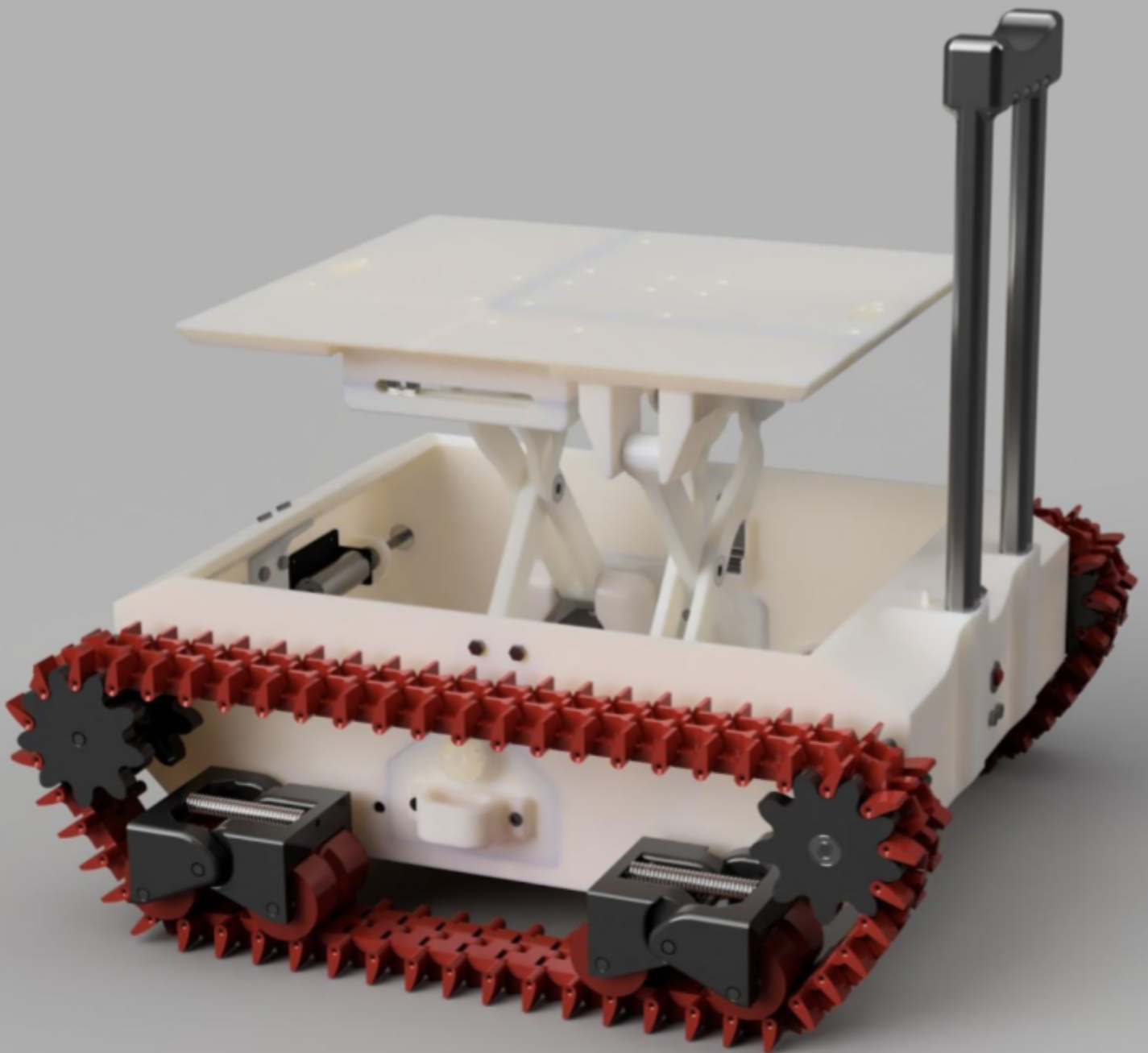


RC SELF-LEVELING CONVEYOR



Mod. MPC 4.F

Marc Puig i Cañellas
Tutora: Mònica Mascort
Institut Vidreres

ÍNDEX

INTRODUCCIÓ.....	Pàg.2
-Objectiu.....	Pàg.2
-Motivació.....	Pàg.2
1. ARDUINO.....	Pàg.3-5
1.1 Què és Arduino.....	Pàg.3
1.2 Història d'Arduino.....	Pàg.3
1.3 Tipus de plaques Arduino.....	Pàg.4
1.4 Per què he triat l'Arduino Mega 2560.....	Pàg.4
1.5 Parts de l'Arduino Mega 2560.....	Pàg.5
2. PRODUCTES SIMILARS.....	Pàg.6
3. DISSENY.....	Pàg.7-12
4. ELECTRÒNICA.....	Pàg.13-14
5. PARTS I COST.....	Pàg.15
6. TRACTAMENTS POST IMPRESSIÓ I CONSTRUCCIÓ.....	Pàg.16-18
- Tractaments postimpresió.....	Pàg.16
- Construcció.....	Pàg.17-18
7. PROGRAMACIÓ.....	Pàg.19-26
- Programació de la tracció.....	Pàg.19-22
- Programació de la anivellació.....	Pàg.23-26
CONCLUSIONS.....	Pàg.27
AGRAÏMENTS.....	Pàg.28
INFOGRAFIA.....	Pàg.29

INTRODUCCIÓ

Objectiu:

L'objectiu d'aquest projecte és produir una maqueta d'un robot que tindrà una tracció de roda d'eruga i una plataforma autoanivellable, tot ell serà controlat per un comandament a radiocontrol que enviarà el senyal al microcontrolador Arduino mega. L'objectiu del mateix és transportar objectes sent capaç de mantenir el producte anivellat mentre el robot es desplaça per pendents. També serà capaç d'eleva la plataforma per facilitar la deposició dels productes al treballador.

Motivació:

La idea d'aquest projecte se'm va ocórrer quan vaig veure el meu tiet que és repartidor de la Frit Ravich baixant de la furgoneta una capsa de llaunes de coca cola. Vaig veure que patia, ja que hi ha capses que poden arribar a pesar molt. Llavors, vaig pensar: Com podria ajudar-lo? Aprofitant que aquest curs havia de fer el treball de recerca vaig pensar en un robot que el pogués ajudar a l'hora de repartir els seus productes estalviant-se així futures lesions cervicals, lumbar i de genolls. Com vaig veure que fer un robot que realment pogués portar tant de pes seria molt més car que no fent una maqueta, vaig decidir fer una maqueta.

1. ARDUINO

1.1 Què és Arduino:

Arduino és una plataforma de codi obert pensada per un ús simple i fàcil d'entendre. Arduino és tant programari com maquinari. Com a programari utilitza una variació del llenguatge C++ el qual és una versió del llenguatge C amb noves modificacions. La versió d'Arduino és diferent del C++ ja que aquesta està pensada per poder controlar les seves pròpies plaques. Aquestes plaques es programen a mitjançant el programa Arduino IDE el qual es basa en el processing.

1.2 Història d'Arduino:

Arduino va ser creada l'any 2005 a l'institut de IVRAE Massimo Banzi. En un principi l'Arduino va ser creada perquè els estudiants aprenguessin electrònica i programació d'una manera més barata, ja que l'institut no es podia permetre comprar altres microcontroladors. El primer prototip d'Arduino no era gens semblant als que tenim ara, ja que en aquell moment només constava d'un microcontrolador i unes resistències de voltatges que permetien connectar Leds i altres resistències sense cap mena de programació. No va ser fins uns anys més tard que es va crear el primer programari per aquella placa: el Wiring. El qual va ser creat per Hernando Barragán, un estudiant de la Universidad de Colombia el qual havia de fer la seva tesi que va ser ajudat per en David Mellis. Més tard es va afegir al grup l'expert en circuits i ordinadors David Cuartielles el qual va millorar la placa, va millorar la plataforma i el llenguatge de programació. Després Massimo Banzi va convidar a Tom Igoe a participar en el projecte i ell va millorar el rendiment de la placa afegint-hi ports USB per poder-la connectar als ordinadors. A més, ell va suggerir la distribució mundial del projecte. Quan van pensar que Arduino estava preparada van contactar amb el publicista Gianluca Martino, qui va donar a conèixer el producte. Per últim va produir en massa la placa gràcies a Natan Sadle. Finalment és reconeguda com una de les millors plataformes per fer els projectes DIY (Do it yourself) que en català vol dir: Fes-ho tu mateix.

1.3 Tipus de plaques Arduino:

Avui en dia hi ha molts tipus de plaques, ja que en ser un projecte open source, tothom pot fer modificacions a la placa i comercialitzar-la, però n'hi ha unes quantes que s'han normalitzat i que són les que més s'utilitzen:

Arduino UNO R3, Arduino Leonardo, Arduino Due, Arduino Ethernet, Arduino Mega, Arduino Mini, Arduino Nano, Arduino Micro, LilyPad Arduino USB, etc.

N'hi ha moltes més i cada una amb la seva utilitat i diferència. Normalment es comença amb una Arduino UNO R3 o una Leonardo, ja que són una versió simple i sense res especial. A partir d'aquí es poden tenir diferents necessitats. Si es necessita una placa que càpiga en un lloc petit potser el millor són les plaques: Nano, Mini, Pro mini o la micro. Si es necessita una placa amb un bon processador i molts pins la millor opció potser la mega ja que té fins a 53 pins digitals. Si es necessita una placa amb connexió a internet, pots utilitzar l'Arduino Ethernet. Si es necessita que sigui una placa circular existeix la placa LilyPad Arduino amb totes les seves modificacions.

