
R2B2-Nano Documentation

Pau Roura

10 de set., 2018

1	1. R2B2-nano OpenSource ROV	3
1.1	1.1. Descàrrega del projecte	3
1.1.1	1.1.1. GitHub R2B2-nano	4
1.1.2	1.1.2. EasyEDA	4
1.1.3	1.1.3. Documentació	4
1.2	1.2. Col·labora	5
2	2. Disseny	7
2.1	2.1. OpenSCAD	8
2.2	2.2. Configuració	8
2.3	2.3. Parts	9
2.3.1	2.3.1. Base de càrrega	9
2.3.2	2.3.2. Xassís	9
2.3.3	2.3.3. Propulsors	11
2.3.4	2.3.4. Boia	15
2.3.5	2.3.5. Carcassa	15
2.4	2.4. Generar STL	16
3	3. Electrònica	21
3.1	3.1. Alimentació	21
3.1.1	3.1.1. Esquemàtics	21
3.1.2	3.1.2. Muntatge	21
3.2	3.2. Control	21
3.2.1	3.2.1. Esquemàtics	24
3.2.2	3.2.2. Muntatge	24
3.2.3	3.2.3 Connexions ESP32	25
3.3	3.3. Components	25
3.3.1	3.3.1. ESP32-WROVER-I (Espressif)	25
3.3.2	3.3.2. MPU-6050 (InvenSense)	26
3.3.3	3.3.3. TB6612FNG (Thosiba)	28
3.3.4	3.3.4. TP4056 & DW01A	28
3.3.5	3.3.5. Cargador Qi i Receptor Qi	28
3.3.6	3.3.6. Bateria LiPo	28
3.3.7	3.3.7. Reguladors de Voltatge	39
3.3.8	3.3.8. Placa PCB R2B2-nano	39
3.3.9	3.3.9. Connector IPEX/U.FL 20278 1.13mm	39
3.3.10	3.3.10 Motor de drone	39

4	4. Software	43
4.1	4.1. MQTT server	43
4.2	4.2. Firmware	45
4.2.1	4.2.1. Arduino IDE	45
4.2.2	4.2.2. Codi del Firmware	45
4.3	4.3. Control	52
4.3.1	4.3.1. Processing IDE	52
4.3.2	4.3.2. Interfície Control	52
4.3.3	4.3.3. Codi del Control	55
5	5. Muntatge	57
6	5.1. Base de Càrrega	61
6.1	5.1.1. Que necessitem?	61
6.1.1	5.1.1.1 Material	61
6.1.2	5.1.1.2 Eines	61
6.2	5.1.2 Muntatge	61
7	5.2. Xassís Alimentació	67
7.1	5.2.1. Que necessitem?	67
7.1.1	5.2.1.1 Material	67
7.1.2	5.2.1.2 Eines	67
7.2	5.2.2 Muntatge	68
8	5.3. Xassís Control	87
8.1	5.3.1. Que necessitem?	87
8.1.1	5.3.1.1 Material	87
8.1.2	5.3.1.2 Eines	87
8.2	5.3.2 Muntatge	88
9	5.4. Propulsors	95
9.1	5.4.1. Que necessitem?	95
9.1.1	5.4.1.1 Material	95
9.1.2	5.4.1.2 Eines	95
9.2	5.4.2 Muntatge	95
10	5.5. Boia	101
10.1	5.5.1. Que necessitem?	101
10.1.1	5.5.1.1 Material	101
10.1.2	5.5.1.2 Eines	101
10.2	5.5.2 Muntatge	101
11	5.6. Carcassa	107
11.1	5.6.1. Que necessitem?	107
11.1.1	5.6.1.1 Material	107
11.1.2	5.6.1.2 Eines	107
11.2	5.6.2 Muntatge	107
12	5.7. Acoblament	111
12.1	5.7.1. Que necessitem?	111
12.1.1	5.7.1.1 Material	111
12.1.2	5.7.1.2 Eines	112
12.2	5.7.2 Muntatge	112
13	A.1. Materials	123

13.1	A.1.1. Cargols	123
13.2	A.1.1. Cable Silicona 28AWG	127
13.3	A.1.1. Connectors Dupont	127
13.4	A.1.1. Cable Coaxial	127
13.5	A.1.1. Estany	127
13.6	A.1.1. Resina Polièster	127
13.7	A.1.1. Funda Termoretractil	127
13.8	A.1.1. Cola Tèmica	127
14	A.2. Eines	133
14.1	A.2.1. Pistola de cola tèrmica	133
14.2	A.2.2. Soldador	133
15	Indices and tables	137



1. R2B2-nano OpenSource ROV

L'R2B2-nano és un projecte OpenSource per a la realització d'un ROV imprimible en 3D. Un ROV, acrònim de l'anglès Remotely Operated underwater Vehicle o vehicle submarí operat remotament, és un vehicle submari controlat a distància per un operador a la superfície.

Principals característiques de l'R2B2-nano:

- Microcontrolador ESP32 (Espressif)
- Giroscopi i acceleròmetre MPU6050 (InvenSense)
- Comunicacions Wifi i Bluetooth
- Càrrega sense fils de les bateries
- 4 Propulsors (2 horitzontals i 2 verticals)
- Actualització del Firmware amb Wireless
- Firmware programable amb Arduino IDE
- Programa de control per Android
- Xassís imprimible en 3D
- Baix cost

1.1 1.1. Descàrrega del projecte

Per a poder crear el teu R2B2-nano cal que et descarreguem els dissenys i programes que hi ha als repositoris de [GitHub](#).

Nota: Els passos que es proposen en tota la documentació estan executats al terminal d'un sistema operatiu **Debian** amb el **git** instal·lat. No hi ha d'haver cap problema per realitzar les mateixes tasques sobre altres sistemes operatius com Window o macOS.

1.1.1 1.1.1. GitHub R2B2-nano

El Disseny 3D, el Firmware pel microcontrolador ESP32 i el codi per generar l'aplicació mòbil és pot trobar al GitHub del projecte [r2b2-nano](#).

També es pot descarregar amb la comanda següent:

```
r2b2@r2b2os:~$ git clone https://github.com/r2b2osrov/r2b2-nano.git
Cloning into 'r2b2-nano'...
remote: Counting objects: 425, done.
remote: Compressing objects: 100% (178/178), done.
remote: Total 425 (delta 114), reused 235 (delta 85), pack-reused 155
Receiving objects: 100% (425/425), 21.77 MiB | 2.52 MiB/s, done.
Resolving deltas: 100% (198/198), done.

r2b2@r2b2os:~$ cd r2b2-nano/
r2b2@r2b2os:~$ ls -l
total 52
drwxrwxr-x 8 r2b2 r2b2 4096 ago 24 08:59 code
drwxrwxr-x 4 r2b2 r2b2 4096 ago 24 08:59 design
drwxrwxr-x 5 r2b2 r2b2 4096 ago 24 08:59 electronics
-rw-rw-r-- 1 r2b2 r2b2 35147 ago 24 08:59 LICENSE
-rw-rw-r-- 1 r2b2 r2b2 32 ago 24 08:59 README.md
```

Tots el dissenys 3D de les peces realitzats amb OpenSCAD i els seus STL per enviar a imprimir estan dins la carpeta **design**. Per com fer-ne ús i personalitzar el vostre R2B2-nano mireu l'apartat [Disseny](#).

Dins la carpeta **code** hi podem trobar diferents programes d'[Arduino IDE](#) per testejar la placa i els components electrònics, el Firmware de l'R2B2-nano i el programa de control fet amb [Processing IDE](#) per controlar l'R2B2-nano des del mòbil. Tots aquests punts estan explicats amb més detall a l'apartat [Software](#).

1.1.2 1.1.2. EasyEDA

Tots els esquemes electrònics i els dissenys de les plaques PCB estan allotjades en un projecte públic de [EasyEDA](#). Des d'aquesta mateixa pàgina es poden demanar les plaques PCB a JLCPCB o descarregar els fitxer Gerber per enviar la placa a producció a qualsevol altre lloc.

1.1.3 1.1.3. Documentació

Aquesta documentació està realitzada amb [SPHINX](#) i també està disponible per descàrrega i edició al GitHub [Documentació R2B2-nano](#).

També es pot descarregar amb la comanda següent:

```
r2b2@r2b2os:~$ git clone https://github.com/r2b2osrov/r2b2-nano-docs-ca
Cloning into 'r2b2-nano-docs-ca'...
remote: Counting objects: 222, done.
remote: Compressing objects: 100% (181/181), done.
remote: Total 222 (delta 26), reused 214 (delta 18), pack-reused 0
Receiving objects: 100% (222/222), 14.79 MiB | 2.56 MiB/s, done.
Resolving deltas: 100% (26/26), done.
```

Un cop descarregada podem editar-la i generar localment els fitxers html per comprovar que ho hem fet correctament. Per generar la documentació cal que estigui instal·lat el programa SPHINX disponible per varis Sistemes Operatius en el següent [enllaç](#).

```

r2b2@r2b2os:~$ cd r2b2-nano-docs-ca/
r2b2@r2b2os:~/r2b2-nano-docs-ca$ make html
Running Sphinx v1.7.2
loading translations [ca]... done
making output directory...
loading pickled environment... not yet created
building [mo]: targets for 0 po files that are out of date
building [html]: targets for 13 source files that are out of date
updating environment: 13 added, 0 changed, 0 removed
reading sources... [100%] software/index
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [100%] software/index
generating indices... genindex
writing additional pages... search
copying images... [100%] assembly/40_thrusters_images/40_06_thrusters_assembly.jpg
copying static files... done
copying extra files... done
dumping search index in English (code: en) ... done
dumping object inventory... done
build succeeded, 6 warnings.

The HTML pages are in build/html.

pau@kisilu:~/r2b2-nano-docs-ca$ cd build/html/
pau@kisilu:~/r2b2-nano-docs-ca/build/html$ ls
assembly design electro genindex.html _images index.html objects.inv r2b2 _
↪search.html searchindex.js software _sources _static

```

1.2 1.2. Col·labora

l'R2B2-nano és un projecte Open Source i està obert a qualsevol tipus de col·laboració. Com pots col·laborar?

Construeix el teu R2B2 i comenta-ho a les xarxes, envia'ns feedback i disfruta del projecte!!!

Escriu Codi per millorar els Firmware i el programa per controlar el R2B2-nano.

Reporta Issues als nostres projectes públics com [GitHub](#) o [EasyEDA](#).

Contribueix en la Documentació tan sigui millorant-la per que es pugui entendre millor com trauint-la a altres idiomes.

2. Disseny

El disseny de l'R2B2-nano és completament parametritzable i es pot adaptar ràpidament a les necessitats o dispositius de que es disposin tan sols canviant pocs paràmetres.

A continuació explicarem com treballar amb el disseny i quins són els paràmetres que afecten a cada part del disseny.

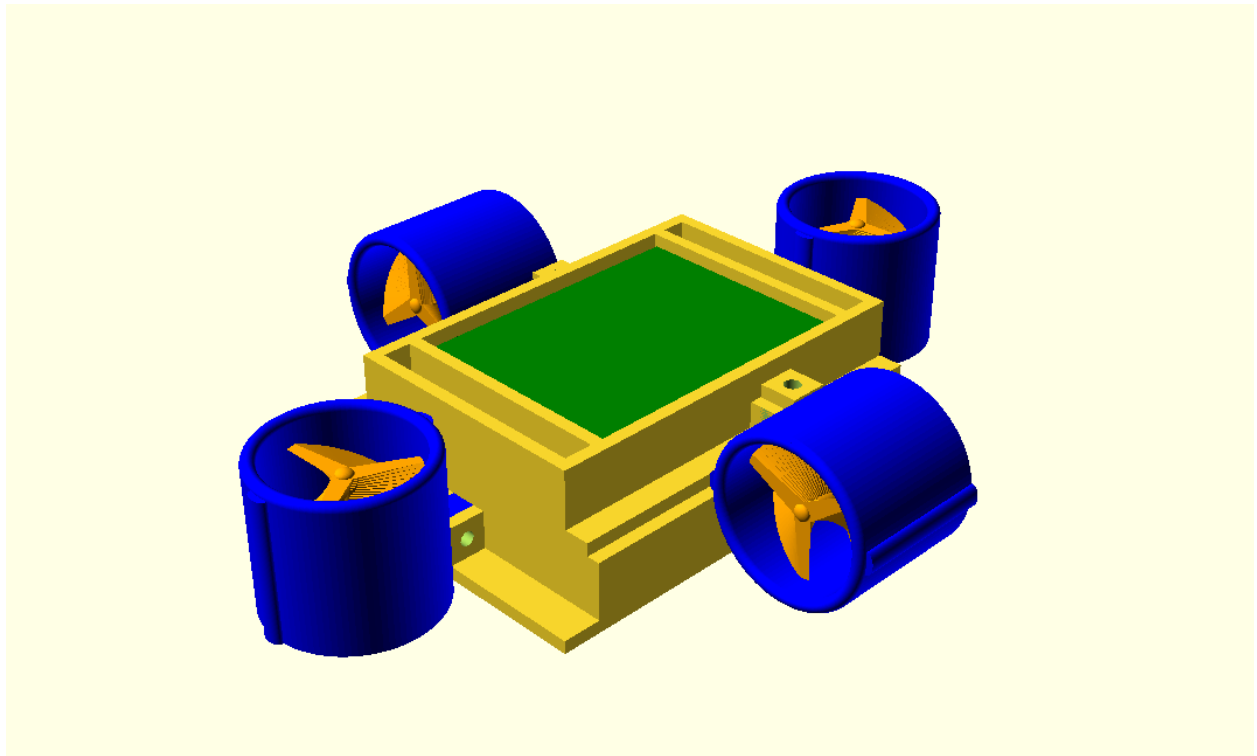


Fig. 1: R2B2-nano 3D (**r2b2-nano.scad**).

2.1 2.1. OpenSCAD

El disseny de l'R2B2-nano està realitzat 100% en [OpenSCAD](#).

OpenSCAD és una aplicació lliure per crear objectes sòlids de CAD. No és un editor interactiu sino un compilador 3D basat en un llenguatge de descripció textual. Un document d'OpenSCAD especifica primitives geomètriques i defineix com son modificades i manipulades per reproduir un model 3D.

Que l'OpenSCAD sigui un llenguatge ens permet per una banda parametritzar tots els objectes de forma molt senzilla i per altra banda ens permet col·laborar d'una manera molt més àgil, ja que la modificació d'un objecte queda reflexada en quina línia de codi s'ha modificat, no en tot un fitxer binari o d'aplicació.

És molt util tindre l'[OpenSCAD CheatSheet](#) a mà cada cop que volem fer algun disseny o modificació.

2.2 2.2. Configuració

Tots els paràmetres es troben al fitxer **config.scad** de la carpeta **design**. Podem definir l'amplada de les parets de tot el 3D, les mides del motors, dels propulsors, la mida dels cargols, etc... i cada paràmetre que ajustem fa que tot el disseny s'hi adapti, així de forma ràpida i sense haver de redibuixar tot el 3D disposem de les peces idònies al nostre muntatge.

A més a més dels paràmetres que especifiquen les mides del disseny hi ha el paràmetre **\$fn** que defineix en número de fragments en que es renderitzaran les circumferències completes. Com més gran sigui el valor de \$fn més tardarà a fer el renderitzat. No és recomanable un valor superior a 100.

```
/*
Title:          config.scad for R2B2 Nano
Description:    All configuration parameters for R2B2-Nano
Authors:       Pau Roura (@prouura)
Date:          20180721
Version:       0.2
Notes:
*/

//General Config
$fn = 100;
w_walls=2.7;           //width of the walls
d_screw_h=3;           //screw thread hole diameter
d_screw_head=6.5;      //screw head diameter
d_screw_p=d_screw_h+0.4; //screw pass through diameter
h_support=8;           //screw support height
w_support=8;           //screw support width
w_chassis=70;          //chassis width
d_chassis=100;         //chassis depth
h_battery=12;          //battery height
h_control=12;          //electronics height

//motor
d_motor=7.6;           //motor diameter
d_motor_shaft=1.5;     //motor shaft diameter
h_motor=16.60;         //motor height

//motor_housing
w_walls_m=1;           //width of the walls
gap_housing=3;         //extra height of the encapsulation
h_motor_body=h_motor+(gap_housing*2); //motor housing height
```

(continues on next page)

(ve de la pàgina anterior)

```
//Thrusters
d_motor_t=d_motor+(w_walls_m*2); //motor support diameter
h_motor_sup=8; //motor support height
o_motor=0; //motor support distance from ground
d_thruster=34; //thruster diameter
h_thruster=30; //thruster height

//module propeller
h_propeller=10; //propeller height
o_propeller=1; //offset of the blades to shaft
n_blade=3; //number of blades
s_blade=20; //angle separation between blades
w_blade=4; //width of the blades
rounded="empty"; //shape of the blades [round | trian | empty]
d_motor_grub=2; //motor grub screw diameter
```

Fitxer config.scad.

2.3 2.3. Parts

2.3.1 2.3.1. Base de càrrega

La base de càrrega és on hi van les dues bobines transmissores capaces de generar el camp magnètic que un cop connectada amb els USBs a la corrent i posem l'R2B2-nano a sobre ens carregarà les bateries.

La base de càrrega tan sols utilitza com a paràmetres les mides del cargols que volem fer servir. Per tant tan sols comprovarem els següents paràmetres a la configuració:

```
/*
Title: config.scad for R2B2 Nano
*/

d_screw_h=3, //screw hole diameter
d_screw_p=d_screw_h+0.4 //screw pass through diameter
d_screw_head=6.5 //screw head diameter
```

Paràmetre	Valor per defecte	Descripció
d_screw_h	3 mm	Diàmetre de la tija del cargol on ha de collar
d_screw_p	d_screw_h+0.4 mm	Diàmetre on ha de passar el cargol sense collar
d_screw_head	6.5 mm	Diàmetre del cap del cargol

2.3.2 2.3.2. Xassís

El xassís de l'R2B2-nano està format per dues parts. Una on hi ha tot el sistema d'alimentació (bobines receptores de corrent, controladors de càrrega per les bateries, les bateries i el regulador de voltatge) i l'altre on hi ha tota l'electrònica de control (Microcontrolador, el giroscopi/acceleròmetre, drivers de motors, etc...).

S'ha fet un disseny modular separant l'alimentació del control a propòsit, ja que per aconseguir impermeabilitzar tota l'electrònica s'omplen els xassís de resina de polièster, que un cop endurida fa que no es pugui accedir a modificar el contingut. El fet de tenir els dos mòduls separats fa que podem reaprofitar-los en cas de voler fer modificacions en alguna de les parts.

Fig. 2: Base de càrrega (**base.scad**).

Els principals paràmetres que cal tenir en compte a l'hora de configurar el xassís és l'amplada, la llargada i la profunditat dels dos mòduls per tal d'assegurar-nos que quan fem el bany de resina de polièster l'electrònica quedi totalment coberta.

```
/*
Title: config.scad for R2B2 Nano
*/

d_screw_h=3,          //screw hole diameter
w_walls=2.7,          //wall width
w_support=8,          //support width
h_support=8,          //support depth
w_chassis=70,         //chassis width
d_chassis=100,        //chassis depth
h_battery=12,         //chassis power height
h_control=12          //chassis control height
```

Paràmetre	Valor per defecte	Descripció
d_screw_h	3 mm	Diàmetre de la tija del cargol on ha de collar
w_walls	2.7 mm	Amplada de les parets
w_support	8 mm	Amplada dels suports del propulsors
h_support	8 mm	Profunditat dels suports del propulsors
w_chassis	70 mm	Amplada del xassís
d_chassis	100 mm	Profunditat del xassís
h_battery	12 mm	Alçada de la zona d'electrònica d'alimentació
h_control	12 mm	Alçada de la zona d'electrònica de control

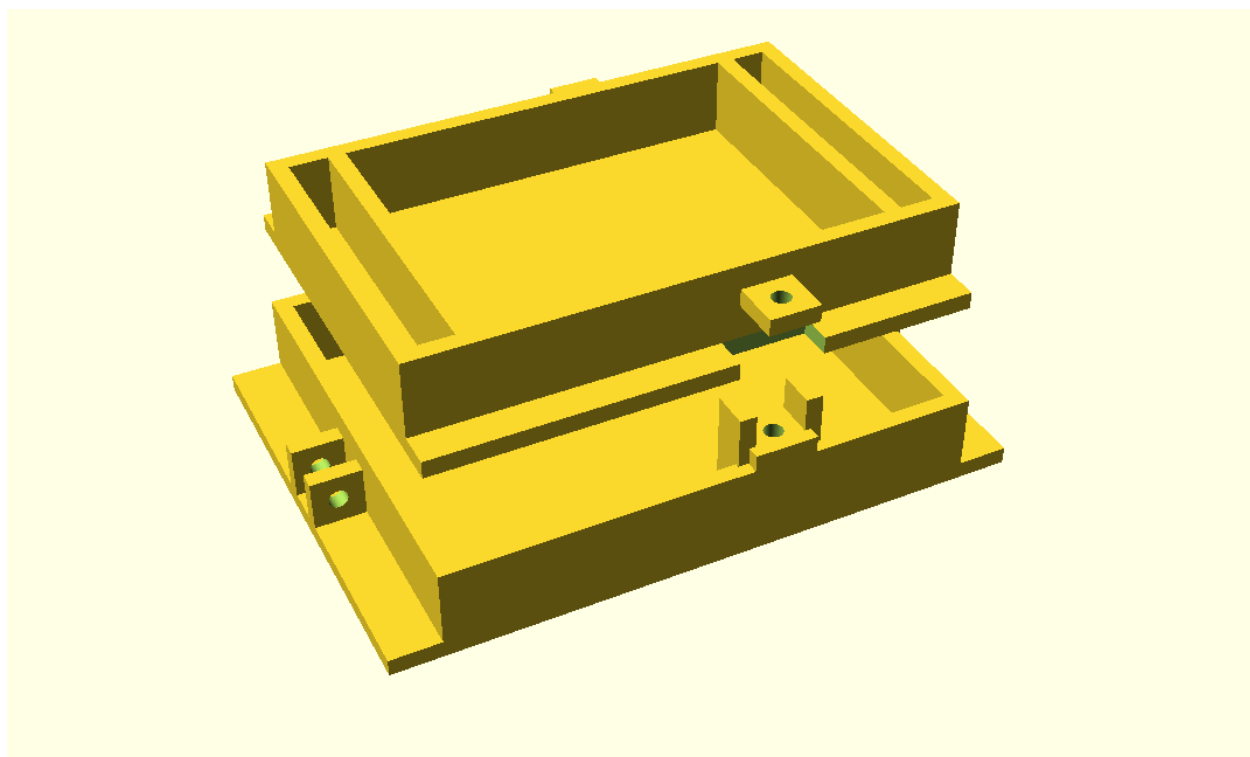


Fig. 3: Xassís d'alimentació i control (**chassis.scad**).

L'R2B2-nano es programa a través de Wifi (OTA / Over-the-air programming), però pot ser que algun cop ens trobem amb problemes, es carregui un firmware erròni i no ens permeti la programació a través de wifi.

Per això és important deixar un port sèrie disponible per poder accedir a programar l'ESP32 amb conversor USB to tty. Aquest port sèrie es compon de 5 cables i mentre no es fan servir cal que estiguin protegits per no crear curtcircuits. Els cables que componen el port es col·loquen dins el protector del port i es farceix de vaselina per que no hi pugui entrar l'aigua.

L'únic paràmtre que hem de tenir en compte és la mida del cargols que farem servir per collar el protector.

```
/*
Title: config.scad for R2B2 Nano
*/

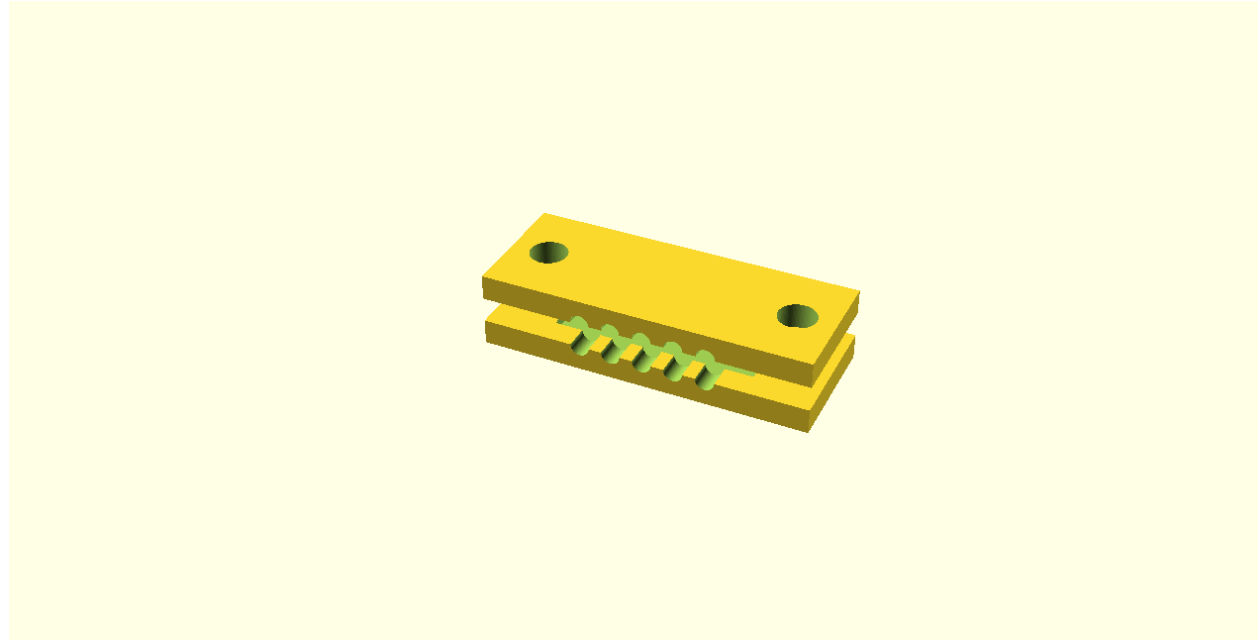
d_screw_h=3 //screw hole diameter
```

Paràmetre	Valor per defecte	Descripció
d_screw_h	3 mm	Diàmetre de la tija del cargol on ha de collar

2.3.3 2.3.3. Propulsors

L'R2B2-nano està format per 4 propulsors que es componen de 3 parts: Carcassa del propulsor, carcassa del motor i hèlix.

La carcassa dels propulsors ens serveix per collar els motors al xassís i per evitar que l'hèlix pugui col·lisionar amb qualsevol objecte quan el motor està en marxa. Els principals paràmetres que cal tenir en compte són les mides del motor.

Fig. 4: Protecció pel port sèrie (**prog_protect.scad**).

A continuació detallem quins paràmetres de la configuració afecten al propulsor:

```

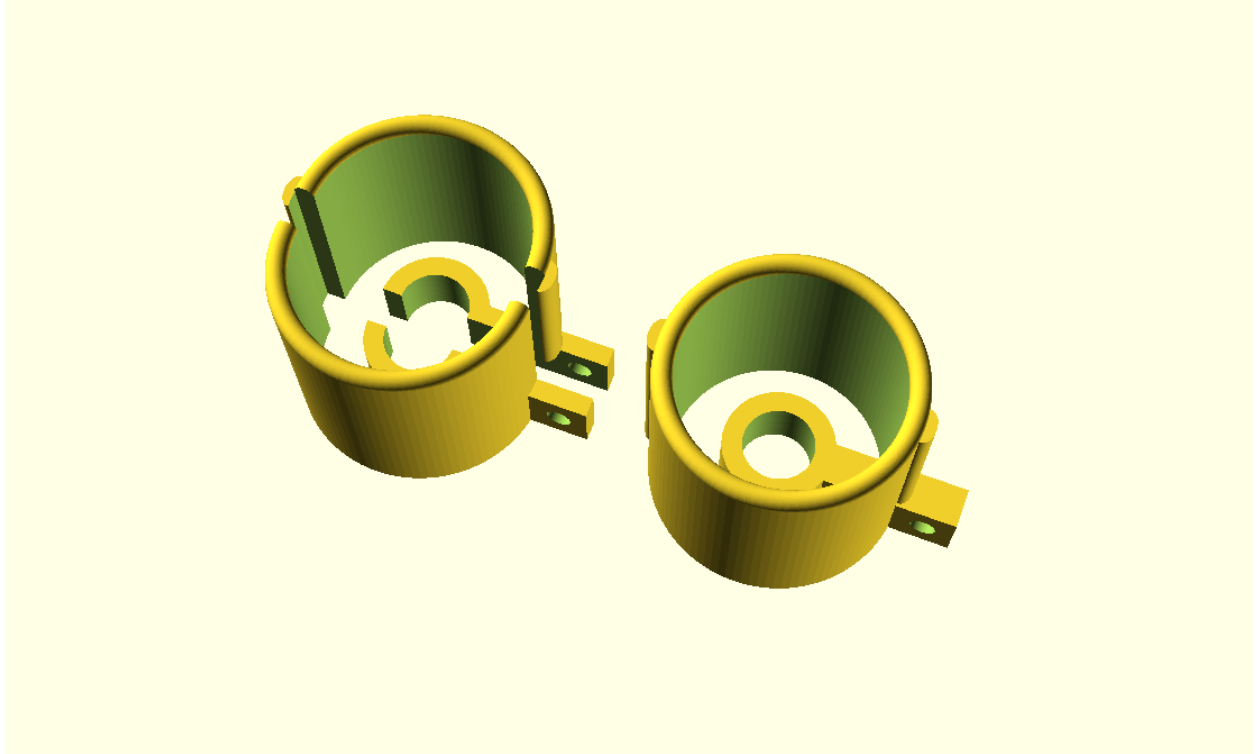
/*
Title: config.scad for R2B2 Nano
*/

d_screw_p=3.4,           //screw diameter
w_walls=2.7,             //width of the walls
w_support=8              //screw support width
h_support=8,             //screw support height
w_walls_m=1,             //width of the motor encapsulation walls
d_motor_t=d_motor+(w_walls_m*2), //motor diameter
h_motor_sup=8,           //motor support height
o_motor=0,               //motor support distance from ground
d_thruster=34,           //thruster diameter
h_thruster=30,           //thruster height

```

Paràmetre	Valor per defecte	Descripció
d_screw_h	3 mm	Diàmetre de la tija del cargol on ha de collar
w_walls	2.7 mm	Amplada de les parets
w_support	8 mm	Amplada dels suports del propulsors
h_support	8 mm	Profunditat dels suports del propulsors
w_walls_m	1 mm	Parets de la carcassa del motor
d_motor_t	d_motor+(w_walls_m*2) mm	Diàmetre total del motor (motor + carcassa)
h_motor_sup	8 mm	Alçada del cos del motor
o_motor	0 mm	Offset del suport respecte la base del propulsor
d_thruster	34 mm	Diàmetre del propulsor
h_thruster	30 mm	Alçada del propulsor

La carcassa del motor ens serveix per aïllar els motors de l'aigua. Els principals paràmetres que hem de tenir en

Fig. 5: Propulsor (**thruster.scad**).

compte son les mides dels motors que volem fer servir i l'amplada de les paret de la carcassa, que amb un valor petit ja serà suficient ja que no ha d'aguantar càrrega mecànica, tan sols aïllar de l'aigua. Per l'amplada de l'eix del motor sempre va bé deixar una mica més de la mida real, sino a l'hora del muntatge haurem de repassar els forats amb un trepant. I per a l'alçada de la carcassa deixar un milimetres extres respecte a l'alçada del motor per tal de poder-hi posar la cola tèrmica.

```
/*
Title: config.scad for R2B2 Nano
*/

w_walls_m=1,          //width of the walls
d_motor=7.6,          //motor diameter
h_motor_body=22.6,    //motor housing height
d_motor_shaft=1.5     //motor shaft diameter
```

Paràmetre	Valor per defecte	Descripció
w_walls_m	1 mm	Amplada de les parets de la carcassa
d_motor	7.6 mm	Diàmetre del motor
h_motor_body	22.6 mm	Alçada de l'encapsulat del motor
d_motor_shaft	1.5 mm	Amplada de l'eix del motor

En aquest objecte 3D s'ha tingut més en compte que s'ha de poder imprimir amb una impressora 3D que la seva eficiència. Per tant s'han fet unes hèlix no gaire primes i amb una forma molt recta per facilitar la impressió. No hi ha cap paràmetre destacable a part de escollir la quantitat d'aspes i la seva disposició ja que la majoria de paràmetres que l'afecten ja estan determinat per la mida del propulsor i del motor descrits anteriorment.

Fig. 6: Encapsulat del motor (**motor_housing.scad**).

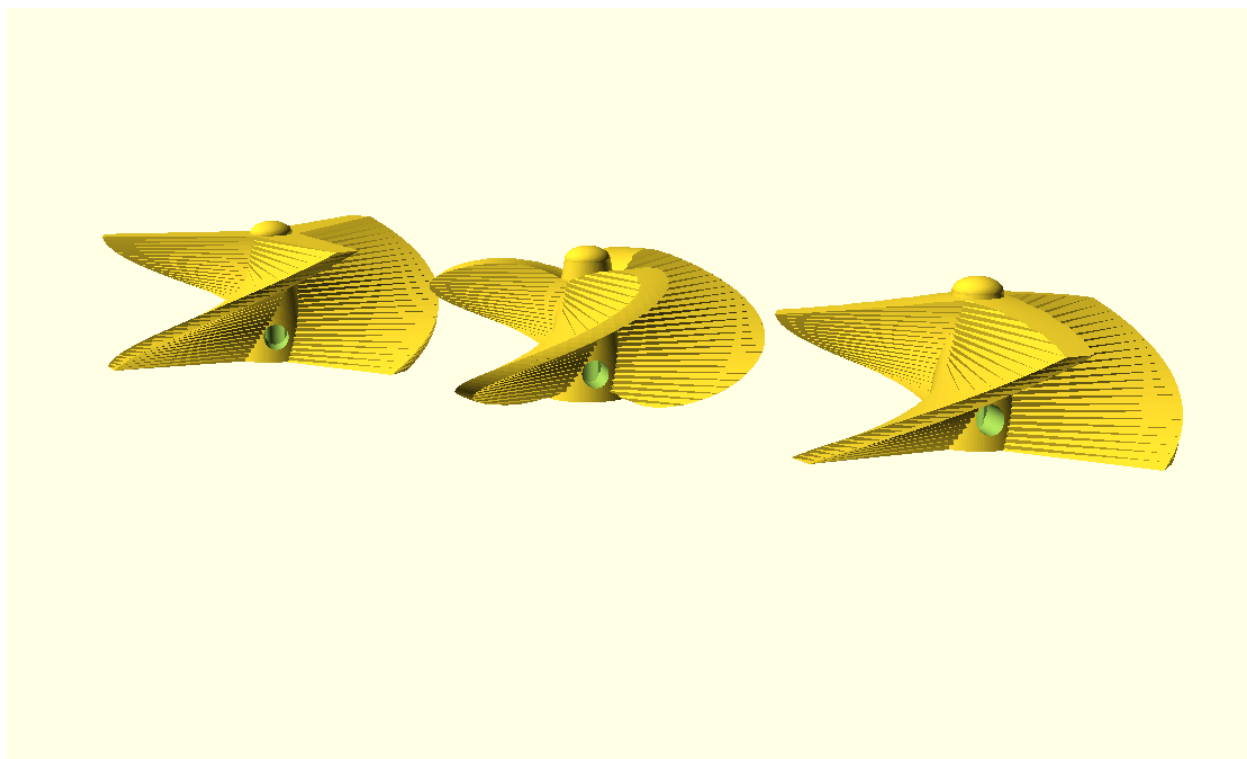
```

/*
Title: config.scad for R2B2 Nano
*/

w_walls=2.7,          //width of the walls
d_thruster=34,        //thruster diameter
h_propeller=10,       //propeller height
o_propeller=1,        //distance from propeller to walls of thruster
n_blade=3,            //number of blades
s_blade=20,           //angle separation between blades
w_blade=4,            //width of the blades
rounded=false,        //shape of the blades [round | trian | empty]
d_motor_shaft=1.5,    //motor shaft diameter
d_motor_grub=2        //motor grub screw diameter

```

Paràmetre	Valor per defecte	Descripció
w_walls	2.7 mm	Amplada de les parets
d_thruster	34 mm	Diàmetre del propulsor
h_propeller	10 mm	Alçada del propulsor
o_propeller	1 mm	Espai entre l'hèlix i les parets del propulsor
n_blade	3 mm	Nombre d'aspes a l'hèlix
s_blade	20 mm	Espai buit entre aspes en °
w_blade	4 mm	Amplada de les parets de les aspes
rounded	false	Forma de les aspes [round trian empty]
d_motor_shaft	1.5 mm	Amplada de l'eix del motor
d_motor_grub	2 mm	Diàmetre del cargol sense cap per collar l'hèlix

Fig. 7: Hèlix (`propeller.scad`).

