. [Online] 3 2012. <http://hyperphysics.phy-astr.gsu.edu/hbase/fcab.html>.
5. *linuxcnc.org*. [Online] 2016. <http://linuxcnc.org/>.
6. *machinekit.io*. [Online] 2016. www.machinekit.io.
7. *linuxcnc.org*. [Online] 2016. http://linuxcnc.org/docs/2.5/html/common/System_Requirements.html#_minimum_requirements.
8. *machinekit.io*. [Online] 2016. http://www.machinekit.io/docs/common/System_Requirements/.
9. *elinux.org*. [Online] 2016. http://elinux.org/Beagleboard:BeagleBoneBlack_Debian#BBW.2FBBB_.28All_Revs.29_Machinekit.
10. *machinekit.io*. [Online] 2016. http://blog.machinekit.io/p/machinekit_16.html.
11. *xenomai.org*. [Online] 2016. <https://xenomai.org/>.
12. *putty.org*. [Online] www.putty.org.
13. *Straightrunning.com*. [Online] 2016. <http://www.straightrunning.com/XmingNotes/>.
14. *winscp.net*. [Online] 2016. <https://winscp.net/eng/download.php>.
15. *sma.de*. [Online] 2013. <http://files.sma.de/dl/4253/SWebBox-BA-US-en-34.pdf>.
16. *cursivedata.co.uk*. [Online] <http://cursivedata.co.uk/>.
17. *freeboard.io*. [Online] 2013. <https://freeboard.io/>.
18. Freeboard Cursivedata. [Online] 2016. <http://freeboard.cursivedata.co.uk/index.html?load=dashboard.json>.
19. *inkscape.org*. [Online] <https://inkscape.org/es/>.
20. *en.wikipedia.org*. [Online] 2016. https://en.wikipedia.org/wiki/Ultra-high-molecular-weight_polyethylene.
21. Collins, Danielle. *motioncontroltips.com*. [Online] 13 1 2016. <http://www.motioncontroltips.com/faq-whats-the-difference-between-detent-torque-and-holding-torque/>.

22. Halfacree, Gareth. *www.flickr.com*. [Online] 2016. <https://www.flickr.com/photos/120586634@N05/>.
23. Venn, Matthew. *github.com*. [Online] 2014. <https://github.com/mattvenn/big-led-driver>.
24. *linear.com*. [Online] 2016. Revision F. <http://cds.linear.com/docs/en/datasheet/4120ff.pdf>.
25. *linear.com*. [Online] 2016. <http://www.linear.com/docs/45434>.
26. *www.mgchemicals.com*. [Online] 2015. <http://www.mgchemicals.com/downloads/msds/01%20English%20UK%20SDS/sds-8463%20en%20uk.pdf>.

APPENDIX I, QUICK SETUP USER MANUAL

User manual and maintenance operations

Rev.27_07_2016

General operations

- Power off
 1. Open cabinet
 2. Press silver button
 3. Wait till all blue lights on the BeagleBone go out
 4. Press the external power button
 5. All lights inside the cabinet should be off

- Power up
 1. Press external power button
 2. Open cabinet
 3. Wait until big digits light up with current energy use
 4. Press the silver button - robot will start homing
 5. Be ready to press the emergency stop button if there is a problem

Scheduled maintenance

- Change pen
 - Power off the robot
 - Remove the pencil and the magnetic support together
 - Remove the pen
 - Place the support in the socket of the calibration tool
 - Place the new pen, and set the length. Tip shall make contact with the limit screw
 - Turn the adjustment screw until pen is centered and firmly attached
 - Place back the support and the assembly on the gondola

- Advance paper
 - Open the cabinet
 - Remove the clips that are holding the paper
 - Attach the stick that is located at the interior of the cabinet to the handle of the paper roll mechanism
 - Pull the paper carefully towards you.
 - Replace the clips
 - Remove stick and store in the cabinet

- Paper change
 - Tools:
 - 2 x 13mm spanners
 - Steps
 - 2 people
 - Paper release stick
 - Supply information on paper roll: <https://www.prizmagraphics.co.uk/uncoated-plotter-paper/plotter-paper-54quot-1372mm-x-50m-90gsm-roll>
 - Expect to be replaced twice a year
- Other tasks
 - Twice a year greasing the lead screws. Might be nice at the paper change.

Troubleshooting

- Robot is drawing in the wrong place
 - Shutdown and turn on the robot. If there still a problem, call Local maintenance responsible.
- Robot won't start up or shut down
 - Call Remote maintenance responsible.
- Robot is unexpectedly off
 - If all the blue lights are off, but the green lights are on, disconnect main power connector located at the back side and try to start up like normal.
 - Otherwise contact local maintenance responsible.
- Energy robot dashboard
 - Useful for monitoring the battery level of the gondola and other parameters. <http://freeboard.cursivedata.co.uk/index.html?load=dashboard.json>
- Gondola LEDs
 - RED LED flickering shows communication with computer system.
 - BLUE LED light & RED LED off = Charging battery
 - RED LED on = Draw mode

Other Resources

- Educational Video
 - The day to day activities. http://www.youtube.com/watch?v=W1kAjR_QbOw

- Contact details of Maintenance Support:
 - Jon Pillai, Local maintenance: jon.pillai@gmail.com 07825661773
 - Matthew Venn, Remote maintenance: matt@mattvenn.net 01290 211 185

APPENDIX II, MATLAB CODE

[tcalc.m](#)

```
%This funtion provide the tension of two strings fixed at points
%p1(0,Ymax) and p2(xmax,ymax)holding the weigth of an object at any point
%of the 2D space.

%(left)Wirep1: T1,alfa
%(right)Wirep2: T2,beta

%introduce weigth in grams and dimensions in centimeters
function [t1, t2] = tcalc(w,xmax,ymax,xpos,ypos)

f=w/1000*9.80665;

alfa=atan((ymax-ypos)/xpos);
beta=atan((ymax-ypos)/(xmax-xpos));

t1=f/(sin(alfa)+(cos(alfa)/cos(beta))*sin(beta));

t2=t1*(cos(alfa)/cos(beta));

end

tcalc.m%This funtion return two matrix of 1 cm resolution coresponding to
%the tensions of two strings fixed at points
%p1(0,Ymax) and p2(xmax,ymax)holding the weigth of an object at any point
%of the 2D space.

%leftside-> Wirep1: T1,gamma,t1values
%rightside->Wirep2: T2,teta,t2values

%introduce weigth in grams and dimensions in centimeters
function [t1values , t2values]= tmatr(w,xmax,ymax)

t1values= zeros(xmax-1,ymax-1);

t2values= zeros(xmax-1,ymax-1);

for i = (ymax-1):-1:1

    for j=1:(xmax-1)

        [a,b]=tcalc(w,xmax,ymax,j,i);
```



```

        t1values(i,j)=a;

        t2values(i,j)=b;

    end

end

end

```

[tpmap.m](#)

```

%This funtion provide a map with a resolution of 1 unit
%with the maximum tensions of two strings fixed at points
%p1(0,Ymax) and p2(xmax,ymax)holding the weigth of an object at any point
%of the 2D space.
%maximum tensions matrix is returned.
%Wirep1: T1,gamma
%Wirep2: T2,teta
%introduce weigth in grams and dimensions in centimeters
%Dependence of

%http://www.mathworks.com/matlabcentral/fileexchange/24253-customizable-heat-
maps/content/html/heatmap\_examples.html

function [tmax]= tpmap(w,xmax,ymax)

tmax= zeros(ymax-1,xmax-1);

%calculate values for two tensions

[t1values , t2values]= tmatr(w,xmax,ymax);

%get maximum and reverse yaxis

for i = 1:(ymax-1)

    for j=1:(xmax-1)

        a= t1values(i,j);

        b=t2values(i,j);

        tmax(ymax-i,j)= max([a b]);

```

```

end

end

%plot maximum

xlab=1:1:xmax;

ylab=ymax:-1:1;

clf

heatmap(tmax, xlab, ylab, '%0.2f', 'Colormap', 'red',...

    'Colorbar', true, 'ColorLevels', 100, 'MaxColorValue', w/1000*9.80665*1.5);

%heatmap(t1values, [], [], '%0.2f', 'Colormap', 'money',...

%    'Colorbar', true, 'ColorLevels', 5);

% heatmap(t2values, [], [], '%0.2f', 'Colormap', 'money',...

%    'Colorbar', true, 'ColorLevels', 5);

end

```

[valuesarea.m](#)

```

function [ maximum, minimum, average] = valuesarea( xmax, ymax,w,p )

%valuesarea provide max min and avg tension of working area

% The working area is centered at the middle of the x and y dimensions

% and is reduced equally in each axis.

% Weigth shall be introduced in grams

% p represent the percentaje of working area used. P=1 means no

% reduction, p=0.7 is equivalent to 70% of the area and a reduction of 30%.

[t1,t2]=tmatr(w,xmax,ymax);

tmax= zeros(ymax-1,xmax-1);

tmin= zeros(ymax-1,xmax-1);

%get maximum,minimum between strings

for i = 1:(ymax-1)

    for j=1:(xmax-1)

```

```

    a= t1(i,j);

    b=t2(i,j);

    tmax(i,j)= max([a b]);

    tmin(i,j)=min([a b]);

    end

end

%values for workspace with equal reduction at x and y

redx=floor(xmax*(1-p)/2);

redy=floor(ymax*(1-p)/2);

x1=0+redx;

x2=xmax-redx;

y1=0+redy;

y2=ymax-redy;

format bank;

maximum=max(max(tmax(y1:y2,x1:x2)))

minimum=min(min(tmin(y1:y2,x1:x2)))

average=mean(mean([t1(y1:y2,x1:x2) t2(y1:y2,x1:x2)]))

end

```

[importjpg.m](#)

```

%Load a picture and plot print with robot

clear

%load the picture

I = imread('square_circle_tringle_100x100.png');

[length,width,~]=size(I);

% robot characteristics

xmax=width;

ymax=length;

```

```

drod=50;

steps=160;

homer1=2;

homer2=-2;

steperr=0;

%color range 255 white 0 black

black=20;

k=1;

for i=1:length

%arriba abajo

    for j=1:width

        %izquierda derecha

            if(I(i,j)>black)

                I(i,j)=0;

            else

                I(i,j)=1;

                pos(1,k)=j;

                pos(2,k)=i;

                k=k+1;

            end

        end

    end

end

possim1(xmax,ymax,drod,steps,homer1,homer2,steperr,pos)

```

[inverseposition.m](#)

```

function [ l1,l2] =inverseposition(xmax,ymax,x,y)

%inverseposition return the robot configuration

% For this procedure is used sin theorem

```

```

base=xmax;

alf=atan((ymax-y)/x);

bet=atan((ymax-y)/(xmax-x));

gam=pi-bet-alf;

a=base;

b=(a/sin(gam))*sin(alf);

c=(a/sin(gam))*sin(bet);

l1=c;

l2=b;

end

```

Maximum position errors.m

```

%error simulation of the given area

%area=0.75;

%xmax=2400;

%ymax=1800;

%drod=55;

%steps=800;

function [maxerrquant,maxerrx,maxerry,maxertotal]=
Maximum_position_errors(xmax,ymax,drod,steps,homer1,homer2,area)

a=1-area;

b=area;

c=0.5;

pos=[a*xmax a*xmax c*xmax c*xmax c*xmax b*xmax b*xmax;a*ymax b*ymax c*ymax
a*ymax b*ymax a*ymax b*ymax];

posquant=pos;

poshomer=pos;

%possiml(xmax,ymax,drod,steps,homer1,homer2,0,pos)

%Introducir xmax, ymax,drod,steps

```

```

%Introducir homing error

%Introducir matriz de puntos

    %pos = [x0 x1 x2 ... xn; y0 y1 y2 ... yn]

    %pos=

    %      x0 x1 x2 .... xn
    %      y0 y1 y2 .... yn

%Obtener puntos caso ideal

%Obtener puntos cuantización

%Obtener puntos cuantización mas homing error and step error

[minrope,lstep,~,~]= paramlcalc(xmax,ymax,drod,steps);

%lengths in ideal case but quantized

l1=(0:lstep:minrope);

l2=l1;

%homing error

l1err=(homer1:lstep:minrope+homer1);

l2err=(homer2:lstep:minrope+homer2);

%blue - ideal points

%quantized position and error positions

[~,width]=size(pos);

for i=1:width

[ la1,la2] =inverseposition(xmax,ymax,pos(1,i),pos(2,i));

    f= l1;

    val = la1; %value to find

    tmp = abs(f-val);

    [~, idx1] = min(tmp); %index of closest value

    l1temp = f(idx1); %closest value

    val = la2; %value to find

    tmp = abs(f-val);

```

```

[~, idx2] = min(tmp); %index of closest value

l2temp = f(idx2); %closest value

% quantized lengths

[posquant(1,i),posquant(2,i)]=positionxy(xmax,ymax,l1temp,l2temp);

%black - plot of quantized lengths + errors

l1temp=l1err(idx1);

l2temp=l2err(idx2);

[poshomer(1,i),poshomer(2,i)]=positionxy(xmax,ymax,l1temp,l2temp);

end

%display(pos)

%display(posquant)

%display(poshomer)

for i=1:width

    %error por cuantización

    errorquant(1,i)=pos(1,i)-posquant(1,i);

    errorquant(2,i)=pos(2,i)-posquant(2,i);

    %error por posicionamiento en coordenadas x e y

    errorhom(1,i)=pos(1,i)-poshomer(1,i);

    errorhom(2,i)=pos(2,i)-poshomer(2,i);

    errortot(i)=sqrt(errorhom(1,i)^2+errorhom(2,i)^2);

end

%display(area);display(homer1);display(homer2);

maxerrquant=max(max(abs(errorquant)));

maxerrx=max(abs(errorhom(1,1:width)));

maxerry=max(abs(errorhom(2,1:width)));

maxertotal=max(errortot);

%center right

```

```
%center down
```

```
end
```

[param1calc.m](#)

```
%this function return the minimum rope required for each side  
  
%the equivalent in linear length of each step and the number of turns  
required for roll all the rope.  
  
%minrope,xmax,lstep, ymax in centimeters  
  
%drod and error in mm  
  
%astep in rad  
  
function [minrope,lstep,astep,nturns]= param1calc(xmax,ymax,drod,steps)  
  
%minimum amount of rope hanging from motor  
  
minrope = sqrt(xmax^2+ymax^2);  
  
%angle of a step rad  
  
astep= ((2*pi)/steps);  
  
%arc rope length. This limitate the resolution  
  
lstep=((drod/2)*astep)/10;  
  
%number of turns required for minrope  
  
nturns= (minrope*10)/(2*pi*(drod/2));  
  
end
```

[posangles.m](#)

```
function [ x,y,al,be,ga] =posangles(xmax,ymax,l1,l2)  
  
%Positionxy return the exact position given the lengths of the triangle  
  
% For this procedure is used cosine theorem  
  
if (l1+l2)<base
```



```

    display('Error input L1,L2 shall be greater than base')

else

    a=base;

    b=l1;

    c=l2;

    syms alf bet gam;

    % a^2=b^2+c^2-2*b*c*cos(alfa);

    %b^2=a^2+c^2-2*a*c*cos(beta);

    %c^2=a^2+b^2-2*a*b*cos(gamma);

    res=solve([a^2==b^2+c^2-2*b*c*cos(alf),b^2==a^2+c^2-
2*a*c*cos(bet),alf+bet+gam==pi,alf>0,bet>0,gam>0],[alf,bet,gam]);

    al=res.alf;

    be=res.bet;

    ga=res.gam;

end

[x,y]=positionxy(xmax,ymax,l1,l2);

end

```

[positionxy.m](#)

```

function [x,y] =positionxy(xmax,ymax,l1,l2)

%Positionxy return the exact position given the base coordinates and
%lengths of wires

% For this procedure is used cosine theorem

base=xmax;

if (l1+l2)<base

    display('Error input L1,L2 shall be greater than base')

else

    a=base;

```

```

b=l1;

c=l2;

t=0.5*sqrt((a+b-c)*(a-b+c)*(-a+b+c)*(a+b+c));

y=t/a;

alfa=asin(y/l1);

x=cos(alfa)*b;

y=ymax-y;

%Debugg plot

%hold on;

%plot(0,0,'d')

%plot(base,0,'d')

%plot(x,-y,'d')

%hold off;

end

end

```

[posmap.m](#)

```

%this function provide a map of the 3d space for isosceles config

%minrope,xmax, ymax in centimeters

%lstep,drod and error in mm

%all angles in rad

function [y]= posmap(xmax,ymax,drod,steps,error)

lbase=xmax/2;

[minrope,lstep,astep,nturns]= paramlcalc(xmax,ymax,drod,steps);

axis([0,xmax,0,ymax])

ql1=(0:lstep:minrope);

y=zeros(1,length(ql1));

%calcular posibles alturas

```

```

for i=1:length(q11)

    a=xmax/2;

    c=q11(1,i);

    y(1,i)=sqrt((c^2)-(a^2));

    %debido a que el origen de cordenadas se encuentra en la esquina
    %inferior izquierda lo trasladamos al sistema global

    y(1,i)=ymax-y(1,i);

    hold on;

    plot(lbase/2,y(1,i),'d')

    if (y(1,i)<0)

        break;

    end

end

hold off;

end

```

[possim1.m](#)

```

function []= possim1(xmax,ymax,drod,steps,homer1,homer2,steperr,pos)

%Introducir xmax, ymax,drod,steps

%Introducir homing error

%Introducir matriz de puntos

    %pos = [x0 x1 x2 ... xn; y0 y1 y2 ... yn]

    %pos=

    %    x0 x1 x2 .... xn

    %    y0 y1 y2 .... yn

%Obtener puntos caso ideal

%Obtener puntos cuantización

%Obtener puntos cuantización mas homing error and step error

```

```

[minrope,lstep,~,~]= paramlcalc(xmax,ymax,drod,steps);

%lengths in ideal case but quantized

l1=(0:lstep:minrope);

l2=l1;

%step error vector

ea=steperr;

eb=-steperr;

steperr = (eb-ea).*rand(1,length(l1)) + ea;

%homing error

l1err=(homer1:lstep:minrope+homer1);

l2err=(homer2:lstep:minrope+homer2);

%total error

l1err=l1err+steperr;

l2err=l2err+steperr;

%plots code

hold on;

xlabel('x')

ylabel('y')

axis([0 xmax -0.25*ymax ymax])

%blue - ideal points

[~,width]=size(pos);

for i=1:width

ideal=plot(pos(1,i),pos(2,i),'bd');

end

%quantized position and error positions

[~,width]=size(pos);

for i=1:width

[ la1,la2] =inverseposition(xmax,ymax,pos(1,i),pos(2,i));

```

```

f= l1;

val = la1; %value to find

tmp = abs(f-val);

[~, idx1] = min(tmp); %index of closest value

l1temp = f(idx1); %closest value

val = la2; %value to find

tmp = abs(f-val);

[~, idx2] = min(tmp); %index of closest value

l2temp = f(idx2); %closest value

%red - plot of quantized lengths

[xtemp,ytemp]=positionxy(xmax,ymax,l1temp,l2temp);

quant=plot(xtemp,ytemp,'r^');

%black - plot of quantized lengths + errors

l1temp=l1err(idx1);

l2temp=l2err(idx2);

[xtemp,ytemp]=positionxy(xmax,ymax,l1temp,l2temp);

equant=plot(xtemp,ytemp,'k*');

end

legend([ideal,quant,equant], 'ideal', 'quantized', 'total')

hold off;

end

```

[trianglexy.m](#)

```

function [x,y] =trianglexy(base,l1,l2)

%Positionxy return the exact position given the lengths of the triangle

% For this procedure is used cosine theorem

% be aware when is used to simulate the robot the base is NOT located at

% x=0,y=0 !

```

```

if (l1+l2)<base

    display('Error input L1 + L2 shall be greater than base')

else

    a=base;

    b=l1;

    c=l2;

    t=0.5*sqrt((a+b-c)*(a-b+c)*(-a+b+c)*(a+b+c));

    y=t/a;

    alfa=asin(y/l1);

    x=cos(alfa)*b;

    %Debugg plot

    %hold on;

    %plot(0,0,'d')

    %plot(base,0,'d')

    %plot(x,y,'d')

    %hold off;

end

end

```

[quadtcalc.m](#)

```

%This funtion provide the tension:

%Of FOUR strings

%two strings fixed between points

%p1(0,Ymax) and p2(xmax,ymax) and center of the basquet

%two strings fixed between points

%center of the basket and weights at corners

%both wights are equal. The total stabilizers weigth is wstab

%(left-up)Wirepl: T1,alfa

```

```

%(right-up)Wirep2: T2,beta

%(left-down)Wirep3: T3,gamma,wstab

%(right-down)Wirep4: T4,theta,wstab

%introduce weigth in grams and dimensions in centimeters

%revisar expresiones modelo

function [t1, t2,t3,t4] = quadtcalc(wbasket,wstab,xmax,ymax,xpos,ypos)

ftot=(wbasket+wstab)/1000*9.80665;

fstab=wstab/1000*9.80665;

alfa=atan((ymax-ypos)/xpos);

beta=atan((ymax-ypos)/(xmax-xpos));

t1=ftot/(sin(alfa)+(cos(alfa)/cos(beta))*sin(beta));

t2=t1*(cos(alfa)/cos(beta));

gamma=atan(ypos/xpos);

theta=atan(ypos/(xmax-xpos));

t3=(fstab/2);

t4=(fstab/2);

%t3=fstab/(sin(gamma)+(cos(gamma)/cos(theta))*sin(theta));

%t4=t3*(cos(theta)/cos(theta));

end

```

[quadmatr.m](#)

```

%This funtion return two matrix of 1 unit resolution coresponding to tensions:

%Of FOUR strings

%two strings fixed between points

%p1(0,Ymax) and p2(xmax,ymax) and center of the basquet

%two strings fixed between points

%center of the basket and weights at lower corners

%both wigths are equal. The total stabilizers weigth is wstab

%(left-up)Wirep1: T1,alfa

```

```

%(right-up)Wirep2: T2,beta
%(left-down)Wirep3: T3,gamma,wstab
%(right-down)Wirep4: T4,theta,wstab

%introduce weigth in grams and dimensions in centimeters

function [t1values , t2values,t3values,t4values]= quadtmatr(w,wstab,xmax,ymax)

t1values= zeros(xmax-1,ymax-1);
t2values= zeros(xmax-1,ymax-1);
t3values= zeros(xmax-1,ymax-1);
t4values= zeros(xmax-1,ymax-1);

for i = (ymax-1):-1:1
    for j=1:(xmax-1)
        [a,b,c,d]=quadtcalc(w,wstab,xmax,ymax,j,i);

        t1values(i,j)=a;
        t2values(i,j)=b;
        t3values(i,j)=c;
        t4values(i,j)=d;
    end
end
end
end

```

[quadtotalpmap.m](#)

```

%This funtion provide a map with a resolution of 1 unit
%with the total tension of:
%Of FOUR strings
%two strings fixed between points
%p1(0,Ymax) and p2(xmax,ymax) and center of the basquet
%two strings fixed between points
%center of the basket and weights at lower corners
%both wigths are equal. The total stabilizers weigth is wstab

```



```

%(left-up)Wirep1: T1,alfa
%(right-up)Wirep2: T2,beta
%(left-down)Wirep3: T3,gamma,wstab
%(right-down)Wirep4: T4,theta,wstab

%introduce weigth in grams and dimensions in centimeters

%Dependence of

%http://www.mathworks.com/matlabcentral/fileexchange/24253-customizable-heat-
maps/content/html/heatmap\_examples.html

function [tmax]= quadttotaltpmap(w,wstab,xmax,ymax)

ttotal= zeros(ymax-1,xmax-1);

%calculate values for tensions

[t1values , t2values,t3values,t4values]= quadtmatr(w,wstab,xmax,ymax);

%get maximum and reverse yaxis

for i = 1:(ymax-1)

    for j=1:(xmax-1)

        a= t1values(i,j);

        b=t2values(i,j);

        c=t3values(i,j);

        d=t4values(i,j);

        ttotal(ymax-i,j)= a+b+c+d;

    end

end

%plot maximum

xlab=1:1:xmax;

ylab=ymax:-1:1;

clf

heatmap(ttotal, xlab, ylab, '%0.2f', 'Colormap', 'red',...

    'Colorbar', true, 'ColorLevels',

    100,'MaxColorValue', (w+wstab)/1000*9.80665*2.5);

```

```

heatmap(t1values, [], [], '%0.2f', 'Colormap', 'money',...
%   'Colorbar', true, 'ColorLevels', 5);

% heatmap(t2values, [], [], '%0.2f', 'Colormap', 'money',...
%   'Colorbar', true, 'ColorLevels', 5);

end

```

[quadtpmap.m](#)

```

%This funtion provide a map with a resolution of 1 unit

%with the maximum tensions of:

%Of FOUR strings

%two strings fixed between points

%p1(0,Ymax) and p2(xmax,ymax) and center of the basquet

%two strings fixed between points

%center of the basket and weights at lower corners

%both wigths are equal. The total stabilizers weigth is wstab

%(left-up)Wirep1: T1,alfa

%(right-up)Wirep2: T2,beta

%(left-down)Wirep3: T3,gamma,wstab

%(right-down)Wirep4: T4,theta,wstab

%introduce weigth in grams and dimensions in centimeters

%Dependence of

%http://www.mathworks.com/matlabcentral/fileexchange/24253-customizable-heat-
maps/content/html/heatmap\_examples.html

function [tmax]= quadtpmap(w,wstab,xmax,ymax)

tmax= zeros(ymax-1,xmax-1);

%calculate values for tensions

[t1values , t2values,t3values,t4values]= quadtmatr(w,wstab,xmax,ymax);

%get maximum and reverse yaxis

for i = 1:(ymax-1)

```

```

for j=1:(xmax-1)

    a= t1values(i,j);

    b=t2values(i,j);

    c=t3values(i,j);

    d=t4values(i,j);

    tmax(ymax-i,j)= max([a b c d]);

end

end

%plot maximum

xlab=1:1:xmax;

ylab=ymax:-1:1;

clf

heatmap(tmax, xlab, ylab, '%0.2f', 'Colormap', 'red',...

        'Colorbar', true, 'ColorLevels',
100, 'MaxColorValue', (w+wstab)/1000*9.80665*1.5);

%heatmap(t1values, [], [], '%0.2f', 'Colormap', 'money',...

%   'Colorbar', true, 'ColorLevels', 5);

% heatmap(t2values, [], [], '%0.2f', 'Colormap', 'money',...

%   'Colorbar', true, 'ColorLevels', 5);

end

```

[quadvaluesarea.m](#)

```

function [ maximum, minimum, average,minmot,avgmot] = quadvaluesarea( xmax,
ymax,w,wstab,p )

%valuesarea provide max min and avg tension of working area

%   The working area is centered at the middle of the x and y dimensions

%   and is reduced equally in each axis.

%   Weights shall be introduced in grams

%   p represent the percentaje of working area used. P=1 means no

%   reduction, p=0.7 is equivalent to 70% of the area and a reduction of 30%.

```

```

[t1,t2,t3,t4]=quadtmatr(w,wstab,xmax,ymax);

tmax= zeros(ymax-1,xmax-1);

tmin= zeros(ymax-1,xmax-1);

%get maximum,minimum between strings

for i = 1:(ymax-1)

    for j=1:(xmax-1)

        a= t1(i,j);

        b=t2(i,j);

        c=t3(i,j);

        d=t4(i,j);

        tmax(i,j)= max([a b c d]);

        tmin(i,j)=min([a b c d]);

    end

end

%values for workspace with equal reduction at x and y

redx=floor(xmax*(1-p)/2);

redy=floor(ymax*(1-p)/2);

x1=0+redx;

x2=xmax-redx;

y1=0+redy;

y2=ymax-redy;

format bank;

maximum=max(max(tmax(y1:y2,x1:x2)))

minimum=min(min(tmin(y1:y2,x1:x2)))

average=mean(mean([t1(y1:y2,x1:x2) t2(y1:y2,x1:x2) t3(y1:y2,x1:x2) t4(y1:y2,
x1:x2)]))

%regarding the motor

[ maximum, minmot, avgmot] = valuesarea( xmax, ymax,w+wsstab,p );

```

```
display(minmot)

display(avgmot)

end
```

[temporalvariables.m](#)

```
area=0.75;

xmax=2400;

ymax=1800;

drod=55;

steps=800;

homer1=0.8;

homer2=0.8;

[a, b, c, d]=Maximum_position_errors(xmax,ymax,drod,steps,homer1,homer2,area)
```

[Forcescomparation.m](#)

```
w=500

ws=0

x=240

y=180

t= zeros(3,5);

i=1;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.9);

i=2;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.75);

i=3;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.5);

display(w);

display(ws);

display(t);
```

```

w=1000

t= zeros(3,5);

i=1;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.9);

i=2;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.75);

i=3;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.5);

display(w);

display(ws);

display(t);

w=1500;

t= zeros(3,5);

i=1;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.9);

i=2;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.75);

i=3;

[t(i,1) t(i,2) t(i,3) t(i,4) t(i,5)]=quadvaluesarea(x,y,w,ws,0.5);

display(w);

display(ws);

display(t);

```

APPENDIX III, RESUME OF SIMULATIONS AND DESIGN PARAMETERS

Appendix III, Resume of simulations and design parameters

Design Parameters	
Rod diameter (mm)	35
X (mm)	2400
Y (mm)	1800
Wire diameter (mm)	1,20
Steps per revolution	200
Microstepping	4
Microsteps per revolution	800
Increase of torque (%)	0,38
Gear ratio between motor & rod	0,56
Step frequency	2000
Displacement per step (mm)	0,08
Maximum speed (mm/s)	76,36

area	Rope length	n°turns	Rod length
0,90	2700,00	24,56	29,47
0,75	2250,00	20,46	24,56
0,50	1500,00	13,64	16,37

hypotenuse	error (mm)
1800,24	0,24
1800,17	0,17
1800,07	0,07

Spool error due to diagonal produced						
x0	x1	y0	y1	width		highth
				a	b	c
120	2280	90	1710	2160	1620	2700
300	2100	225	1575	1800	1350	2250
600	1800	450	1350	1200	900	1500

Configuration									
Dual 240 x 180									
Gondola Weight (g)	500			1000			1500		
Working area (%)	0,90	0,75	0,5%	0,90	0,75	0,5	0,90	0,75	0,5
Maximum tension (N)	36,86	13,60	7,01	73,71	27,20	14,02	110,58	40,80	21,03
Mimumum tension (N)	0,37	1,02	2,04	0,75	2,04	4,08	1,11	3,06	6,12
Average tension (N)	5,14	4,48	4,19	10,29	8,96	8,38	15,42	13,44	12,57
Minimum Holding Torque (Ncm)	64,51	23,80	12,27	128,99	47,60	24,54	193,52	71,40	36,80
Holding torque with microstepping (Ncm)	168,55	62,19	32,06	337,06	124,38	64,11	505,66	186,57	96,17
Minimum Holding Torque (Ncm) WITH GEAR	93,64	34,55	17,81	187,26	69,10	35,62	280,92	103,65	53,43

[maximum, minimum, average] = valuesarea(xmax, ymax,w,p)

Appendix III, Resume of simulations and design parameters

Configuration	Quad with wstab of 200g 240x180											
	500			1000			1500			1500		
Gondola Weight (g)	0.90	0.75	0.5	0.90	0.75	0.5	0.90	0.75	0.5	0.90	0.75	0.5
Working area (%)	51,60	19,04	9,82	88,46	32,64	16,84	125,31	46,24	23,85			
Maximum tension (N)	0,52	0,98	0,98	0,90	0,98	0,98	0,98	0,98	0,98			
Minimum tension (N)	4,09	3,63	3,43	6,66	5,87	5,52	9,23	8,11	7,62			
Average tension (N)	0,52	1,43	2,86	0,90	2,45	4,90	1,27	3,47	6,95			
Minimum tension motor strings (N)	7,20	6,27	5,87	12,34	10,76	10,06	17,49	15,24	14,26			
Average tension motor strings (N)	90,30	33,32	17,19	154,81	57,12	29,47	219,29	80,92	41,74			
Minimum Holding Torque (Ncm)												

[maximum, minimum, average,minmot,avgmot] = quadvaluesarea(xmax, ymax,w,wstab,p)

Type of errors & maximum values for 2400x1800 mm *	55					
	0.90		0.75		0.5	
Spool diameter (mm)						
Area (%)						
Steps (n°)	400	800	1600	400	800	1600
Max quantization error(mm)				0,02	0,02	0,01
Max x error (mm)				1,25	1,25	1,25
Max y error (mm)				0,33	0,33	0,33
Max distance error (mm)				1,29	1,29	1,29

[maxerrquant,maxerrx,maxerry,maxertotal]= Maximum_position_errors(xmax,ymax,drod,steps,homer1,homer2,area)

Maximum errors for 2400x1800 mm config.	55					
	0.75 %					
Spool diameter (mm)						
Area (%)						
Homer1	0,80	0,80	0,60	0,60	0,20	0,20
Homer2	-0,80	0,80	-0,60	0,60	-0,20	0,20
Max x error (mm)	1,25	0,36	0,94	0,27	0,32	0,02
Max y error (mm)	0,33	2,30	0,25	1,73	0,08	0,59
Max distance error (mm)	1,29	2,33	0,97	1,75	0,33	0,59

[maxerrquant,maxerrx,maxerry,maxertotal]= Maximum_position_errors(xmax,ymax,drod,steps,homer1,homer2,area)

Appendix III, Resume of simulations and design parameters

Design parameters of interrupter	
leadscrew pitch	2,00
turns	20,46
total interrupter movement	40,93
interrupter length	30,46

Microstepping	Increase of torque (%)
1	100,00
2	70,71
4	38,27
8	19,51

Notes

Quad is the configuration with four strings, two motors and two extra weights.

Dual is the configuration with two strings, two motors.

Wstab is the total extra weight in quad configuration.

Homeerror1 and homeerror2 are the homming errors of axis x and y.

The function under some tables shows the function used for simulate the behaviour and get the data

Microstepping vs increase of torque are approximate and can change between drivers.