1.4 Per què he triat l'Arduino Mega 2560:

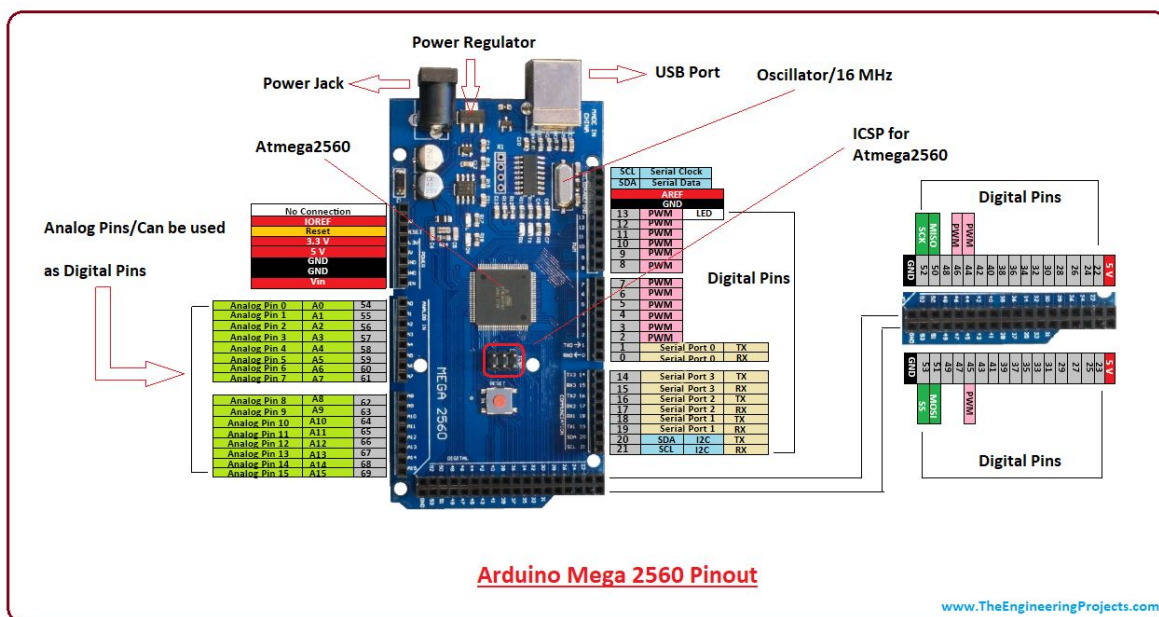
En el meu cas he utilitzat dues Arduino mega 2560 per una simple raó. Les he reciclat d'un antic projecte el qual necessitava molts pins. L'espai que ocupen no és un inconvenient, ja que el tanc té prou espai. La raó per la qual utilitzo dues plaques en lloc de només una és perquè utilitzaré una per controlar la tracció del robot i l'altre per controlar l'acceleròmetre que controla la plataforma anivellable. D'aquesta manera és més fàcil programar les funcions per individual i no estressaré tant les plaques.

1.5 Parts de l'Arduino Mega 2560:

L'Arduino Mega2560 conté 53 pins digitals, 15 d'ells amb PWM (Modulació per amplada de polsos), 5 dels digitals de TX (transmissió de senyal) i 5 de RX (rebuda) i 2 de SDA (Serial data line) i 2 SCL (Serial clock line). També té 15 pins analògics.

També comte amb 3 sortides de 5V, 1 de 3,3V, 1 de Vin (Voltage in) i 5 GND(Ground).

El seu nucli és un microcontrolador Atmega2560 amb l'ICSP per l'Atmega2560. També té un oscil·lador de 16 MHz, un controlador de corrent, una entrada de corrent amb port jack de fins a 9V i un port USB per informació.



Im.1 Parts de l'Arduino

2.PRODUCTES SIMILARS:

En el mercat hi ha diversos productes similars els quals tenen la mateixa funció que el meu robot, però tots tenen algun inconvenient. Alguns d'aquests productes són:



Im.2 Producte similar 1

<https://www.pianoplan.info/es/>

Aquest és el producte més complet que hi ha al mercat, puja les escales sol, s'aixeca sol, es mou amb un control manual motoritzat i mou la plataforma amb pistons hidràulics com la majoria de màquines d'aquest tipus.

La part negativa d'aquest producte és que s'ha de moure manualment i no amb un comandament radiocontrol.



Im.3 Producte similar 2

Aquest és un concepte diferent de les màquines enfilea escales, ja que aquest treballa verticalment i no horitzontalment com altres.



Im.4 Producte similar 3

Aquest és un altre exemple de robot que puja escales, a diferència del primer, aquest pot aguantar més càrrega i està controlat a distància tot i que amb un cable que podria ser trepitjat pel mateix robot. Una altra desavantatge és que necessita ser ajudat per un pla inclinat per

poder pujar els primers esglaons, ja que no té prou angle d'atac.

<https://www.materialshandling.com.au/products/stair-robot/>

3.DISSENY

Per dissenyar aquest robot, he utilitzat el programa de disseny Fusion 360, un programa d'Autodesk el qual compta amb tot el que se li pot demanar a un programa de disseny 3D. Té una interfície molt intuïtiva amb molts tutorials per aprendre. Pot renderitzar imatges per fer-les semblar reals, simular esforços, disseny generatiu (Ell mateix crea el disseny a partir d'unes ordres, restriccions i forces que ha d'aguantar), crear animacions, etc.

He escollit aquest programa de disseny per tot el que es pot fer amb ell i perquè tinc una llicència gratuïta de 3 anys per ser estudiant.

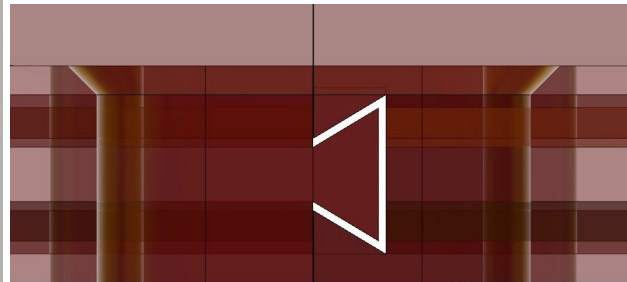
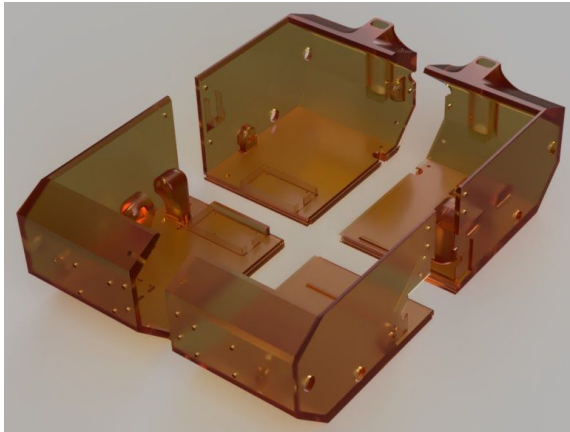
Amb el Fusion 360 he utilitzat les funcions de disseny, renderitzat i animació. Sobretot he utilitzat la funció de disseny, ja que és la que realment és necessària per crear el model.

El funcionament del programa és relativament senzill. Primer es crea un esbós en dues dimensions on es poden posar rectes, corbes, circumferències, patrons, arrodoniments i tota classe de restriccions, després, s'extrudeix en disseny donant volum a l'esbós i a partir d'aquest cos de 3 dimensions pots dibuixar a qualsevol cara i seguir modelant el cos. Un cop es té una pesa dissenyada, es pot crear un nou disseny on es pot ajuntar peses i d'aquesta manera veure com queda tot en conjunt i el seu funcionament, ja que aquest programa permet afegir moviments a les peces per poder veure tots els problemes que puguin haver-hi.

En el disseny del robot hi ha 8 parts principals:

El cos: Aquest és el cos del robot, és l'estructura principal on tots els components s'uneixen. Aquesta part està dividida en 4 parts, ja que s'ha de poder imprimir amb la impressora 3D Cr-10S la qual té una superfície d'impressió de 300x300 mm. Cada una d'aquestes peces fa al voltant de 300*203 sense contar els encaixos que afegeixen 5 mm d'allargada i amplitud en dues de les parts. Això fa que les parts siguin més grans que la plataforma, però en ser només 5mm es pot desplaçar el llit d'impressió 5 mm endavant i la impressora funciona perfectament, ja que no té cap restricció en el seu programari. Cada una d'aquestes peces està encaixada amb guies. Les parets, com són més difícils d'encaixar, tenen uns forats per poder-los ajuntar com si fos una reblada per tapajunts però amb visos. En l'interior del cos, hi

ha les 2 plaques Arduino, 3 l289N, 1 receptor, 2 motors dc, 2 motors pas a pas, 1 interruptor i el mànec telescòpic. A la part exterior hi ha la suspensió i rodes tractores.

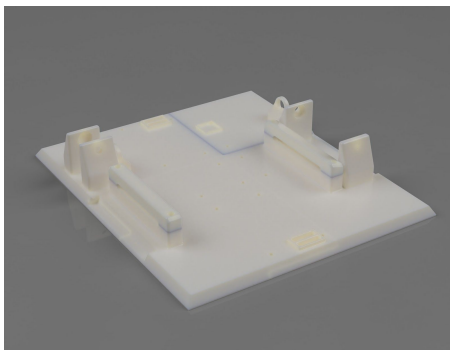


Im.5 Vista general del cos

Im.6 Guia d'unió del cos

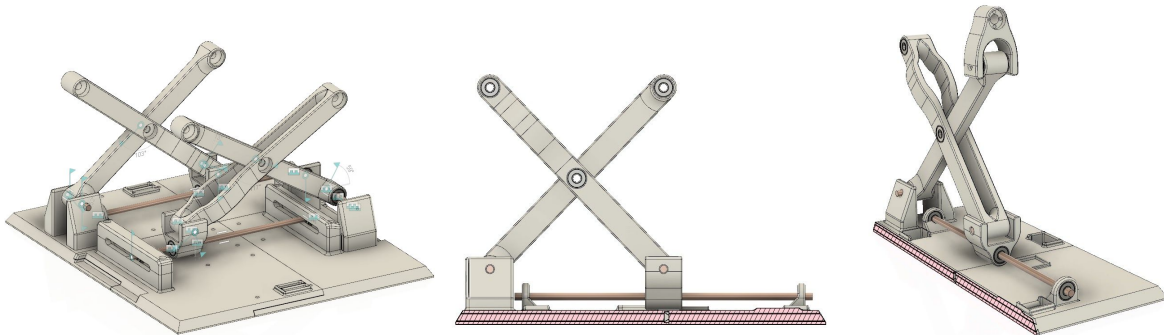
La plataforma: Aquesta és la plataforma que s'anivella automàticament, està impresa en 4 parts per la mateixa raó que el cos del robot, amb la diferència que cada una de les peces són de mides diferents, ja que podien ser més petites i d'aquesta manera evitar que el tall estigui en zones fràgils. Les unions de la plataforma són diferents que les del cos, ja que com s'ha imprès després del cos, s'ha pogut veure que a l'hora d'encaixar les peces es fa molt difícil, per tant, s'ha optat per fer un encavalcament el qual és més fàcil de muntar.

L'únic component electrònic que té la plataforma és l'acceleròmetre que llegeix la inclinació de la plataforma.