2.3.4 2.3.4. Boia

Una de les majors complicacions que es presenten en un ROV són les comunicacions, degut a que l'aigua no és un medi amic de les altes freqüències que utilitzen el Wifi o el Bluetooth. Per això es deu que la majoria de ROVs duen un cable fins a la superfície, ja sigui per emetre les ones de comunicació com per connectar directament a la consola de control.

Com a paràmetres a la boia només hi ha les mides del cargol que utilitzarem per unir-ne les dues parts.

```
/*
Title: config.scad for R2B2 Nano
*/

d_screw_h=3,           //screw hole diameter
d_screw_head=6.5       //screw head diameter
```

Paràmetre	Valor per defecte	Descripció
d_screw_h	3 mm	Diàmetre de la tija del cargol on ha de collar
d_screw_head	6.5 mm	Diàmetre del cap del cargol

2.3.5 2.3.5. Carcassa

La carcassa **NO ÉS IMPRIMIBLE** en 3D!!! Està feta a mà amb espuma FOAM o alguna altra espuma que es pugui moldejar fàcilment amb un cúter, un trepant petit amb una fresa, llimes...

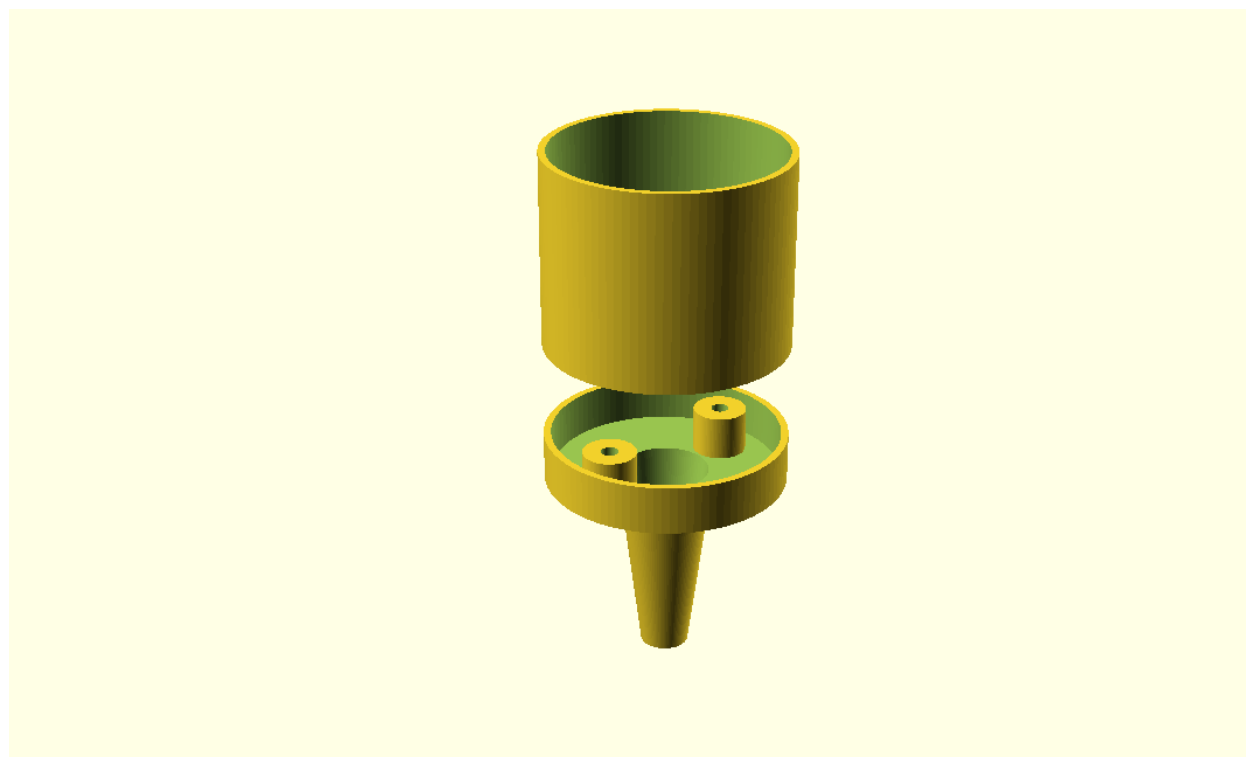


Fig. 8: Boia (**buoy.scad**).

La seva funció principal és aconseguir la flotabilitat zero de l'R2B2-nano, és a dir que un cop col·loquen la carcassa a l'R2B2-nano i el submergim sota l'aigua ni s'enfonsi ni floti cap a la superfície.

Nota: Sempre va bé deixar una mica de flotabilitat positiva per que l'R2B2-nano vagi cap a la superfície en el cas de que ens quedem sense bateria o es produeix algun error de comunicacions.

Per tant la carcassa de R2B2-nano es pot fer al gust i forma de cadascú tinguent en compte els principis d'Arquimedes. Si fem una forma de carcassa asimètrica, l'eix que es crea entre el punt de flotació i el punt de pes farà que l'R2B2-nano s'estabilitzi en una posició incòmode i de més difícil controlar.

Per altra banda si aconseguim alinear tan la flotació com el pes de l'R2B2-nano en una posició on ens queden tots els propulsors a la mateixa alçada, tindrem un R2B2-nano molt més estable i fàcil de controlar.

Una proposta de carcassa podria ser com la mostrada a continuació:

2.4 2.4. Generar STL

Per poder imprimir les peces en una impressora 3D convencional cal que els fitxers estiguin en un format compatible com pot ser STL. Des de l'OpenSCAD podem generar els fitxers STL un a un, però per agilitzar aquesta tasca hi ha un Script dins la carpeta design que recorre el contingut de la carpeta **/to_stl** i converteix cada fitxer .scad que hi troba a un STL. Els STLs generats es desen a la carpeta **/stl**.

<p>Atenció: Generar tots el fitxers pot requerir una espera llarga.</p>
--

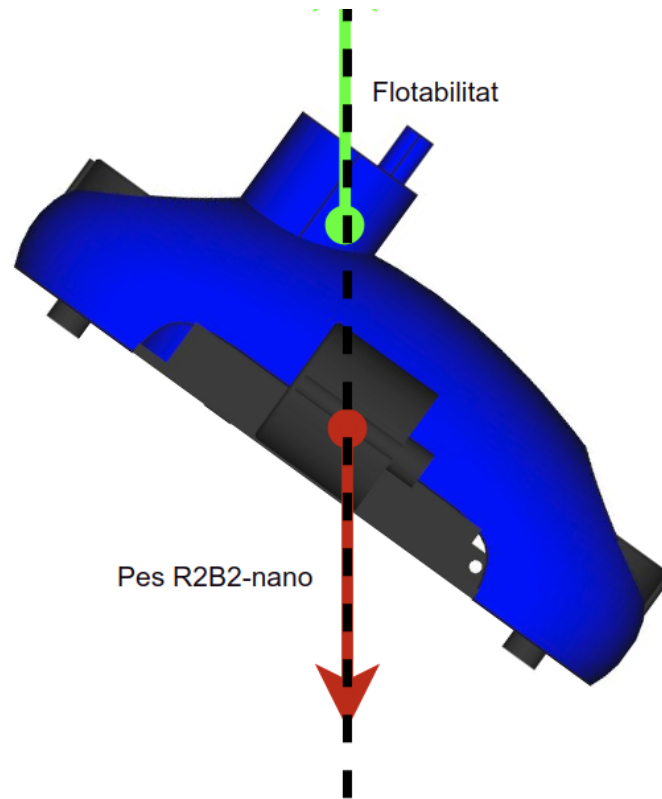


Fig. 9: Disseny mala carcassa.

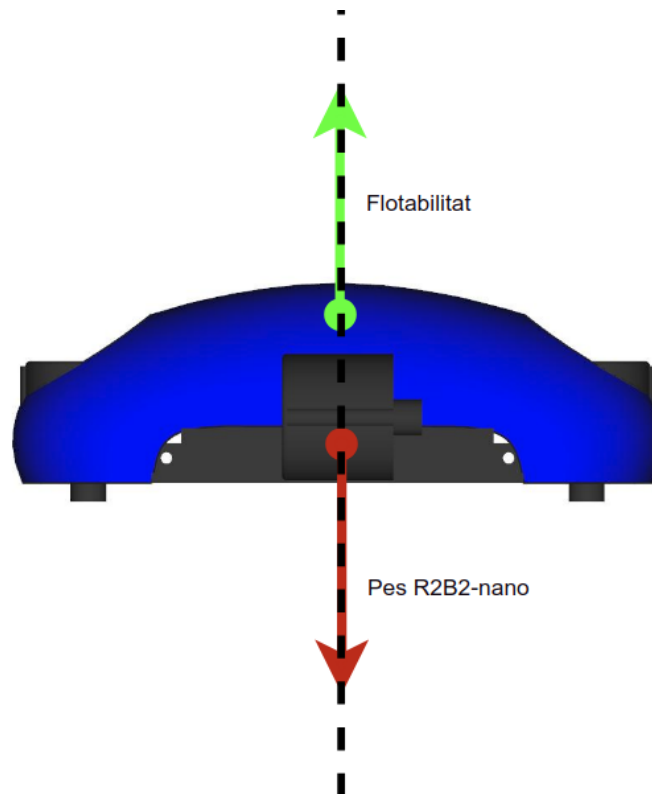


Fig. 10: Disseny carcassa correcte.

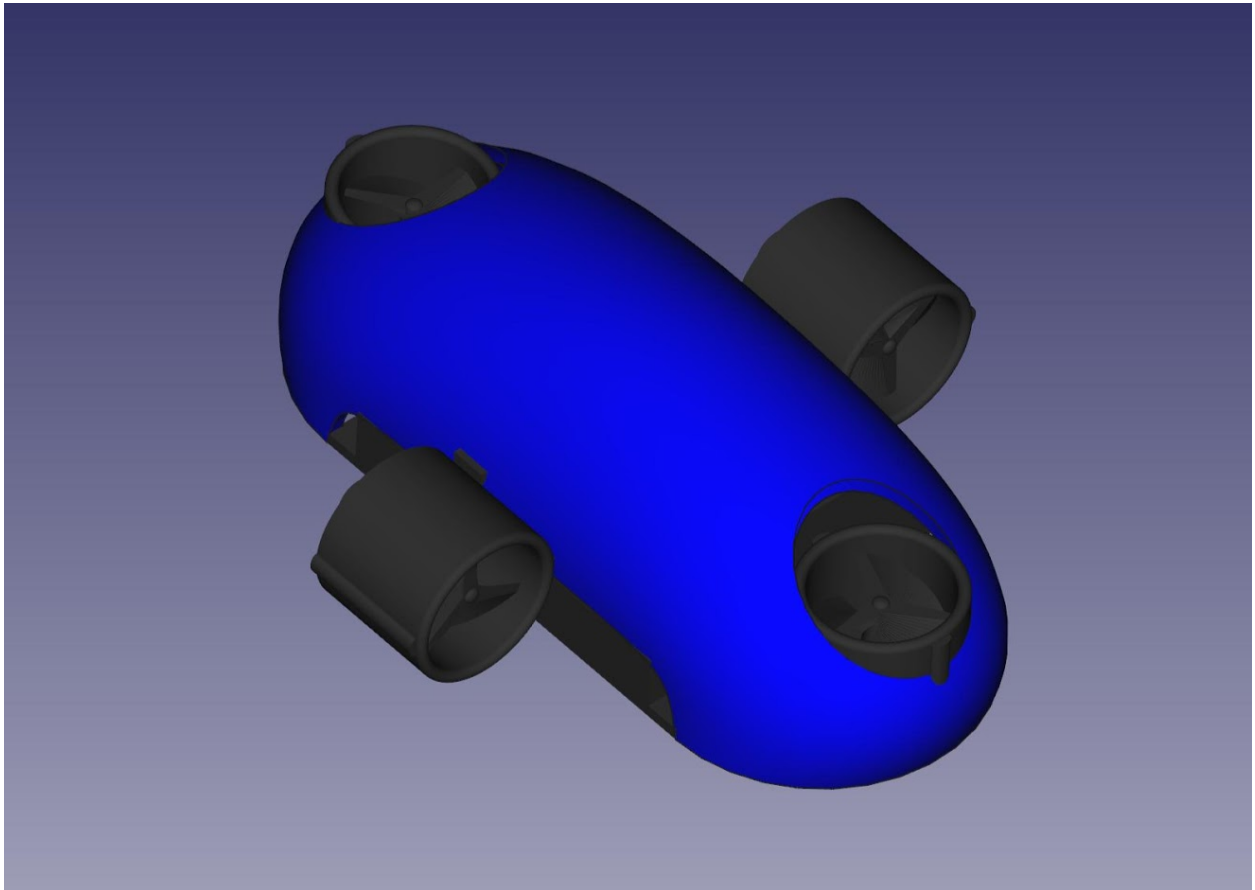


Fig. 11: Proposta de carcassa.

```
r2b2@r2b2os:/r2b2-nano/design$ ./generate_stl.sh
Converting base_b.scad to base_b.stl
Converting base.scad to base.stl
Converting buoy_down.scad to buoy_down.stl
Converting buoy_top.scad to buoy_top.stl
Converting chassis_b.scad to chassis_b.stl
Converting chassis.scad to chassis.stl
Converting half_thruster_A.scad to half_thruster_A.stl
Converting half_thruster_B.scad to half_thruster_B.stl
Converting motor_housing.scad to motor_housing.stl
Converting prog_protec.scad to prog_protec.stl
Converting propeller.scad to propeller.stl
Converting r2b2_nano.scad to r2b2_nano.stl
```

Per altra banda, si només ens interessa regenerar l'STL d'una sola peça, podem cridar l'script passant com a paràmetre el nom .scad del fitxer que volem convertir.

```
r2b2@r2b2os:/r2b2-nano/design$ ./generate_stl.sh to_stl/propeller.scad
```

3. Electrònica

L'R2B2-nano consta de dos mòduls. Un és l'encarregat de suplir corrent elèctrica, tan a motors com a components electrònics, i l'altre de controlar els dispositius i les comunicacions.

Els esquemes i PCB dels dos mòduls estan en un projecte públic de la web EasyEDA anomenat [R2B2-nano](#).

3.1 3.1. Alimentació

La base d'alimentació està formada per dues *bateries LIPO* de 3.7 volts connectades en sèrie arribant a un total de 7.4 volts. Cadascuna d'aquestes dues bateries està connectada a un *mòdul de càrrega TP4056* que a més a més de gestionar la càrrega les protegeix tan de sobrecàrregues com de infracàrregues. El mòdul de càrrega TP4056 disposa d'un port USB com a entrada de corrent, però per evitar connectors amb possible contacte amb l'aigua s'ha optat per posar un receptor de corrent sense fils (*Qi Receiver*).

Per ajustar la sortida de voltatge als requeriments del mòdul de control s'ha connectat un *Regulador de Voltatge* a la sortida de les dues bateries en sèrie.

3.1.1 3.2.1. Esquemàtics

Es pot trobar l'esquema electronic «r2b2-nano-power» al projecte públic [R2B2-nano](#) allotjat a EasyEDA.

3.1.2 3.2.2. Muntatge

Podeu veure com muntar aquest mòdul a l'apartat [Xassís Alimentació](#)

3.2 3.2. Control

El mòdul de control de l'R2B2-nano és l'encarregat de comunicar-se amb l'exterior i transmetre les ordres sol·licitades als diferents components. Té com a microcontrolador un *ESP32 WROVER* que incorpora WiFi/Bluetooth i a través

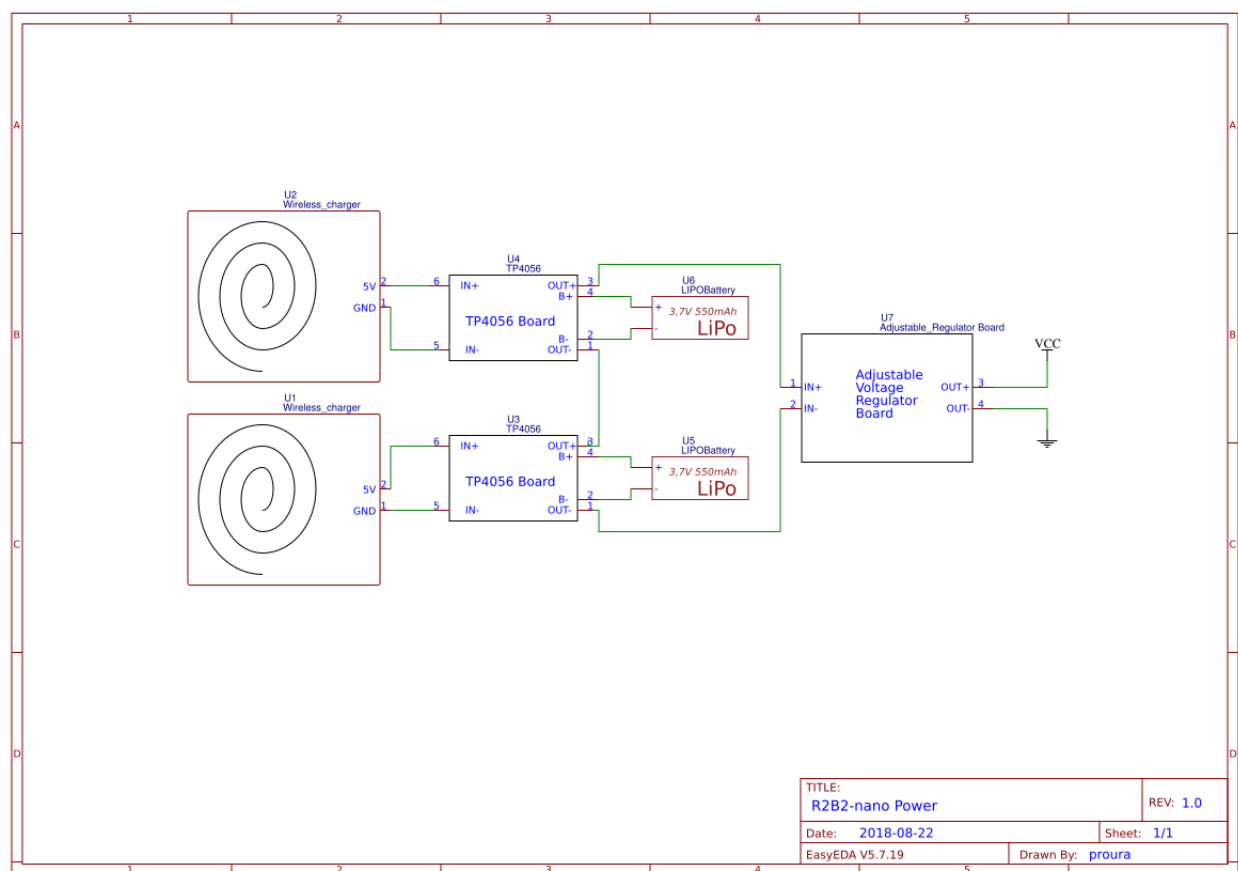


Fig. 1: Esquema del mòdul d'alimentació.

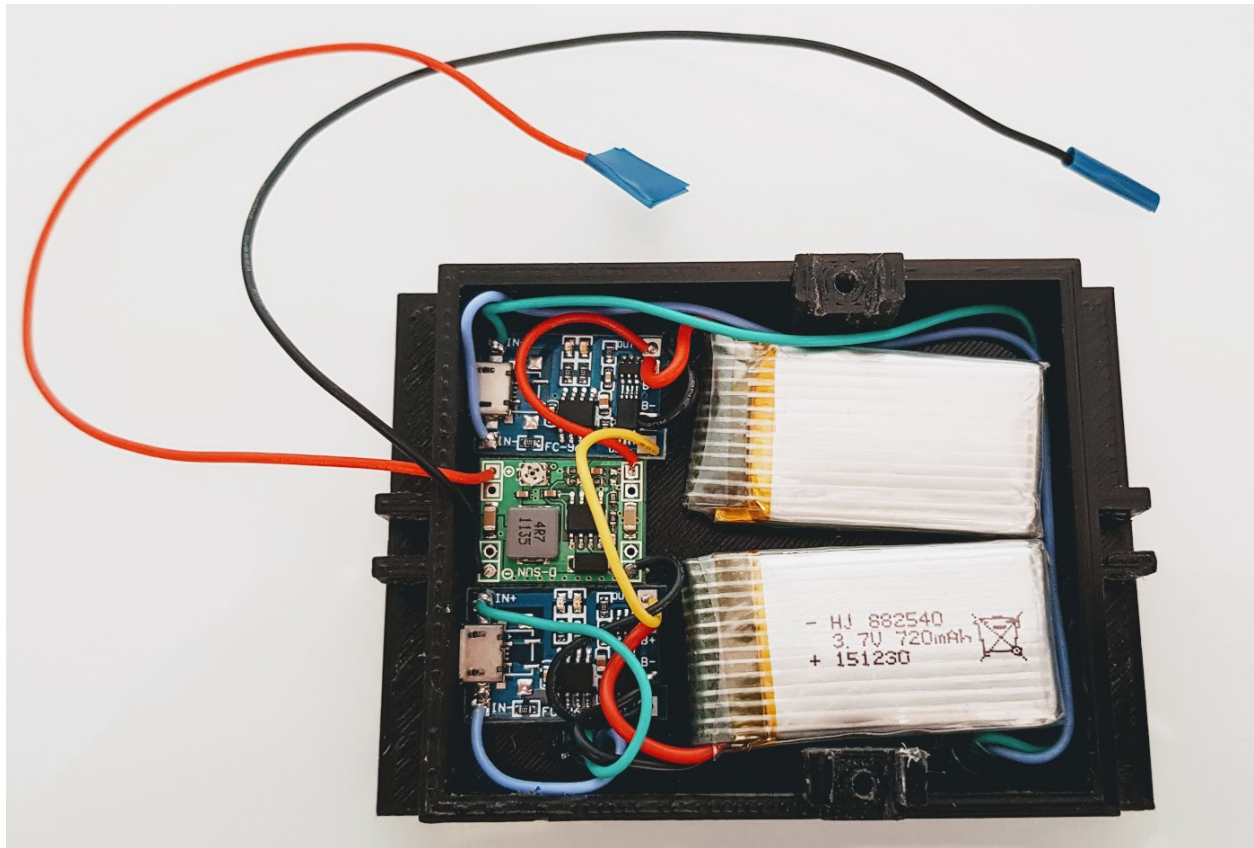


Fig. 2: Mòdul d'alimentació en construcció.

d'un connector coaxial ens permet treure un cable fins a la superfície per a realitzar les comunicacions.

Disposem de dos controladors de motors *TB6612FNG* que ens permet controlar a través de modulació per amplada de polsos (PWM) fins a 4 dispositius, en el nostre cas els motors dels propulsors.

També disposa d'un giroscopi i accelerometer *MPU 6050* que ens pot enviar informació del posicionament que té l'R2B2-nano en cada moment i realitzar les operacions que es creguin oportunes.

Tots els dispositius que hi ha al mòdul de control funcionen a 3.3 volts, per tant ens cal un *Regulador* per baixar el voltatge que ens proporciona el mòdul d'alimentació als 3.3 volts.

3.2.1 3.2.1. Esquemàtics

Es pot trobar l'esquema electronic «r2b2-nano-board» al projecte públic *R2B2-nano* allotjat a EasyEDA.

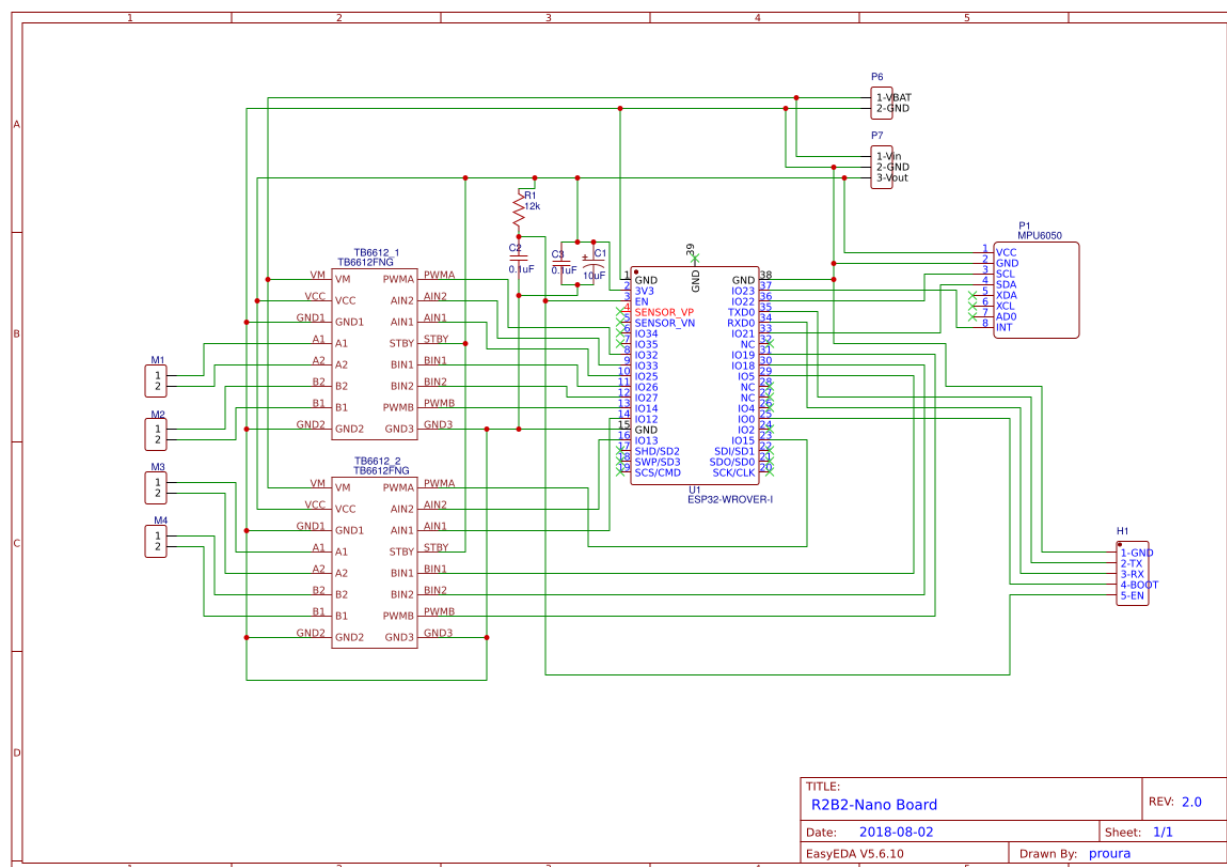


Fig. 3: Esquema del mòdul de control.

3.2.2 3.2.2. Muntatge

Per a muntar el mòdul de control hi ha varies opcions:

- Es pot cablejar sobre els mateixos components com es mostra a la següent imatge. Això ens pot servir per fer un prototip, però no es recomana ja que és molt fàcil cometre errors o que se'ns malmetin les soldadures al manipular la placa.

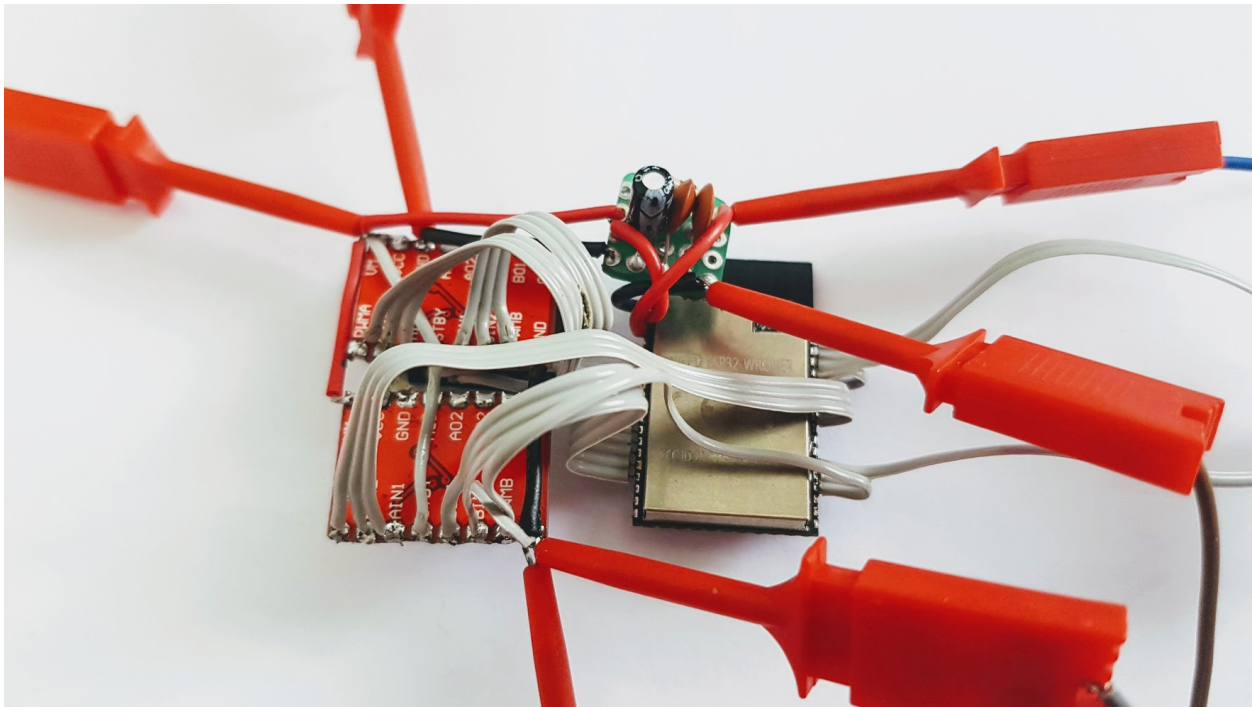


Fig. 4: Mòdul de control sense placa PCB.

- A la web del projecte R2B2-nano de EasyEDA hi ha disponible dos disseny de placa PCB una amb els components SMD (Surface-mount device) i l'altre amb components convencionals.

Podeu veure com muntar aquest mòdul a l'apartat [Xassís Control](#)

3.2.3 Connexions ESP32

Al l'esquema següent podem veure que hi ha connectat a cada port de l'EPS32 WROVER.

Relació entre els drivers de motors i els ports de la placa:

Driver	Sortida	PCB v0.1	PCB v0.2
TB6612FNG_1	Out A	P2	M1
TB6612FNG_1	Out B	P3	M2
TB6612FNG_2	Out A	P4	M3
TB6612FNG_2	Out B	P5	M4

3.3 Components

3.3.1 ESP32-WROVER-I (Espressif)

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC ultra-low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios.

- Link oficial a [Espressif ESP32](#).

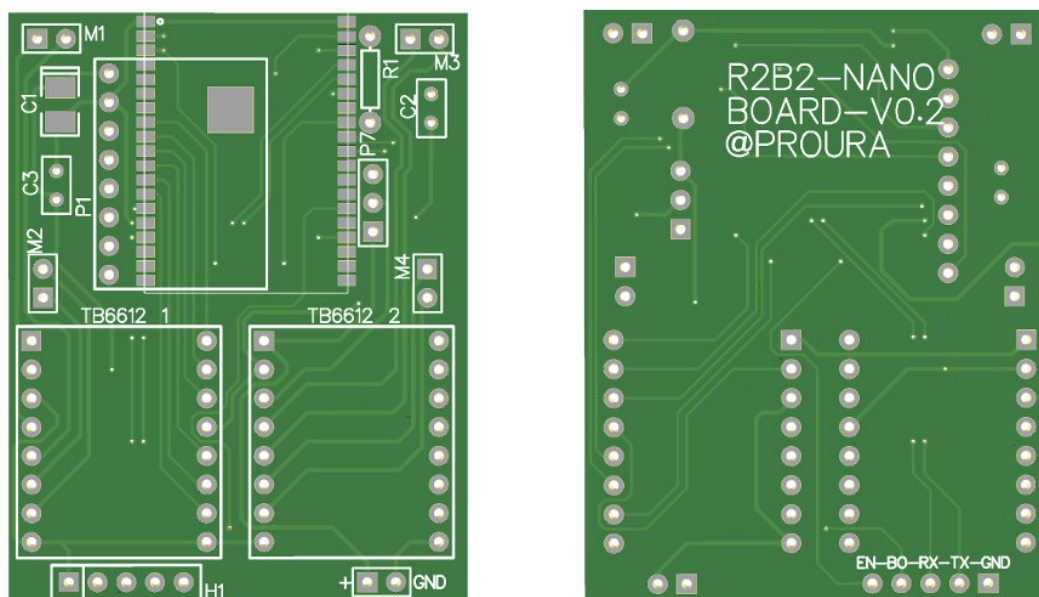


Fig. 5: Placa PCB del mòdul de control.

3.3.2 MPU-6050 (InvenSense)

The MPU-6050™ parts are the world's first MotionTracking devices designed for the low power, low cost, and high-performance requirements of smartphones, tablets and wearable sensors.

The MPU-6050 incorporates InvenSense's MotionFusion™ and run-time calibration firmware that enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices in motion-enabled products, guaranteeing that sensor fusion algorithms and calibration procedures deliver optimal performance for consumers.

The MPU-6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die, together with an onboard Digital Motion Processor™ (DMP™), which processes complex 6-axis MotionFusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master I²C bus, allowing the devices to gather a full set of sensor data without intervention from the system processor. The devices are offered in a 4 mm x 4 mm x 0.9 mm QFN package.

The InvenSense MotionApps™ Platform that comes with the MPU-6050 abstracts motion-based complexities, offloads sensor management from the operating system, and provides a structured set of APIs for application development.

For precision tracking of both fast and slow motions, the parts feature a user-programmable gyro full-scale range of ± 250 , ± 500 , ± 1000 , and ± 2000 °/sec (dps), and a user-programmable accelerometer full-scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$. Additional features include an embedded temperature sensor and an on-chip oscillator with $\pm 1\%$ variation over the operating temperature range.

- Link oficial a [MPU-6050](#).

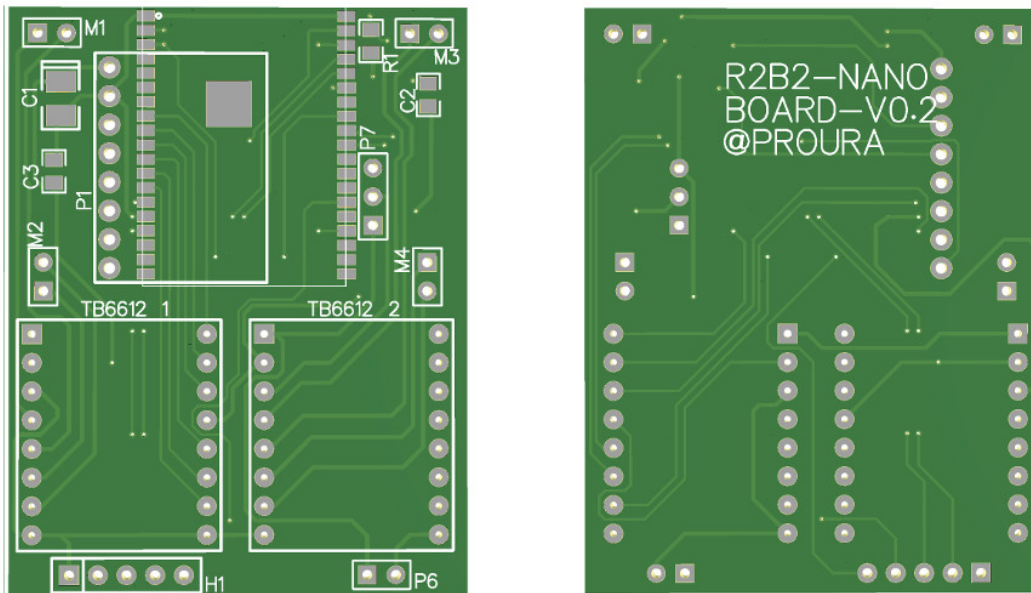


Fig. 6: Placa PCB del mòdul de control amb SMD (Surface-mount device).

