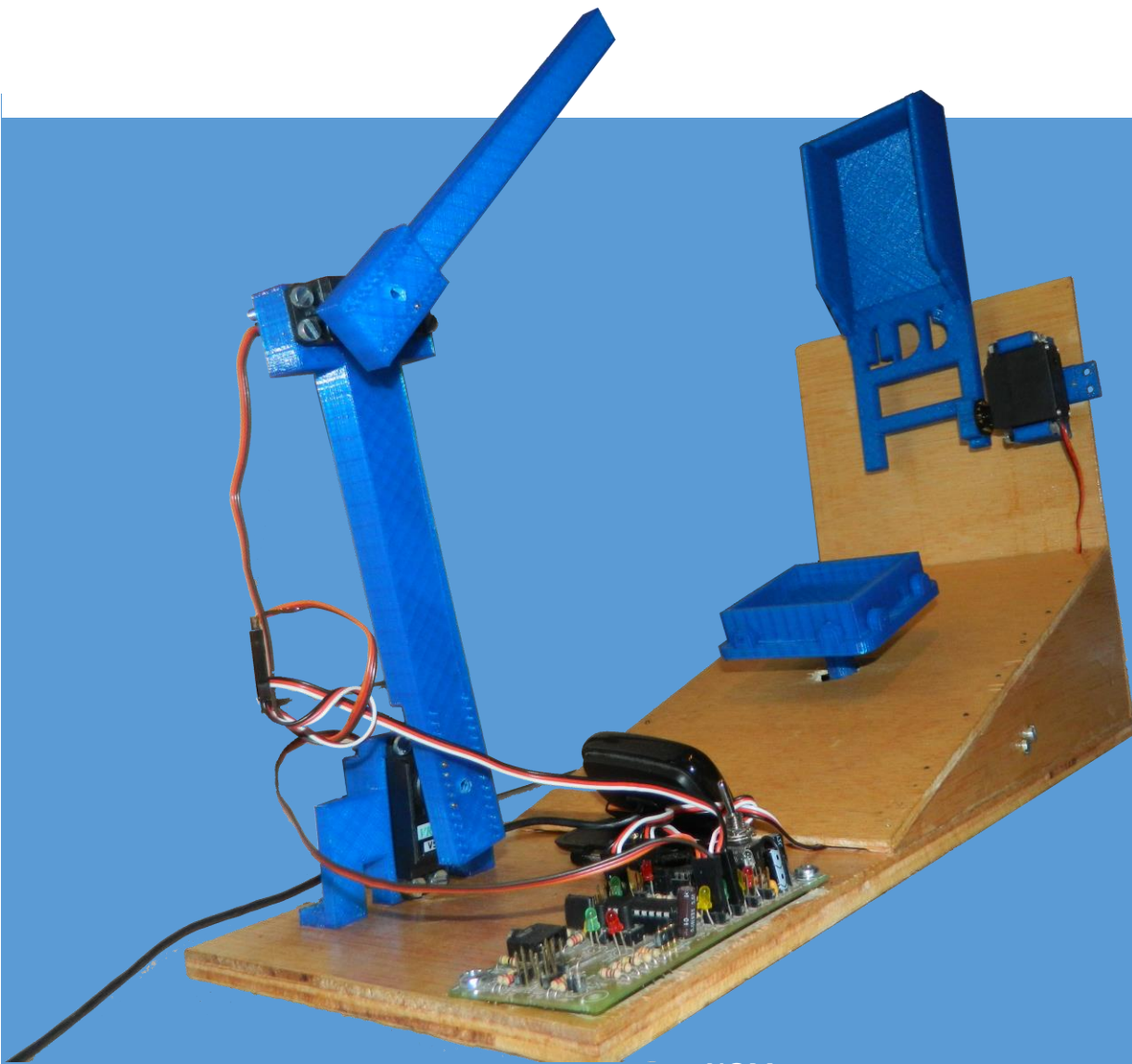


RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK



NOM:

MODALITAT: 2n de Batxillerat Tecnològic

TUTOR:

CURS: 2014 - 2015

“Abans de llegir el treball de recerca és oportú visitar següent enllaç (https://www.youtube.com/watch?v=9ZeD7_q8Y3U), on trobareu un vídeo de la demostració del funcionament del robot.”

ÍNDEX

1. INTRODUCCIÓ	6
1.1. Objectius	7
1.2. Metodologia	7
2. EL CUB DE RUBIK	8
2.1. Què és?	8
2.2. Història	8
2.3. Curiositats sobre el <i>Cub de Rubik</i>	9
2.4. Altres tipus de cubs	9
2.5. Mètodes de resolució	10
2.6. Combinacions	10
2.6.1. Número Màgic: 20	11
2.7. Notació	11
3. PLACA IMAGINA	12
3.1. Característiques	12
4. DISSENY EN 3D – GOOGLE SKETCHUP	13
4.1. Com funciona?	14
5. CREACIÓ D'UNA APLICACIÓ PER A ANDROID – APP INVENTOR	15
6. CONSTRUCCIÓ DE ROBOT CAPAÇ DE RESOLDRE CUB DE RUBIK	16
7. BLOC 0: CONEIXEMENT I FAMILIARITZACIÓ DEL CUB DE RUBIK – CUBE EXPLORER	17
7.1. Càmera web	18
8. BLOC 1:CONSTRUCCIÓ DE LA BASE DEL ROBOT	19
8.1. Construcció de les primeres peces	20
8.2. Construcció de l'estructura	20
8.3. Aparició d'alternatives durant el procés	21
9. BLOC 2: DISSENY, IMPRESSIÓ I MUNTATGE DE LES PECES EN 3D	22

9.1.	Impressora 3D – REPRAP BCN+	22
9.1.1.	Com funciona l'extrusió de material?	23
9.1.2.	Esquema de la impressora <i>REPRAP BCN+</i>	24
9.1.3.	Materials d'impressió – Filaments	25
9.2.	Disseny de les peces	26
9.3.	Impressió de les peces amb impressora 3D	26
9.3.1.	Exportar dissenys a .STL - SketchUp STL	27
9.3.2.	Reparar possibles errors d'impressió – NetFabb .	27
9.3.3.	Creació del GCODE	28
9.3.4.	Impressió del GCODE	29
9.4.	Impressió de les peces pel robot	29
9.5.	Muntatge del robot	30
10.	BLOC 3: ELECTRÒNICA I PROGRAMACIÓ – PLACA	
	PICAXE IMAGINA	33
10.1.	Electrònica	33
10.1.1.	<i>Bluetooth</i>	34
10.1.2.	Servomotors	34
10.1.3.	Font d'alimentació	36
10.2.	Programació de la placa	36
10.2.1.	Picaxe Editor 6	36
10.2.2.	Programar un robot capaç de moure i resoldre un <i>Cub de Rubik</i>	37
11.	BLOC 4: CREACIÓ D'UNA APLICACIÓ PER A ANDROID –	
	APP INVENTOR	39
11.1.	Disseny de l'aplicació	39
11.1.1.	<i>User Interface</i>	40
11.1.2.	<i>Layout</i>	40
11.1.3.	<i>Connectivity</i>	40
11.1.4.	Propietats i recursos usats	41
11.2.	Programació de l'aplicació – Blocks Editor	41
12.	PROBLEMES SORGITS DURANT EL PROCÉS	43

13. ÚLTIMS DETALLS	48
14. POSSIBLES MILLORES DEL TREBALL	49
15. CONCLUSIÓ I OPINIÓ PERSONAL	49
16. AGRAÏMENTS	50
17. BIBLIOGRAFIA	50
18. ANNEX	50

1. INTRODUCCIÓ

Ja fa bastants anys que vaig tenir el primer contacte amb el cub de Rubik i mai no m'ha deixat indiferent, ni em deixa de sorprendre avui en dia.

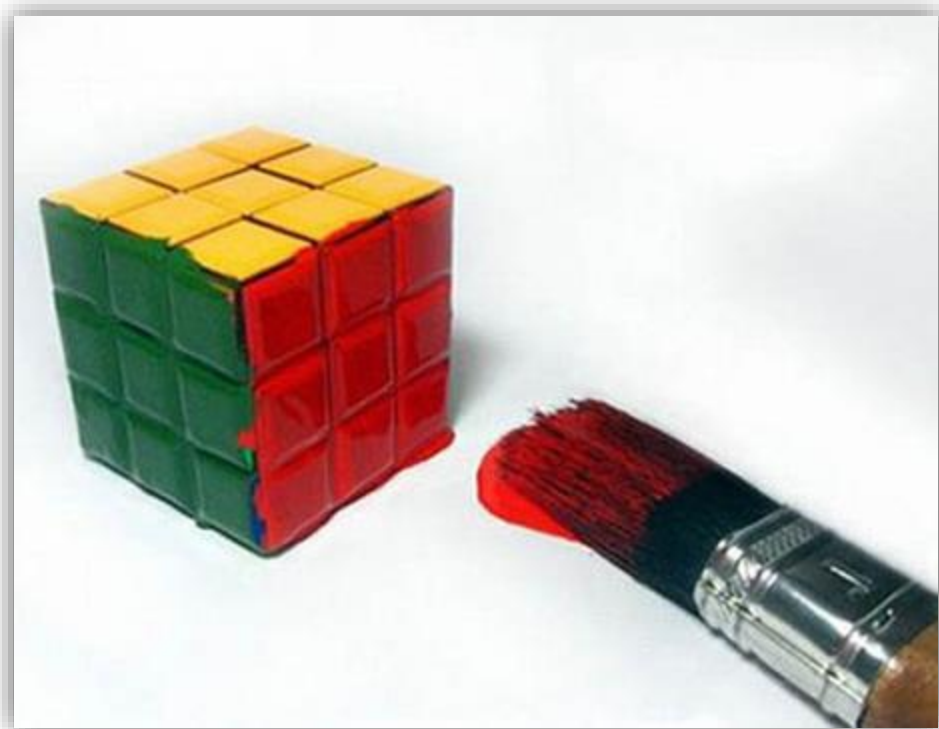
Pel meu cap hi han recorregut infinitat de preguntes relacionades amb el *Què*, el *Com* i el *Per què* d'aquest objecte, i fins i tot la idea esbojarrada de trobar quelcom, alguna ajuda, que, sense la necessitat de desenganxar els gomets de colors i tornar-los a posar a lloc, t'instruís per solucionar aquest complicat trencaclosques.

Llavors vaig pensar que, amb tots els avantatges que aporten les noves tecnologies, la meva afició al cub acompanyada de ganes i dedicació em podrien portar a aconseguir resoldre alguna d'aquestes preguntes.

És per tot això, i després de mesos de reflexió, que em vaig decidir a construir un robot que permetés fer tots els moviments possibles del cub i, per què no?, resoldre'l.

Es pot dir que les dues coses que més m'han marcat la infantesa són els ordinadors i el *cub de Rubik*. Fer un treball en què pugui aglutinar aquests dos mons possiblement serà un dels punts d'inflexió de la meva adolescència i qui sap si de la vida.

Estava una mica preocupat, primer, per aprendre a programar i, segon, per mentalitzar-me del que volia i del que havia de fer per aconseguir completar el treball. M'esperava un estiu llarg i insòlit...



1.1. Objectius

Els meus objectius bàsics des de bon principi van ser els següents:

1. Dissenyar una mecànica que pugui fer moure el cub en totes les seves combinacions.
2. Dur a terme el robot només servomotors i un motor pas a pas.
3. Programar amb el llenguatge *Picaxe* tots els moviments que pot fer el cub.
4. Controlar els moviments amb comandament a distància.

A més a més, em vaig proposar uns altres objectius més complicats si assolía els bàsics:

5. Crear una aplicació per a Android per controlar els moviments amb *Bluetooth*.
6. Dissenyar les peces del robot en 3D i imprimir-les amb una impressora 3D.
7. Resoldre el cub automàticament, és a dir, ordenar-lo en 6 homogènies i monocromàtiques cares.

1.2. Metodologia

Per fer el treball he seguit els passos següents:

- Informar-me sobre programació i aprendre a programar amb llenguatge *Basic Picaxe*.
- Muntar i soldar tots els components a la placa usada pel treball (*Picaxe Imagina*).
- Fer un esbós de la mecànica que hauria de tenir el robot.
- Construir el robot amb fusta.
- Fer acoblaments pels servomotors i comprovar que tots funcionaven correctament connectats alhora.
- Programar els moviments del cub.
- Implementar el comandament a distància per dur a terme aquests moviments.

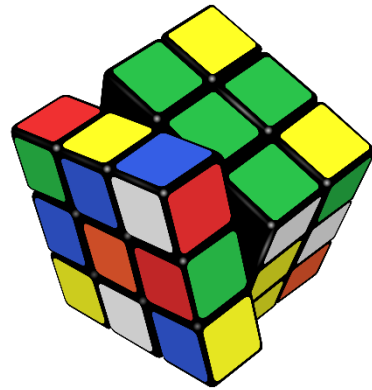
A més, quan em vaig tornar a replantejar els objectius més complicats, vaig seguir la següent metodologia:

- Informar-me i aprendre a fer aplicacions per a Android amb *AppInventor* – Fer una aplicació per controlar els moviments a través de *Bluetooth* que substitueixi el comandament a distància.
- Informar-me i aprendre a dissenyar peces en 3D amb *SketchUp*. – Dissenyar totes les peces del robot en 3D que substitueixin i millorin les peces amb fusta.
- Imprimir les peces amb impressora 3D.
- Posar-ho tot al seu lloc i comprovar-ne el correcte funcionament.

2. EL CUB DE RUBIK

2.1. Què és?

El cub de Rubik és un dels trencaclosques d'un sol jugador més coneguts del món. El disseny del cub té com a característica 27 blocs alineats en un patró de 3 per 3 per 3. Cada costat del cub consta de 9 gomets de colors. El cub és barrejat i el desafiament és revertir tots els costats al disseny original d'un color per costat. El cub de Rubik és primerament un entreteniment domèstic, però també té clubs d'aficionats i es fan tornejos dedicats.



2.2. Història



L'original trencaclosques va ser inventat el 1974 per Erno Rubik¹, un escultor i professor d'arquitectura de la Universitat de Budapest. Aleshores, Rubik no podia imaginar-se que aquesta idea original es convertiria amb els anys en la joguina més venuda del món. Per a ell, el cub era més que una joguina, suposava un instrument per explicar als seus alumnes conceptes acadèmics.

Després d'haver presentat el prototip a alumnes i amics, Rubik va decidir registrar el seu invent el 1975 a l'oficina de patents d'Hongria amb el nom de *cub màgic*.

Els primers cubs màgics van ser fabricats i distribuïts a Hongria per Politechnika, amb una gran acceptació entre els habitants del país. Durant aquests anys, Hongria pertanyia al règim comunista, motiu pel qual resultava difícil que els cubs màgics de Rubik fossin exportats de forma massiva a altres països.

No va ser fins a la celebració de la fira del Joguet de Nuremberg quan la destinació dels cubs màgics canviaria per sempre. En aquesta fira, el cub va sorprendre als assistents i Tom Kremer, un especialista del món de les joguines, es va comprometre a vendre'l i distribuir-lo per tot el món a través de l'empresa Toy Company, i establir un nou nom per al producte: *cub de Rubik*.

¹ Erno Rubik, biografia: http://es.wikipedia.org/wiki/Ern%C5%91_Rubik

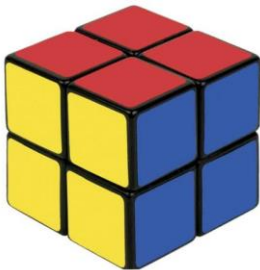
2.3. Curiositats sobre el Cub de Rubik

- Erno Rubik no sabia resoldre el seu cub.
- Hi ha 43 trilions de combinacions de colors possibles, exactament 43.252.003.274.489.856.000 combinacions. Si cada segon féssim una combinació diferent, tardaríem 1400 bilions d'anys a fer-les totes. De la mateixa manera que, si per cada combinació col·loquéssim un cub de Rubik al costat d'un altre, en línia recta, arribaríem a cobrir la distància que separa la Terra de la constel·lació Columba, és a dir, uns 261 anys llum.
- El cub de Rubik no només és la joguina més popular, sinó la més venuda de la història, amb més de 350 milions d'unitats.
- La persona més ràpida a resoldre el cub de Rubik és l'holandès Mats Valk, amb 5.55 segons.

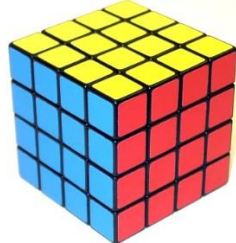
2.4. Altres tipus de cubs

De la mateixa manera que es va crear el cub original de 3x3, també s'han ideat i fabricat altres cubs com el que us mostro a continuació:

- Cub de 2x2



- Cub de 4x4



- Cub de 5x5



- Altres



2.5. Mètodes de resolució

Hi ha moltíssims mètodes de resolució del cub de Rubik, però els més famosos i més usats són:

- **Mètode per a principiants:** Aquest mètode consisteix a resoldre el cub de Rubik per capes, primer la superior, després la central, i finalment la inferior. Aquest és el mètode que farem servir per resoldre el cub durant el treball.
- **Mètode per a experts (Fridrich):** Aquest mètode va ser creat inicialment per Jessica Fridrich, i consisteix a resoldre el cub en tres passos. La part difícil d'aquest mètode que requereix aprendre's un munt d'algorismes complicats de memoritzar. El primer pas consisteix a resoldre les dues primeres capes del cub simultàniament. El segon pas consisteix a orientar les peces de l'última capa i finalment, en el tercer pas, permutant les peces de l'última capa, s'acaba amb la resolució del cub.
- **A cegues (Blindfold Cubing):** El blindfold consisteix a resoldre el cub sense mirarlo, havent-lo inspeccionat prèviament.