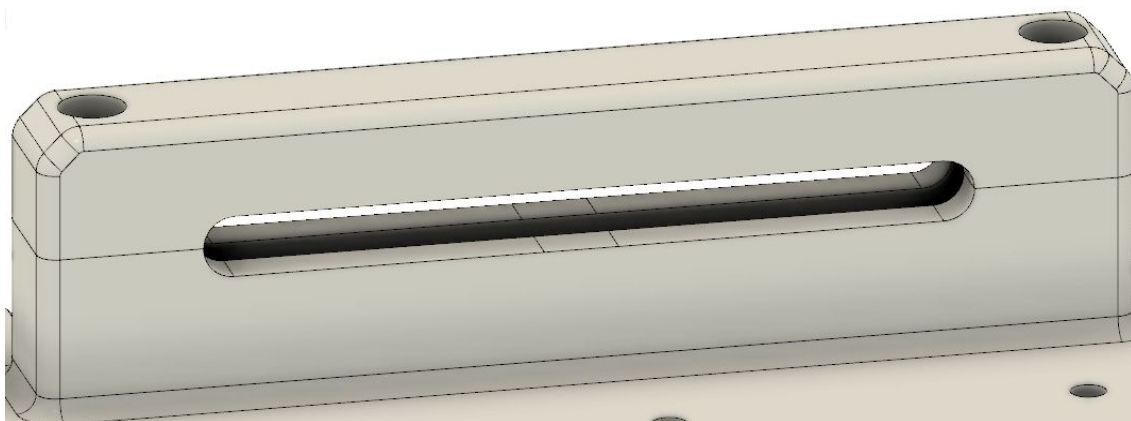
Im.7 Vista general de la plataforma

El sistema d'anivellació de la plataforma: La plataforma utilitza un sistema d'elevació de tisora per poder-se anivellar. Per poder pujar i baixar gira l'eix roscat un motor pas a pas que fa que es mogui horitzontalment la peça que està en contacte amb l'eix. Com es pot observar, una de les tisores està acoblada a un raïl el qual es mou lliurement, aquest és el que permet que la plataforma pugui canviar d'inclinació, ja que si no hi fos, només podria pujar horitzontalment, altrament, es trencarien les tisores. El sistema de tisora té 4 parts que es mouen: dues que roten i que en conseqüència varien l'alçada i dues que tenen un moviment horitzontal que fan rotar les tisores. Les tisores estan una dins de l'altre, per poder-les posar així, s'ha hagut d'eixamplar una de les dues peces per tal de poder inserir una dins de l'altre, ja que la zona on hi ha els coixinets és més ample que la resta.



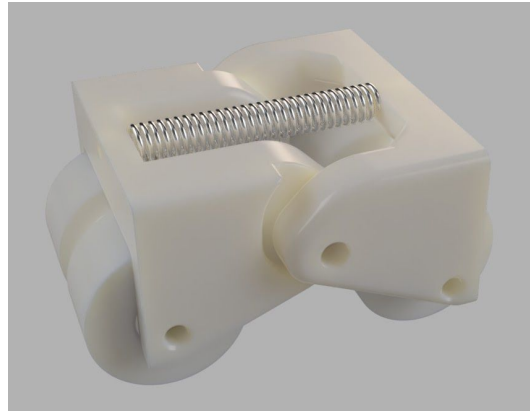
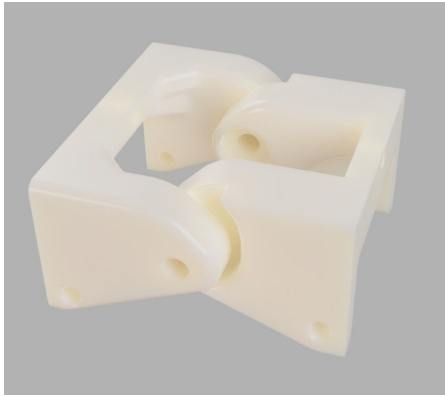
Im.8 Vista general del sistema d'elevació Im.9 Vista frontal de les tisores Im.11 Vista general de les tisores

Tancadora del raïl mòbil: Aquesta és la part superior del raïl, està dividit en dues parts perquè si no, no hi hauria manera de posar el coixinet a dins, per tant s'ha dividit entre 2 i està unit amb 2 caragols M6.



Im.12 Raïl

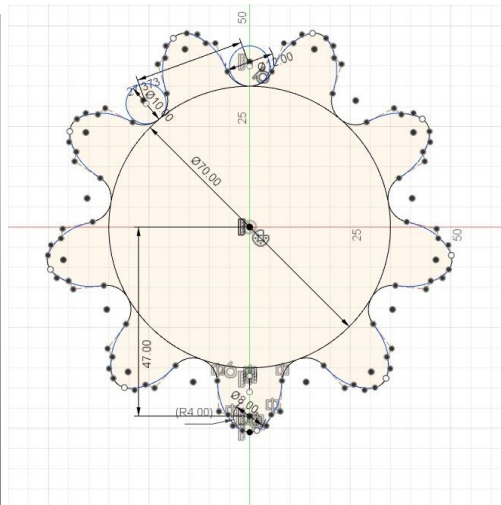
La suspensió: Aquesta és la suspensió que amorteix els cops del robot. Està format per quatre parts principals i una molla: Hi ha dues peces que encaixen i s'uneixen amb un eix. A cada una d'aquestes peces hi ha dues rodetes que són les que estan en contacte amb les erugues.



Im.13 Peces principals de la suspensió

Im.14 Vista general de la suspensió montada

La roda tractora: Aquesta és la roda que engrana amb l'eruga, aquesta peça està impresa en dues parts per evitar gastar material en suports i per millorar-la estèticament. La roda té una transmissió directa amb el motor per evitar pèrdues d'energia i per facilitar el disseny.

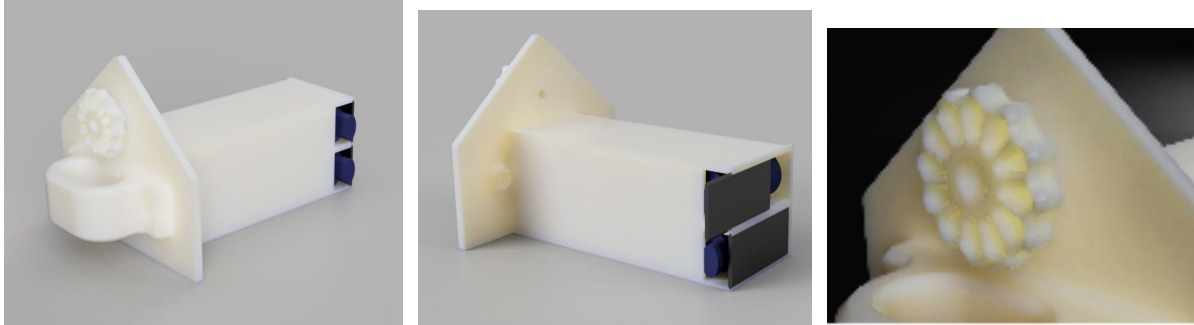


Im.15 Vista general de la roda tractora

Im.16 Croquis de la roda tractora

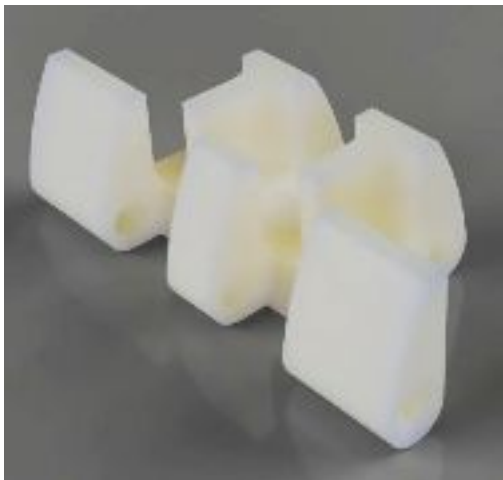
La capsa de la bateria: Aquesta és la capsa on hi ha les bateries. És un component extraïble, ja que és necessari poder treure les bateries per carregar-les i en cas que s'acabés la bateria, poder treure-la manualment. Per fixar la bateria, s'utilitza un vis amb un cap imprès per fer més fàcil i estètic el disseny. Per treure-la amb facilitat hi ha una petita maneta. Per poder treure i posar les bateries de la capsa hi ha dues tapes amb guies que llisquen. Aquesta peça està impresa en dues parts per qüestions de consum de material i acabats finals. La part que s'ha de veure, que és la que té forma de casa, queda brillant perquè aquesta és la primera capa

i ha estat impresa en un llit de vidre. La maneta s'ha imprès a part perquè si no l'acabat no hagués sigut el mateix i s'hauria hagut d'imprimir molt material de suport i sempre s'intenta evitar costos extrems.



Im.17 Vista general de la capsa de la bateria Im.18 Vista posterior de la capsa de la bateria Im.19 Maneta

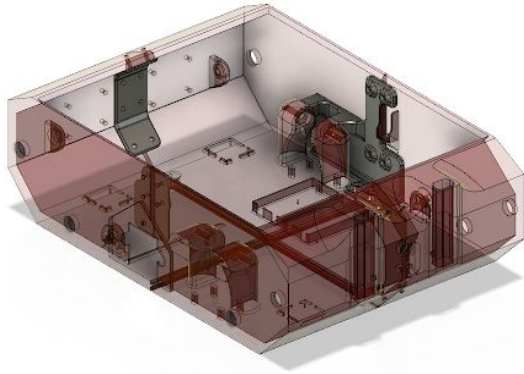
La cadena d'eruga: Aquesta és una peça de la cadena, cada peça té 5 dents, 3 davant i 2 darrere les quals encaixen entre elles. Cada cadena s'encaixa amb una altra amb una vareta de llautó de 5mm. En total hi ha 102 unitats, 51 a cada costat.



Im.20 Vista general de la cadena

Juntes d'unió entre peces: Aquestes són les 4 peces que donen rigidesa a tot el cos, ja que sense ells no hi ha res que uneixin les parets del robot i per tant, es trencaria amb molta facilitat. Dues d'aquestes peces tenen més d'una funció, la primera peça, també té la funció de suportar la part del cos on van les tisores, ja que en un disseny inicial no hi havia res que evités el seu moviment horitzontal, un cop impreses, es va observar que aquella part es podia trencar i per tant es va decidir aprofitar la peça d'unió per reforçar la part feble. Aquest problema va ser corregit a la resta del cos imprès posteriorment. La tercera peça té com a funció extra de tenir el caragol femella que subjecta la bateria un cop inserida en el cos. Aquestes dues peces, a més a més, tenen afegides uns forats per encaixar els caragols femella.