* Values with input home error config. Homeer1= 0.8 homeer2=-0.8

APPENDIX IV, MACHINEKIT CODE MODIFICATIONS

Bipod.hal

```
# #####
#
# HAL file for BeagleBone + BeBoPr cape with 4 steppers
#
# Derived from example hm2-stepper config
#
# #####

# Launch the setup script to make sure hardware setup looks good
loadusr -w ./setup.sh
loadusr -W xbee.py 20

# #####
# Core EMC/HAL Loads
# #####

# kinematics
# loadrt trivkins
loadrt bipodkins

setp bipodkins.Bx 2140 # 2200 wide - 60 for the pulleys

# motion controller, get name and thread periods from ini file
# trajectory planner
loadrt tp
loadrt [EMCMOT]EMCMOT servo_period_nsec=[EMCMOT]SERVO_PERIOD
num_joints=[TRAJ]AXES tp=tp kins=bipodkins
#loadrt [EMCMOT]EMCMOT servo_period_nsec=[EMCMOT]SERVO_PERIOD
#loadrt threads name3=xbee-thread period3=5000000 # 50ms, xbee takes max 25ms

# load low-level drivers
#loadrt hal_bb_gpio output_pins=816,822,823,824,825,826,923,925
input_pins=807,808,809,810,817,911,913,922,921
loadrt hal_bb_gpio output_pins=914 input_pins=922,921,911
loadrt [PRUCONF](DRIVER) prucode=$(HAL_RTMOD_DIR)/[PRUCONF](PRUBIN)
[PRUCONF](CONFIG) halname=hpg pru_period=4000
loadrt limit1 count=2

# #####
# THREADS
# #####

addf hpg.capture-position servo-thread
addf bb_gpio.read servo-thread
addf motion-command-handler servo-thread
addf motion-controller servo-thread
addf limit1.0 servo-thread
addf limit1.1 servo-thread
addf hpg.update servo-thread
addf bb_gpio.write servo-thread

#addf xbee xbee-thread

# #####
# Axis-of-motion Specific Configs (not the GUI)
# #####

# #####
# X [0] Axis
# #####
```

```

# axis enable chain
newsig emcmot.00.enable bit
sets emcmot.00.enable FALSE

net emcmot.00.enable <= axis.0.amp-enable-out
net emcmot.00.enable => hpg.stepgen.00.enable

# position command and feedback
net emcmot.00.pos-cmd <= axis.0.motor-pos-cmd
net emcmot.00.pos-cmd => hpg.stepgen.00.position-cmd

net motor.00.pos-fb <= hpg.stepgen.00.position-fb
net motor.00.pos-fb => axis.0.motor-pos-fb

# timing parameters
setp hpg.stepgen.00.dirsetup      [AXIS_0]DIRSETUP
setp hpg.stepgen.00.dirhold      [AXIS_0]DIRHOLD

setp hpg.stepgen.00.steplen      [AXIS_0]STEPLEN
setp hpg.stepgen.00.stepspace    [AXIS_0]STEPSPACE

setp hpg.stepgen.00.position-scale [AXIS_0]SCALE

setp hpg.stepgen.00.maxvel       [AXIS_0]STEPGEN_MAX_VEL
setp hpg.stepgen.00.maxaccel     [AXIS_0]STEPGEN_MAX_ACC

#setp hpg.stepgen.00.step_type      0
setp hpg.stepgen.00.steppin       917
setp hpg.stepgen.00.dirpin        918

# #####
# Y [1] Axis
# #####

# axis enable chain
newsig emcmot.01.enable bit
sets emcmot.01.enable FALSE

net emcmot.01.enable <= axis.1.amp-enable-out
net emcmot.01.enable => hpg.stepgen.01.enable

# position command and feedback
net emcmot.01.pos-cmd <= axis.1.motor-pos-cmd
net emcmot.01.pos-cmd => hpg.stepgen.01.position-cmd

net motor.01.pos-fb <= hpg.stepgen.01.position-fb
net motor.01.pos-fb => axis.1.motor-pos-fb

# timing parameters
setp hpg.stepgen.01.dirsetup      [AXIS_1]DIRSETUP
setp hpg.stepgen.01.dirhold      [AXIS_1]DIRHOLD

setp hpg.stepgen.01.steplen      [AXIS_1]STEPLEN
setp hpg.stepgen.01.stepspace    [AXIS_1]STEPSPACE

setp hpg.stepgen.01.position-scale [AXIS_1]SCALE

setp hpg.stepgen.01.maxvel       [AXIS_1]STEPGEN_MAX_VEL
setp hpg.stepgen.01.maxaccel     [AXIS_1]STEPGEN_MAX_ACC

#setp hpg.stepgen.01.step_type      0
setp hpg.stepgen.01.steppin       915

```

```

setp hpg.stepgen.01.dirpin          916

# #####
# Z [2] Axis
# #####

setp xbee.scale 20
net zpos axis.2.motor-pos-cmd => xbee.pos axis.2.motor-pos-fb

# #####
# Standard I/O - EStop, Enables, Limit Switches, Etc
# #####

# Create estop signal chain
# Drive software estop to hardware
#net estop-out iocontrol.0.user-enable-out => bb_gpio.p8.out-26
#setp bb_gpio.p8.out-26.invert 1

# Monitor estop input from hardware
net estop-loop bb_gpio.p9.in-11 => iocontrol.0.emc-enable-in
#setp bb_gpio.p8.in-17.invert 1

# Axis enable signal (active low)
net emcmot.00.enable => bb_gpio.p9.out-14
#setp bb_gpio.p9.out-14.invert 1

# #####
# Limit Switches
# #####
newsig home-x bit
newsig home-y bit

net home-x <= bb_gpio.p9.in-21
net home-y <= bb_gpio.p9.in-22

# Uncomment if you actually have limit switches setup
# You probably want to setup homing in the INI file, as well
net home-x => axis.0.home-sw-in
net home-y => axis.1.home-sw-in

```

[Bipod.ini](#)

```

[PRUCONF]
DRIVER=hal_pru_generic
CONFIG=pru=0 num_stepgens=3
PRUBIN=xenomai/pru_generic.bin

[EMC]

# Name of machine, for use with display, etc.
MACHINE =          Bipod

# Debug level, 0 means no messages. See src/emc/nml_int/emcglb.h for others
#DEBUG =          0x00000003
DEBUG =          0x00000007
#DEBUG = 0

[DISPLAY]

# Name of display program, e.g., tkemc

```

```

#DISPLAY = axis
DISPLAY = ./bipod.py

# Cycle time, in seconds, that display will sleep between polls
CYCLE_TIME = 0.200

# Path to help file
HELP_FILE = tklinucnc.txt

# Initial display setting for position, RELATIVE or MACHINE
POSITION_OFFSET = RELATIVE

# Initial display setting for position, COMMANDED or ACTUAL
POSITION_FEEDBACK = ACTUAL

# Highest value that will be allowed for feed override, 1.0 = 100%
MAX_FEED_OVERRIDE = 1.5

# Prefix to be used
PROGRAM_PREFIX = /home/machinekit/machinekit/nc_files

# Increments for the JOG section
INCREMENTS = 10 1 0.1 0.01

[TASK]

# Name of task controller program, e.g., milltask
TASK = milltask

# Cycle time, in seconds, that task controller will sleep between polls
CYCLE_TIME = 0.010

[RS274NGC]

# File containing interpreter variables
PARAMETER_FILE = pru-stepper.var

[EMCMOT]

EMCMOT = motmod

# Timeout for comm to emcmot, in seconds
COMM_TIMEOUT = 1.0

# Interval between tries to emcmot, in seconds
COMM_WAIT = 0.010

# Servo task period, in nanoseconds
SERVO_PERIOD = 500000

[HAL]

# The run script first uses halcmd to execute any HALFILE
# files, and then to execute any individual HALCMD commands.

# list of hal config files to run through halcmd
# files are executed in the order in which they appear

HALFILE = bipod.hal

# list of halcmd commands to execute
# commands are executed in the order in which they appear

```

```

#HALCMD =          save neta

[TRAJ]

AXES =              3
COORDINATES =      X Y Z
LINEAR_UNITS =     mm
ANGULAR_UNITS =    degree
CYCLE_TIME =       0.010
DEFAULT_VELOCITY = 100.00
MAX_LINEAR_VELOCITY = 200.00
MAX_ACCELERATION = 400
# lots of joint following errors until I added this:
DEFAULT_ACCELERATION = 200

[AXIS_0]

TYPE =              LINEAR
MAX_VELOCITY =      200
MAX_ACCELERATION =  400
# Set Stepgen max 20% higher than the axis
STEPGEN_MAX_VEL =   300.0
STEPGEN_MAX_ACC =   1200.0

BACKLASH =          0.000

SCALE = -25.46

MIN_LIMIT =         150.0
MAX_LIMIT =         2600.0

FERROR =            1.0
MIN_FERROR =        0.25

HOME =              1500
HOME_OFFSET =       1500
HOME_SEARCH_VEL =   -50.0
HOME_LATCH_VEL =    -10.0
HOME_IGNORE_LIMITS = YES
HOME_USE_INDEX =    NO
HOME_SEQUENCE =     1

# Set to zero if you don't have physical home/limit switches
# Set to the desired homing and latch velocity if you have switches
# See: https://github.com/machinekit/machinekit-docs

# these are in nanoseconds
DIRSETUP =          2000
DIRHOLD =            2000
STEPLEN =           10000
STEPSPACE =         10000

[AXIS_1]

TYPE =              LINEAR
MAX_VELOCITY =      200.0
MAX_ACCELERATION =  400.0
# Set Stepgen max 20% higher than the axis
STEPGEN_MAX_VEL =   300.0
STEPGEN_MAX_ACC =   1200.0

BACKLASH =          0.000

# decreasing scale increases length per step
SCALE = -25.46

```



```

MIN_LIMIT =          150.0
MAX_LIMIT =          2600.0

FERROR =      1.0
MIN_FERROR = 0.25

HOME =              1515
HOME_OFFSET =      1515
HOME_SEARCH_VEL =          -50.0
HOME_LATCH_VEL =         -10.0
HOME_IGNORE_LIMITS =    YES
HOME_USE_INDEX =        NO
HOME_SEQUENCE =         1

# Set to zero if you don't have physical home/limit switches
# Set to the desired homing and latch velocity if you have switches
# See: https://github.com/machinekit/machinekit-docs

# these are in nanoseconds
DIRSETUP =           2000
DIRHOLD =             2000
STEPLEN =             10000
STEPSPACE =           10000

[AXIS_2]

TYPE =                LINEAR
MAX_VELOCITY =        100.0
MAX_ACCELERATION =    200.0
# Set Stepgen max 20% higher than the axis
STEPGEN_MAX_VEL =     200.0
STEPGEN_MAX_ACC =     400.0

MIN_LIMIT =           -1
MAX_LIMIT =           8.9

FERROR =      1.0
MIN_FERROR = 0.5

HOME = 8
HOME_OFFSET = 0
HOME_SEARCH_VEL =     0.0
HOME_LATCH_VEL =     0.0
HOME_SEQUENCE = 0

[EMCIO]

# Name of IO controller program, e.g., io
EMCIO =                io

# cycle time, in seconds
CYCLE_TIME =           0.100

# tool table file
TOOL_TABLE =           tool.tbl

```

Bipod.py

```
#!/usr/bin/env python
from subprocess import Popen, PIPE
from interrupt import Interrupt
import os
import glob
import logging
import linuxcnc
import time
import pickle

# setup linuxcnc control/feedback channels
com = linuxcnc.command()
sta = linuxcnc.stat()
err = linuxcnc.error_channel()

class ShutdownException(Exception):
    pass

# where the ngc files are
ngc_dir = '/tmp/gcodes/*ngc'

# gondola flags
GOND_FLAG_CHARGE = 1

# count how many programs have been run
program_count = 0
log_interval = 5 #seconds

# setup logging
log = logging.getLogger('')
log.setLevel(logging.DEBUG)

# create console handler and set level to info
log_format = logging.Formatter('%(asctime)s - %(levelname)-8s - %(message)s')
ch = logging.StreamHandler()
ch.setFormatter(log_format)
log.addHandler(ch)

# create file handler and set to debug
fh = logging.FileHandler('bipod.log')
fh.setFormatter(log_format)
log.addHandler(fh)

# this should come from .hal
width = 2140
g54 = { 'x': 0, 'y': 0, 'z': 0 } # this is where the g54 0,0 point will be
# what files to run to charge the gondola battery
dir_path = os.path.dirname(os.path.realpath(__file__))
charge_gcode = dir_path + '/charge.ngc'
precharge_gcode = dir_path + '/precharge.ngc'

#####
# start of functions

def run_program(file):
    log.info("starting program %d: %s" % (program_count, file))
    log.debug("changing to auto mode")
    com.mode(linuxcnc.MODE_AUTO)
    com.wait_complete() # wait until mode switch executed
    sta.poll()

    if sta.task_mode == linuxcnc.MODE_AUTO:
        log.debug("success")
```

```

com.program_open(file)
com.auto(linuxcnc.AUTO_RUN, 0) # second arg is start line
wait_till_done()

def turn_on_charger():
    log.info("turning on charger")
    os.system("config-pin p9.13 hi")

def turn_off_charger():
    log.info("turning off charger")
    os.system("config-pin p9.13 lo")

def move_to_precharge():
    log.info("moving to precharge position")
    turn_off_charger()
    run_program(precharge_gcode)

def move_to_charge():
    log.info("moving to charging position")
    run_program(charge_gcode)

    # turn on power
    turn_on_charger()

    # wait for charge connection
    time.sleep(4)

    # then check if it's charging
    gond_flags = Popen('halcmd getp xbee.gond_flags', shell=True,
stdout=PIPE).stdout.read().strip()
    gond_flags = int(gond_flags)
    if gond_flags & GOND_FLAG_CHARGE:
        log.info("docked and charging")
    else:
        log.warning("docked but not charging")
#     turn_off_charger() # leave it on

def gondola_touched():
    gond_touch = Popen('halcmd getp xbee.gond_touch', shell=True,
stdout=PIPE).stdout.read().strip()
    if gond_touch is not None:
        try:
            gond_touch = int(gond_touch)
            if gond_touch >= 4:
                log.warning("gondola touch = %d" % gond_touch)

                return True
        except ValueError:
            log.warning("got gondola flag %s couldn't convert to int" %
gond_flags)

def wait_till_done():
    paused = False
    last_log = 0
    while True:
        """
        if gondola touched() and not paused:
            log.warning("gondola touch detected - pausing")
            paused = True
            com.auto(linuxcnc.AUTO_PAUSE)
        elif not gondola_touched() and paused:
            log.warning("gondola touch OK - resuming")
            paused = False
            com.auto(linuxcnc.AUTO_RESUME)
        """
        sta.poll()
        error = err.poll()

```

```

    if error:
        kind, text = error
        log.warning("error: %s" % text)
    """
    1 EXEC_ERROR
    2 EXEC_DONE
    3 EXEC_WAITING_FOR_MOTION
    4 EXEC_WAITING_FOR_MOTION_QUEUE
    5 EXEC_WAITING_FOR_PAUSE
    6 EXEC_WAITING_FOR_MOTION_AND_IO
    7 EXEC_WAITING_FOR_DELAY
    8 EXEC_WAITING_FOR_SYSTEM_CMD

    2, 3 (not much), 5 or 7 (lots)
    """
    if time.time() - last_log > log_interval:
        last_log = time.time()
        log.debug("exec state %d" % sta.exec_state)

        #1 to 4: INTERP_IDLE, INTERP_READING, INTERP_PAUSED,
INTERP_WAITING
        # 1, 2 or 4
        log.debug("interp state %d" % sta.interp_state)

        # RCS_DONE, RCS_EXEC, RCS_ERROR.
        # 1, 2 (mostly) or 3 - worth investigating state 3
        log.debug("state %d" % sta.state)

        #1 to 6: INTERP_OK, INTERP_EXIT, INTERP_EXECUTE_FINISH,
INTERP_ENDFILE, INTERP_FILE_NOT_OPEN, INTERP_ERROR
        # always 0 so far
        log.debug("interp errcode %d" % sta.interpreter_errcode)
        log.debug("line in file %d" % sta.motion_line)

    time.sleep(0.1)
    if sta.interp_state == linuxcnc.INTERP_IDLE:
        log.info("finished")
        break

    # button pressed? shutdown
    if not button_int.isAlive():
        raise ShutdownException()

def set_g54():
    log.info("changing to mdi mode")
    com.mode(linuxcnc.MODE_MDI)
    com.wait_complete() # wait until mode switch executed
    sta.poll()
    if sta.task_mode == linuxcnc.MODE_MDI:
        log.debug("success")

    log.info("resetting g54 to x%d y%d z%d" % (g54['x'], g54['y'], g54['z']))
    com.mdi("g10 l2 p1 x%d y%d" % (g54['x'], g54['y']))
    com.feedrate(200)

#####
# start of main

# wait for button
button_int = Interrupt()
button_int.start()
log.info("started - waiting for button")
button_int.join()

# start another button interrupt
button_int = Interrupt()
button_int.daemon = True # don't have to clean up after

```

```

button_int.start()

# get state right
com.state(linuxcnc.STATE_ESTOP_RESET)
com.wait_complete()
com.state(linuxcnc.STATE_ON)
com.wait_complete()

# home all
log.info("homing all")
com.home(0)
com.home(1)
com.home(2)

while not sta.homed[0:3] == (1,1,1):
    log.debug("homing...")
    sta.poll()
    time.sleep(1)

# switch to world mode
log.info("teleop mode")
com.teleop_enable(1)
com.wait_complete()
set_g54()

# move to charge pos

move_to_precharge()
move_to_charge()

# wait for files to appear
try:
    while True:
        if not button_int.isAlive():
            raise ShutdownException()

        files = glob.glob(ngc_dir)
        if len(files) == 0:
            log.info("no files, sleeping")
            time.sleep(10)
            continue

        set_g54() # in case the program changed it
        move_to_precharge()
        run_program(files[0])
        program_count += 1

        os.remove(files[0])
        move_to_precharge()
        move_to_charge()

except KeyboardInterrupt:
    log.info("interrupted!")
except ShutdownException:
    log.warning("shutdown")
    # do the shutdown
    os.system("sudo halt")
except Exception as e:
    log.error("unexpected exception! %s" % e)

turn_off_charger()
log.info("done")

```

Bipodkins.c

```
/*
*****
* Description: bipodkins.c
*
* License: GPL Version 2
*
*****
*/

#include "motion.h"          /* these decls */
#include "rtapi_math.h"

/* ident tag */
#ifndef __GNUC__
#ifndef __attribute__
#define __attribute__(x)
#endif
#endif

#ifdef RTAPI
#include "hal.h"
struct {
    hal_float_t bx;
} *haldata = 0;

#define Bx (haldata->bx)
#else
double Bx;
#endif

#define sq(x) ((x)*(x))

int kinematicsForward(const double * joints,
                      EmcPose * pos,
                      const KINEMATICS_FORWARD_FLAGS * fflags,
                      KINEMATICS_INVERSE_FLAGS * iflags)
{
#define AD (joints[0])
#define BD (joints[1])
#define Dx (pos->tran.x)
#define Dy (pos->tran.y)
#define Dz (pos->tran.z)

    double AD2 = sq(joints[0]);
    double BD2 = sq(joints[1]);

    double x = (AD2 - BD2 + sq(Bx)) / (2 * Bx);
    double y2 = AD2 - x * x;

    if (y2 < 0)
        return -1;

    Dx = x;
    Dy = rtapi_sqrt(y2);
    Dz = joints[2];

    if (*fflags) {
        Dy = -Dy;
    }

    return 0;
}

#undef AD
```

```

#undef BD
#undef Dx
#undef Dy
#undef Dz
}

int kinematicsInverse(const EmcPose * pos,
                    double * joints,
                    const KINEMATICS_INVERSE_FLAGS * iflags,
                    KINEMATICS_FORWARD_FLAGS * fflags)
{
#define AD (joints[0])
#define BD (joints[1])
#define Dx (pos->tran.x)
#define Dy (pos->tran.y)
#define Dz (pos->tran.z)

    double x2 = sq(Dx);
    double y2 = sq(Dy);
    AD = rtapi_sqrt(x2 + y2);
    BD = rtapi_sqrt(sq(Bx - Dx) + y2);
    joints[2] = Dz;

    *fflags = 0;
    if (Dy < 0.0) {
        *fflags = 1;
    }

    return 0;

#undef AD
#undef BD
#undef Dx
#undef Dy
#undef Dz
}

KINEMATICS_TYPE kinematicsType()
{
    return KINEMATICS_BOTH;
}

#ifdef MAIN

#include <stdio.h>
#include <string.h>

/*
    Interactive testing of kins.

    Syntax: a.out <Bx>
*/
int main(int argc, char *argv[])
{
#ifdef BUFFERLEN
#define BUFFERLEN 256
#endif
    char buffer[BUFFERLEN];
    char cmd[BUFFERLEN];
    EmcPose pos, vel;
    double joints[3], jointvels[3];
    char inverse;
    char flags;
    KINEMATICS_FORWARD_FLAGS fflags;

    inverse = 0;          /* forwards, by default */
    flags = 0;           /* didn't provide flags */
    fflags = 0;          /* above xy plane, by default */

```

```

if (argc != 2 || 1 != sscanf(argv[1], "%lf", &Bx)) {
    fprintf(stderr, "syntax: %s Bx\n", argv[0]);
    return 1;
}

while (! feof(stdin)) {
    if (inverse) {
        printf("inv> ");
    }
    else {
        printf("fwd> ");
    }
    fflush(stdout);

    if (NULL == fgets(buffer, BUFFERLEN, stdin)) {
        break;
    }
    if (1 != sscanf(buffer, "%s", cmd)) {
        continue;
    }

    if (! strcmp(cmd, "quit")) {
        break;
    }
    if (! strcmp(cmd, "i")) {
        inverse = 1;
        continue;
    }
    if (! strcmp(cmd, "f")) {
        inverse = 0;
        continue;
    }
    if (! strcmp(cmd, "ff")) {
        if (1 != sscanf(buffer, "%*s %d", &fflags)) {
            printf("need forward flag\n");
        }
        continue;
    }

    if (inverse) {          /* inverse kins */
        if (3 != sscanf(buffer, "%lf %lf %lf",
            &pos.tran.x,
            &pos.tran.y,
            &pos.tran.z)) {
            printf("need X Y Z\n");
            continue;
        }
        if (0 != kinematicsInverse(&pos, joints, NULL, &fflags)) {
            printf("inverse kin error\n");
        }
        else {
            printf("%f\t%f\t%f\n", joints[0], joints[1], joints[2]);
            if (0 != kinematicsForward(joints, &pos, &fflags, NULL)) {
                printf("forward kin error\n");
            }
            else {
                printf("%f\t%f\t%f\n", pos.tran.x, pos.tran.y, pos.tran.z);
            }
        }
    }
    else {                  /* forward kins */
        if (flags) {
            if (4 != sscanf(buffer, "%lf %lf %lf %d",
                &joints[0],
                &joints[1],
                &joints[2],
                &fflags)) {
                printf("need 3 strut values and flag\n");
            }
        }
    }
}

```



```

        continue;
    }
}
else {
if (3 != sscanf(buffer, "%lf %lf %lf",
                &joints[0],
                &joints[1],
                &joints[2])) {
    printf("need 3 strut values\n");
    continue;
}
}
if (0 != kinematicsForward(joints, &pos, &fflags, NULL)) {
printf("forward kin error\n");
}
else {
printf("%f\t%f\t%f\n", pos.tran.x, pos.tran.y, pos.tran.z);
if (0 != kinematicsInverse(&pos, joints, NULL, &fflags)) {
    printf("inverse kin error\n");
}
else {
    printf("%f\t%f\t%f\n", joints[0], joints[1], joints[2]);
}
}
}
} /* end while (! feof(stdin)) */

return 0;
}

#endif /* MAIN */

#ifdef RTAPI
#include "rtapi.h" /* RTAPI realtime OS API */
#include "rtapi_app.h" /* RTAPI realtime module decls */
#include "hal.h"
static vtkins_t vtk = {
    .kinematicsForward = kinematicsForward,
    .kinematicsInverse = kinematicsInverse,
    .kinematicsType = kinematicsType
};

MODULE_LICENSE("GPL");
static const char *name = "bipodkins";

int comp_id, vtable_id;
int rtapi_app_main(void) {
    int res = 0;

    comp_id = hal_init("bipodkins");
    if(comp_id < 0) return comp_id;

    haldata = hal_malloc(sizeof(*haldata));
    if(!haldata) goto error;
    Bx = 1.0;

    if((res = hal_param_float_new("bipodkins.Bx", HAL_RW, &haldata->bx,
    comp_id)) < 0) goto error;

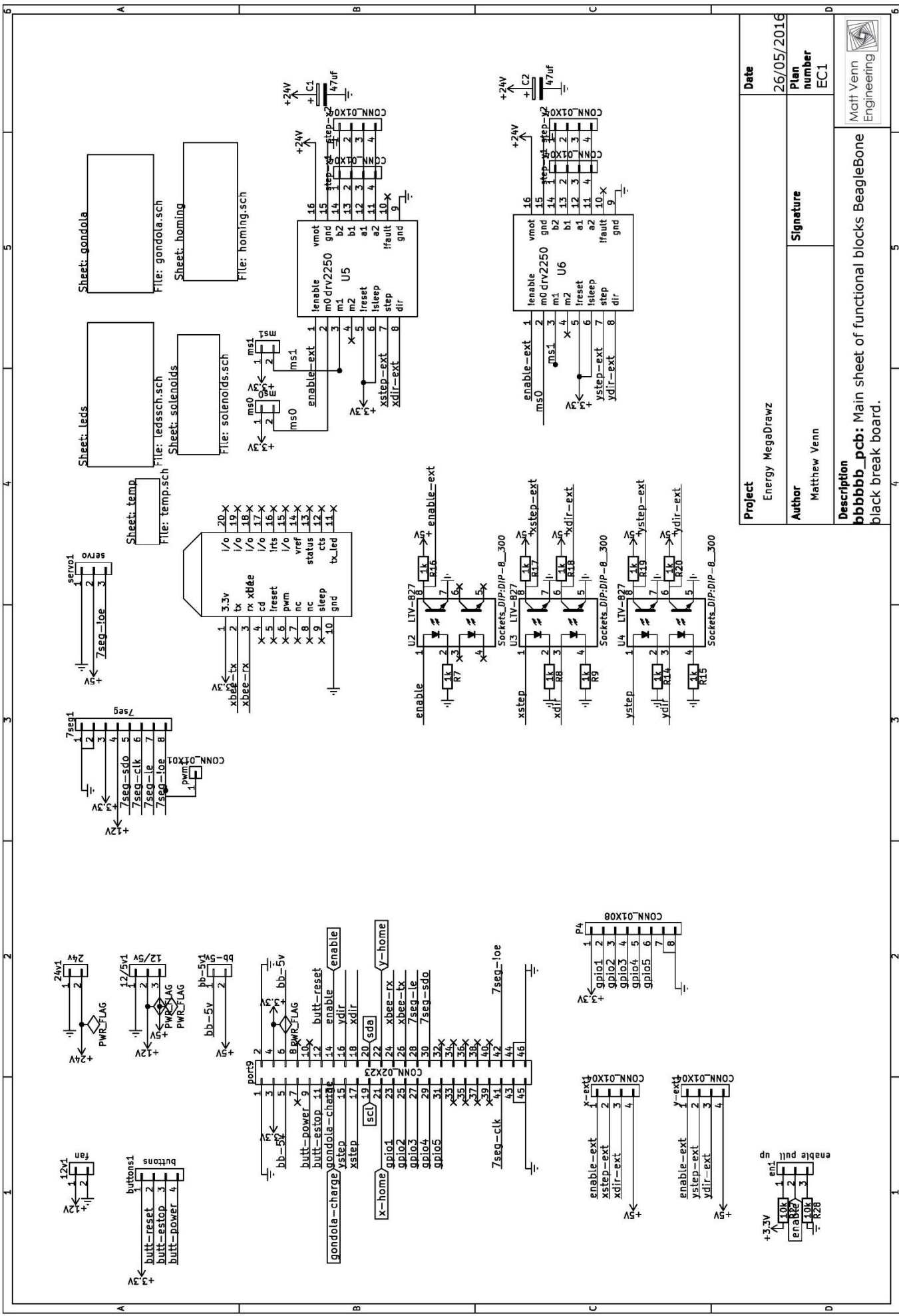
    vtable_id = hal_export_vtable(name, VTKINEMATICS_VERSION1, &vtk, comp_id);

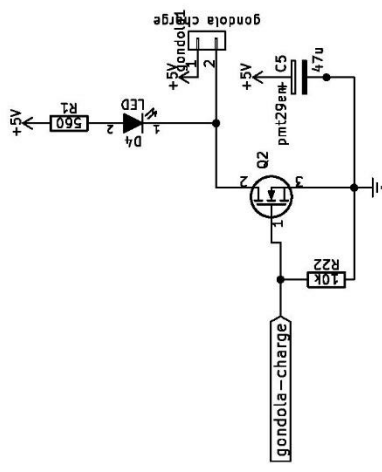
    if (vtable_id < 0) {
        rtapi_print_msg(RTAPI_MSG_ERR,
            "%s: ERROR: hal_export_vtable(%s,%d,%p) failed: %d\n",
            name, name, VTKINEMATICS_VERSION1, &vtk, vtable_id);
        return -ENOENT;
    }
}


```

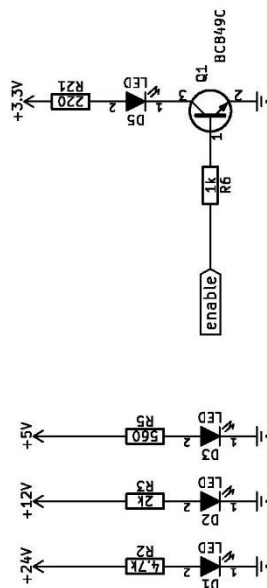
```
    hal_ready(comp_id);  
    return 0;  
  
error:  
    hal_exit(comp_id);  
    return res;  
}  
  
void rtapi_app_exit(void)  
{  
    hal_remove_vtable(vtable_id);  
    hal_exit(comp_id);  
}  
#endif
```

APPENDIX V, PCB SCHEMATICS



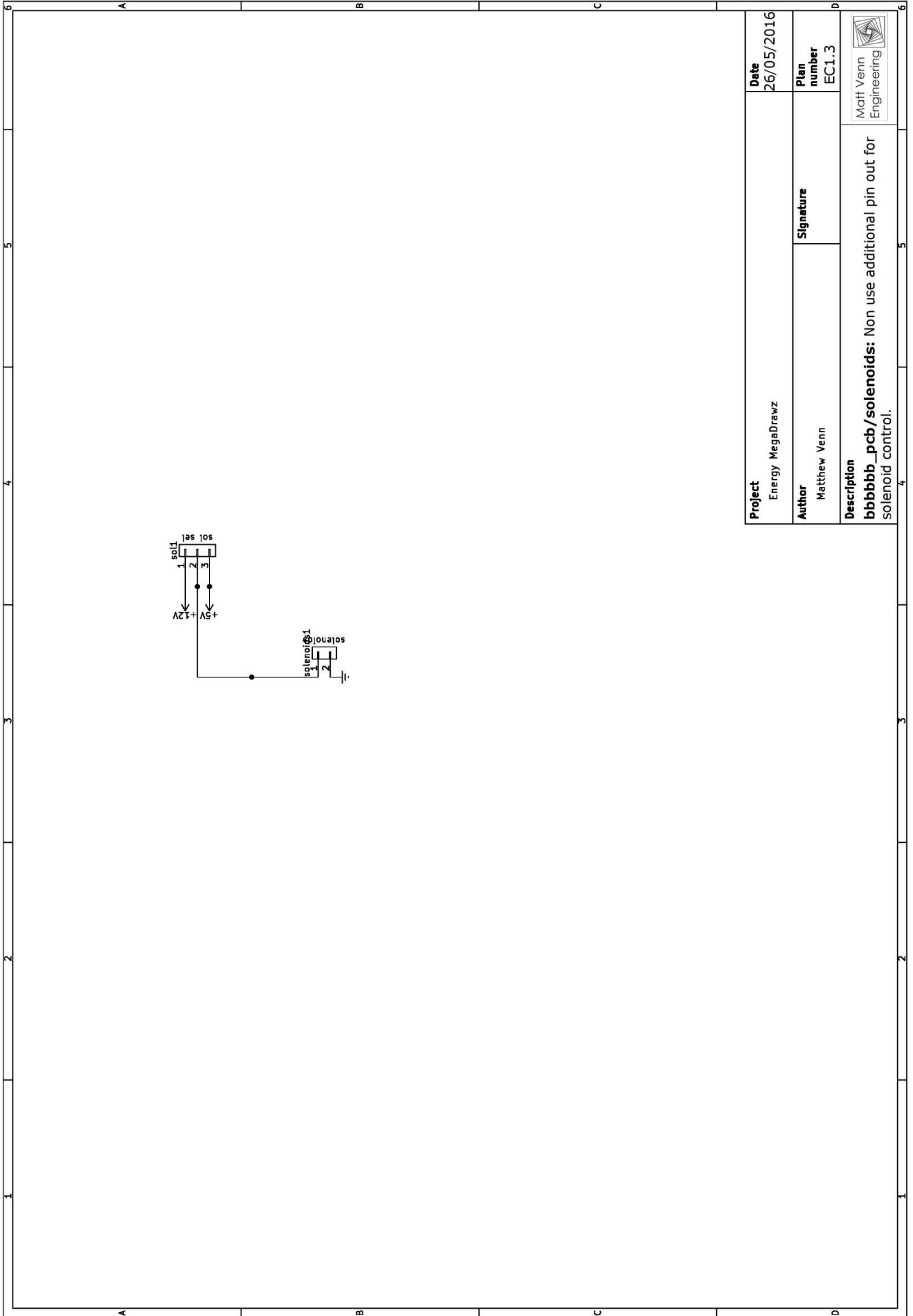


Project Energy MegaDrawz		Date 26/05/2016
Author Matthew Venn	Signature	Plan number EC1.1
Description bbbbbb_pcb/gondola: charge control circuit and LED indicator.		 Matt Venn Engineering



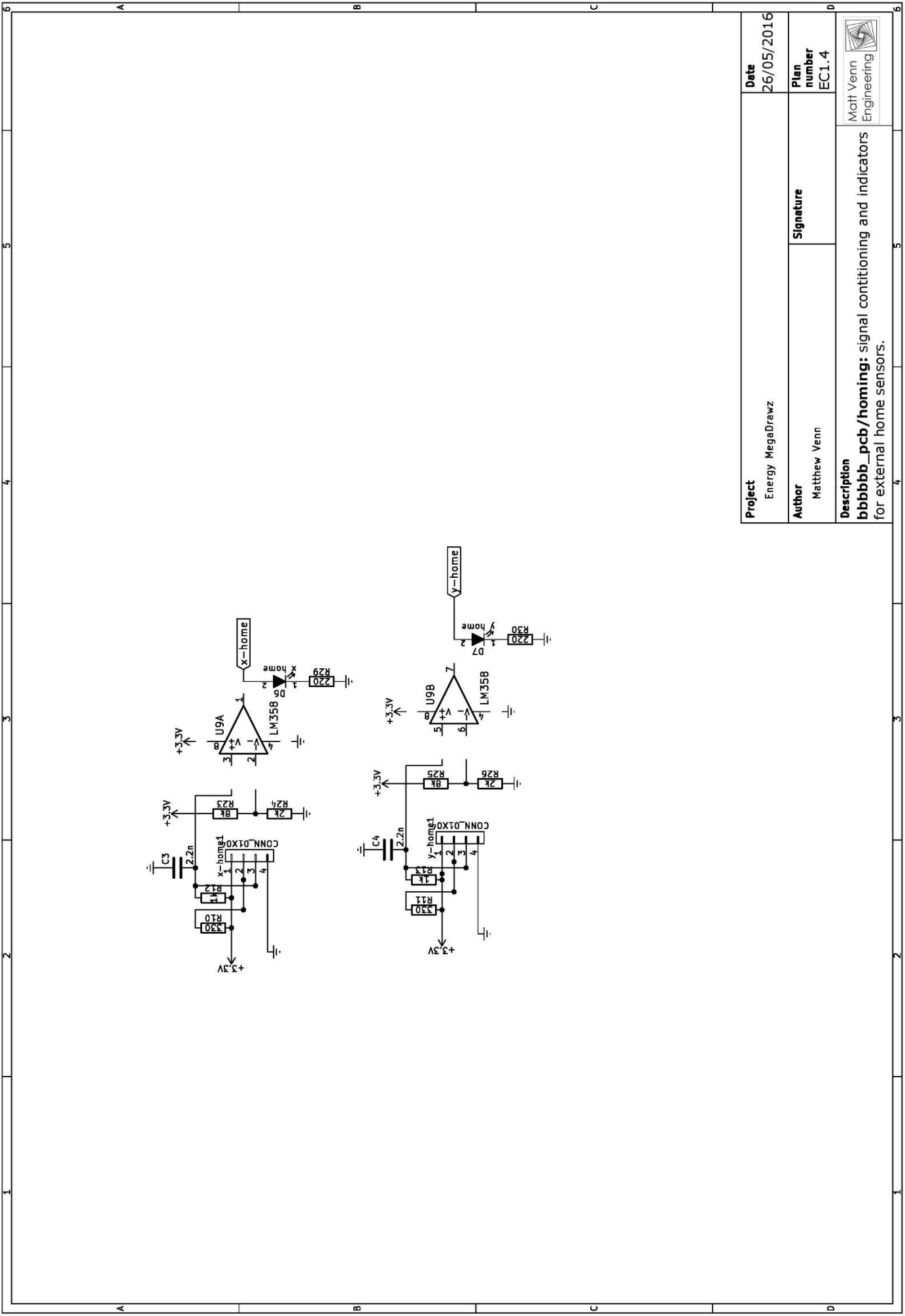
Project Energy MegaDrawz	Date 26/05/2016
Author Matthew Venn	Plan number EC1.2
Description bbbbbb_pcb/leds:status leds for power supplies.	




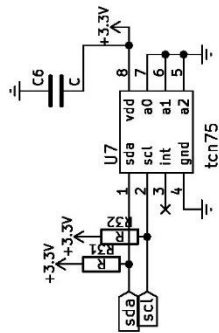


Project Energy MegaDrawz	Date 26/05/2016
Author Matthew Venn	Plan number EC1.3
Description bbbbbb_pcb/solenoids: Non use additional pin out for solenoid control.	





Project Energy MegaDrawz	Date 26/05/2016
Author Matthew Venn	Plan number EC1.4
Description bbbbbb_pcb/homing: signal conditioning and indicators for external home sensors.	
 Matt Venn Engineering	



Project Energy MegaDrawz	Date 26/05/2016
Author Matthew Venn	Plan number EC1.5
Description bbbbbb_pcb/temp: referent circuit for I2C TCM75 temperature sensor.	



1

2

3

4

5

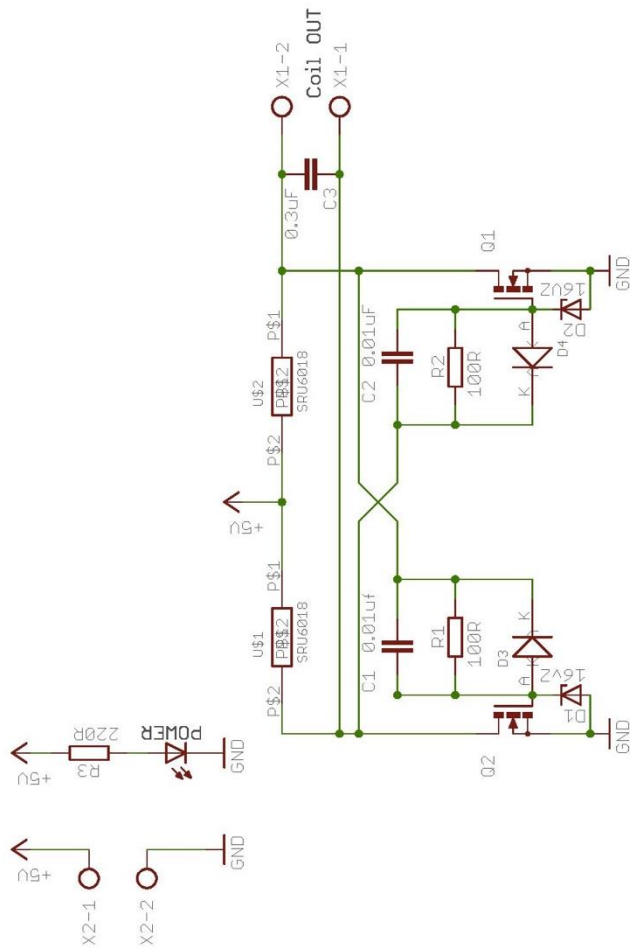
6

A

B

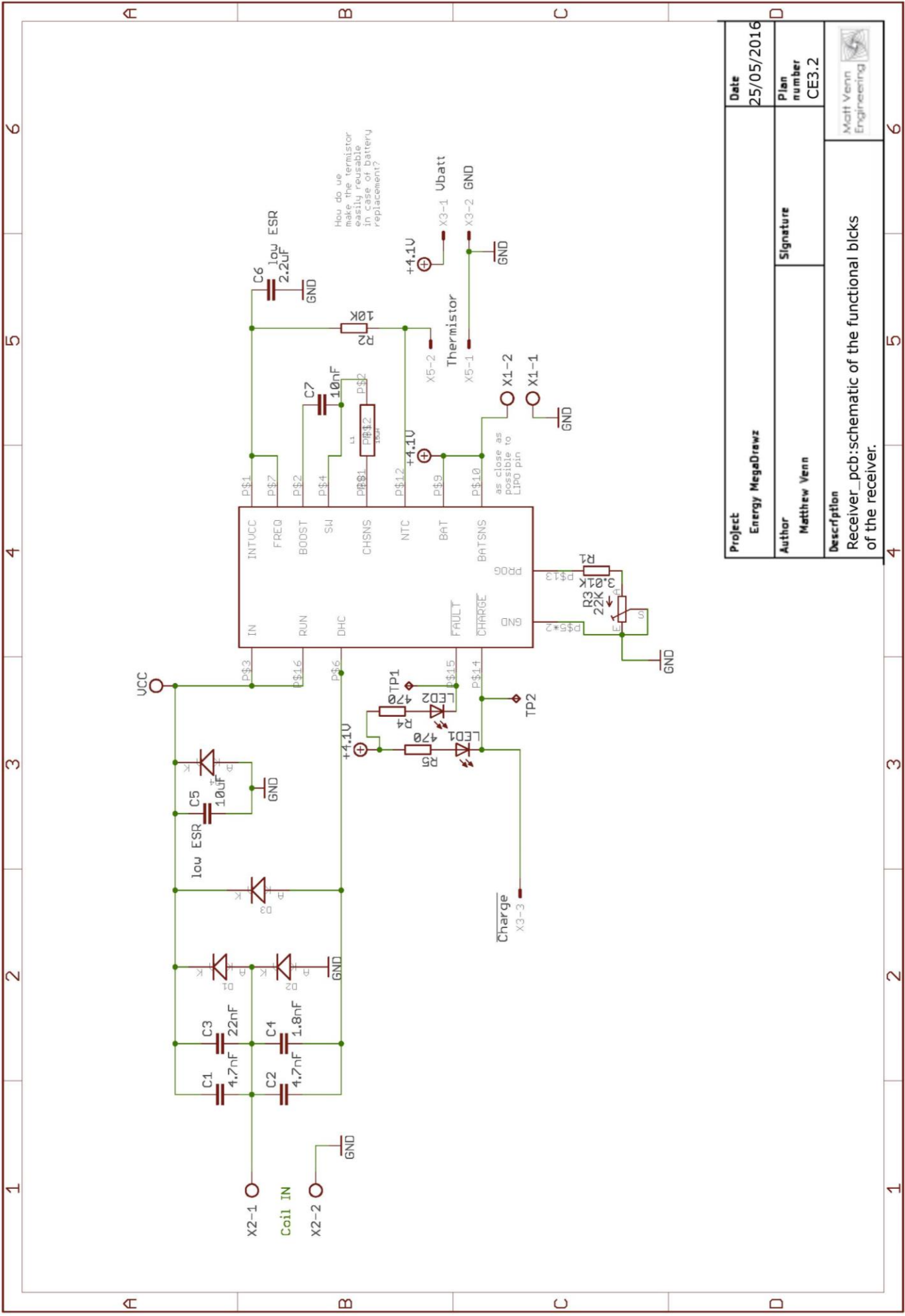
C

D




Project Energy MegaDrawz	Date 25/05/2016
Author Matthew Venn	Plan number EC3.1
Description Charger_pcb: Schematic of the functional blocks of the inductive charger.	