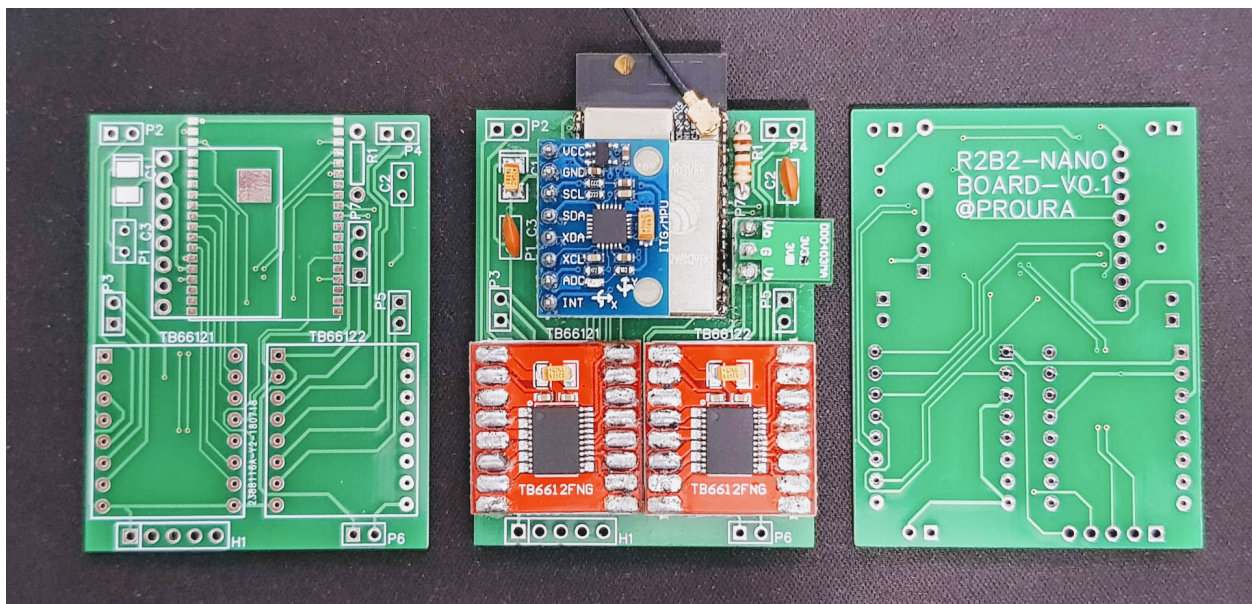


Fig. 7: Mòdul de control muntat.

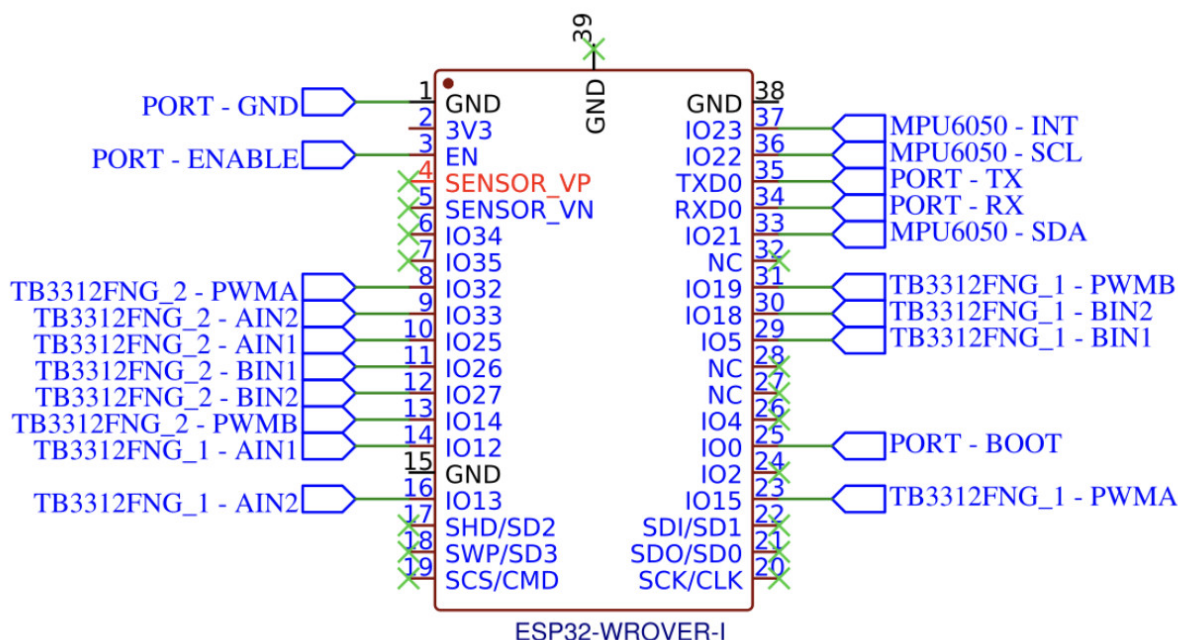


Fig. 8: Connexions de l'ESP32.

3.3.3 3.3.3. TB6612FNG (Thosiba)

TB6612FNG is a driver IC for DC motor with output transistor in LD MOS structure with low ON-resistor. Two input signals, IN1 and IN2, can choose one of four modes such as CW, CCW, short brake, and stop mode.

- [Link oficial TB6612FNG Toshiba.](#)

3.3.4 3.3.4. TP4056 & DW01A

The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. Furthermore, the TP4056 can work within USB and wall adapter.

The DW01A battery protection IC is designed to protect lithium-ion/polymer battery from damage or degrading the lifetime due to overcharge, overdischarge, and/or overcurrent for one-cell lithium-ion/polymer battery powered systems, such as cellular phones.

3.3.5 3.3.5. Cargador Qi i Receptor Qi

Qi és un estàndard obert d'interface desenvolupat pel consorci de potència sense fils (WPC) que té com a objectiu de transferir energia a una distància fins a 4cm. És usat per a carregar dispositius en múltiples aplicacions.

3.3.6 3.3.6. Bateria LiPo

Bateria d'ió liti en polímer o Bateria d'ió liti-polímer abreviat Li-Po és una bateria en què l'electròlit és una solució de sals de liti en un polímer sòlid o gelatinós, l'ànode és de carboni i el càtode és un òxid metàl·lic.



Fig. 9: ESP32 WROVER.

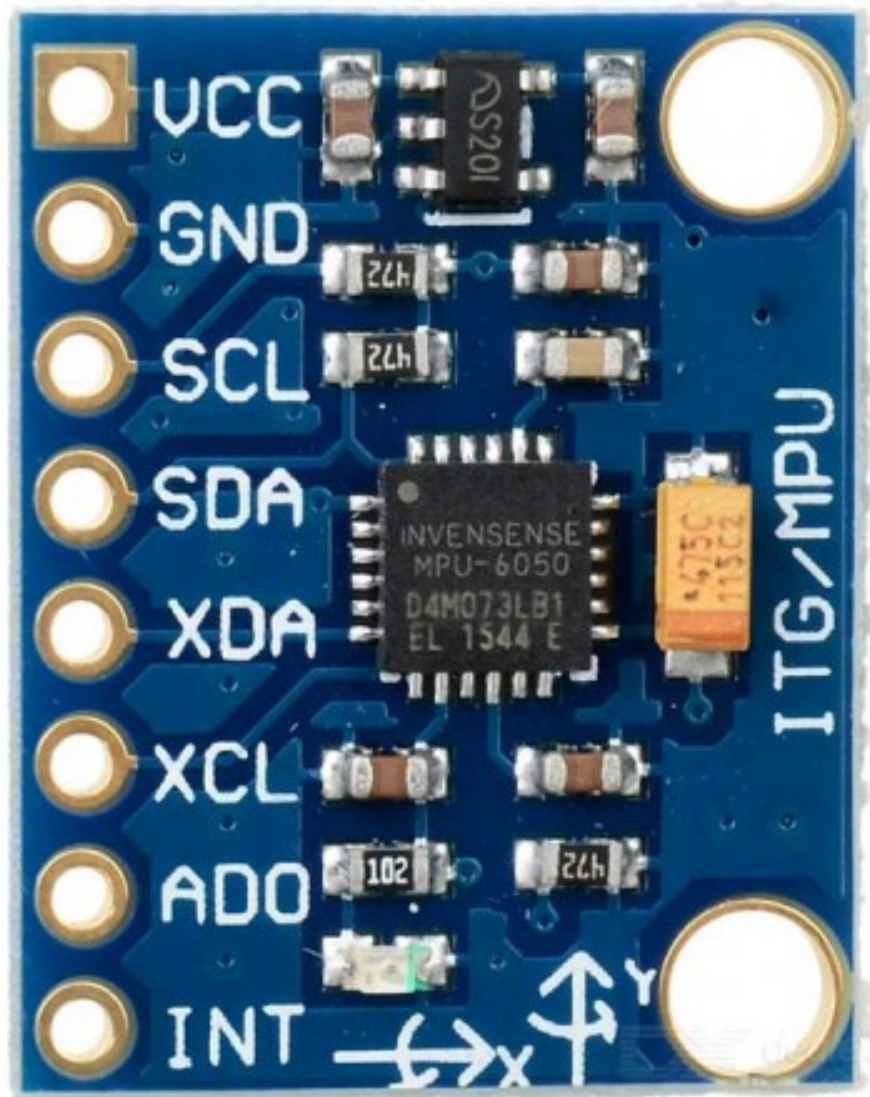


Fig. 10: Mòdul MPU-6050.



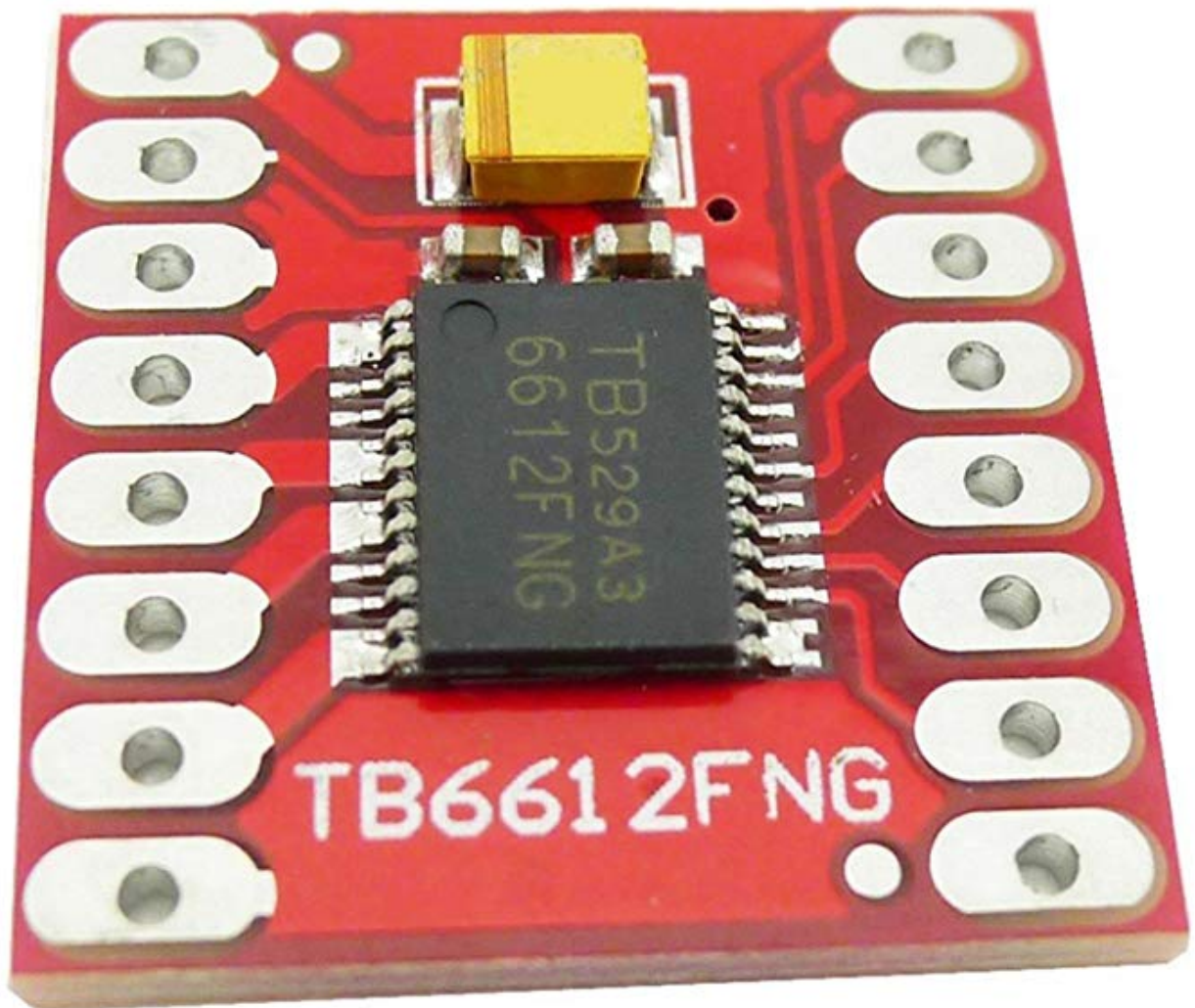


Fig. 11: Driver de motors TB6612FNG.

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

Fig. 12: TB6612FNG H-SW Control Function.

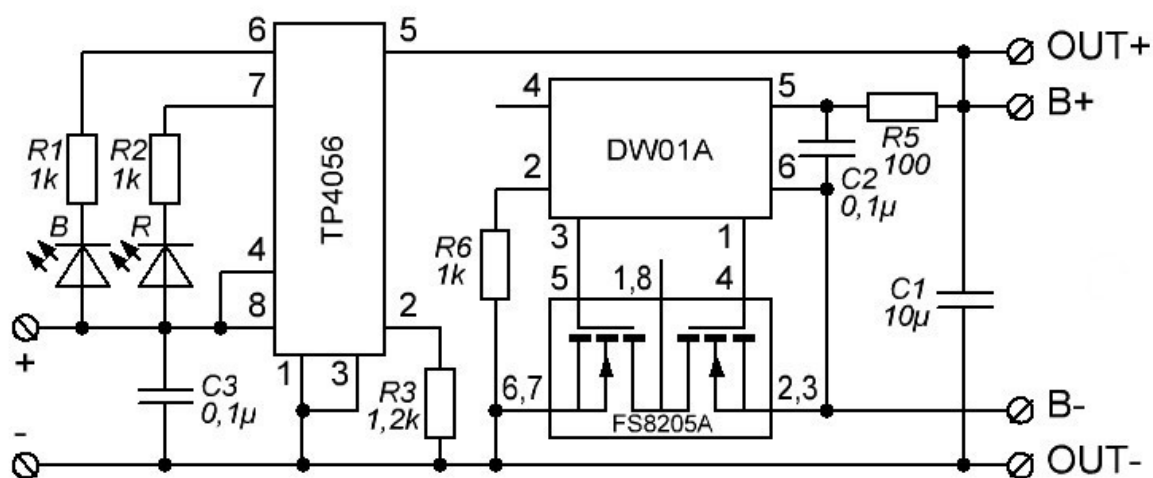


Fig. 13: Esquema elèctric del mòdul de càrrega.

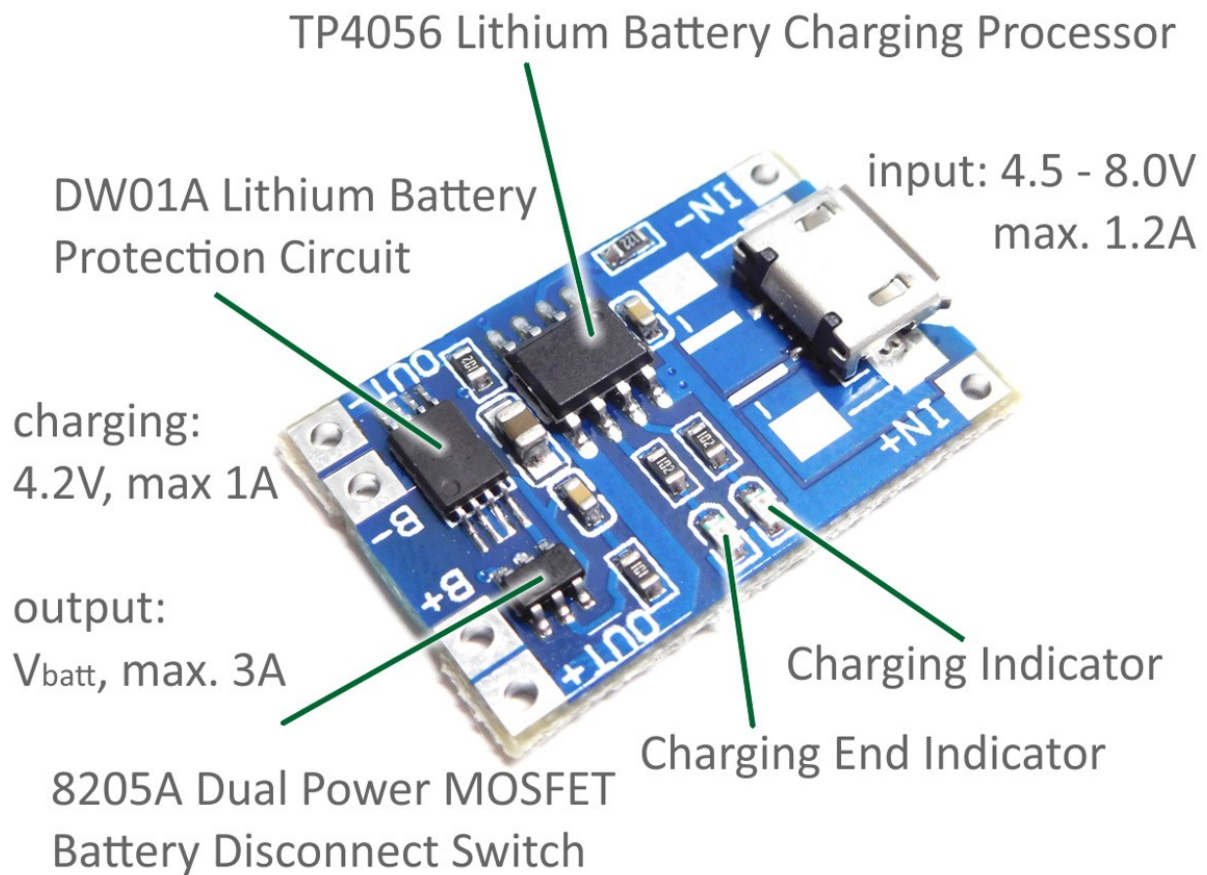


Fig. 14: Mòdul de càrrega TP4056.



Fig. 15: Receptor Qi.

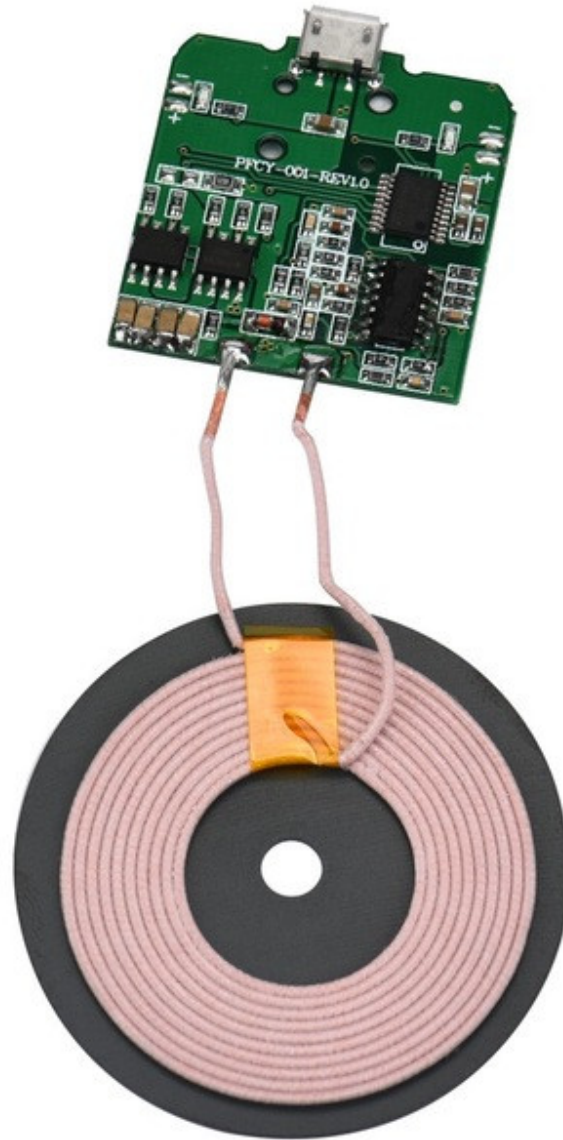


Fig. 16: Emissor Qi.

És una bateria recarregable que evolucionà a partir de les bateries d'ió liti on l'electròlit de sals de liti ja no es troba en una solució orgànica sinó en un polímer compost sòlid com ara el Poliacrylonitril.

Avís: Cal anar en compte amb la manipulació de les bateries LiPo!!! Ja que les bateries LiPo són força delicades. No s'han de deixar desateses mentre es carreguen. Una manipulació incorrecte pot fer que cremin!!!



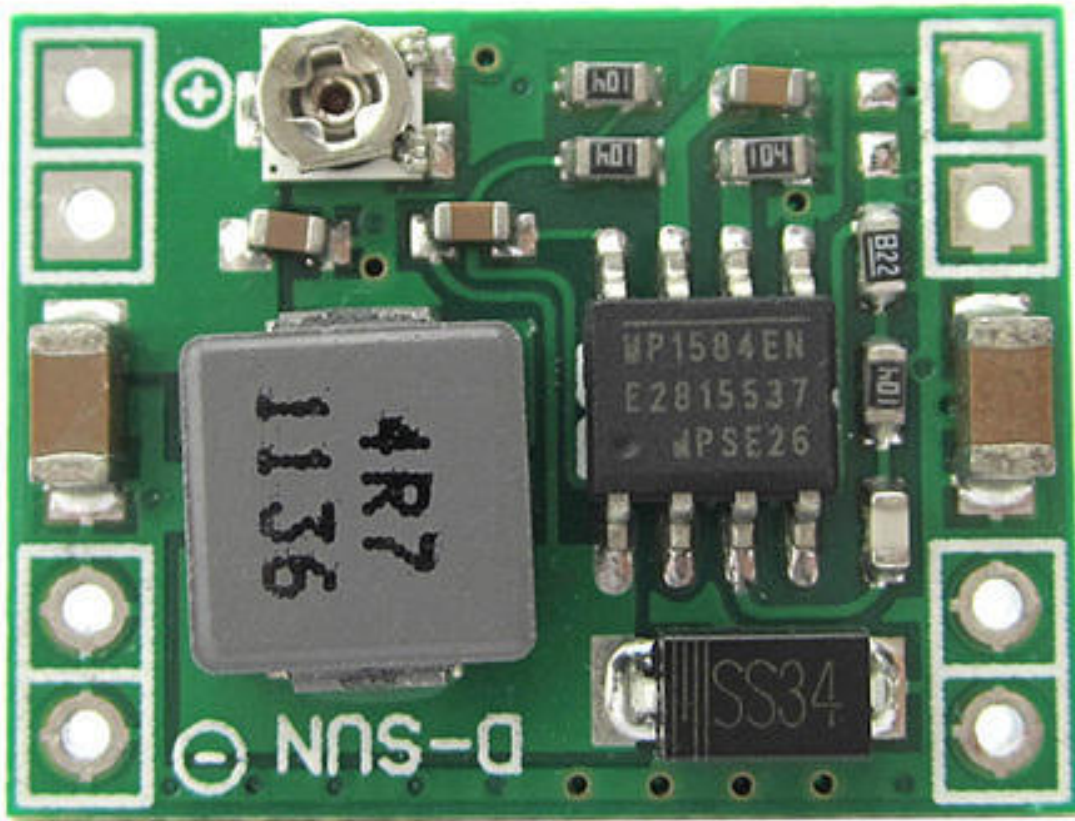


Fig. 17: Regulador IN: 4.5 - 28 volts → OUT: 0.8 -20 volts (3A. max.)

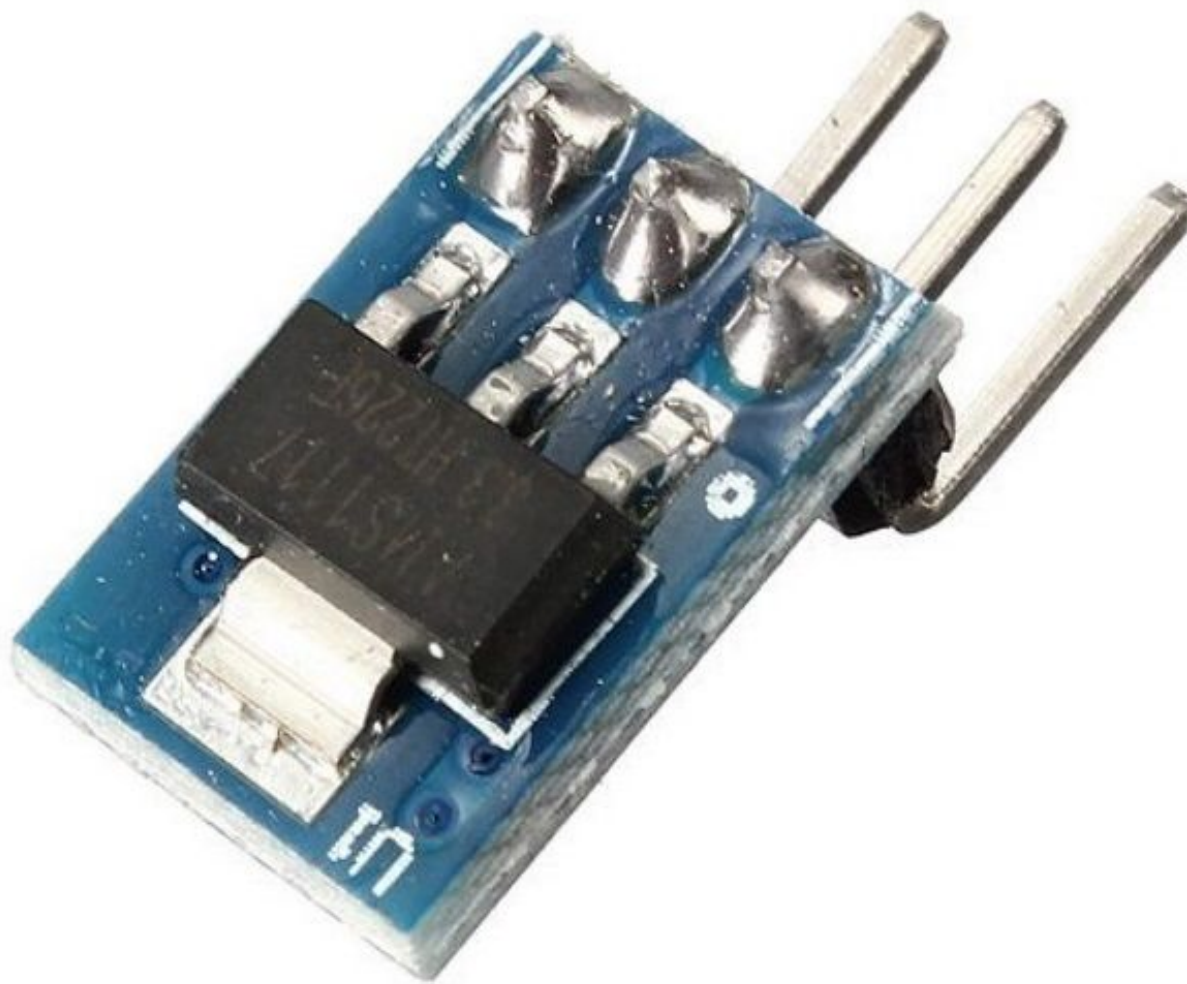


Fig. 18: Regulador OUT: 3.3v

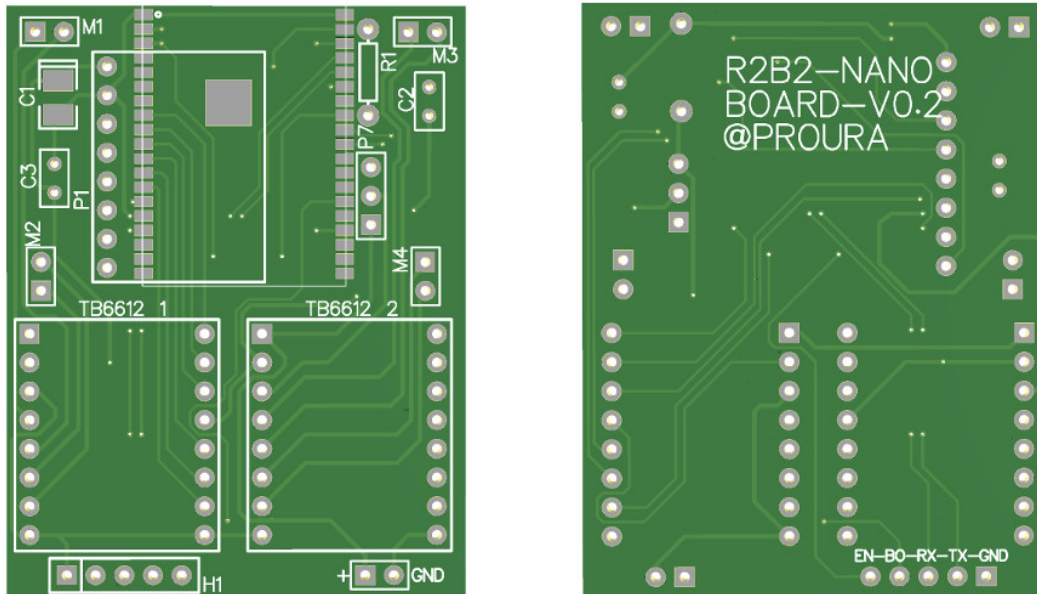


Fig. 19: Placa PCB del mòdul de control.

3.3.7 3.3.7. Reguladors de Voltatge

3.3.8 3.3.8. Placa PCB R2B2-nano

3.3.9 3.3.9. Connector IPEX/U.FL 20278 1.13mm

3.3.10 3.3.10 Motor de drone

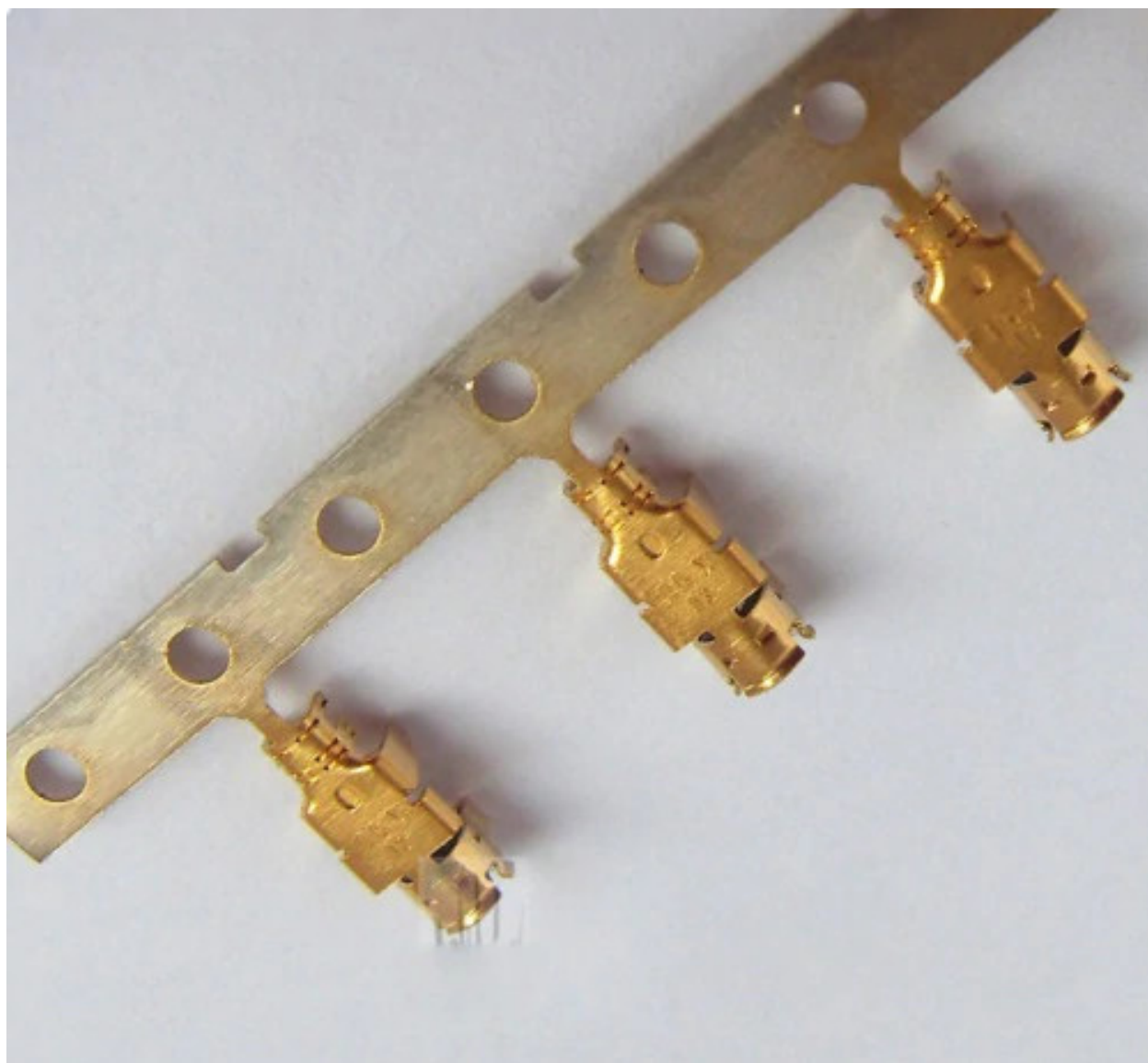


Fig. 20: Connector IPEX/U.FL 20278 1.13mm.

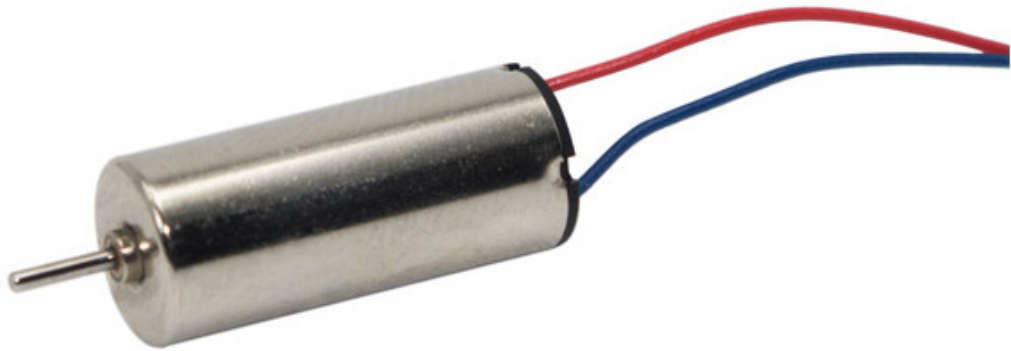


Fig. 21: Motors Drone (716 Coreless Motor).

4. Software

En aquest apartat es descriurà una proposta de com programar i utilitzar l'R2B2-nano, però cadascú pot realitzar o adaptar la proposta a les seves necessitats o tecnologies.

La programació està separada en tres grans apartats: el servidor de missatgeria MQTT que s'utilitza com a sistema de comunicació, el firmware de l'R2B2-nano fet amb Arduino IDE i el programa de control fet amb Processing IDE.

Nota: Els passos que es proposen en tota la documentació estan executats al terminal d'un sistema operatiu GNU/Linux **Debian** i amb el **mosquito client MQTT** instal·lat.

4.1 4.1. MQTT server

MQTT (Message Queuing Telemetry Transport) és un protocol de missatgeria publish-subscribe basat en el protocol TCP/IP.

Propietats de MQTT :

- Estàndard obert.
- Estructura senzilla (mínim nombre de bytes per cada missatge).

Acció	Prot. HTTP	Prot. MQTT	Relació HTTP/MQTT
Rebre 1 unitat de dades	320 bytes	69 bytes	4,6
Enviar 1 unitat de dades	320 bytes	47 bytes	6,8
Rebre 100 unitats de dades	12600 bytes	2445 bytes	5,1
Enviar 100 unitats de dades	14100 bytes	2126 bytes	6,6

- Fiabilitat (existeix la funció QoS Quality of Service), que ens informa de l'estat de la comunicació.
- Simplicitat (protocol que està definit en 43 pàgines).