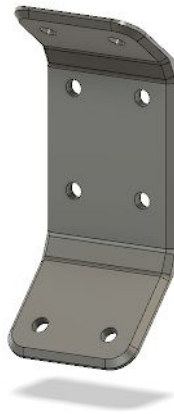
Les altres dues peces no tenen aquesta modificació perquè van ser impreses abans de fer aquesta modificació, de fet, va ser gràcies a aquestes peces que es va veure el problema que tenien els caragols.



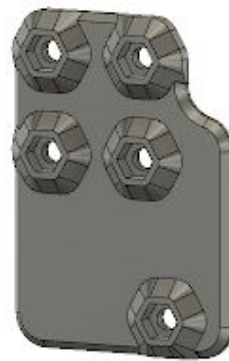
Im.21 Vista general de les unions muntades



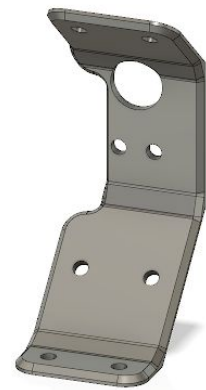
Im.22 Unió 1



Im.23 Unió 2



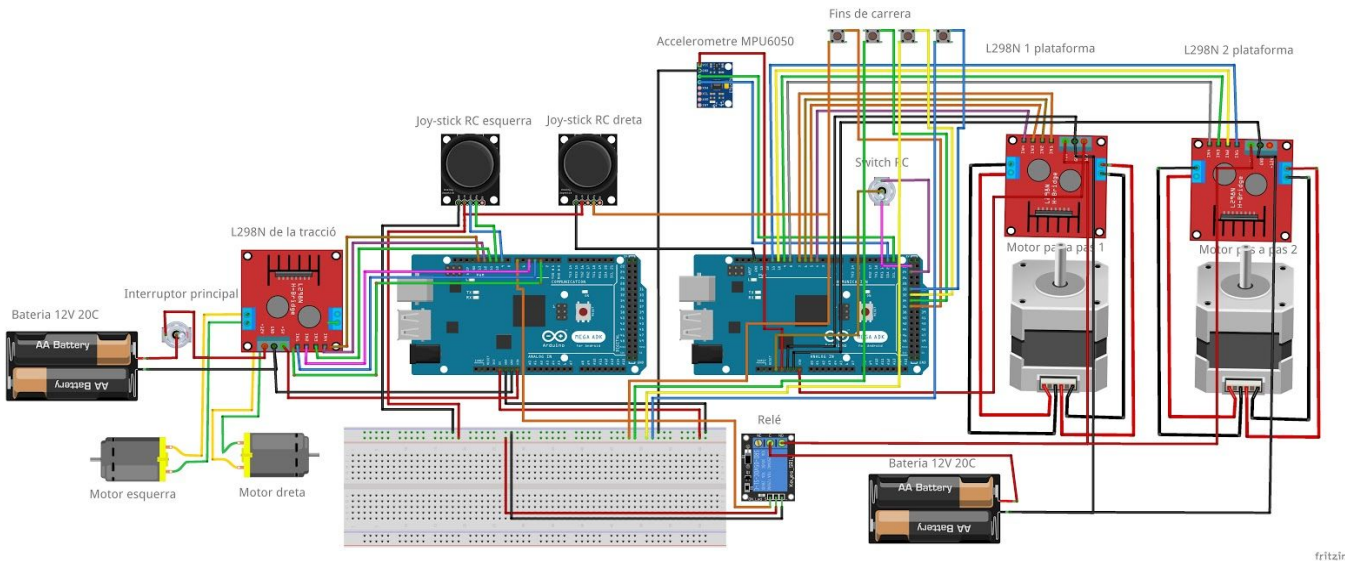
Im.24 Unió 3



Im.25 Unió 4

4. ELECTRÒNICA

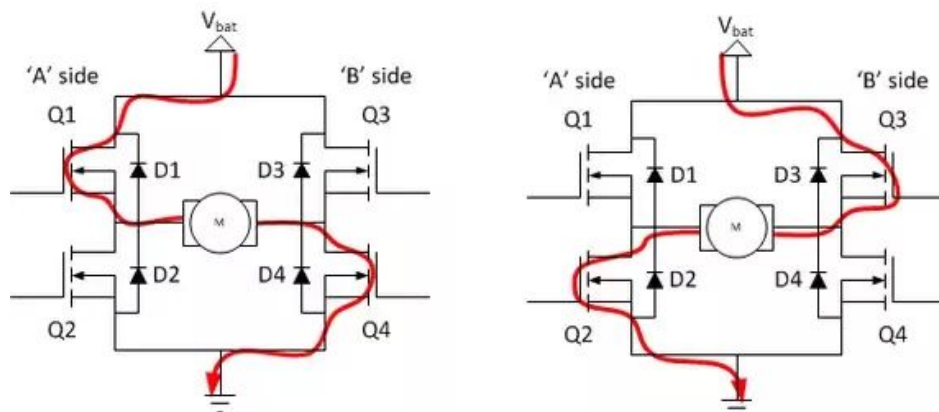
Aquest és el circuit elèctric que té el robot, aquest conté un total de 14 elements connectats a les dues plaques, 16 comptant les bateries.



Im.26 Circuit elèctric

El circuit es divideix en dues parts: la de la tracció i la de l'anivellació.

En el de la tracció, hi ha una bateria de 12 volts de 20C que alimenta els controladors dels motors que té l'interruptor connectat el qual és el que alimenta el driver dels motors. Hi ha 2 motors dc de 12V amb reductora que són el component principal de la tracció del robot, els motors, són controlats a través del driver L298N el qual pot suportar una càrrega de 35V. Aquests drivers, són capaços de moure 2 motors DC alhora en ambdues direccions i lliuren 5V per alimentar la placa Arduino, per poder moure els motors en el sentit de les agulles del rellotge i del revers, tenen 2 ponts H els quals permeten la inversió del sentit del corrent i per tant del motor.



Im.27 Esquema del pont H

La placa Arduino, està connectada al receptor de radiofreqüència que alhora està connectat al joystick de comandament a radiocontrol. El joystick és controlat pel treballador que controli el robot. La placa de la tracció, quan s'encén, acciona un relé que actua com a interruptor pels drivers que controlen els drivers dels motors pas a pas, els components principals de l'anivellació. A diferència dels motors dc, els motors pas a pas requereixen tot un driver L298N a causa que necessiten el doble de cables, per tant, com hi ha 2 motors, hi ha 2 drivers. Un d'aquests drivers alimenta amb 5 volts a la placa, la qual té connectada l'acceleròmetre que llegeix la inclinació de la plataforma, un Switch del comandament radiocontrol que canvia de mode la plataforma (mode autoanivellació a mode elevació i del revers). Amb l'objectiu d'evitar el trencament del robot, hi ha 4 finals de carrera que delimiten el recorregut de les tises. Per últim, té el segon joystick de radiocontrolador que controla l'elevació de la plataforma.

5.PARTS I COST

Producte	Quantitat	Cost individual	Cost total
Arduino mega 2560	2	13,99	27,98
Motor DC 12V amb reductora	2	16	32
Bateria 5000 mah 20C 3cel·les	2	22,99	45,98
Commandament radiocontrol 6 canals 2.4 Hz	1	26,69	26,69
Receptor 6 canals	1	10,33	10,33
Motor pas a pas	2	10,99	21,98
Driver L298N	3	2,164	6,492
Acceleròmetre MPU-6050	1	5,54	5,54
Coixinets	89	0,6215	55,3135
Connectors bullet 4 mm	4	0,5495	2,198
Material d'impressió 3D PLA	8	17	136
Varilla d'acer roscada de 8mmØ i 1 m de llargada	4	10,99	43,96
Varilla roscada d'eix Z d'impresora	2	11,99	23,98
Cargol M3	30	0,0586	1,758
Cargol M6*15mm	29	0,24	6,96
Cargol M6*20mm	10	0,26	2,6
Cargol femella M6	55	0,1565	8,6075
Acoplador de 5mm a 8 mm	2	1,9975	3,995
Acoplador de 6mm a 8 mm	2	1,9975	3,995
Arandela 8mm	20	0,22925	4,585
Màneg de maleta telescòpic	1	20	20
Antiguspies	2	1,56	3,12
Interruptor	1	0,8	0,8
Relé	1	1	1
Fi de carrera	4	2,038	8,152
Cixinet amb cargols	8	2,75	22
		TOTAL	526,016

Im.28 Taula de parts i cost

6. TRACTAMENTS POSTIMPRESSIÓ I CONSTRUCCIÓ

6.1 Tractaments postimpressió:

Totes les peces d'aquest robot han passat per diferents processos postimpressió. Totes elles tenen un procés en comú, tenen una capa d'un epoxi, de 2 components l'A i el B, el qual està pensat per endurir el PLA. El component A s'ha d'utilitzar en el 66,67% de la mescla, aquest component és una resina transparent la qual sense el component B és líquid, el component B, també és líquid i és el 33.33% restant de la mescla, aquest component és l'enduridor el qual fa que la mescla dels 2 creï un sol component sòlid. Aquest material endureix les peces dipositant-se entre les capes d'impresió evitant que quedi cap esquerdada o bombolla que s'hagi pogut produir durant el procés d'impresió, ja que el PLA és un material molt porós i sol deixar petits espais que poden arribar a convertir-se en esquerdes i més tard provocar el trencament de la peça.

Moltes de les peces han passat per un polit amb paper de vidre o amb una dremel per aconseguir bons acabats, que acabi d'encaixar una peça o arreglar imperfeccions que hagin pogut sorgir a causa de la impressió com petits fils per culpa d'algun error de retracció.

L'últim tractament de postimpressió és per les erugues del robot. Com les impressions amb PLA són molt llises, rellisquen molt en el terra, per tant, se'ls hi ha aplicat tres capes de pintura impermeabilitzadora per terrats que no rellisca gens. Amb aquesta pintura s'aconsegueix una bona adherència i que no llisqui en plans inclinats o escales.