2.6. Combinacions

Com ja he esmentat anteriorment durant el treball, el cub de Rubik de 3x3 té 43 trilions de combinacions possibles i s'obté a partir d'aquest càlcul:

$$\frac{8! \cdot 12! \cdot 3^7 \cdot 2^{11}}{2}$$

En què:

- **8!**² – Correspon a totes les combinacions possibles dels vèrtexs.
- **12!/2** – Correspon a totes les combinacions possibles de les arestes. Dividit per dos, perquè el cub consta d'un eix simètric.
- **3⁷** – Per cada moviment de peces, podem orientar els vèrtexs de manera que la torsió total sigui múltiple de 3.
- **2¹¹** – Per cada moviment de peces, podem orientar les arestes de manera que la torsió total sigui múltiple de 2.

² El signe "!" (factorial) és el producte de tots els nombres naturals menors i iguals en aquest cas de 8. És a dir: 1x2x3x4x5x6x7x8.

2.6.1. Número Màgic: 20

Caescuna dels 43 trilions combinacions possibles del cub de Rubik pot ser resolta en vint moviments o menys.

Un equip d'investigadors de Google que han resolt, gracies a un ordinador potent, totes les combinacions del cub de Rubik mostren que cap posició requereix més d'una vintena de moviments, tenint en compte que qualsevol gir de qualsevol cara és un moviment.

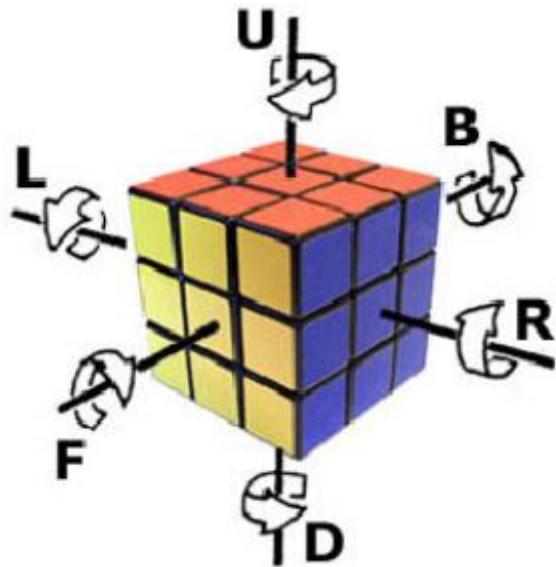
És per això que totes les solucions del cub que s'efectuaran en el treball no requeriran més de 20 moviments³ (per problemes d'optimització, en el robot s'acceptaran un màxim de 25 moviments).

2.7. Notació

El cub de Rubik també té un llenguatge propi a l'hora de referir-se als moviments que cal efectuar. En l'aprenentatge d'algoritmes de resolució, s'hi pot trobar això: "R U R' U R' F R F' U2 R' U R F". Si es desconeix la notació del cub, no es pot interpretar.

Com es veu en la imatge:

- **U (UP)** – Capa superior en sentit horari.
- **B (BACK)** – Capa posterior en sentit horari.
- **D (DOWN)** – Capa inferior en sentit horari.
- **F (FRONT)** – Capa frontal en sentit horari.
- **L (LEFT)** – Capa esquerra en sentit horari.
- **R (RIGHT)** – Capa dreta en sentit horari.



A més, quan es posa una cometa (') després de qualsevol lletra, indica que el sentit de gir en aquell cas serà antihorari. En el cas dels números després de qualsevol lletra, indiquen que cal efectuar *n* vegades aquell moviment.

³ Serien 20 moviments en les condicions dels humans (és a dir: amb dues mans). En el cas del robot (una sola mà), es necessiten més del doble de moviments (més informació a la part pràctica).

3. PLACA IMAGINA

La placa Imagina és una placa base que ha estat desenvolupada per l'Equip de Robolot. Està dissenyada sobretot per alumnes que s'inicien en la programació i per a la realització de projectes tecnològics.



Les seves funcionalitats són extenses, ja que concentra diversos àmbits de l'electrònica i de la mecatrònica (combinació de l'enginyeria mecànica, l'enginyeria electrònica, l'enginyeria de control i l'enginyeria de programari). Es pot fer des d'un senzill projecte d'un semàfor, fins a aplicacions per a la domòtica i tota mena de robots.

La placa Imagina treballa amb el sistema PICAXE. Es tracta d'un sistema de microcontroladors fàcil de programar que utilitza un llenguatge BASIC. El sistema PICAXE explota les característiques de la nova generació de microcontroladors de baix cost FLASH. Aquests microcontroladors poden ser programats una i altra vegada sense la necessitat d'un programador. El poder del sistema PICAXE radica en la seva senzillesa. No necessita cap programador o complex sistema electrònic: el microcontrolador és programat (amb un simple programa en BASIC o un diagrama de flux) mitjançant la connexió al port sèrie (COM) de l'ordinador.

3.1. Característiques

Les principals característiques tècniques són:

- Reduïdes dimensions que faciliten la seva integració en la majoria de muntatges.
- Els components són de baix cost.
- Disposa de 8 sortides⁴ i 8 entrades⁵: 3 d'analògiques/digitals i 5 de digitals.
- Molt versàtil: admet diferents tipus de sensors (digitals/analògics) i col·locacions: (CNY70, GP2Y0D310k, GP2Y0D340K, GP2Y0D805Z, GP2Y0D810z, GP2D120, GP2Y0A21, GP2Y0A41, LM35, TSOP4838, LED020, LDR, finals de cursa...)
- Admet tres tipus de microcontroladors PICAXE, que són: el 20M, el 20X2 i el 20M2.
- Software de programació gratuït.

⁴ Entrades: encarregades de transmetre la informació del microcontrolador als actuadors connectats, ja siguin servomotors, pantalles...

⁵ Sortides: encarregades de rebre informació de l'exterior de la placa, a través de sensors, per processar-la al microcontrolador.

4. DISSENY EN 3D – GOOGLE SKETCHUP

El Google SketchUp és un programa de modelat i disseny en 3D que ofereix una bona alternativa a programes professionals poc intuïtius, com *AutoCAD* o *DraftSight*. Per tant, l'SketchUp és un bon programa per als qui s'inicien en el món del disseny en 3 dimensions i l'arquitectura.

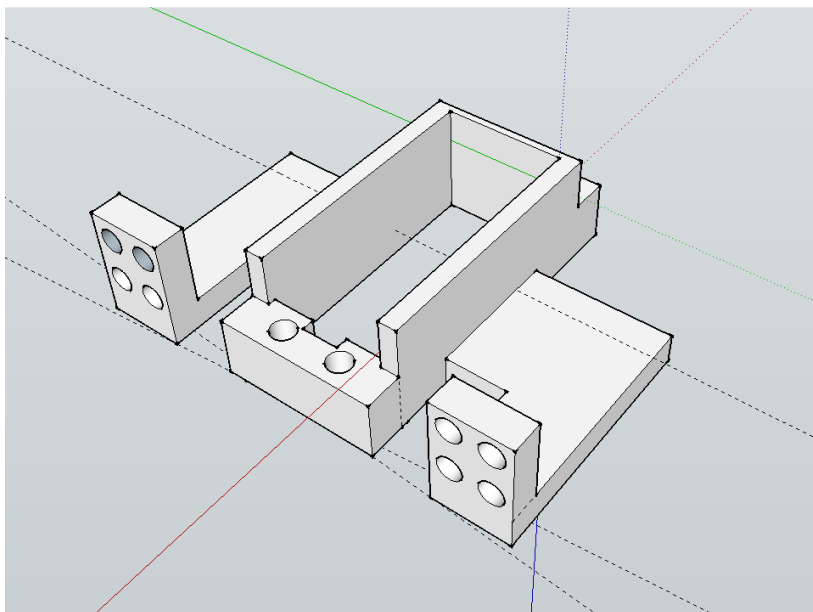


Permet modelar en 3D edificis, paisatges, escenaris, mobiliari, persones i qualsevol objecte o article que imagini el dissenyador o dibuixant. El programa inclou una galeria d'objectes, textures i imatges disponibles per descarregar.

Alguns dels complements de l'SketchUp són:

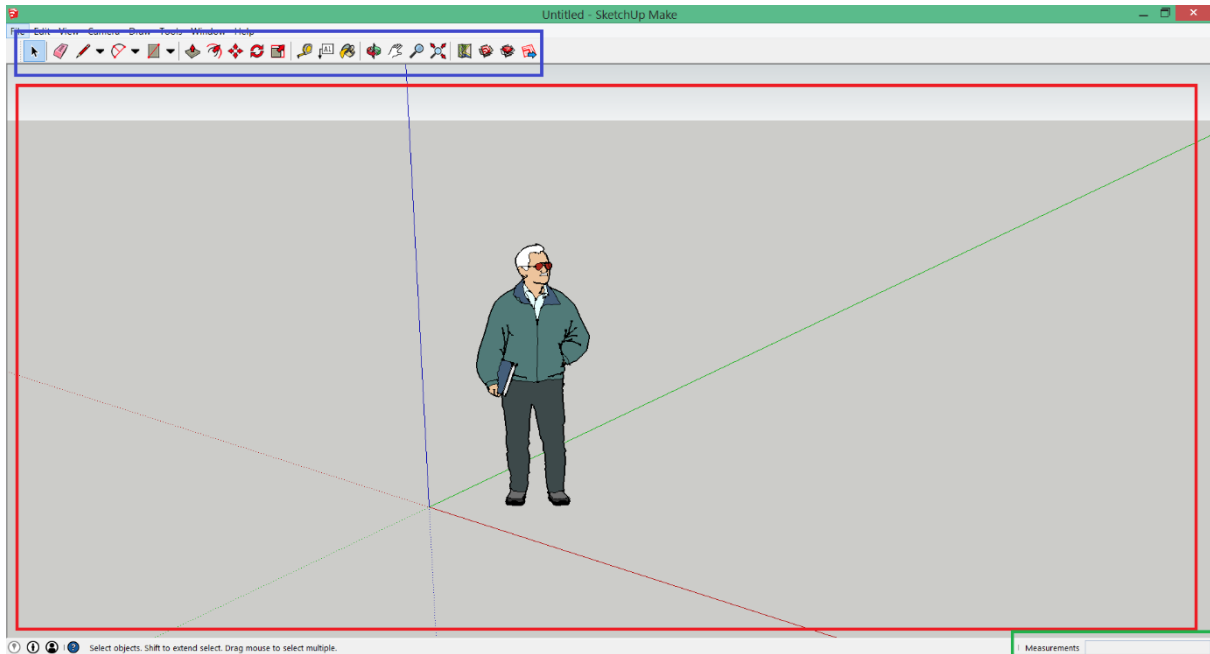
- **Plugins:** Aquests petits o grans programes que permeten fer petits canvis en la funcionalitat del programa es diuen *plugins*. N'hi ha una gran varietat, amb aplicacions particulars com el dibuix automatitzat de sostres, peces d'acer, cornises, etc.
- **Render:** Sketchup s'ocupa de la geometria, però hi ha programes per fer una representació *fotorealista*, els més populars són Vray, Kerkythea, Twilight.

A més, a l'SketchUp s'hi poden integrar eines de maquetació gràfica per realitzar-hi presentacions professionals. Hi ha dues versions d'aquest programa: l'SketchUp (gratuït) i l'SketchUP PRO (de pagament).



Exemple de peça dissenyada en 3D pel treball

4.1. Com funciona?



Pàgina inicial Google SketchUp 15

- La zona **BLAVA** és la barra d'eines on s'ubiquen tots els recursos que caldran per fer el disseny. S'hi poden trobar (d'esquerra a dreta):

Eina de selecció, goma d'esborrar, eina de línies, eina de figures irregulars, eina de figures regulars, eina de volum regular, eina de volum irregular, eina de moviment, eina de rotació, eina d'escala, mesurador de distància, eina d'inserció de text, eina de pintura, eina rotació en 3D, eina moviment en 3D, lupa, eina de centrar disseny.



- La zona **VERMELLA** és el camp de treball i on es veuen les peces que s'estan dissenyant.
- La zona **VERDA** és on s'indiquen les mesures del disseny. En treballar en 3D, segons l'angle en què s'observin les peces, les mesures poden semblar diferents a la real. Aquesta zona s'encarrega de donar el valor exacte de la mesura, independentment de l'angle o punt de vista que s'observi.

5. CREACIÓ D'UNA APLICACIÓ PER A ANDROID – APP INVENTOR

App Inventor és un *framework* (entorn de treball) creat inicialment pel MIT (Institut tecnològic de Massachusetts) i comprat després per Google perquè qualsevol persona amb interès pugui crear la seva pròpia aplicació mòbil, ja sigui per a la seva empresa, a nivell personal o per a altres interessos.

Per crear una aplicació amb *App Inventor* calen tres passos:

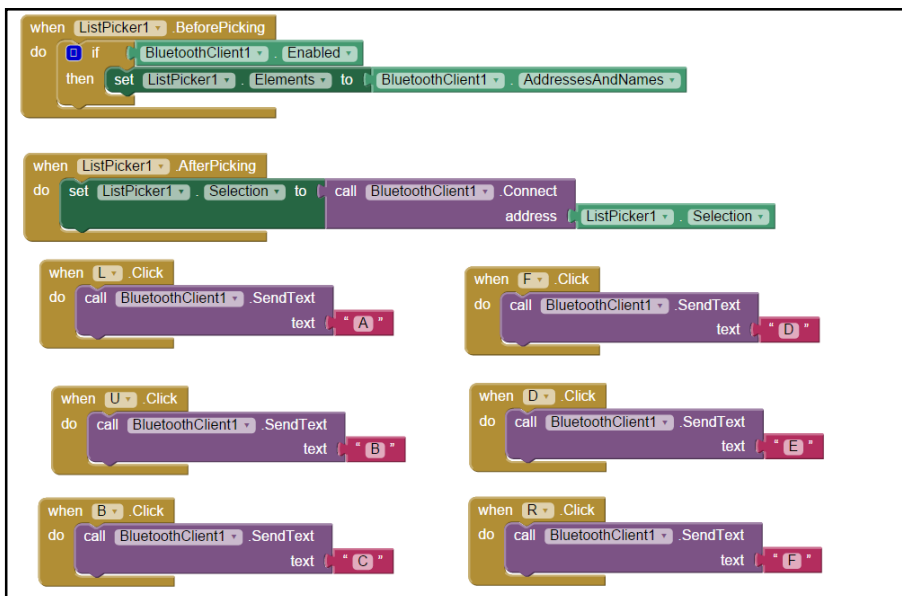
- El disseny de l'aplicació, en la qual se seleccionen els components per a la seva aplicació, l'estructuració i el contingut en pantalla.
- L'editor de blocs o programació de blocs, on s'aniran escollint els blocs (o talls de programació) que siguin necessaris segons l'aplicació que es vulgui fer.



L'aplicació apareixerà pas a pas a la pantalla del telèfon a mesura que s'hi afegeixi contingut, perquè es pugui comprovar la feina.

- Per acabar, es pot empaquetar l'aplicació i produir una aplicació independent per instal·lar-la.

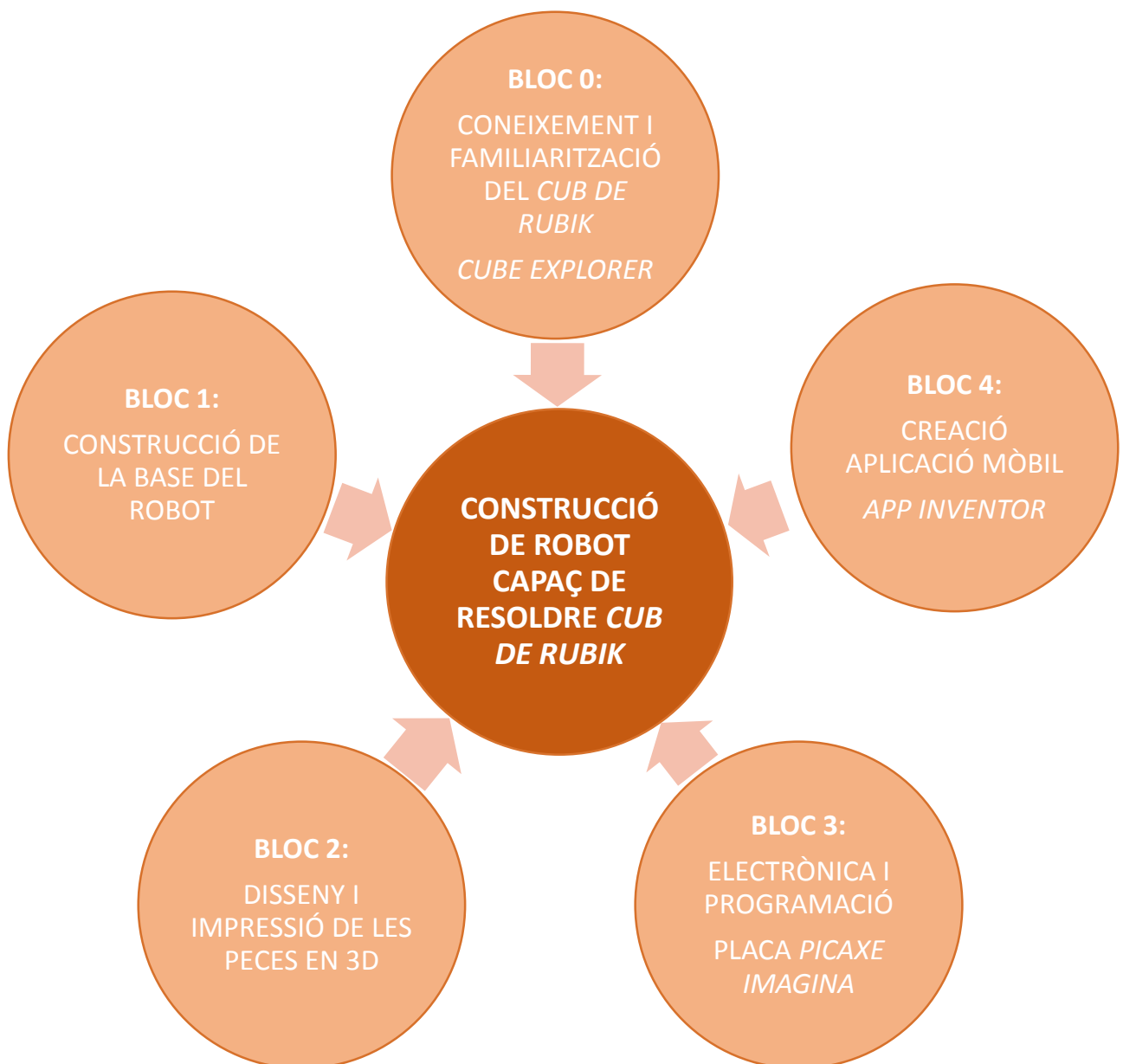
Si no es té un telèfon Android, es poden construir les pròpies aplicacions utilitzant l'emulador d'Android, el programari que s'executa a l'ordinador i es comporta com el telèfon.