En un protocol de publicació i subscripció pots realitzar les següents funcions:

- Publicar un missatge a un tòpic.
- Subscriure't a un tòpic i per tant rebre tot el que es publiqui en aquell tòpic.
- Anul·lar una subscripció i per tant deixar de rebre missatges d'aquell tòpic.

Cada R2B2-nano publica informació en un tòpic i aquest tòpic està format per una primera part on indica que és un R2B2, a continuació el seu número d'identificació i una petita informació que indiqui a que fa referència el contingut de les dades.

Exemples:

```
client.publish("r2b2/123456789/info", "Hello from R2B2-nano!"); --> Cada R2B2-nano_
↳ publica aquest missatge a l'arrencar.
client.publish("r2b2/123456789/mpu", "425 125 124 25 36 41"); --> Per enviar_
↳ informació dels valors de Giroscopi/Acceleròmetre.
client.publish("r2b2/123456789/tmp", "25.4568"); --> Per enviar la temperatura.
```

Llavors els dispositius que volen rebre la informació dels R2B2-nanos o d'un sol R2B2-nano tan sols s'ha de subscriure al tòpic que correspongui.

```
client.subscribe("r2b2/123456789/#"); --> Qui estigui subscript a aquest tòpic rebrà_
↳ tot el que publiqui l'R2B2-nano 123456789.
client.subscribe("r2b2/123456789/tasks"); --> Es rebran els missatges publicats a_
↳ tòpic r2b2/123456789/tasks.
```

Per tant ja tenim una manera de comunicar dispositius i R2B2-nanos. Cada R2B2-nano publica les dades amb el tòpic format pel seu propi ID («r2b2/r2b2Id/XXXX») i es subscriu al tòpic que li correspon a ell mateix r2b2/r2b2Id/tasks per rebre les ordres a executar i cada dispositiu que vulgui controlar un R2B2-nano en concret es subscriu al tòpic r2b2/r2b2Id/# i publica a r2b2/r2b2Id/tasks.

Daquesta manera també podem fer que un dispositiu controlï i/o observi més d'un R2B2-nano. Si es subscriu al tòpic «r2b2/#» rebrà la informació de tots els R2B2-nanos.

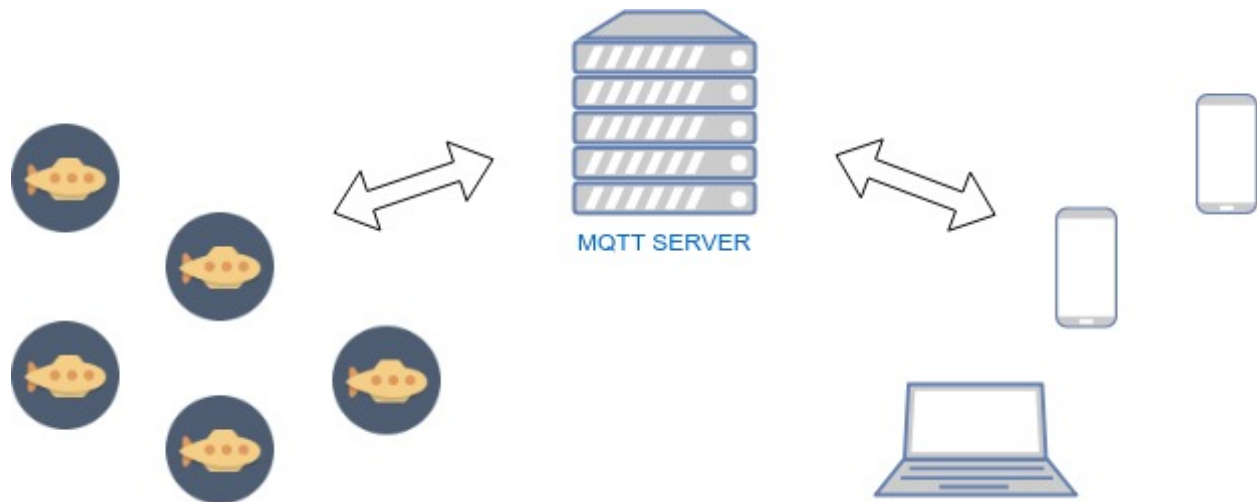


Fig. 1: Comunicacions R2B2-nano.

Des de la consola ens podem subscriure al servidor MQTT que estiguem fent servir i veure tots els missatges que s'estan intercanviant. Això pot ser molt útil per depurar les comunicacions entre R2B2-nanos i els programes de control.

```
r2b2@mqttsrv:~$ mosquitto_sub -h 192.168.82.106 -t "#" -v
r2b2/3291342516/info Hello from R2B2-nano!
r2b2/3291342516/info r2b2-3291342516
r2b2/3291342516/mpu -116 -32 22944 244 204 23
r2b2/3291342516/tasks mFF 50
r2b2/3291342516/tasks mFS
r2b2/3291342516/tasks mBF 100
r2b2/3291342516/tasks mBS
```

També podem publicar missatges des de la consola per mirar si el Firmware de l'R2B2-nano respon correctament a l'ordre llançada sense necessitat de fer ús de cap programa de control.

```
r2b2@mqttsrv:~$ mosquitto_pub -h 192.168.82.106 -t "r2b2/3291342516/tasks" -m
↪ "getGyrAcc"
r2b2@mqttsrv:~$ mosquitto_pub -h 192.168.82.106 -t "r2b2/3291342516/tasks" -m "mFF 50"
r2b2@mqttsrv:~$ mosquitto_pub -h 192.168.82.106 -t "r2b2/3291342516/tasks" -m "mFS"
r2b2@mqttsrv:~$ mosquitto_pub -h 192.168.82.106 -t "r2b2/3291342516/tasks" -m "mBF 100"
↪ "
r2b2@mqttsrv:~$ mosquitto_pub -h 192.168.82.106 -t "r2b2/3291342516/tasks" -m "mFS"
r2b2@mqttsrv:~$ mosquitto_pub -h 192.168.82.106 -t "r2b2/3291342516/tasks" -m "mBS"
```

4.2. Firmware

4.2.1. Arduino IDE

El firmware de l'R2B2-nano està desenvolupat amb l'[Arduino IDE](#).

Per poder compilar el firmware caldrà afegir la placa ESP32 seguint les instruccions oficials del [GitHub d'espressif](#) i instal·lar les llibreries **PubSubClient** i **MPU6050_tockn** a través del menú Sketch->Include Library->Manage Libraries.

La llibreria PubSubClients és per publicar i subscriure en un servidor MQTT i la llibreria MPU6050_tockn per llegir els valors de Giroscopi i acceleròmetre.

Un cop fet això ja podem obrir el projecte 99_R2B2_Nano_Firmware d'Arduino IDE que trobarem a la carpeta codi del projecte R2B2-nano, configurar la placa com a ESP32 Dev Module i seleccionar el Port on estigui connectat.

Avís: El primer cop que es puja el firmware s'ha de fer amb un conversor de USB a TTY connectant el port RX de R2B2-nano al TX del conversor TTY, el port TX de R2B2-nano al RX del conversor TTY, connectar el GND de l'R2B2-nano amb el GND del conversor TTY i posant la placa en mode «serial bootloader», això es fa posant el port GPIO0 (BOOT) a LOW/GND i fent un reset posant l'enable (EN) a LOW/GND. Un cop feta la primera pujada ens apareixerà l'R2B2 al llistat de ports i podrem actualitzar el firmware a través de WiFi.

4.2.2. Codi del Firmware

4.2.2.1. Configuració

Cal configurar 4 variables abans de pujar el firmware a l'R2B2-nano. La variable **ssid** que defineix a quina WiFi connectar, **password** per la contrasenya de la WiFi, a la variable **mqttServer** hi hem d'especificar la direcció IP del servidor MQTT i per finalitzar **mqttPort** per definir a través de quin port ha de realitzar la connexió MQTT.

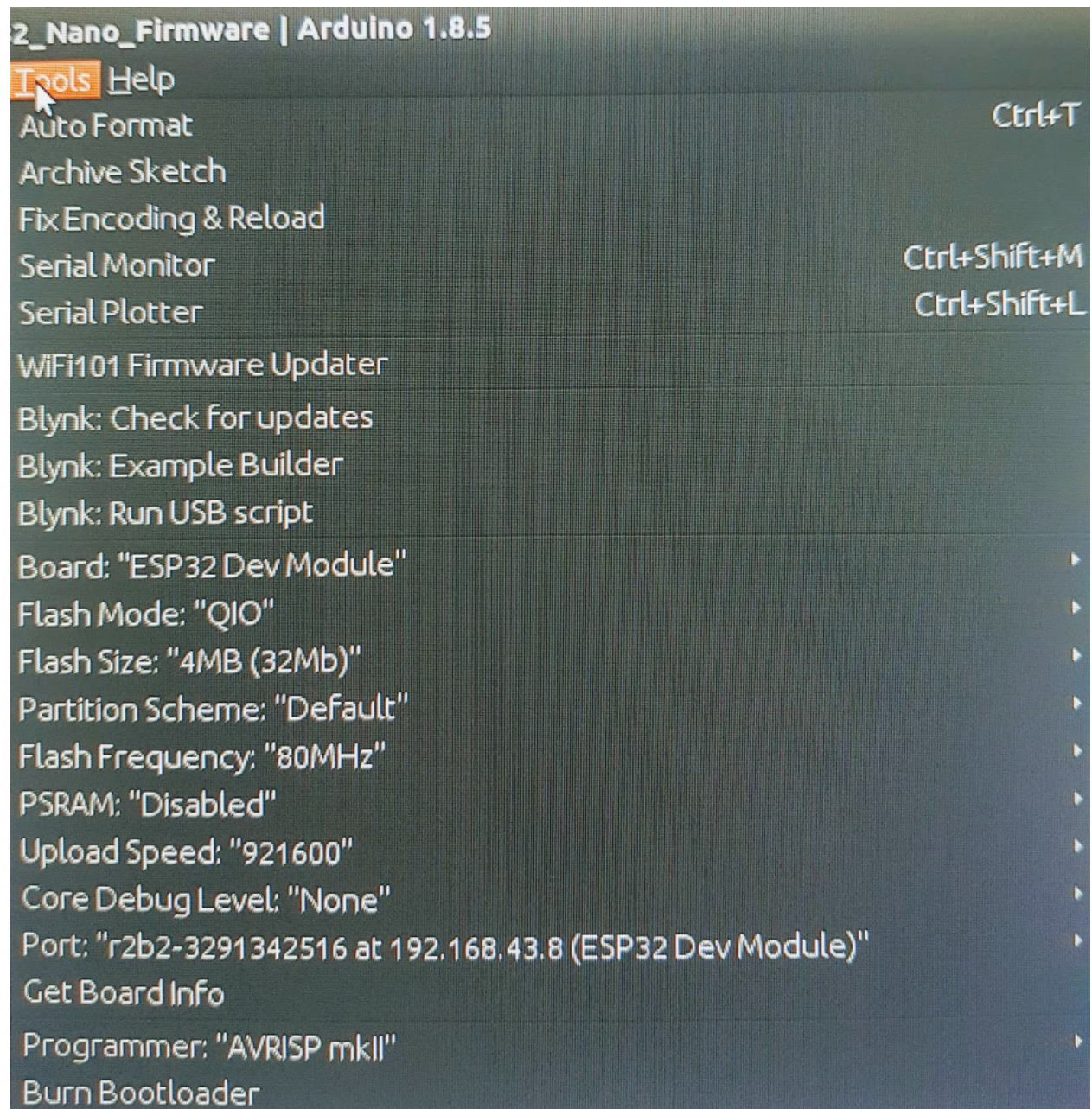


Fig. 2: Configuració Arduino IDE.

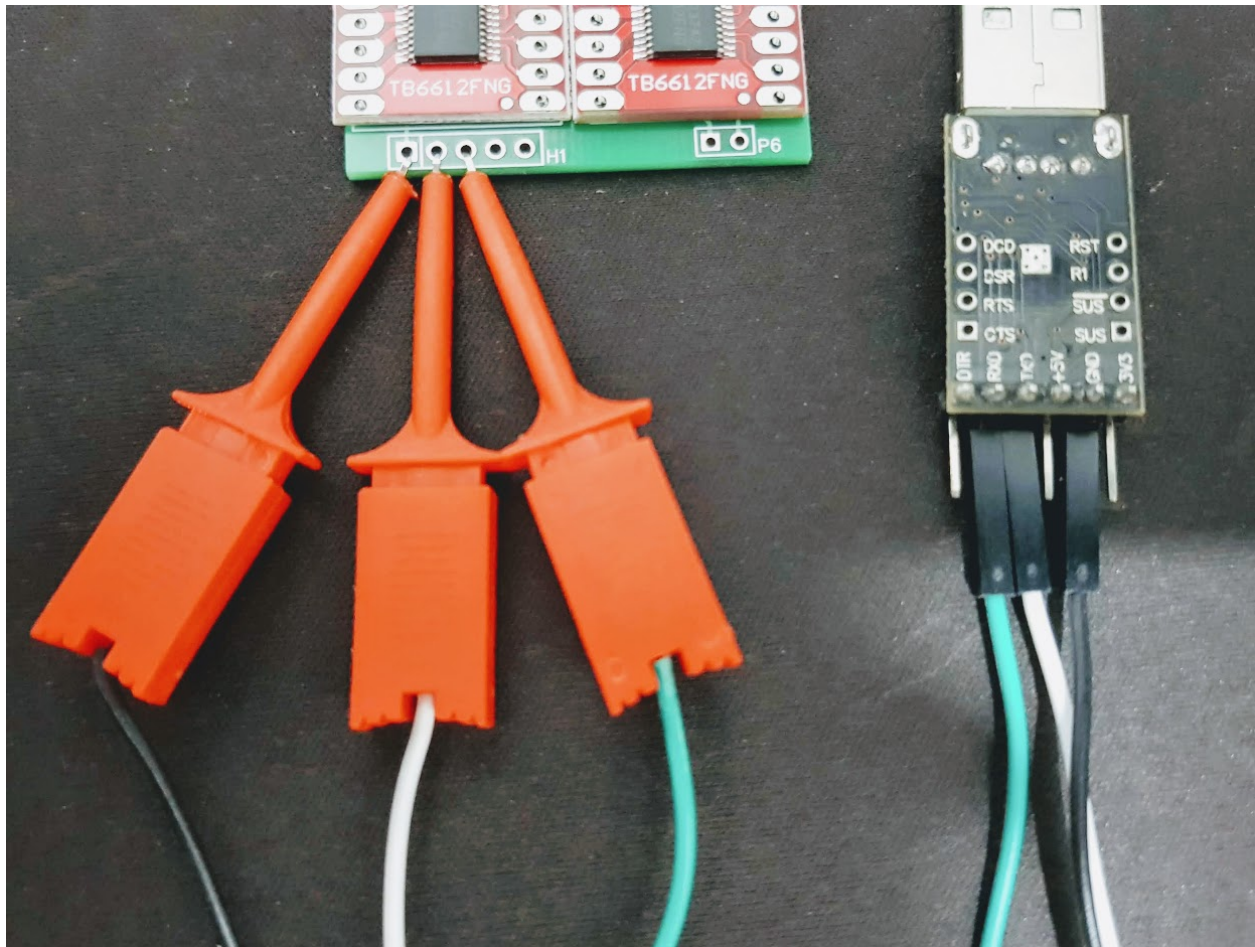


Fig. 3: R2B2-nano connectat per USB.

```
...
const char* ssid = "BuLan";
const char* password = "00009999";
const char* mqttServer = "192.168.82.106";
const int    mqttPort = 1883;
...
```

Unes altres variables que cal tindre en compte son els pins on estan connectats els motors. Pot ser que durant el muntatge invertim la polaritat d'algun motors i per tant a l'hora de configurar els pins haurem d'invertir-los també.

```
...
//Configuració del PWM pels 4 motors
const int freq = 2000;
const int resolution = 8;
const int MotorChannelR = 0;
const int MotorChannelL = 1;
const int MotorChannelF = 2;
const int MotorChannelB = 3;
...

//Pins del motor Dret
const int motorR_IN1 = 33;
const int motorR_IN2 = 25;
const int motorR_PWM = 32;

//Pins del motor Esquerra
const int motorL_IN1 = 5;
const int motorL_IN2 = 18;
const int motorL_PWM = 19;

//Pins del motor Frontal
const int motorF_IN1 = 26;
const int motorF_IN2 = 27;
const int motorF_PWM = 14;

//Pins del motor Posterior
const int motorB_IN1 = 12;
const int motorB_IN2 = 13;
const int motorB_PWM = 15;
...
```

4.2.2.2. OTA Over-the-Air

Per que poguem actualitzar el firmware a través de WiFi cal que sempre hi hagi els següent talls de codi a l'Sketch d'Arduino. Si no ho posem o fem una mala configuració haurem d'actualitzar el firmware l'R2B2-nano amb el conversor USB to TTY.

```
//Dins a SETUP
...
initializeOTA((char*)R2B2id);
...

//Dins a LOOP
...
ArduinoOTA.handle();
...
```

(continues on next page)

(ve de la pàgina anterior)

```
//FUNCIÓ

void initializeOTA(const char * host_name){
    sprintf(host, "r2b2-%u", host_name);
    ArduinoOTA.setHostname(host);
    ArduinoOTA
        .onStart([]() {
            String type;
            if (ArduinoOTA.getCommand() == U_FLASH) type = "sketch";
            else type = "filesystem";
            Serial.println("Start updating " + type);
        })
        .onEnd([]() {
            Serial.println("\nEnd");
        })
        .onProgress([](unsigned int progress, unsigned int total) {
            Serial.printf("Progress: %u%%\r", (progress / (total / 100)));

            //Send Upload Info to MQTT
            if((millis() - timer > 250) || ((progress / (total / 100)) == 100 )){
                sprintf(msg, "upload %u%%\r", (progress / (total / 100)));
                sprintf(topic, "r2b2/%u/info", R2B2id);
                client.publish(topic, msg);
                timer = millis();
            }

        })
        .onError([](ota_error_t error) {
            Serial.printf("Error[%u]: ", error);
            if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
            else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
            else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
            else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
            else if (error == OTA_END_ERROR) Serial.println("End Failed");
        });

    ArduinoOTA.begin();
}
```

4.2.2.3. Estructura del codi

En el **setup** del codi inicialitzarem els motors, el WiFi, l'OTA, la connexió al servidor MQTT i el Giroscopi/Acceleròmetre. Llavors dins al **loop** estarem pendents de si arriba alguna petició d'actualització de firmware o algun missatge MQTT.

```
//Dins a SETUP
...
initializeMotors();
initializeWifi();
initializeOTA((char*)R2B2id);
initializeMQTT();
initializeMPU();
...
```

(continues on next page)

(ve de la pàgina anterior)

```
//Dins a LOOP
...
ArduinoOTA.handle();
client.loop();
...
```

A l'**inicialització del motors** crearem quatre canals PWM amb una freq i resolució i els assignarem als motors.

```
void initializeMotors() {
    ledcSetup(MotorChannelR, freq, resolution);
    ledcSetup(MotorChannelL, freq, resolution);
    ledcSetup(MotorChannelF, freq, resolution);
    ledcSetup(MotorChannelB, freq, resolution);

    ledcAttachPin(motorR_PWM, MotorChannelR);
    ledcAttachPin(motorL_PWM, MotorChannelL);
    ledcAttachPin(motorF_PWM, MotorChannelF);
    ledcAttachPin(motorB_PWM, MotorChannelB);

    ledcWrite(MotorChannelR, 0);
    ledcWrite(MotorChannelL, 0);
    ledcWrite(MotorChannelF, 0);
    ledcWrite(MotorChannelB, 0);

    pinMode (motorR_IN1, OUTPUT);
    pinMode (motorR_IN2, OUTPUT);
    pinMode (motorL_IN1, OUTPUT);
    pinMode (motorL_IN2, OUTPUT);
    pinMode (motorF_IN1, OUTPUT);
    pinMode (motorF_IN2, OUTPUT);
    pinMode (motorB_IN1, OUTPUT);
    pinMode (motorB_IN2, OUTPUT);

    digitalWrite (motorR_IN1, LOW);
    digitalWrite (motorR_IN2, LOW);
    digitalWrite (motorL_IN1, LOW);
    digitalWrite (motorL_IN2, LOW);
    digitalWrite (motorF_IN1, LOW);
    digitalWrite (motorF_IN2, LOW);
    digitalWrite (motorB_IN1, LOW);
    digitalWrite (motorB_IN2, LOW);
}
```

A la **inicialització del WiFi** assignarem una funció per caçar els events de canvi d'estat del WiFi i així poder detectar quan perdem la connexió. Llavors inicialitzam la connexió amb **ssid** i **password** definits.

```
void initializeWifi() {
    WiFi.mode(WIFI_STA);
    WiFi.onEvent(WiFiEvent);
    WiFi.begin(ssid, password);
    Serial.println("Connecting WiFi...");
    while (WiFi.waitForConnectResult() != WL_CONNECTED) {
        Serial.println("Connection Failed! Rebooting...");
        delay(5000);
        ESP.restart();
    }
}
```

(continues on next page)

(ve de la pàgina anterior)

```

}

void WiFiEvent(WiFiEvent_t event){
  switch(event) {
    case SYSTEM_EVENT_STA_GOT_IP:
      Serial.print("WiFi connected! IP address: ");
      Serial.println(WiFi.localIP());
      connected = true;
      break;
    case SYSTEM_EVENT_STA_DISCONNECTED:
      Serial.println("WiFi lost connection");
      connected = false;
      break;
  }
}

```

Quan **inicialitzem l'MQTT**, a més a més d'indicar a quin servidor i port connectar, assignem una **Callback Function**. Aquesta funció és la que s'executarà quan es rep un missatge. La funció de **Callback** rep com a paràmetres el tòpic, el bytes d'informació i la longitud de la trama i en funció del missatge que rebí executarà una acció o una altre.

```

void initializeMQTT(){
  client.setServer(mqttServer, mqttPort);
  client.setCallback(callback);
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i=0;i<length;i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();

  String str = String((char *)payload);
  str = str.substring(0, length);

  if ( str == "ping" ) pong();
  if ( str == "calibrate" ) calibrate();
  if ( str == "getGyrAcc" ) getGyrAcc();
  if ( str == "getTemp" ) getTemp();
  if ( str == "stop" ) motorStop();

  if ( str.substring(0, 3) == "mRF" ) motorRForward(str.substring(4, length).
↳toInt());
  if ( str.substring(0, 3) == "mRB" ) motorRBackward(str.substring(4, length).
↳toInt());
  if ( str.substring(0, 3) == "mRS" ) motorRStop();

  if ( str.substring(0, 3) == "mLF" ) motorLForward(str.substring(4, length).
↳toInt());
  if ( str.substring(0, 3) == "mLB" ) motorLBackward(str.substring(4, length).
↳toInt());
  if ( str.substring(0, 3) == "mLS" ) motorLStop();

  if ( str.substring(0, 3) == "mFF" ) motorFFForward(str.substring(4, length).
↳toInt());
}

```

(continues on next page)

(ve de la pàgina anterior)

```

    if ( str.substring(0, 3) == "mFB" ) motorFBackward(str.substring(4, length) .
↳toInt());
    if ( str.substring(0, 3) == "mFS" ) motorFStop();

    if ( str.substring(0, 3) == "mBF" ) motorBForward(str.substring(4, length) .
↳toInt());
    if ( str.substring(0, 3) == "mBB" ) motorBBackward(str.substring(4, length) .
↳toInt());
    if ( str.substring(0, 3) == "mBS" ) motorBStop();
}

```

A la **inicialització del MPU6050** Giroscopi/Acceleròmetre tan sols indiquem a quins pins està connectat.

```

void initializeMPU() {
    pinMode (21, INPUT_PULLUP);
    pinMode (23, INPUT_PULLUP);
    Wire.begin();
    mpu6050.begin();
}

```

4.3 4.3. Control

Com hem vist en els punts anteriors de la documentació la comunicació entre els R2B2-nanos i els programes de control es fa a través de missatges MQTT. Per tant es pot desenvolupar el programa de control amb qualsevol eina que tingui una llibreria MQTT i pugui subscriure's i enviar missatges a un tòpic.

4.3.1 4.3.1. Processing IDE

El programa de control que hi ha desenvolupat al projecte del GitHub està realitzat amb el [Processing IDE](#), ja que dona molta facilitat a l'hora de realitzar un programa executable en un sistema operatiu Android.

Per preparar el Processing IDE caldrà instal·lar el mode Android, fent clic al desplegable de la cantonada superior dreta on posa Java i seleccionar «Add Mode...». Dins el menú que se'n obre seleccionarem l'Android Mode i clicarem el botó install.

També farà falta afegir les llibreries MQTT i TockLibs per poder executar el programa de control. Per fer això ens hem de dirigir al punt de menú Sketch→Import Library...→Add Library... i fer la cerca de les dues llibreries esmentades.

Un cop fet això ja podem obrir el projecte **r2b2_control** que hi ha dins la carpeta code del projecte R2B2-nano i ja podem executar el codi en un mobil connectat per USB (menú Sketch→Run on Device) o en un emulador (menú Sketch→Run in Emulator).

4.3.2 4.3.2. Interfície Control

L'Interfície de control permet processar les dades del Giroscopi/Acceleròmetre i dibuixar en pantalla en quina posició es troba l'R2B2-nano. També té botons per interactuar amb els propulsors i d'altres per sol·licitar informació i realitzar tasques de manteniment.

Funcionalitats de la interfície:

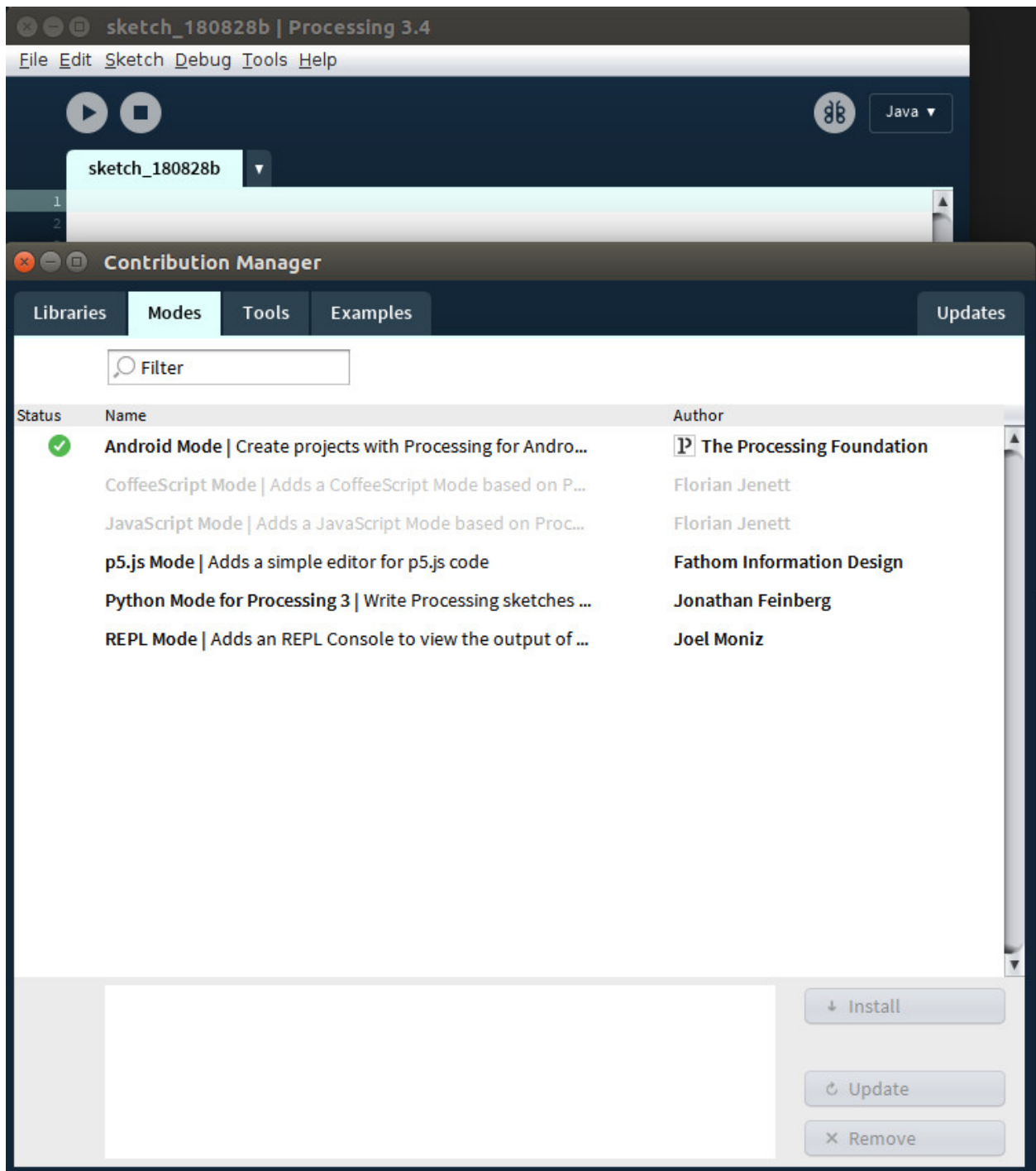


Fig. 4: Instal·lació Mode Android.

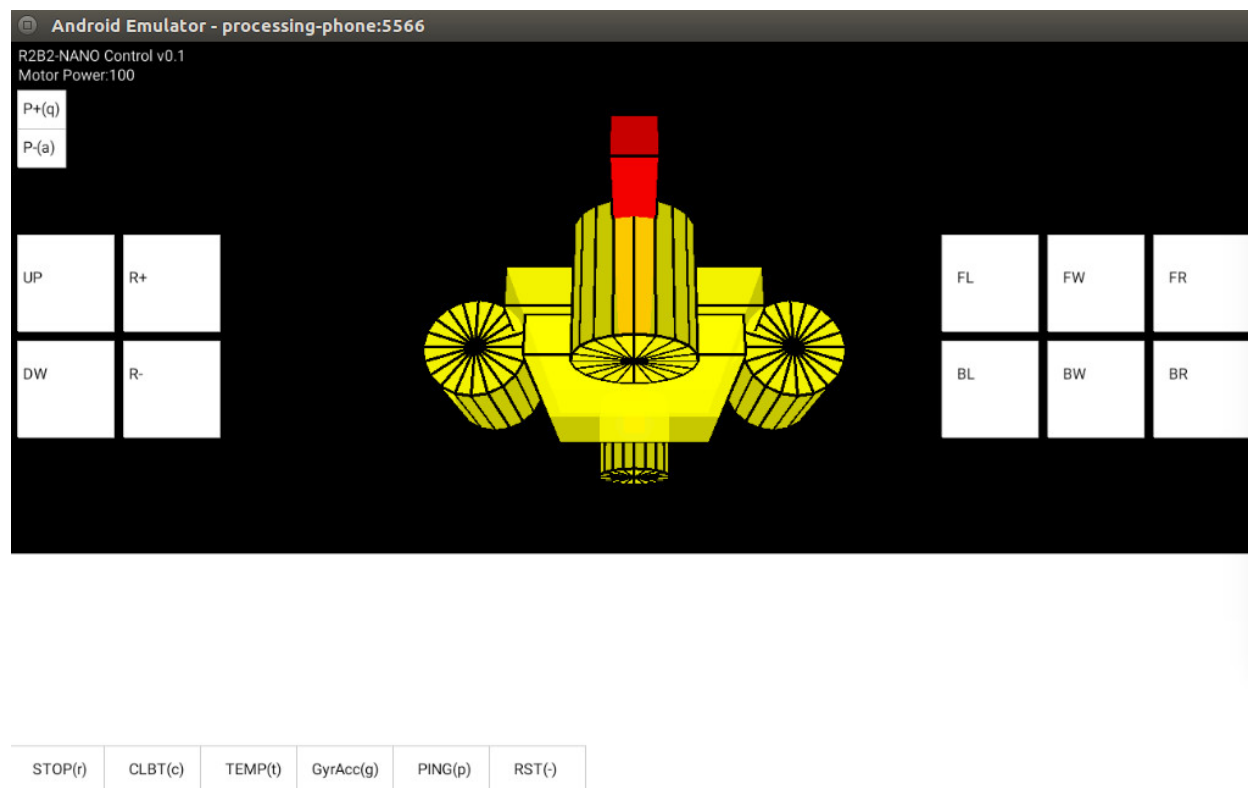


Fig. 5: Programa de control R2B2-nano.

Botó	Funció
P+(q)	Incrementar la potència dels motors
P-(a)	Decrementar la potència dels motors
UP	Pujar cap a la superfície
DW	Baixar cap al fons
R+	Rotar sobre els propulsors laterals +
R-	Rotar sobre els propulsors laterals -
FL	Girar endavant cap a l'esquerra
FW	Anar endavant
FR	Girar endavant cap a la dreta
BL	Girar endarrera cap a l'esquerra
BW	Anar endarrere
BR	Girar endarrera cap a la dreta
STOP	Parar tots els motors
CLBT	Calibrar el Giroscopi/Acceleròmetre
TEMP	Obtenir la temperatura de l'R2B2-nano
GyrAcc	Demana la posició al Gir/Acc (no reflexa resultats en pantalla)
PING	Demana senyals de vida (no reflexa resultats en pantalla)
RST	Test de motors (no reflexa resultats en pantalla)

4.3.3. Codi del Control

4.3.3.1. Configuració

Com hem vist en el punt 4.1 cada R2B2-nano te un identificador que fa servir per publicar al un servidor MQTT. Per tant cal que especifiquem al programa de control quin R2B2-nano volem controlar i a quin servidor MQTT ens hem de connectar. Això ho fem definint les variables **r2b2Id** i **MQTTServer** al principi del codi.

```
String r2b2Id = "3291342516";
String MQTTServer = "192.168.82.106";
```

4.3.3.2. Estructura del codi

En el setup del codi ens subscrim al tòpic corresponent a l'R2B2-nano que volem controlar. Un cop fet això anem sol·licitant cada 300 mil·lisegons els valors del Giroscopi/Acceleròmetre. Mirem si hi ha alguna ordre de moviment activa i en cas afirmatiu activem els motors corresponents. Llavors mirarem quins punts de la pantalla s'estan tocant i si corresponen a algun dels botons. Per finalitzar dibuixem els botons i l'R2B2-nano en la posició que hem llegit del Giroscopi/Acceleròmetre.

```
...
//SETUP
...
//Configure MQTT
client = new MQTTClient(this);
client.connect("mqtt://" + MQTTServer, "Pro-" + r2b2Id);
client.subscribe("r2b2/" + r2b2Id + "/"#");
//
...

//DRAW
...
if (millis() - interval > 300) {
    client.publish("r2b2/" + r2b2Id + "/tasks", "getGyrAcc");
    ping();
    interval = millis();
}

//Look for active motors////////////////////////////////////
//Look for Touches////////////////////////////////////
//Draw Buttons////////////////////////////////////
//Draw R2B2-nano////////////////////////////////////
```

5. Muntatge

Fases del Muntatge

1. Base de Càrrega
2. Xassís Alimentació
3. Xassís Control
4. Propulsors
5. Boia
6. Carcassa
7. Acoblament

Llistat del material necessari

Tipus	Material	Quantitat
3D	Base Càrrega Top	1
3D	Base Càrrega Down	1
3D	Propulsor Part A	4
3D	Propulsor Part B	4
3D	Hèlix	4
3D	Encapsulat Motors	4
3D	Xassís Alimentació	1
3D	Xassís Control	1
3D	Protecció port	2
3D	Boia Top	1
3D	Boia Down	1
MAT	Volandera M3 Inox	6
MAT	Cargol M3x6 Inox	4
MAT	Cargol M3x16 Inox	6
MAT	Femella M3 Inox	2

Continua a la pàgina següent

Table 1 – ve de la pàgina anterior

Tipus	Material	Quantitat
MAT	Cargol sense cap M2x3	4
MAT	Escuma FOAM	
MAT	Cable electronica de silicona 28 AWG	
MAT	Funda Termoretractil - 15cm	
ELECT	Passador DuPont	2
ELECT	Connector Mascle DuPont	1
ELECT	Coonector Femella DuPont	1
ELECT	R2B2_nano Board	1
ELECT	ESP32 WROVER	1
ELECT	Mòdul TB6612FNG	2
ELECT	Mòdul MPU6050	1
ELECT	Mòdul TP4056 (Battery Charger)	2
ELECT	Registència 12k (Op.: 2012/0805 SMD)	1
ELECT	10 UF 16 V 106C 1206/3216 SMD	1
ELECT	0.1 UF 104 Ceramic (Op.:2012/0805 SMD)	2
ELECT	Regulador DD0403MB_3V3	1
ELECT	716 Coreless Motor	4
ELECT	Connector IPEX/U.FL 20278 1.13mm	1
ELECT	1.13mm Wires/Coaxial Antenna Cable 50ohm	5
ELECT	Receptor Qi Wireless Charger Receiver	2
ELECT	Emisor Qi Wireless Power Charging	2
ELECT	Bateria 3.7V 550mAh	2
ELECT	Regulador lm2596 (8v -> 6v)	1

Altres Materials

- Cola tèrmica
- Estany
- Resina de poliéster i catalitzador
- Cinta aïllant
- Vaselina
- Massilla de poliéster
- Esprai D'imprimació
- Esprai de pintura

Eines i material de protecció

- Soldador
- Estenalles d'electrònica
- Motlles de Silicona per contenir la resina
- Guants
- Mascara
- Ulleres de protecció
- Trepant amb broca petita i fresadora
- Pistola Cola Tèrmica

- Serra/llima per treure els excessos de resina
- Tester
- Espatula petita
- Clau Allen 0.9mm
- Cutter
- Llimes
- Paper de vidre
- USB to TTY

5.1. Base de Càrrega

6.1 5.1.1. Que necessitem?