6.2 Construcció:

La primera part que es va muntar va ser el cos del tanc, es van encaixar les 4 peces. Després es van posar les unions amb caragols, un cop tot el cos estava unit, es va procedir a pintar tot el cos amb l'epoxi enduridor per acabar de fusionar tot. Un cop tot sec es van posar tots els coixinets al seu lloc a pressió i es van afegir alguns dels coixinets que actuaven com a separadors (a les rodes), després, es van instal·lar els motors tractors i amb ells els eixos que unien la roda engranada amb el motor. Quan les rodes motrius van ser instal·lades es van instal·lar les rodes engranades no conduïdes amb els coixinets per donar-les plena llibertat per rotar, a continuació, es van instal·lar les suspensions de la mateixa manera que les rodes engranades no motrius. Quan tota la part exterior va ser acoblada, es va posar les cadenes d'erugues i es van ajustar el nombre de peces que eren necessàries perquè l'eruga no estigués massa tensada o massa solta. Un cop tota la tracció va ser instal·lada, es van posar els motors pas a pas amb la "fulla" de la tisora que va acoblada als eixos dels motors. Amb això fet, es va muntar la plataforma posant els coixinets al seu forat, després es va enganxar l'acceleròmetre amb un caragol M3 i després es va posar el raïl mòbil al seu lloc tancant-lo amb la peça superior amb 2 caragols M6 després de posar el coixinet a dins. Un cop instal·lada la plataforma, es va poder observar a on estaven els límits de la plataforma, i amb aquesta informació, vam posar tots els finals de carrera al seu lloc. Quan tot això va ser fet, es va posar la resta de l'electrònica al seu lloc (les dues plaques Arduino Mega 2560, els drivers L298N, el relé, interruptor, el receptor i les bateries) i va ser fixada amb caragols M3. Llavors, es va accedir a fer les connexions. Primer es van connectar els motors DC a la placa motriu i es va confirmar que funcionaven, després es van refer les connexions per connectar els motors amb els seus drivers i a les bateries amb l'interruptor. Es va utilitzar un programa de prova per saber si tot funcionava bé i després es va connectar el receptor i es va poder provar el comandament a radiocontrol. Després de fer tota la tracció, es va connectar el relé a la placa motriu i els drivers dels motors pas a pas, després es va connectar els motors pas a pas per comprovar si funcionaven juntament amb els finals de carrera, quan tot això va funcionar, es va connectar l'acceleròmetre i es van fer els ajustaments en el programa, ja que aquests acceleròmetres no són molt precisos i no venen ben calibrats de fàbrica. Després de calibrar-lo es va provar el programa i es van fer els canvis convenientes per millorar el funcionament del robot. Després que funcionés la tracció i l'anivellació, es va afegir la funció

d'elevació la qual només requeria un nou cable que connectes un canal del receptor a la placa nivelladora, ja que s'acciona el mode d'elevació amb el Switch del comandament a radiocontrol. Per últim, se li va afegir la maneta telescòpica per poder guiar el robot en cas que es quedi sense bateria o es desviï del camí.

7. PROGRAMACIÓ

7.1 Programació de la tracció:

El programa de la tracció del tanc és la següent:

La primera part del programa és la definició dels pins que s'utilitzen i s'entren noms per què l'Arduino entengui aquests pseudònims que donem als pins.

```
/*Còdig de la tracció del robot*/

#define rc3 9 //Canal 3 connectat al pin 9 de l'arduino
#define rc4 10 //Canal 4 connectat al pin 10 de l'arduino

#define EnA 3 //EnA(motor dret) del L298N al pin 3 de l'arduino
#define EnB 13 //EnB(motor esquerra) del L298N al pin 13 de l'arduino

#define N1 4 //In 1 del L298N al pin 4 de l'arduino
#define N2 5 //In 2 del L298N al pin 5 de l'arduino
#define N3 11 //In 3 del L298N al pin 11 de l'arduino
#define N4 12 //In 4 del L298N al pin 12 de l'arduino

int ch3; //canal(channel) 3 del receptor
int ch4; //canal(channel) 4 del receptor
```

Im.29 Primera part del programa de tracció

La segona part és el void setup que és on es diu si un pin és d'entrada o de sortida i si des d'un inici volem que els pins estiguin oberts o tancats. A més, obrim el pin serial (el pin d'informació) per poder rebre el senyal.

```
void setup() {

  pinMode (rc3, INPUT); //Pin d'entrada de senyal
  pinMode (rc4, INPUT); //Pin d'entrada de senyal

  pinMode (EnA, OUTPUT); //Pin de sortida de senyal
  pinMode (EnB, OUTPUT); //Pin de sortida de senyal

  pinMode (N1, OUTPUT); //Pin de sortida de senyal
  pinMode (N2, OUTPUT); //Pin de sortida de senyal
  pinMode (N3, OUTPUT); //Pin de sortida de senyal
  pinMode (N4, OUTPUT); //Pin de sortida de senyal

  pinMode (7, OUTPUT); //pin del relé

  digitalWrite(N1, LOW); //Comencen tots els pins apagats
  digitalWrite(N2, LOW);
  digitalWrite(N3, LOW);
  digitalWrite(N4, LOW);

  digitalWrite(7, HIGH); //Menys el del relé, que alimenta la placade nivellació

  Serial.begin(115200); //S'inicia el port serial a 115200Hz

}
```

Im.30 Void setup del programa de tracció

Aquí s'inicia el void loop, aquesta és la part del programa que s'anirà repetint infinitament.

Aquesta part del void loop és la que diu que es llegeixi el senyal dels pins del receptor constantment, i així, no deixar de rebre senyal.

```
void loop() {  
  ch3 = pulseIn(rc3, HIGH); //Llegeix els pulsos del receptor del canal 3  
  ch4 = pulseIn(rc4, HIGH); //Llegeix els pulsos del receptor del canal 4
```

Im.31 Void loop del programa de tracció

Aquí és on el programa diu quan s'ha de parar

```
if (ch4 < 1500 && ch4 > 1300 && ch3 < 1500 && ch3 > 1300) { // Si el joystick està al centre  
  
  digitalWrite(EnA, LOW);  
  digitalWrite(N1, LOW);  
  digitalWrite(N2, LOW);  
  
  digitalWrite(N3, LOW);  
  digitalWrite(N4, LOW);  
  digitalWrite(EnB, LOW);  
  // Tot parat  
}
```

Im.32 Si els joysticks estan centrats

Aquesta part és l'encarregada de fer anar el robot endavant a la dreta, a l'esquerra o recte

```
if (ch4 > 1600) { //Si endavant a dreta/esquerra  
  
  if (ch3 < 1300) { //cap a la dreta  
  
    digitalWrite(EnA, HIGH); //Endavant a la dreta  
    digitalWrite(N1, LOW);  
    digitalWrite(N2, HIGH);  
  
    digitalWrite(N3, LOW);  
    digitalWrite(N4, LOW);  
    digitalWrite(EnB, LOW);  
  }  
  
  else if (ch3 > 1600) { //cap a l'esquerra  
  
    digitalWrite(EnA, LOW);  
    digitalWrite(N1, LOW);  
    digitalWrite(N2, LOW);  
  
    digitalWrite(N3, HIGH); //Andavant a l'esquerra  
    digitalWrite(N4, LOW);  
    digitalWrite(EnB, HIGH);  
  
  }  
}
```

Im.33 Si va endavant

Aquesta part s'encarrega de què fer en cas que el robot vagi enrere a la dreta, esquerra o recte

```
if (ch4 < 1200) { //Si enrere a la dreta/esquerra

    if (ch4 < 1300) { //Cap a la dreta

        digitalWrite(EnA, HIGH); //Enrere a la dreta
        digitalWrite(N1, HIGH);
        digitalWrite(N2, LOW);

        digitalWrite(N3, LOW);
        digitalWrite(N4, LOW);
        digitalWrite(EnB, LOW);

    }

    else if (ch3 > 1600) { //Cap a l'esquerra

        digitalWrite(EnA, LOW);
        digitalWrite(N1, LOW);
        digitalWrite(N2, LOW);

        digitalWrite(N3, LOW ); //Enrera l'esquerra
        digitalWrite(N4, HIGH);
        digitalWrite(EnB, HIGH);
    }

    else { //Qualsevol de opció que no sigui ni a la dreta ni a l'esquerra però si enrere

        digitalWrite(EnA, HIGH); //Enrere a la dreta
        digitalWrite(N1, HIGH );
        digitalWrite(N2, LOW );

        digitalWrite(N3, LOW); //Enrera a la esquerra
        digitalWrite(N4, HIGH);
        digitalWrite(EnB, HIGH);

        //Va enrere recta
    }
}
```

Im.34 Si va enrere

L'última part és la que s'encarrega de quan no va ni endavant ni enrere, sinó que només gira a la dreta, a l'esquerra o s'està quiet.

```
else { //Si no és ni davant ni enrere

    if (ch3 > 1300) { //Dreta

        digitalWrite(EnA, HIGH); //Dreta enrere
        digitalWrite(N1, LOW);
        digitalWrite(N2, HIGH);

        digitalWrite(N3, LOW ); //Esquerra endavant
        digitalWrite(N4, HIGH );
        digitalWrite(EnB, HIGH);

        //Dreta en el seu propi eix
    }

    else if (ch3 > 1600) { //Esquerra

        digitalWrite(EnA, HIGH); //Dreta endavant
        digitalWrite(N1, HIGH);
        digitalWrite(N2, LOW);

        digitalWrite(N3, HIGH ); //Esquerra enrere
        digitalWrite(N4, LOW );
        digitalWrite(EnB, HIGH);

        //Esquerra en el seu propi eix
    }

    else {

        digitalWrite(EnA, LOW); //Quiet
        digitalWrite(N1, LOW);
        digitalWrite(N2, LOW);

        digitalWrite(N3, LOW);
        digitalWrite(N4, LOW);
        digitalWrite(EnB, LOW);
    }
}
}
```

Im.35 Si gira sobre el seu propi eix

Programació de l'anivellació:

El programa de l'anivellació i elevació de la plataforma és el següent:

En aquesta part del programa és on s'anomenen tots els pins, llibreries, motors i valors.

```
//GND - GND
//VCC - VCC
//SDA - Pin A4
//SCL - Pin A5

#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

#include <Stepper.h>

const int PasPerRev = 200; // Passos per revolució

Stepper Motor1(PasPerRev, 8, 9, 10, 11); // inicialitzar la llibreria stepper en els pins 8,9,10 i 11

Stepper Motor2(PasPerRev, 4, 5, 6, 7); //Inicialitzar la llibreria stepper en els pins 4,5,6 i 7

const int addressMPU = 0x68; // Pot ser 0x68 o 0x69
MPU6050 mpu(addressMPU);

int ax, ay, az;
int gx, gy, gz;

int filb = 31; //fi de carrera al màxim de baix del motor 1
int fila = 32; //fi de carrera al màxim d'alt del motor 1

int fi2b = 33; //fi de carrera al màxim de baix del motor 2
int fi2a = 34; //fi de carrera al màxim d'alt del motor 2

int valfilb; //valor dels finals de carrera dels motors 1 i 2 a alt i baix
int valfila;
int valfi2b;
int valfi2a;

int Switch = 24; //canals del switch i canal 1 i 2
int rcl;
int rc2;

int ChSwitch; //Calor dels canals del switch i canal 1 i 2
int Ch1;
int Ch2;

long temps_prev, dt;
float girosc_ang_x, girosc_ang_y;
float girosc_ang_x_prev, girosc_ang_y_prev;
```