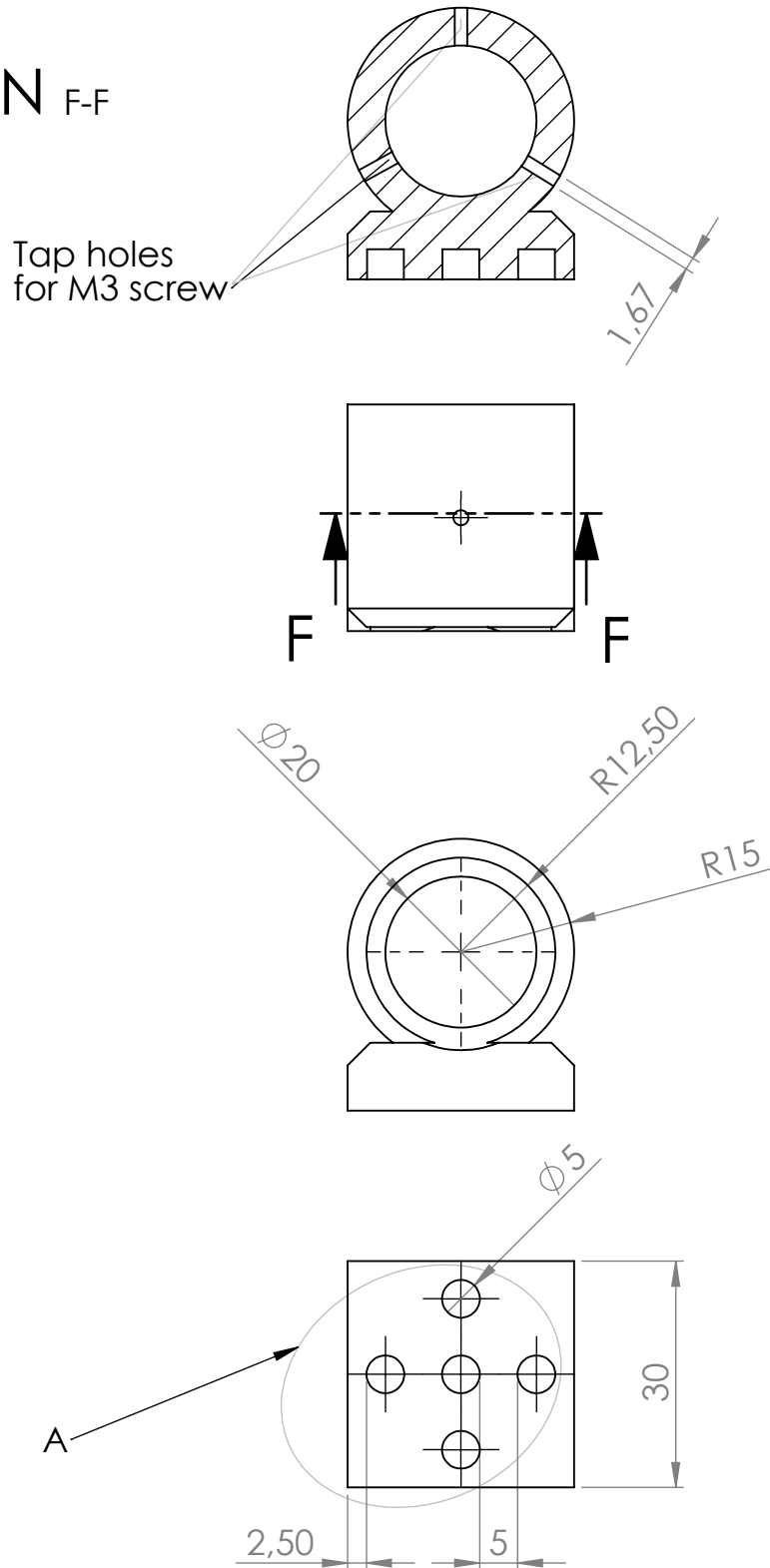



How do we
make the thermistor
easily reusable
in case of battery
replacement?

Project Energy MegaDrawz	Date 25/05/2016
Author Matthew Venn	Signature
Description Receiver_pcb:schematic of the functional blocks of the receiver.	
Plan number CE3.2	


APPENDIX VI, PLANS OF OTHER MECHANICAL COMPONENTS

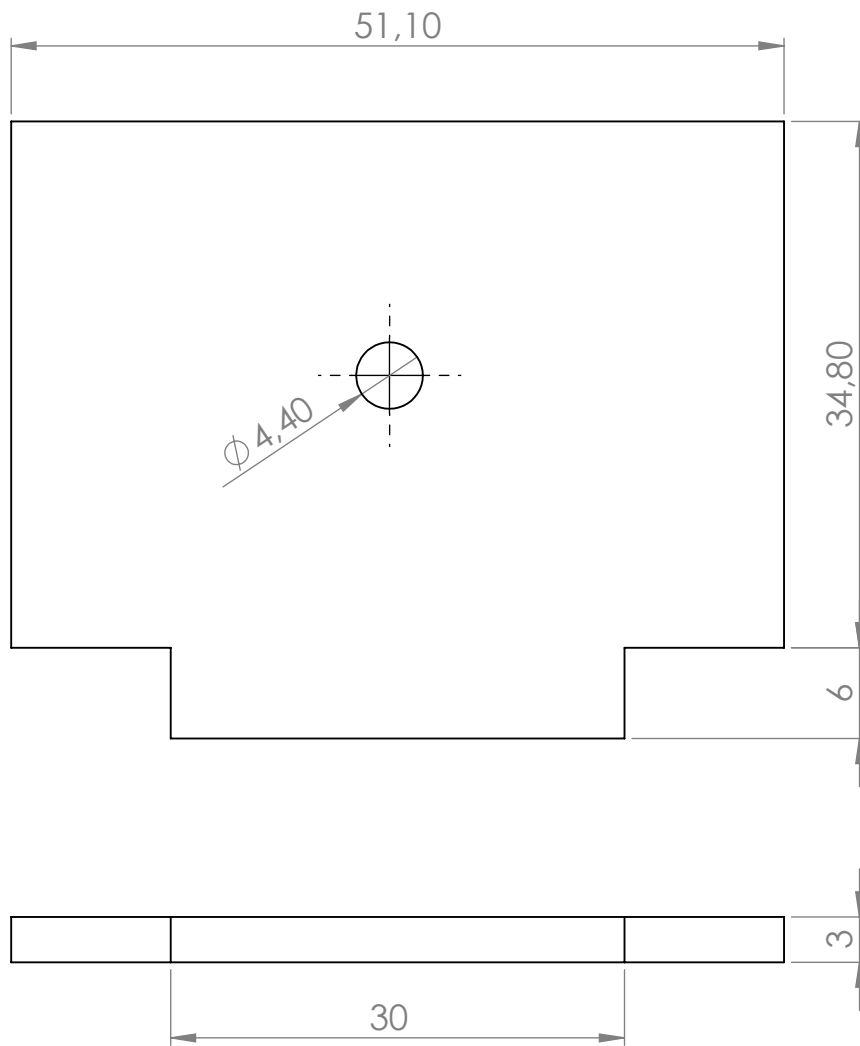
SECTION F-F




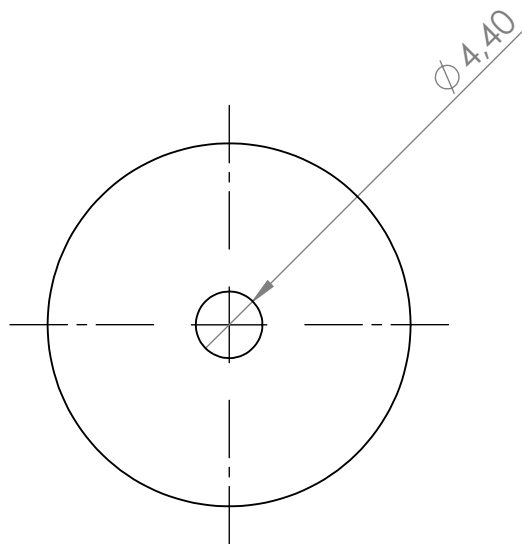
Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number C1
Description Bolisoportemagnetico_v2.1: Produce with PLA and 3D printer. Holes indicated must be taped by hand. Holes of 2.5mm of section A must have a magnet inside, use a soldering iron for heat them until 250 °C and push them until be aligned with the base. Remember place them in the opposite polarity that in the base.		Matt Venn Engineering 




Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number C2
Description Layer1: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Verify dimensions and thickness for a push fit.		Matt Venn Engineering 




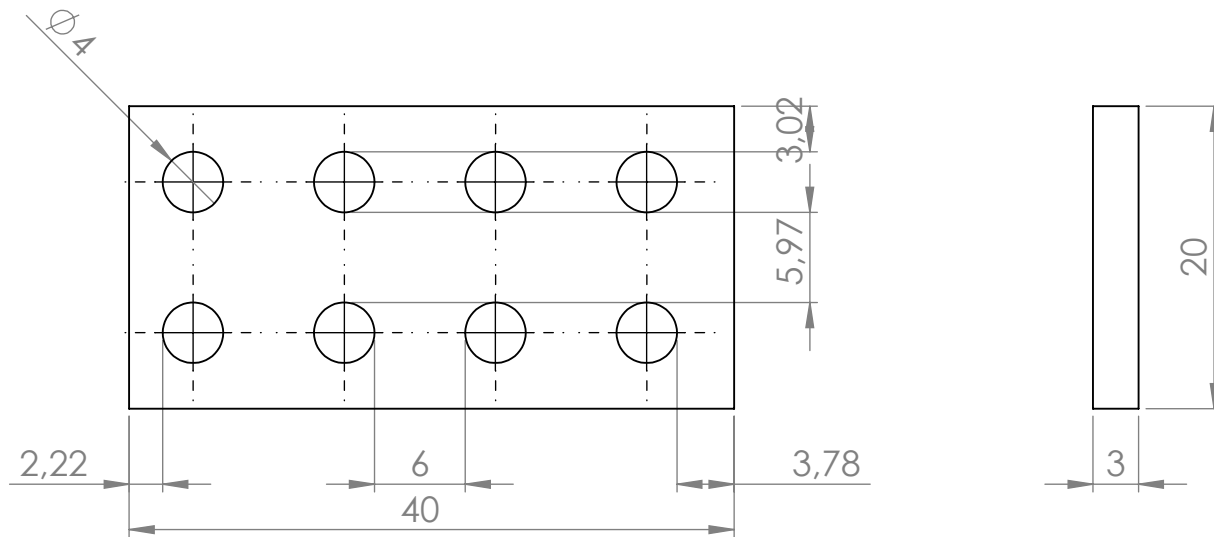
Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number C3
Description Screw_adj: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Verify thickness for a push fit.		Matt Venn Engineering 




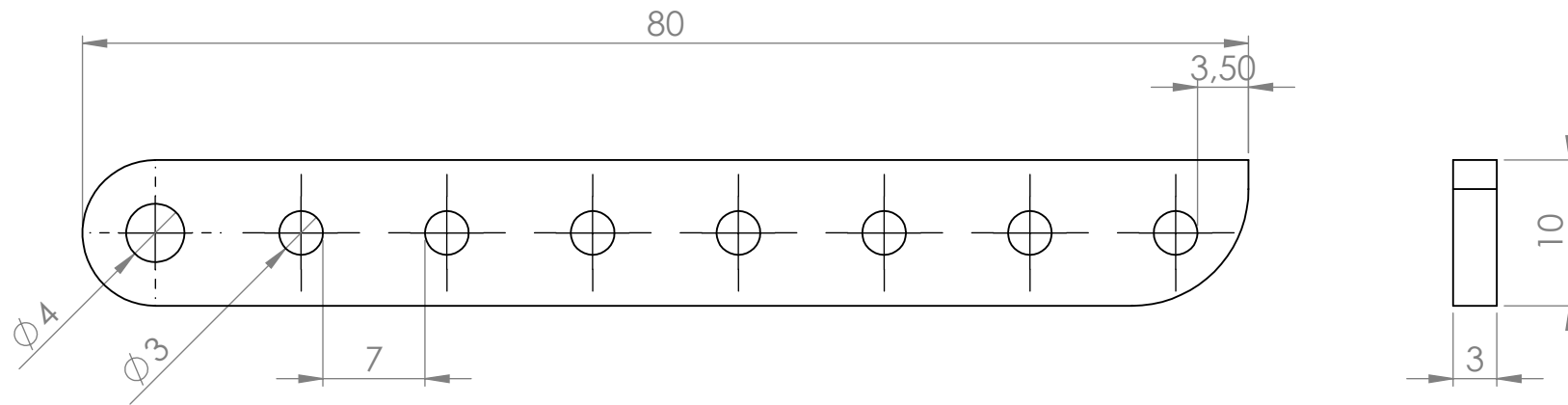
Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number C4
Description screw_limit: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




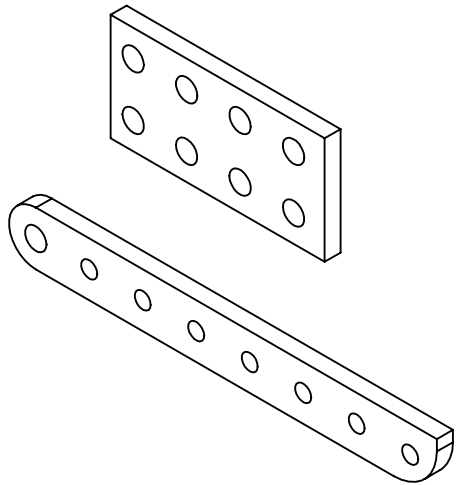
Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number C5
Description Layer0: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Verify dimensions and thickness for a push fit.		Matt Venn Engineering 



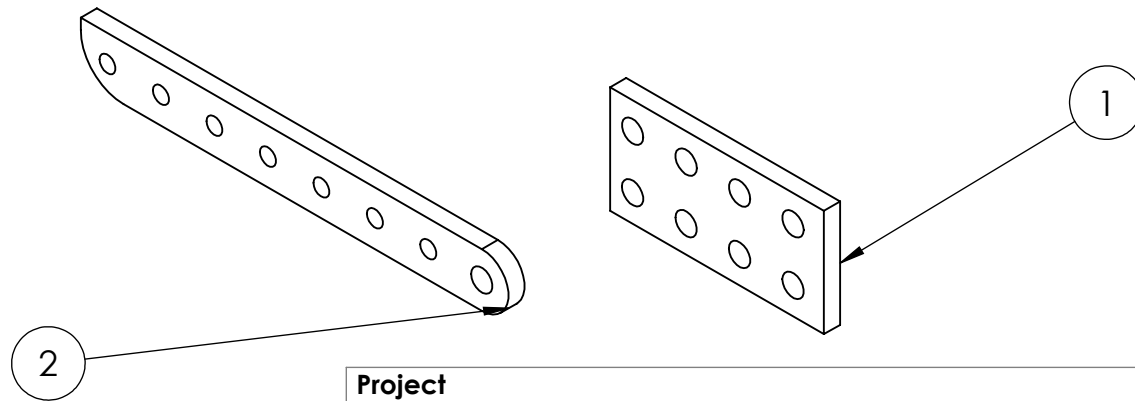
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number CS0.1
Description 20x40_anchorpoint: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




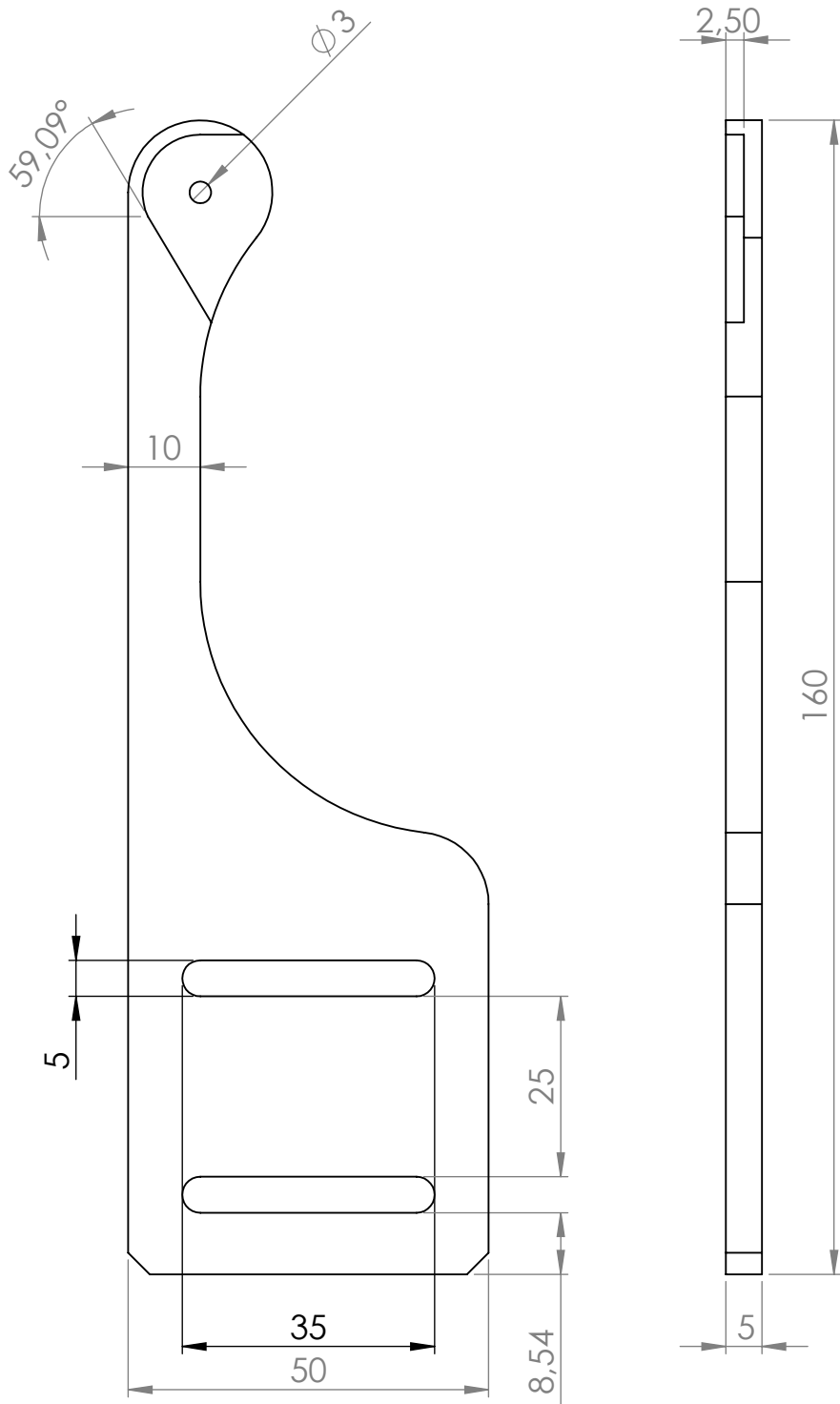
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number CS0.2
Description Lever_10x75_m4: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




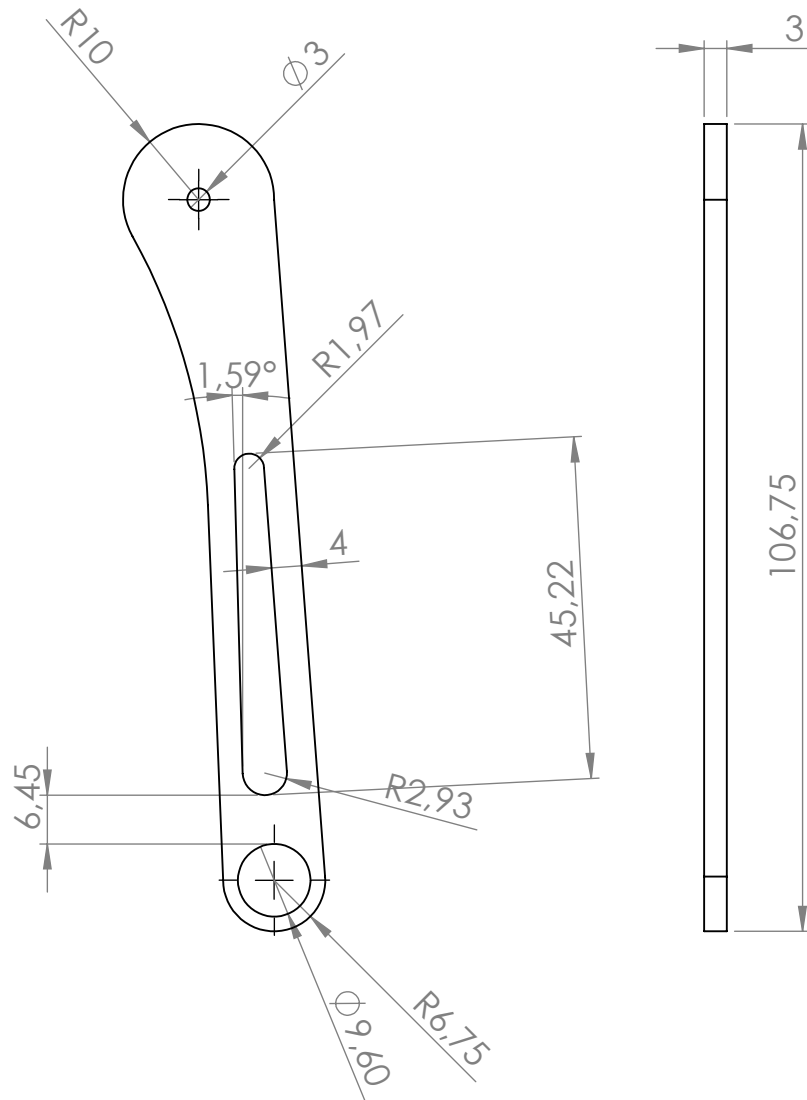
NUMBER	ID	QUANTITY
1	20x40_anchorpoint	2
2	lever_10x75_m4	2




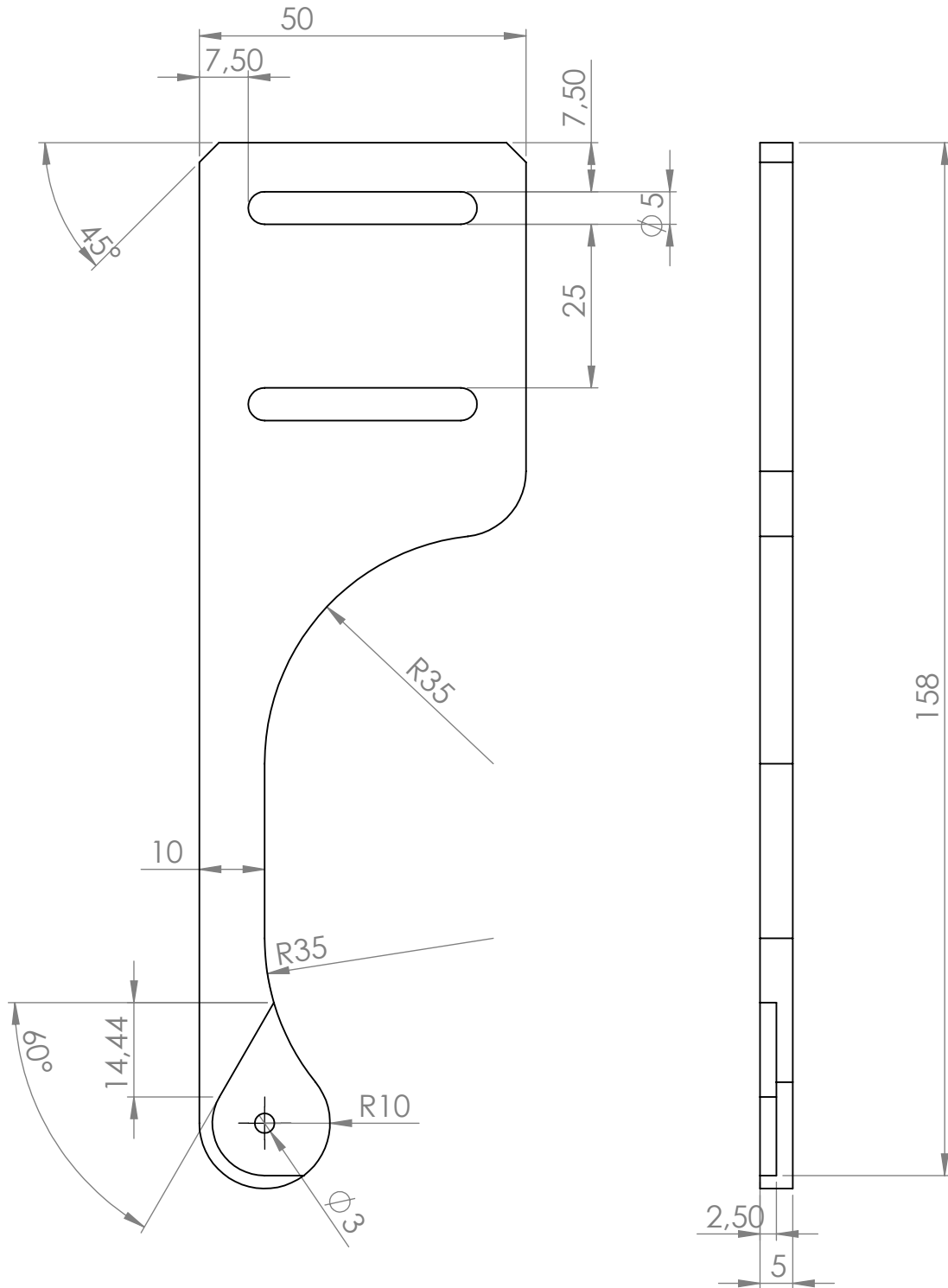
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number CS0.a
Description E_gondolacharger_v1: Cover contact area with charging pads of 2with aluminum tape.		Matt Venn Engineering 




Project Energy MegaDrawz		Date 06/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number CS1.1
Description 180x60_5mm_aluminum_arm_rigth: Use an aluminium plate of 5mm, face the top surface and machine with a CNC drilling machine. Chamfers can be added to the external edge. File the edges. Use 1mm tabs.		Matt Venn Engineering 




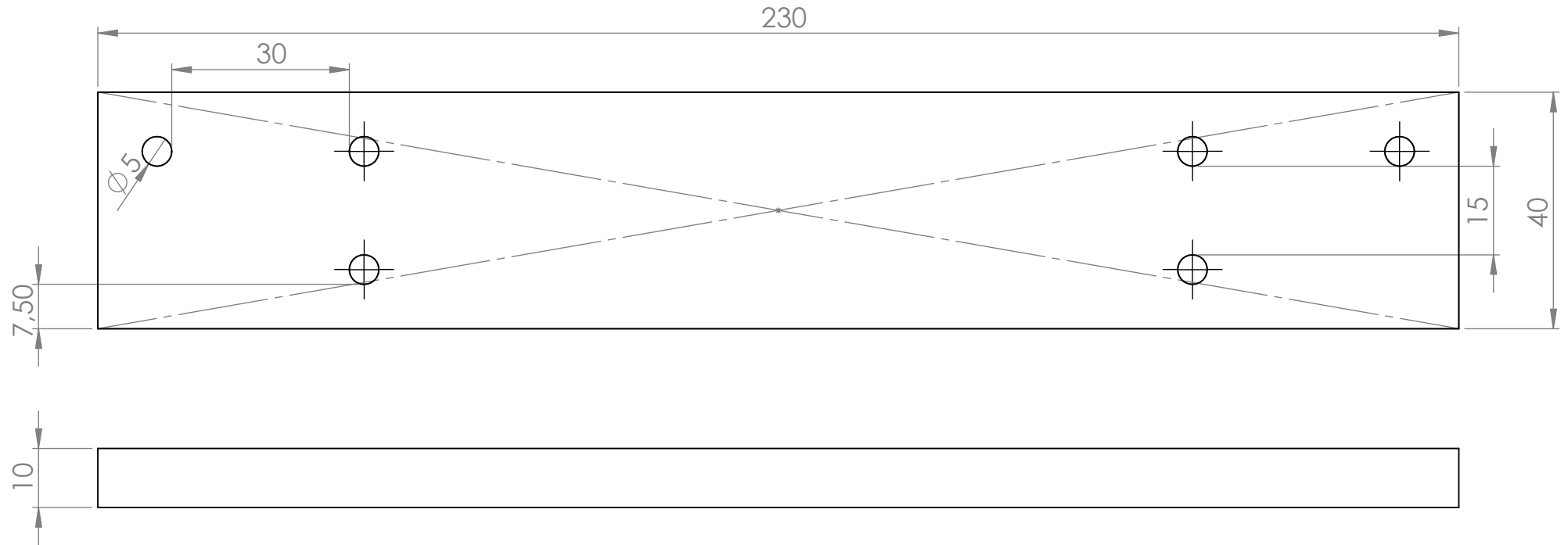
Project Energy MegaDrawz		Date 06/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number CS1.2
Description 110x30_3mm_aluminum_door: Laser cut from 3mm MDF or Use an aluminium plate of 3mm, face the top surface and machine with a CNC drilling machine. For aluminium piece chamfers can be added to the external edge, file the edges and use 1mm tabs.		Matt Venn Engineering 




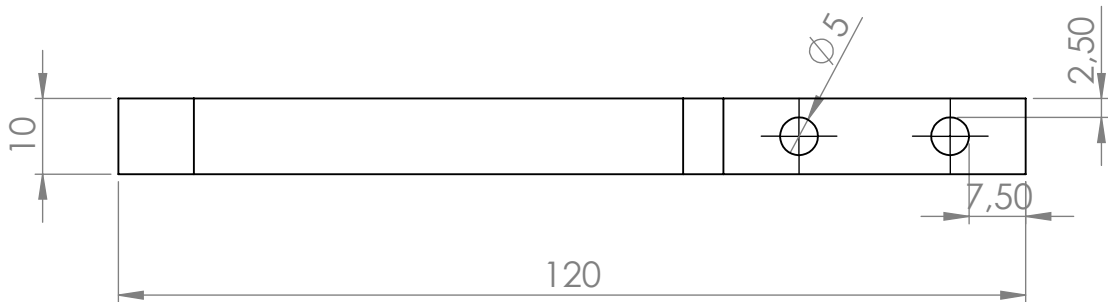
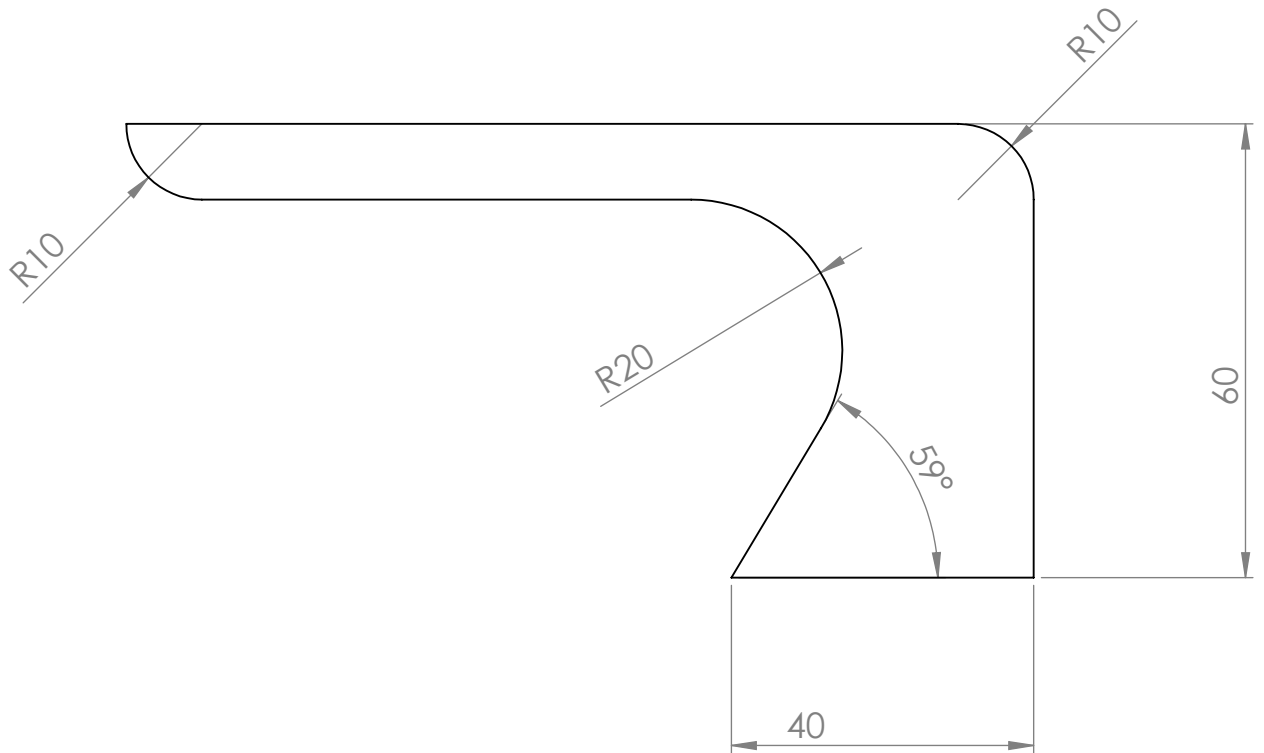
Project Energy MegaDrawz		Date 06/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number CS1.3
Description 180x60_5mm_aluminum_arm_left: Use an aluminium plate of 5mm, face the top surface and machine with a CNC drilling machine. Chamfers can be added to the external edge. File the edges. Use 1mm tabs.		Matt Venn Engineering 




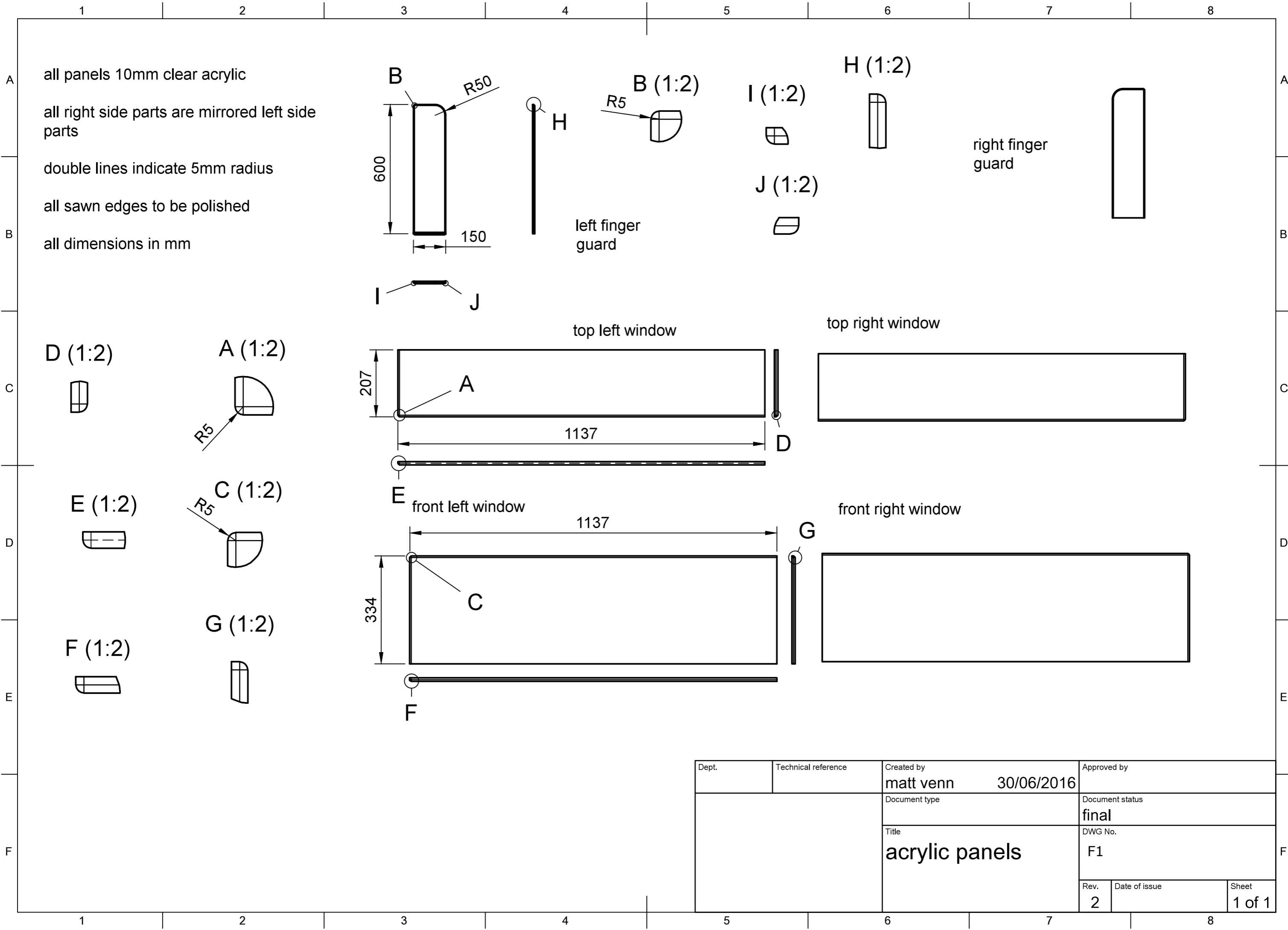
Project Energy MegaDrawz		Date 06/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number CS1.4
Description 1300x60_acrylic: Use a table saw for cut a piece of 10 mm acrylic with two 5mm diameter holes with a drilling tower. Do not remove the protector film of the acrylic.		Matthew Venn Engineering 



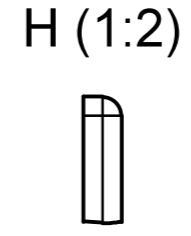
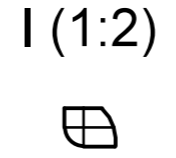
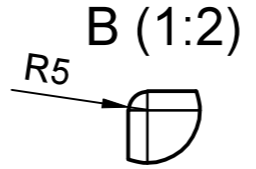
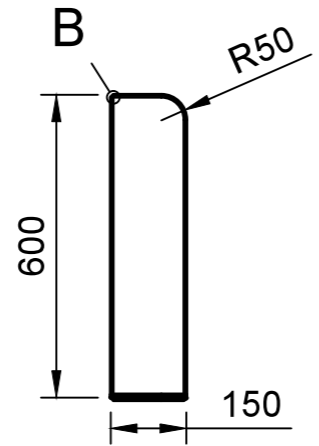
Project Energy MegaDrawz		Date 06/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number CS1.5
Description 230x40_acrylic_top: Use a laser cutter for manufacture this piece. Check internal diameters at both sides. If piece present problems mark the holes with the laser cutter and machine them with a tower driller.		Matt Venn Engineering 



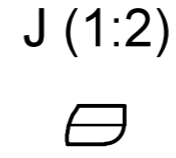
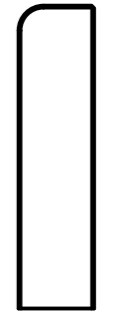
Project Energy MegaDrawz		Date 06/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number CS1.6
Description supportacrilic: Use a laser cutter for manufacture this piece. Check internal diameters at both sides. Machine the holes with a tower drilling machine.		Matt Venn Engineering 