6.1.1 5.1.1.1 Material

- 1x Base Càrrega Top - (Base Top STL)
- 1x Base Càrrega Down - (Base Down STL)
- 2x Cargols M3 x 6mm
- 2x Emisor Qi Wireless Power Charging
- Cola Tèrmica

6.1.2 5.1.1.2 Eines

- Tornavís
- Pistola Cola Tèrmica
- Guants

6.2 5.1.2 Muntatge

Agafem la base de càrrega, amb la pistola de cola tèrmica posem dues piles petites de cola per enganxar els dos Emisors Qi com es mostra a continuació. Cal tindre en compte que els port micro USB han de coincidir amb els forats de la base de càrrega.

Assegurem les plaques de càrrega amb una mica de cola tèrmica per que no es moguin al connectar els USBs.

Cargolem la tapa de la base de càrrega amb els dos cargols M3 de 6mm.

I ja tenim la base de càrrega apunt per connectar a qualsevol USB o carregador de mòbil.

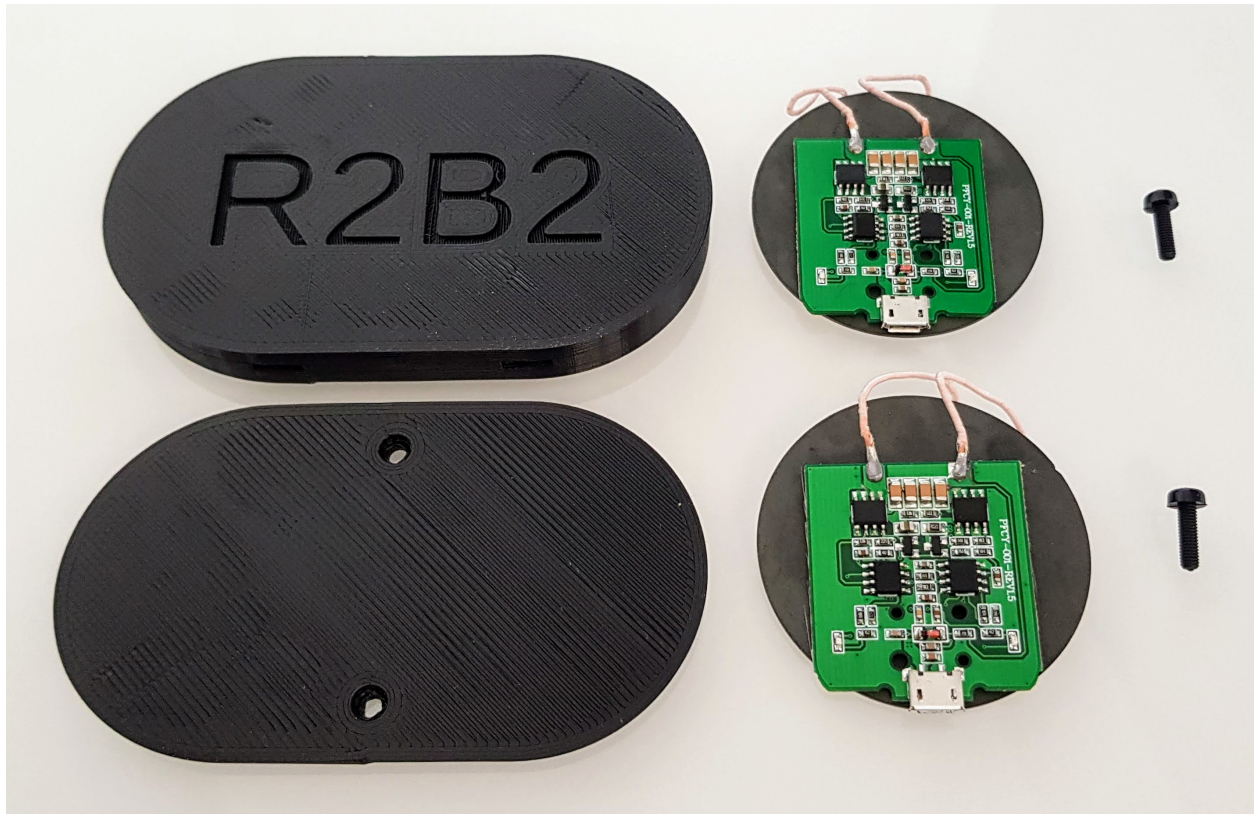
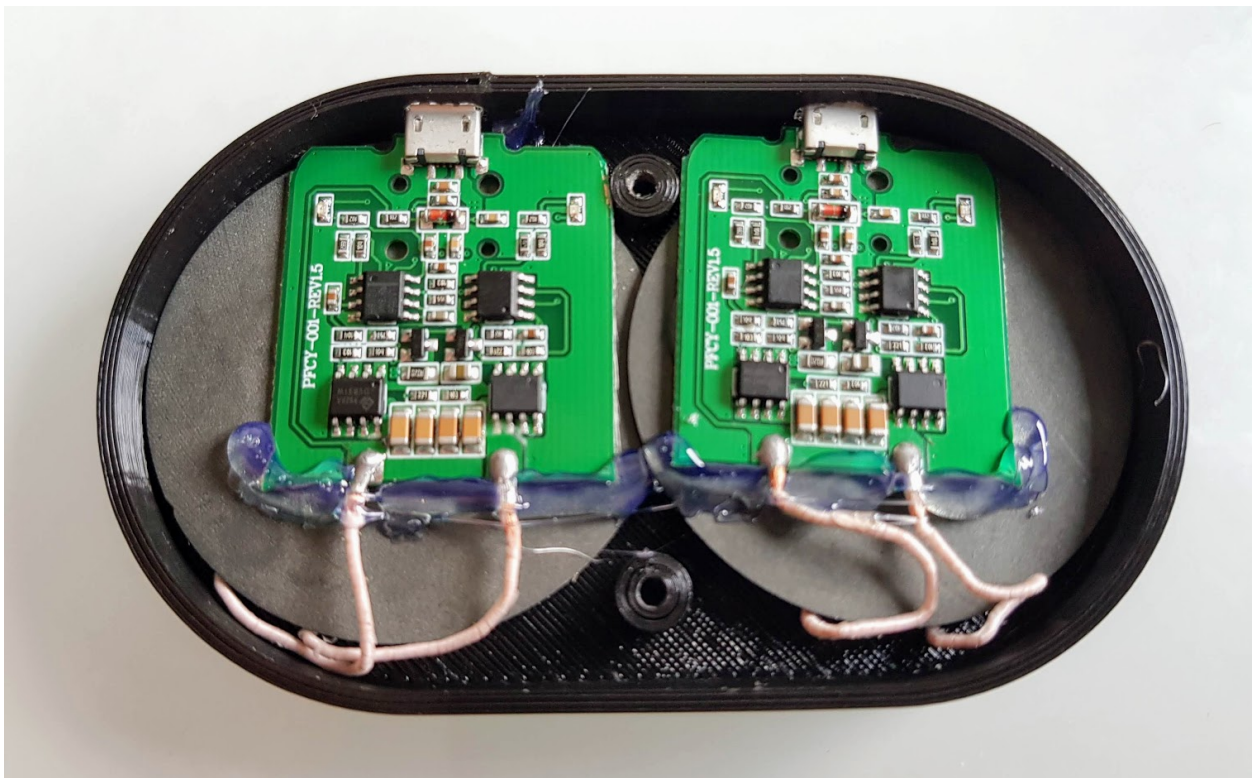
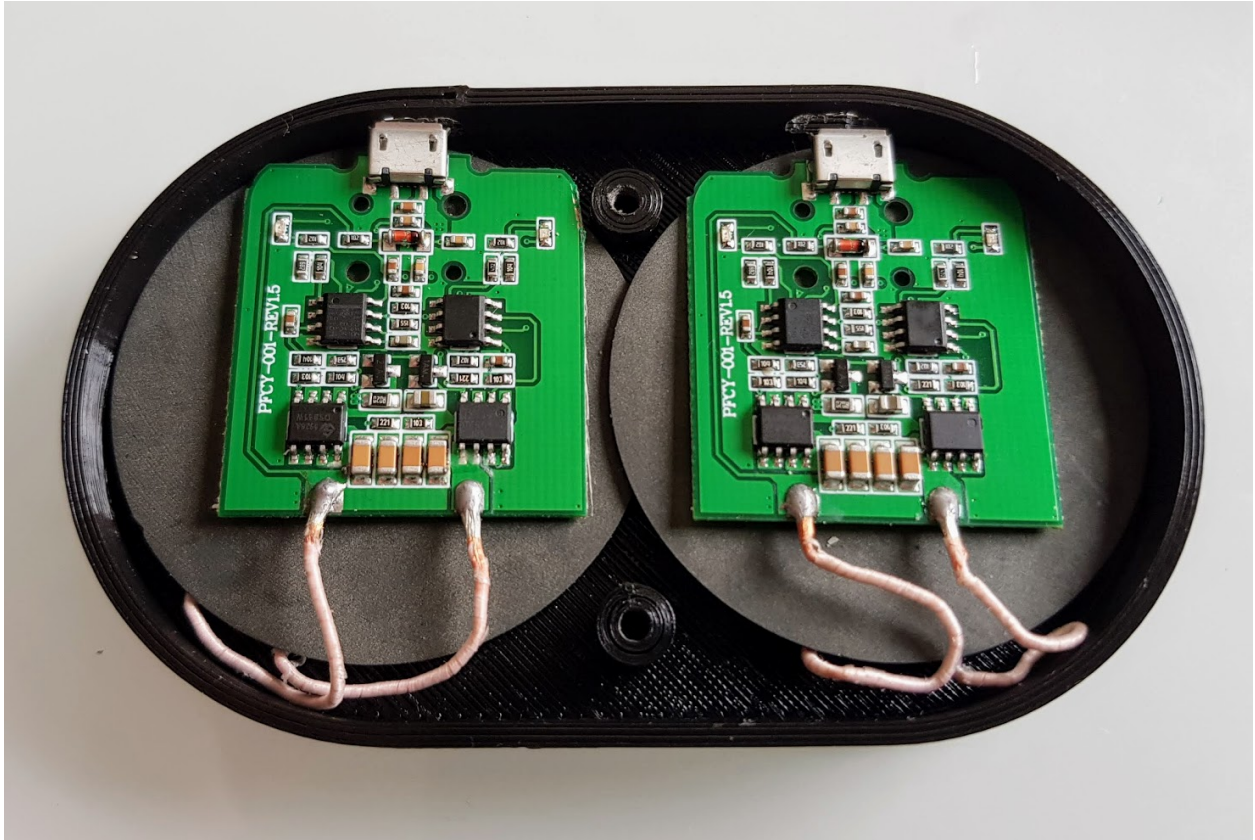


Fig. 1: Material per muntar la base de càrrega.







5.2. Xassís Alimentació

7.1 5.2.1. Que necessitem?

7.1.1 5.2.1.1 Material

- 1x Xassís Alimentació (Fitxer STL)
- 2x Mòdul TP4056 (Cargador de bateria)
- 2x Receptor Qi Wireless Charger
- 2x 3.7V 550mAh Battery
- 1x Regulador voltage (8v → 5.5v)
- Cable elèctric de silicona 28 AWG
- Estany
- Cola Tèrmica
- Resina de polièster i catalitzador
- Cinta aïllant

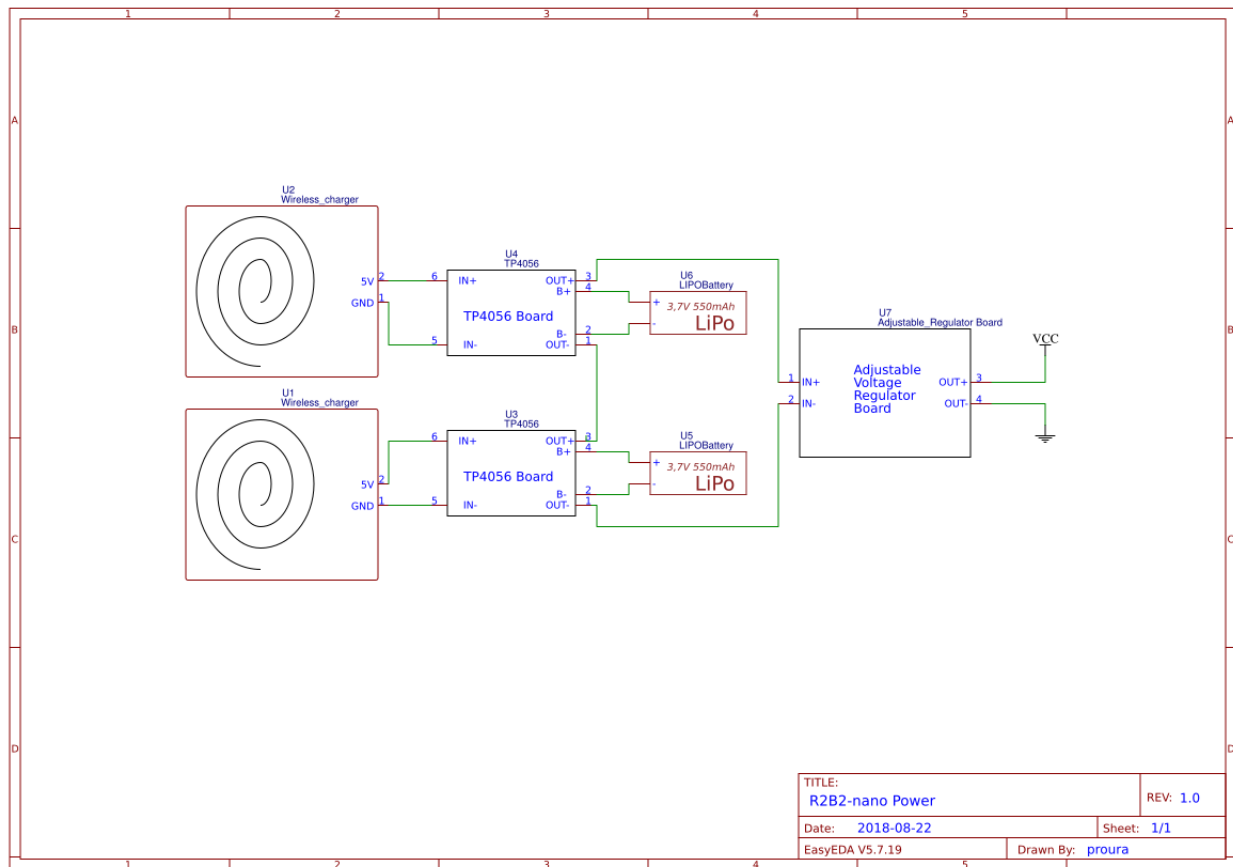
7.1.2 5.2.1.2 Eines

- Soldador
- Estenalles d'electrònica
- Motlles de Silicona per contenir la resina
- Guants
- Mascara
- Ulleres de protecció

- Trepant i broca petita
- Pistola Cola Tèrmica
- Serra/llima per treure els excessos de resina
- Tester

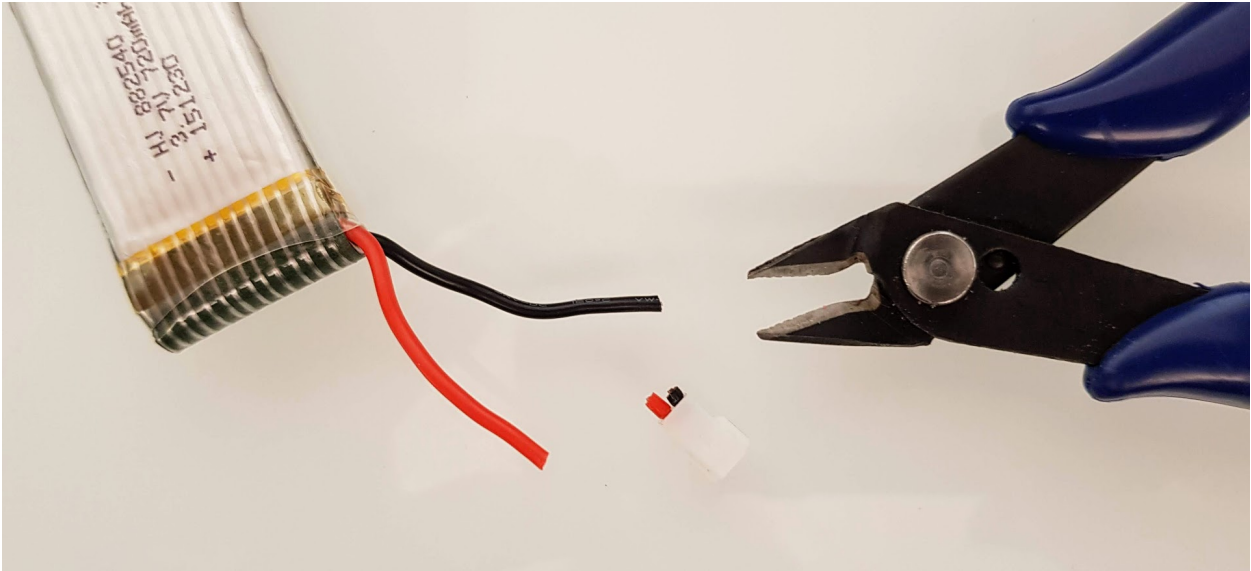
7.2 5.2.2 Muntatge

Es pot trobar l'esquema electrònic «r2b2-nano-power» al projecte públic [R2B2-nano](#) allotjat a EasyEDA.

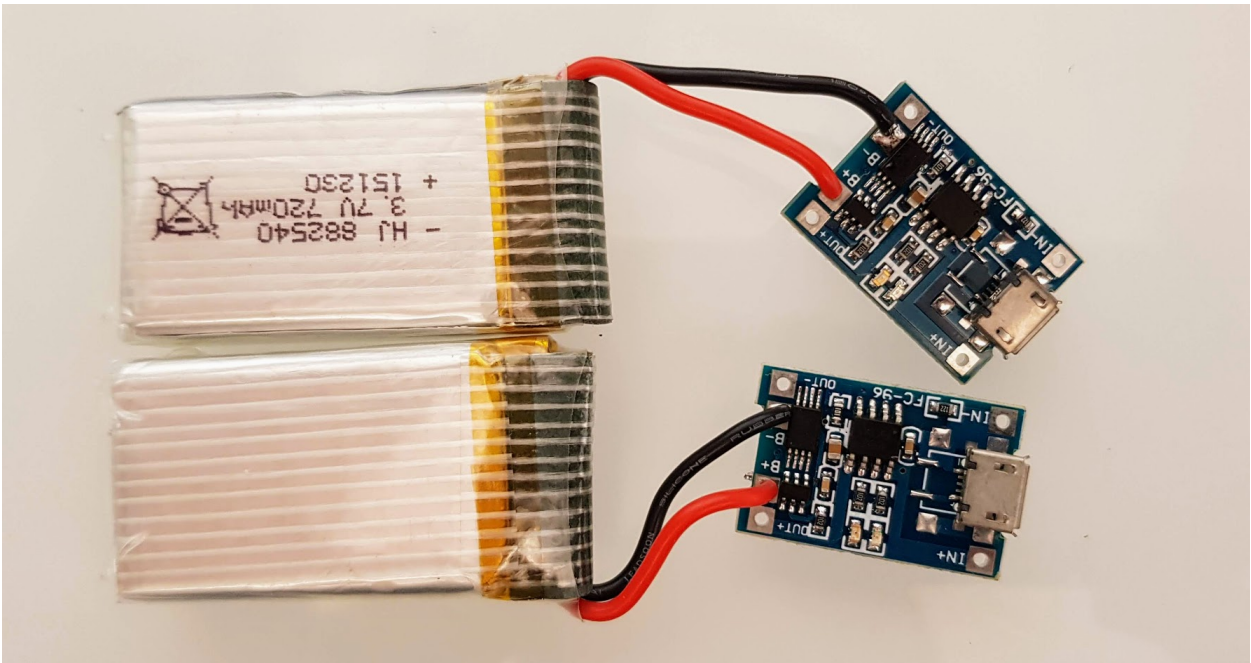


Primer de tot tallem els cables de la bateria LiPo.

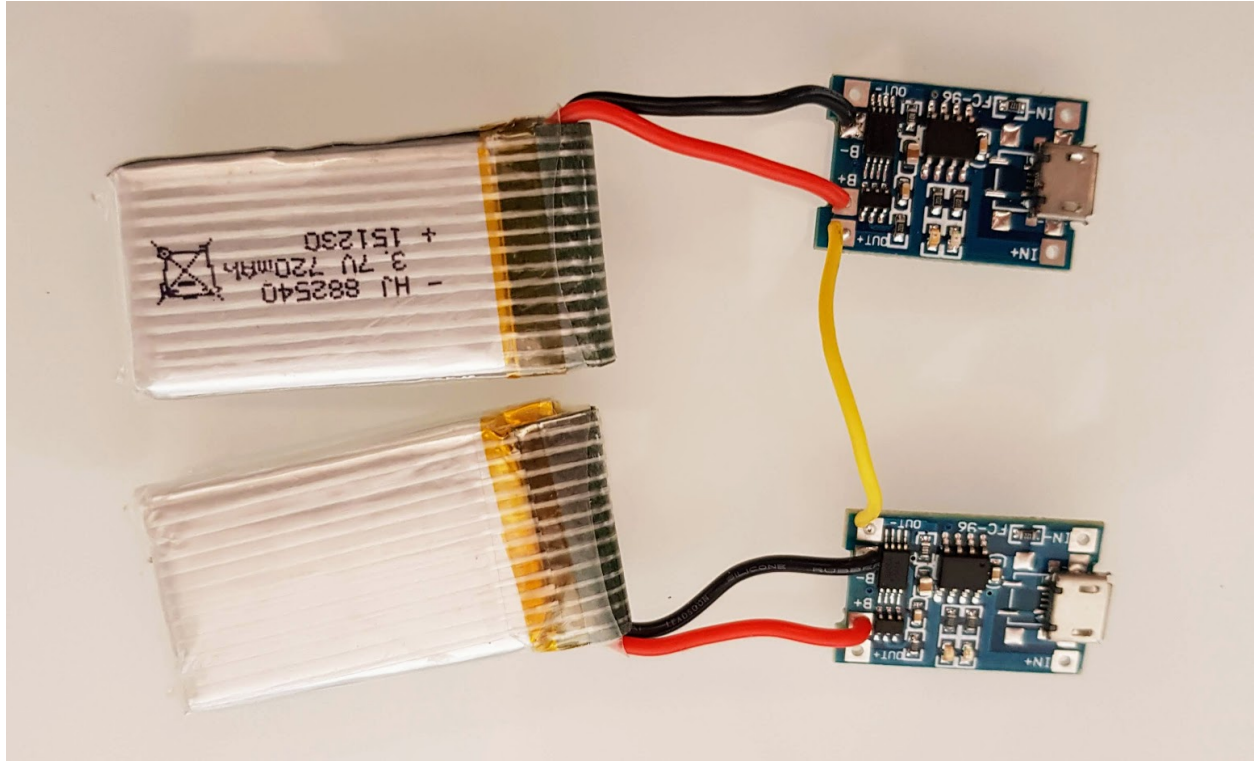
Avís: Cal tallar els cables d'un en un per no creuar els cables i dañar les bateries!!!



Soldem els cables de les bateries als mòduls de càrrega TP4056 tinguent en compte que el vermell ha d'anar al B+ i el negre al B-.

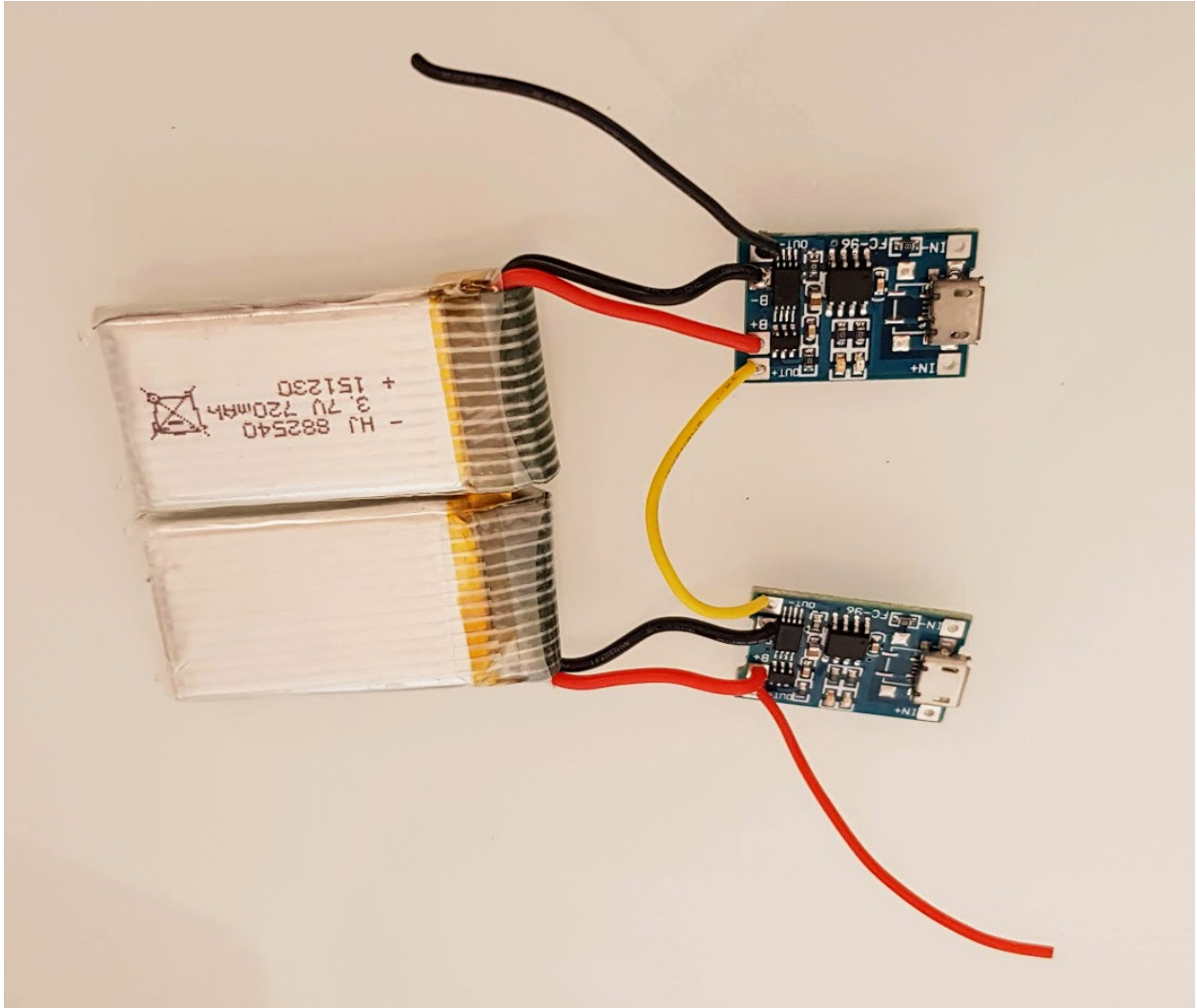


Per posar les bateries en sèrie connectarem la sortida OUT+ d'un mòdul TP4065 amb l'OUT- de l'altre mòdul.

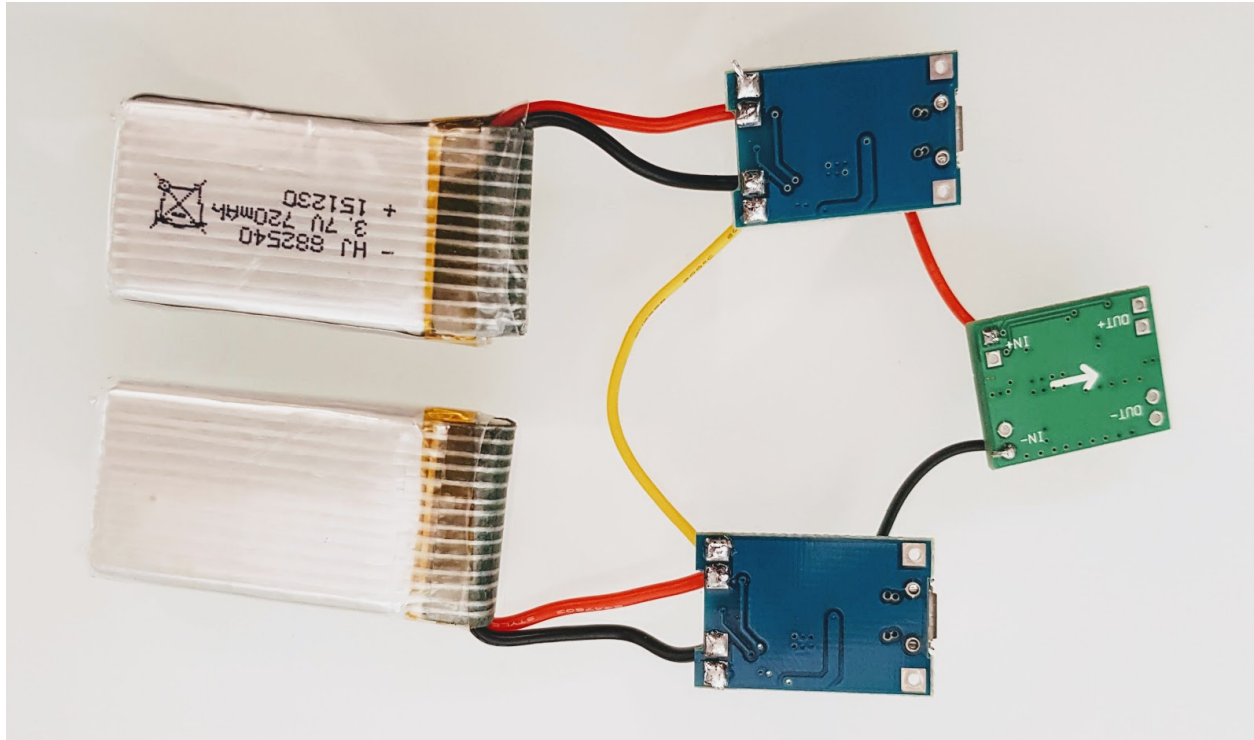


Soldarem un cable a la sortides restants de cada mòdul TP4056 un a l'OUT+ i l'altre a l'OUT-.

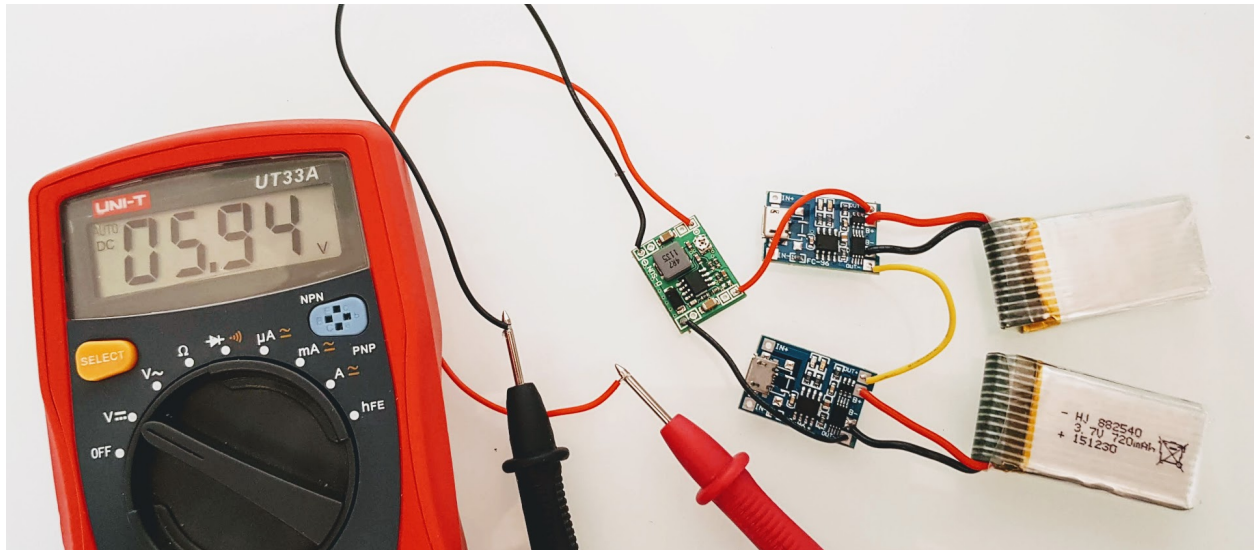
Nota: Si sempre cablejem les masses amb color negre i els pols positius amb vermell ens ajudarà a identificar els cables més facilment i comentre menys errors.



Ara connectem el OUT+ del modul TP4056 a l'IN+ del regulador de voltatge i l'OUT- a l'IN-.



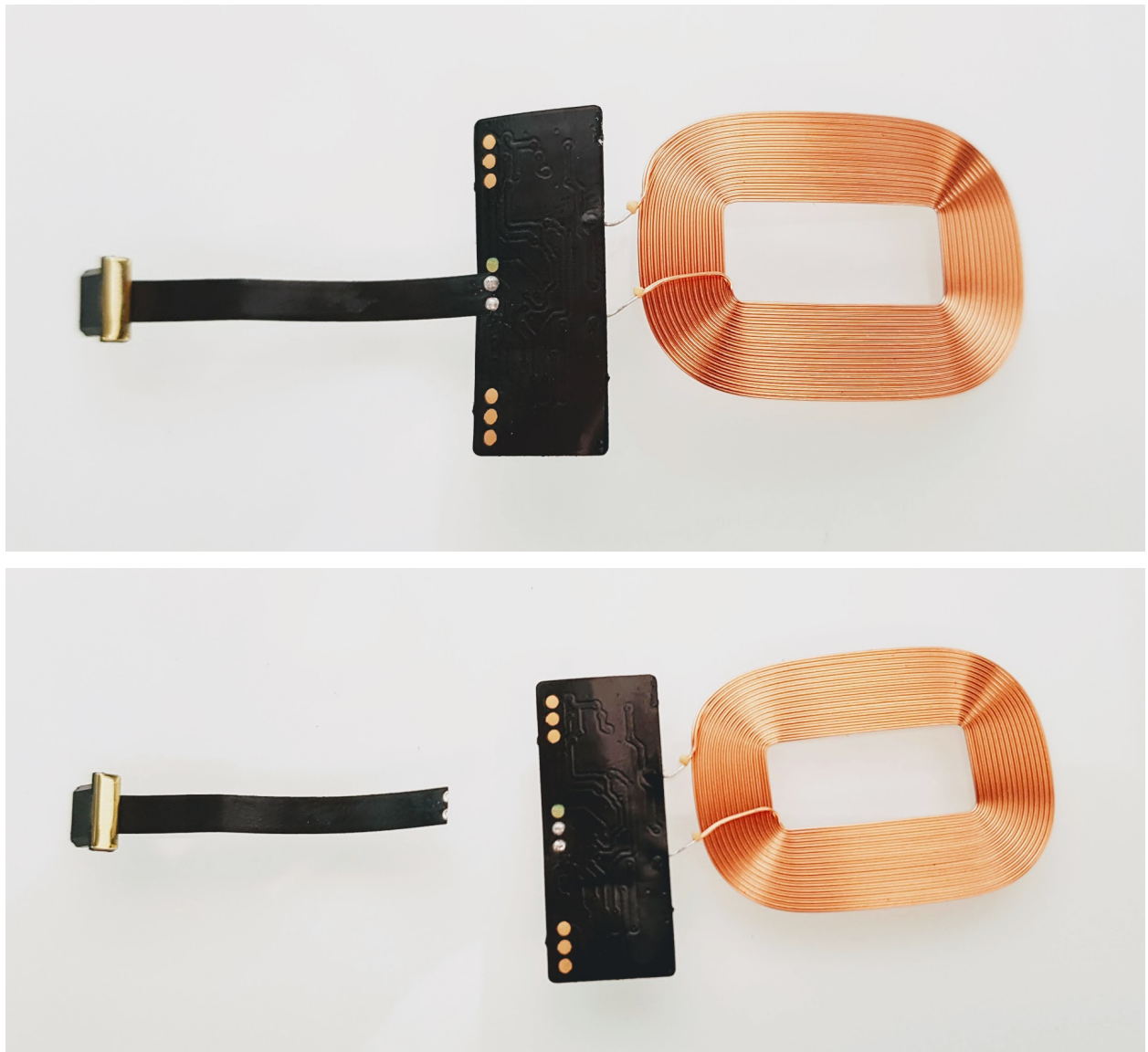
Arribats a aquest punt podem comprovar amb un tester que el voltage que ens dona és correcte.