Im.36 Primera part del programa d'anivellació

Aquest és el void `updateGiro`, és la funció la qual s'encarrega de llegir els valors de l'accelerometre i traduir-los en graus.

```
void updateGiro() //Per llegir els valors del giroscopi
{
  dt = millis() - temps_prev;
  temps_prev = millis();

  girosc_ang_x = (gx / 131) * dt / 1000.0 + girosc_ang_x_prev;
  girosc_ang_y = (gy / 131) * dt / 1000.0 + girosc_ang_y_prev;

  girosc_ang_x_prev = girosc_ang_x;
  girosc_ang_y_prev = girosc_ang_y;
}
```

Im.37 Void `updateGiro`

Aquí està el void `setup`, en el qual es diu si els pins són d'entrada o de sortida, s'obre el port serial, se li posa un valor a la velocitat dels motors pas a pas i es fa elevar els motors fins a la meitat de la seva alçada màxima.

```
void setup()
{
  Motor1.setSpeed(100); //posar la velocitat a 100 rpm
  Motor2.setSpeed(100); //posar la velocitat a 100 rpm

  Serial.begin(9600); //inicialització del port serial
  Wire.begin();
  mpu.initialize();
  Serial.println(mpu.testConnection() ? F("IMU iniciat correctament") : F("Error a l'iniciar IMU"));

  pinMode(fila, INPUT); //Els pins amb valor d'entrada d'informació
  pinMode(fila, INPUT);
  pinMode(fila, INPUT);
  pinMode(fila, INPUT);
  pinMode(Switch, INPUT);
  pinMode(rc1, INPUT);
  pinMode(rc2, INPUT);

  Motor1.step(400); //Els dos extrems pugen a la meitat
  Motor2.step(400);
}
```

Im.38 Void `setup` del programa d'anivellació

Aquest és l'inici del void loop, aquí es llegeixen tots els valors del receptor i del giroscopi.

```
void loop()
{

  ChSwitch = pulseIn(Switch, HIGH); //Llegir el valor dels canals
  Ch1 = pulseIn(rc1, HIGH);
  Ch2 = pulseIn(rc2, HIGH);

  valfilb = digitalRead(filb); //Donar nom al valor dels finals de carrera
  valfila = digitalRead(fila);
  valfi2b = digitalRead(fi2b);
  valfi2a = digitalRead(fi2a);

  ChSwitch = digitalRead(Switch); //Donar nom al valor del Switch

  mpu.getRotation(&gx, &gy, &gz); // Llegir les velocitats angulars

  updateGiro(); //regir valors del giroscopi

  // Mostrar resultats
  Serial.print(F("Rotacio en X: "));
  Serial.print(girosc_ang_x);
  Serial.print(F("\tRotacio en Y: "));
  Serial.println(girosc_ang_y);
}
```

Im.39 Inici del void loop del programa d'anivellació

Aquí hi ha el que passa en cas que el Switch estigui en mode autoanivellació

```
if (ChSwitch > 1600) { //Si el Switch està en mode autoanivellació

  Serial.begin(9600);

  if (valfila == HIGH || valfi2a == HIGH) { //Si la plataforma està massa alta

    Motor1.step(-PasPerRev); //Baixa per ambdós costats
    Motor2.step(-PasPerRev);

    delay(50);
  }

  if (valfilb == HIGH || valfi2b == HIGH) { //Si la plataforma està massa baixa

    Motor1.step(PasPerRev); //Puja per ambdós costats
    Motor2.step(PasPerRev);
  }

  else { //Si cap dels 2 finals de carrera està activat, es pot autoanivellar

    if (girosc_ang_x < -20) { //Si la plataforma està inclinada cap avall

      Motor1.step(PasPerRev); //Eleva la part baixa de la plataforma
      delay(5);
    }
    else if (girosc_ang_x > 20) { //Si la plataforma està inclinada cap amunt

      Motor1.step(-PasPerRev); //Baixa la part elevada
      delay(5);
    }

    else { //Si està anivellada
      Motor1.step(0); //Motors quietes
      delay(5);
    }
  }
}
}
```

Im.40 Mode autoanivellació

I aquí quan el Switch està en mode anivellació/elevació manual

```
else { //Si els Switch està en mode elevació manual

    Serial.begin(115200);

    if (Ch1 < 1500 && Ch1 > 1300 && Ch2 < 1500 && Ch2 > 1300) { //Si el joystick està centrat

        Motor1.step(0); //Motors quiets
        Motor2.step(0);
    }

    if (Ch1 > 1500) { //Si el joystick està elevat

        Motor1.step(PasPerRev); //Elevar davant
        Motor2.step(0);
    }

    if (Ch1 < 1300) { //Si el joystick està baix

        Motor1.step(-PasPerRev); //Baixar davant
        Motor2.step(0);
    }

    if (Ch2 > 1500) { //Si el joystick està a la dreta

        Motor2.step(PasPerRev); //Pujar darrere
        Motor1.step(0);
    }

    if (Ch2 < 1300) { //Si el joystick etsà a l'esquerra

        Motor2.step(-PasPerRev); //Baixar darrere
        Motor1.step(0);
    }
}
}
```

Im.41 Mode anivellació manual

CONCLUSIONS

L'objectiu d'aquest projecte era crear una maqueta d'un robot controlat a radiocontrol i que fos capaç de transportar objectes i que aquests objectes sempre estiguessin en posició horitzontal. També havia de tenir un control manual per poder elevar la plataforma independentment de la seva inclinació.

Fer un robot de tan grans dimensions i d'una complexitat en el disseny com la d'aquest robot comporta moltes hores de feina i d'esforç. Haver d'imprimir tantes hores provoca un desgast a la impressora molt elevat i per tant hi ha moltes peces que s'espatllen com ara els rodaments, que s'han hagut de canviar, i el hot end que s'ha arreglat, espatllat i substituït per un de nou. Durant el procés de disseny també es troben molts entrebancs com errors de programa o qüestions físiques de disseny amb les quals s'ha de lluitar per poder aconseguir el tipus de moviment desitjat. Per últim, la programació, s'ha hagut d'aprendre noves funcions i inventar línies de codi perquè tot funcioni com ho ha de fer, per tant, no hi ha hagut res fàcil.

Fer un projecte com aquest, t'ensenya a valorar tot el que hi ha radera de qualsevol producte del mercat, ja que la recerca i desenvolupament d'un producte porta un gran esforç d'equip on participen tot tipus de tècnics especialistes com enginyers de materials, mecànics, electrònics, etc. els quals han de assumir el cost de proves i fracassos fins aconseguir l'èxit.

El resultat final del projecte ha sigut exitós, ja que ha complert tots els objectius inicials. La tracció del robot funciona perfectament gràcies als seus grans motors i la pintura antilliscant i la plataforma s'eleva sense problema. L'únic punt fluix que es podria millorar és l'autoanivellació, ja que no té una alta precisió ni velocitat. Es podria haver millorat amb millors sensors i detectors de posició dels raïls.

En conclusió, el projecte ha complert tot el que s'havia proposat en un principi i per tant ha sigut un èxit.

AGRAÏMENTS

Primer de tot vull agrair a la meva tutora de treball de recerca, Mònica Mascort, pel seu seguiment constant i per ajudar-me en tots els dubtes que hagi tingut.

En segon lloc voldria agrair a la meva família i sobretot al meu pare per estar amb mi en tot el camí amb el cost del robot per ajudar-me en entrebancs tècnics i morals, i per donar-me tot el suport que han pogut donar-me.

Per últim, voldria agrair a tot el professorat per donar-me tots els coneixements necessaris per poder fer anar endavant el projecte.

INFOGRAFIA

IOBOTIC (17 d'agost 2016) *Tanque robot con Arduino (Control Rover Tank)*

<https://www.youtube.com/watch?v=24nDwRqKFBk&t=9s>

Nikodem Bartnik(5 de juliol 2018) *How to use RC with Arduino (RC Arduino robot)*

<https://www.youtube.com/watch?v=-yTet2BCdck>

Vt en linea(19 de juny 2017) *Como usar el driver controlador de motores L298N*

<https://www.youtube.com/watch?v=c0L4gNKwjRw>

How To Mechatronics(8 d'agost de 2017) *Arduino DC Motor Control Tutorial -L298N | H-Bridge | PWM | Robot Car*

<https://www.youtube.com/watch?v=I7IFsQ4tQU8>

Ed Lukas(23 de març 2013) *Blinding the Hobby King HK-T4 V2 radio*

https://www.youtube.com/watch?v=JyAyM1f_O3Q

Productes amb el mateix objectiu:

<https://lesjardinsdemathieu.net/immagini-idea-di-carrello-saliscala-elettrico-prezzo/carrello-saliscala-domino-mobile-carrello-saliscala-elettrico-prezzo-e-carrelli-saliscala-zonzini-con-carrello-saliscala-elettrico-prezzo-e-1200x700px/>