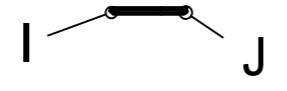
all panels 10mm clear acrylic
 all right side parts are mirrored left side parts
 double lines indicate 5mm radius
 all sawn edges to be polished
 all dimensions in mm



right finger guard

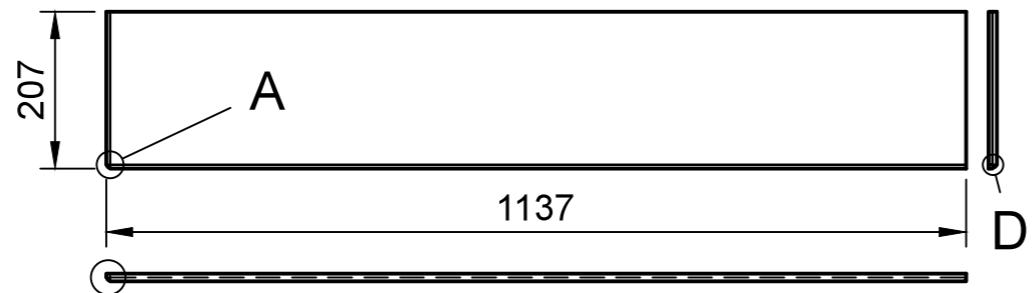


left finger guard



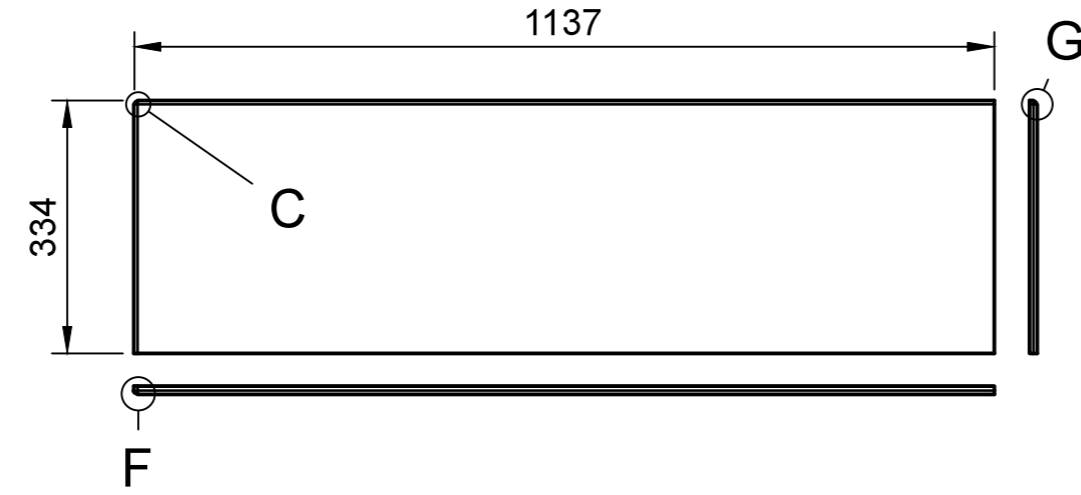
top left window

top right window



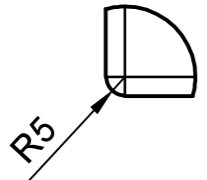
front left window

front right window



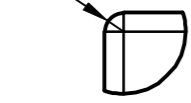
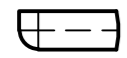
D (1:2)

A (1:2)



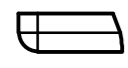
E (1:2)

C (1:2)

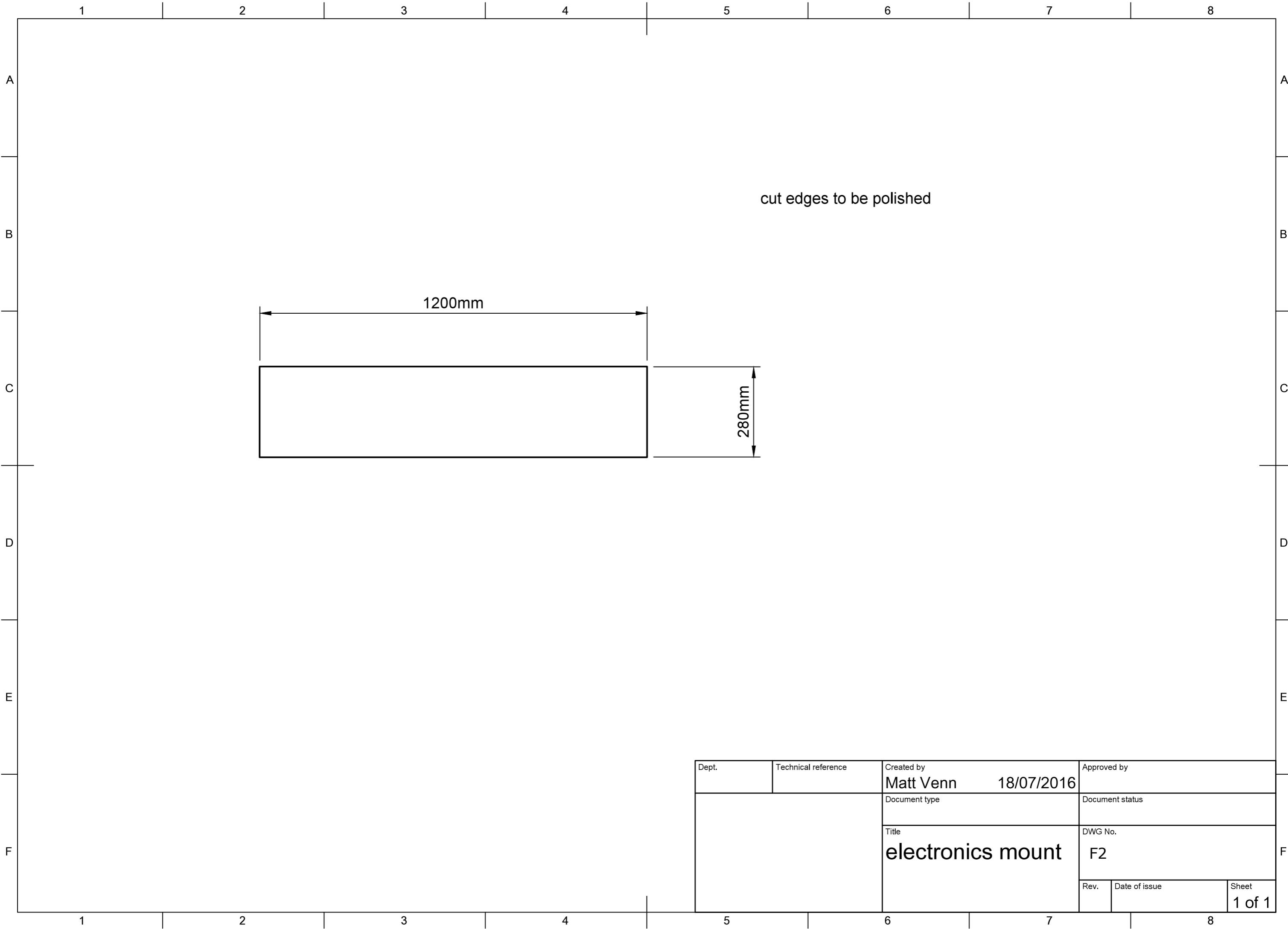


F (1:2)

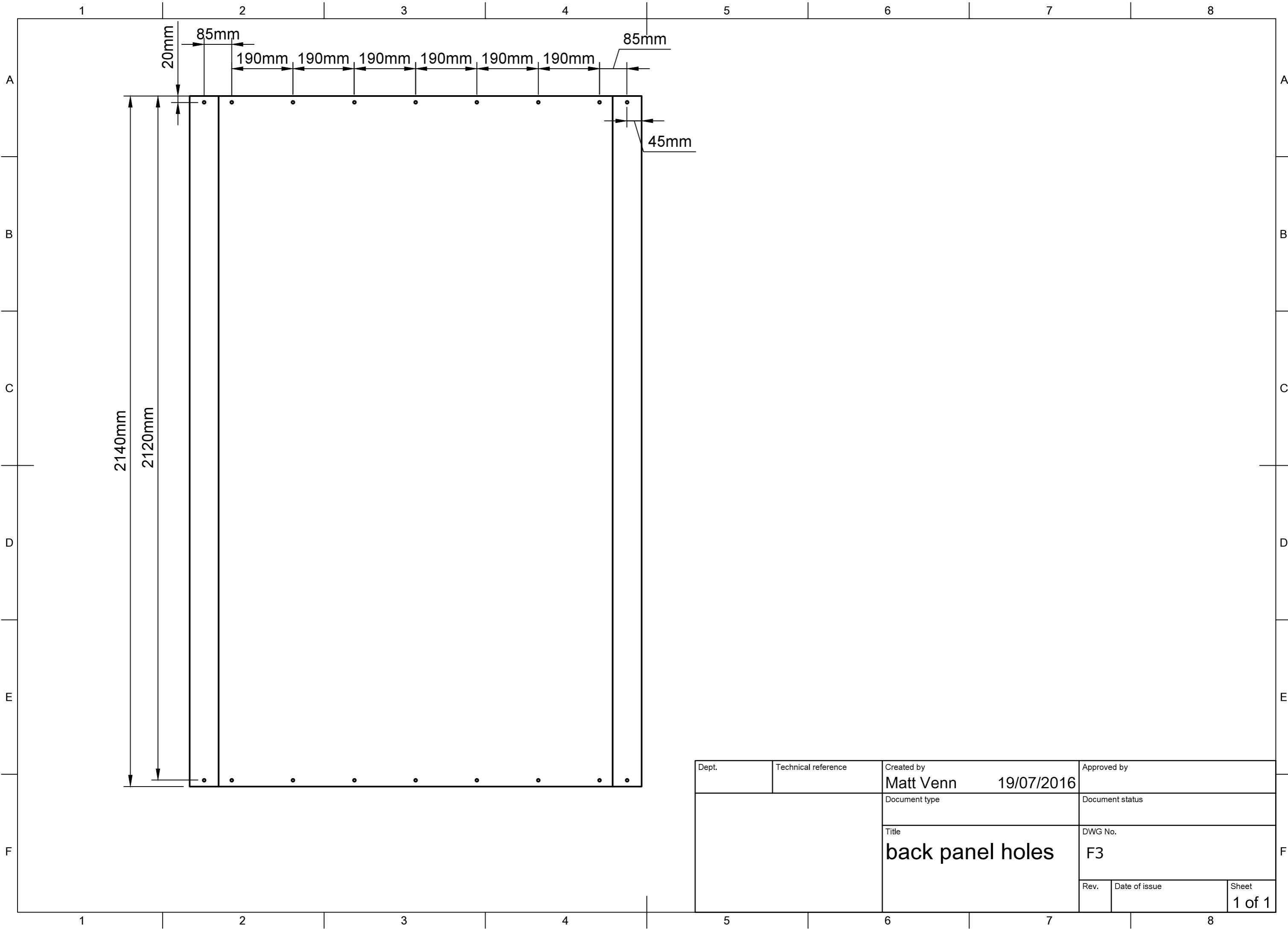
G (1:2)



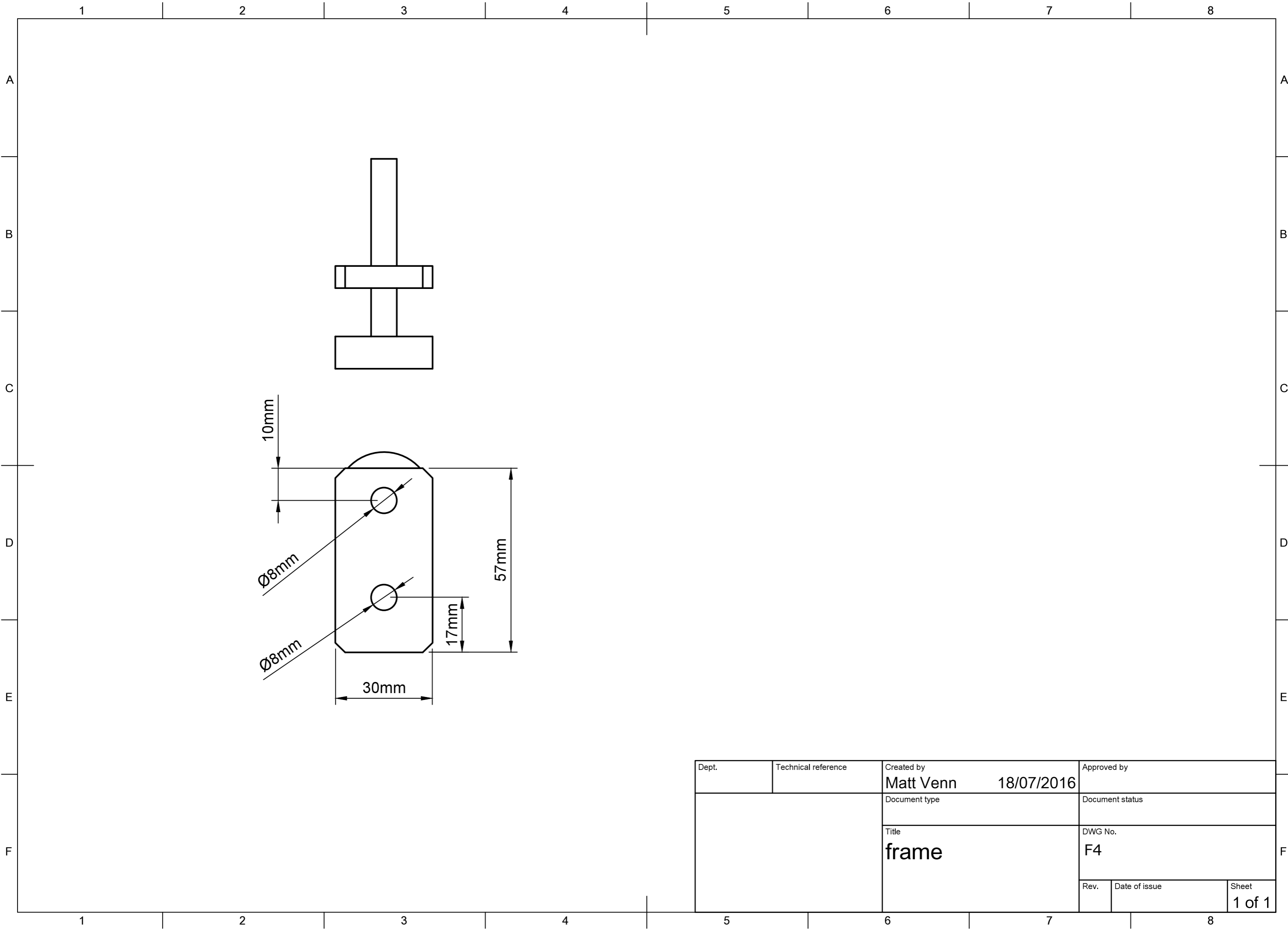
Dept.	Technical reference	Created by matt venn	30/06/2016	Approved by
		Document type	Document status final	
		Title acrylic panels	DWG No. F1	
Rev.	Date of issue	Sheet		
2		1 of 1		



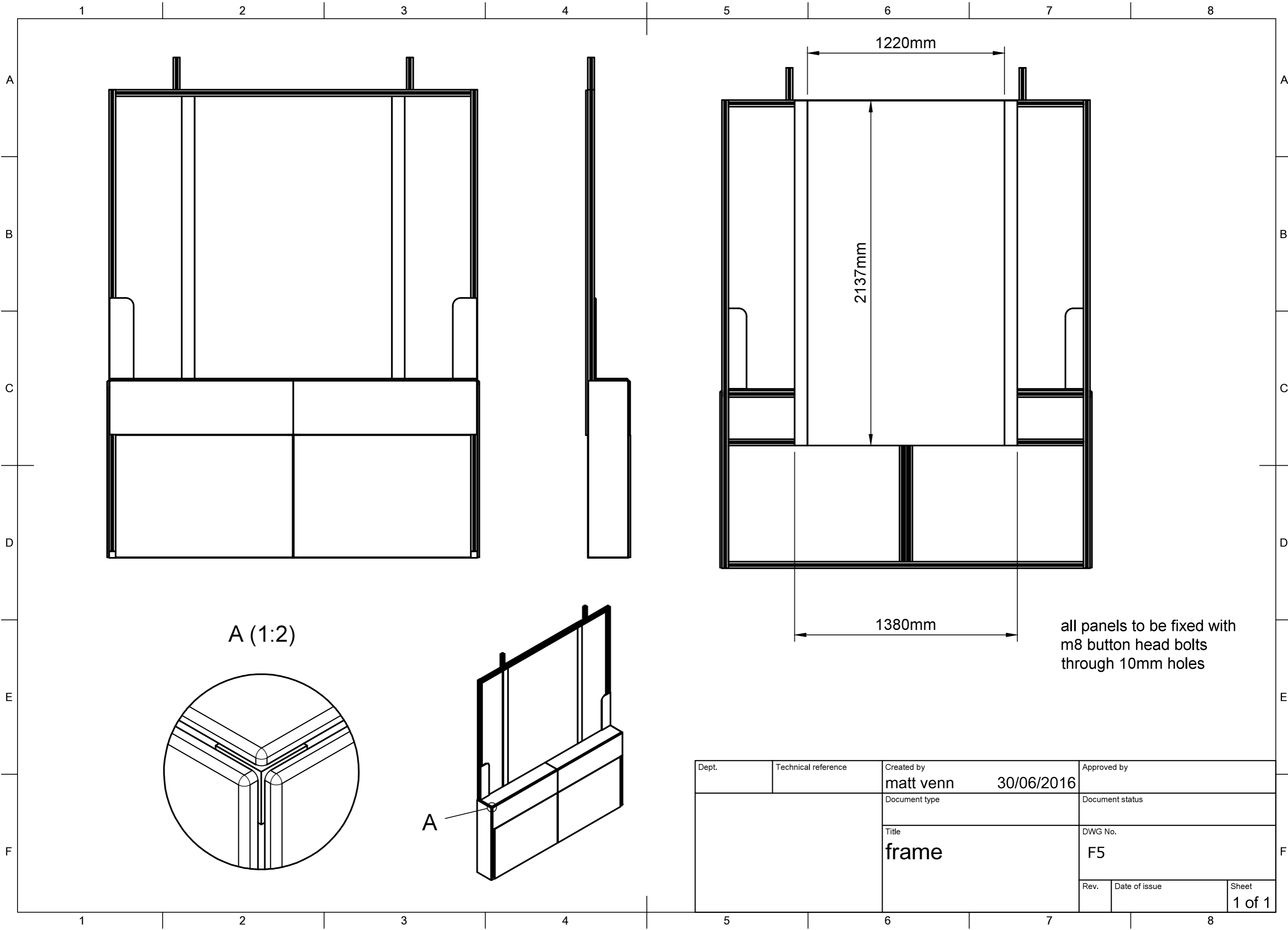
Dept.	Technical reference	Created by Matt Venn	18/07/2016	Approved by
		Document type	Document status	
		Title electronics mount	DWG No. F2	
		Rev.	Date of issue	Sheet 1 of 1



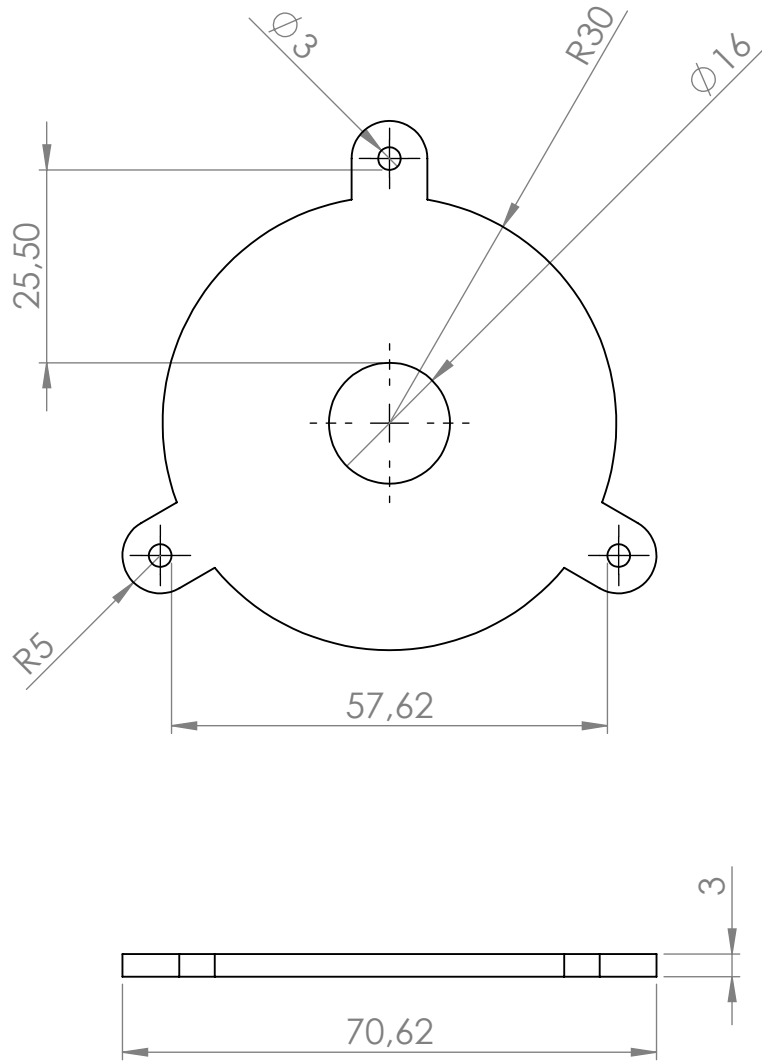
Dept.	Technical reference	Created by Matt Venn	19/07/2016	Approved by
		Document type	Document status	
		Title back panel holes	DWG No. F3	
		Rev.	Date of issue	Sheet 1 of 1




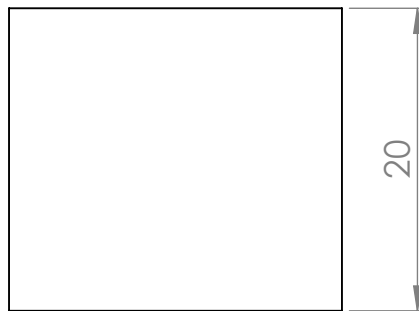
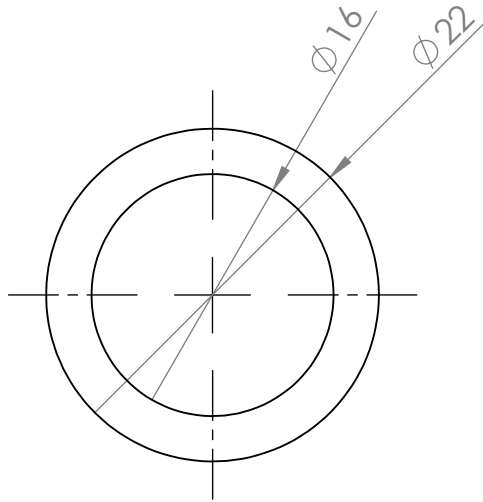
Dept.	Technical reference	Created by Matt Venn	18/07/2016	Approved by
		Document type	Document status	
		Title frame	DWG No. F4	
		Rev.	Date of issue	Sheet 1 of 1




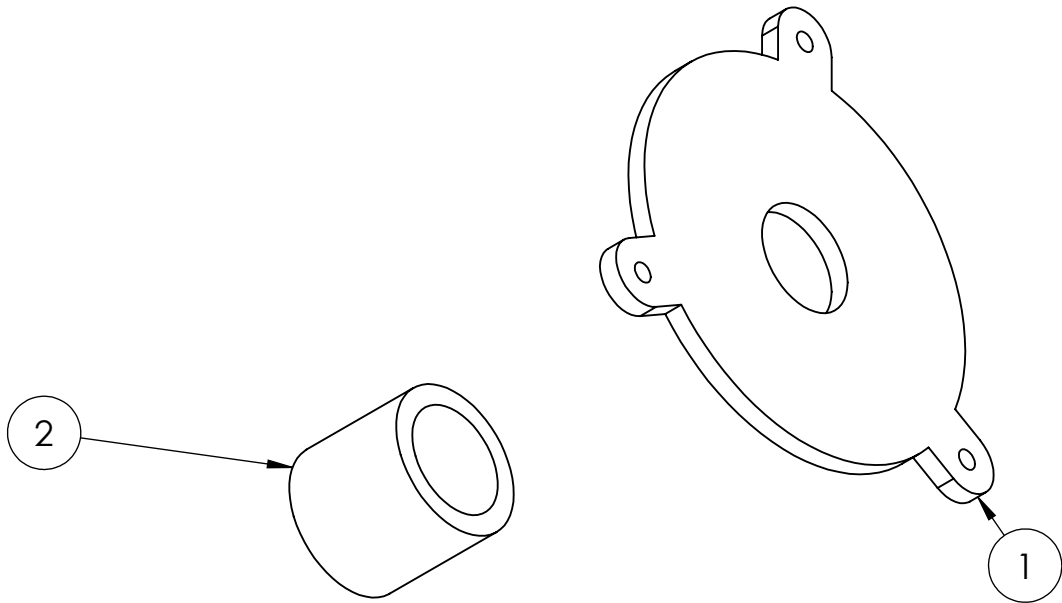
Dept.	Technical reference	Created by matt venn	30/06/2016	Approved by
		Document type	Document status	
		Title frame	DWG No. F5	
Rev.	Date of issue	Sheet		1 of 1




Project Energy MegaDrawz		Date 24/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G1.1
Description Simple_3points_gondola_a : Manufacture with 3mm MDF or acrylic, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

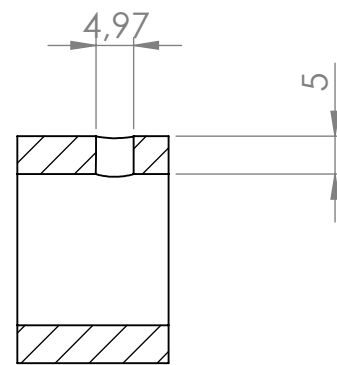
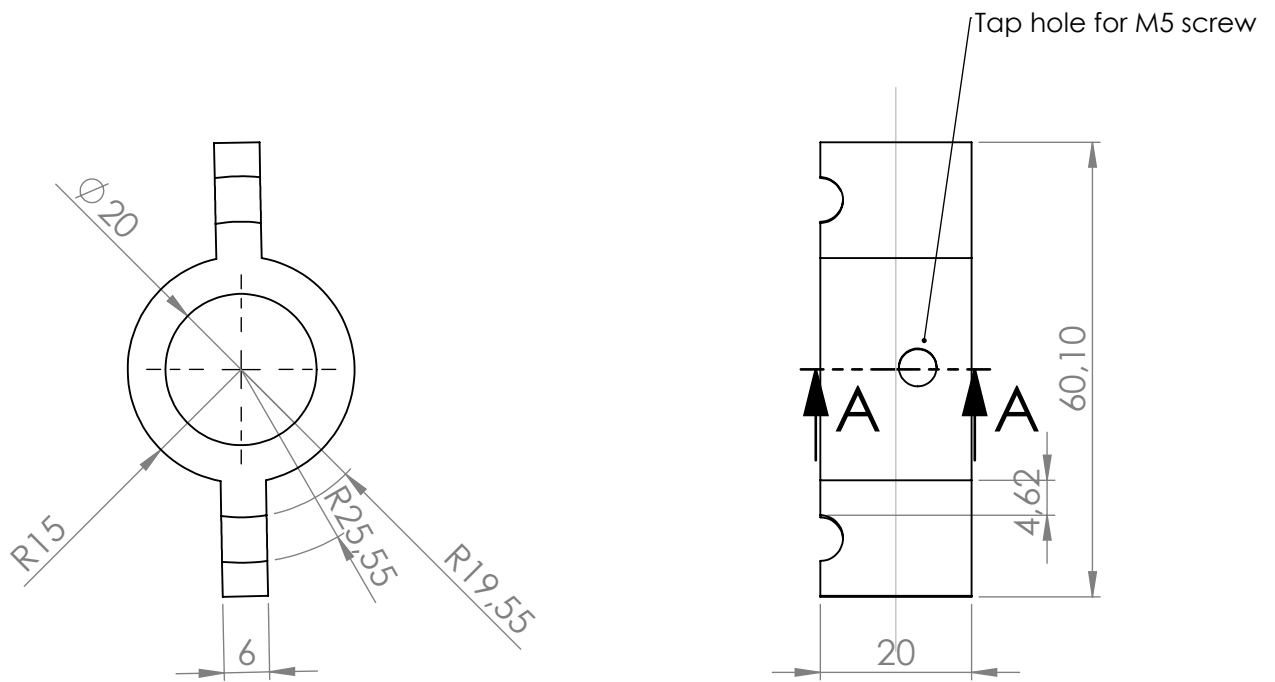


Project Energy MegaDrawz		Date 24/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G1.2
Description Simple_3points_gondola_b: Machine with lathe from an aluminum rod or 3D print with PLA.		Matt Venn Engineering 




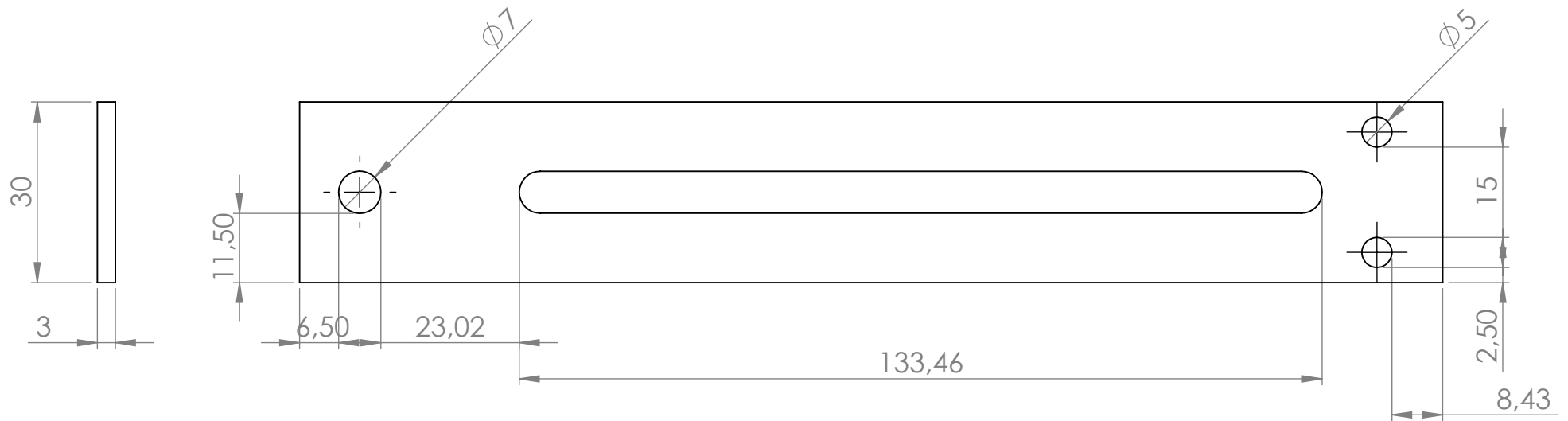
NUMBER	ID	CANTIDAD
1	simple_3points_gondol a_a	1
2	simple_3points_gondol a_b	1


Project Energy MegaDrawz		Date 24/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G1.a
Description Esimple: Use glue for the assembly of part 1 and 2 if is required.		Matt Venn Engineering 

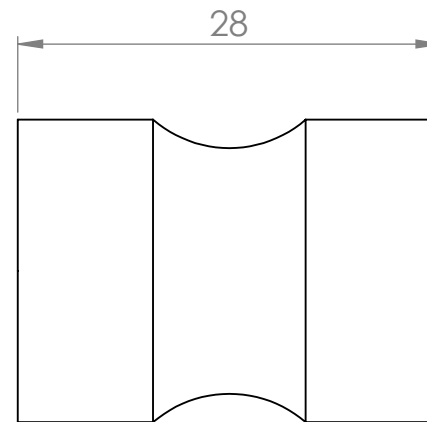
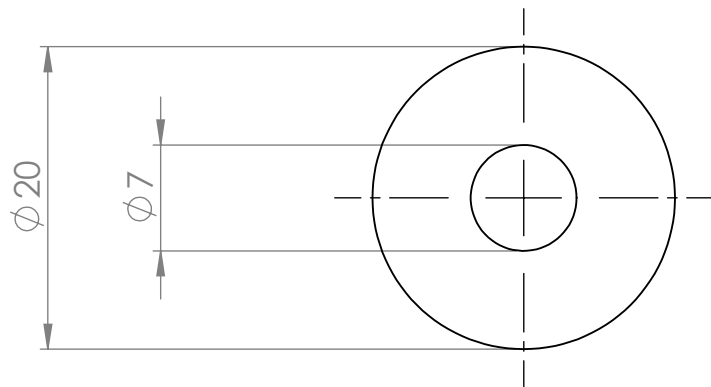



Section A-A

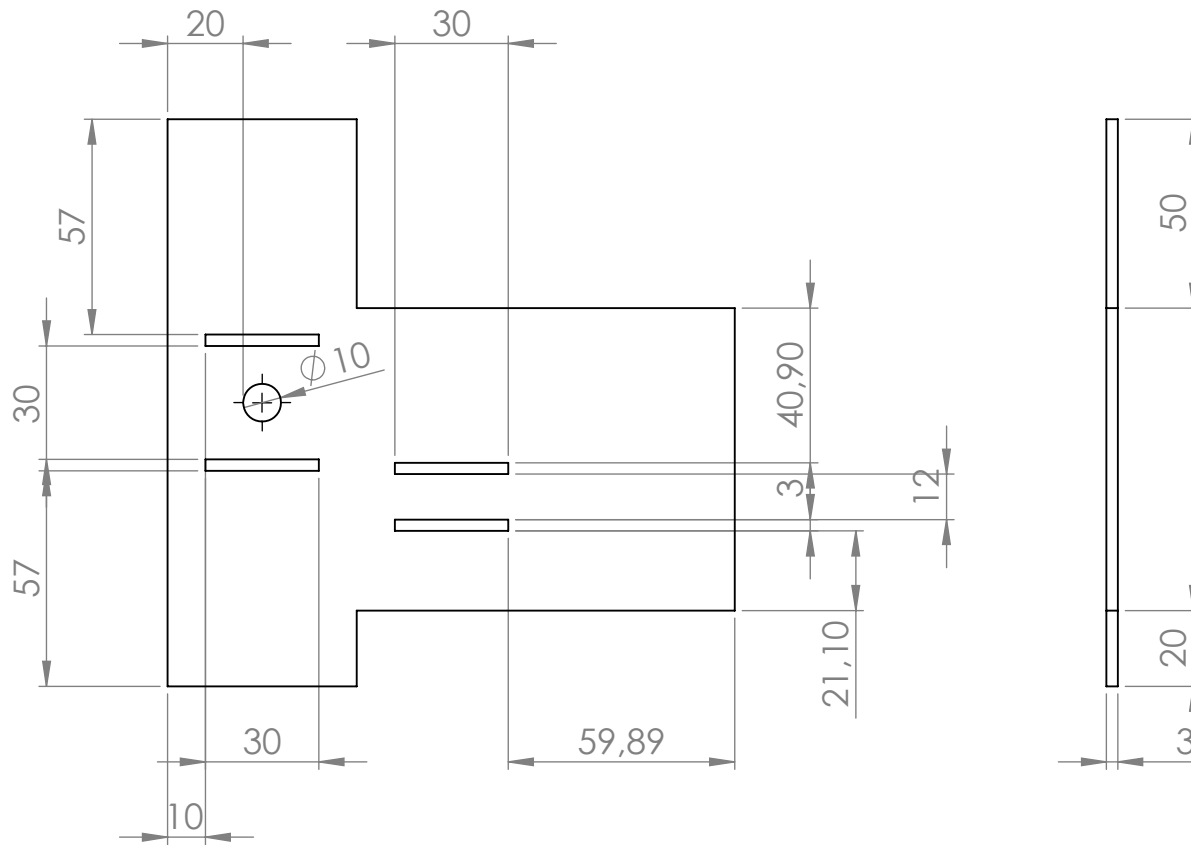
Project Energy MegaDrawz		Date 12/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G.2.1
Description Amarre.sldprt: Manufacture with PLA with a 3D printer. Hole of diameter 4,97 must be tapped for a M5 screw.		Matt Venn Engineering 




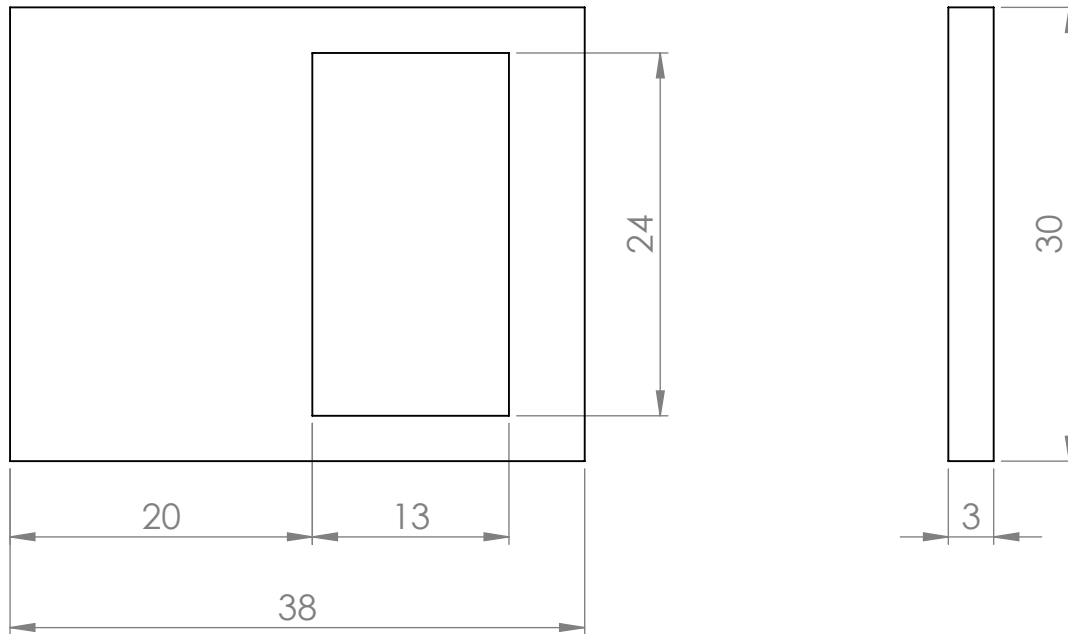
Project Energy MegaDrawz		Date 12/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G.2.2
Description Pilar: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1 mm. Verify dimensions thickness for a push fit.		Matt Venn Engineering 




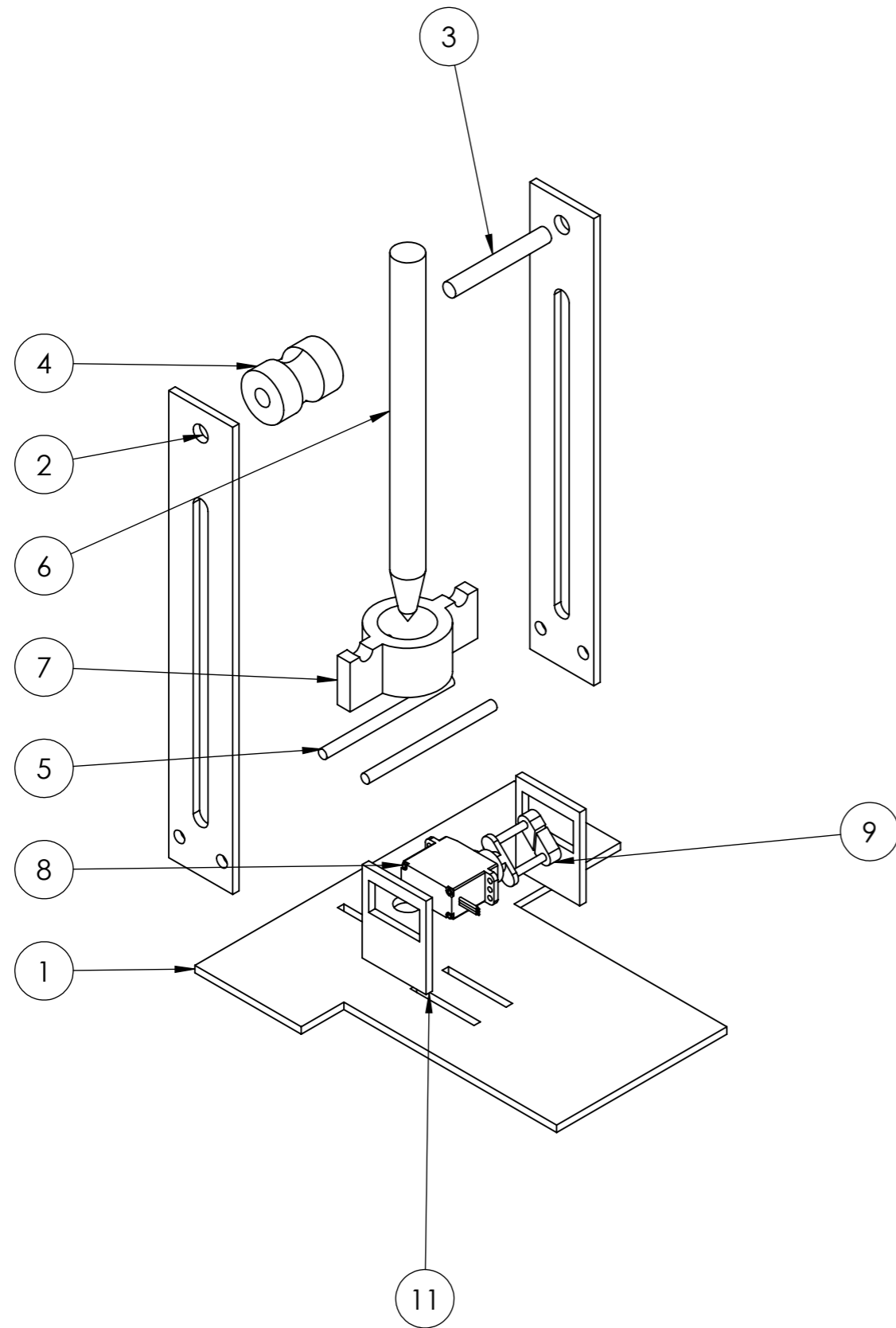
Project Energy MegaDrawz		Date 12/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G2.3
Description Pulley: Manufacture with PLA and a 3D printer. Internal diameter shall be verify. Can be replaced with a BNL acetal pulley of similar dimensions.		Matt Venn Engineering 