Exemple de programació per blocs amb AppInventor

6. PART PRÀCTICA: CONSTRUCCIÓ DE ROBOT CAPAÇ DE RESOLDRE CUB DE RUBIK

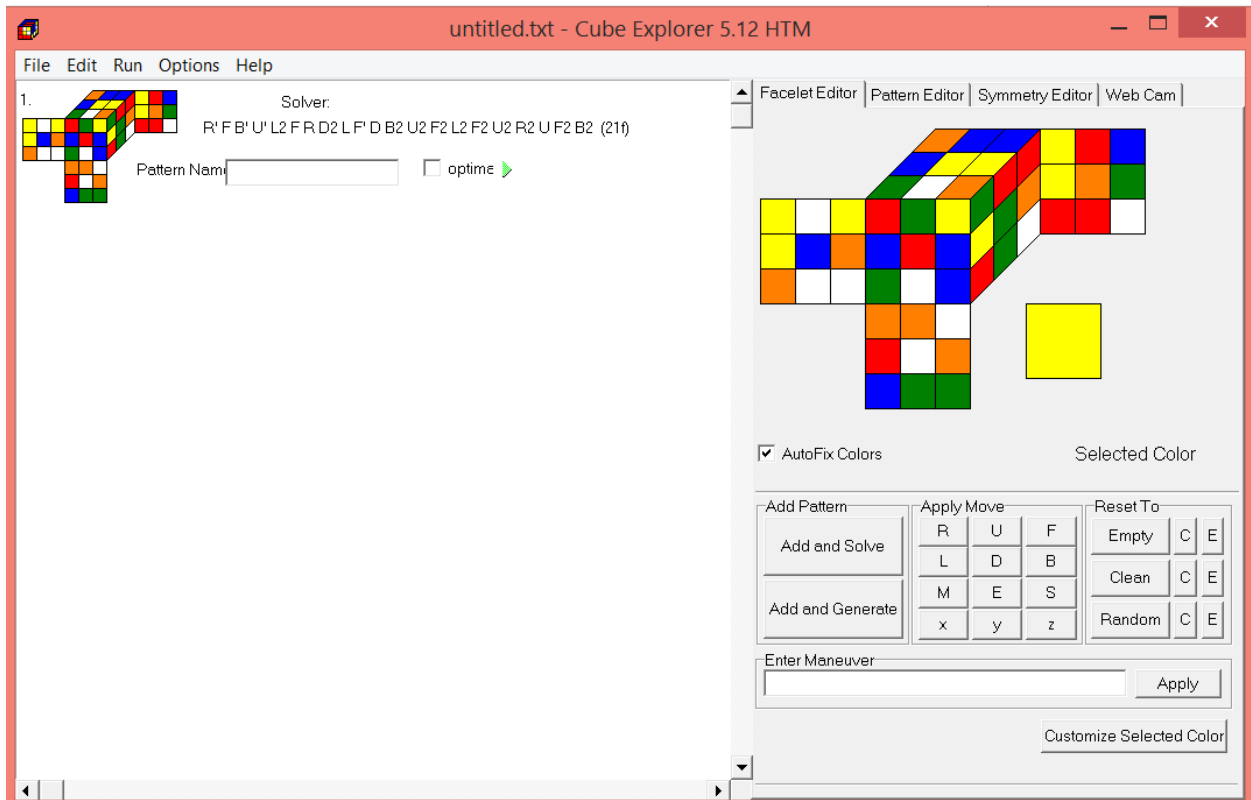
Per dur a terme la part pràctica del treball s'han de compaginar les quatre parts d'escrites a la part teòrica.



7. BLOC 0: CONEIXEMENT I FAMILIARITZACIÓ DEL CUB DE RUBIK – CUBE EXPLORER

Aquest bloc serveix d'introducció a la construcció del robot, ja que abans de decidir-me pel treball actual vaig passar molt de temps buscant informació sobre com podia realitzar-lo.

Finalment, vaig trobar un *software* lliure creat per Herbert Kociemba anomenat *Cube Explorer* que permetia escanejar el cub de Rubik amb una càmera web i, usant el mètode Kociemba "*The Two-Phase-Algorithm*", et donava en un màxim de 25 moviments la solució exacta del cub escanejat. És per això que, construint una mecànica adequada que pogués efectuar aquests moviments, podria crear un robot capaç de resoldre el cub de Rubik.



Pàgina principal *Cube Explorer*.

Amb la descoberta d'aquest programa se m'obria una porta a poder realitzar aquest treball, ja que la meua intenció era des de bon principi crear un robot capaç d'automatitzar aquest procés i no crear un nou mètode de resolució.

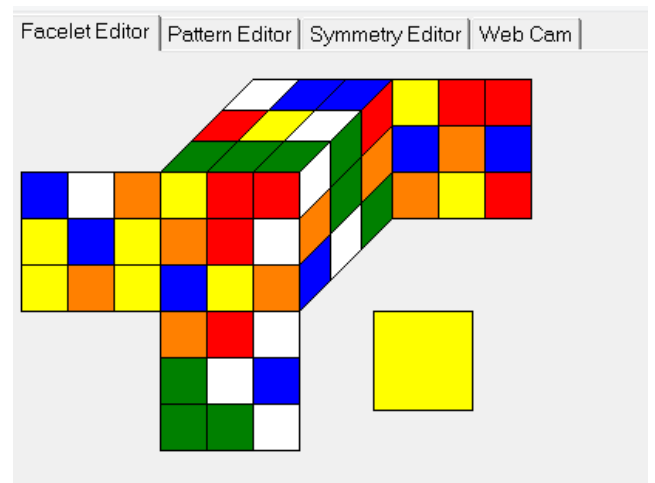
7.1. Càmera web

El programa *Cube Explorer* té com a opció la implementació de la càmera web a l'hora d'escanejar els colors del cub. És a dir, permet crear un model en 2D del teu cub simplement enfocant totes les cares a la càmera. Es segueixen els passos següents:

- Primer de tot, s'enfoquen totes les cares del cub a la càmera seguint un ordre determinat que estableix el programa.



- Una vegada escanejat, el programa crea el model en 2D del teu cub.



- Finalment, calcula la solució amb els mínims moviments del cub. En el cas següent, s'aconsegueix resoldre el cub amb 22 moviments.

1. Solver: `B U D R U' B' R2 F' L' B' U2 R' D2 R D2 R2 F2 R U2 F2 D2 F2 (22f)`

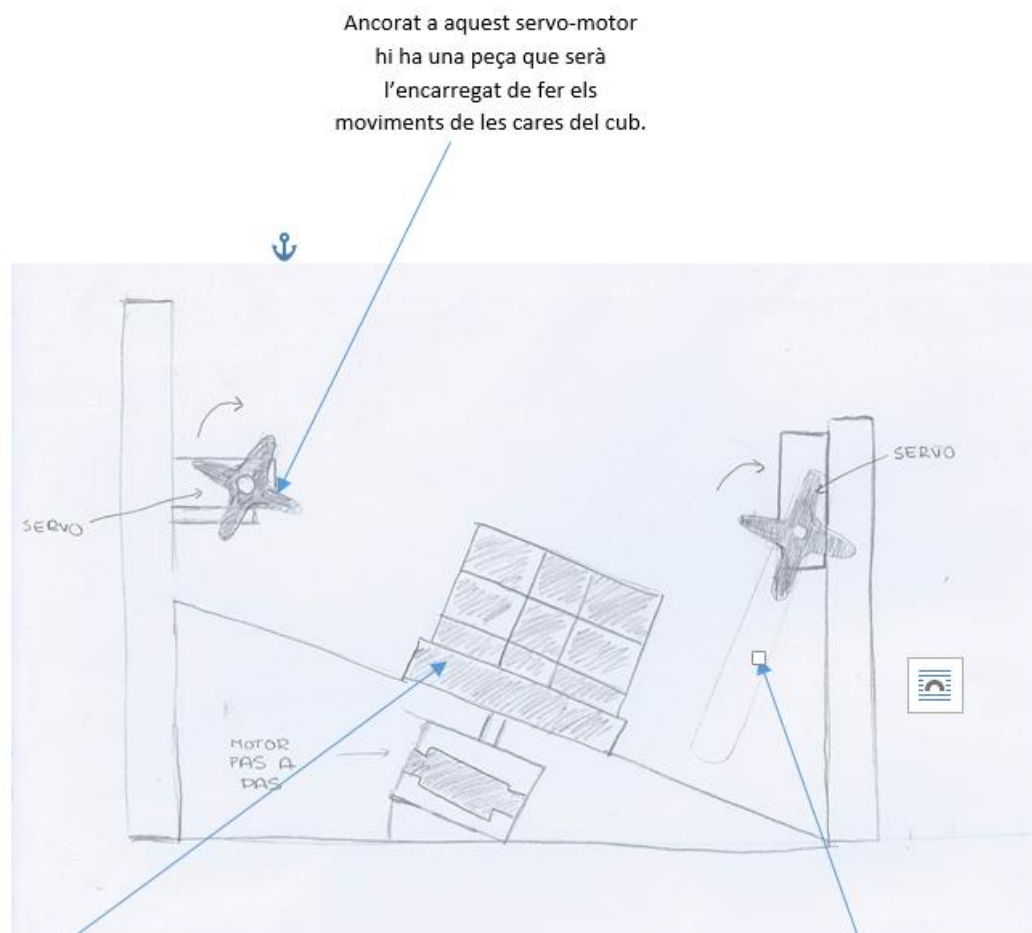
Pattern Name: optime

8. BLOC 1: CONSTRUCCIÓ DE LA BASE DEL ROBOT

Com a primer contacte amb el robot, vaig pensar en com havia de ser la mecànica per aconseguir moure el cub en totes les direccions sense imitar complicats i costosos dissenys creats per intentar resoldre el cub més ràpid que les persones.

És per això que, després de fer la pluja d'idees, em vaig decidir per fer el robot dels materials més barats i quotidians possibles.

Per aquesta raó vaig prendre la decisió de fer la base de fusta i les altres peces de "palets" (l'eina de fusta amb què els metges miren el coll dels pacients). Aquest va ser el primer esbós de com havia de ser el robot:



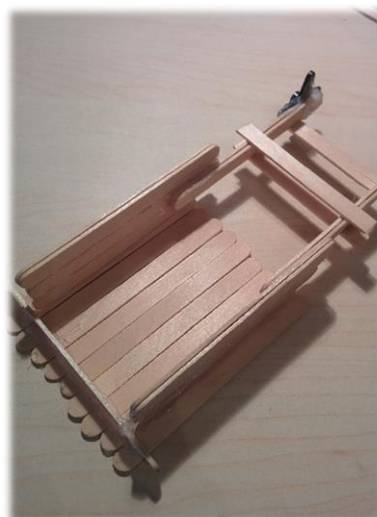
Ancorat a aquest servo-motor hi ha una peça que serà l'encarregat de fer els moviments de les cares del cub.

Base pel cub: Una plataforma de mides lleugerament superiors al cub que té un suport a la part inferior que permet el moviment de l'eix X del cub (moviments de 90,180,270,360 graus perpendiculars a la base)

Aquesta peça és l'encarregada de fer els moviments de l'eix Y del cub. (moviments de 0 i 90 graus que el cub pot fer paral·lels a la base)

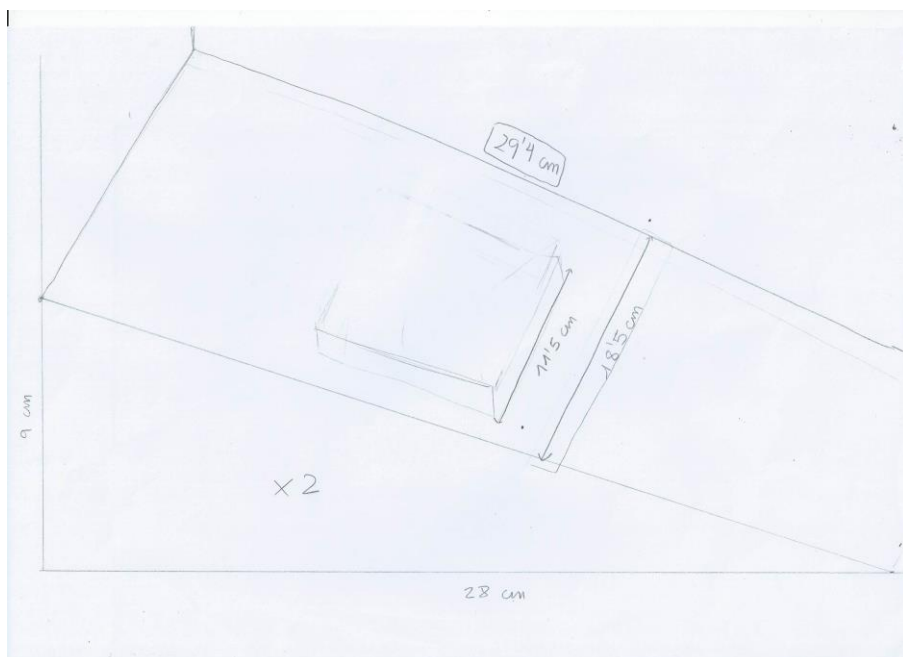
8.1. Construcció de les primeres peces

Aquestes són les primeres peces que vaig construir del robot amb l'ajuda dels "palets" de fusta i silicona calenta. La primera consistia en el lloc on s'estacionava el cub i la segona era la peça que feia girar les cares.



8.2. Construcció de l'estructura

L'estructura del robot havia de ser suficientment gran com per enquibir tots els components del robot, per tant, vaig fer aquest croquis amb les mides i ho vaig tallar amb un llistó de fusta, tal com es veu a continuació:



8.3. Aparició d'alternatives durant el procés

Durant el procés de construcció del robot amb fusta, l'INS La Bisbal va adquirir una impressora 3D; això va fer canviar moltíssim les meves aspiracions, ja que s'obria una porta importantíssima per poder millorar el robot en seguretat i estètica.

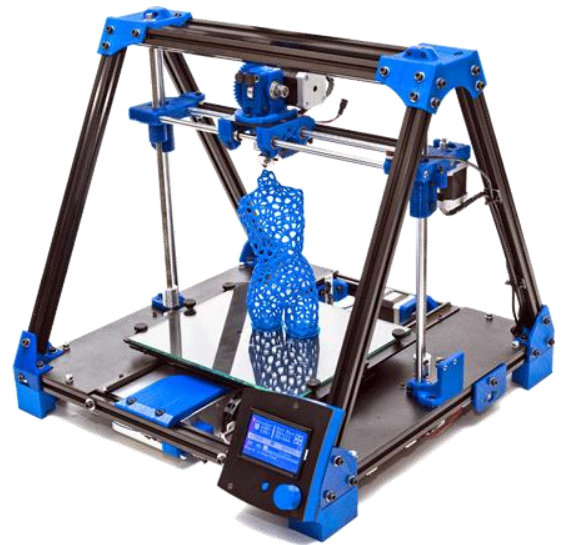
Per aquesta raó i pel fet que estava tenint problemes de subjecció dels servomotors a l'estructura, vaig decidir canviar el rumb del treball i dissenyar les meves pròpies peces en 3D amb les mides exactes dels servomotors. Gràcies a això, es van eliminar les antiestètiques brides que els subjectaven i se'n millorava la fiabilitat. Per tant, vaig decidir deixar la construcció de peces en fusta (només conservant la base del robot) i endinsar-me en el disseny d'aquestes peces en 3D. En el següent bloc, hi explicaré el procés de disseny i impressió de les peces en 3D.

9. BLOC 2: DISSENY, IMPRESSIÓ I MUNTATGE DE LES PECES EN 3D

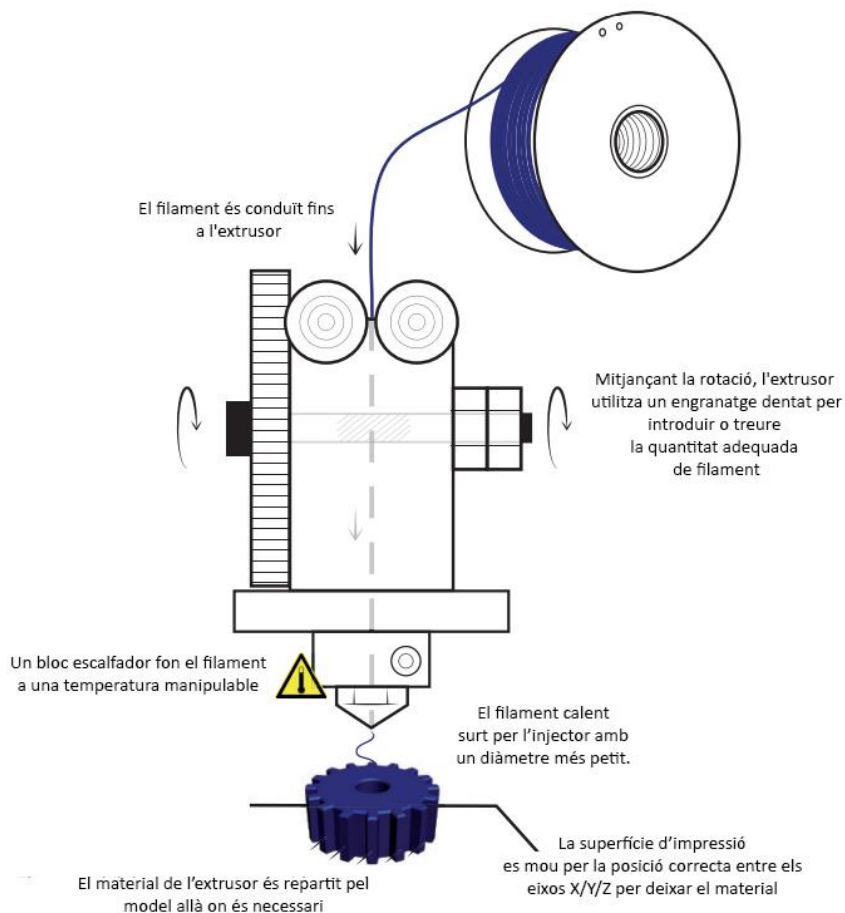
9.1. Impressora 3D – REPRAP BCN+

A l'institut disposem de la BCN3D+, que és una impressora 3D basada en la tecnologia FFF (per les seves sigles en anglès: *Fused Filament Fabrication*).