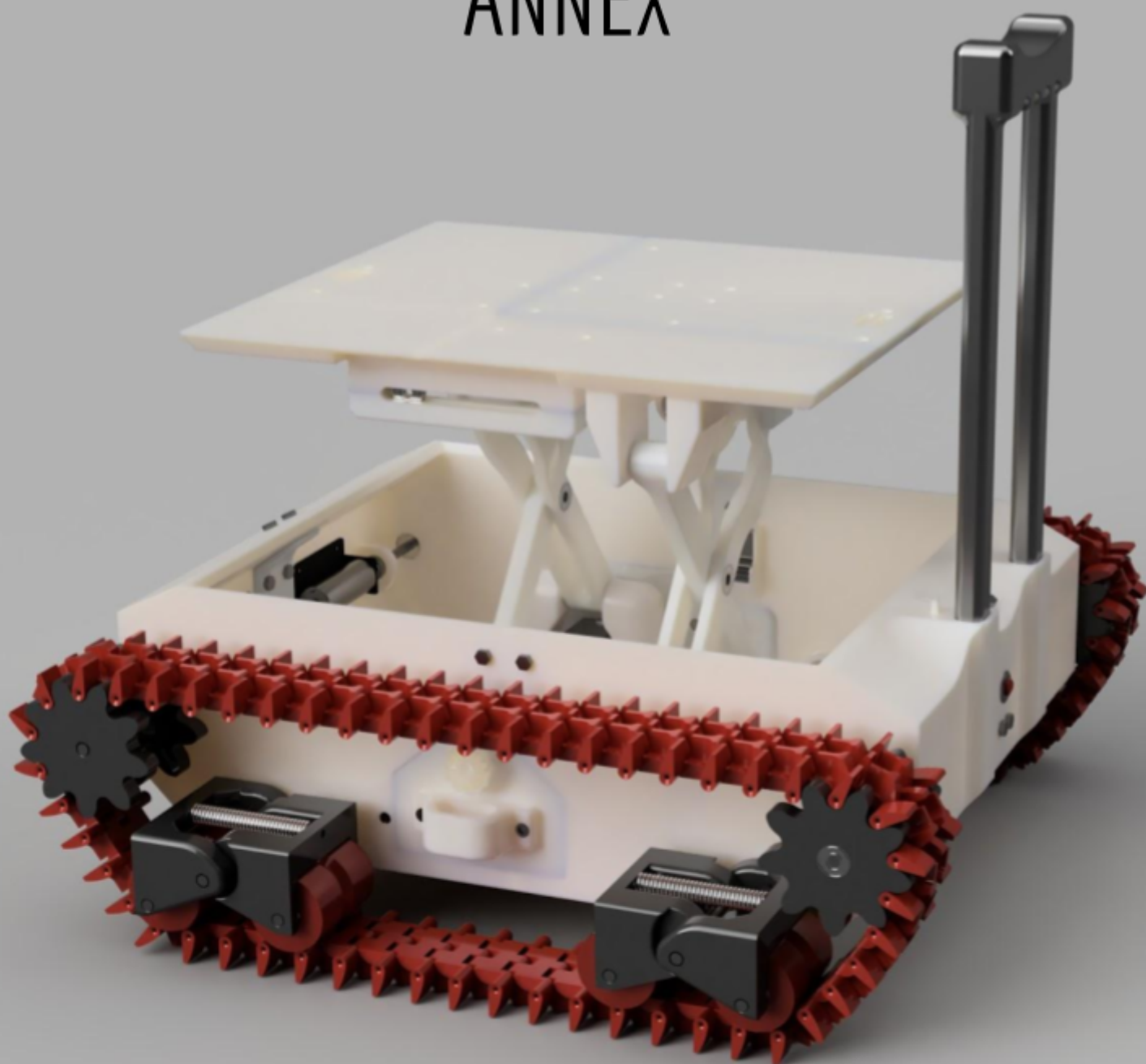
Arduino: Tecnología para todos

<https://arduinodhtics.weebly.com/historia.html>

Arduino

<https://www.arduino.cc/reference/en/>

RC SELF-LEVELING CONVEYOR ANNEX



Mod. MPC 4.F

Marc Puig i Cañellas

Tutora: Mònica Mascort

ÍNDIX de l'Annex

INTRODUCCIÓ.....	Pàg.32
1. DISSENY.....	Pàg.33-34
2.ELECTRÒNICA.....	Pàg.34
3.PROGRAMACIÓ.....	Pàg.35-42

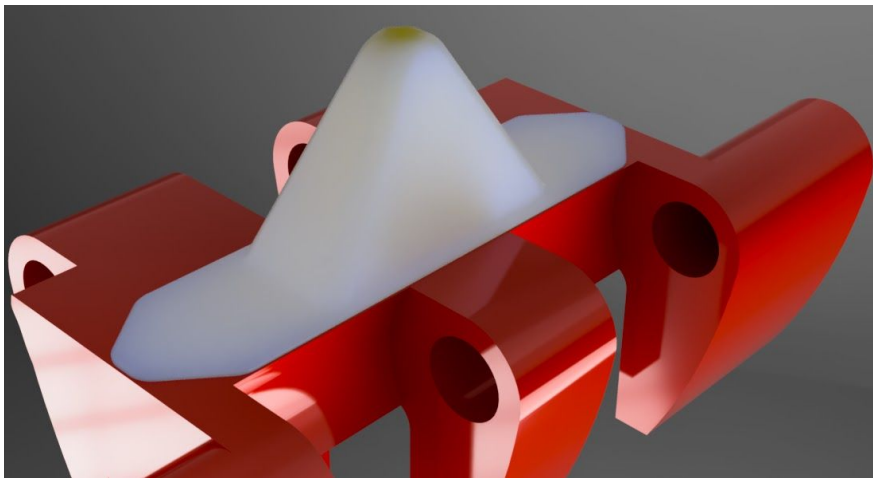
INTRODUCCIÓ

Aquest annex és el producte de la millora del projecte físic, la maqueta, i per evitar incongruències amb el treball escrit, es va veure oportú fer un annex amb els canvis que es van veure convenients com: l'acoblament d'alguna peça, canvis en la programació i en l'electrònica.

1. DISSENY

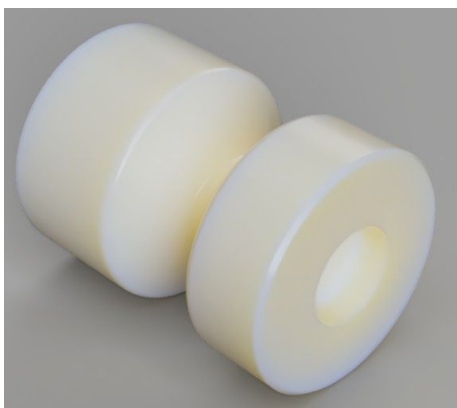
En el disseny hi ha hagut un canvi principal en les erugues per evitar l'escapament de la cadena a l'hora de girar, ja que si es forçava molt i el terreny era poc lliscant, saltava, per tant, es van adherir amb l'epoxi de 2 components unes dents que encaixaven amb les rodets de la suspensió.

Dents: Les dents s'adhereixen amb l'epoxi i s'encaixen entre la ranura de les rodets evitant el lliscament de l'eruga fora del recorregut principal aconseguit així el seu correcte funcionament.

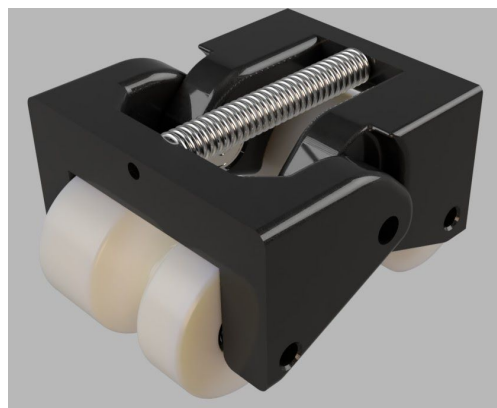


Im.42 Dents de la cadena

Rodeta: A conseqüència de la modificació de la cadena, es va haver de canviar la rodeta per poder encaixar les dents i com no estaven totalment centrades, el centre de la guia es troba desviat uns centímetres.

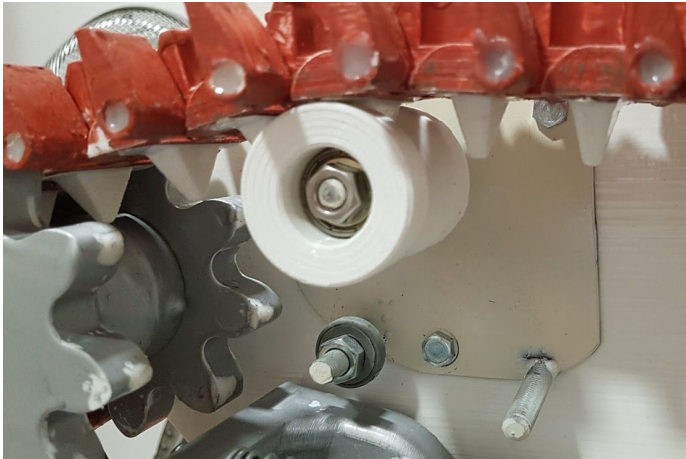


Im.43 Rodeta amb guia



Im.44 Rodeta a la suspensió

Tensor: Per millorar la tensió a la part superior de la cadena, es va haver de fer una estructura metàl·lica a mà, ja que el material d'impressió no aguantaria la força. Perquè les dents poguessin passar adequadament, es va haver d'afegir una rodeta de PLA.



Im.45 Estructura tensor



Im.46 Rodeta tensor

2.ELECTRÒNICA

Hi Ha hagut un canvi en l'electrònica a causa de la simplificació de la programació i de les connexions. S'ha canviat l'acceleròmetre per 2 interruptors d'inclinació de mercuri que tenen un funcionament molt senzill, si s'inclinen cap a un costat es tenca el circuit i si s'inclina cap a l'altre, s'obra, així doncs, si en poses 2 un al costat contrari de l'altre amb una petita inclinació de marge d'error la màquina sap si ha de girar cap a un costat, ca per l'altre o quedar-se quiet tot i que mai se sap quina és la inclinació exacta però no cal perquè el projecte funcioni.

Im.47 Sensor d'inclinació de mercuri

3.PROGRAMACIÓ

Hi ha hagut un canvi radical a la programació, ja que es va reescriure tot totalment tant la tracció com l'anivellació.