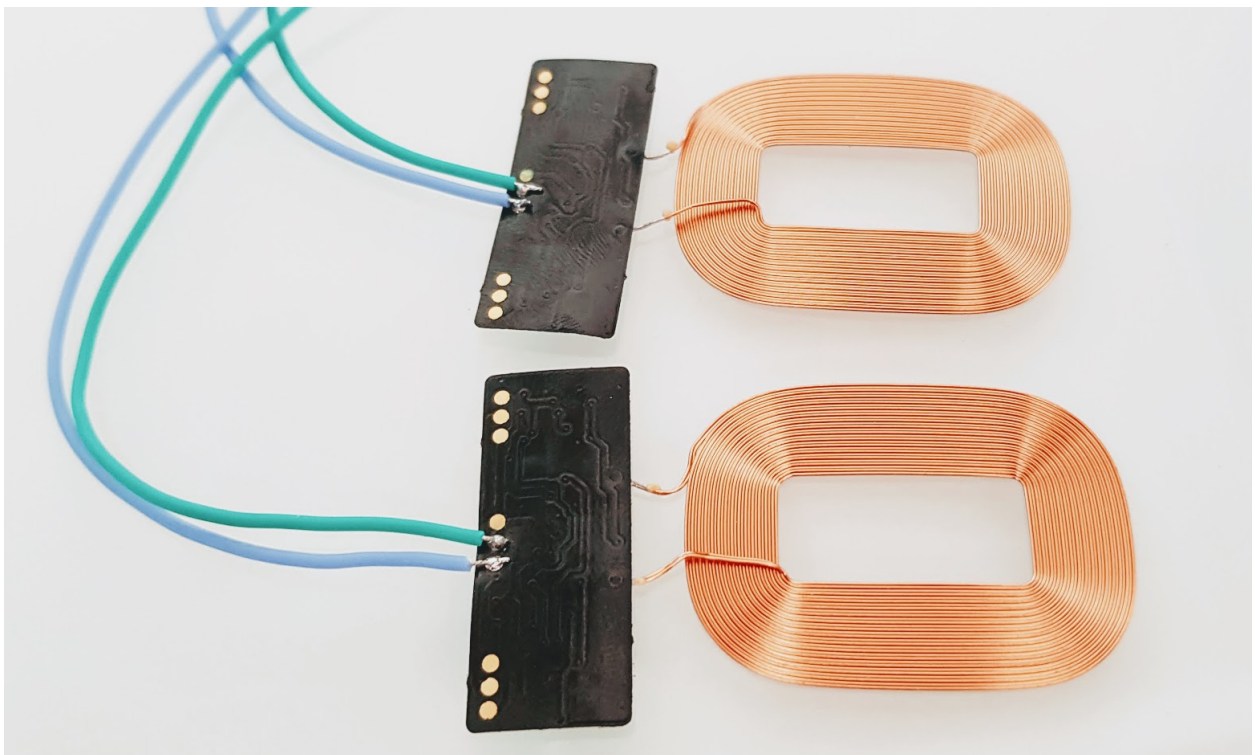
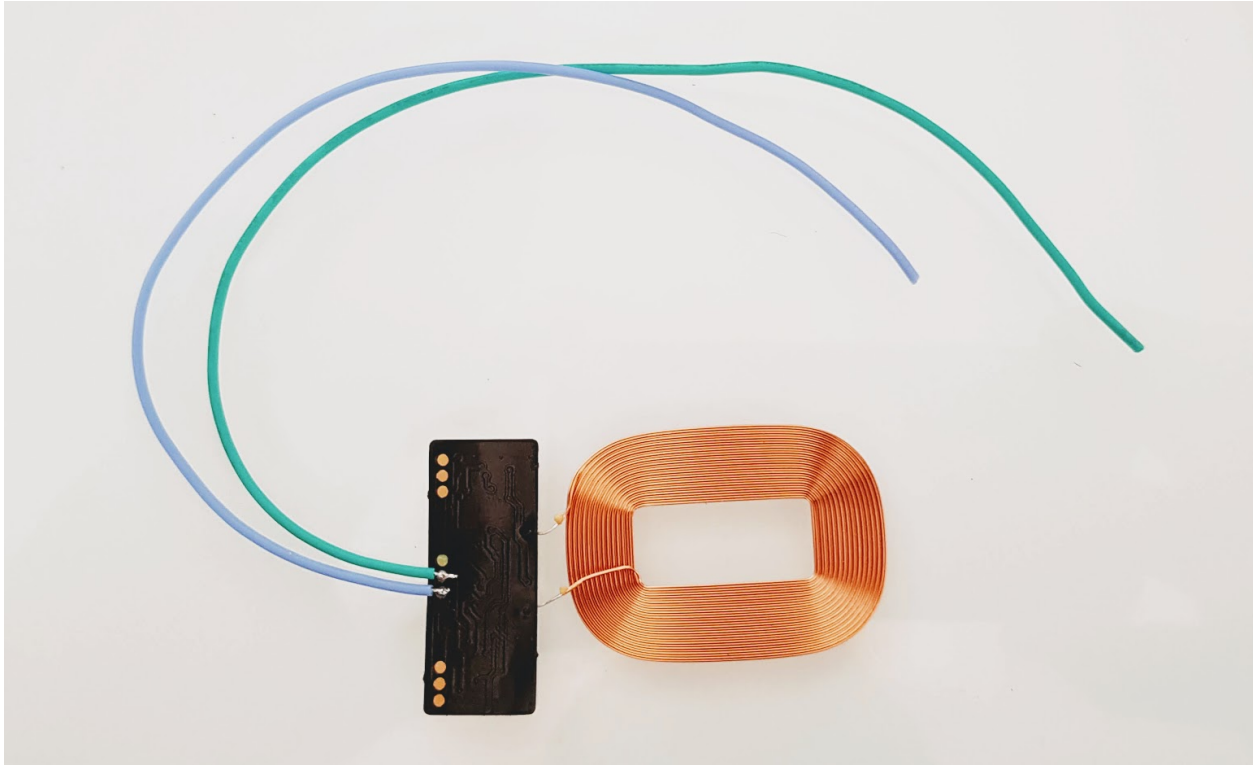
Ara agafem els receptors Qi i els hi traiem els adhesius.



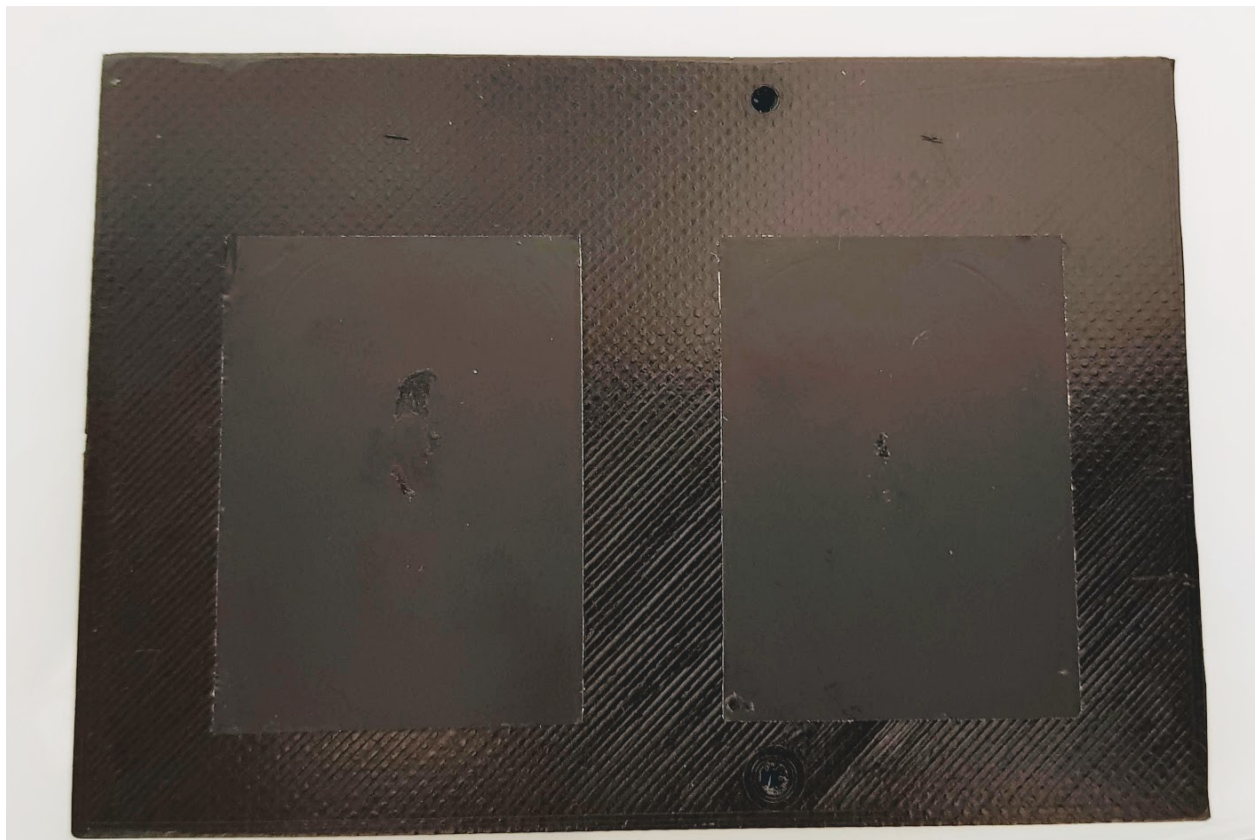
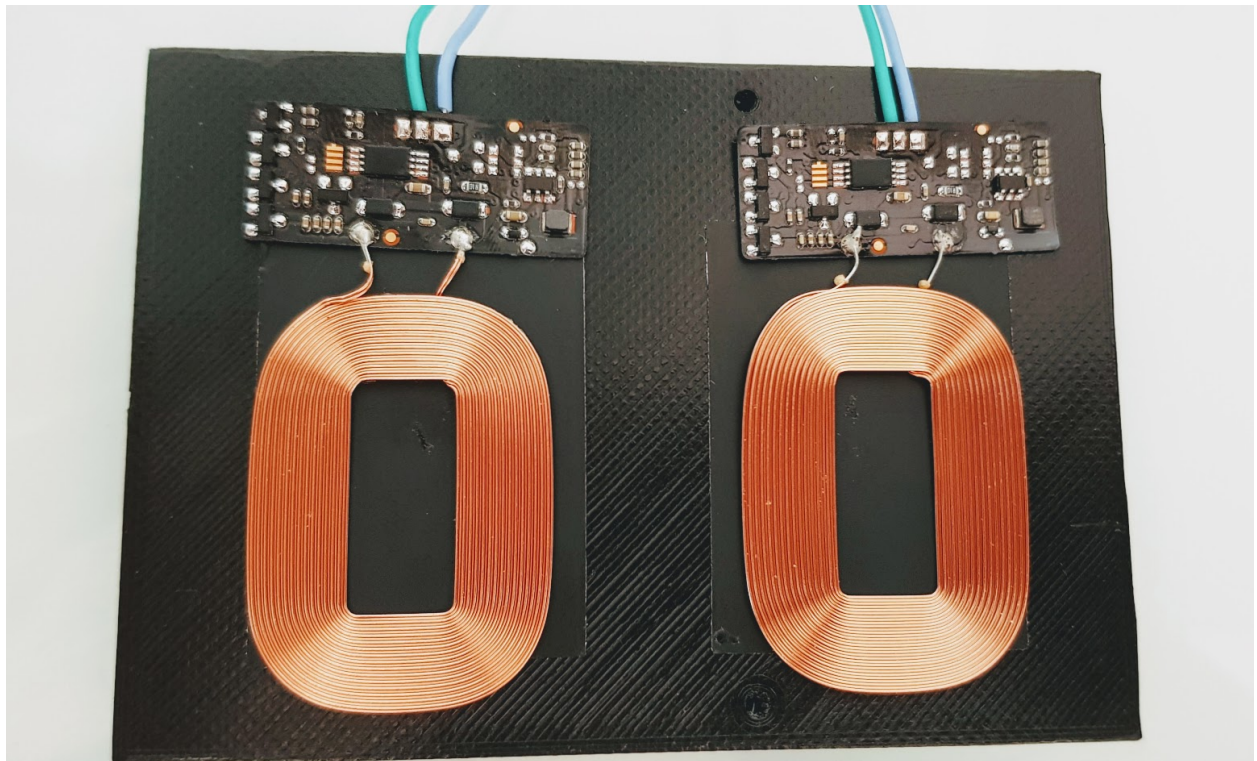
Amb el soldador, dessoldem el port USB.



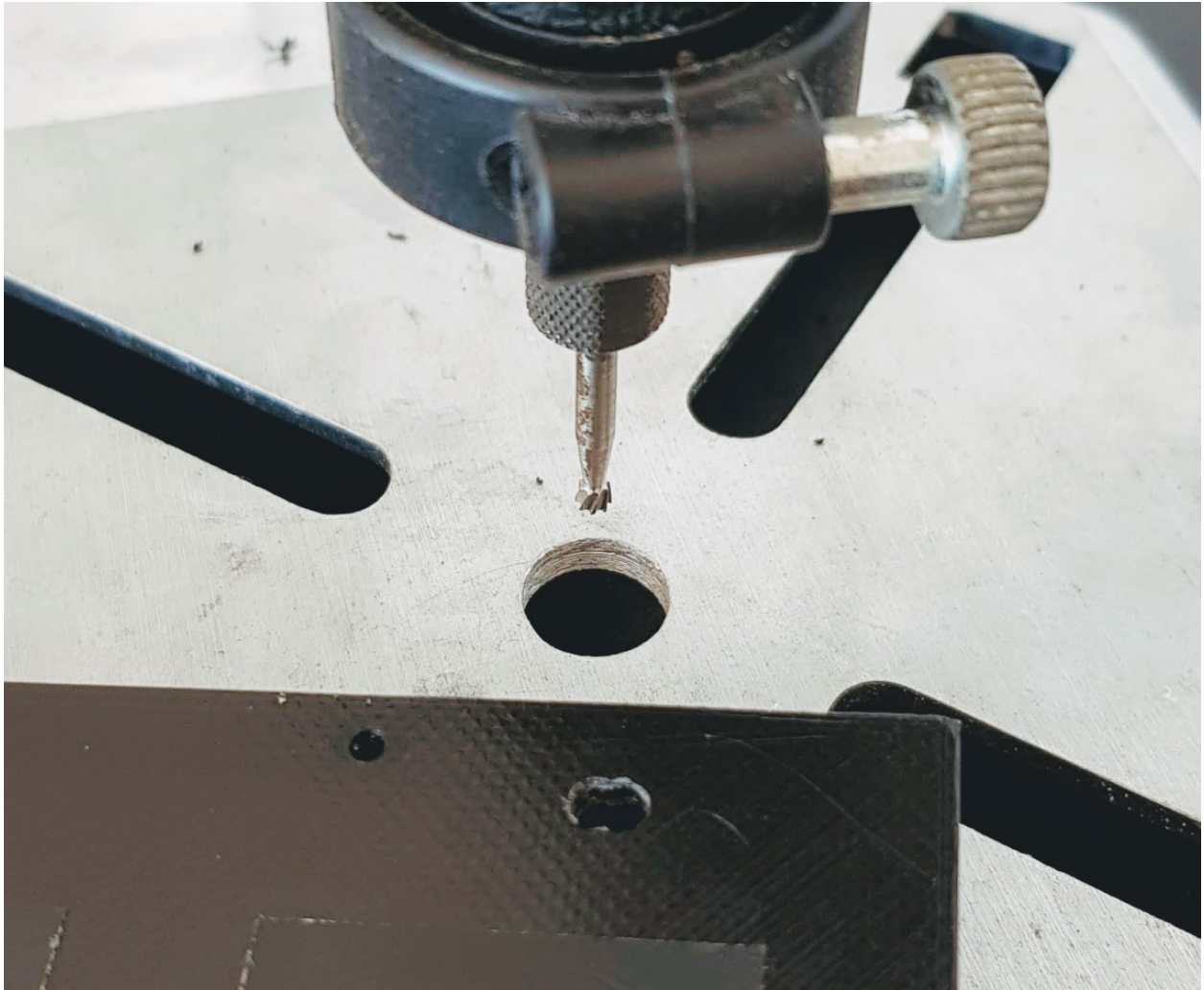
I hi soldem dos cables de silicona nous.

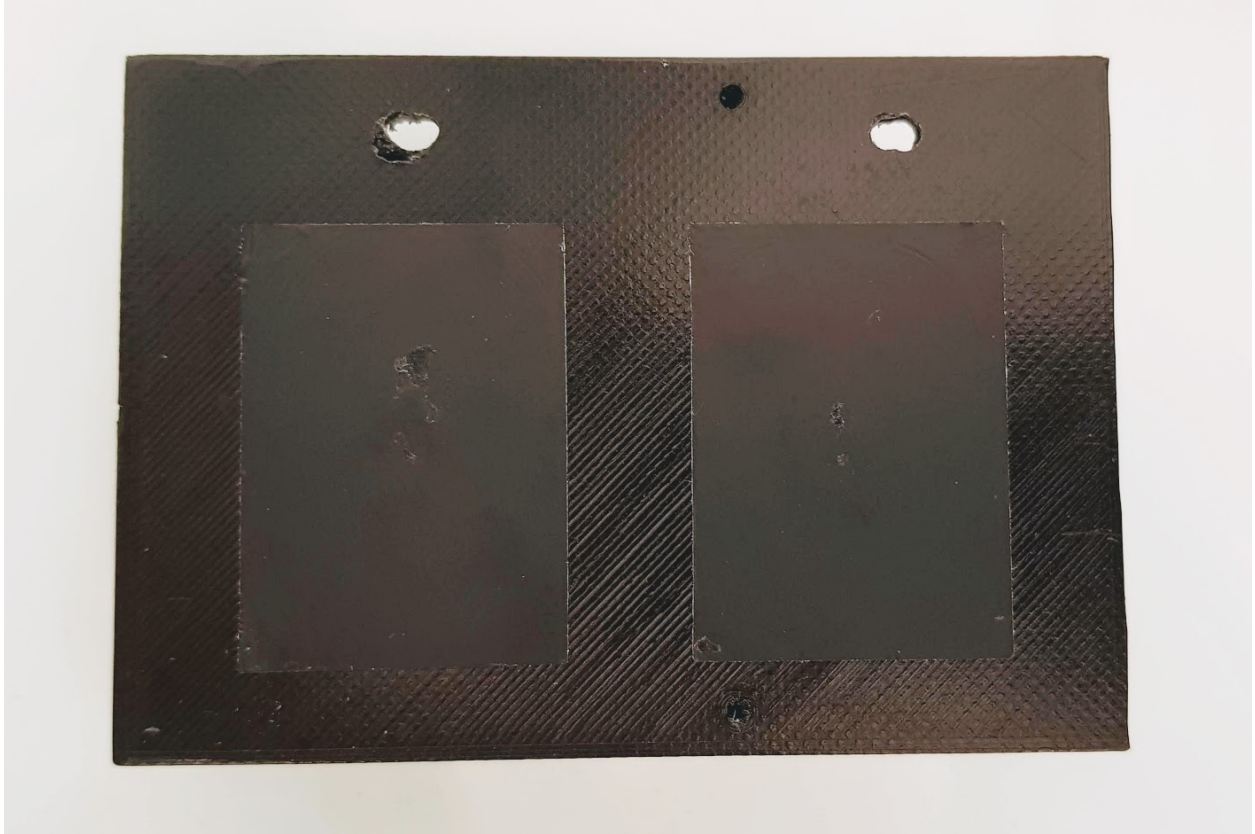


Col·loquem els dos carregadors a la base de l'R2B2-nano i fem unes marques a on foradarem per passar els cables.

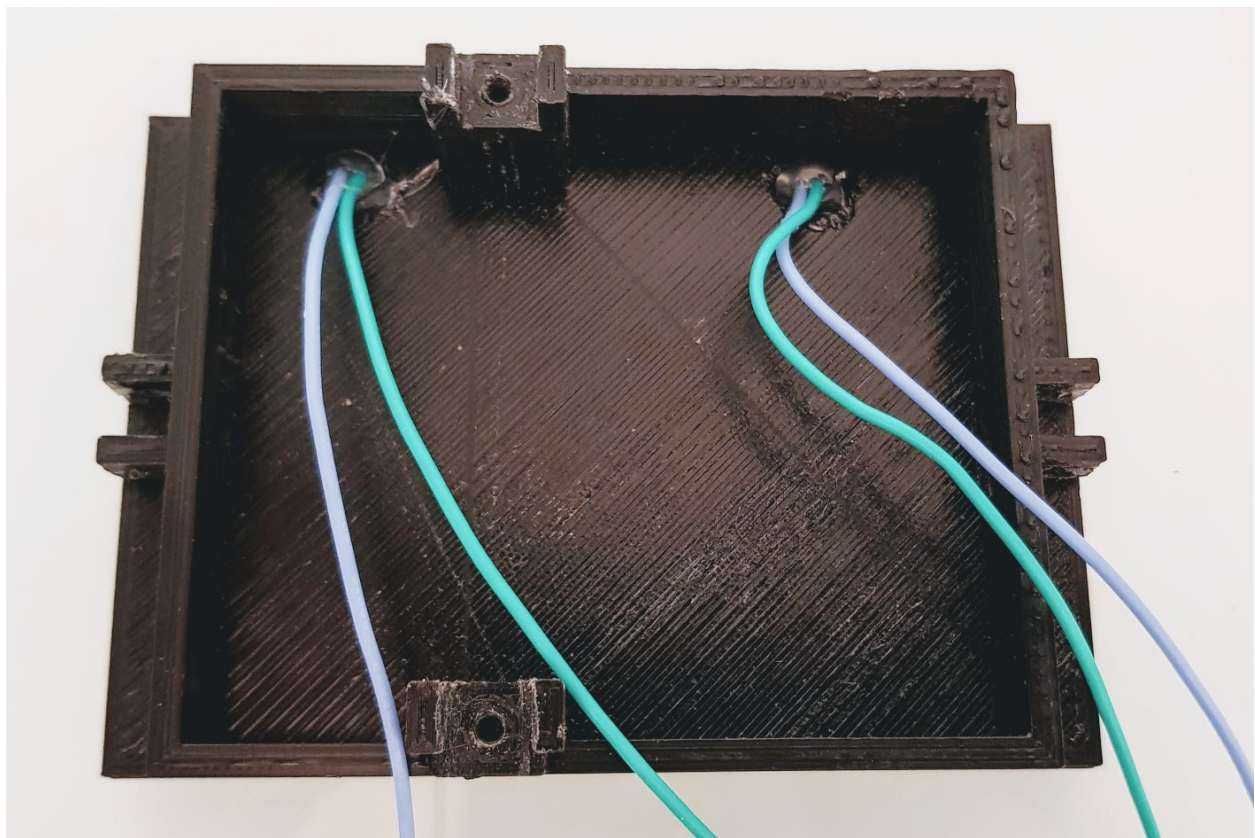
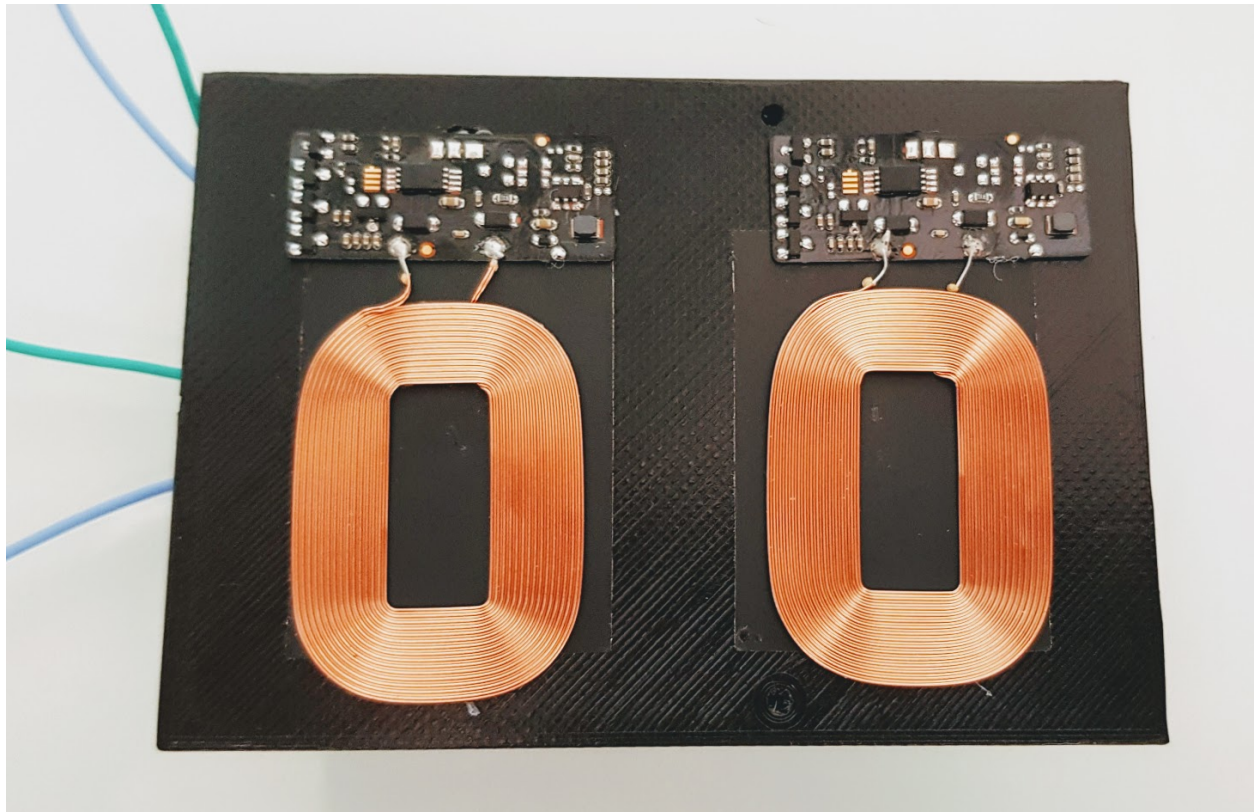


Foradem amb un trepant petit i una broca o fresadora.





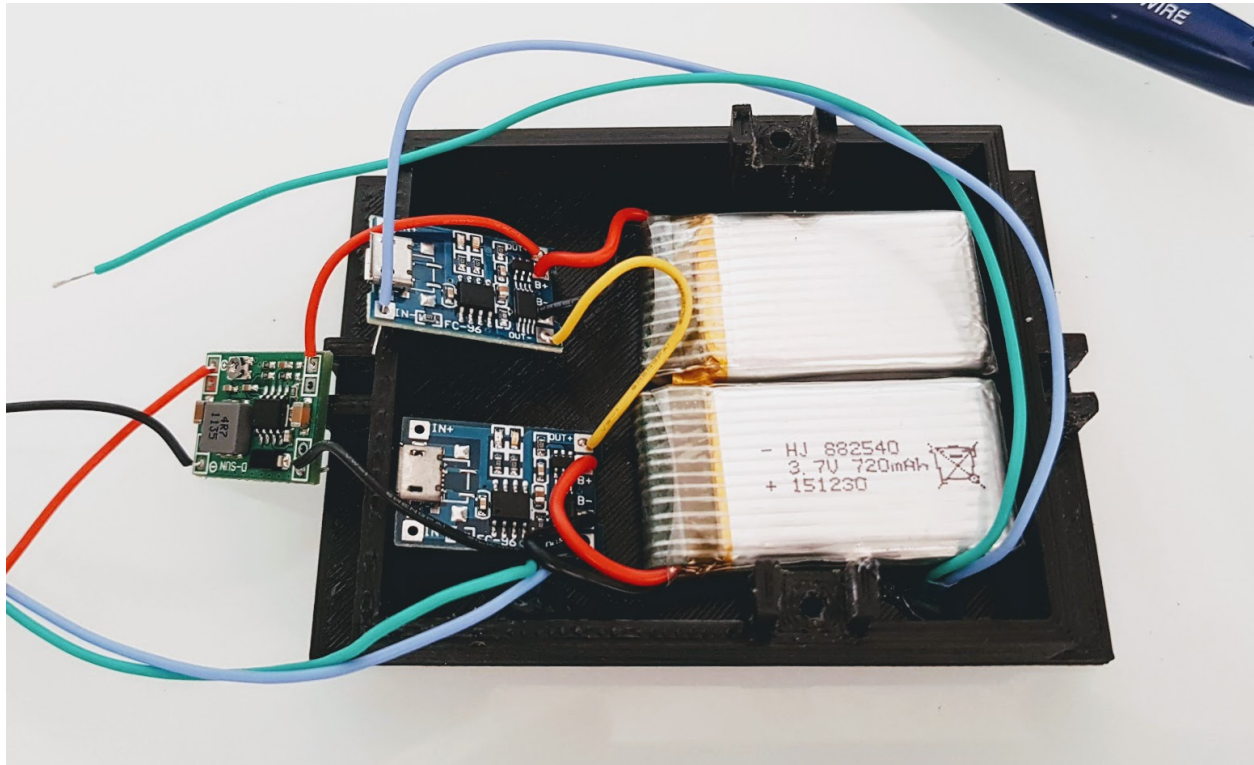
Tornem a col·locar els receptors Qi a la base, però ara fent passar els cables nous pels forat i enganxant-los amb cola tèrmica.



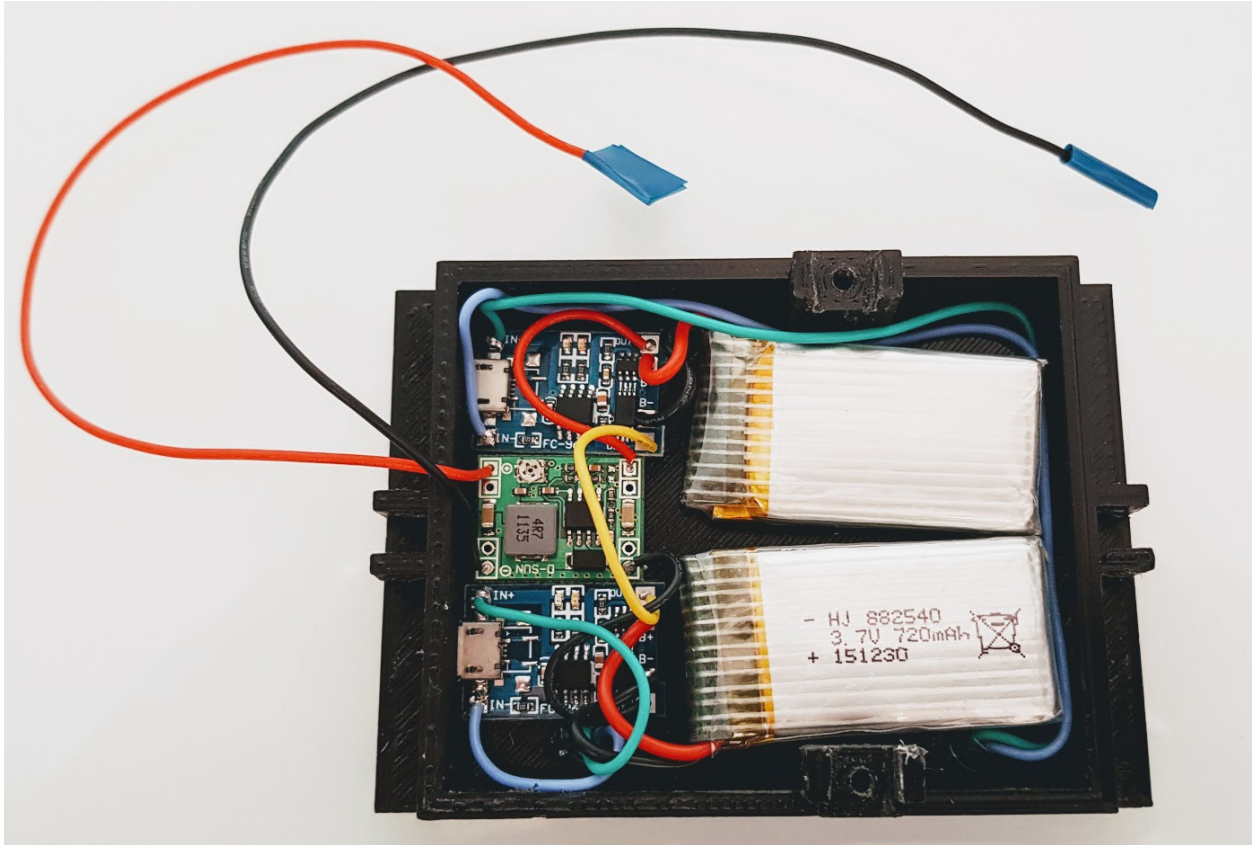
Ara ja podem soldar cadascun dels receptors Qi al mòdul de càrrega.

Avís: Cal comprovar amb un tester quin és el pol positiu i negatiu de cada receptor Qi. Per fer això posarem el receptor Qi mòdul sobre un emisor Qi endollat i connectarem els dos cables al tester.

És molt important també posar els components amb la orientació correcta, ja que els cables d'alimentació accedeixen al xassís de control per un forat fet a propòsit.



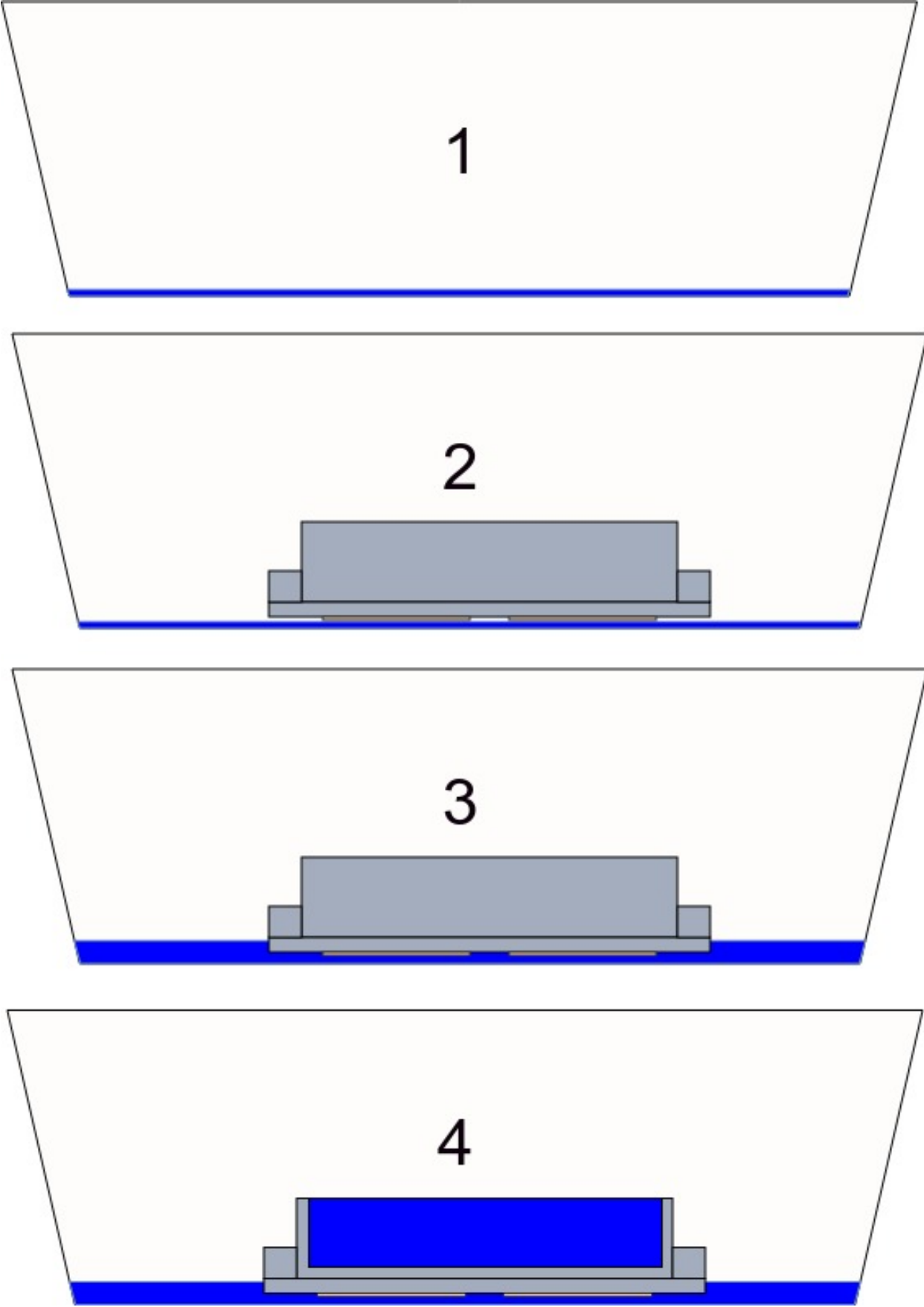
Arribats a aquest punt ja tenim la base d'alimentació a punt per enresinar. Només caldrà verificar que tenim el voltatge de sortida desitjat amb un tester, tapar amb cinta aïllant els dos cables i col·locar els element i cables dins al xassís de tal manera que no sobresurtin.