Aquesta tecnologia permet fabricar peces per mitjà de la deposició de material fos. Per això, es parteix d'un model digital que es secciona en capes fines. El capçal realitza els moviments per dipositar el material d'una capa, després realitza un petit desplaçament vertical i continua amb la següent capa. El procés es repeteix fins a realitzar la figura completa.

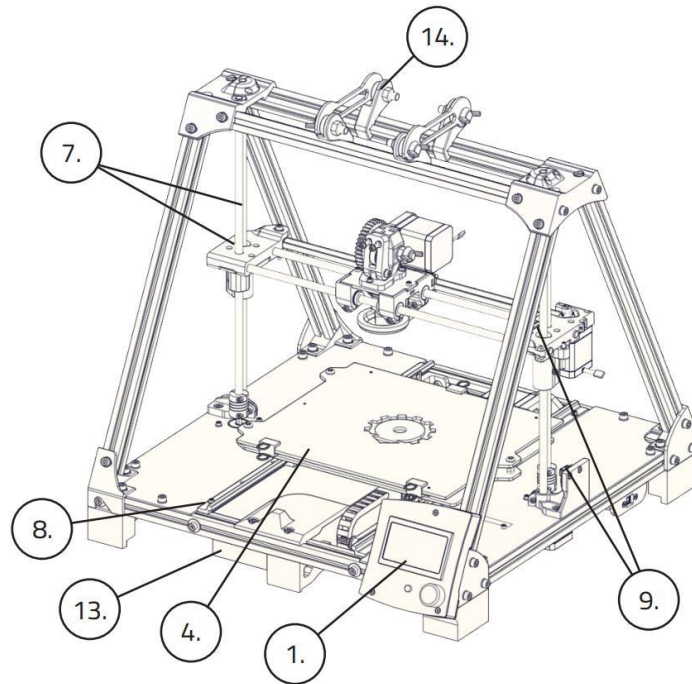


9.1.1. Com funciona l'extrusió de material?

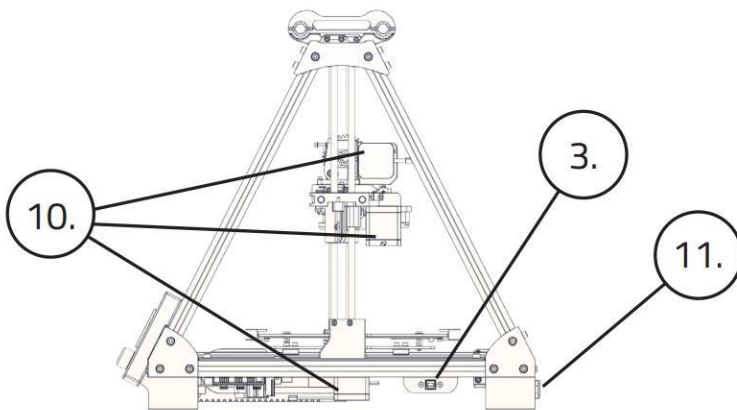


9.1.2. Esquema de la impressora REPRAP BCN+

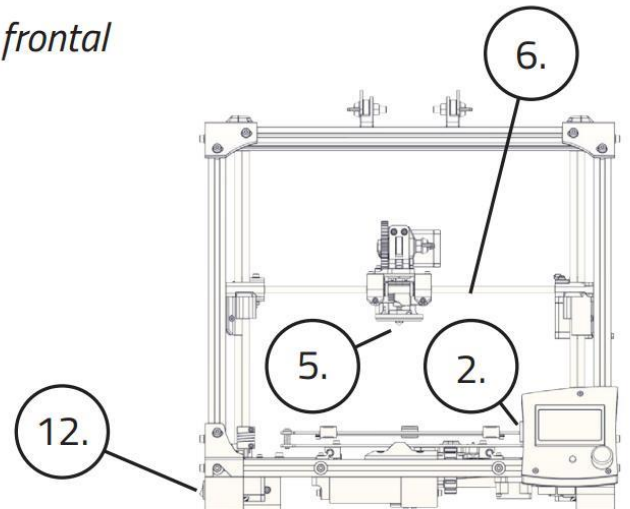
Perspectiva



Vista esquerra



Vista frontal



1. Pantalla LCD

2. Ranura targeta SD

3. Connector USB

4. Hotbed (base calefactada)

5. Hotend (extrusor)

6. Guies eix X

7. Varettes roscades i guies eix Z

8. Guies lineals eix Y

9. Finals de carrera mecànics (Endstop)

10. Motors pas a pas

11. Presa de corrent

12. Botó d'encesa

13. Font d'alimentació

14. Suport per material

9.1.3. Materials d'impressió – Filaments

La BCN3D + presenta els seus millors resultats d'impressió amb PLA, ABS i Nylon, encara que accepta molts altres materials, en format filament de tipus plàstic de 3 mm. A continuació, es presenta una breu descripció dels materials acceptats actualment:

- **PLA** (poliàcid làctic) - és el material d'impressió 3D per excel·lència. És el material recomanat per iniciar-se en aquest tipus de tecnologia de fabricació. Hi ha una gran varietat de colors i permet aconseguir geometries complexes sense una gran dificultat. L'acabat és lleugerament brillant o semimat.

Per a la impressió, es treballa a temperatures que oscil·len entre els 195 °C i els 220 °C, en funció del proveïdor de material i de la velocitat d'impressió. Presenta una gran adherència a la base de vidre.

- **ABS** (acrilonitril butadiè estirè) - és un altre dels materials més usats en la impressió 3D, i supleix alguna de les febleses del PLA. La seva impressió és més complexa i requereix certes cures. No es recomana l'ABS per a usuaris inexperts. La varietat de colors també és molt alta i l'acabat és semimat o mat.

L'ABS s'imprimeix a temperatures entre 210 °C i 240 °C en funció del proveïdor de material i de la velocitat d'impressió. L'ABS presenta un millor comportament que el PLA davant els voladissos. L'adherència a la base és més delicada que en el PLA, a causa de la contracció del material en refredar-se, i sol requerir laca adhesiva per adherir-se a la base de vidre.

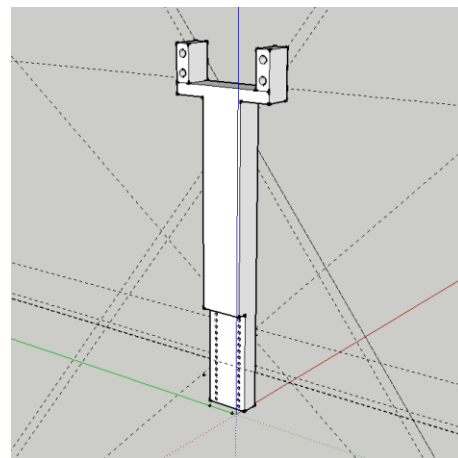
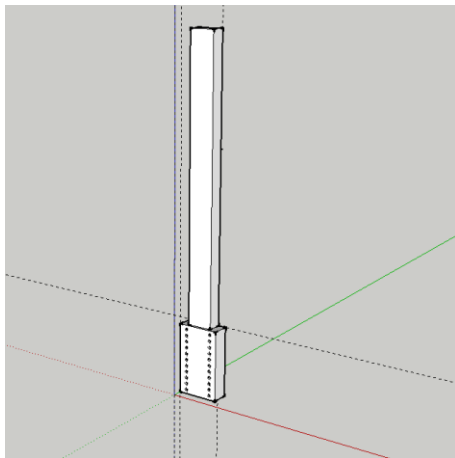
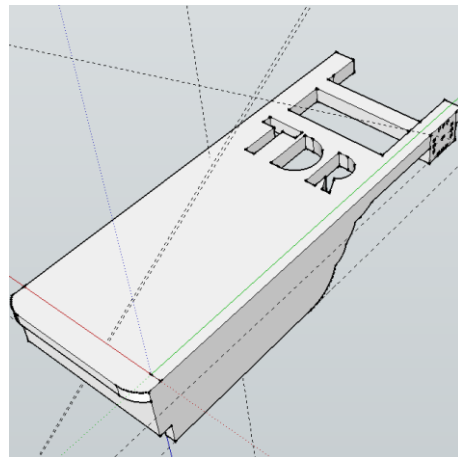
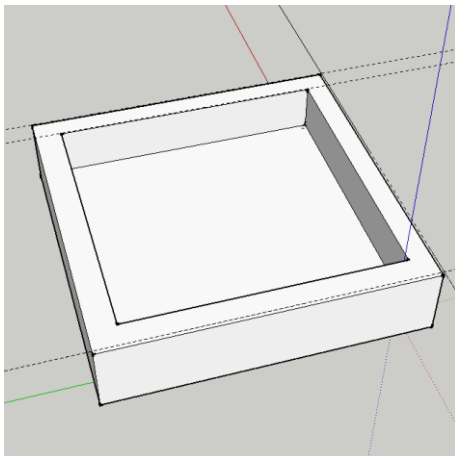
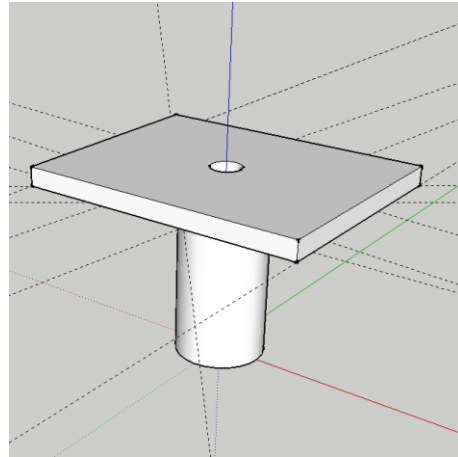
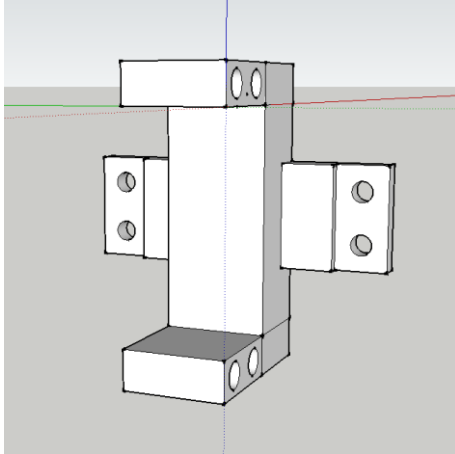
- **Nylon** - És una poliamida fàcilment imprimible, però amb grans problemes d'adherència a la base. És un material lleugerament translúcid i que presenta certa flexibilitat. El color és blanc, tot i que accepta el tintat. El seu ús és recomanat per evitar el desgast en peces que pateixin fricció en el seu ús habitual. S'imprimeix a una temperatura semblant a la del PLA, i sol requerir l'ús de tècniques alternatives per garantir l'adherència a la base.

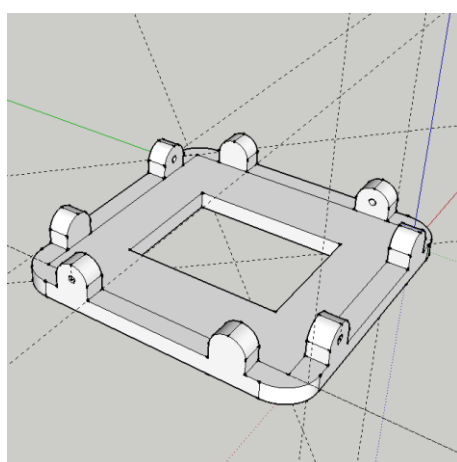
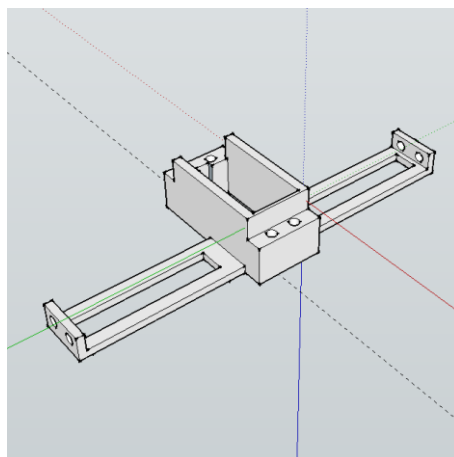
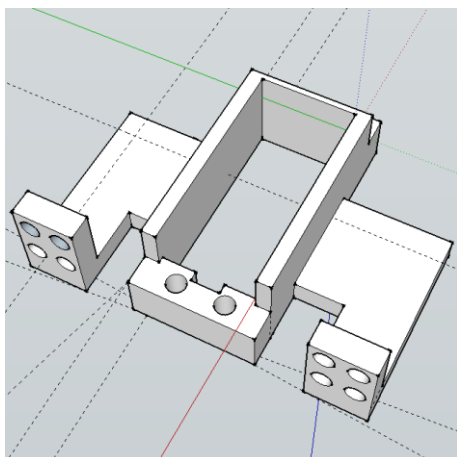
A més, hi ha uns materials en fase experimental:

- **HIPS** - És un material amb unes propietats similars a l'ABS, però que reacciona davant de solvents diferents (limonè). Se sol utilitzar com a material de suport per imprimir en ABS.
- **PVA** (polyvinyl alcohol) - és un material enfocat únicament en la fabricació de suports per a la impressió de volades. És soluble en aigua, i, per tant, pot eliminar-se d'una manera senzilla i segura.

9.2. Disseny de les peces

Per dissenyar les peces en 3D pel robot vaig fer servir el Google SketchUp, el funcionament del qual està prèviament explicat a la part teòrica del treball. A continuació mostro un seguit d'imatges dels dissenys en 3D que vaig crear pel treball.





9.3. Impressió de les peces amb impressora 3D

Per imprimir els dissenys creats amb SketchUP vaig seguir els següents passos:

- Descarregar l'extensió *SketchUp STL* per SketchUp
- Exportar els dissenys amb format .STL
- Obrir l'.STL amb un programa anomenat NetFabb
- Configurar els paràmetres de la impressora 3D a el Cura
- Importar els arxius .STL reparats al Cura.
- Crear l'arxiu .GCODE
- Posar GCODE a targeta SD, connectar-la a la impressora i posar-la a imprimir.

9.3.1. Exportar dissenys a .STL – Extensió SketchUp ST

Per exportar els dissenys a STL amb l'SketchUp, primer de tot cal descarregar-se una extensió anomenada *SketchUp STL*; aquesta extensió es pot baixar gratuïtament des de l'*Extension Warehouse*, que és com una botiga d'extensions diverses pel programa.

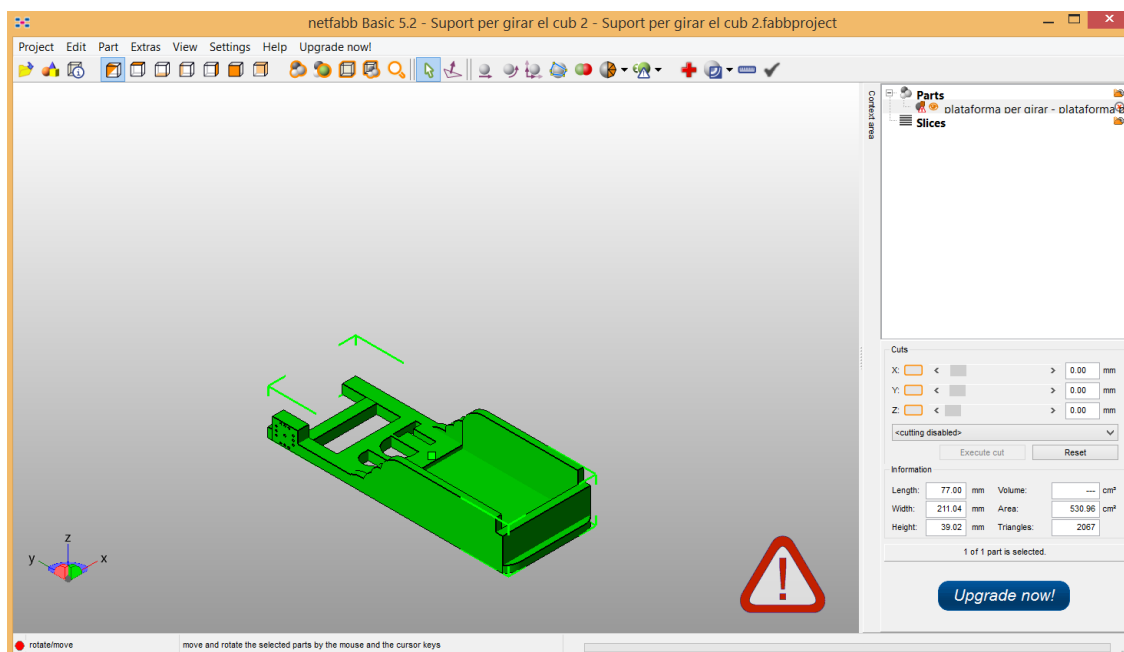
Una vegada instal·lada, quan s'exporti el disseny es mostrarà l'opció: "Export .STL"

9.3.2. Reparar possibles errors d'impressió - NetFabb

Abans d'imprimir un disseny en 3D, cal verificar que no tingui problemes d'impressió, que tots els costats es toquin entre si, és a dir, que no hi hagi forats. Per aquesta raó, s'haurà de passar l'arxiu .STL per un programa anomenat NetFabb, que s'encarrega de trobar irregularitats a les peces dissenyades. A part de trobar aquestes irregularitats, les repara i et deixa exportar l'arxiu .STL lliure de problemes d'impressió.