Tracció:

```
#define rc3 9 //Canal 3 connectat al pin 9 de l'arduino
#define rc4 10 //Canal 4 connectat al pin 10 de l'arduino

#define EnA 11 //EnA(motor dret) del L298N al pin 3 de l'arduino
#define EnB 3 //EnB(motor esquerra) del L298N al pin 13 de l'arduino

#define N1 12 //In 1 del L298N al pin 4 de l'arduino roda dreta
#define N2 13 //In 2 del L298N al pin 5 de l'arduino roda dreta
#define N3 4 //In 3 del L298N al pin 11 de l'arduino roda esquerra
#define N4 5 //In 4 del L298N al pin 12 de l'arduino roda esquerra

int ch3; //canal(channel) 3 del receptor
int ch4; //canal(channel) 4 del receptor

int vel;
int valvel;

void setup() {

  pinMode (rc3, INPUT); //Pin d'entrada de senyal
  pinMode (rc4, INPUT); //Pin d'entrada de senyal

  pinMode (EnA, OUTPUT); //Pin de sortida de senyal
  pinMode (EnB, OUTPUT); //Pin de sortida de senyal

  pinMode (N1, OUTPUT); //Pin de sortida de senyal
  pinMode (N2, OUTPUT); //Pin de sortida de senyal
  pinMode (N3, OUTPUT); //Pin de sortida de senyal
  pinMode (N4, OUTPUT); //Pin de sortida de senyal

  pinMode (7, OUTPUT); //pin del relé

  digitalWrite(N1, LOW); //Comencen tots els pins apagats
  digitalWrite(N2, LOW);
  digitalWrite(N3, LOW);
  digitalWrite(N4, LOW);

  digitalWrite(7, HIGH); //Menys el del relé, que alimenta la placade nivellació

  Serial.begin(115200); //S'inicia el port serial a 115200Hz

}

void loop() {
  ch3 = pulseIn(rc3, HIGH); // Lee los pulsos del receptor
  ch4 = pulseIn(rc4, HIGH); // Lee los pulsos del receptor

  if (ch4 < 1550 && ch4 > 1350 && ch3 < 1550 && ch3 > 1350)
  {
    analogWrite(EnA, LOW); //Quiet
    digitalWrite(N1, LOW);
    digitalWrite(N2, LOW);

    digitalWrite(N3, LOW);
    digitalWrite(N4, LOW);
    analogWrite(EnB, LOW);
  }
}
```

Im.48 Inici tracció

Im.49 Loop tracció

```

if (ch4 > 1650)
{
  if (ch3 < 1350)
  {
    vel = map(ch3,1650,1000,0,255);
    analogWrite(EnA, HIGH); //Endavant dreta
    digitalWrite(N1, LOW );
    digitalWrite(N2, HIGH);

    digitalWrite(N3, LOW);
    digitalWrite(N4, LOW);
    analogWrite(EnB, LOW);
  }
  else
  {
    if (ch3 > 1650)
    {
      vel = map(ch3,1500,2000,0,255);
      analogWrite(EnA, LOW); //Endavant esquerra
      digitalWrite(N1, LOW);
      digitalWrite(N2, LOW);

      digitalWrite(N3, HIGH);
      digitalWrite(N4, LOW);
      analogWrite(EnB, HIGH);
    }
    else
    {
      vel = map(ch4,1500,2000,0,255);
      digitalWrite(EnA, HIGH); //Endavant
      digitalWrite(N1, LOW );
      digitalWrite(N2, HIGH);

      digitalWrite(N3, HIGH);
      digitalWrite(N4, LOW);
      digitalWrite(EnB, HIGH);
    }
  }
}

else
{
  if (ch4 < 1350)
  {
    if (ch3 < 1350)
    {
      vel = map(ch3,2000,1500,0,255);
      digitalWrite(EnA, HIGH); // Enrere dreta
      digitalWrite(N1, HIGH);
      digitalWrite(N2, LOW);

      digitalWrite(N3, LOW);
      digitalWrite(N4, LOW);
      analogWrite(EnB, LOW);
    }
    else
    {
      if (ch3 > 1650) //else if b)3)
      {
        vel = map(ch3,1500,2000,0,255);
        analogWrite(EnA, LOW); //Enrere esquerra
        digitalWrite(N1, LOW);
        digitalWrite(N2, LOW);

        digitalWrite(N3, LOW );
        digitalWrite(N4, HIGH);
        digitalWrite(EnB, HIGH);
      }
      else
      {
        vel = map(ch4,1500,2000,0,255);
        digitalWrite(EnA, HIGH); //Enrere
        digitalWrite(N1, HIGH );
        digitalWrite(N2, LOW );

        digitalWrite(N3, LOW);
        digitalWrite(N4, HIGH);
        digitalWrite(EnB, HIGH);
      }
    }
  }
}

```

Im.50 Endavant tracció

Im.51 Enrere tracció

```

else
{
  if (ch3 < 1350)
  {
    vel = map(ch3,1360,1000,0,255);
    analogWrite(EnA, vel); //Dreta
    digitalWrite(N1, LOW);
    digitalWrite(N2, HIGH);

    digitalWrite(N3, LOW );
    digitalWrite(N4, HIGH );
    analogWrite(EnB, vel);
  }
  else
  {
    if (ch3 > 1650) //(C)2
    {
      vel = map(ch3,1500,2000,0,255);
      analogWrite(EnA, vel); //Esquerra
      digitalWrite(N1, HIGH);
      digitalWrite(N2, LOW);

      digitalWrite(N3, HIGH );
      digitalWrite(N4, LOW );
      analogWrite(EnB, vel);
    }
    else
    {
      analogWrite(EnA, LOW); //Quiet
      digitalWrite(N1, LOW);
      digitalWrite(N2, LOW);

      digitalWrite(N3, LOW);
      digitalWrite(N4, LOW);
      analogWrite(EnB, LOW);
    }
  }
}
}
}
}
}

```

Im.52 Dreta, esquerra, quiet tracció

Anivellació:

```
//GND - GND
//VCC - VCC
//SDA - Pin A4
//SCL - Pin A5

#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

#include <Stepper.h>

const int PasPerRev = 200; // Passos per revolució

Stepper Motor1(PasPerRev, 6, 7, 8, 9); // inicialitzar la llibreria stepper en els pins 8,9,10 i 11

Stepper Motor2(PasPerRev, 2, 3, 4, 5); //Inicialitzar la llibreria stepper en els pins 4,5,6 i 7

int filb = 31; //fi de carrera al màxim de baix del motor 1
int fila = 32; //fi de carrera al màxim d'alt del motor 1

int fi2b = 33; //fi de carrera al màxim de baix del motor 2
int fi2a = 34; //fi de carrera al màxim d'alt del motor 2

int valfilb; //valor dels finals de carrera dels motors 1 i 2 a alt i baix
int valfila;
int valfi2b;
int valfi2a;

int Switch = 44; //canals del switch i canal 1 i 2
int rc1 = 10;
int rc2 = 11;

int ChSwitch; //Calor dels canals del switch i canal 1 i 2
int Ch1;
int Ch2;

int HgDavant = 22;
int HgAtras = 25;

int ValHgD;
int ValHgA;
```

Im.53 Defició de variables nivellació

```

void setup() {
  Motor1.setSpeed(100); //posar la velocitat a 100 rpm
  Motor2.setSpeed(100); //posar la velocitat a 100 rpm

  Serial.begin(115200);

  ChSwitch = pulseIn(Switch, HIGH); //Llegir el valor dels canals
  Ch1 = pulseIn(rc1, HIGH);
  Ch2 = pulseIn(rc2, HIGH);

  ChSwitch = digitalRead(Switch); //Donar nom al valor del Switch

  pinMode(fila, INPUT); //Els pins amb valor d'entrada d'informació
  pinMode(fila, INPUT);
  pinMode(fila, INPUT);
  pinMode(fila, INPUT);
  pinMode(Switch, INPUT);
  pinMode(rc1, INPUT);
  pinMode(rc2, INPUT);

  pinMode(50, OUTPUT);
  pinMode(51, OUTPUT);
  pinMode(52, OUTPUT);
  pinMode(53, OUTPUT);

  pinMode(28, OUTPUT);
  pinMode(29, OUTPUT);

  digitalWrite(50, HIGH);
  digitalWrite(51, HIGH);
  digitalWrite(52, HIGH);
  digitalWrite(53, HIGH);
  digitalWrite(28, HIGH);
  digitalWrite(29, HIGH);

  pinMode(HgDavant, INPUT);
  pinMode(HgAtras, INPUT);

  Motor1.step(400); //Els dos extrems pugen a la meitat
  Motor2.step(400);
}

```

Im.54 Setup anivellació

```

void loop() {
  ChSwitch = digitalRead(Switch);
  ChSwitch = pulseIn(Switch, HIGH);
  Ch1 = pulseIn(rc1, HIGH);
  Ch2 = pulseIn(rc2, HIGH);

  valfila = digitalRead(fila);
  valfi2a = digitalRead(fi2a);
  valfilb = digitalRead(filb);
  valfi2b = digitalRead(fi2b);
  ValHgD = digitalRead(HgDavant);
  ValHgA = digitalRead(HgAtras);
}

```

Im.55 Loop anivellació


```

if (ChSwitch < 1500 && ChSwitch > 995) //Si està en mode automàtic
{
  if (valfila == LOW || valfi2a == LOW) { //Si la plataforma està massa alta

    Serial.println("Massa alt!");
    Motor1.step(-PasPerRev); //Baixa per ambdos costats
    Motor2.step(-PasPerRev);

    delay(50);
  }

  else if (valfilb == LOW || valfi2b == LOW) { //Si la plataforma està massa baixa

    Serial.println("Massa baix!");
    Motor1.step(PasPerRev); //Puja per ambdos costats
    Motor2.step(PasPerRev);
  }

  else {

    if (ValHgA == HIGH && ValHgD == HIGH) {
      Serial.println("Anivellat");
      Motor1.step(0);
      Motor2.step(0);
    }

    else {

      if (ValHgD == HIGH) {

        Serial.println("Hg D");
        Motor1.step(PasPerRev);
        Motor2.step(0);
      }
      else if (ValHgA == HIGH) {

        Serial.println("Hg A");
        Motor1.step(-PasPerRev);
        Motor2.step(0);
      }
    }
  }
}
}

```

Im.56 Automàtic anivellació

```

else { //Si està en mode manual
  if (valfila == LOW || valfi2a == LOW) { //Si la plataforma està massa alta
    Serial.println("Massa alt!");
    Motor1.step(-PasPerRev); //Baixa per ambdós costats
    Motor2.step(-PasPerRev);

    delay(50);
  }

  else if (valfilb == LOW || valfi2b == LOW) { //Si la plataforma està massa baixa
    Serial.println("Massa baix!");
    Motor1.step(PasPerRev); //Puja per ambdós costats
    Motor2.step(PasPerRev);
  }

  else { //Si cap dels 2 finals de carrera està activat, es pot autoanivellar

    Serial.println(Ch2);

    if (Ch1 < 1500 && Ch1 > 1300 && Ch2 < 1600 && Ch2 > 1300) { //Si el joystick està centrat
      Serial.println("parat");
      Motor1.step(0); //Motors quiets
      Motor2.step(0);
    }

    else if (Ch1 > 1500) { //Si el joystick està elevat
      Serial.println("dreta endavant");
      Motor1.step(PasPerRev); //Elevar davant
      Motor2.step(0);
    }

    else if (Ch1 < 1300) { //Si el joystick està baix
      Serial.println("dreta abaix");
      Motor1.step(-PasPerRev); //Baixar davant
      Motor2.step(0);
    }

    else if (Ch2 > 1600) { //Si el joystick està a la dreta
      Serial.println("esquerra adalt");
      Motor2.step(PasPerRev); //Pujar darrere
      Motor1.step(0);
    }

    else if (Ch2 < 1300) { //Si el joystick està a l'esquerra
      Serial.println("esquerra abaix");
      Motor2.step(-PasPerRev); //Baixar darrere
      Motor1.step(0);
    }
  }
}
}
}

```

Im.57 Manual anivellació