Project Energy MegaDrawz		Date 12/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G2.4
Description Base: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Verify dimensions thickness and slots for a push fit.		Matt Venn Engineering 

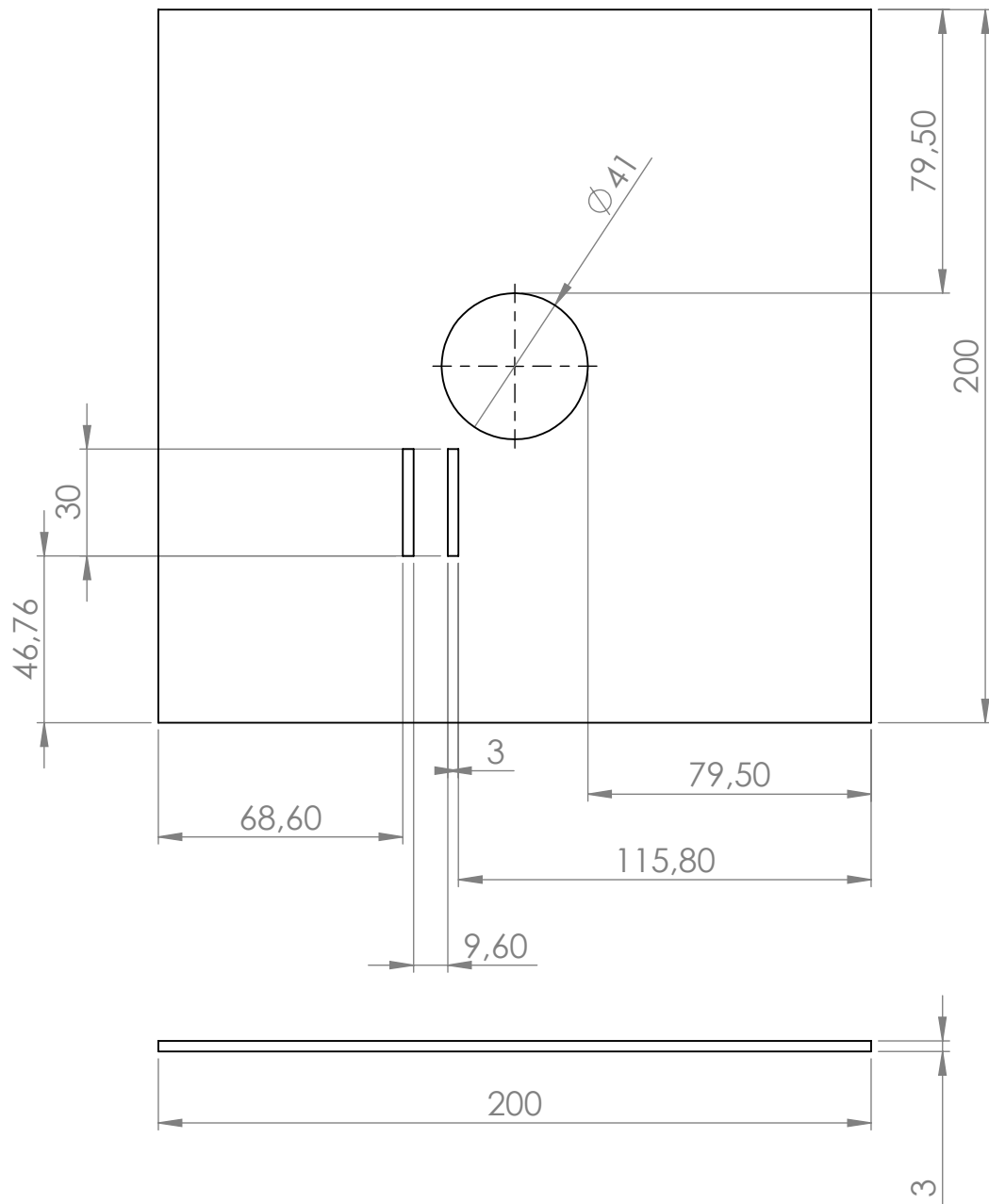



Project Energy MegaDrawz		Date 12/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G2.5
Description Soporteservo1: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Verify dimensions and thickness for a push fit.		Matt Venn Engineering 

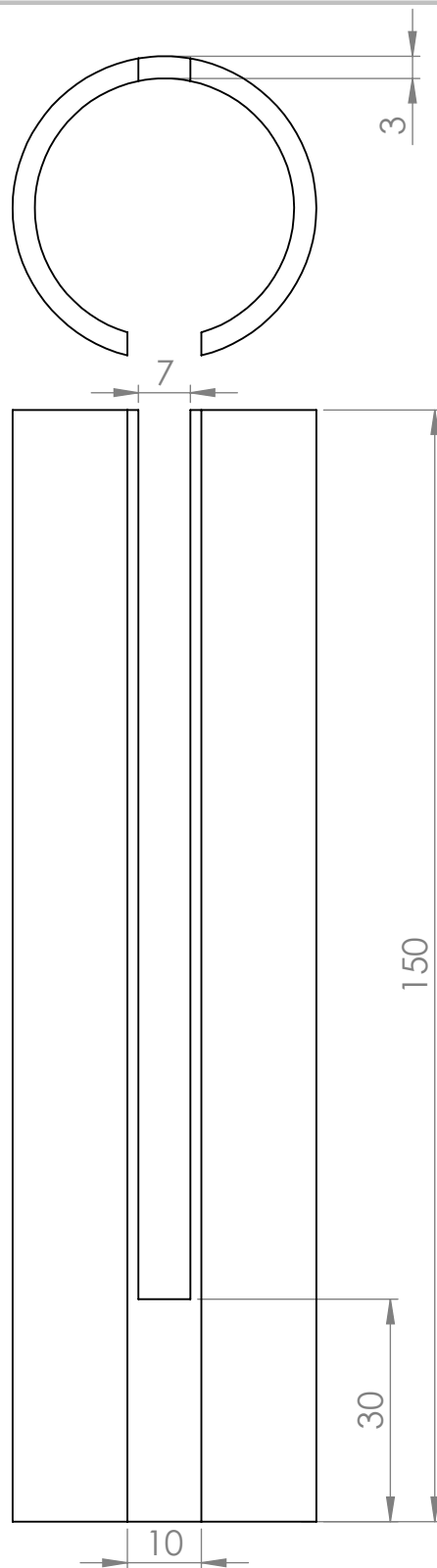



NUMBER	ID	QUANTITY
1	base	1
2	pilar	2
3	rod6x45	1
4	pulley	1
5	rod4x60	2
6	pen	1
7	amarre	1
8	SM-S2309S	1
9	SM-S2309S-Arm	1
10	SM-S2309S-Screw	1
11	soporteservo1	2

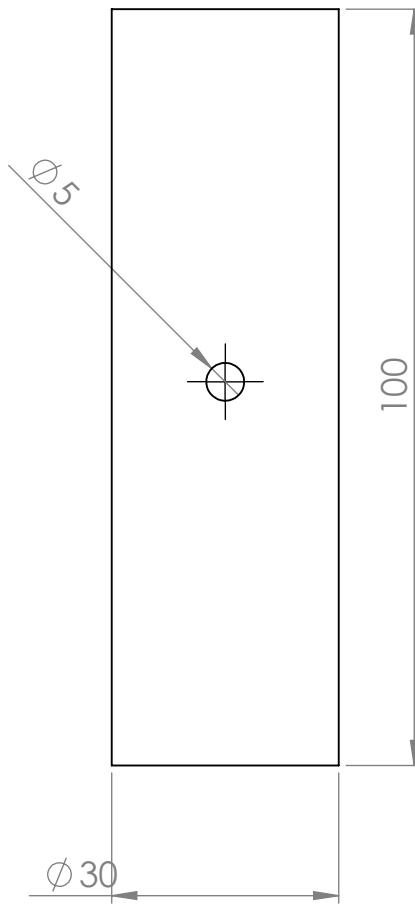
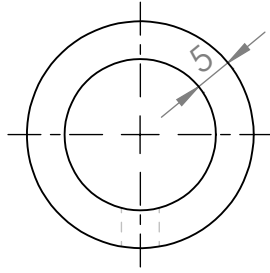
Project Energy MegaDrawz		Date 12/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G2.a
Description Ensamblaje: If push fits are not good enough use wood glue. Rods 3 and 5 can need nuts and washers for stability. Tests place springs between 5 and 7.		Matt Venn Engineering 




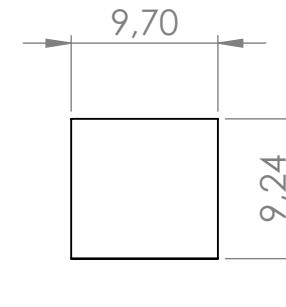
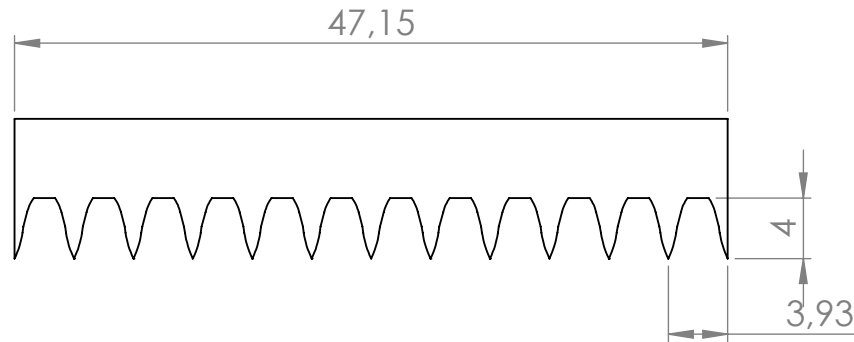
Project Energy MegaDrawz		Date 28/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G3.1
Description Base_20x20: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




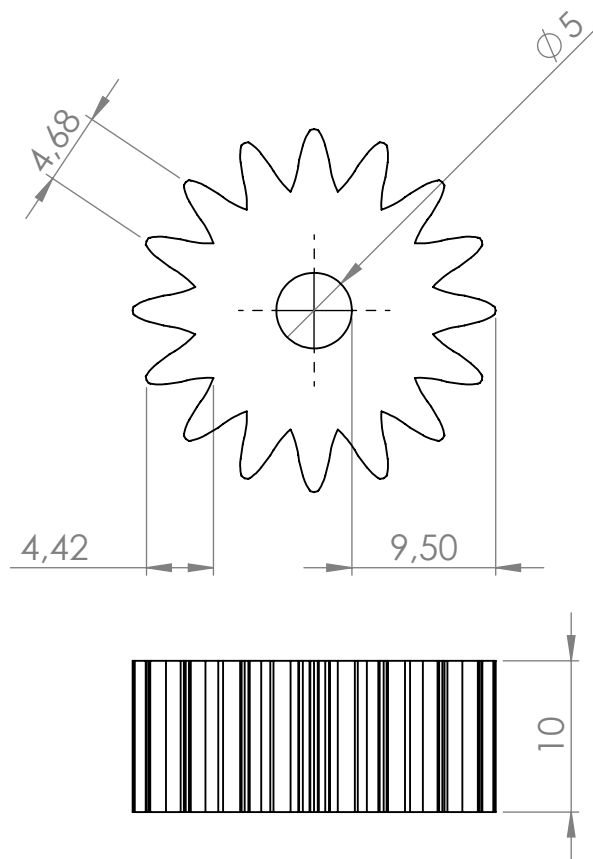
Project Energy MegaDrawz		Date 24/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G3.2
Description Cylinder: use a 3D printer with PLA for manufacture. Verify width of slots		Matt Venn Engineering 




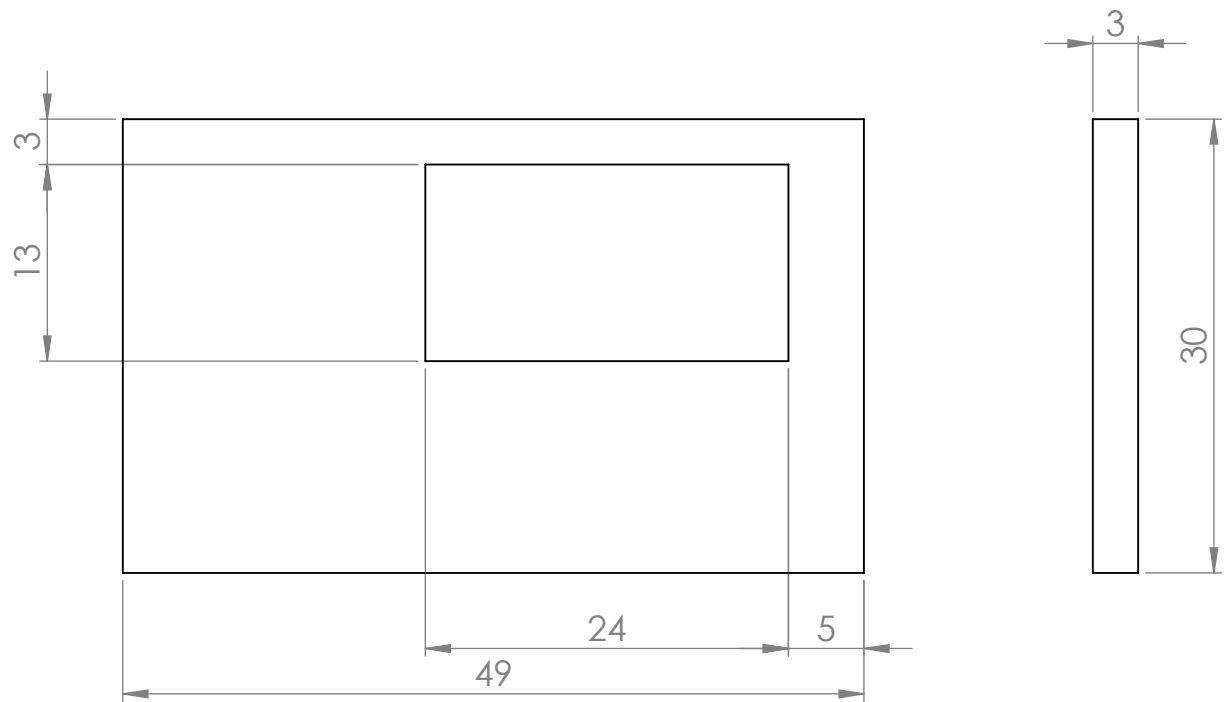
Project Energy MegaDrawz		Date 28/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G3.3
Description Cilindrocentral: 3D print with PLA or use an acrylic cylinder of the given dimensions.		Matt Venn Engineering 




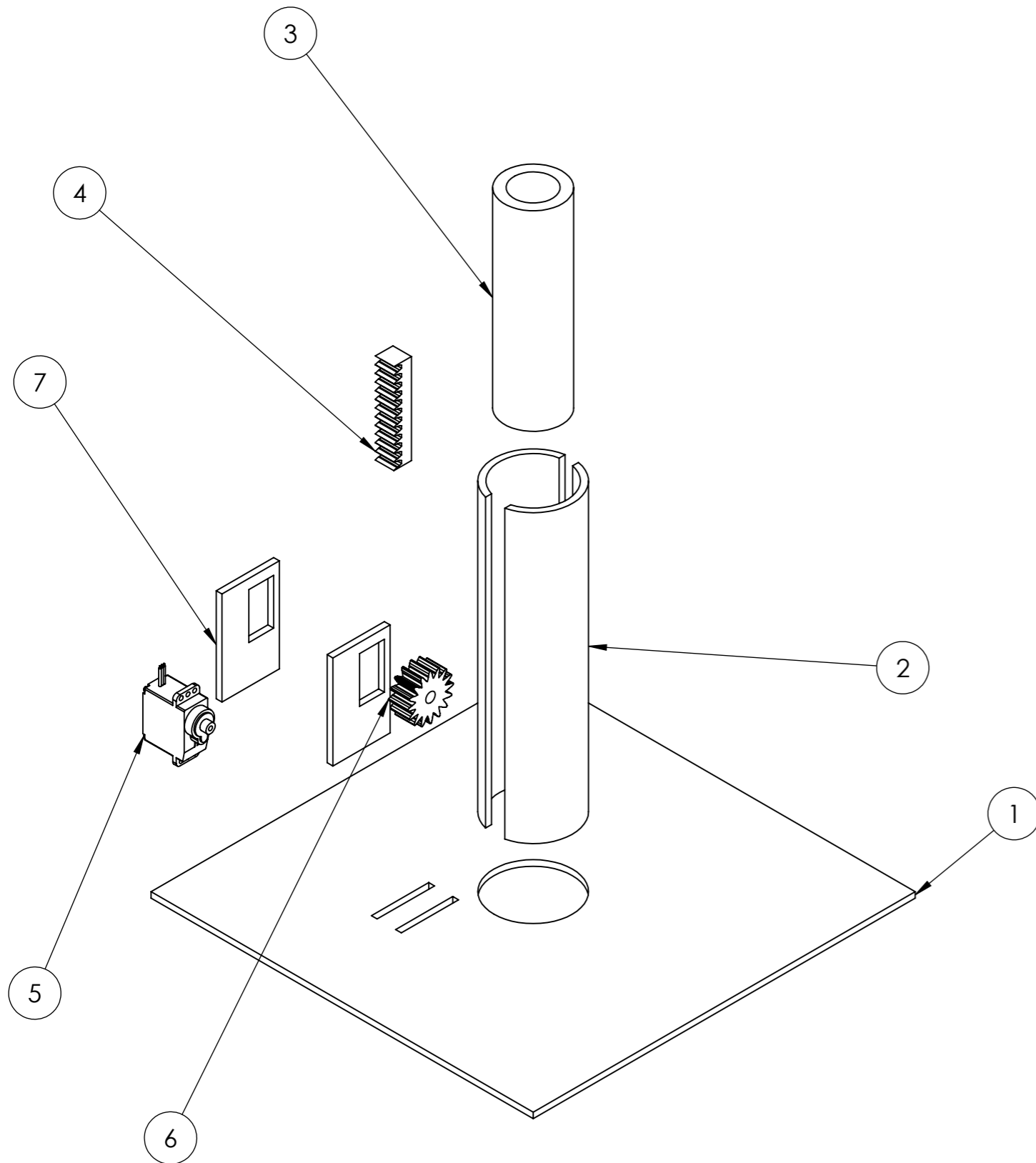
Project Energy MegaDrawz		Date 28/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G3.4
Description Cremallera: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Thickness shall be accurate in ± 1 mm		Matt Venn Engineering 




Project Energy MegaDrawz		Date 28/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G3.5
Description Piñon: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

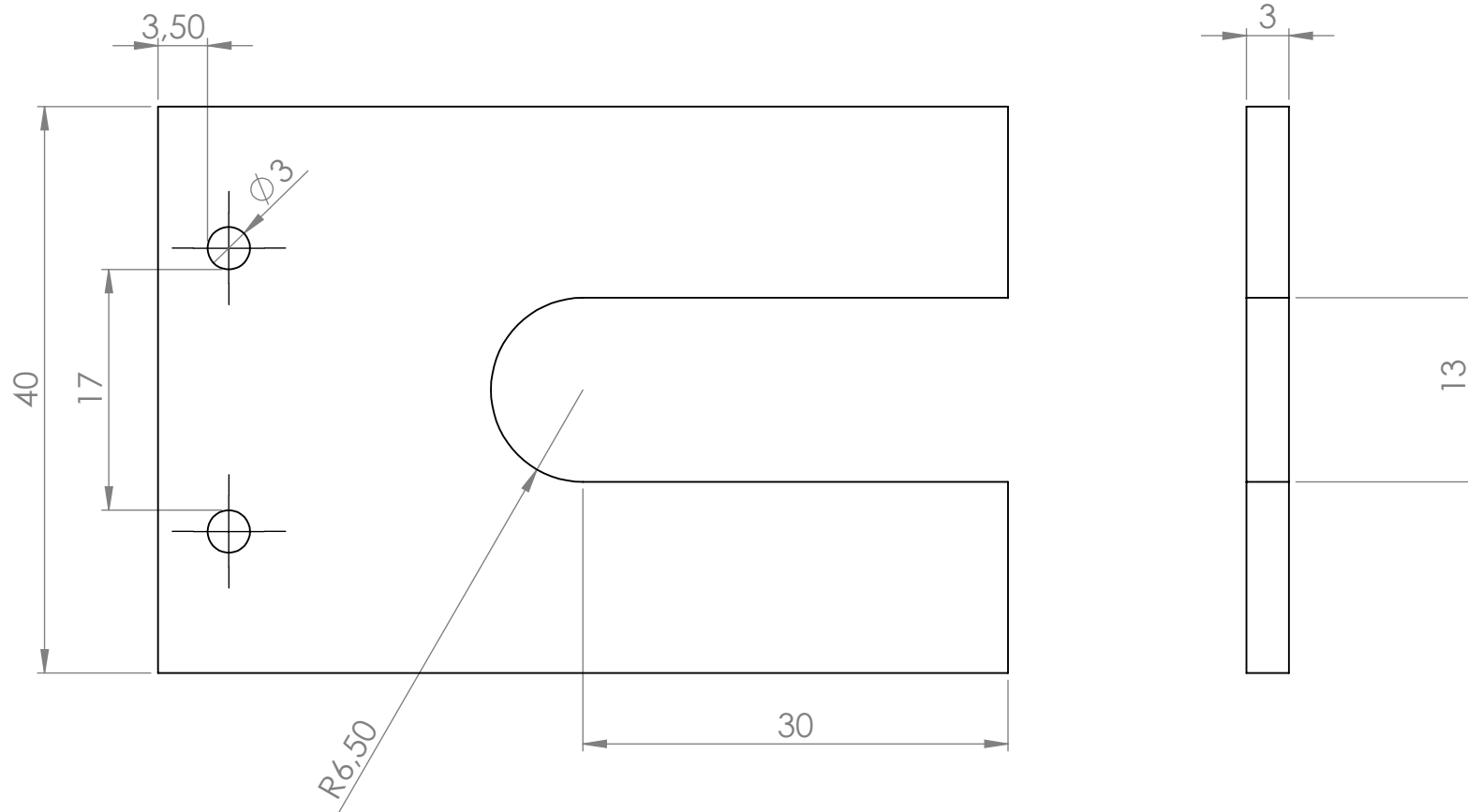



Project Energy MegaDrawz		Date 24/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G3.6
Description Soporteservo2: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

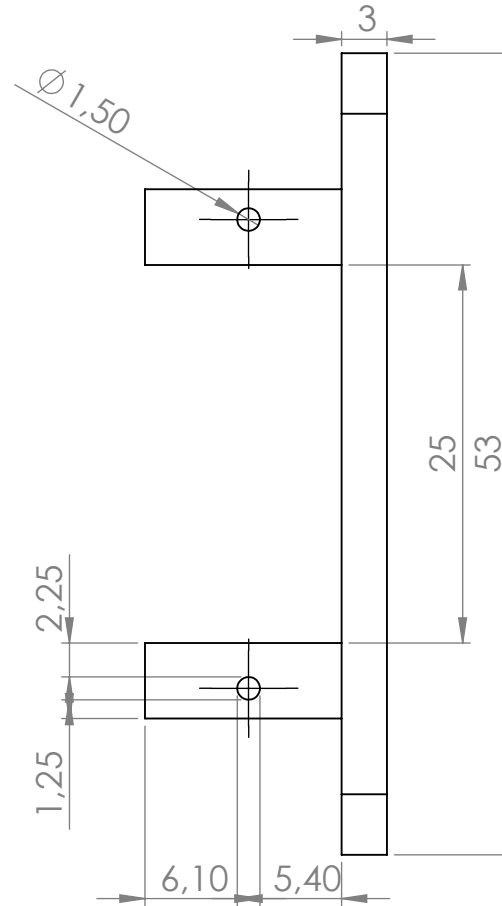
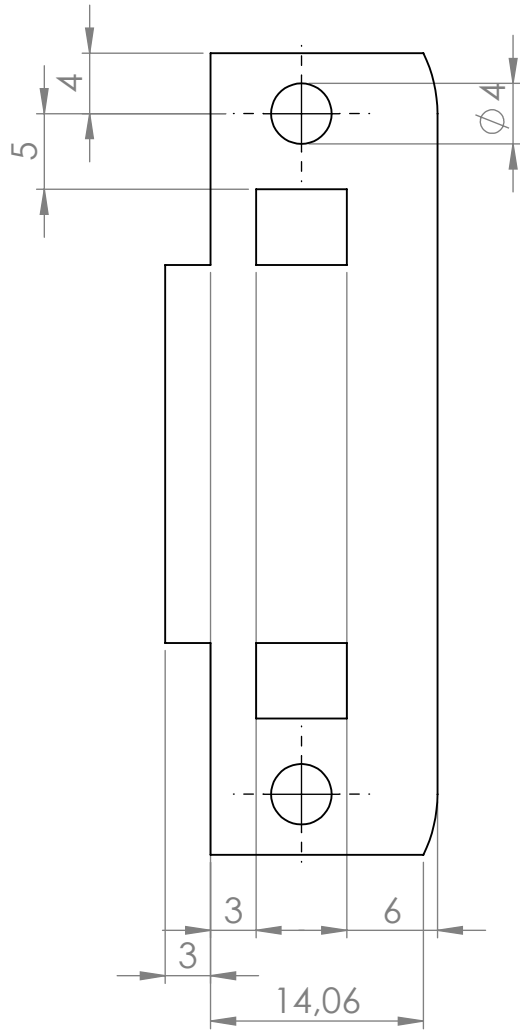



NUMBER	ID	QUANTITY
1	base_20x20	1
2	cilinder	1
3	cilindrocentral	1
4	cremallera	1
5	SM-S2309S	1
6	piñon	1
7	soporteservo2	2

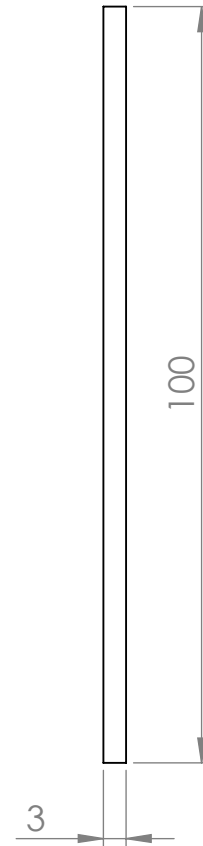
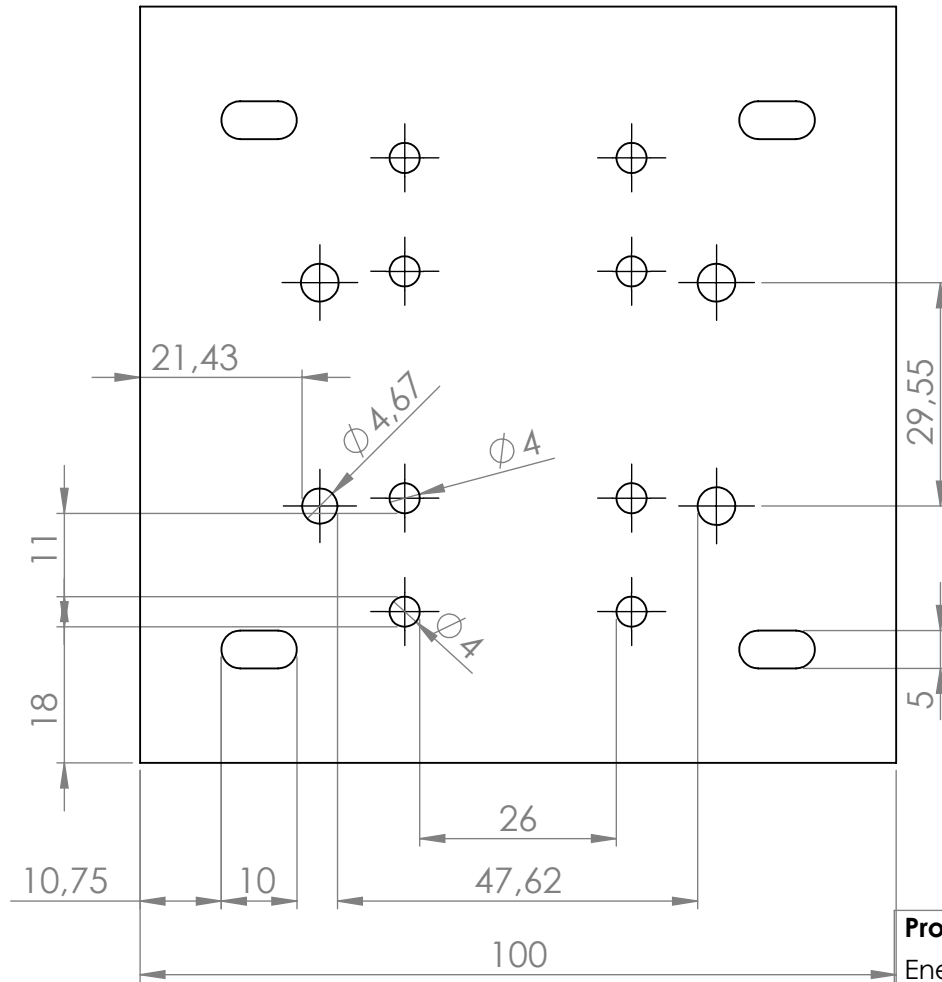
Project Energy MegaDrawz		Date 24/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G3.a
Description Eprototype1: Glue component 2 and 1. Extra screw is required for component 3. Component 4 can be attached with zip tires. Glue component 7 to 1 if it is required.		Matt Venn Engineering 




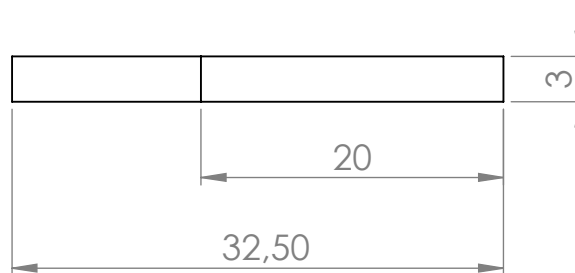
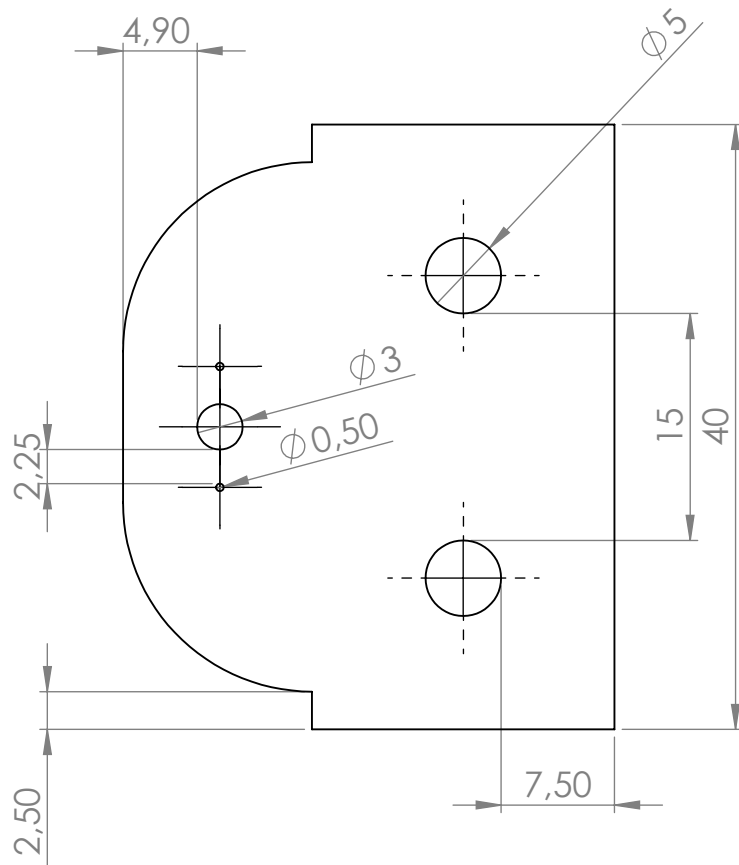
Project Energy MegaDrawz		Date 01/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G4.1
Description Guialeva: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




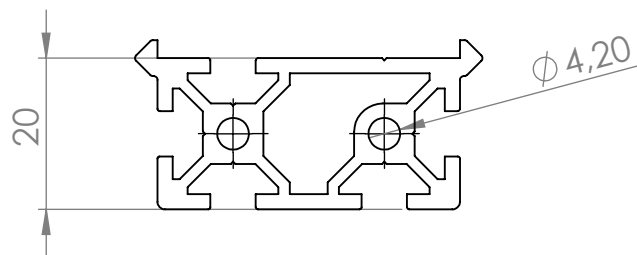
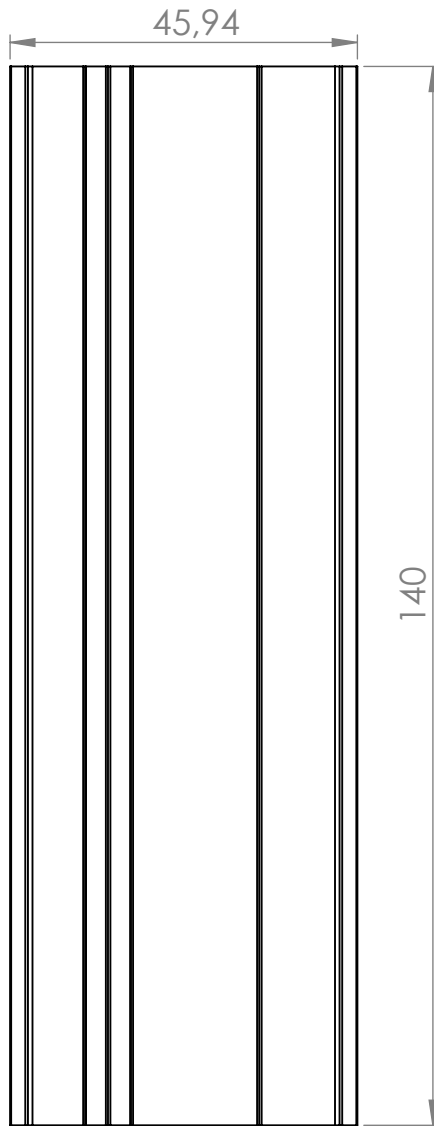
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G4.2
Description Soporteservov2: Manufacture with PLA with a 3D printer.		Matt Venn Engineering 




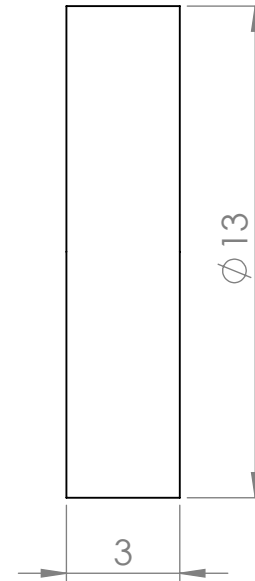
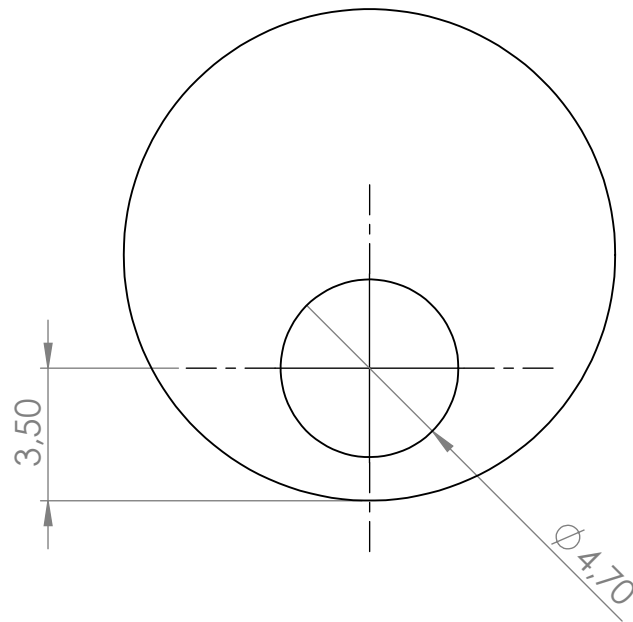
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G4.3
Description Soporterodamientos: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




Project Energy MegaDrawz		Date 30/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G4.4
Description Partesuperior: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

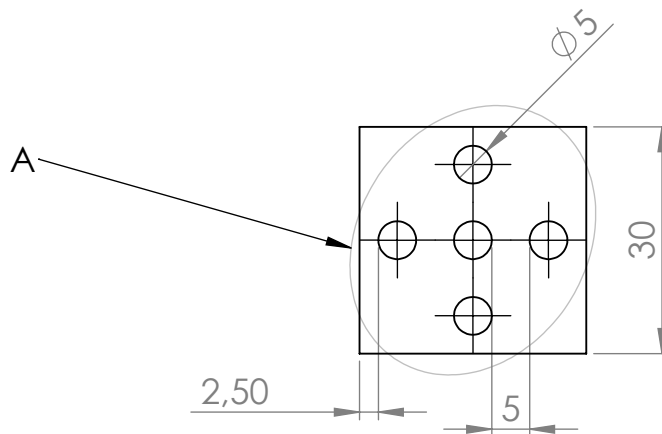
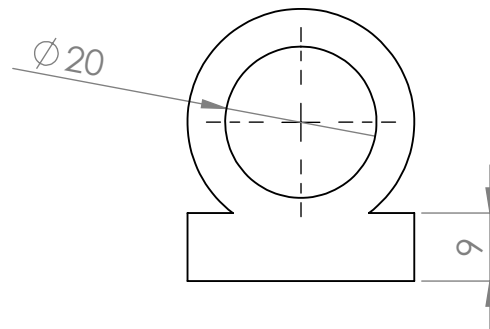
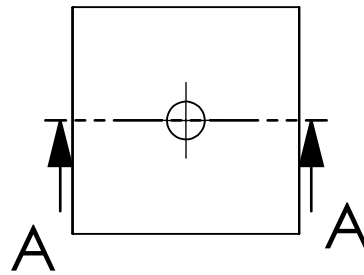
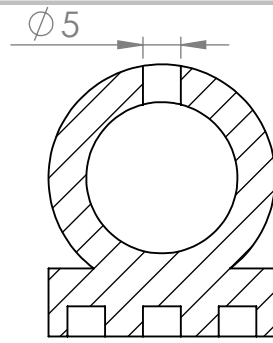