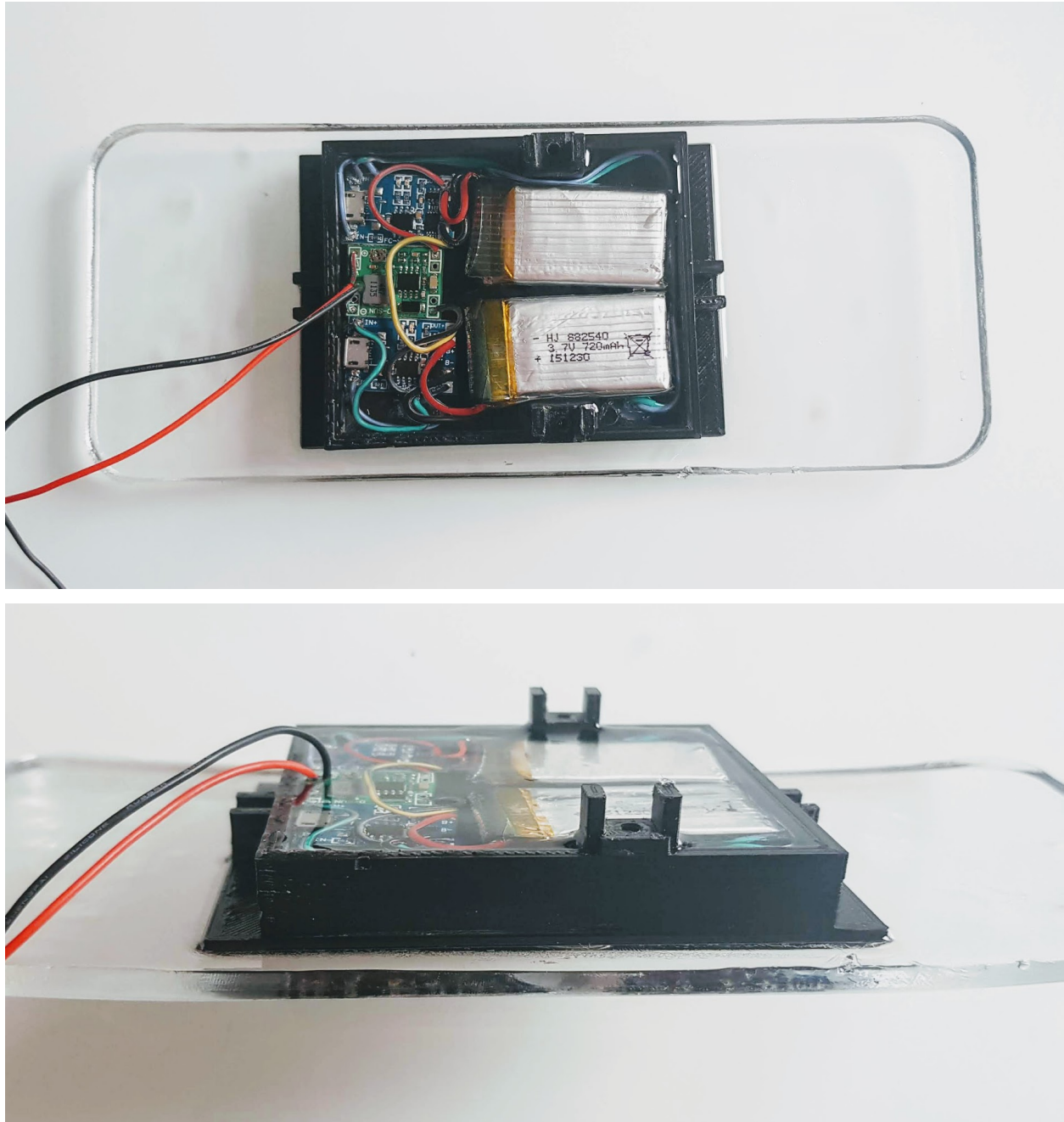


Per enresinar podem fer servir qualsevol motlle de silicona, tan per fer la mescla com per contenir-la a l'hora de bolcar-la.



1. Primer de tot farem una base d'1 a 2 milimetres de resina i la deixarem assecar.
2. Col·locarem la base a dins el motlle sobre la resina seca.
3. Prepararem més resina i la bolcarem dins el motlle fins a quasi arribar a l'alçada del suports.
4. Un cop assecada la resina omplirem la base d'alimentació fins a cobrir tots els dispositius electrònics, cables i bateries. Només deixarem sortir els dos cables, positiu i negatiu, tal i com es mostra a les següents imatges.

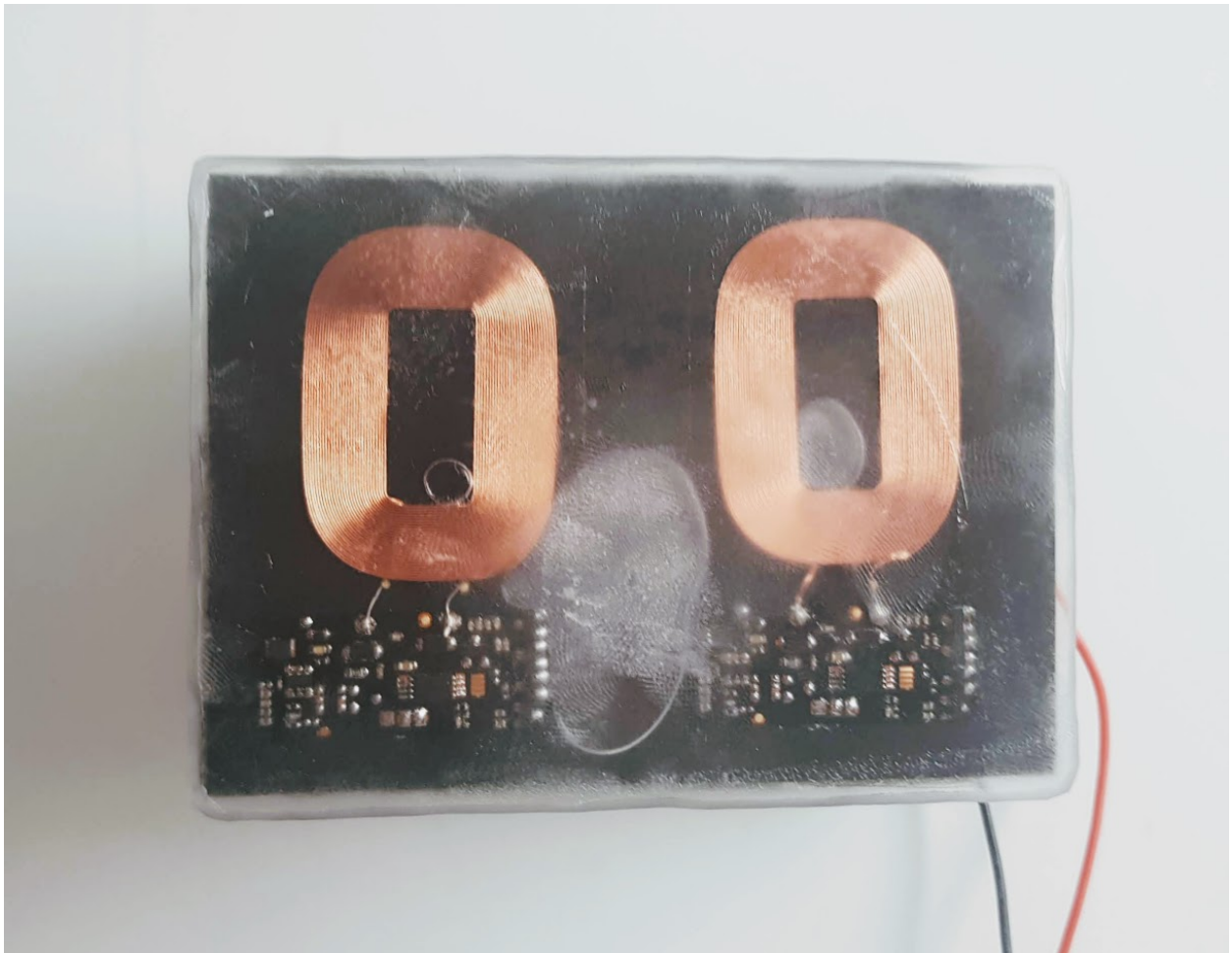


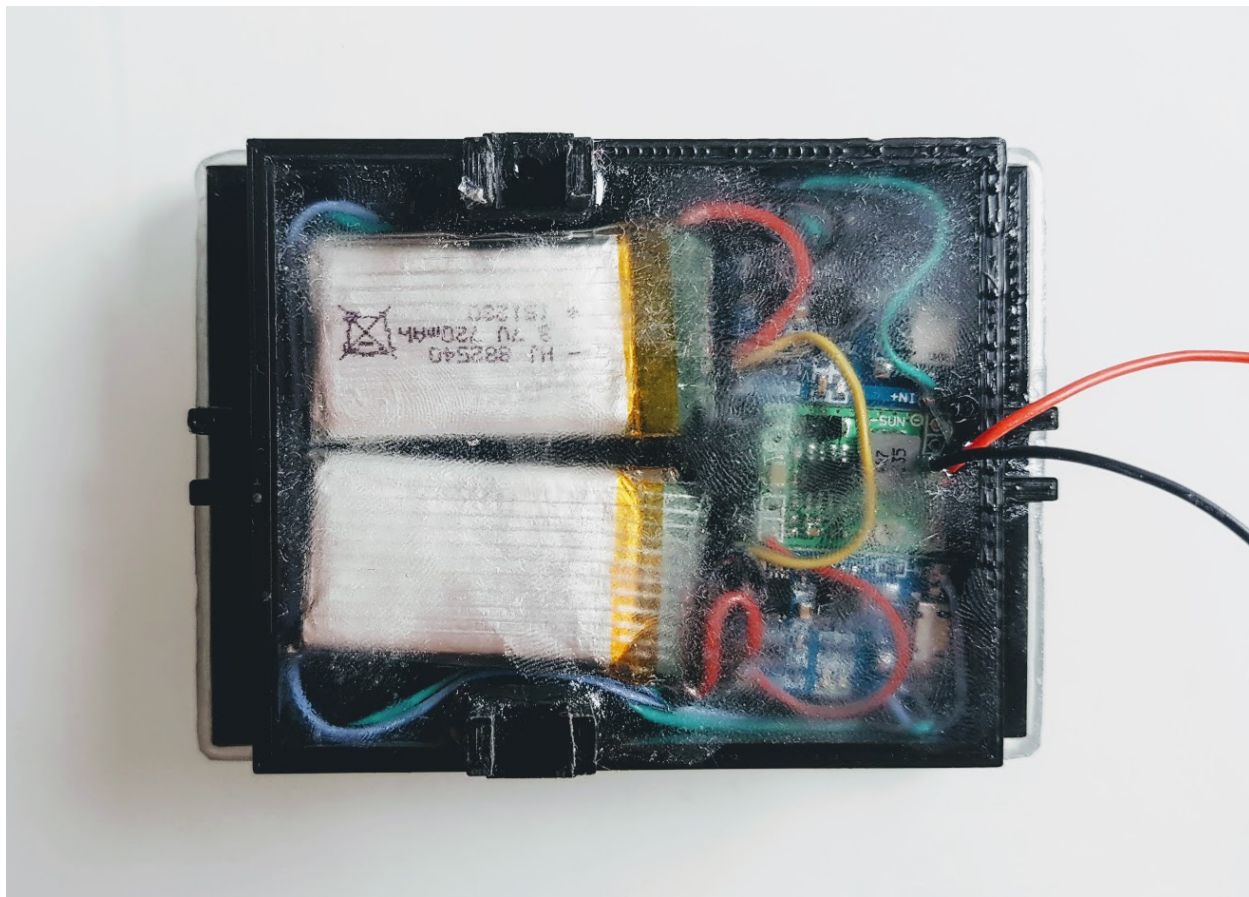


Un cop asseçada la resina ja podem treure la base del motlle i retallar els sobrants amb una serra o trepant petit.



Ja tenim la base d'alimentació a punt!!!





5.3. Xassís Control

8.1 5.3.1. Que necessitem?

8.1.1 5.3.1.1 Material

- 1x Xassís Control (Fitxer STL)
- 1x R2B2-nano-board R2B2-nano
- 1x ESP32 WROVER
- 2x Mòdul diver motor TB6612FNG
- 1x Mòdul Giroscopi/Acceleròmetre MPU6050
- 1x Registència 12k
- 1x Condensador tantalum 10 UF 16 V 106C
- 2x Condensador ceràmic 0.1 UF 104
- 1x Regulador voltage 3.3V
- 1x Connector IPEX/U.FL 20278 1.13mm
- 1x Cable coaxial d'antena 1.13mm i 50ohm de 5metres
- Cable elèctric de silicona 28 AWG
- Estany
- Resina de polièster i catalitzador

8.1.2 5.3.1.2 Eines

- Soldador
- Estenalles d'electrònica

- Guants
- Mascara
- Ulleres de protecció

8.2 5.3.2 Muntatge

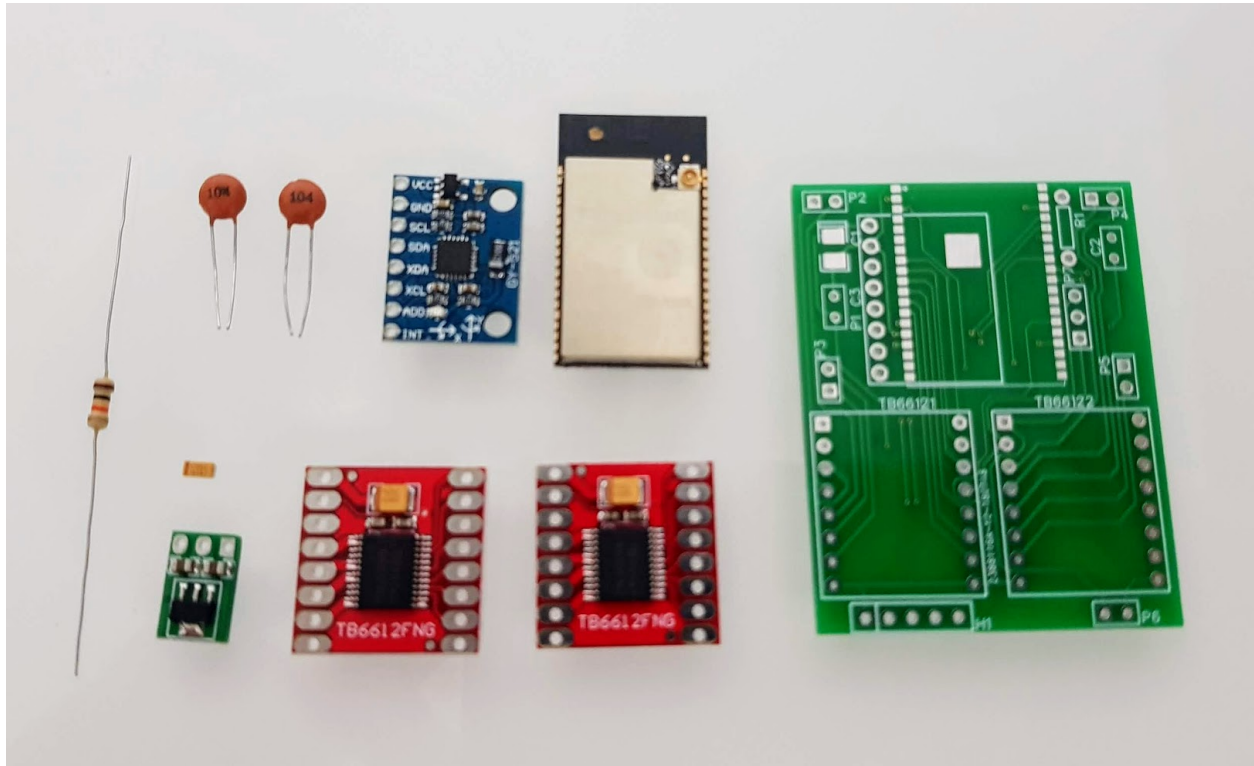
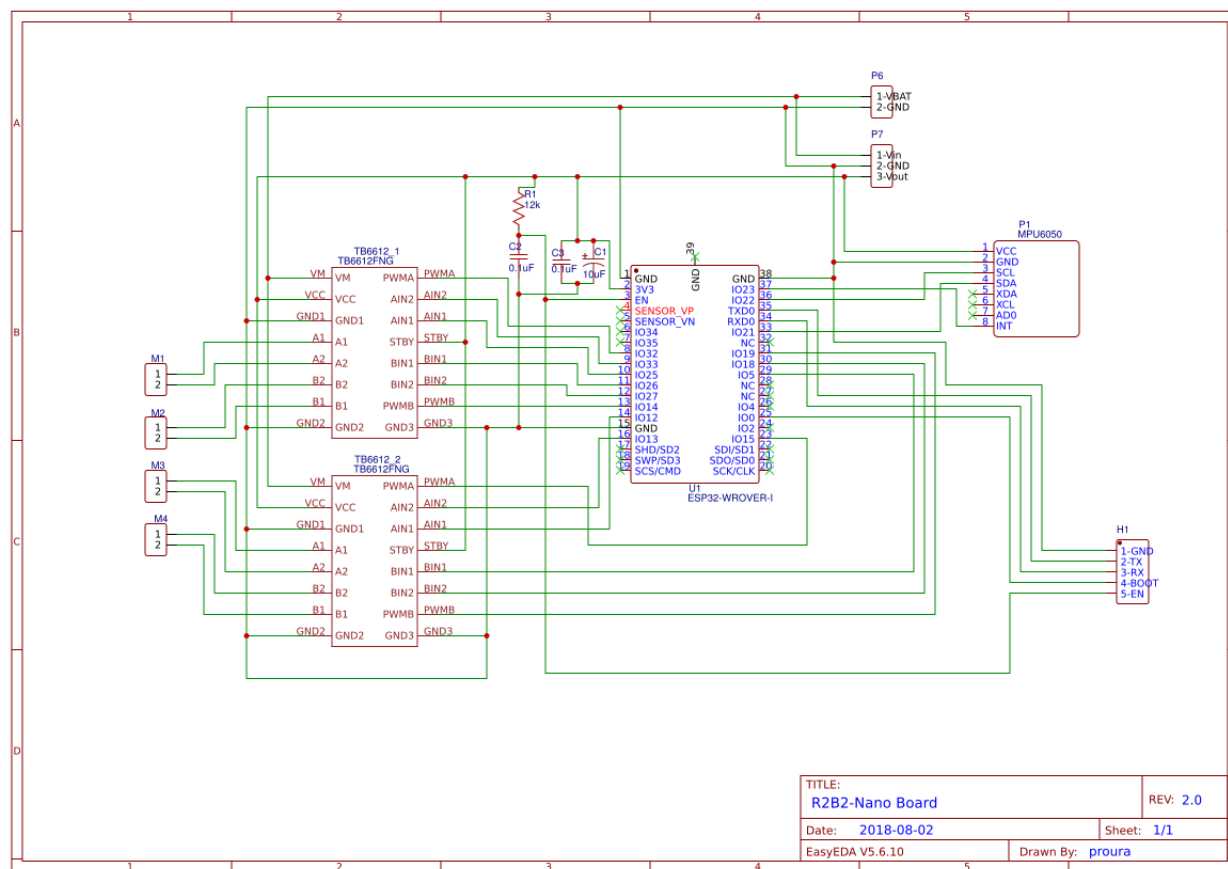


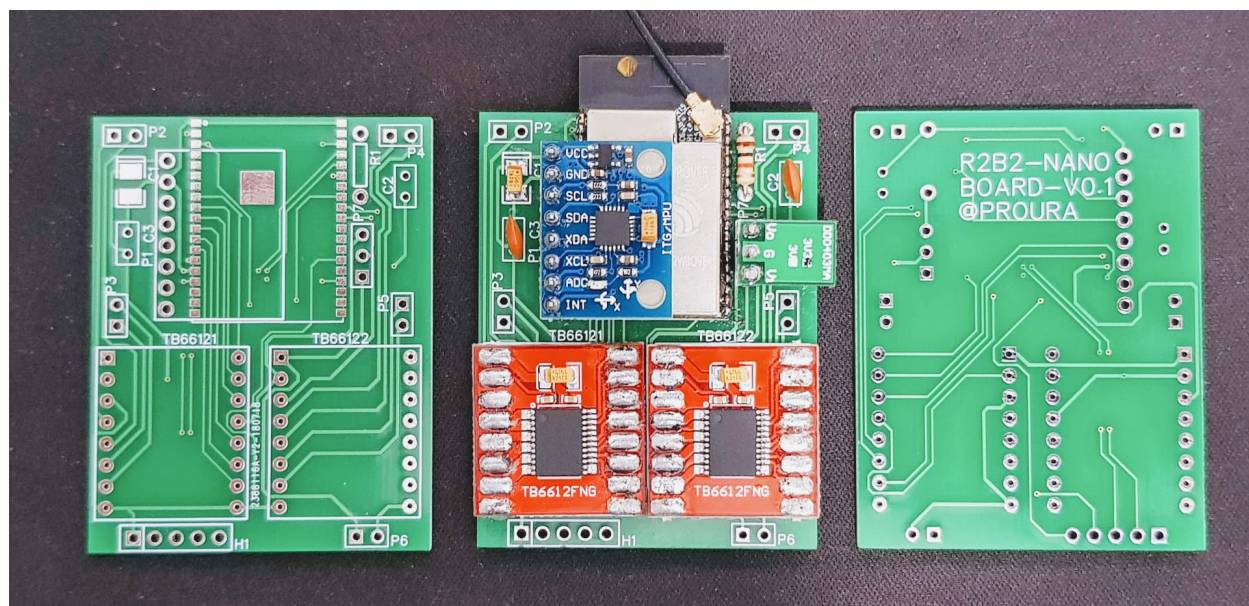
Fig. 1: Material per sla R2B2-nano-board.

Si hi ha cap dubte de on soltar cada component es pot consultar l'esquema electrònic «r2b2-nano-board» al projecte públic [R2B2-nano](#) allotjat a EasyEDA.

A l'apartat de [Electrònica->Control](#) es proposen altres maneres de muntar el sistema de control.



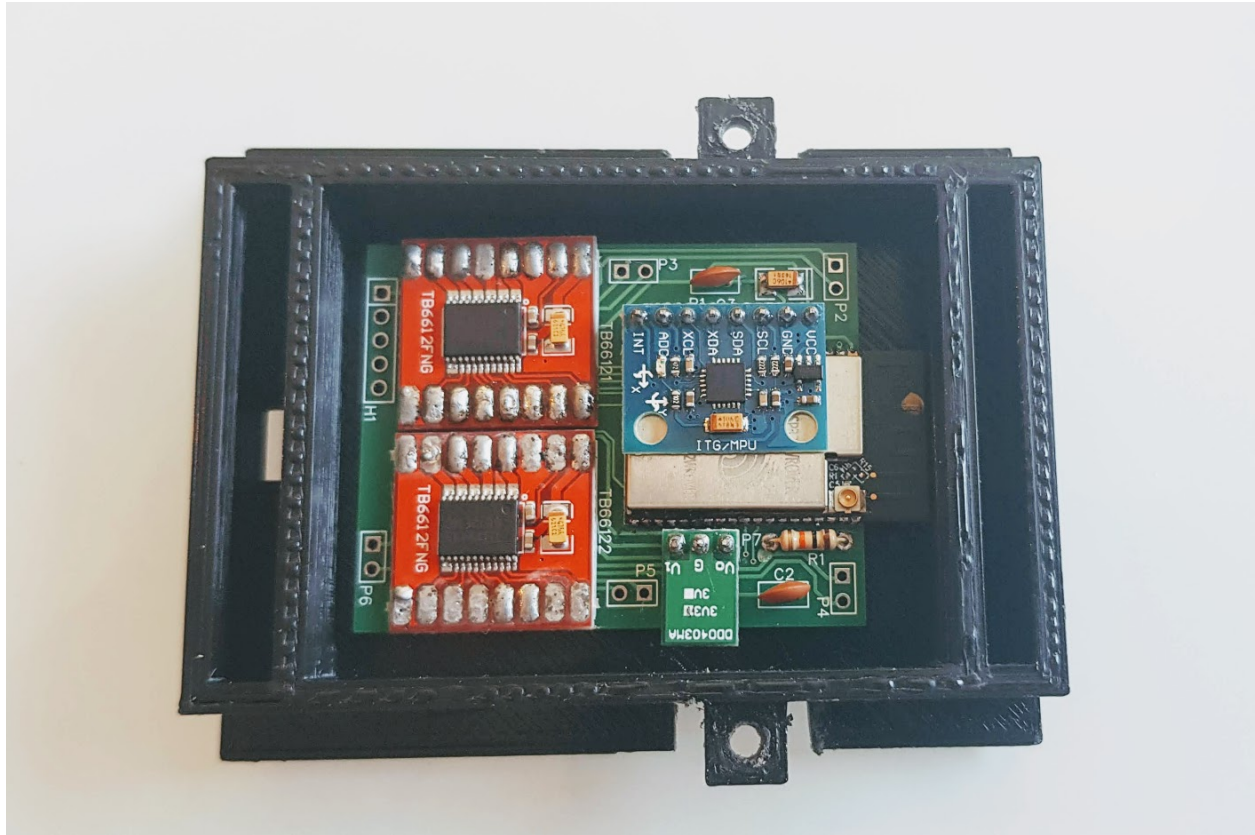
Per a soldar els components a sobre la placa és recomanable començar per l'ESP32 (per facilitar la soldadura d'aquest component és recomanable fer servir algun tipus de pasta de soldadura Flux NC-559 o similars). Llavors podeu anar soldant tots els altres components tal i com és mostren a la següent imatge.



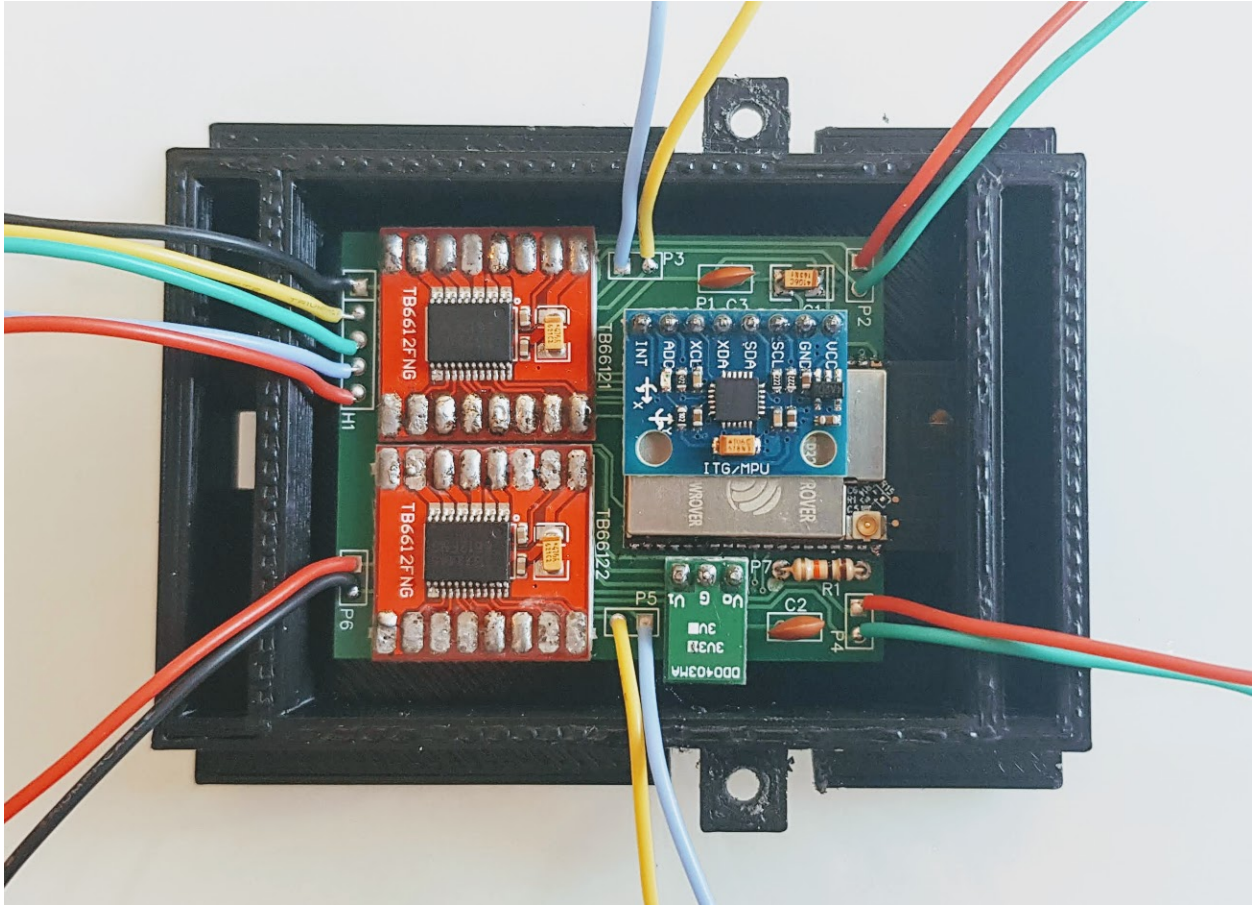
Un cop es tinguin tots els components soldats és convenient comprovar amb un tester que totes les connexions estan ben soldades. Per això posarem el tester que ens avisi amb senyal acústica si hi ha continuïtat i comprovarem els punts

de soldadura seguint l'esquema.

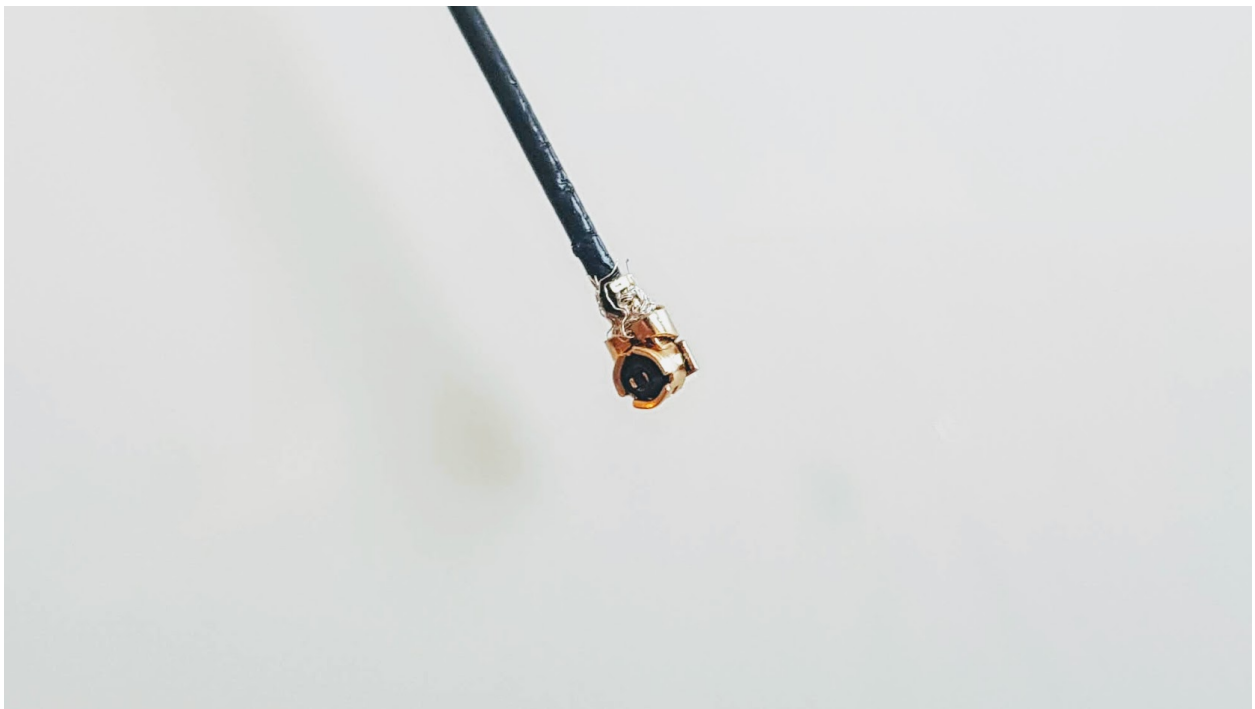
Si tot els punts de soldadura estan bé, alimentarem la placa i farem la primera pujada del Firmware com s'explica a l'apartat de [Software→Firmware](#). Així podrem comprovar que tots els ports a on connectar els motors i el Giroscopi/Acceleròmetre funcionen correctament.



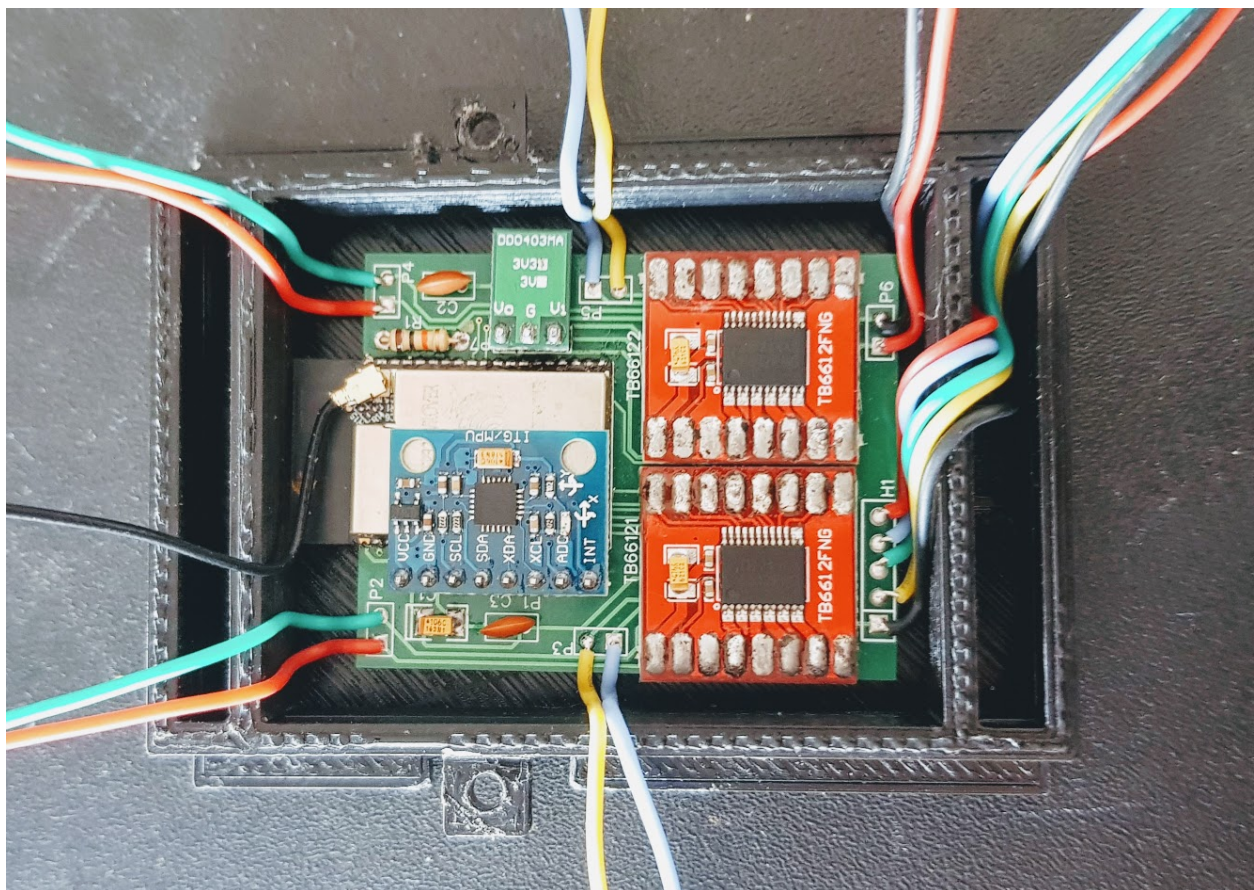
Un cop tinguem la placa de control funcionant ja hi poden soldar tots els cables i posar-la dins el xassís de control.