En cas que el disseny no estigui correcte apareixerà el signe d'exclamació a baix a la dreta com a la imatge següent. Això es soluciona amb l'opció "Automatic Repair", fent clic dret a l'objecte i extraient l'arxiu .STL reparat.

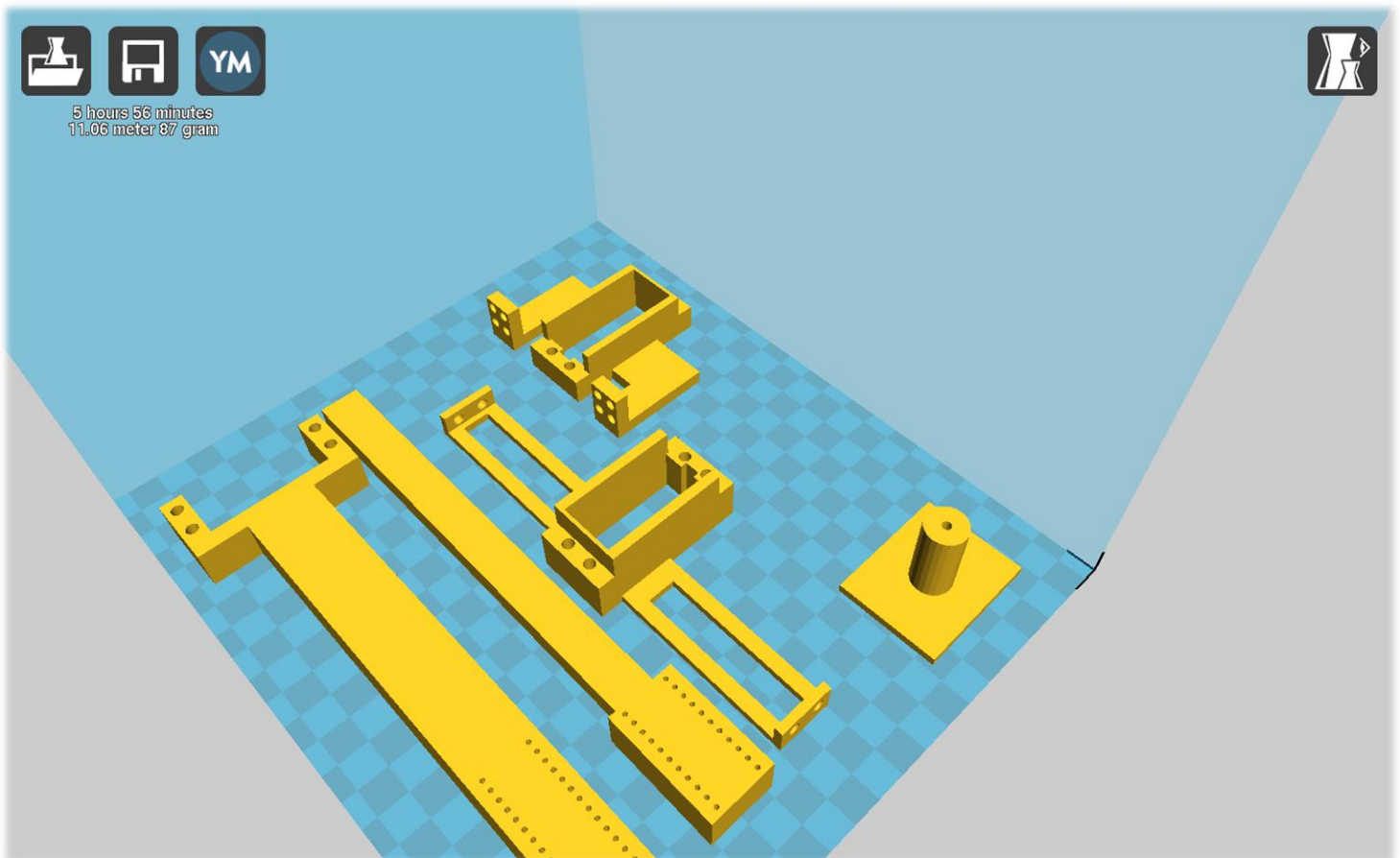


9.3.3. Creació del GCODE

Arribats a aquest punt, ja disposava d'un model 3D a punt per imprimir. No obstant això, calia un últim pas. S'havia de convertir l'arxiu en format .STL a un arxiu que la impressora pogués entendre per fabricar el model. En altres paraules, s'ha de generar l'arxiu de GCODE. El GCODE és un llenguatge informàtic usat per controlar màquines de CNC⁶, incloses la majoria d'impressores 3D.

Es tracta, bàsicament, d'una llista d'instruccions que indiquen les posicions i velocitats a les quals s'han de moure cadascun dels eixos, i les temperatures de fusió del material.

Per crear l'arxiu GCODE vaig usar un programa anomenat *Cura*. Amb aquest programa no solament pots visualitzar, rotar i col·locar les peces de la mateixa manera que s'imprimiran, sinó que pots saber el temps que tardarà a imprimir si has configurat prèviament les característiques de la impressora.



Exemple de peces col·locades al Cura a punt per imprimir. 5 hores 56 minuts d'impressió.

⁶ CNC: Màquines de control numèric, és a dir, pot controlar el posicionament d'un òrgan mecànic mòbil mitjançant ordres elaborades de forma totalment automàtica a partir d'informacions numèriques en temps real.

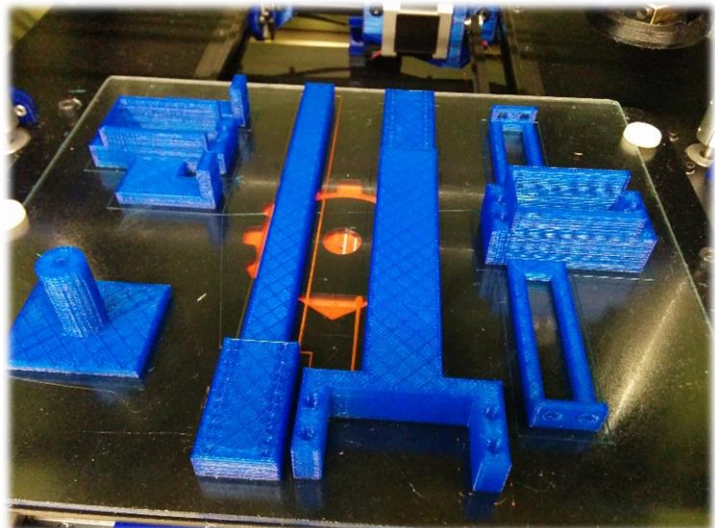
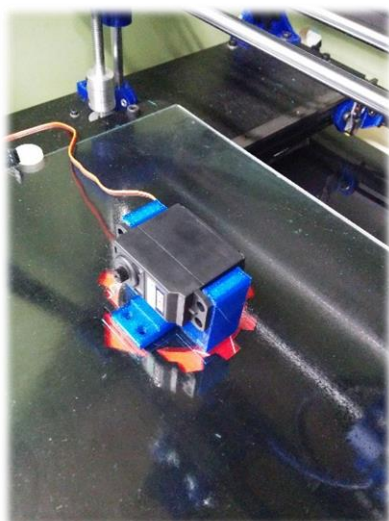
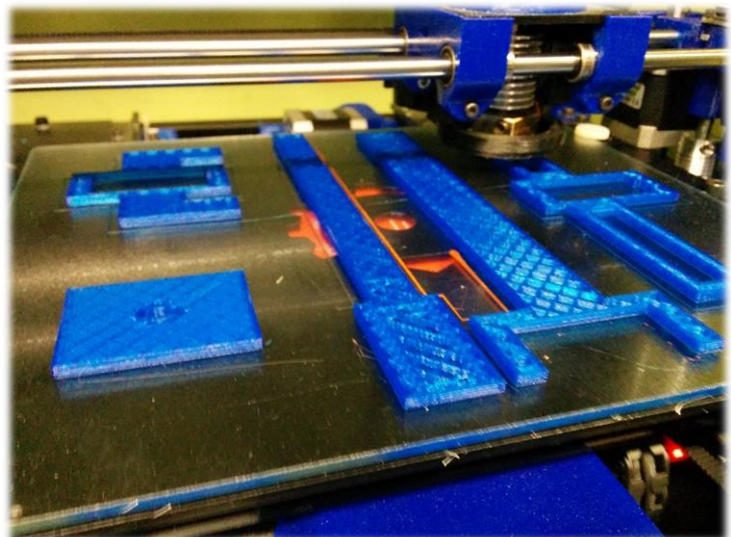
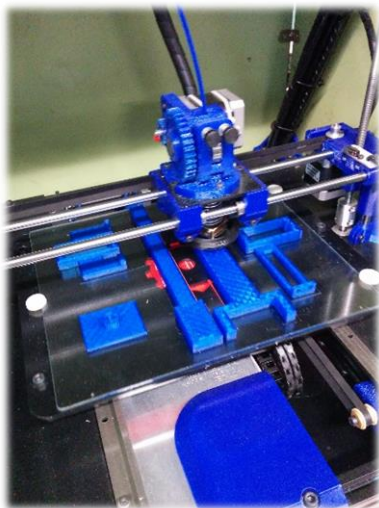
9.3.4. Impressió del GCODE

Finalment, només cal transferir el Gcode a la impressora per començar a imprimir. Per a això, hi ha dos procediments diferents, en funció de la comoditat i de la disposició de la impressora i l'ordinador:

- Mitjançant pantalla LCD. Utilitzar una targeta SD, carregada amb l'arxiu Gcode a imprimir, i introduir-la en la ranura lateral de la pantalla de la BCN3D+. Un cop inserida, prémer el botó de control i escollir l'opció "Print from SD" per seleccionar la peça.
- Connexió al PC. Fer servir un programari per connectar la impressora a l'ordinador a través d'USB.

9.4. Impressió de les peces pel robot

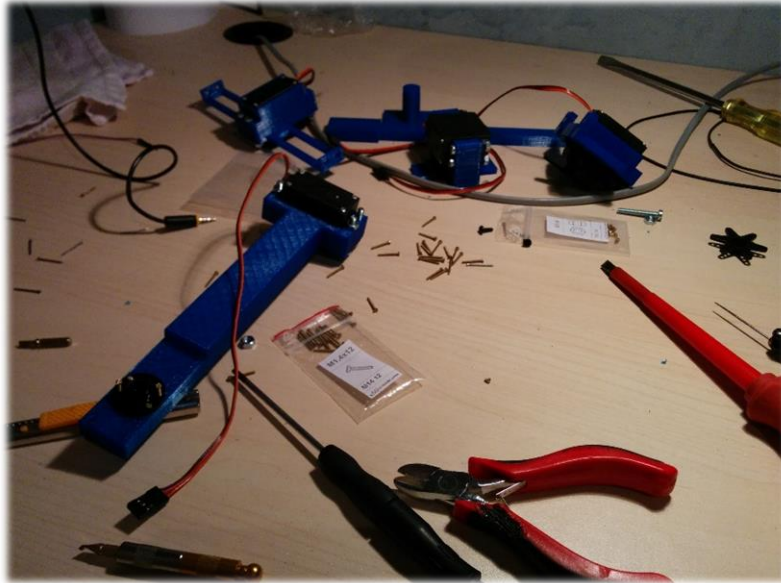
Per imprimir totes les peces del robot vaig necessitar 3 impressions. Dues de 7 hores de durada i una de 2 hores. Aquestes són algunes fotografies dels processos d'impressió:



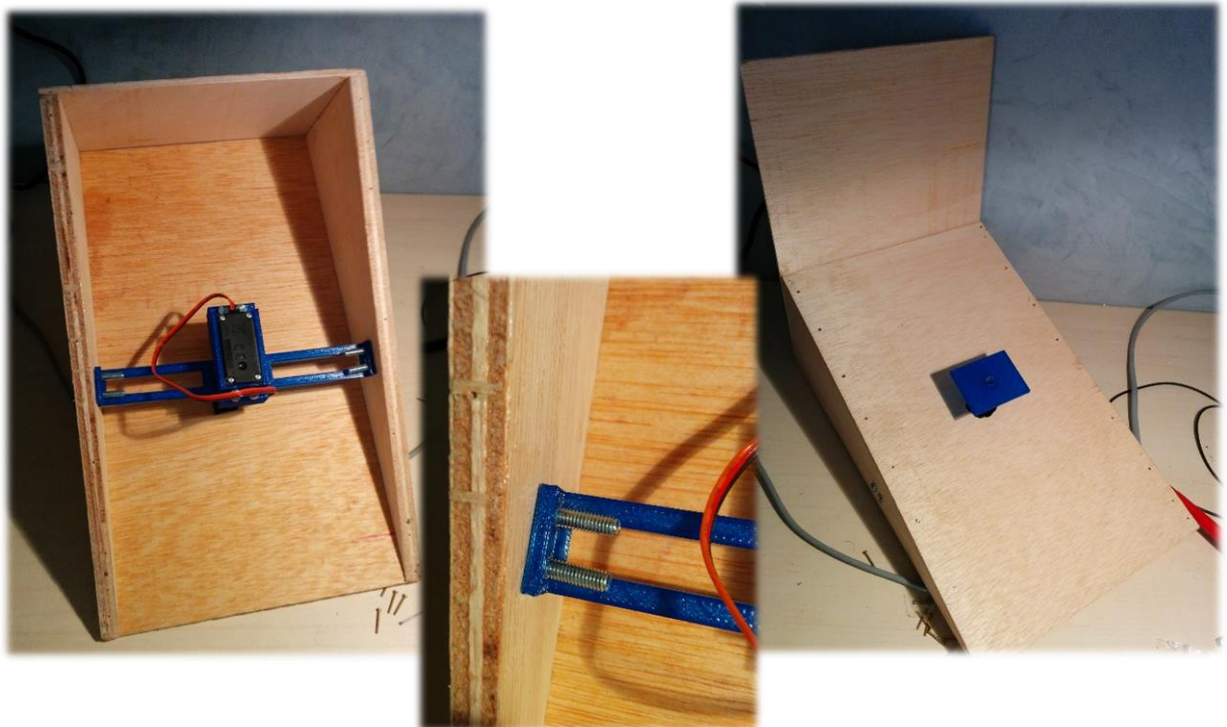
9.5. Muntatge del robot

Com que ja tenia les fustes de la base tallades des del primer bloc, només vaig haver d'acoblar-hi les peces impreses, que ja estaven dissenyades amb els corresponents forats per ser fixades amb cargols. Vaig seguir els següents passos:

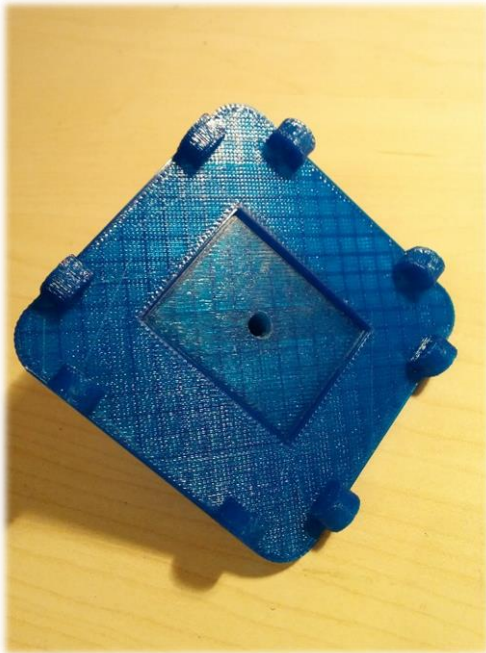
- Fixar tots els servomotors en els seus corresponents llocs de les peces impreses amb rosques de mètric 4.



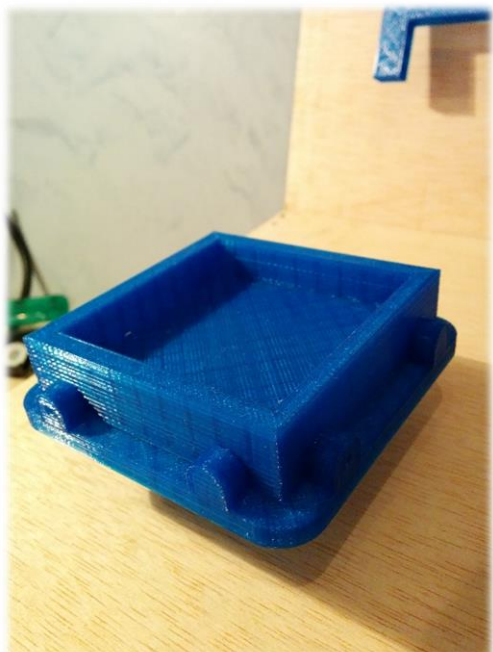
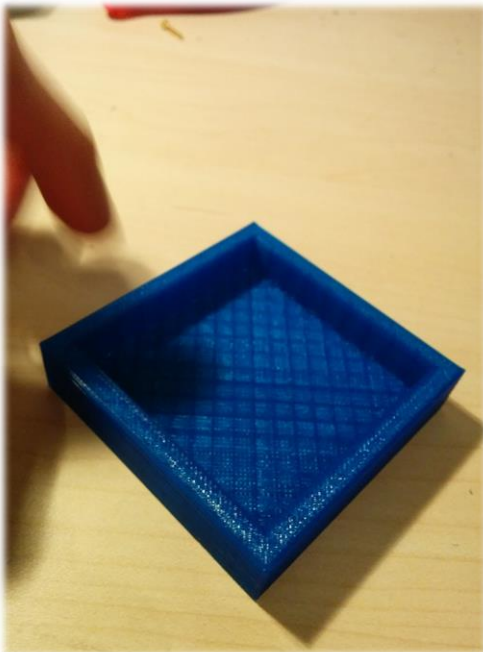
- Introduir el servo dins el suport per al rotor i foradar la base de fusta per poder unir-lo amb l'eix, que ha d'anar perpendicular a la base.



- Enquibir el rotor sobre el seu suport.