Project Energy MegaDrawz		Date 26/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G4.5
Description Maker slide400: MakerSlide is an open source hardware profile made o aluminum. Tis project is limited to make use of it		Matt Venn Engineering 

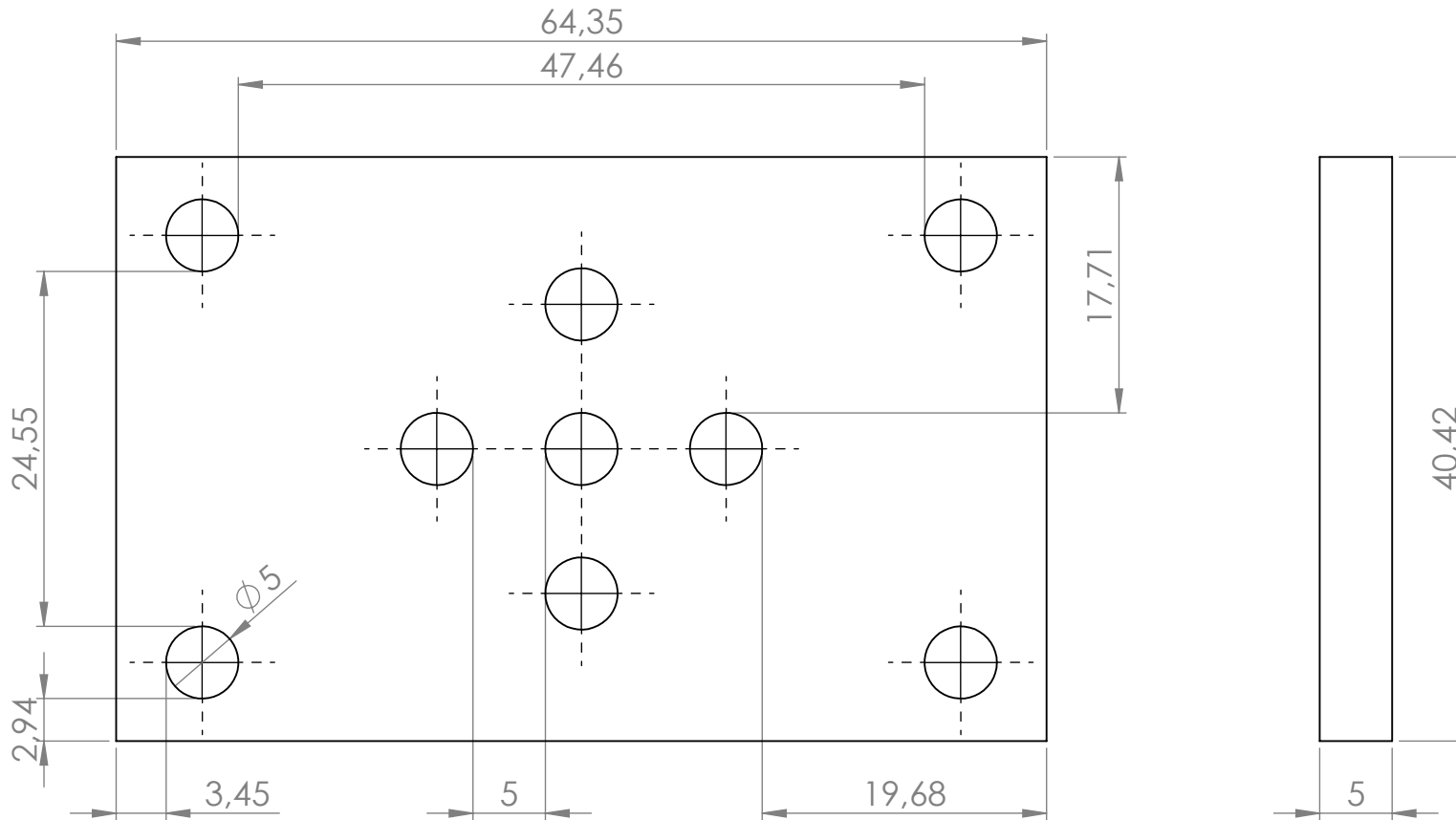



Project Energy MegaDrawz		Date 31/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 5:1
Approval Matthew Venn	Signature	Plan number G4.6
Description Leva_capa0: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Cut from aluminum rod and machine hole with tower driller		Matt Venn Engineering 

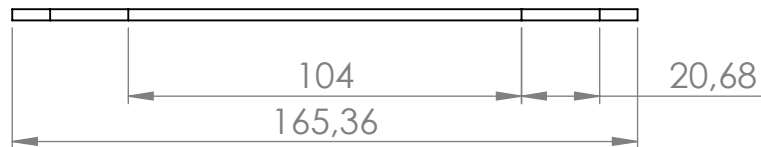
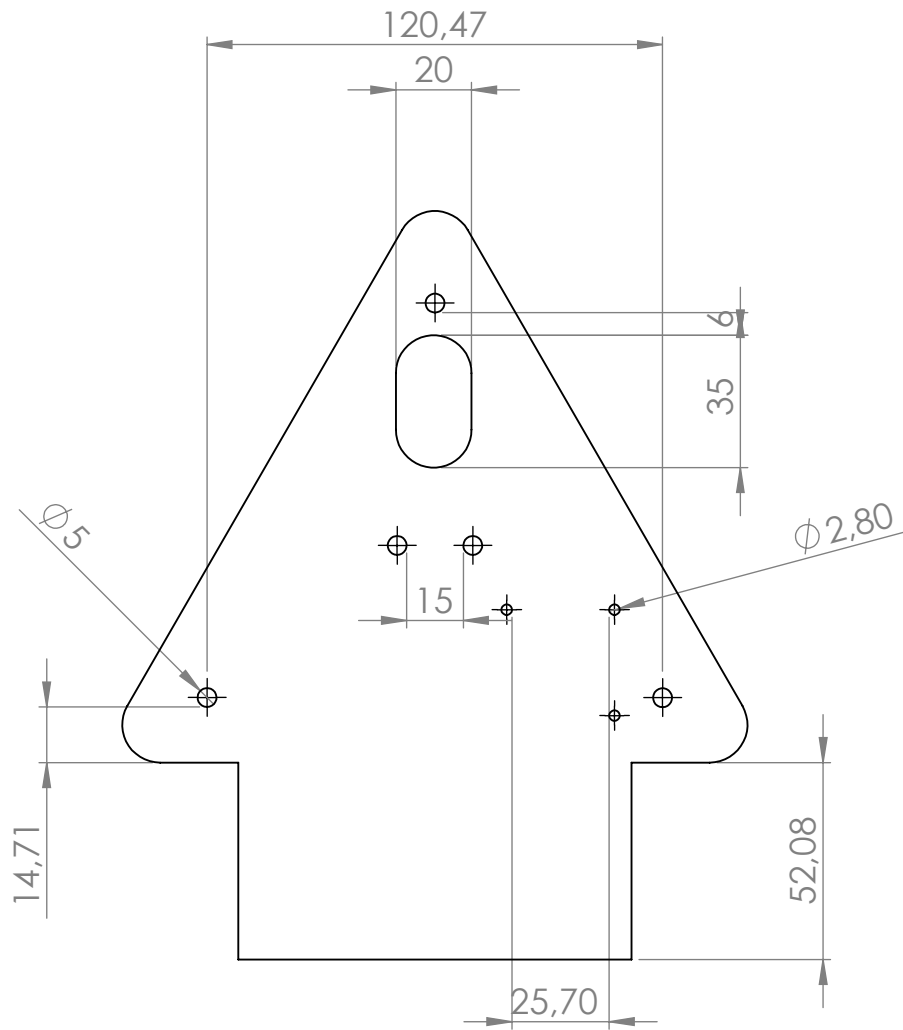
SECTION A-A




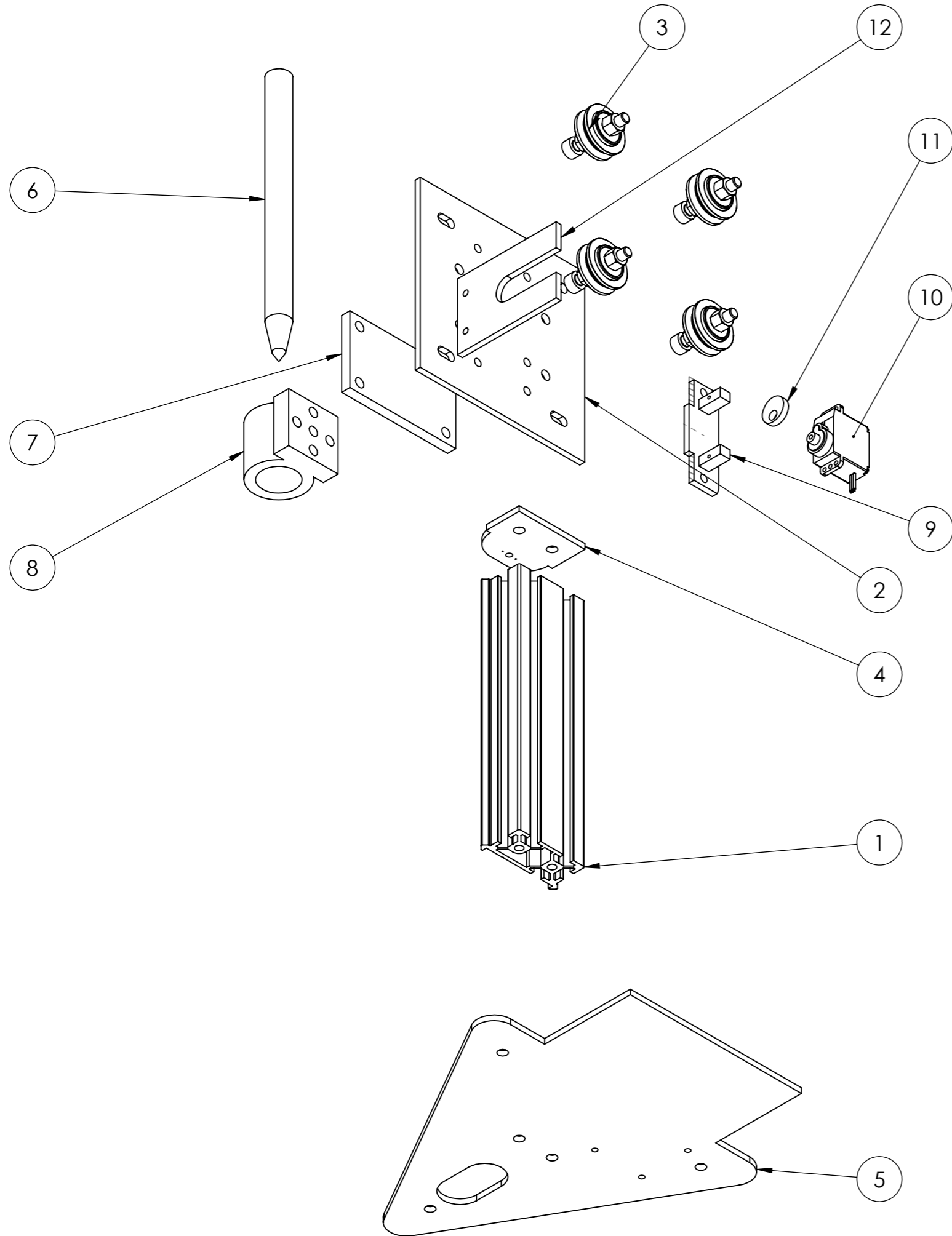
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G4.7
Description Bolisoportemagnético: Produce with PLA and 3D printed. Holes of 2.5mm of section A must have a magnet inside, use a soldering iron for heat them until 250°C and push them until be aligned with the base. Remember place them in the opposite polarity that in the base.		Matt Venn Engineering 




Project Energy MegaDrawz		Date 30/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G4.8
Description Basesoportemagnético: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

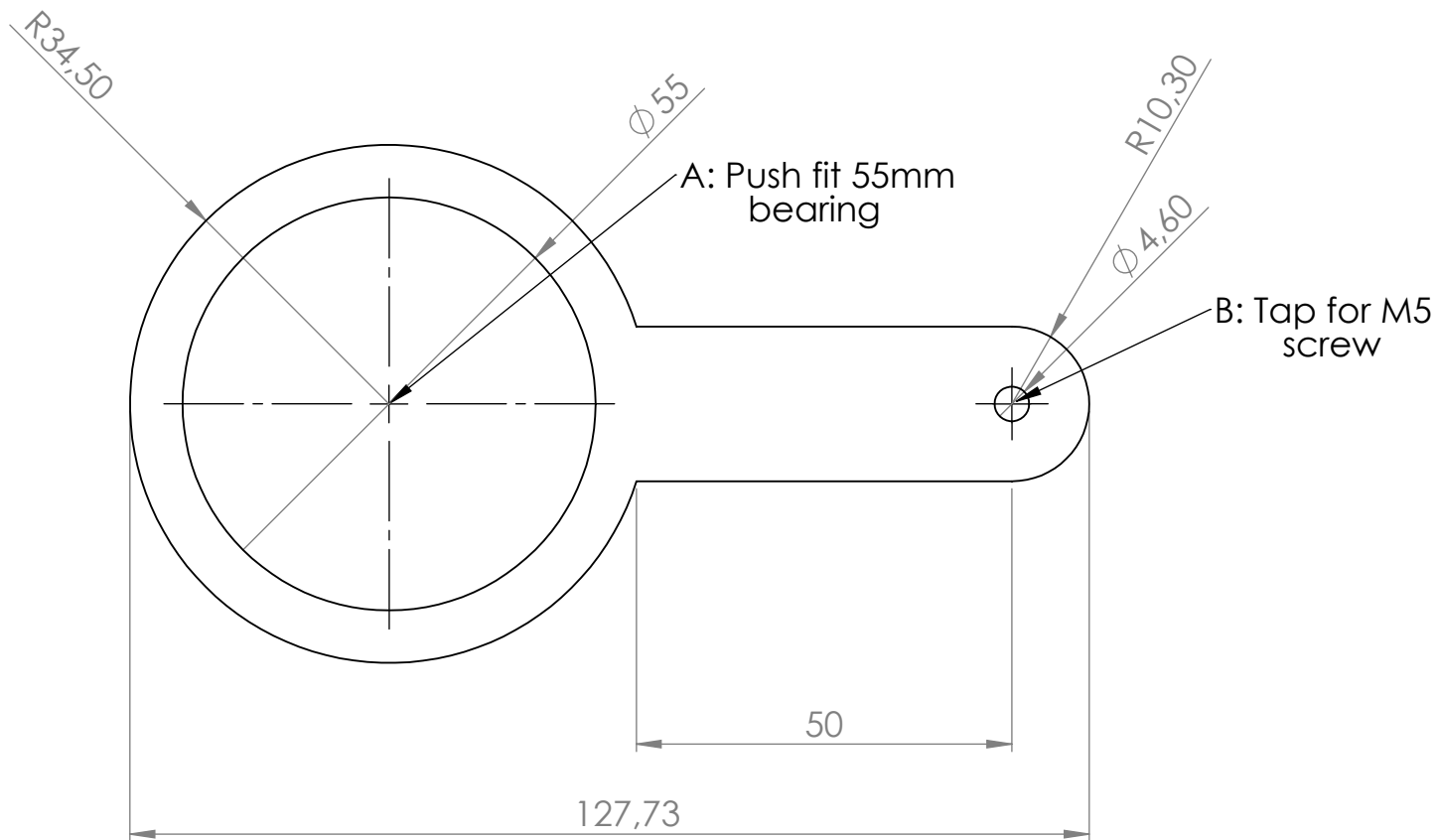



Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G4.9
Description Basegondola_v2: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

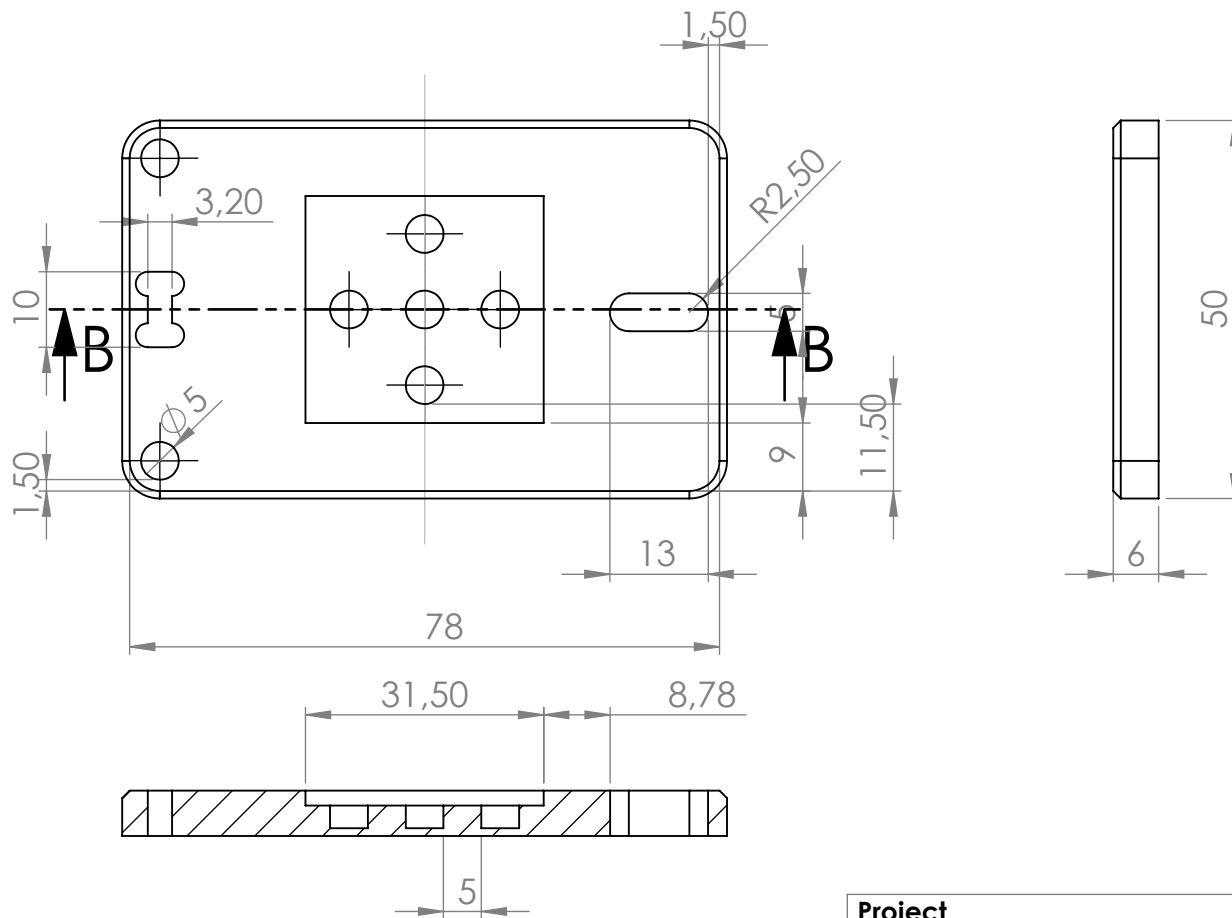


NUMBER	ID	QUANTITY
1	maker slide 140mm	1
2	soporterodamientos	1
3	G5.sa.2	4
4	partesuperior	1
5	basegondola_v2	1
6	pen1	1
7	basesoportemagnetic o	1
8	bolisoportemagnetico	1
9	soporteservov2	1
10	SM-S2309S	1
11	leva_capa0	1
12	guialeva	1


Project Energy MegaDrawz		Date 26/05/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number G4.a
Description Esliderprot: tap holes when it is required. Extra screws can be needed. Verify push fit between 11 and 10. Piece 9 shall be review for next version.		Matt Venn Engineering 

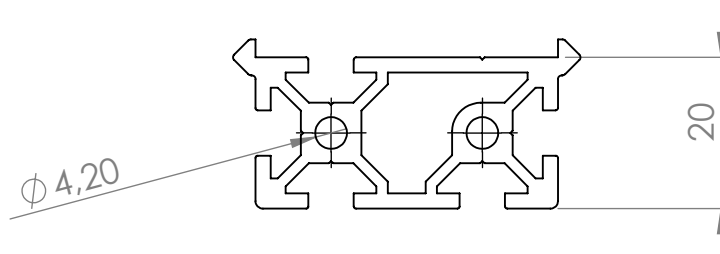
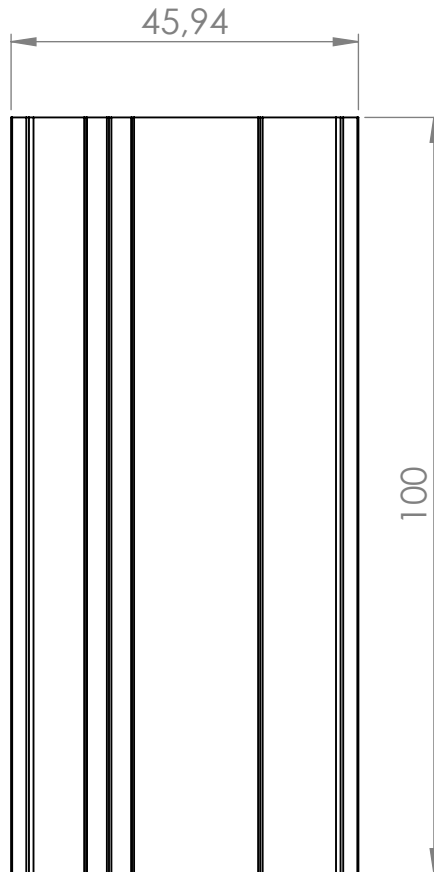



Project Energy MegaDrawz		Date 07/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G5.1
Description Wings_bearings_55mm: Laser cut from MDF from 3 to 6 mm. Note that actual values are adjusted for obtain a push fit with a laser cutter of 0.1mm. For machine it with delrin adjust diameter of circumference A to 55 mm. Tap hole B as indicated.		Matt Venn Engineering 

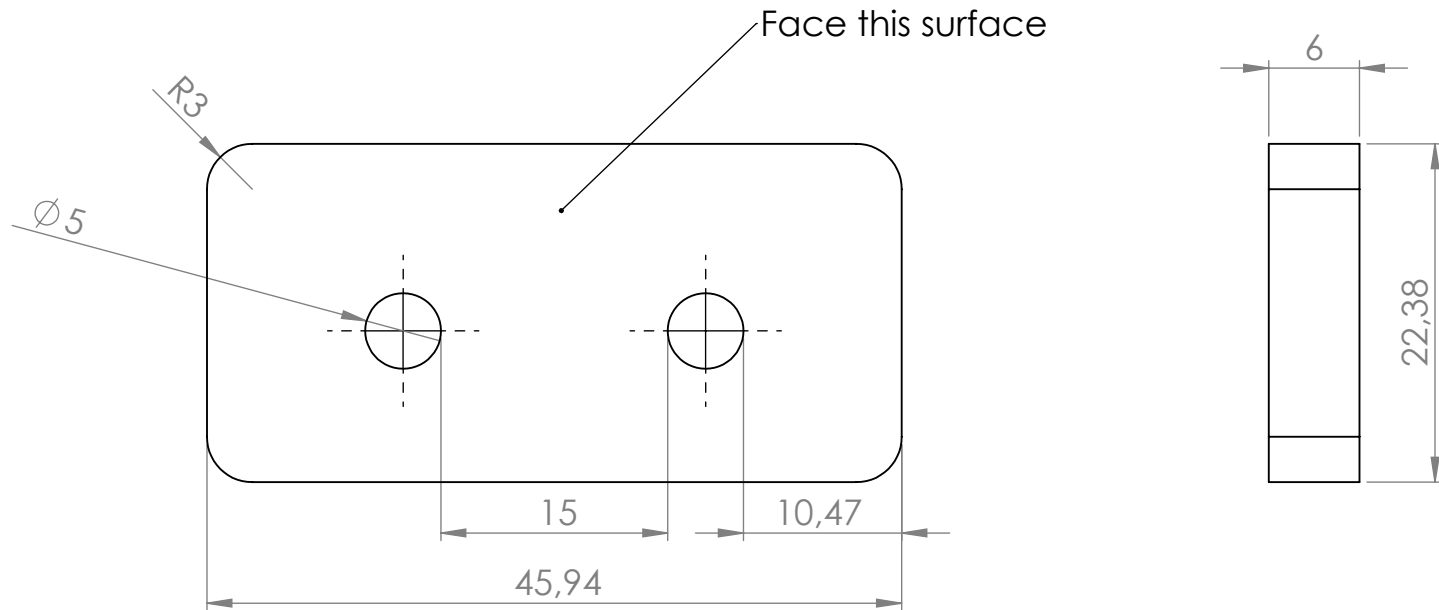



SECTION B-B

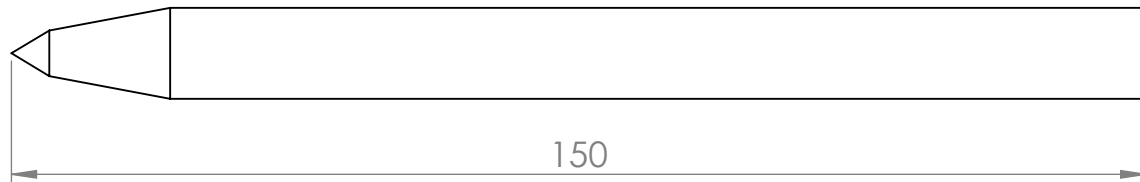
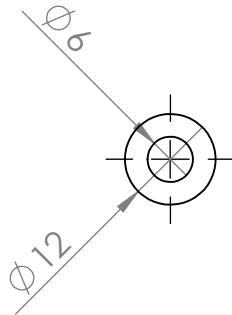
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G5.2
Description Soporterodamientosv2: Machine with 6mm delrin plate, if isrequired face top surface. Chamfer is optional.Modify corners of the socket when CNC drilling machine trajectory is generated. Corners can be exeeded for guaranty a good fit. Modify CAD in next revision.		Matt Venn Engineering 




Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G5.3
Description Maker slide100: MakerSlide is an open source hardware profile made of aluminum. This project is limited to make use of it.		Matt Venn Engineering 

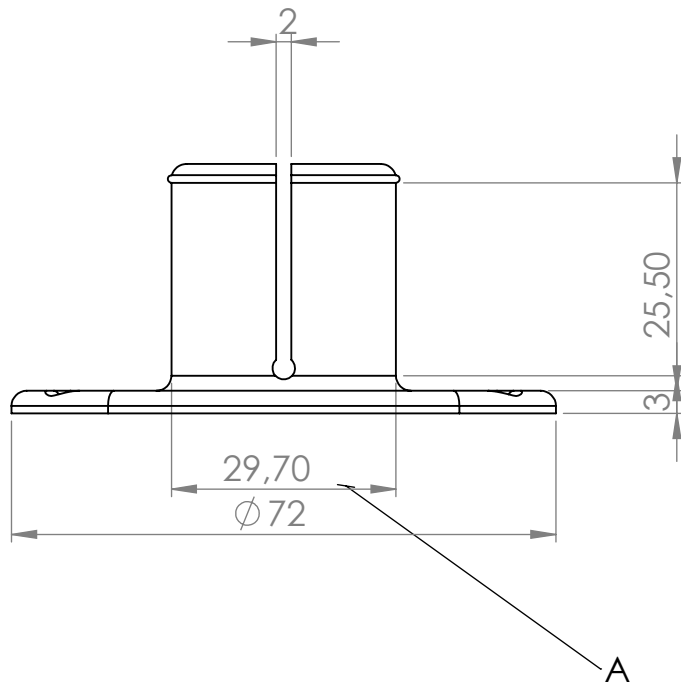
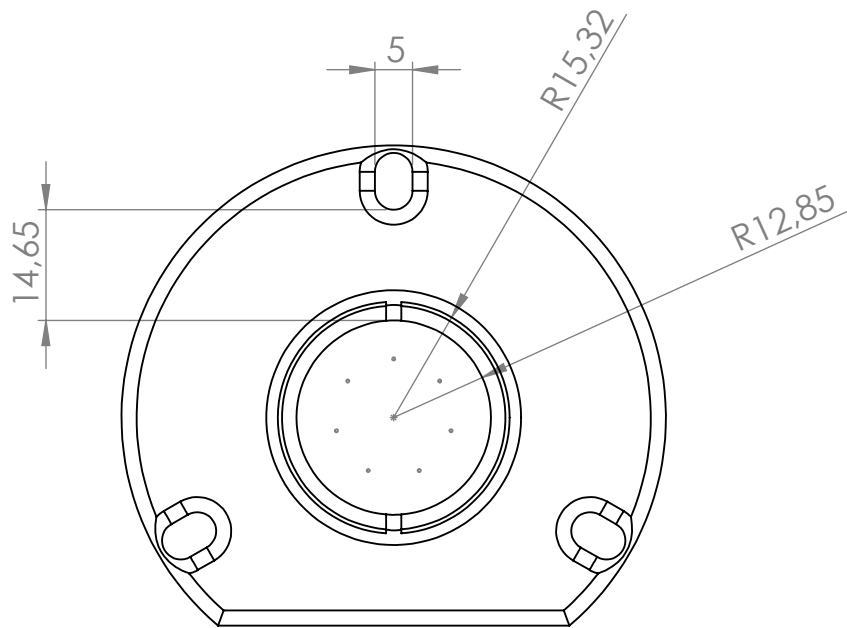



Project Energy MegaDrawz		Date 11/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G5.4
Description Slidertop_cover: Use a CNC drilling machine and 6mm plate of delrin for manufacture this component. Face top surface.		Matt Venn Engineering 

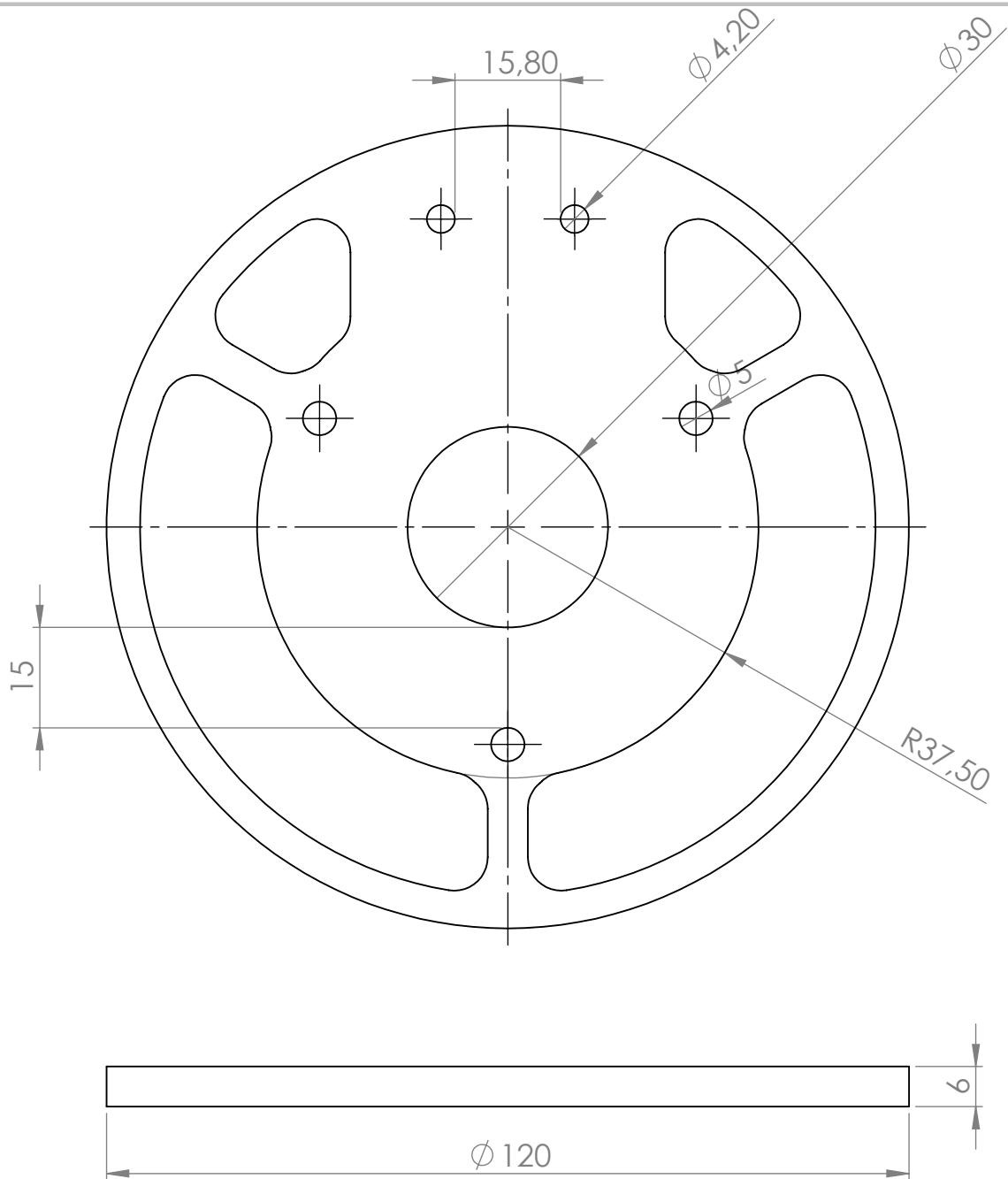



Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G5.5
Description Pen1: Use only as reference for design other components.		Matt Venn Engineering 

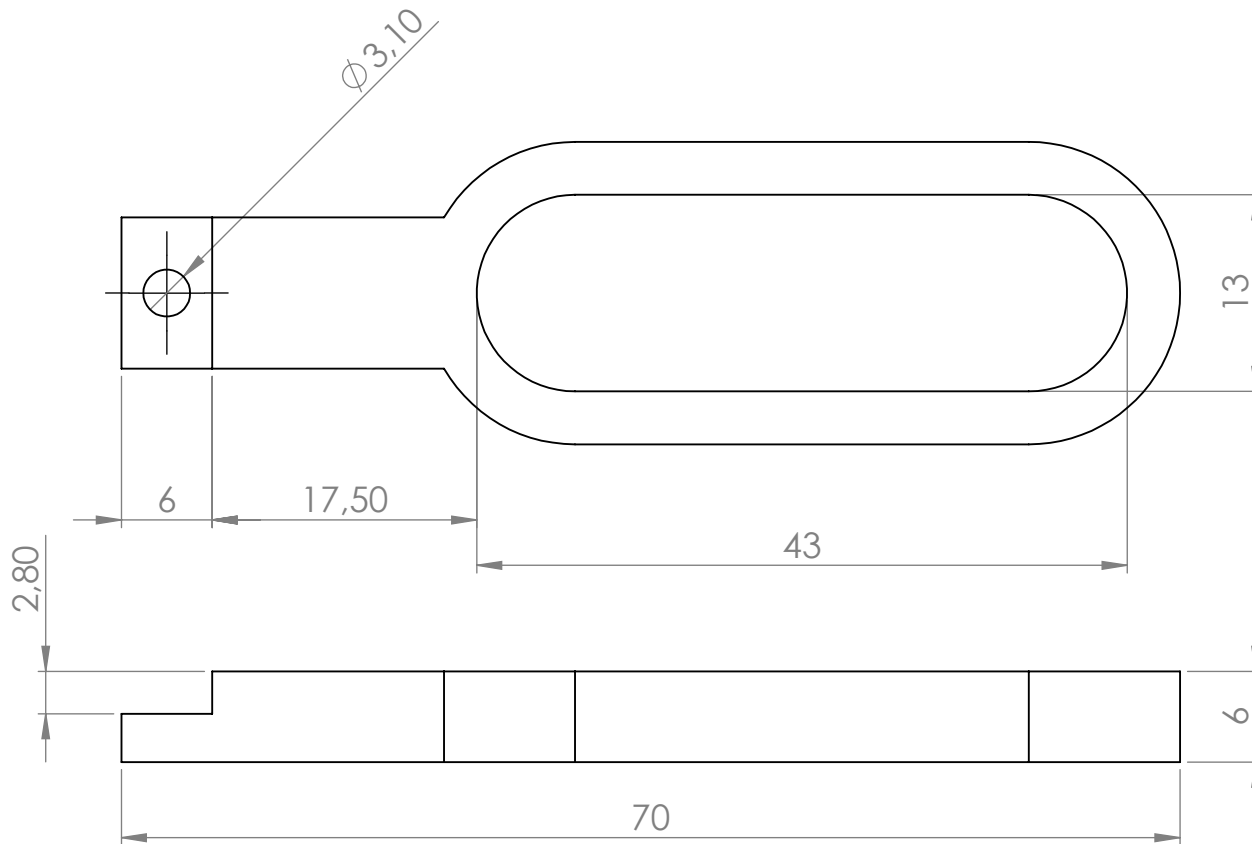
SOLIDWORKS Student Edition.
Solo para uso académico.