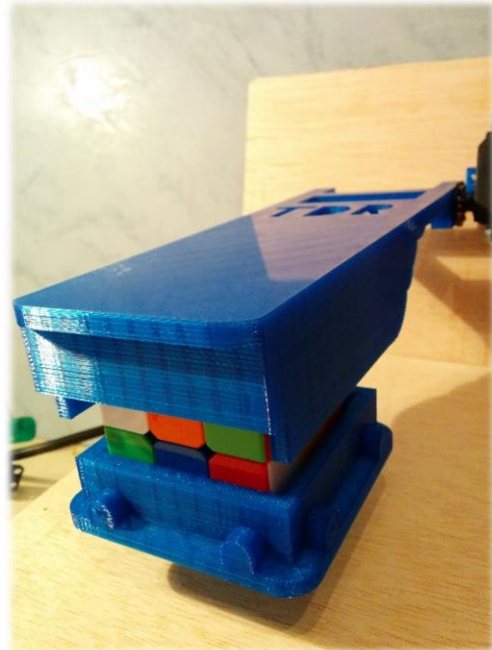
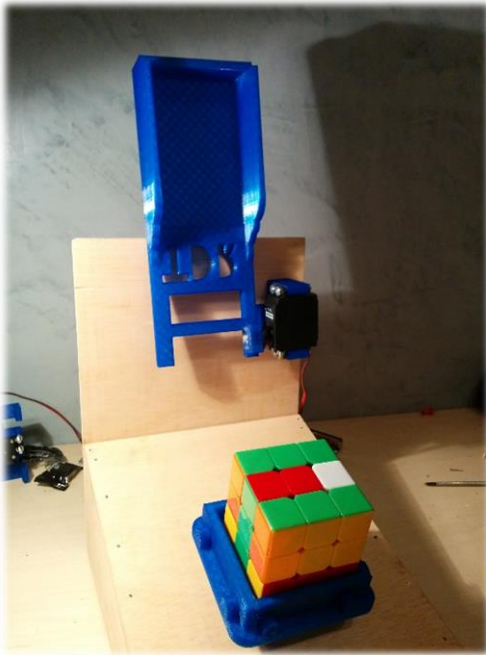


- Quadrar-hi la plataforma del cub.

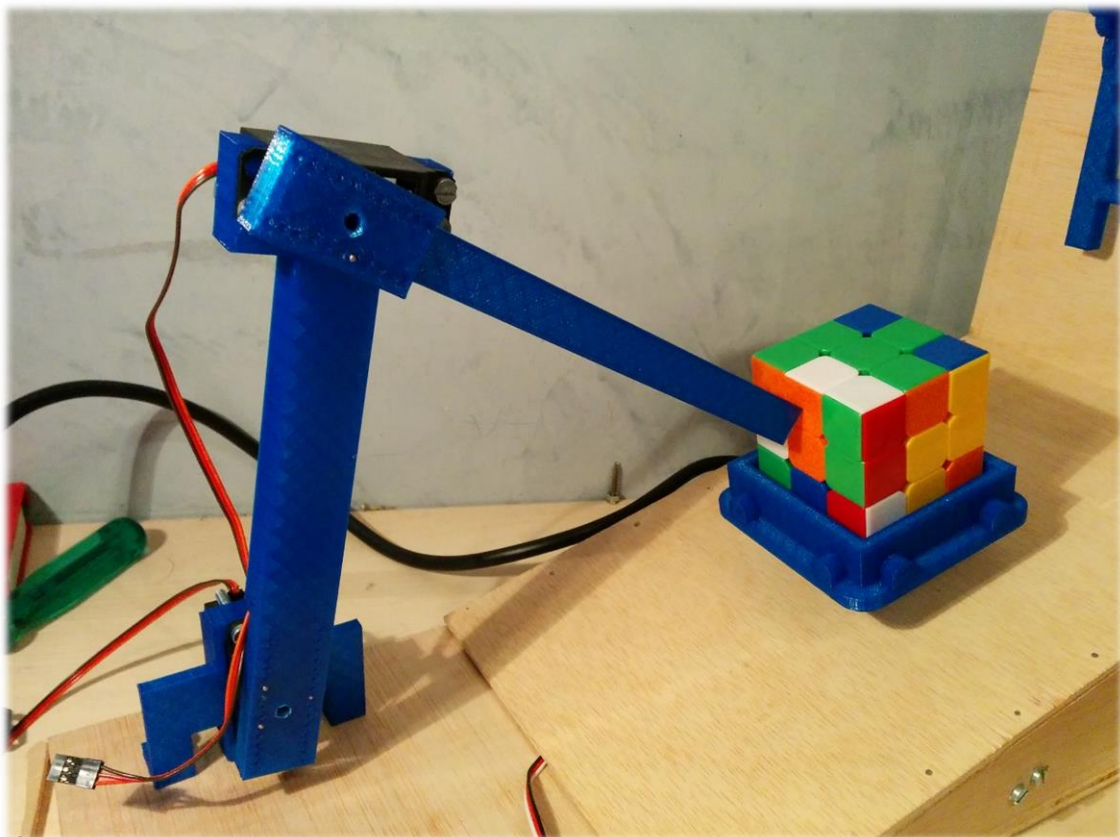


RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

- Muntar el fixador i el seu suport en la posició correcta, de manera que l'angle, quan faci el moviment, quedi paral·lel al cub.



- Per últim, fixar els dos braços i fer que quedi centrat



10. BLOC 3: ELECTRÒNICA I PROGRAMACIÓ – PLACA *PICAXE IMAGINA*

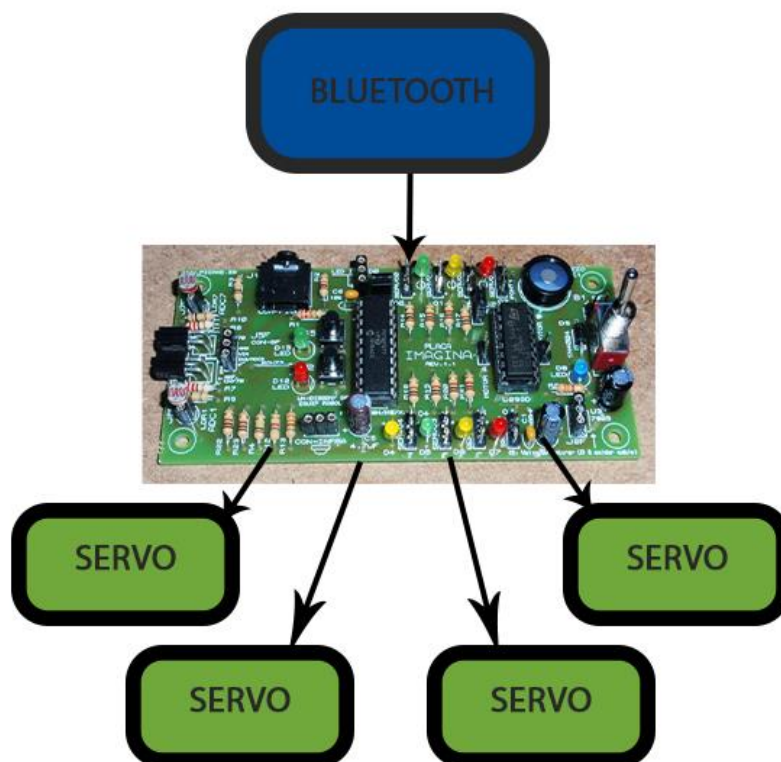
La tercera part del treball va consistir en l'electrònica i la programació de la placa. És el bloc del treball que vaig tenir més clar des de bon principi, però que, a mesura que anava avançant, se'm complicava més.

Per a la part electrònica, la meva intenció era usar 3 servomotors per a les "mans" del robot i 1 motor pas a pas per al "rotor", és a dir, l'encarregat de girar el cub perquè les mans puguin manipular-lo. Al final, vaig decidir usar només 4 servomotors per a tots els moviments tot i fer més lent el procés de resolució del cub.

A l'apartat de programació, al començament vaig decidir usar la placa *PICAXE IMAGINA* amb el xip *PICAXE 20M2*, ja que es programava amb llenguatge *Basic* i durant l'estiu havia après a programar amb aquest llenguatge.

10.1. Electrònica

La part electrònica del treball es pot esquematitzar amb el següent quadre:



Marcades en **BLAU** les entrades i en **VERD** les sortides a la placa.

A més d'això vaig fer servir dues fonts d'alimentació per alimentar el sistema, una va ser el *Bluetooth* i l'altra els servos.

10.1.1. Bluetooth

Bluetooth és una tecnologia sense fils que permet l'intercanvi de dades a través dels dispositius fixos i mòbils en distàncies curtes.



Necessitava una manera de comunicar-me amb la placa per ordenar-li quins moviments havia de fer. Vaig dubtar entre 3 maneres de comunicació: Posar un teclat físic i que cada tecla correspongués a una determinada instrucció; fer el mateix però amb un simple comandament de televisió per infraroig; o, per últim, usar un mòbil i la comunicació *Bluetooth*.



Em vaig decantar per la darrera opció per la comoditat de poder controlar el robot des del mòbil. Per aconseguir-ho necessitava un receptor de *Bluetooth* que pogués rebre les dades que li fossin enviades des del telèfon mòbil i vaig usar el receptor JY-MCU, que no tan sols funcionava com a receptor, sinó també com a emissor.

10.1.2. Servomotors

Els servos són un tipus especial de motor de corrent contínua que es caracteritzen per la seva capacitat per posicionar-se de forma immediata en qualsevol posició dins del seu interval d'operació.

Per això, el servomotor espera un tren de polsos que es correspon amb el moviment a realitzar. Estan generalment formats per un amplificador, un motor, un sistema reductor format per rodes dentades i un circuit de realimentació, tot en una mateixa caixa de petites dimensions. El resultat és un servo de posició amb un marge d'operació de 180° aproximadament.



Es diu que el servo és un dispositiu amb un eix de rendiment controlat, ja que pot ser portat a posicions angulars específiques en enviar

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

una senyal codificada. El servo mantindrà la posició angular de l'engranatge mentre no canviï la senyal. Quan la senyal codificada canvia, la posició angular dels pinyons canvia. Un servo normal proporciona un parell de $3 \text{ kg} \cdot \text{cm}$ a 4.8 V , la qual cosa és suficient per la seva grandària, sense consumir molta energia. El corrent que requereix depèn de la mida del servo. Normalment el fabricant indica quina és la corrent que consumeix.

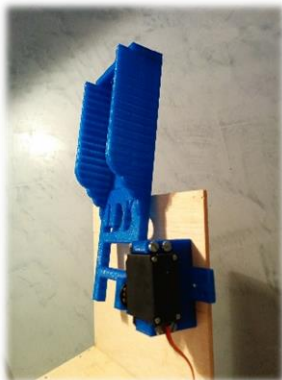
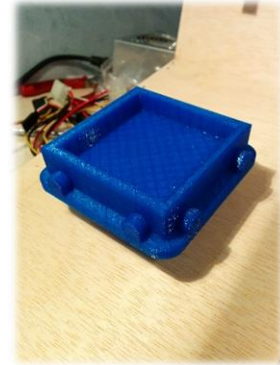
El servo té 3 cables de connexió:

- **VERMELL** - Per l'alimentació, Vcc (5V)
- **NEGRE** - Connexió a terra (GND) (0V)
- **BLANC** o **GROC** - és la línia de control per la qual se li envia la senyal codificada per comunicar l'angle en què s'ha posicionar.

La raó per la qual vaig usar servos pel robot és pel parell que proporcionen (la força que fan a part del moviment). Havia de fer servir uns motors que poguessin aixecar i moure el pes del cub sense problemes i sense importar l'angle des del qual fes el treball.

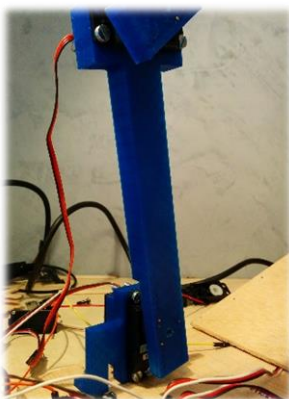
Les funcions dels 4 servos del treball són:

- **ROTOR**. S'encarrega dels moviments del cub paral·lels a la base (eix X).



- **FIXADOR**. S'encarrega de baixar la plataforma que ha de fixar el cub perquè el rotor pugui moure la cara inferior.

- **BRAÇ 2**. S'encarrega de fer els moviments perpendiculars a la base.



- **BRAÇ 1**. És simplement un ajudant del braç 2.

10.1.3. Font d'alimentació

En fer servir 4 servos i el *Bluetooth* actuant alhora, unes simples piles no eren suficients, ja que la intensitat que recorria el sistema era inestable. Per això, vaig optar per fer servir dues fonts d'alimentació:

- **Font d'alimentació per la placa i els servomotors** – Vaig reutilitzar una font d'alimentació d'un ordinador vell per alimentar la placa.
- **4 piles d'1,5V per alimentar el mòdul de *Bluetooth*.**

COM REUTILITZAR UNA FONT D'ALIMENTACIÓ D'ORDINADOR?

Per engegar una font d'alimentació sense estar connectada a un ordinador cal fer un pont entre el cable de senyal, normalment de color **VERD**, i el cable de connexió de terra, normalment **NEGRE**, de la manera com es mostra la imatge.

Després de comprovar que la font d'alimentació s'engega, cal identificar amb un tester quin és el voltatge de cada pin; en el cas del robot, vaig buscar quins eren els pins de +5V, que normalment són de color **VERMELL**.

Llavors només faltava connectar el cable vermell al pol positiu de la placa i qualsevol cable de terra de color negre al pol negatiu per aconseguir una connexió de +5V més estable que amb unes piles.



10.2. Programació de la placa

Per controlar tot el que ha de fer el robot s'ha de programar la placa d'una manera determinada que s'ajusti a les necessitats que es vulguin i transmetre-ho a la placa a través d'un cable SERIE.

10.2.1. *Picaxe Editor 6*

Per a la programació de la placa vaig usar el programa *PICAXE EDITOR 6*, la versió nova, encara en estat beta del programa original *PICAXE PROGRAMMING EDITOR*. Alguns avantatges de la nova versió són la millora de l'aparença del programa i la funció de veure l'evolució de les variables del xip en tot moment sense haver d'escriure cap instrucció.

10.2.2. Programar un robot capaç de moure i resoldre un cub de Rubik

He organitzat la programació de la següent manera:

- Primerament vaig posar els noms als valors fixos, és a dir, valors com la sortida en què es connecta el servo o el temps d'espera entre operació i operació, ja que d'aquesta manera el programa queda molt més clar. Per això, faig servir la instrucció *symbol*:

```

1 configuracions_inicials:
2 'Espera entre moviments
3 symbol espera = 1
4 'Espera entre final de moviment i retorn a posicio inicial del CUB
5 symbol espera2 = 3
6
7 'Configuracio variable de Bluetooth
8 symbol bluetooth=b5
9
10
11 'Configuracio de tots els moviments dels servos en variables
12 symbol rotor = 1
13     symbol rotor_0 = 45
14     symbol rotor_90 = 125
15     symbol rotor_180 = 215
16     symbol rotor_corrector_0=135
17     symbol rotor_corrector=rotor_corrector_0
18     symbol rotor_corrector_180=115
19
20 symbol brac1 = 5
21     symbol brac1_inici = 235
22     symbol brac1_mig = 220
23     symbol brac1_final=210
24
25 symbol brac2 = 4
26     symbol brac2_inici = 50
27     symbol brac2_mig=132
28     symbol brac2_final1= 131
29     symbol brac2_final2=138
30     symbol brac2_final3=145
31     symbol brac2_final=155
32
33
34 symbol fixador = 2
35     symbol fixador_inici = 243
36     symbol fixador_final = 140
37

```

- Seguidament, vaig programar que els servomotors es posicionessin a el lloc inicial amb la instrucció *servo*.
- Després vaig programar que cada moviment del cub correspongués a un número i que, si aquest número era rebut per *Bluetooth*, s'efectués el moviment amb la instrucció *serin* i configurant la freqüència de treball del xip a les necessitats del *Bluetooth* amb la instrucció *setfreq*.

```

;REBUDA DE DADES DE LA APP PER BLUETOOTH
inici:
setfreq m8
serin C.3,T9600_8,bluetooth

'Moviments horaris
if bluetooth=65 then L2
if bluetooth=66 then U
if bluetooth=67 then Bb
if bluetooth=68 then F
if bluetooth=69 then Dd
if bluetooth=70 then R

'Moviments antihoraris
if bluetooth=71 then L2
if bluetooth=72 then U2
if bluetooth=73 then bbb2
if bluetooth=74 then F2
if bluetooth=75 then D2
if bluetooth=76 then R2

goto inici

```

- Per acabar, hi ha tots els moviments possibles del cub organitzats en sub-rutines. Per moure el servo vaig fer servir la instrucció *servopos*.

```

U: 'Moviment part superior (UP) HORARI
setfreq m4
servopos rotor,rotor_90
wait espera
    wait espera
    servopos rotor,rotor_0
    wait espera
    gosub moviment_bracos
    wait espera
    gosub moviment_bracos
    wait espera
    servopos fixador,fixador_final
    wait espera
    servopos rotor,rotor_corrector
    wait espera
    servopos fixador, fixador_inici
    wait espera
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    wait espera
    servopos rotor,rotor_0
    wait espera
    gosub moviment_bracos
    wait espera
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    wait espera
goto inici
    
```

Exemple de programació per a moviment "U" del cub

- També vaig programar una sub-sub-rutina anomenada *moviment_bracos*, la seqüència de moviments que els braços havien de fer per girar el cub a la posició perpendicular a la base que ajudava a optimitzar moltíssim el programa (passava de 1500 línies de programa a unes 1000).

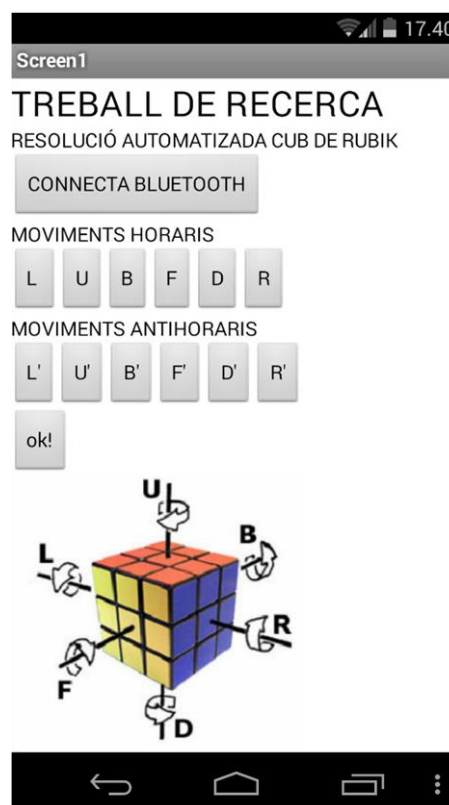
```

moviment_bracos:      'Moviment simple de gir de cub
setfreq m4
    servopos bracl,bracl_mig
    wait espera
    servopos brac2,brac2_mig
    wait espera2
    setfreq m4
    servopos bracl,bracl_final
    servopos brac2,brac2_final
    wait espera
    servopos brac2,180
    wait espera
    servopos bracl,bracl_inici
    wait espera
    servopos brac2,brac2_inici
    wait espera
return
    
```

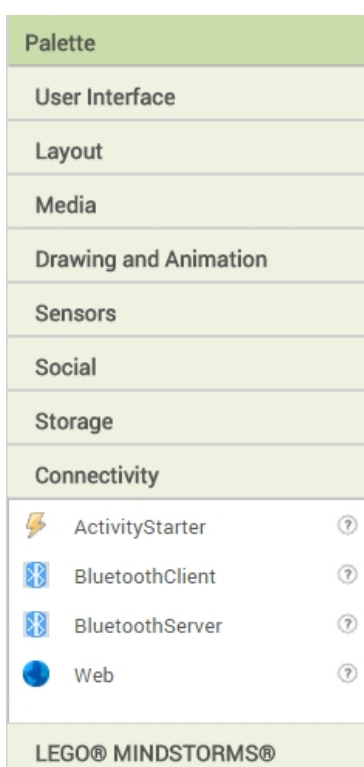
11. BLOC 4: CREACIÓ D'UNA APLICACIÓ PER A ANDROID – APP INVENTOR

En l'últim bloc del treball vaig crear una aplicació per a *Android* que pogués controlar els moviments del robot amb *AppInventor*. Per fer-ho primerament vaig fer el disseny i, una vegada acabat, la programació.

Per dissenyar l'aplicació em vaig basar en la senzillesa, és a dir, alguna cosa intuïtiva i fàcil d'usar. Aquest és el resultat:



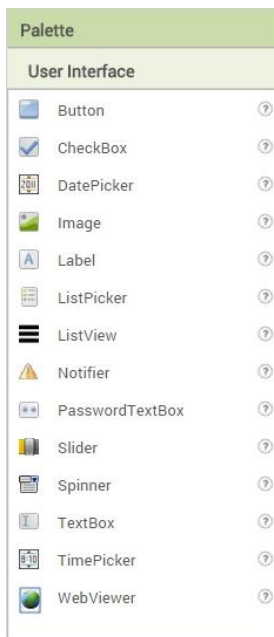
11.1. Disseny de l'aplicació



A la imatge de l'esquerra es mostren les opcions que et proporciona el programa per crear l'aplicació.

En el cas del robot, només usarem les opcions de "User Interface", "Layout" i "Connectivity".

11.1.1. User Interface

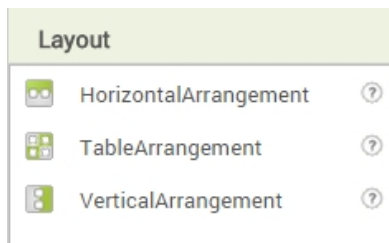


En aquest apartat vaig fer servir les següents funcions:

- *ListPicker* – Serveix per agregar un botó clicable a l'aplicació que et redirigeix a una llista d'elements. En el cas del robot, vaig posar un "ListPicker" per mostrar les connexions *Bluetooth* disponibles per enllaçar la placa amb el mòbil.
- *Image* – Serveix per inserir fotos. En el cas del robot, vaig col·locar una il·lustració del cub i els moviments.
- *Label* – Serveix per col·locar text. A diferència del "Text box", el text és fix i la persona que usa l'aplicació no el pot modificar.

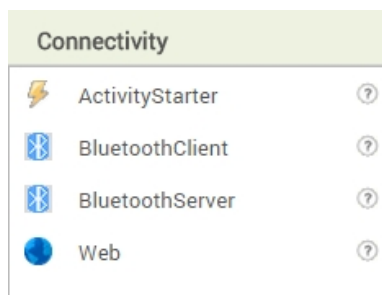
11.1.2. Layout

L'apartat de *layout* serveix per organitzar l'aplicació, és a dir, posar tot el que vulguis posar al seu lloc.



En el cas que es vulgui situar un element a la part dreta de la pantalla s'haurà de posar prèviament un *HorizontalArrangement* i posar-hi l'element a dins. I si es vol crear una taula o col·locar-hi elements verticals es necessitaran o bé el *TableArrangement* o el *VerticalArrangement*.

11.1.3. Connectivity



En aquest apartat hi ha totes les opcions de connexions del mòbil amb altres dispositius. Per al robot vaig fer servir el *BluetoothClient* per connectar la placa amb el mòbil. Les opcions d'aquest apartat no modifiquen gràficament el que es veurà en pantalla, només activen una funció determinada, en aquest cas, el *Bluetooth*.

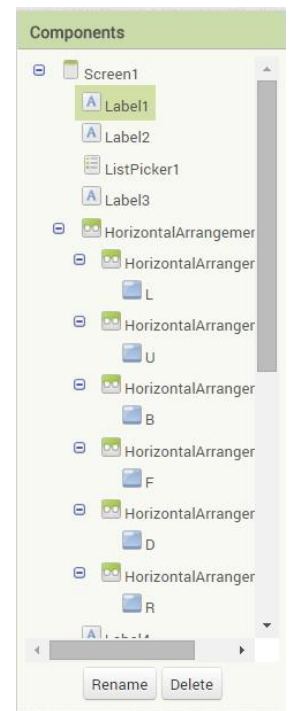
11.1.4. Propietats i recursos usats

Quan afegia un element a l'aplicació apareixia una taula de propietats que et permetia modificar qualsevol paràmetre com la mida, el tipus de lletra i color, estar o no estar visible en pantalla, entre d'altres.

A mesura que vaig afegir elements a l'aplicació s'anava creant una taula de components com a la imatge, on es poden veure tots els elements usats en l'aplicació del robot.



Taula de propietats



Taula de components

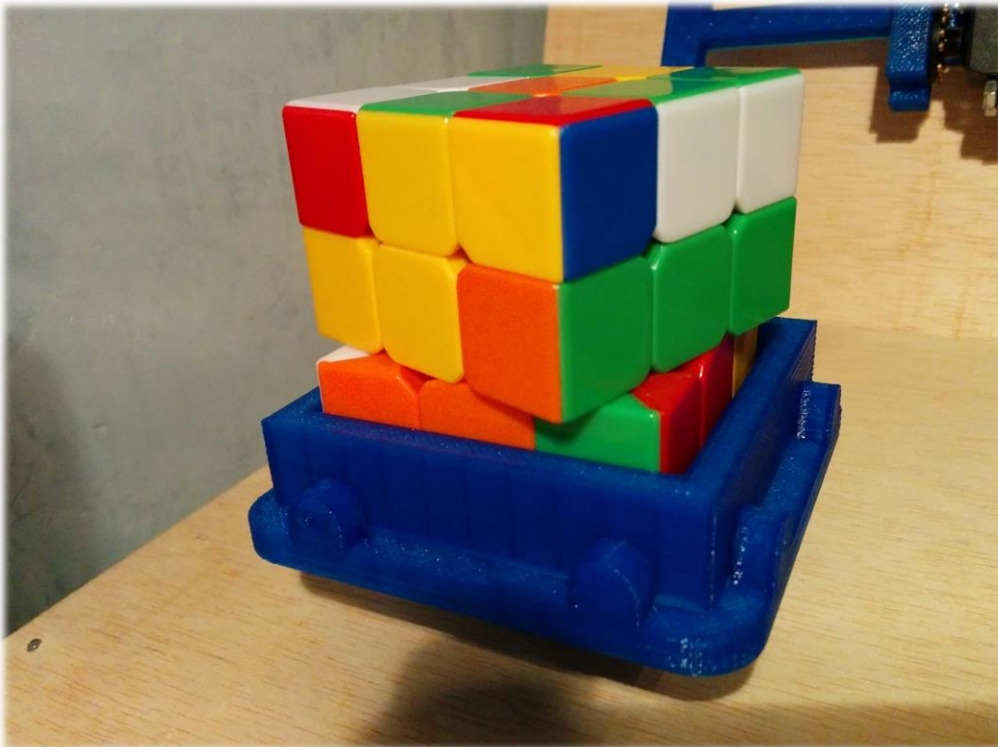
11.2. PROGRAMACIÓ DE L'APLICACIÓ – BLOCKS EDITOR

Després de decidir el disseny, vaig obrir el *Blocks Editor* per establir el comportament de cada element. És un mètode de programació per blocs (trossos de programació) per iniciats en la programació. Aquest és el resultat final:



12. PROBLEMES SORGITS DURANT EL PROCÉS

- A causa d'un problema de marges, quan vaig dissenyar els suports per als servomotors només vaig deixar 0,75 mm de marge entre la mida exacta i la mida del suport. Això va comportar que qualsevol mínim marge d'error de la impressora em fes anar malament el suport. És per això que per introduir els servomotors en els seus suports vaig haver de desmuntar-los, introduir-los desmuntats i tornar-los a collar.
- Una vegada tenia tot el robot muntat i programat, el fixador, la part que assegura el cub perquè el rotor pugui moure la cara inferior, era més ample del compte i el cub no quedava quadrat com es veu en la imatge.



La solució a aquest problema va ser afegir un factor corrector al rotor, és a dir, que el rotor girés més del compte fins a quadrar el cub i que, després, quan el fixador ja hagués tornat a la posició inicial, retornés a la posició correcta amb el cub quadrat.

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

- Com que el servomotor que controla el rotor ha de tenir un angle determinat, $19,27^\circ$ perquè el cub pugui fer els moviments de l'eix Y, el suport del servo també s'havia de collar amb aquest angle i en collar-ho em vaig adonar que el servomotor tocava a terra abans que l'estructura.



O bé havia de pujar el nivell de la fusta perquè el servo quedés a dintre, o bé havia de rebaixar la fusta mig centímetre. Vaig decidir rebaixar la fusta, ja que amb una fresadora no era complicat rebaixar mig centímetre suficient perquè el servo pogués cabre-hi.



- Quan compres un servomotor hi vénen uns encaixos com els de la imatge. Roscats amb un bis, els encaixos queden fixats a l'eix del servomotor.

Quan dissenyava les peces del robot les vaig fer pensant que es poguessin ajuntar amb aquests encaixos, però no vaig tenir en compte que el bis era petit i que si posava una peça davant l'encaix el bis no cargolaria. La solució va ser comprar bisos mes llargs i el problema es va solucionar. El problema gros va venir quan ja tenia tot el robot muntat i funcionant, vaig voler assegurar-me que els bisos estaven ben enroscats i els vaig enfortir tots els eixos. Quan



vaig engegar el robot vaig descobrir que un servomotor no feia el que havia de fer i que no responia als polsos de voltatge. Després de trencar-me el cap moltes

hores del que podia haver passat vaig desmuntar el servomotor i vaig veure que en roscar havia enfonsat i foradat el potenciòmetre que porten tots els servos, que és l'encarregat de saber la posició en què es troba el servo. Sense potenciòmetre el servomotor es converteix en un motor de corrent contínua, és a dir, gira sense límit. Vaig tenir la sort que fent palanca vaig aconseguir posar el potenciòmetre al seu lloc i que el forat no havia tocat cap component electrònic. Vaig tornar a

posar en marxa el sistema i va funcionar perfectament.

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

- Quan ja tenia tot el robot acabat vaig veure que, tot i alimentar el robot amb una font d'alimentació, quan tots els servomotors actuaven alhora, la intensitat baixava molt considerablement fins al punt de la desconexió del mòdul de *Bluetooth*. Si se'm desconnectava el mòdul, cada vegada que volia fer un moviment del cub havia de connectar-lo. La solució a aquest problema va ser alimentar el mòdul *Bluetooth* amb un porta piles de 6V, ja que el consum és mínim, i alimentar la placa i els motors amb la font d'alimentació.
- Com he dit abans, vaig dissenyar les peces pensant en els suports que vénen amb el servomotor quan en compres un. És per això que les peces del robot van ser dissenyades amb la mateixa mida que els forats dels suports del servomotor, és a dir, cargols de 1,6 mm de diàmetre. Vivia sense saber que les mides normalitzades que es troben en una ferreteria convencional comencen als 3 mm de diàmetre. Un cop amb les peces impreses, em vaig trobar amb forats de 1,4 mm de diàmetre a causa del marge d'error de la impressora, per ser tan petit, i sense la mínima idea d'on trobar una rosca més petita de 3 mm de diàmetre. Em va passar pel cap fer el forat més gran amb un trepant, però això era sinònim de canvi total del que havia planejat. Finalment, buscant per Internet vaig trobar en una botiga de modelistes, els qui fan maquetes a escala i allà vaig trobar rosques i volanderes de M1,4, és a dir, 1,4 mm de diàmetre.

El segon problema arran del primer va ser que, un cop vaig rebre els cargols, eren tan petits i amb tan poc radi que, en cargolar-los (ho havia de fer amb unes alicates perquè no es troben tornavisos tant petits), la majoria es trencaven.



RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

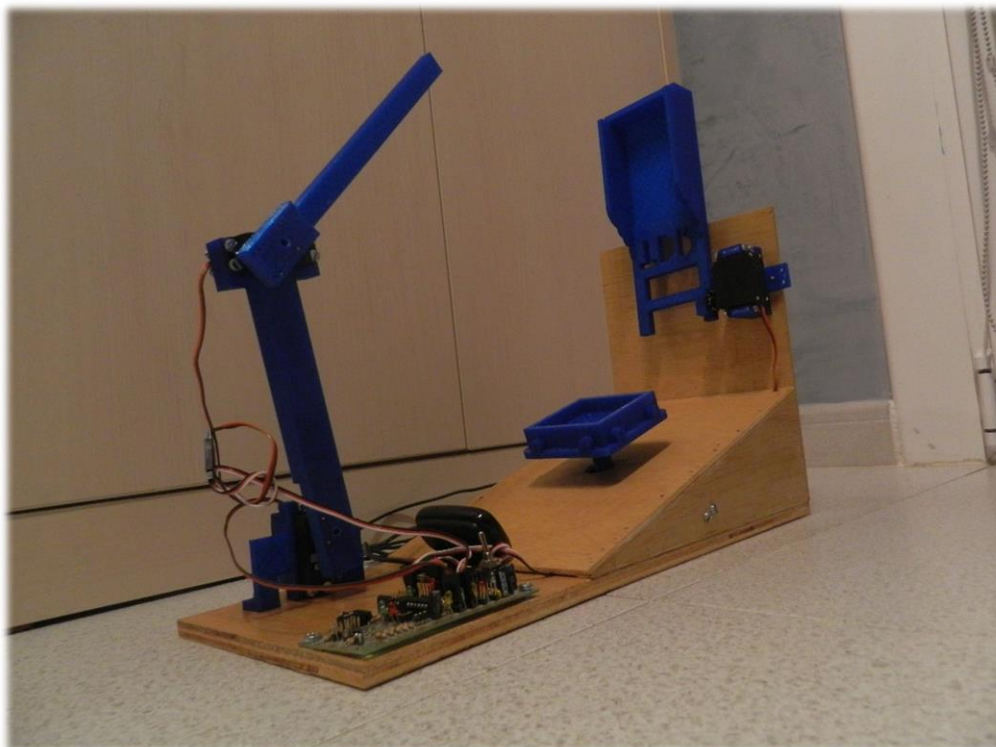
- Quan ja tenia tot el codi de programació estructurat em vaig disposar a enviar el codi a la placa per comprovar si tot funcionava. Em vaig trobar amb un error continu que m'impedia dur a terme aquesta tasca. Vaig arribar a la conclusió que el xip que usava, el Picaxe 20M2, s'havia quedat petit i no tenia suficient memòria ni variables per acollir tot el que volia fer. És per això que vaig haver d'anar a Vilafant, on hi ha el magatzem del distribuïdor de Picaxe de la península Ibèrica a buscar un xip més potent i amb més memòria com era el 20X2. Aquestes són les principals diferències dels xips:

	PINS	MEMÒRIA	SORTIDES	ENTRADES	RAM	VARIABLES
20M2	20	2048 Bytes	8	8	512 Bytes	28
20X2	20	4096 Bytes	8	8	256 Bytes	56

Tot i constar de menys memòria RAM, el 20X2 m'anava millor pel treball perquè necessitava 40 variables i el 20M2 tant sols en tenia 28.

13. ÚLTIMS DETALLS

Per acabar, vaig decidir envernissar el treball, ja que li donava un molt millor aspecte.



14. POSSIBLES MILLORES DEL TREBALL

El treball es podria millorar de la següent manera:

- Posar un motor pas a pas que controli el rotor en comptes d'un servomotor. Així, estalviariem molts moviments extrems que el servo ha de fer, ja que només pot moure's 180 graus. Amb el motor pas a pas aconseguiríem més precisió i evitaríem moviments innecessaris. No he implementat aquesta millora per un problema de tensions entre *Bluetooth*, placa, servos i motor pas a pas.
- Fer els braços que mouen l'eix Y del cub més consistents. En dissenyar en 3D no tens una visió 100% certa de les peces que estàs fent fins que no les agafes amb les mans. Amb unes peces amb més gruix aconseguiríem reduir tremolors i seguretat.
- Programar una aplicació que calculi les solucions del cub escanejant amb sensors de color. No ho he dut a terme perquè aquesta no era la meua intenció, ja que em volia centrar exclusivament en la part electrònica i crec que aquesta opció sola podria ser un treball de recerca diferent.