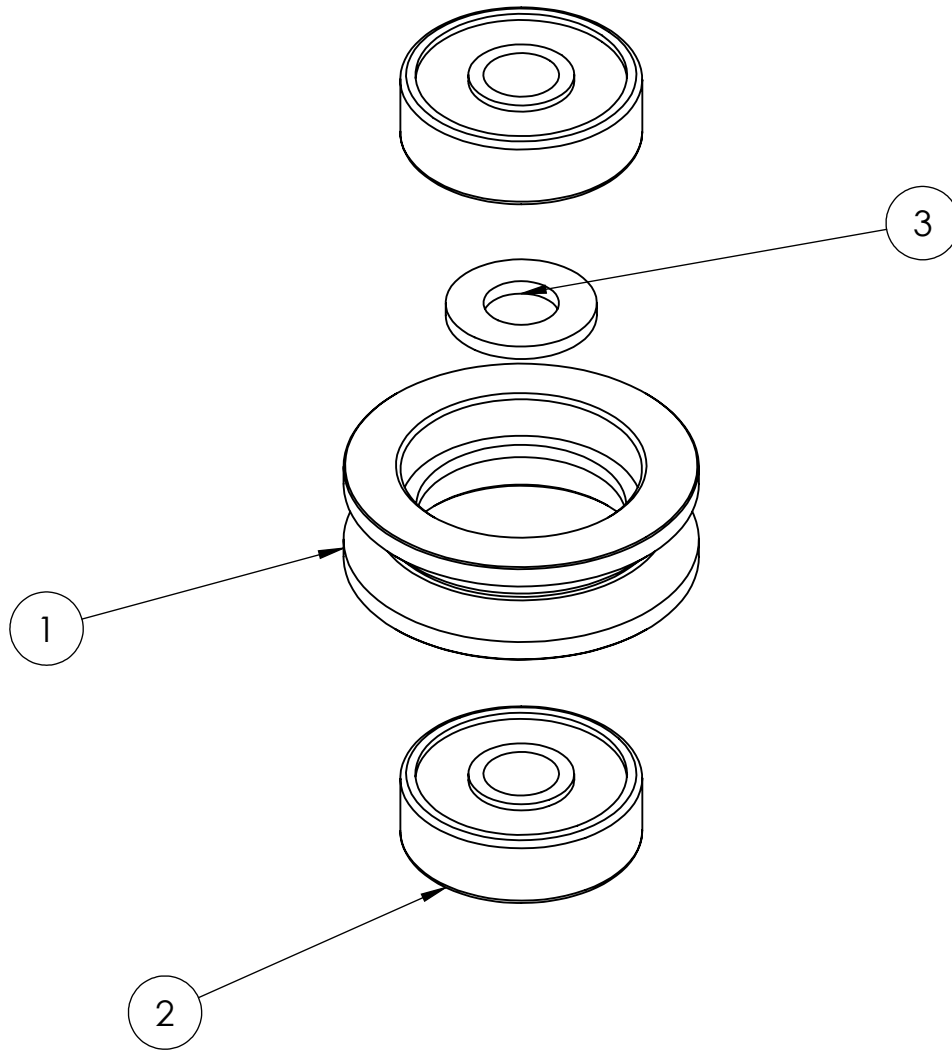
Project Energy MegaDrawz		Date 11/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G5.6
Description Bearing_connector_30mm_v2: Use PLA and a 3D printer. Dimensions are adjusted for RB_Delta 3D printer, for different tolerances A shall be adjusted for get a push fit with a bearing of 30mm of internal diameter.		Matt Venn Engineering 




Project Energy MegaDrawz		Date 11/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number G5.7
Description Base_v2: Machine a 6mm Delrin with a CNC milling machine. Top face can be chamfered. Use 1.5 mm tabs while machining.		Matt Venn Engineering 

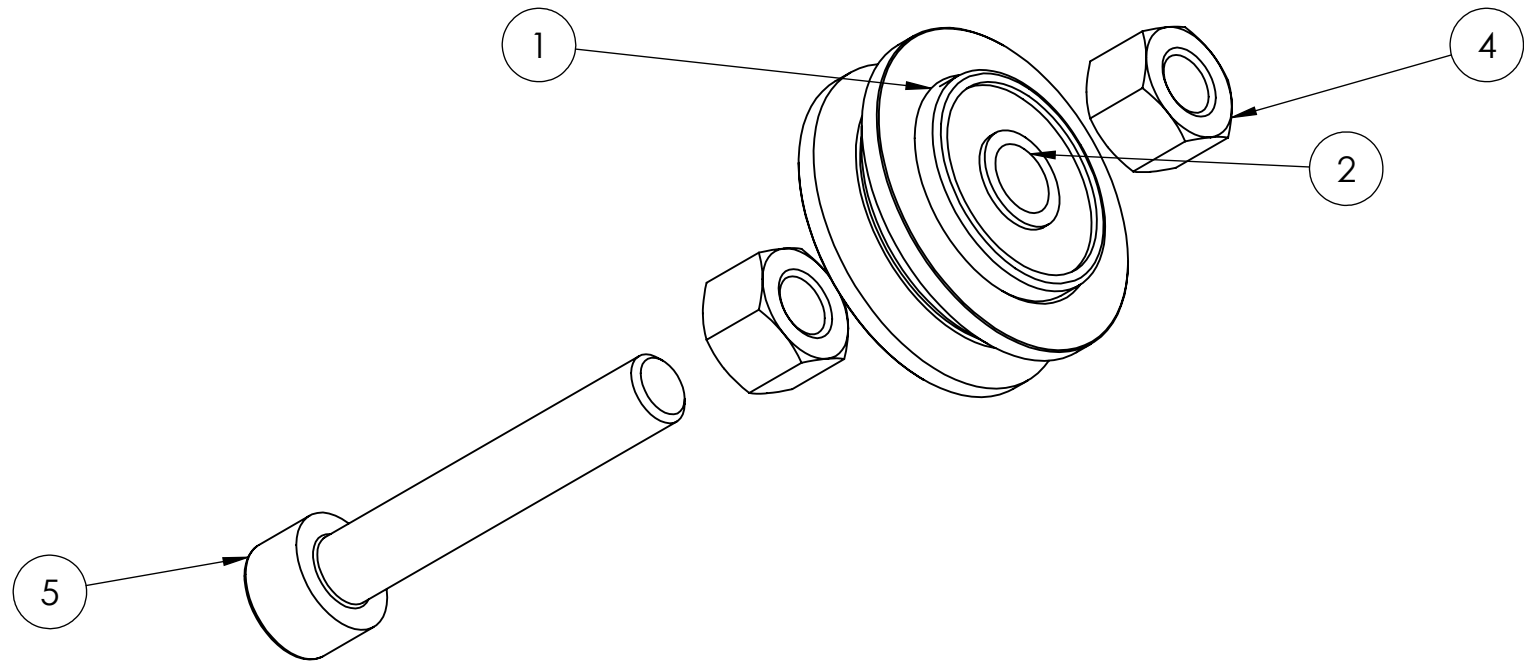


Project Energy MegaDrawz		Date 28/08/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G5.8
Description Guialeva_6mm: Machine a 6mm Delrin plate with a milling machine.		Matt Venn Engineering 




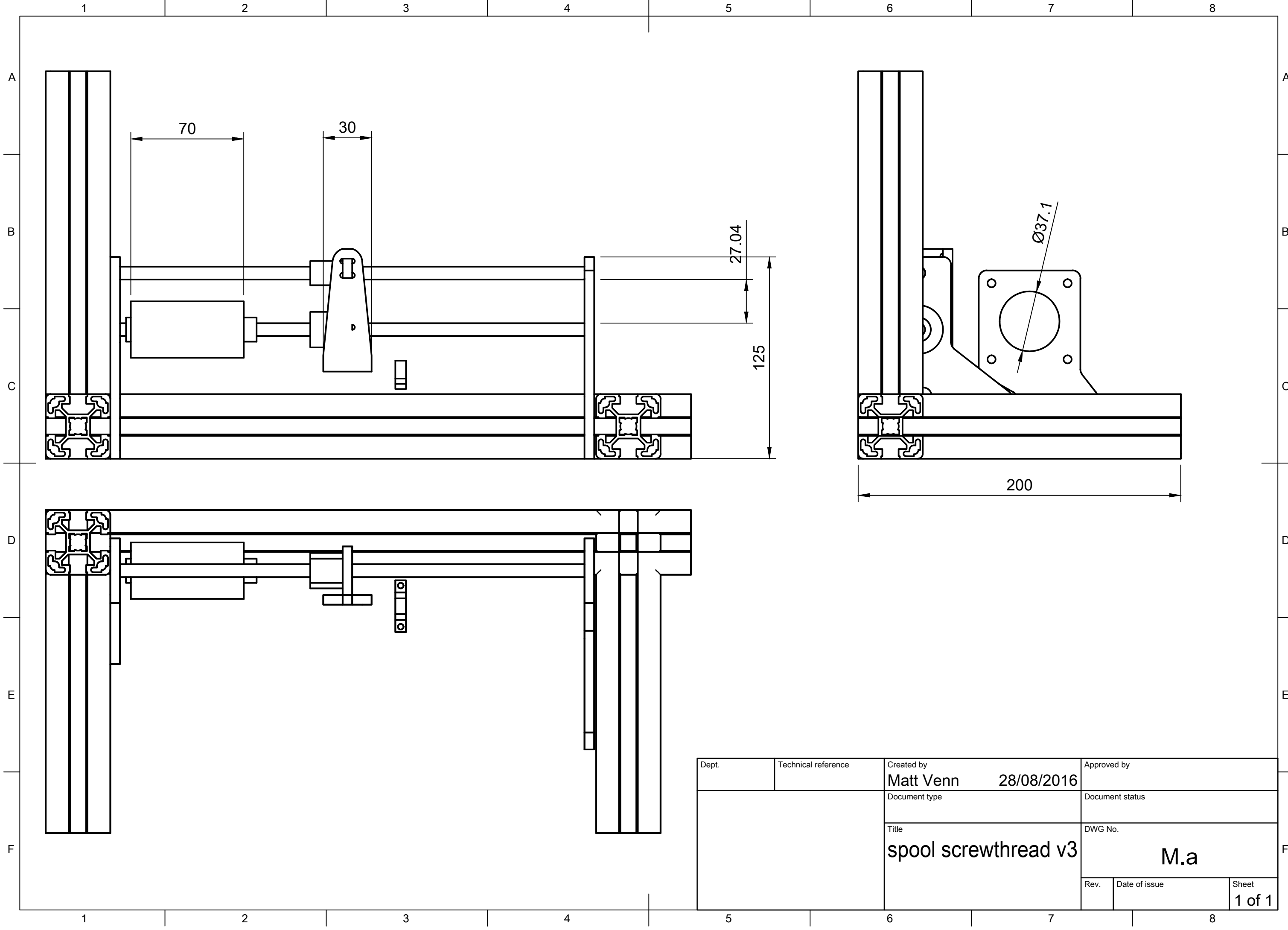
NUMBER	ID	QUANTITY
1	G5.sa	1
2	10mm bearing	2
3	socket head cap screw_iso	1

Project Energy MegaDrawz		Date 04/09/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G5.sa.1
Description V-Wheel_5x16x5: Verify that piece 3 is placed and lubricate before place the second bearing.		Matt Venn Engineering 

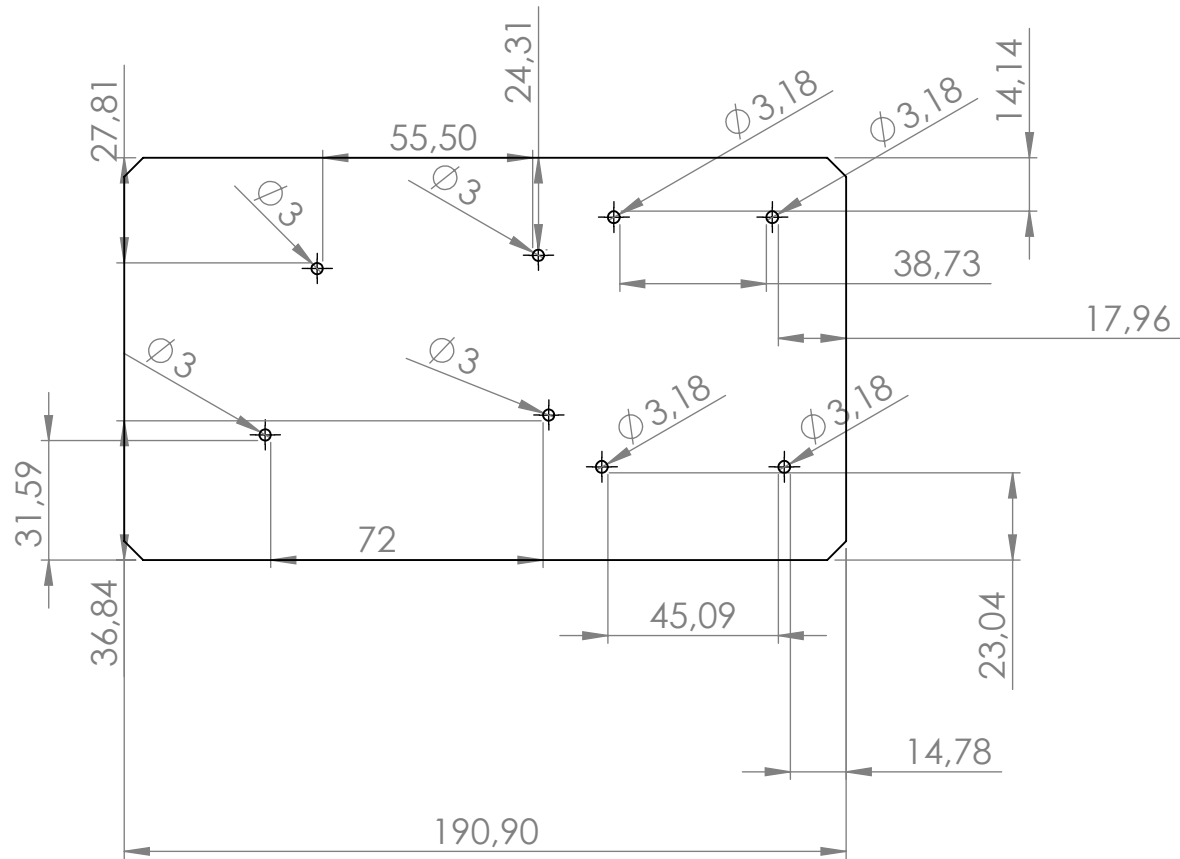



NUMBER	ID	QUANTITY
1	G5.sa	1
2	hex nut gradec_iso	2
3	socket head cap screw_iso	1
4	hex nut gradec_iso	2
5	socket head cap screw_iso	1

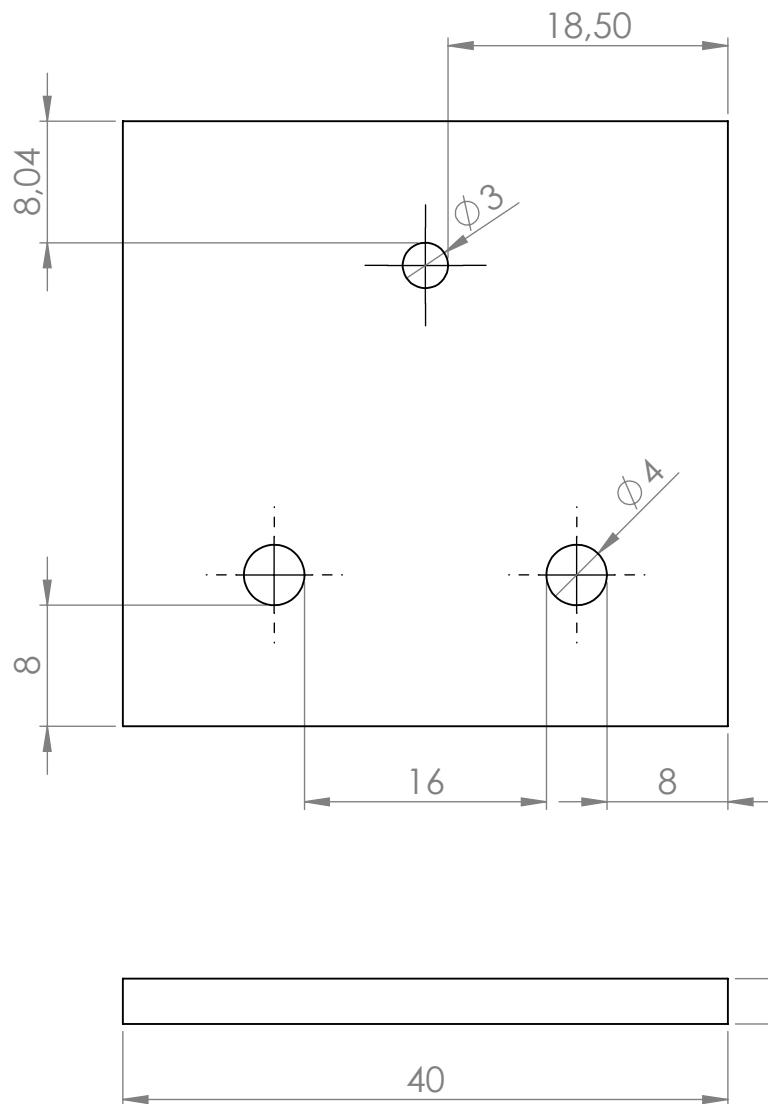
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number G5.sa.2
Description Se_rodamientos: Assembly at the same time that G5.a.		Matt Venn Engineering 




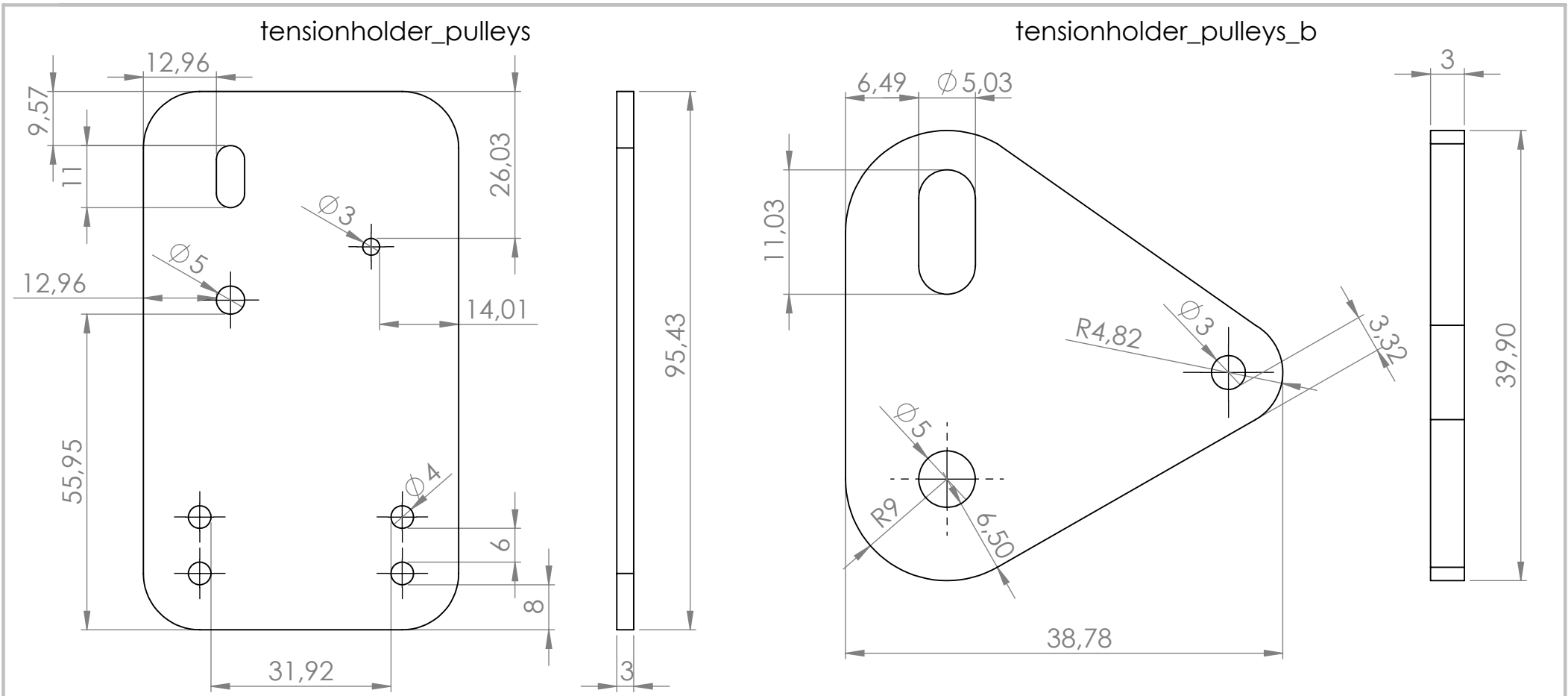
Dept.	Technical reference	Created by Matt Venn	28/08/2016	Approved by
		Document type		Document status
		Title spool screwthread v3		DWG No. M.a
		Rev.	Date of issue	Sheet 1 of 1




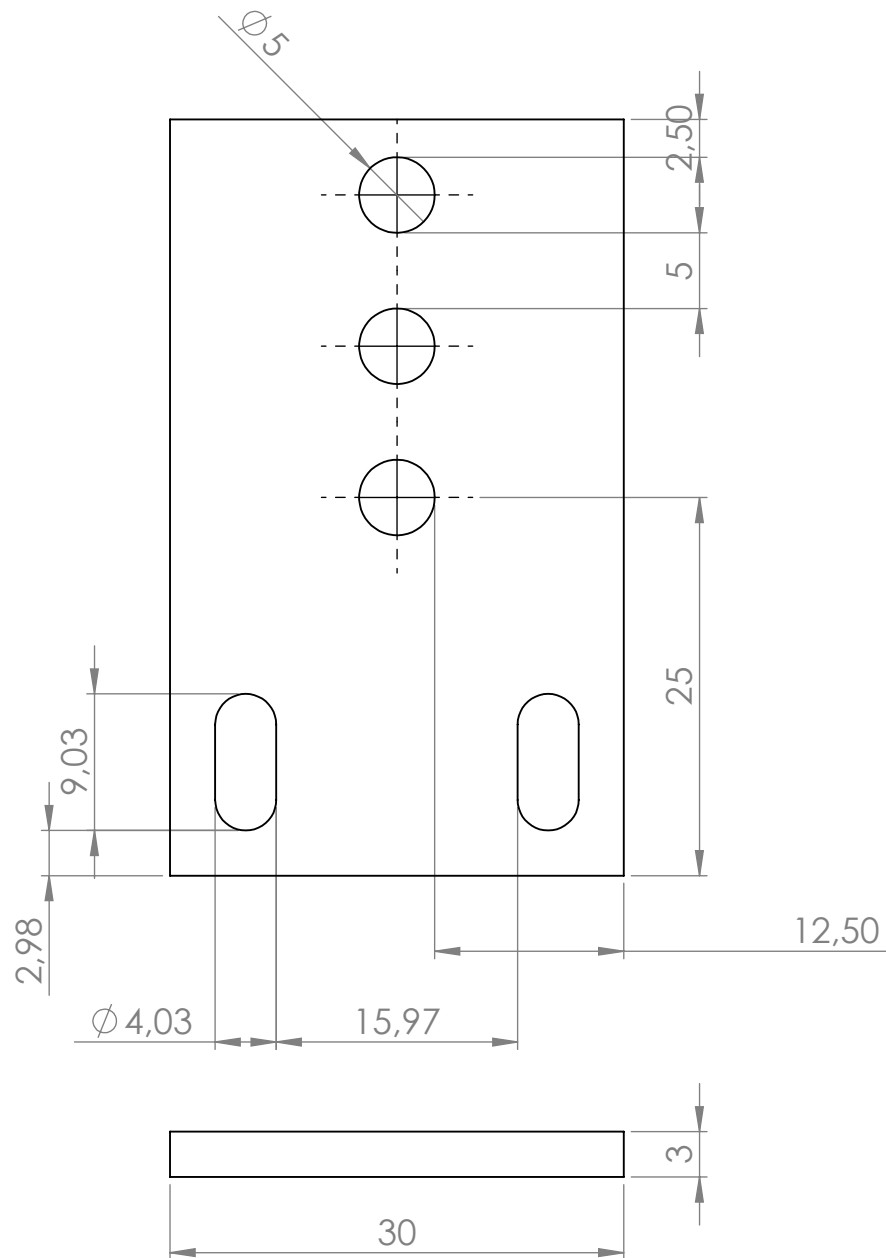
Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number O1.1
Description Base_v1: Laser with MDF from 3 to 10 mm. Verify the position of the holes. Screws shall be perpendicular to the surface, if screws are angled or have problems to fit mark the holes with laser cutter and machine with a tower driller.		Matt Venn Engineering 




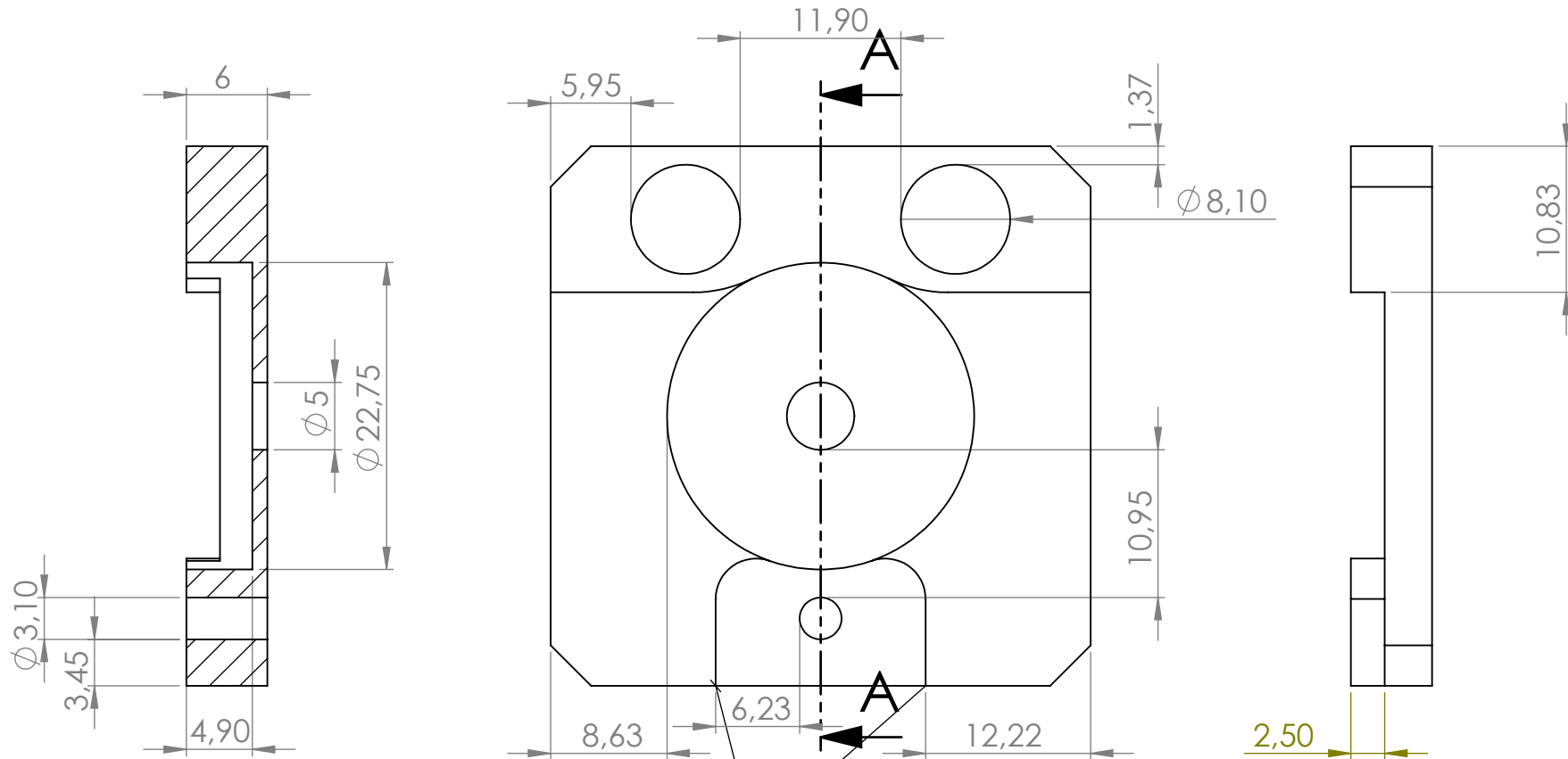
Project Energy MegaDrawz		Date 07/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number O2.1.1
Description Soportepinpoleas: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 



Project Energy MegaDrawz		Date 14/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number O2.1.2
Description Tensionholder_pulleys: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm. Keep parts together for assembly.		Matt Venn Engineering 




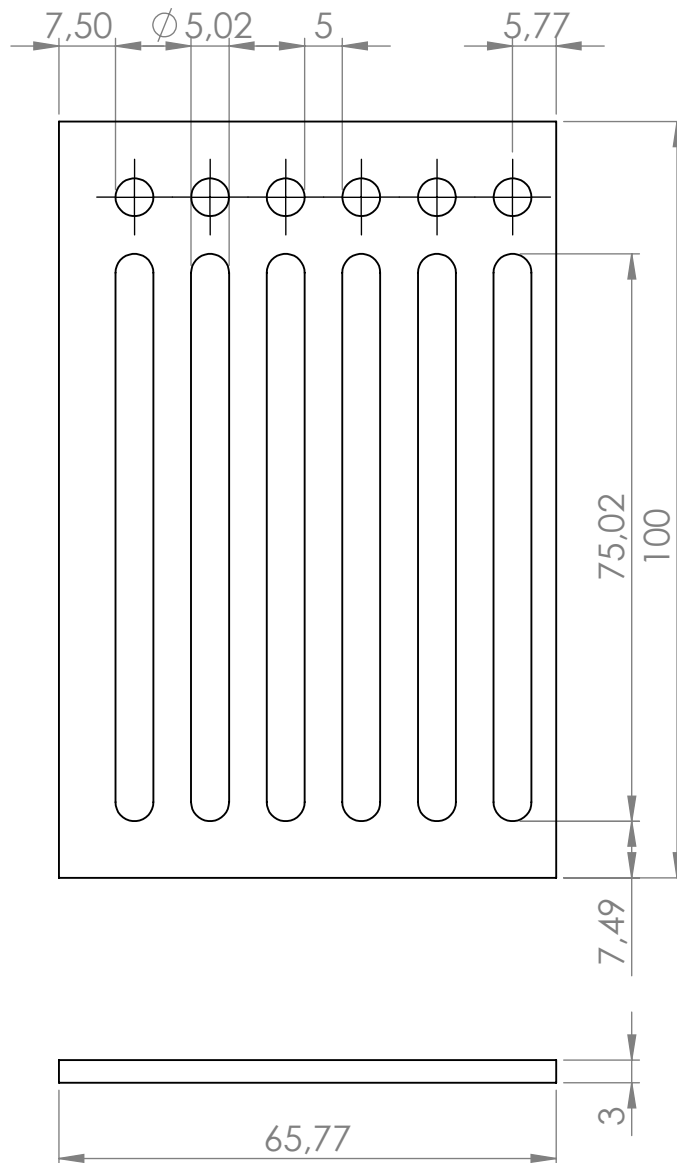
Project Energy MegaDrawz		Date 20/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number O2.1.3
Description Pullysupp_bigprot: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




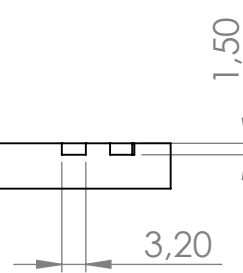
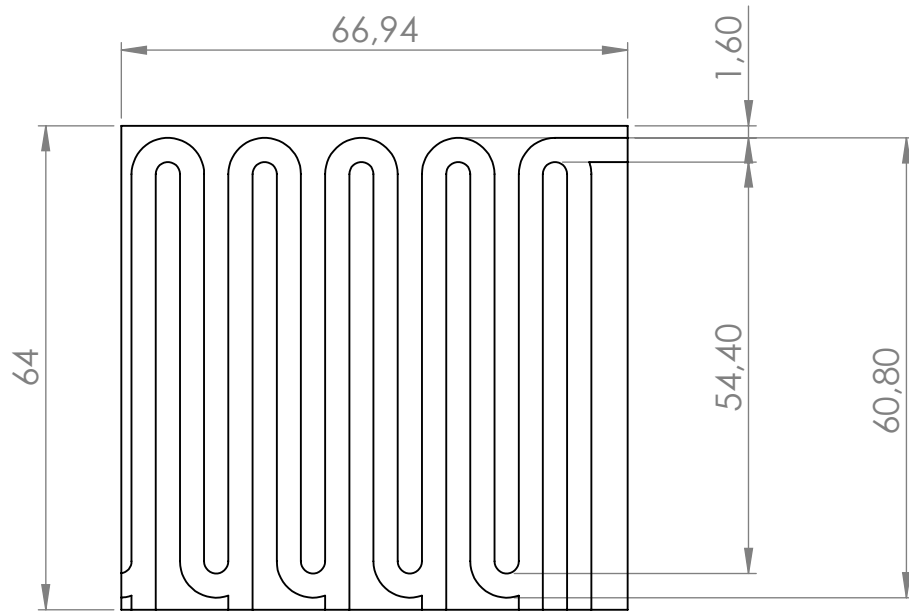
SECTION A-A


Round edges

Project Energy MegaDrawz		Date 11/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number O2.1.4
Description Puleymount: Machine a 6mm Delrin with a CNC milling machine. Round marked edges with sandpaper. Do not machine this piece again without round 5mm in the CAD model.		Matt Venn Engineering 

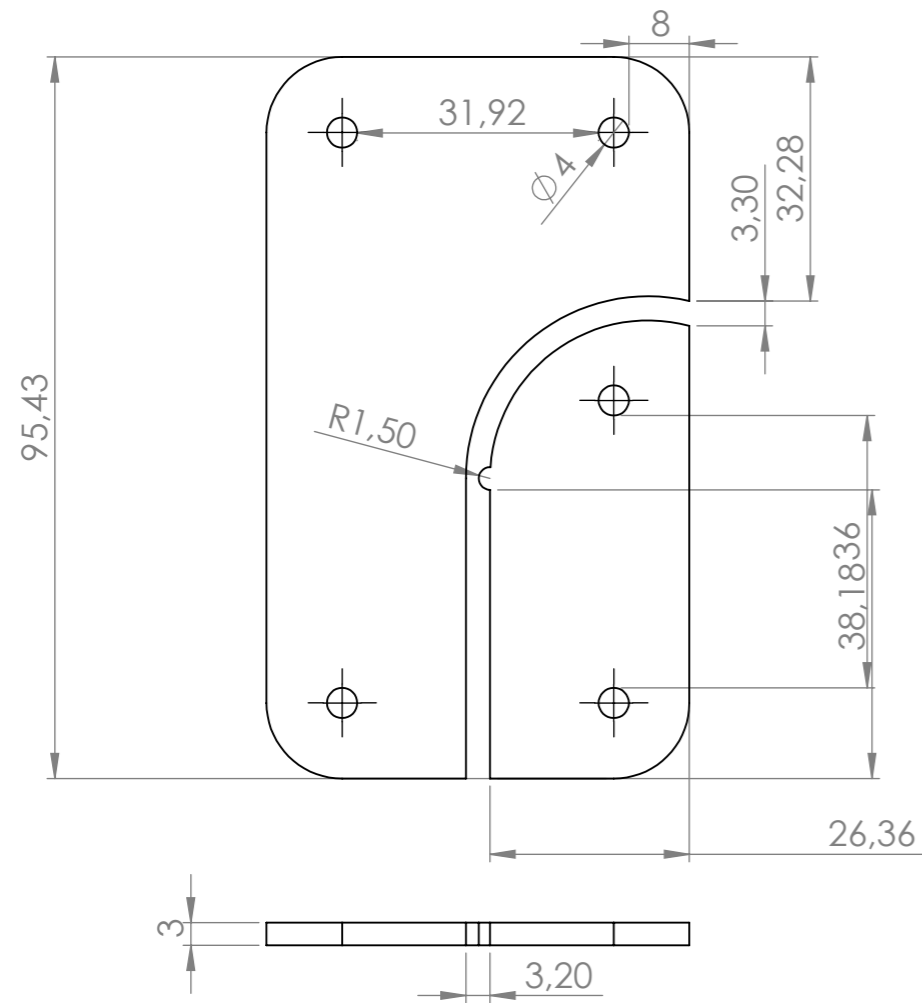


Project Energy MegaDrawz		Date 13/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number O2.2.1
Description Bancodepruebas_tensionholder: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

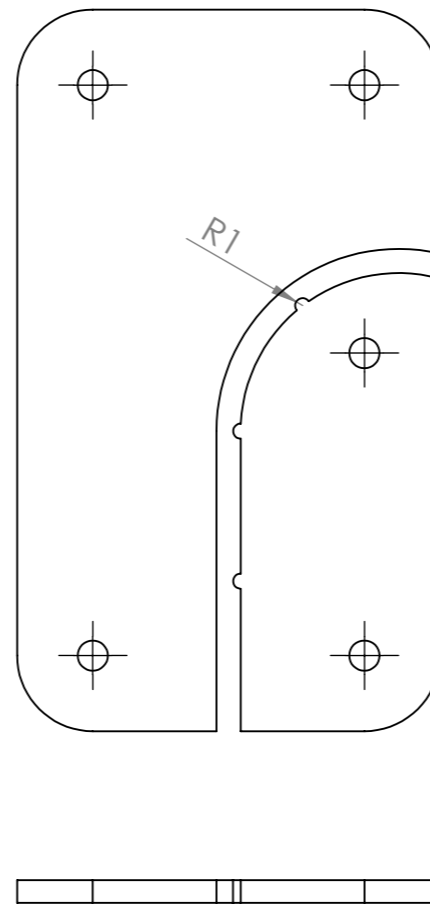


Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number O2.2.2
Description Tensionholder_cnc_3.2mm: Machine a 6mm Delrin plate with a CNC milling machine. Two of this pices are required for the test.		Matt Venn Engineering 

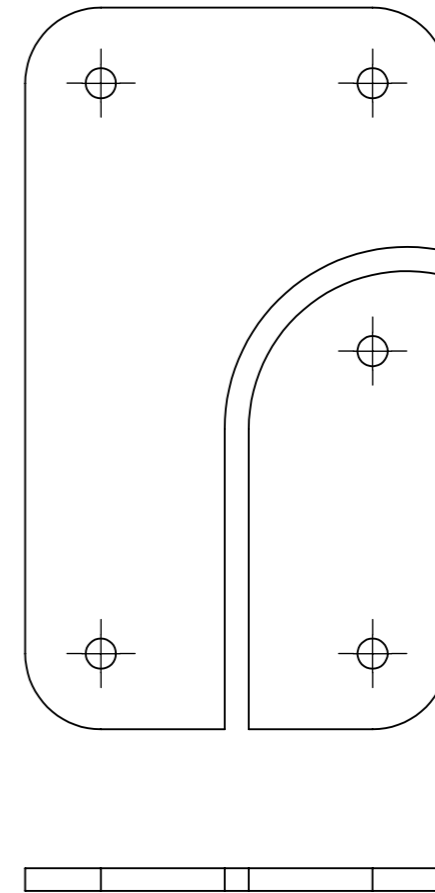
tensionholder_tube_b_v4.3_subassembly




tensionholder_tube_b_v4.2_subassembly

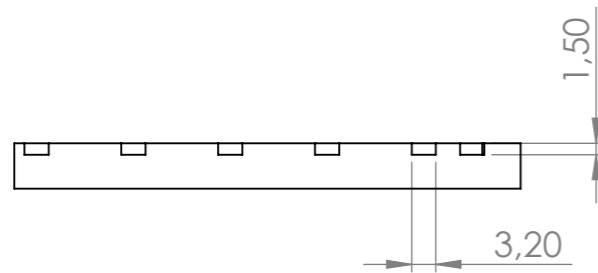
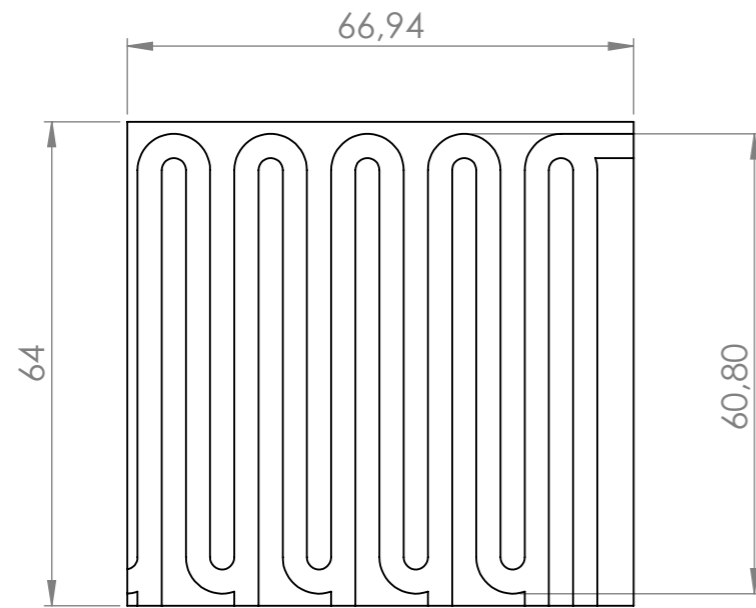


tensionholder_tube_b_v4

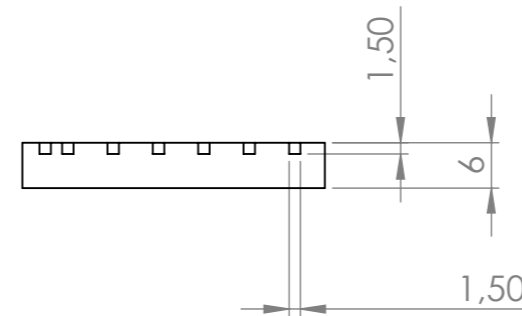
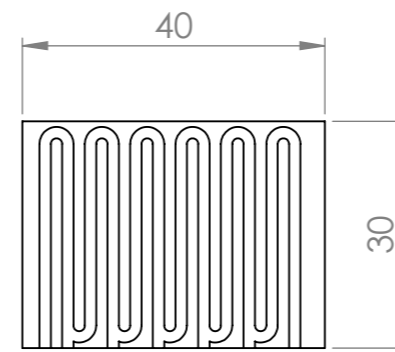