15. CONCLUSIÓ I OPINIÓ PERSONAL

La realització d'aquest treball m'ha servit moltíssim com a punt d'inflexió a la meua vida, ja que mai m'havia proposat un repte com aquest. Un projecte que va des de principi fins al final, des de la idea a la realitat passant pel concepte.

Han estat moltes i moltes hores davant l'ordinador capficant-me quan les coses no sortien i alegrant-me quan veia que alguna cosa anava bé. Tot i així, no dono per perdut ni un minut dels que he dedicat a aquest projecte, ja que no tan sols m'han servit per adquirir una bona base formativa, sinó que, a més a més, m'ho he passat molt bé fent-lo. Pots ser molt feliç quan gaudeixes de la feina que fas perquè t'agrada fer el que fas.

He complert tots els objectius, tant els principals com els secundaris que em vaig proposar a mig curs de l'any passat. El robot aconsegueix resoldre el cub sencer amb una mitjana de 9 minuts.

16. AGRAÏMENTS

Haver completat aquest treball ha estat possible gracies a unes persones que no dubtaré mai en agrair-los el seu suport. Per tant, vull agrair:

- A en Jordi Fanals, el meu tutor, que em va iniciar en el món del Picaxe i la programació durant el mes de juny, quan anava a l'institut just quan havíem acabat les classes de primer de batxillerat. A més a més de que em va obrir la porta a poder construir les peces del robot amb la impressora 3D.
- A els meus pares i el meu germà, que van haver de suportar hores, dies i setmanes de maldecaps quan les coses no em funcionaven a la primera. Sense el seu suport moral tampoc hagués estat possible.
- A la meva cosina, Bruna, pel seu assessorament lingüístic a la memòria del treball.
- I finalment, a tu, lector, per interessar-te pel meu treball!

17. BIBLIOGRAFIA

<https://sites.google.com/site/bluetoothpicaxe/home/android---picaxe>

<http://humanoideimagina.weebly.com/>

<http://picaxe.es/>

<http://todopic.com/>

<http://rubikaz.com/>

<http://rubiks.com/>

<http://kociemba.org/cube.htm>

18. ANNEX

Per la comoditat de tots a la hora de veure el resultat, he creat un canal al YouTube amb vídeos del funcionament del robot. Els podeu trobar a el següent enllaç:

<http://www.youtube.es/user/roger21gm/videos>

CODI DE PROGRAMACIÓ DEL ROBOT:

```

#picaxe 20x2

configuracions_inicials:
'Espera entre moviments
symbol espera = 1
'Espera entre final de
moviment i retorn a posicio
inicial del CUB
symbol espera2 = 2
'Espera entre moviments de
bracos en milisegons
symbol espera3=375
'Espera moviment de bracos 2
en ms
symbol espera4=1100

'Configuracio variable de
Bluetooth
symbol bluetooth=b55

'Configuracio de tots els
moviments dels servos en
variables
symbol rotor = 3
    symbol rotor_0 = 45
    symbol rotor_90 = 125
    symbol rotor_180 = 215
    symbol
rotor_corrector_0=138
    symbol
rotor_corrector=rotor_correct
or_0
    symbol
rotor_corrector_180=106

symbol bracl = 4
    symbol bracl_inici = 240
    symbol bracl_mig = 221
    symbol bracl_final=209

symbol brac2 = 5
    symbol brac2_inici = 180
    symbol brac2_mig=134
    symbol brac2_migbaix=105
    symbol brac2_final=155

symbol fixador = 1
    symbol fixador_inici =
243
    symbol fixador_final =
120
;POSADA EN MARXA DELS SERVOS

posicionament:
servo rotor,rotor_90
wait espera
servo fixador,fixador_inici
servo bracl,bracl_inici
servo brac2,brac2_inici
wait espera

;REBUDA DE DADES DE LA APP
PER BLUETOOTH I
EMMAGATZEMADES EN VARIABLES

Pos:
setfreq m8
serin C.3,t9600_8,b0
if b0=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b1
if b1=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b2
if b2=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b3
if b3=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b4
if b4=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b5
if b5=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b6
if b6=109 then seq
gosub comprovant_pos

```

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

```
setfreq m8
serin C.3,t9600_8,b7
if b7=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b8
if b8=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b9
if b9=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b10
if b10=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b11
if b11=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b12
if b12=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b13
if b13=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b14
if b14=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b15
if b15=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b16
if b16=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b17
if b17=109 then seq
gosub comprovant_pos
```

```
setfreq m8
```

```
serin C.3,t9600_8,b18
if b18=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b19
if b19=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b20
if b20=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b21
if b21=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b22
if b22=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b23
if b23=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b24
if b24=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b25
if b25=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b26
if b26=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b27
if b27=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b28
if b28=109 then seq
gosub comprovant_pos
```

```
setfreq m8
serin C.3,t9600_8,b29
```

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

```
if b29=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b30
if b30=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b31
if b31=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b32
if b32=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b33
if b33=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b34
if b34=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b35
if b35=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b36
if b36=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b37
if b37=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b38
if b38=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b39
if b39=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b40
if b40=109 then seq

gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b41
if b41=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b42
if b42=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b43
if b43=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b44
if b44=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b45
if b45=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b46
if b46=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b47
if b47=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b48
if b48=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b49
if b49=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b50
if b50=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b51
if b51=109 then seq
gosub comprovant_pos
```

```

setfreq m8
serin C.3,t9600_8,b52
if b52=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b53
if b53=109 then seq
gosub comprovant_pos

setfreq m8
serin C.3,t9600_8,b54
if b54=109 then seq
gosub comprovant_pos

stop

;INICI DE LA SEQUENCIA:
Variable BLUETOOTH serà igual
a totes les variables
trobades.
seq:
bluetooth = b0
gosub bluetooth_seq

bluetooth = b1
gosub bluetooth_seq

bluetooth = b2
gosub bluetooth_seq

bluetooth = b3
gosub bluetooth_seq

bluetooth = b4
gosub bluetooth_seq

bluetooth = b5
gosub bluetooth_seq

bluetooth = b6
gosub bluetooth_seq

bluetooth = b7
gosub bluetooth_seq

bluetooth = b8
gosub bluetooth_seq

bluetooth = b9
gosub bluetooth_seq

bluetooth = b10
gosub bluetooth_seq

bluetooth = b11
gosub bluetooth_seq

bluetooth = b12
gosub bluetooth_seq

bluetooth = b13
gosub bluetooth_seq

bluetooth = b14
gosub bluetooth_seq

bluetooth = b15
gosub bluetooth_seq

bluetooth = b16
gosub bluetooth_seq

bluetooth = b17
gosub bluetooth_seq

bluetooth = b18
gosub bluetooth_seq

bluetooth = b19
gosub bluetooth_seq

bluetooth = b20
gosub bluetooth_seq

bluetooth = b21
gosub bluetooth_seq

bluetooth = b22
gosub bluetooth_seq

bluetooth = b23
gosub bluetooth_seq

bluetooth = b24
gosub bluetooth_seq

bluetooth = b25
gosub bluetooth_seq

bluetooth = b26
gosub bluetooth_seq

bluetooth = b27
gosub bluetooth_seq

bluetooth = b28
gosub bluetooth_seq

bluetooth = b29
gosub bluetooth_seq

```

```

bluetooth = b30
gosub bluetooth_seq

bluetooth = b31
gosub bluetooth_seq

bluetooth = b32
gosub bluetooth_seq

bluetooth = b33
gosub bluetooth_seq

bluetooth = b34
gosub bluetooth_seq

bluetooth = b35
gosub bluetooth_seq

bluetooth = b36
gosub bluetooth_seq

bluetooth = b37
gosub bluetooth_seq

bluetooth = b38
gosub bluetooth_seq

bluetooth = b39
gosub bluetooth_seq

bluetooth = b40
gosub bluetooth_seq

bluetooth = b41
gosub bluetooth_seq

bluetooth = b42
gosub bluetooth_seq

bluetooth = b43
gosub bluetooth_seq

bluetooth = b44
gosub bluetooth_seq

bluetooth = b45
gosub bluetooth_seq

bluetooth = b46
gosub bluetooth_seq

bluetooth = b47
gosub bluetooth_seq

bluetooth = b48
gosub bluetooth_seq

bluetooth = b49
gosub bluetooth_seq

bluetooth = b50
gosub bluetooth_seq

bluetooth = b51
gosub bluetooth_seq

bluetooth = b52
gosub bluetooth_seq

bluetooth = b53
gosub bluetooth_seq

bluetooth = b54
gosub bluetooth_seq
goto final

final:
servopos rotor, rotor_0
wait espera
servopos rotor, rotor_180
stop

comprovant_pos:
setfreq m8
high b.7
pause 350
low b.7
return

bluetooth_seq:
setfreq m8
'Moviments per escanejar
el CUB
if bluetooth=109 then
final

'Moviments horaris
if bluetooth=97 then L
if bluetooth=98 then U
if bluetooth=99 then Bb
if bluetooth=100 then F
if bluetooth=101 then Dd
if bluetooth=102 then R

'Moviments antihoraris
if bluetooth=103 then L2

```

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

```
    if bluetooth=104 then U2
    if bluetooth=105 then
bbb2
    if bluetooth=106 then F2
    if bluetooth=107 then D2
    if bluetooth=108 then R2

goto bluetooth_seq

'Programa per escanejar el
CUb amb l'ordre del
CubeExplorer.
escaneig:
bluetooth=0
setfreq m8
    wait espera
    servopos rotor,rotor_180
    gosub espera_escaneig
    servopos rotor,rotor_90
    gosub espera_escaneig
    servopos rotor,rotor_0
    gosub espera_escaneig
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    gosub moviment_bracos
    servopos rotor,
rotor_180
    wait espera
    gosub moviment_bracos
    gosub moviment_bracos
    servopos rotor,rotor_90
    gosub espera_escaneig
    gosub moviment_bracos
    gosub moviment_bracos
    gosub moviment_bracos
    gosub moviment_bracos
    gosub espera_escaneig
    gosub moviment_bracos
    gosub moviment_bracos
let pinB.2=1
    gosub espera_escaneig
    servopos rotor,rotor_180
    wait espera
    gosub moviment_bracos
    servopos rotor,rotor_90
    gosub moviment_bracos
    gosub moviment_bracos
    gosub moviment_bracos

goto pos

L:  'Moviment part esquerra
    (LEFT) HORARI
    bluetooth=0
    setfreq m8
    servopos rotor,rotor_90
    wait espera
        servopos rotor,rotor_0
        wait espera
        gosub moviment_bracos
        servopos
    fixador,fixador_final
        wait espera
        servopos
    rotor,rotor_corrector
        wait espera
        servopos
    fixador,fixador_inici
        wait espera
        servopos rotor,rotor_90
        wait espera
        servopos rotor,rotor_180
        wait espera
        gosub moviment_bracos
        servopos rotor,rotor_90
        wait espera
        gosub moviment_bracos
        gosub moviment_bracos
        gosub moviment_bracos
        wait espera

return

U: 'Moviment part superior
    (UP) HORARI
    bluetooth=0
    setfreq m8
    servopos rotor,rotor_90
    wait espera
        servopos rotor,rotor_0
        wait espera
        gosub moviment_bracos
        gosub moviment_bracos
        servopos
    fixador,fixador_final
        wait espera
        servopos
    rotor,rotor_corrector
        wait espera
        servopos fixador,
    fixador_inici
        wait espera
        servopos rotor,rotor_90
        wait espera
        gosub moviment_bracos
```


RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

```
servopos rotor,rotor_0
wait espera
gosub moviment_bracos
servopos rotor,rotor_90
wait espera
gosub moviment_bracos
wait espera

return

Bb: 'Moviment part posterior
(BACK) HORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
    gosub moviment_bracos
    servopos rotor,rotor_0
    wait espera
    servopos
fixador,fixador_final
    wait espera
    servopos
rotor,rotor_corrector
    wait espera
    servopos fixador,
fixador_inici
    wait espera
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    servopos rotor,rotor_180
    wait espera
    gosub moviment_bracos
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    gosub moviment_bracos
    wait espera

return

F: 'Moviment part frontal
(FRONT) HORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
    gosub moviment_bracos
    gosub moviment_bracos
    gosub moviment_bracos
    servopos rotor, rotor_0

    wait espera
    servopos fixador,
fixador_final
    wait espera
    servopos
rotor,rotor_corrector
    wait espera
    servopos fixador,
fixador_inici
    wait espera
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    wait espera
    servopos rotor,
rotor_180
    wait espera
    gosub moviment_bracos
    wait espera
    servopos rotor,rotor_90
    wait espera

return

Dd: 'Moviment part inferior
(DOWN) HORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
    servopos rotor,rotor_0
    wait espera
    servopos
fixador,fixador_final
    wait espera
    servopos
rotor,rotor_corrector
    wait espera
    servopos
fixador,fixador_inici
    wait espera
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    servopos rotor,rotor_180
    wait espera
    gosub moviment_bracos
    servopos rotor,rotor_90
    wait espera
    gosub moviment_bracos
    gosub moviment_bracos
    wait espera

return
```

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

```
R: 'Moviment part dreta del
cub (RIGHT) HORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
servopos rotor,rotor_0
wait espera
gosub moviment_bracos
gosub moviment_bracos
gosub moviment_bracos
servopos
fixador,fixador_final
wait espera
servopos
rotor,rotor_corrector
wait espera
servopos
fixador,fixador_inici
wait espera
servopos rotor,rotor_90
wait espera
servopos rotor,rotor_0
wait espera
gosub moviment_bracos
wait espera
servopos rotor,rotor_90
wait espera
gosub moviment_bracos
wait espera

return

'SENTIT ANTIHORARI!

L2: 'Moviment part esquerra
(LEFT) ANTIHORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
servopos rotor,rotor_180
wait espera
gosub moviment_bracos
gosub moviment_bracos
gosub moviment_bracos
servopos
fixador,fixador_final
wait espera
```

```
servopos
rotor,rotor_corrector_180
wait espera
servopos
fixador,fixador_inici
wait espera
servopos rotor,rotor_90
wait espera
servopos rotor,rotor_180
wait espera
gosub moviment_bracos
servopos rotor,rotor_90
wait espera
gosub moviment_bracos
wait espera

return
```

```
U2: 'Moviment part superior
(UP) ANTIHORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
servopos rotor,rotor_180
wait espera
gosub moviment_bracos
gosub moviment_bracos
servopos
fixador,fixador_final
wait espera
servopos
rotor,rotor_corrector_180
wait espera
servopos fixador,
fixador_inici
wait espera
servopos rotor,rotor_90
wait espera
gosub moviment_bracos
servopos rotor,rotor_180
wait espera
gosub moviment_bracos
servopos rotor,rotor_90
wait espera
gosub moviment_bracos
wait espera

return
```

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

```
Bbb2: 'Moviment part posterior (BACK) ANTIHORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
  gosub moviment_bracos
  servopos rotor,rotor_180
  wait espera
  servopos
fixador,fixador_final
  wait espera
  servopos
rotor,rotor_corrector_180
  wait espera
  servopos fixador,
fixador_inici
  wait espera
  servopos rotor,rotor_90
  wait espera
  gosub moviment_bracos
  servopos rotor,rotor_0
  wait espera
  gosub moviment_bracos
  servopos rotor,rotor_90
  wait espera
  gosub moviment_bracos
  gosub moviment_bracos
  wait espera

return

F2: 'Moviment part frontal (FRONT) ANTIHORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
  gosub moviment_bracos
  gosub moviment_bracos
  gosub moviment_bracos
  servopos rotor,
rotor_180
  wait espera
  servopos fixador,
fixador_final
  wait espera
  servopos
rotor,rotor_corrector_180
  wait espera
  servopos fixador,
fixador_inici
  wait espera

servopos rotor, rotor_90
wait espera
gosub moviment_bracos
servopos rotor, rotor_0
wait espera
gosub moviment_bracos
servopos rotor,rotor_90
wait espera

return

D2: 'Moviment part inferior (DOWN) ANTIHORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
  servopos rotor,rotor_180
  wait espera
  servopos
fixador,fixador_final
  wait espera
  servopos
rotor,rotor_corrector_180
  wait espera
  servopos
fixador,fixador_inici
  wait espera
  servopos rotor,rotor_90
  wait espera
  gosub moviment_bracos
  servopos rotor,rotor_0
  wait espera
  gosub moviment_bracos
  servopos rotor,rotor_90
  wait espera
  gosub moviment_bracos
  gosub moviment_bracos
  wait espera

return

R2: 'Moviment part dreta del cub (RIGHT) ANTIHORARI
bluetooth=0
setfreq m8
servopos rotor,rotor_90
wait espera
  servopos rotor,rotor_180
  wait espera
  gosub moviment_bracos
  servopos
fixador,fixador_final
```

RESOLUCIÓ ROBOTITZADA DEL CUB DE RUBIK

```
wait espera
servopos
rotor,rotor_corrector_180
wait espera
servopos
fixador,fixador_inici
wait espera
servopos rotor,rotor_90
wait espera
servopos rotor,rotor_0
wait espera
gosub moviment_bracos
servopos rotor,rotor_90
wait espera
gosub moviment_bracos
gosub moviment_bracos
gosub moviment_bracos
wait espera

return

moviment_bracos:
  'Moviment simple de gir
de cub
setfreq m8
servopos
bracl,bracl_inici
if pinc.5=1 then goto
espera_escaneig

pause espera3
servopos
brac2,brac2_migbaix
pause espera3
servopos bracl,bracl_mig
pause espera3
servopos brac2,brac2_mig
pause espera4

servopos
bracl,bracl_final
servopos
brac2,brac2_final
pause 600
servopos
brac2,brac2_inici
servopos
bracl,bracl_inici
pause espera3

return

espera_escaneig:
pause 200
if pinC.5=0 then return
endif
```