Project Energy MegaDrawz		Date 23/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number O2.2.2
Description Prototypes tensionholder: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 

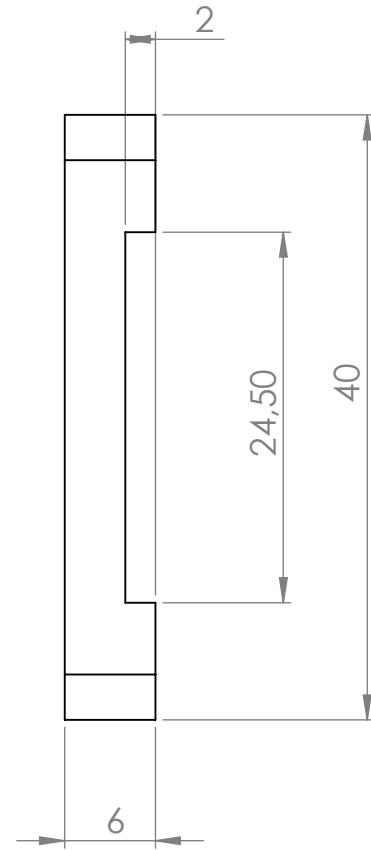
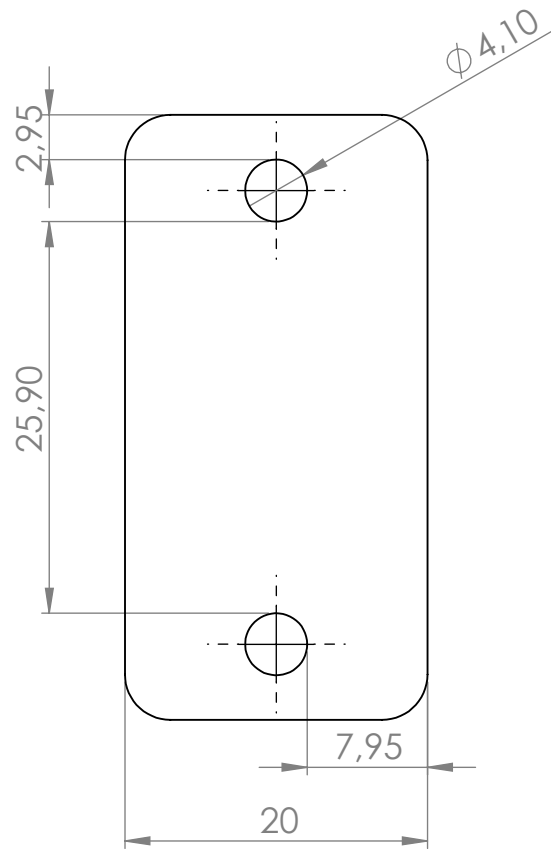
tensionholder_cnc_3.2mm




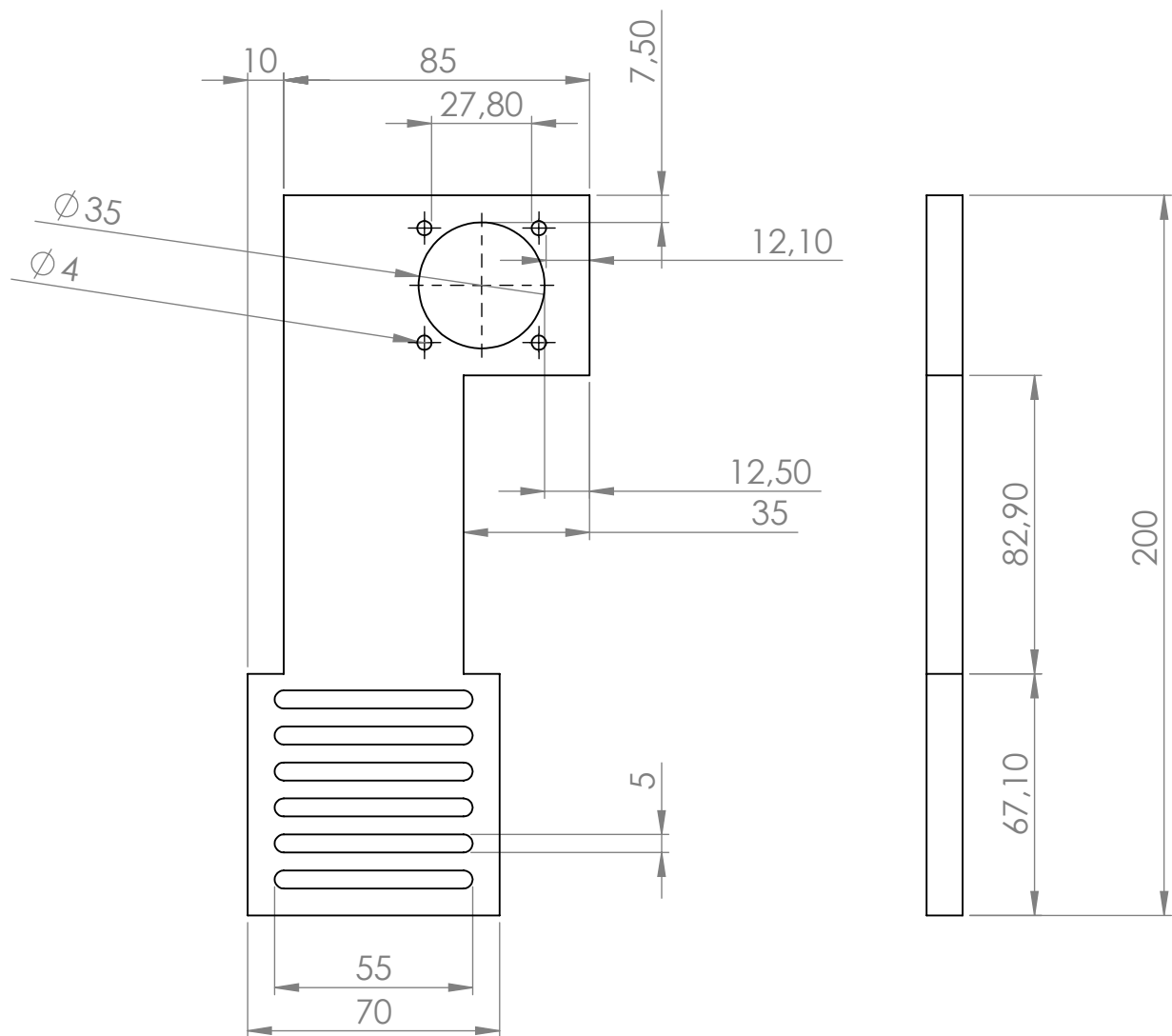
tensionholder_cnc_test1




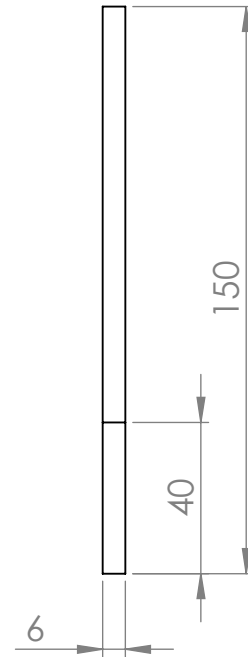
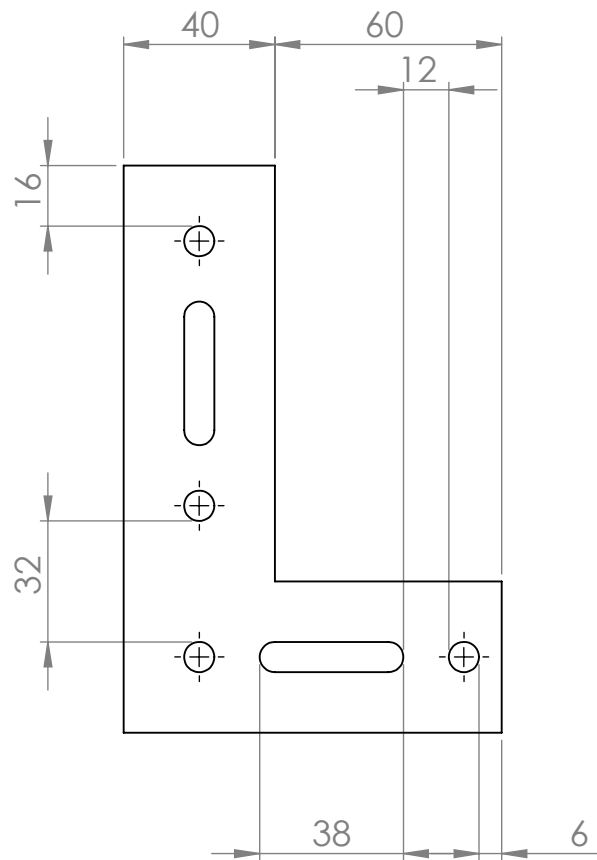
Project Energy MegaDrawz		Date 23/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number O2.2.3
Description Prototypes delrin tensionholder: Machine a 6mm Delrin plate with a CNC milling machine		Matt Venn Engineering 




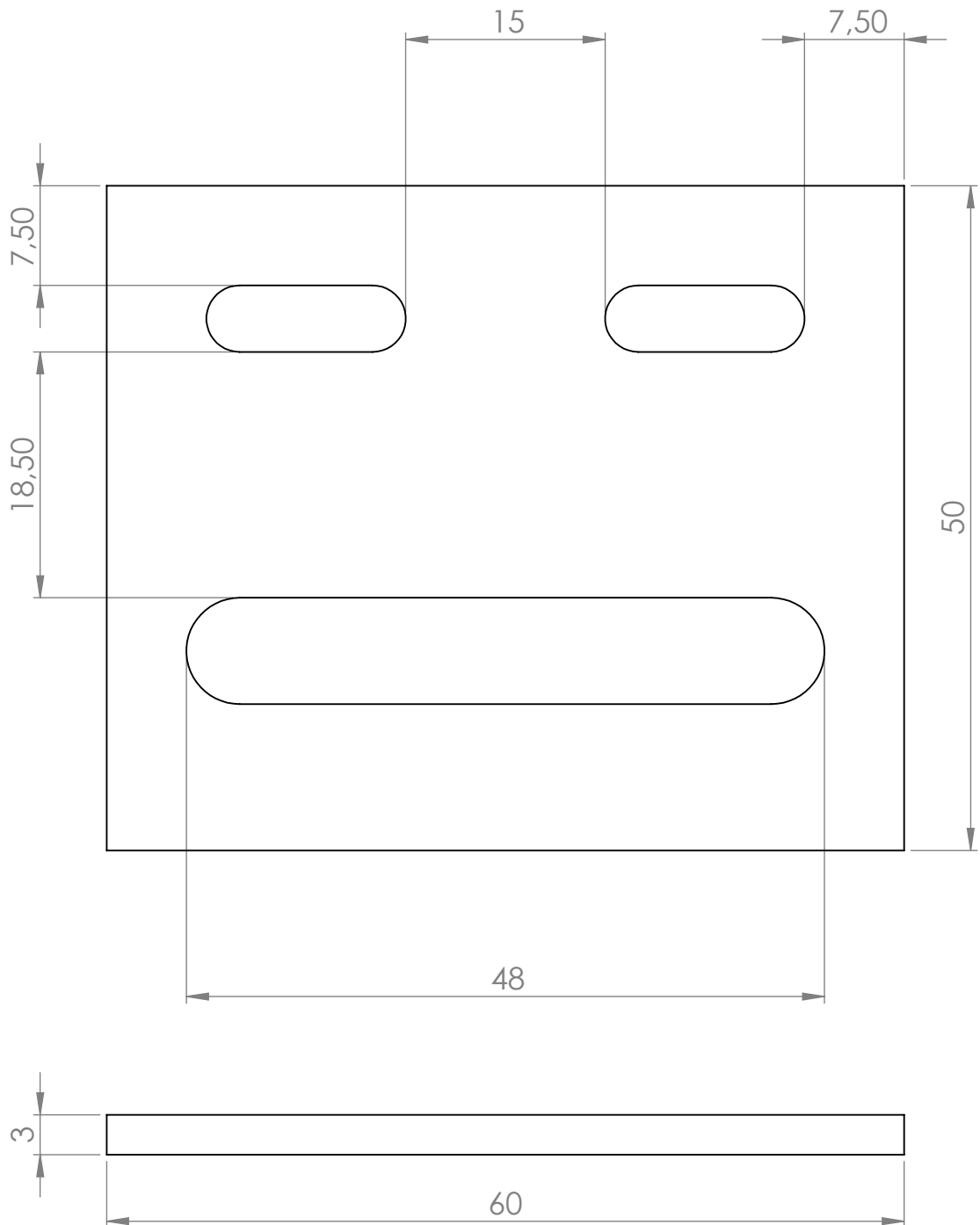
Project Energy MegaDrawz		Date 11/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number O3.1
Description Servo_mount: Machine a 6mm Delrin plate with a milling machine or use PLA and a 3D printer.		Matt Venn Engineering 




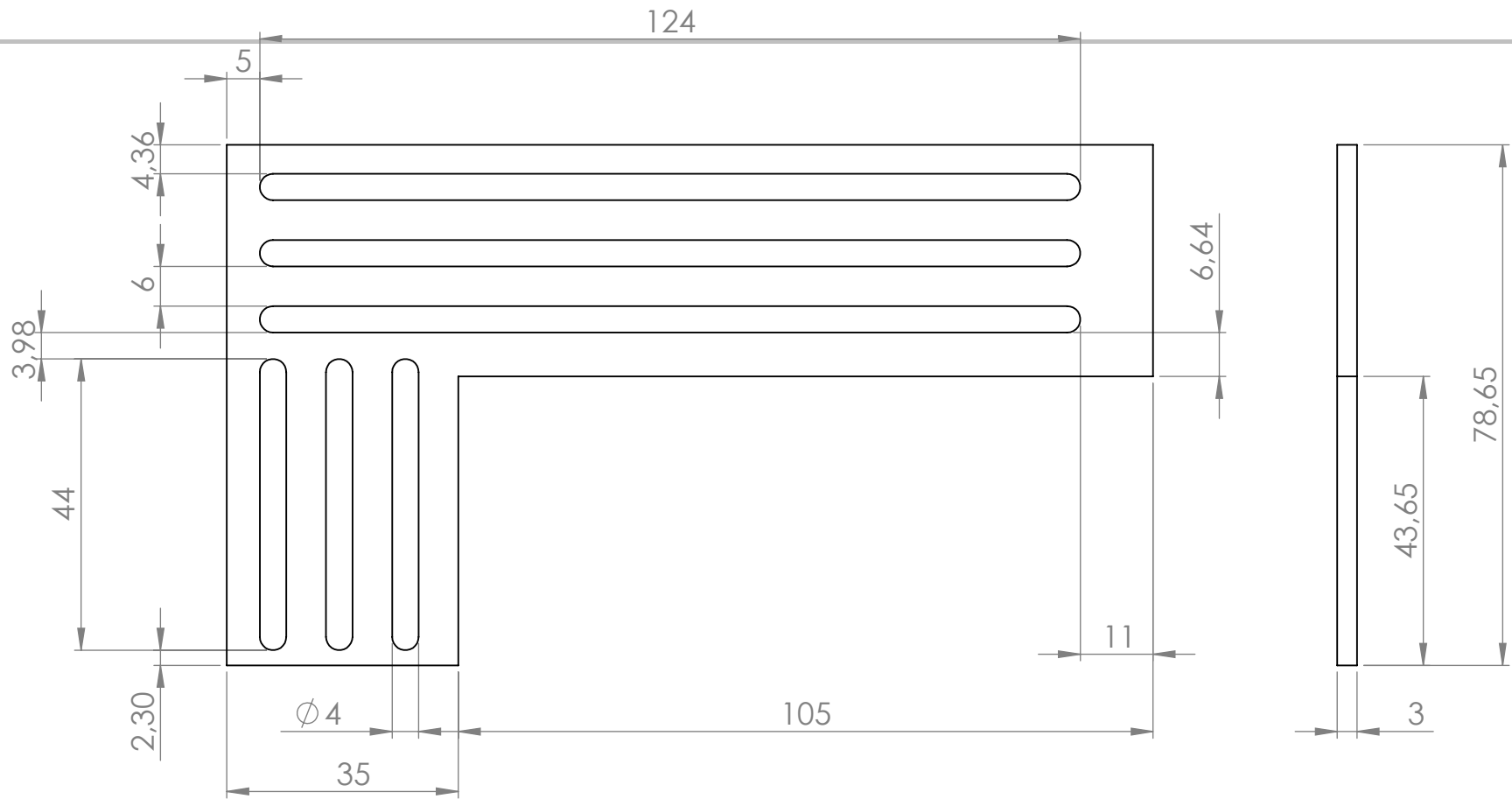
Project Energy MegaDrawz		Date 05/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number O3.2
Description fan_support: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




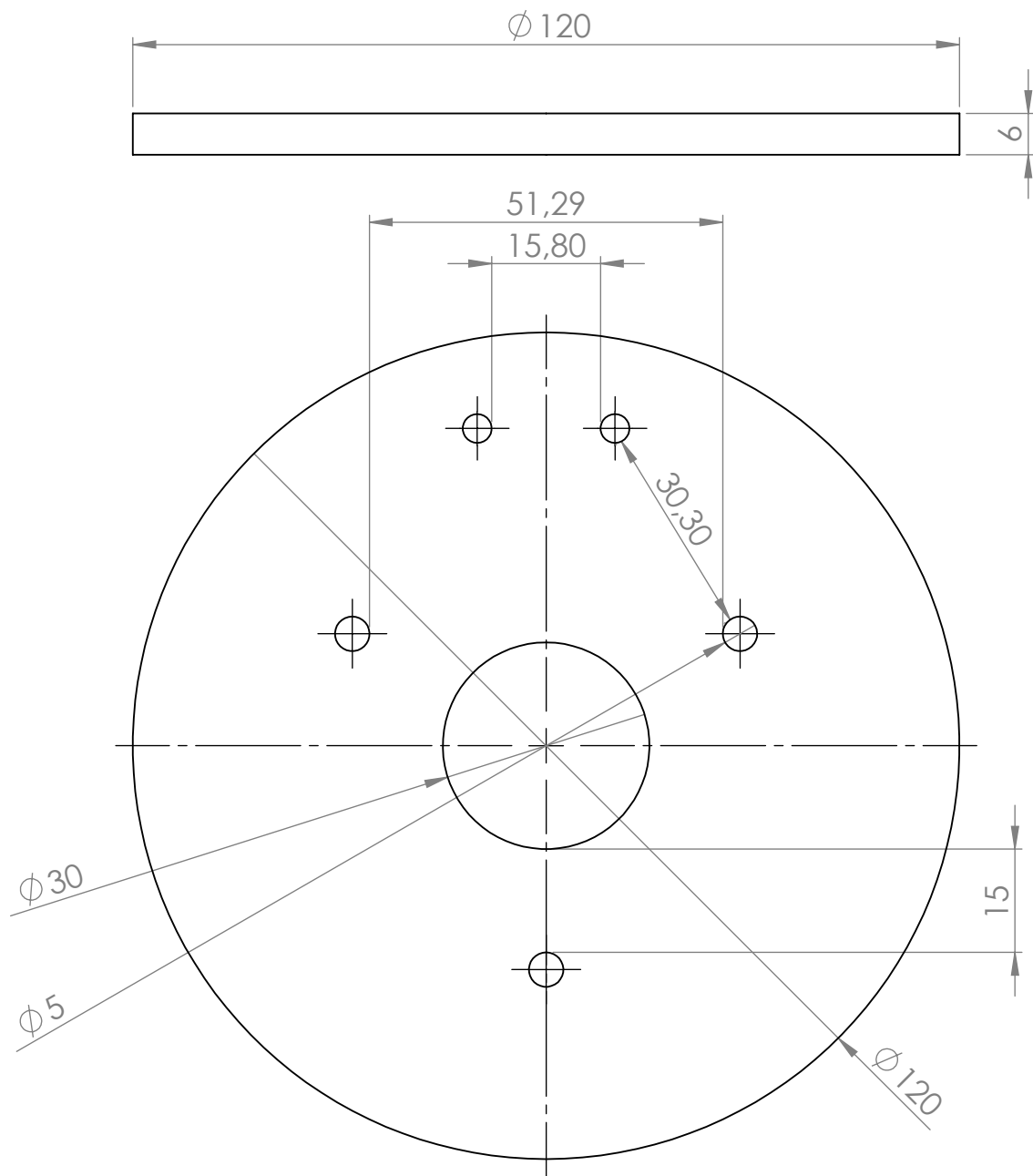
Project Energy MegaDrawz		Date 08/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number O3.3
Description 40x40_6mm_mdf_gusset: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




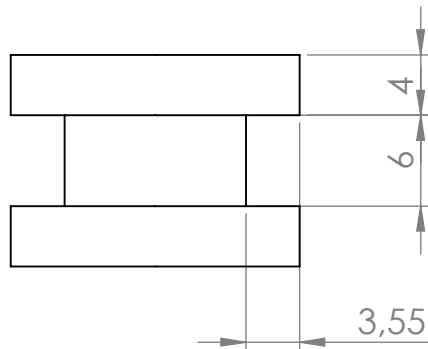
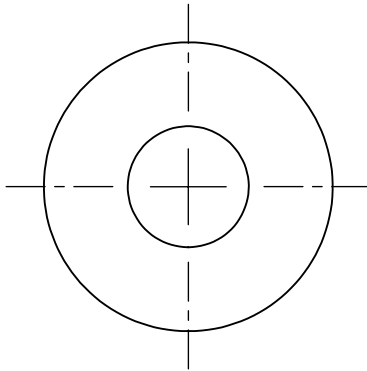
Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number O3.4
Description M5 to M8 mount: manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




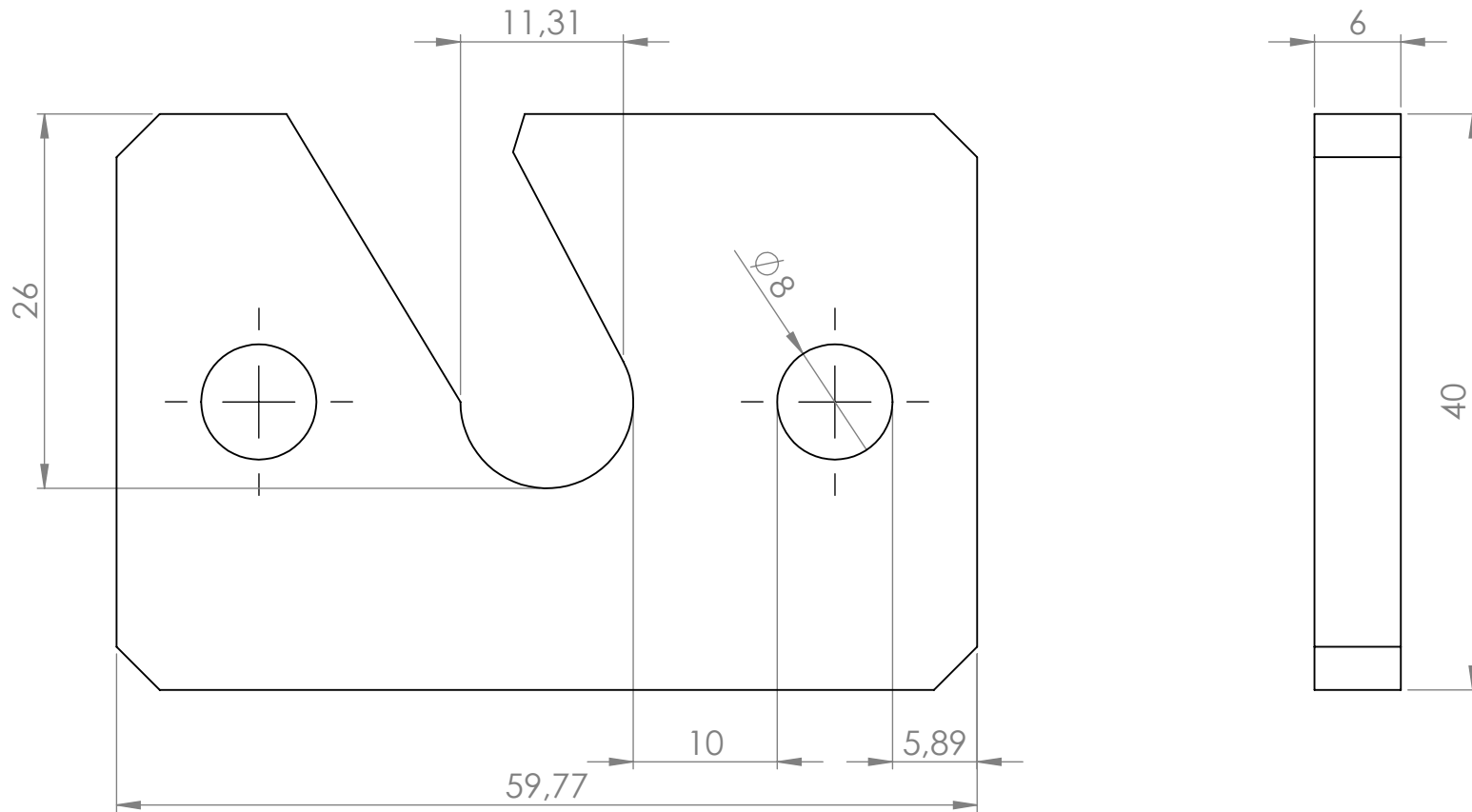
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number O3.5
Description Escuadras_m4: Manufacture with 3mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 




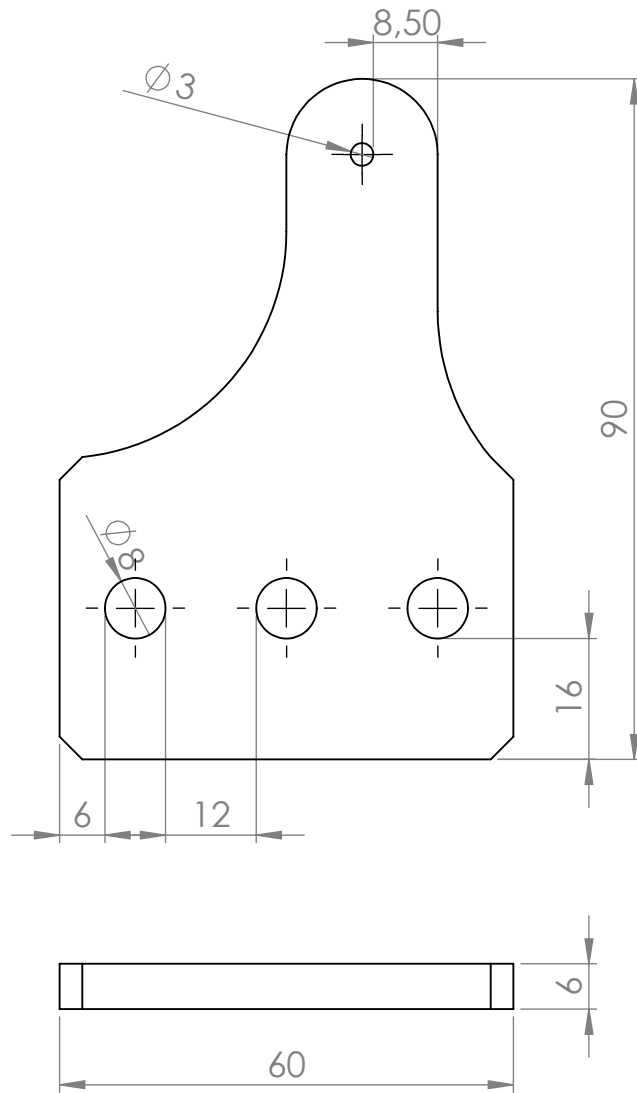
Project Energy MegaDrawz		Date 04/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number O4.1
Description Base_p4: Manufacture with 6mm MDF, use a laser cutter with tolerances lower than 0,1mm.		Matt Venn Engineering 



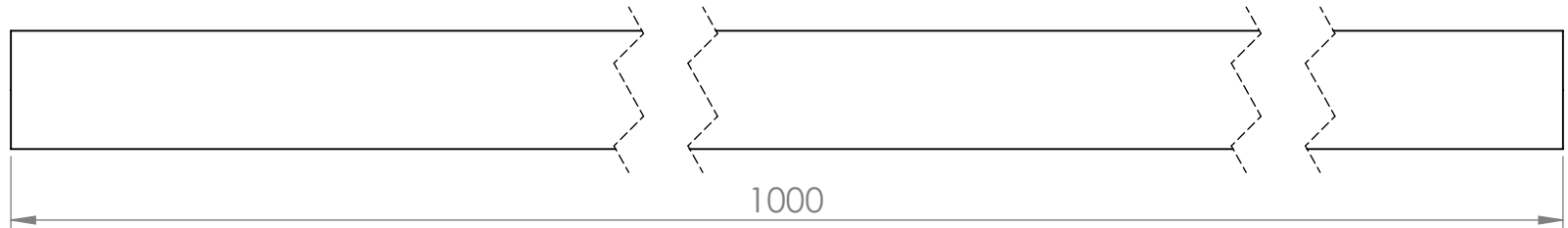
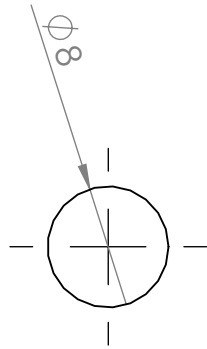
Project Energy MegaDrawz		Date 28/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number P1
Description h_beam_holder: Machine from an aluminum rod with lathe.		Matt Venn Engineering 



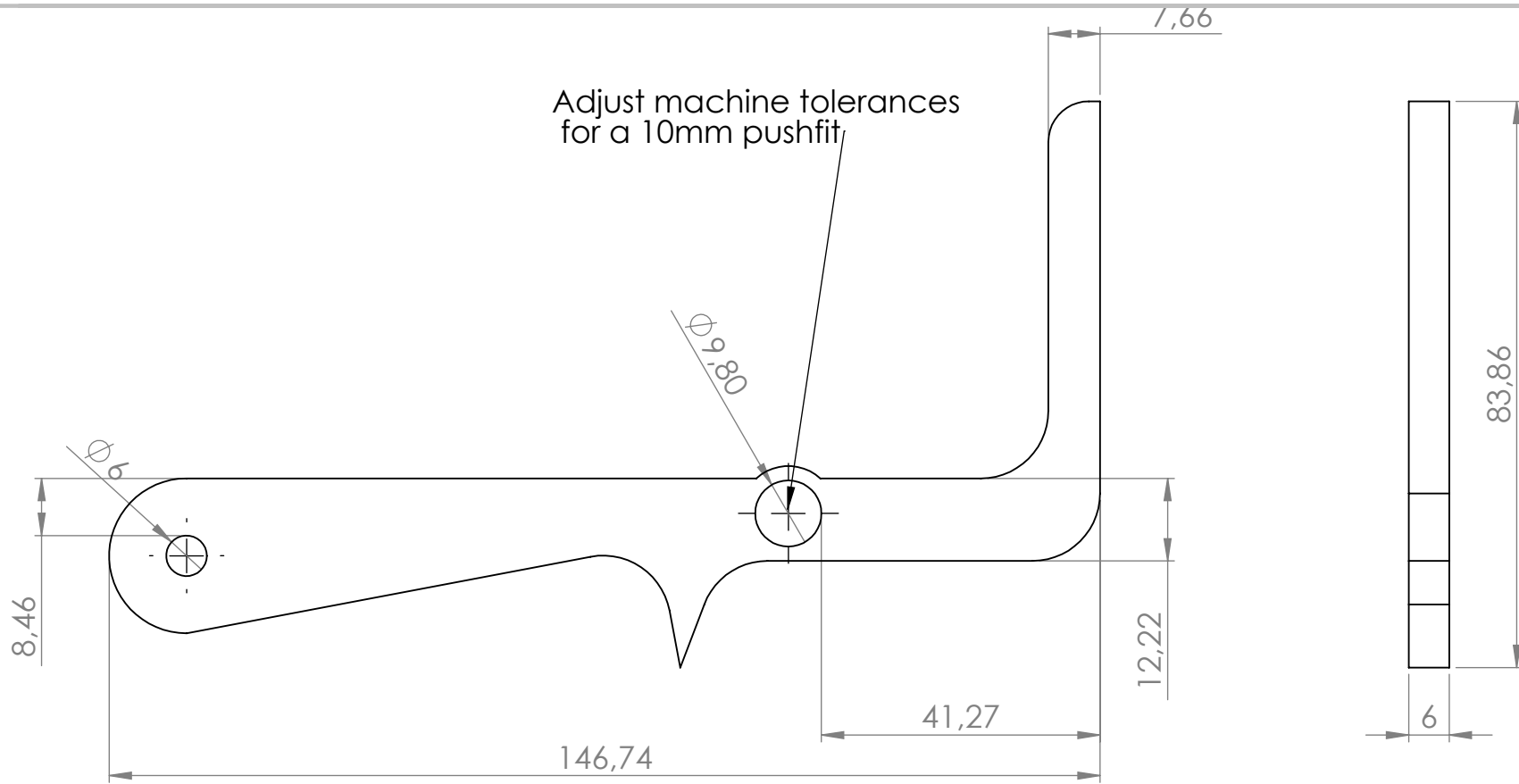
Project Energy MegaDrawz		Date 11/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number P2
Description Lateral_v3_izq: Machine a 6mm Delrin plate with a milling machine.		Matt Venn Engineering 




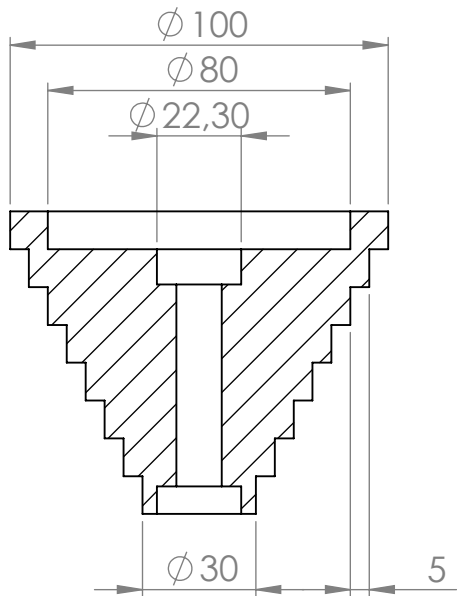
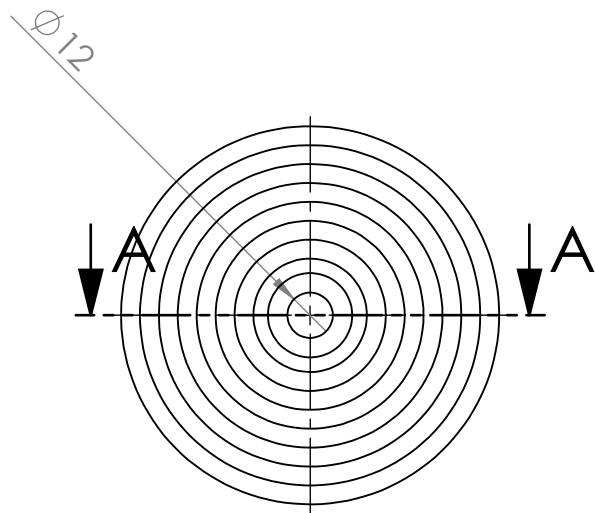
Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number P3
Description Lateral_v4: Machine a 6mm Delrin plate with a milling machine.		Matt Venn Engineering 



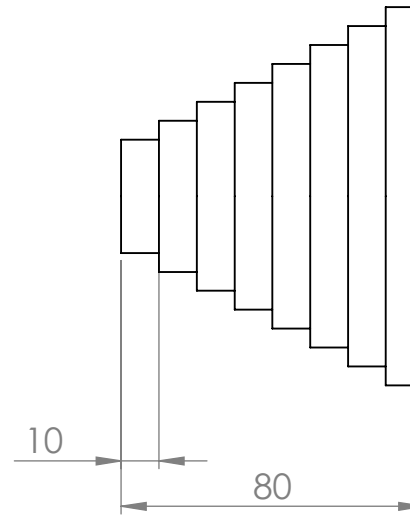
Project Energy MegaDrawz		Date 27/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 2:1
Approval Matthew Venn	Signature	Plan number P4
Description m8_varilla_1000mm: File edges.		Matt Venn Engineering 




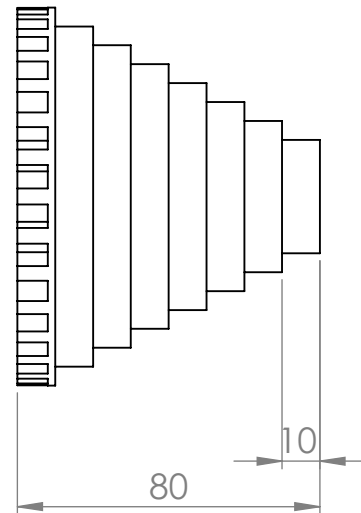
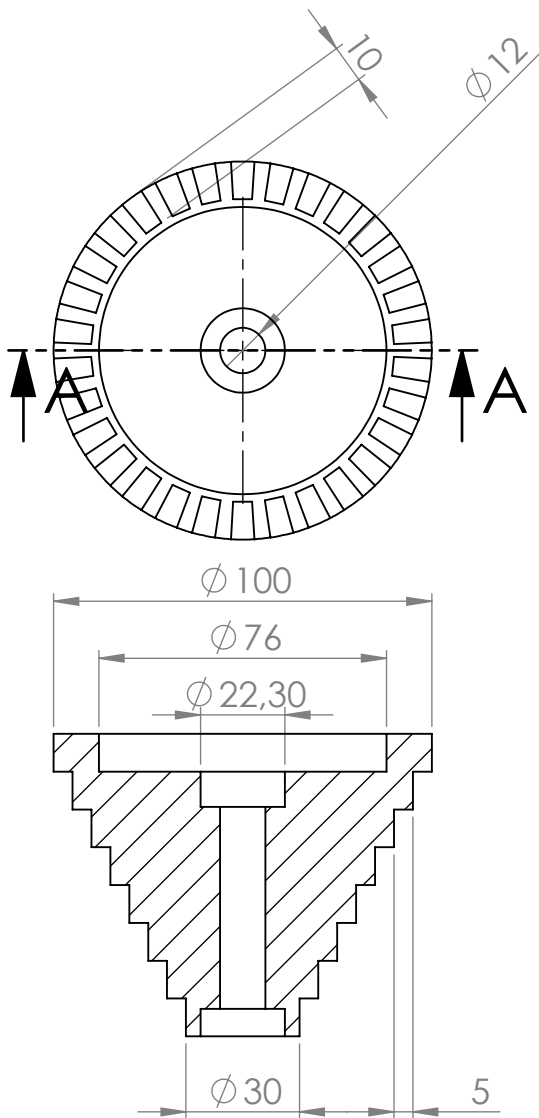
Project Energy MegaDrawz		Date 28/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:1
Approval Matthew Venn	Signature	Plan number P5
Description Palanca_freno: Laser cut from 3mm MDF or Use an aluminium plate of 3mm, face the top surface and machine with a CNC drilling machine. For aluminium piece chamfers can be added to the external edge, file the edges and use 1mm tabs.		Matt Venn Engineering 



SECTION A-A



Project Energy MegaDrawz		Date 14/07/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number P6
Description Soporte_liso_v2: Manufacture with PLA with a 3D printer. Supports are required. Verify that all the support material has been removed.		Matt Venn Engineering 



SECTION A-A

Project Energy MegaDrawz		Date 28/06/2016
Author Guillem Cabo Pitarch	Signature	Scale 1:2
Approval Matthew Venn	Signature	Plan number P7

Description
Soporte_ranuras: Manufacture with PLA with a 3D printer. Supports are required. Verify that all the support material of the slots has been removed.

Matt Venn
 Engineering 