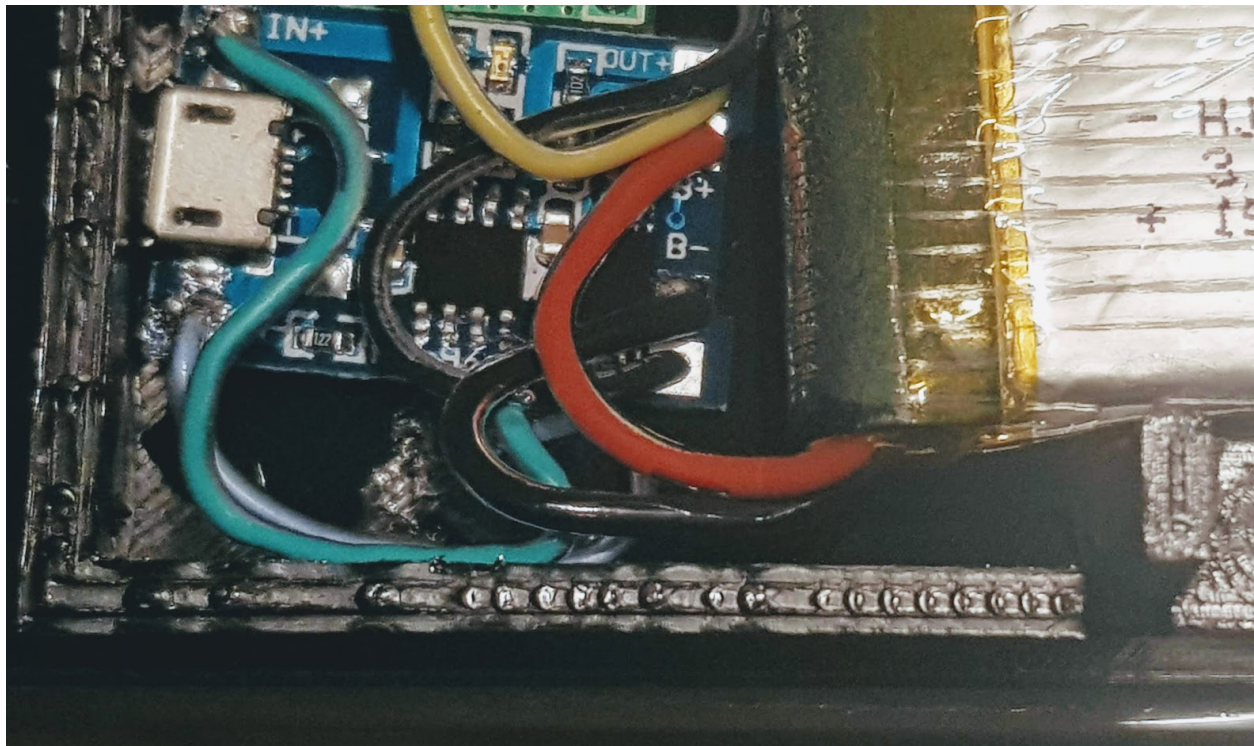
Agafarem el cable coaxial i hi muntarem el connector IPEX.



Connectarem l'antena a la placa i col·locarem tots els cables a punt per enresinar el xassís.



8.2. 5.3.2 Muntatge



5.4. Propulsors

9.1 5.4.1. Que necessitem?

9.1.1 5.4.1.1 Material

- 4x Propulsor Part A (Propulsor A STL)
- 4x Propulsor Part B (Propulsor B STL)
- 4x Hèlix (Hèlix STL)
- 4x Encapsulat Motors (Encapsulat STL)
- 4x Cargols sense cap M2 x 3mm
- 4x Motors Drone (716 Coreless Motor)
- Vaselina
- Cola Tèrmica

9.1.2 5.4.1.2 Eines

- Pistola Cola Tèrmica
- Trepant i broca petita
- Espatula petita
- Clau Allen 0.9mm

9.2 5.4.2 Muntatge

Primer de tot verificarem que els eixos dels motors poden entrar bé dins les hèlixs i que passen pel forat de l'encapsulat dels motors. En cas que sigui necessari farem el forat més ample amb el trepant petit.



Fig. 1: Material per a muntar els Propulsors.

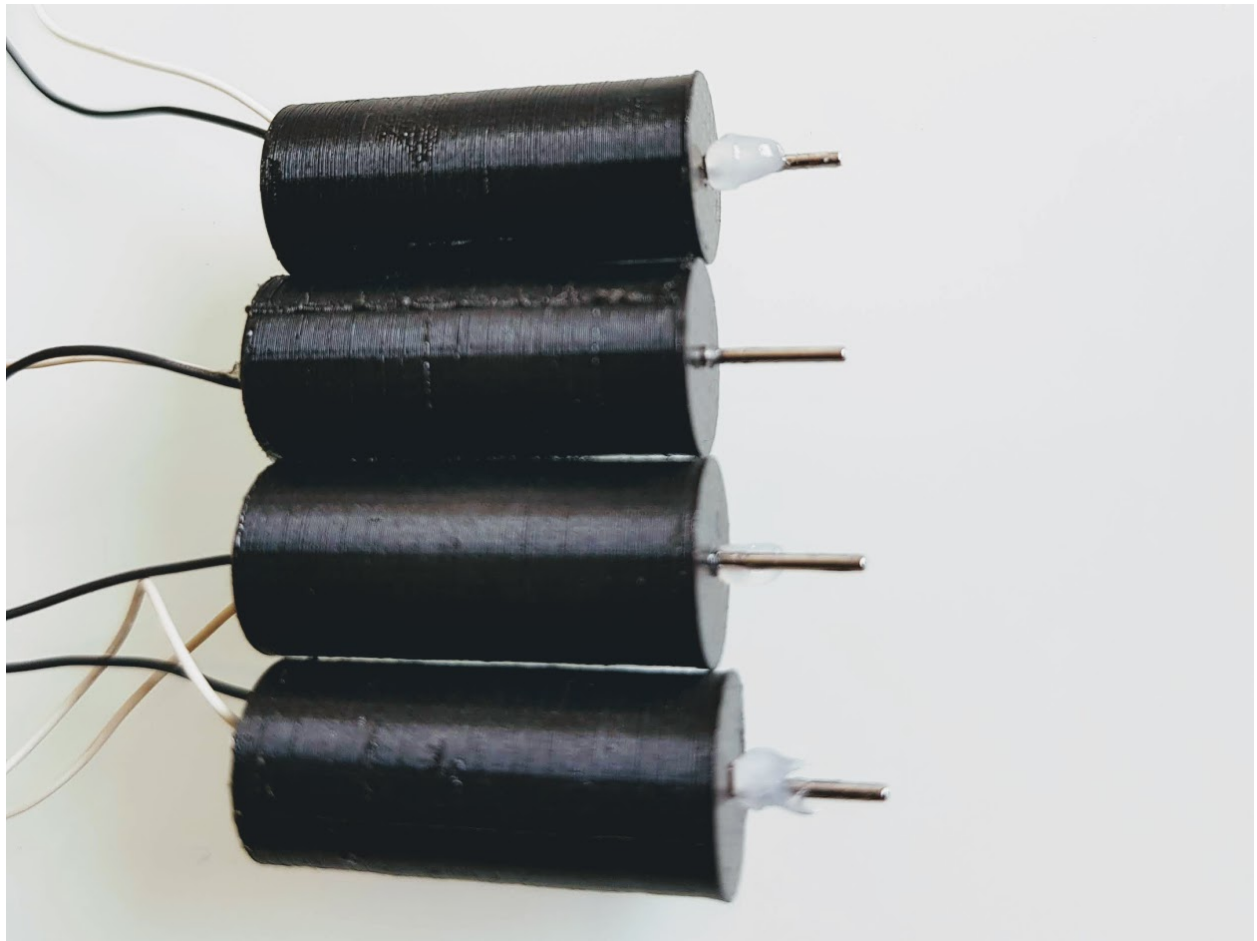


Si el motors no te cap forat a la part d'on surt l'eix, hi posarem la vaselina directament. Si hi ha algun forat el taparem amb cinta adhesiva.

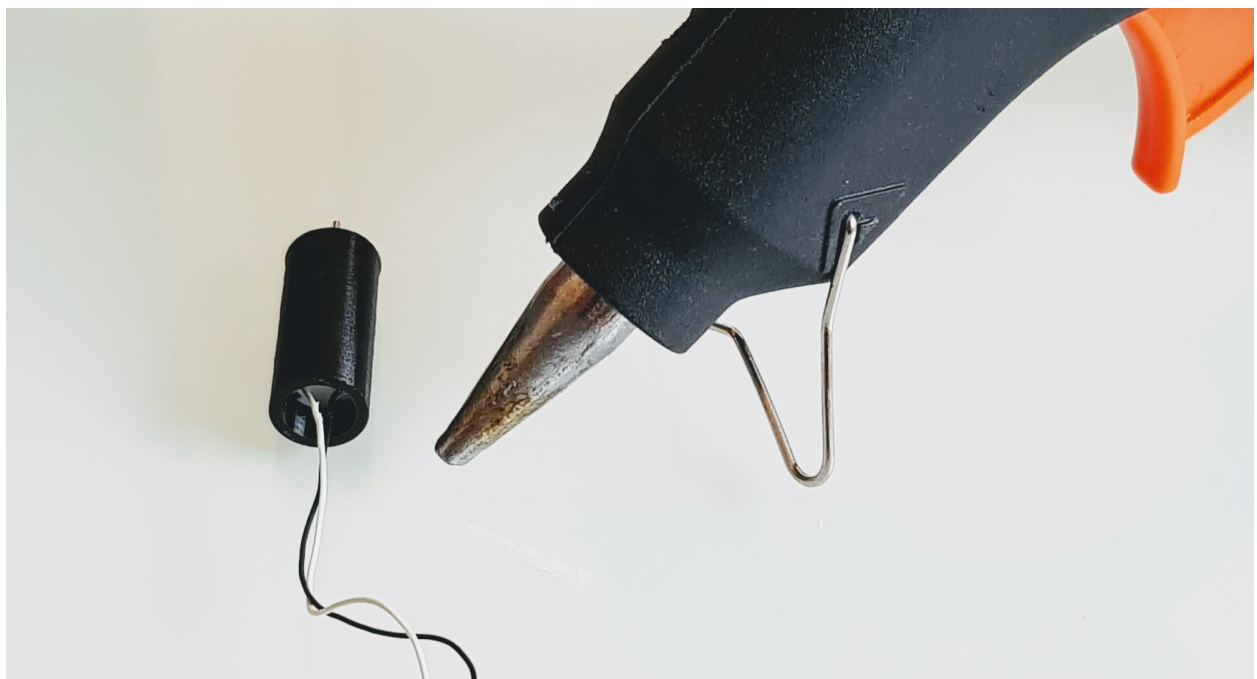
Posant la vaselina entre l'eix i l'encapsulat aconseguirem que es pugui moure lliurement sense que hi entri l'aigua.



A continuació posarem el motor dins l'encapsulat i apretarem fins que surt una mica de vaselina per el forat de l'eix.

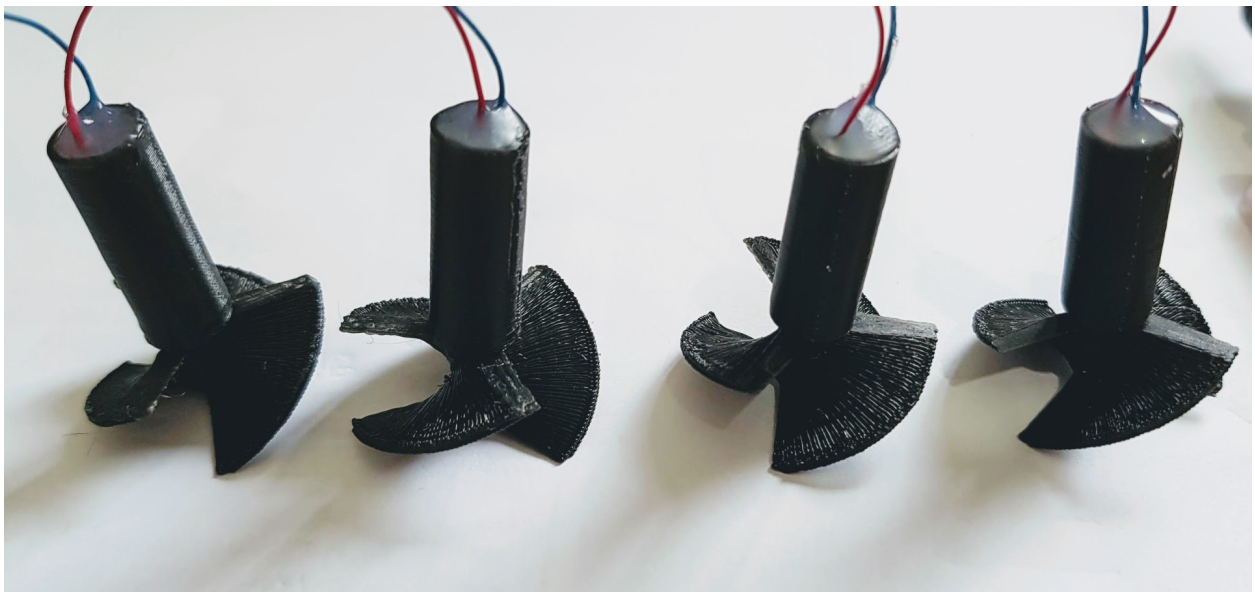


Per evitar que ens entri aigua per la part posterior taparem l'encapsulat del motor amb cola tèrmica.





Ara ja només ens quedarà cargolar les hèlixs amb el cargol sense cap per que no s'escapin i tindrem els propulsors apunt.



10.1 5.5.1. Que necessitem?

10.1.1 5.5.1.1 Material

- 1x Boia Part A (Boia A STL)
- 1x Boia Part B (Boia B STL)
- 2x Cargols M3 16mm
- Escuma FOAM

10.1.2 5.5.1.2 Eines

- Cutter
- Tornavís

10.2 5.5.2 Muntatge

Primer de tot tallarem l'escuma fins que en entri dins la boia.

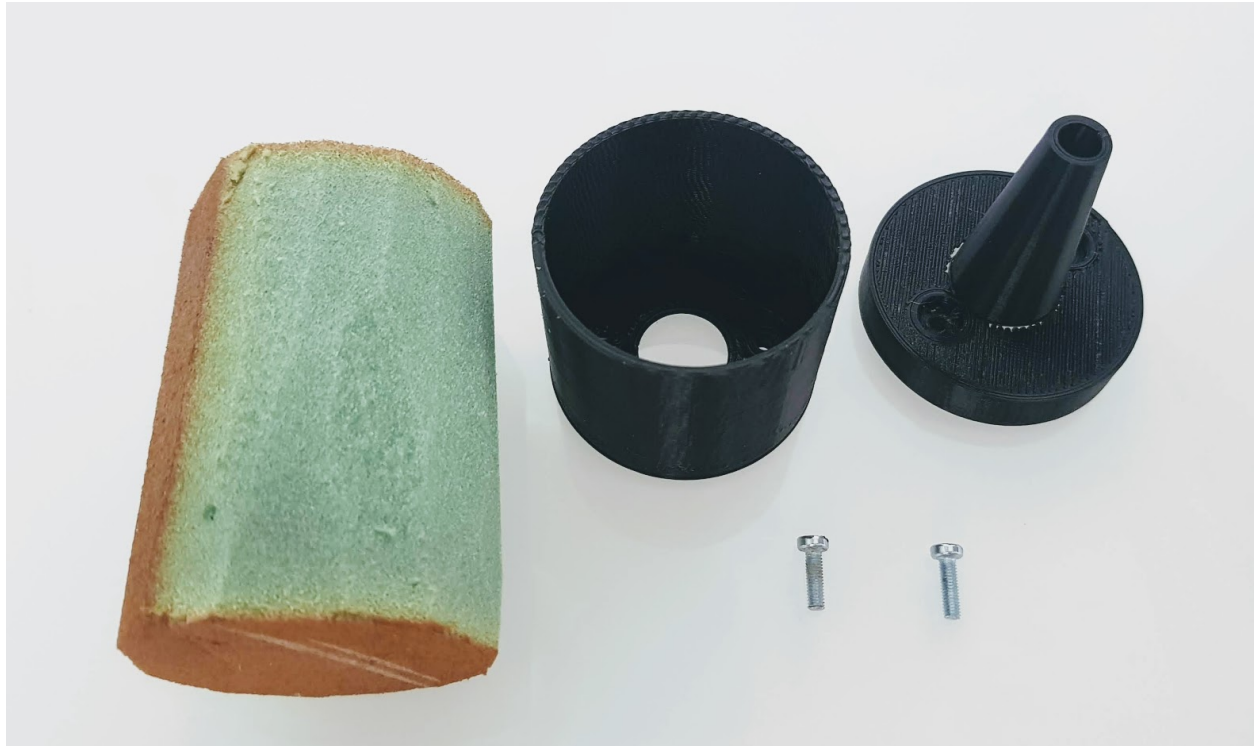
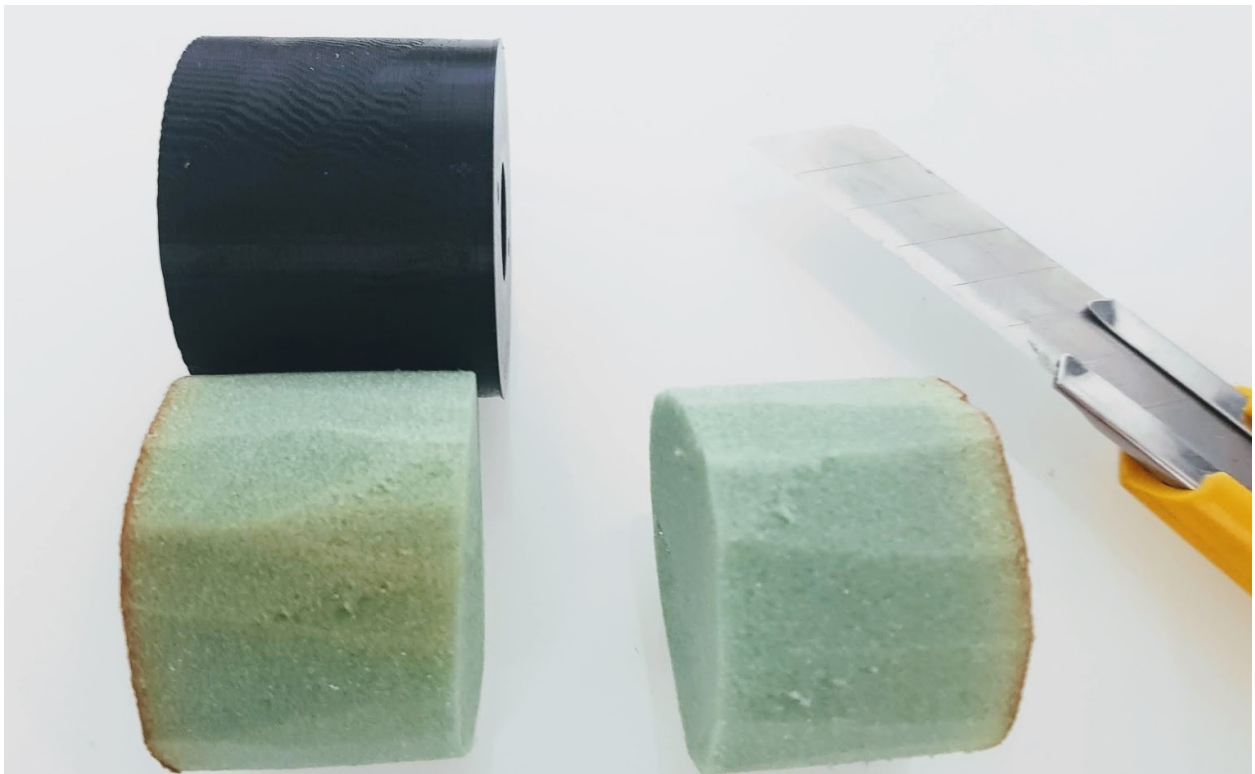


Fig. 1: Material per la boia de comunicacions.



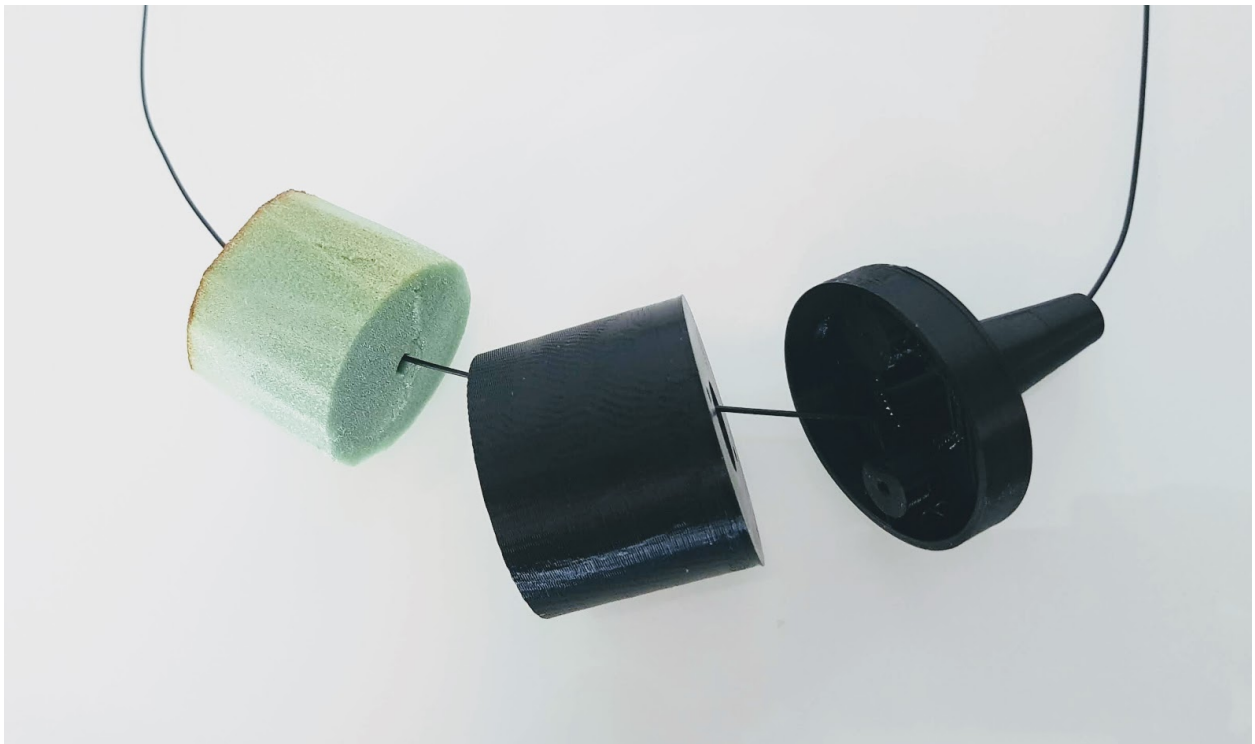




Hi farem un forat per poder passar el cable.



Hi passarem el cable.



Avís: El disseny de la boia no està acabat. Un cop es tingui un model que funcioni correctament és penjarà la documentació.

5.6. Carcassa

11.1 5.6.1. Que necessitem?

11.1.1 5.6.1.1 Material

- Escuma FOAM
- Cola Tèrmica
- Massilla de polièster
- Esprai d'imprimació
- Esprai pintura

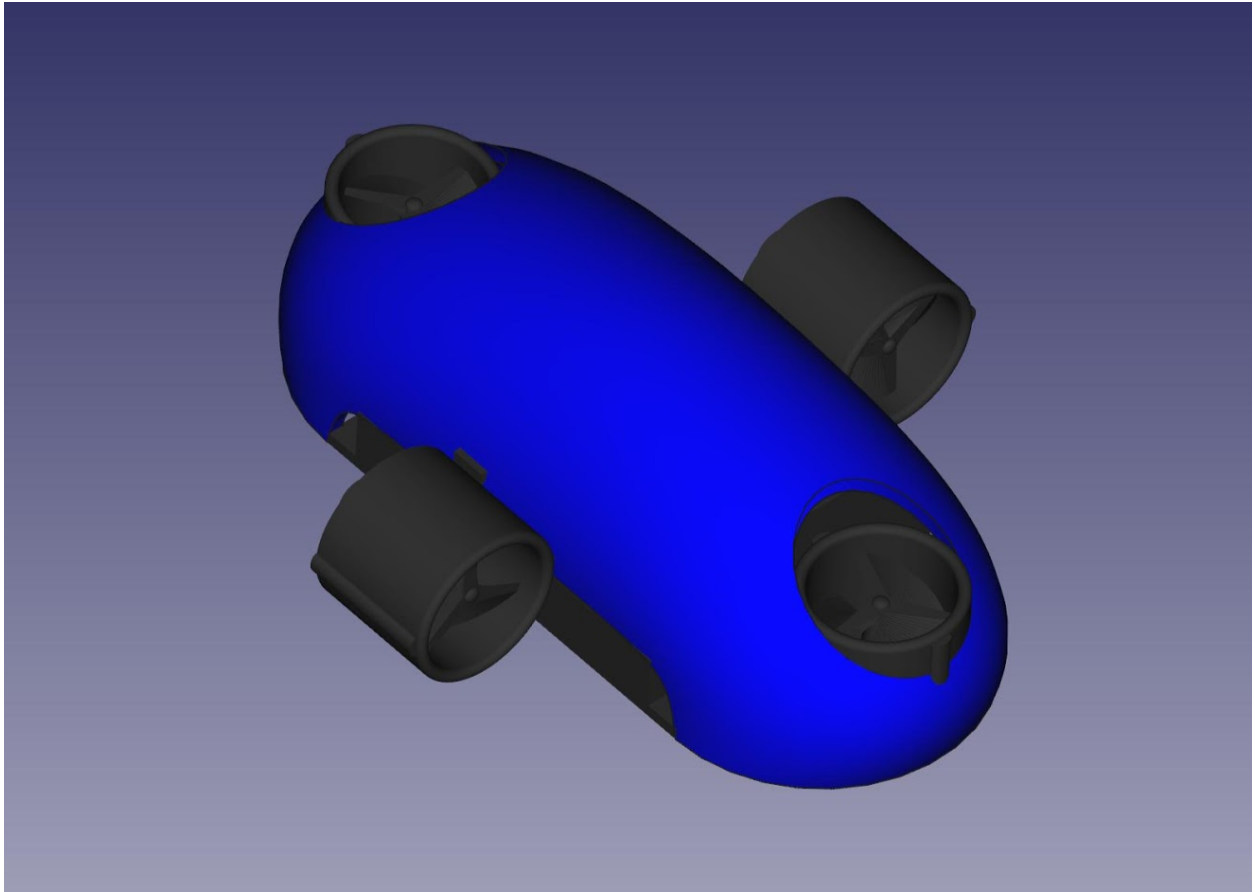
11.1.2 5.6.1.2 Eines

- Cutter
- Trepant amb fresadora
- Pistola Cola Tèrmica
- Guants
- Mascara
- Ulleres de protecció
- Llimes/Paper de vidre

11.2 5.6.2 Muntatge

El disseny de la carcassa és completament lliure i la seva finalitat està explicada a l'apartat de [Disseny](#)—>Carcassa.

A continuació veurem com s'ha treballat la proposta de carcassa que es motra en la següent imatge.



S'ha agafat un tall d'escuma FOAM i s'ha marcat la línia a seguir per la carcassa.



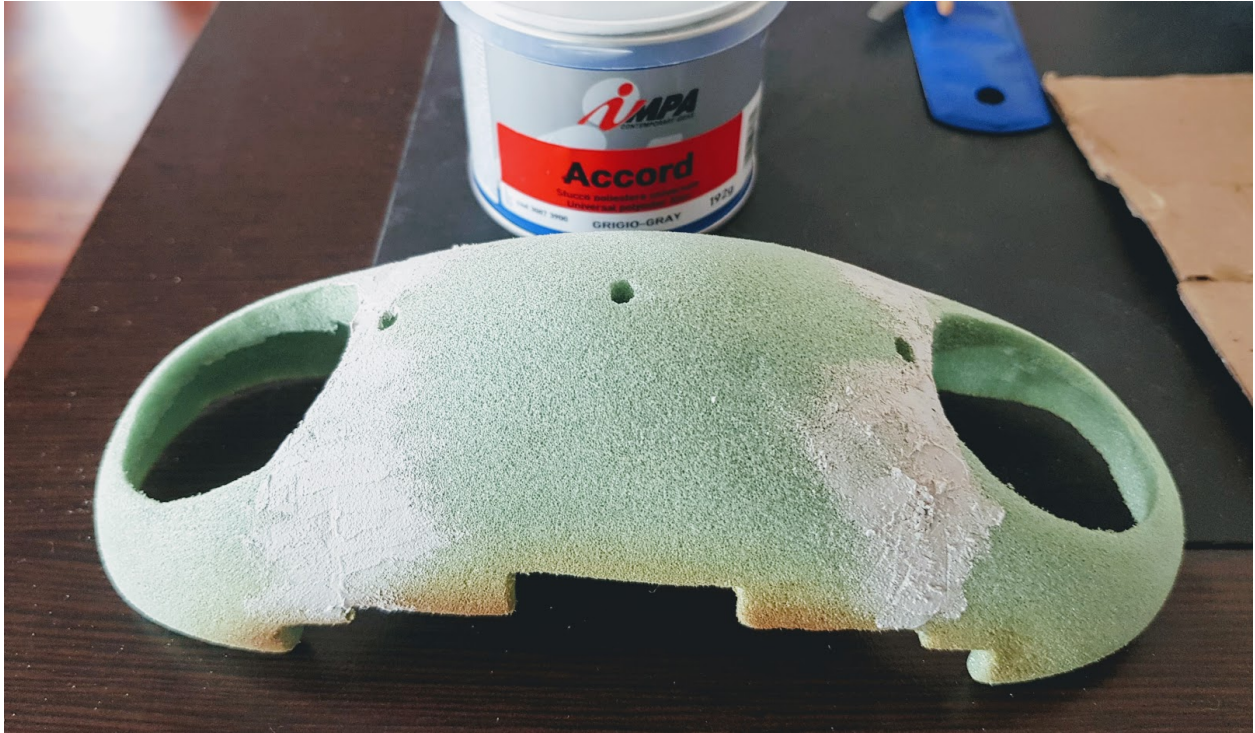
S'ha anat retallant i buidant amb el cutter i el trepant petit amb la fresadora.



S'ha fet un allisat més fi amb paper de vidre.



S'han fet forats per que no s'hi acumuli aire a l'entrar dins l'aigua.



Finalment s'hi ha fet unes capes d'imprimació i pintura.



5.7. Acoblament

12.1 5.7.1. Que necessitem?

Avís: Per poder realitzar el muntatge cal tindre tots els passos anteriors realitzats.

12.1.1 5.7.1.1 Material

- 2x protecció Port Serie (Fitxer STL)
- 4x Cargols M3x16 Inox
- 2x Cargols M3x6 Inox
- 4x Volandera M3
- 2x Roca femella M3
- 2x Passador DuPont
- 1x Connector mascle DuPont
- 1x Connector femella DuPont
- Funda termoretràctil
- Cola Tèrmica
- Baselina
- Estany
- Brides petites

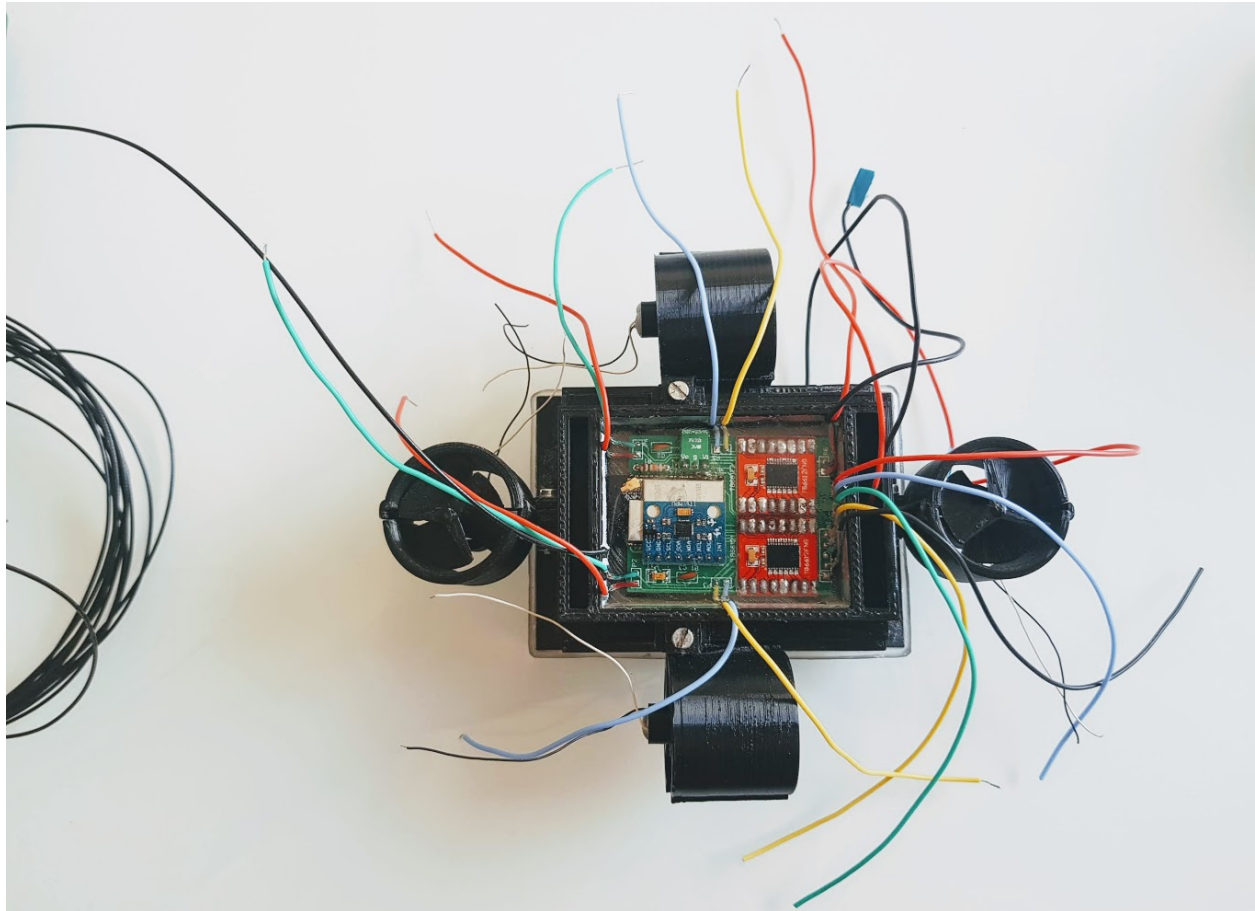
12.1.2 5.7.1.2 Eines

- Estenalles d'electrònica
- Pistola Cola Tèrmica
- Tornavís
- Soldador

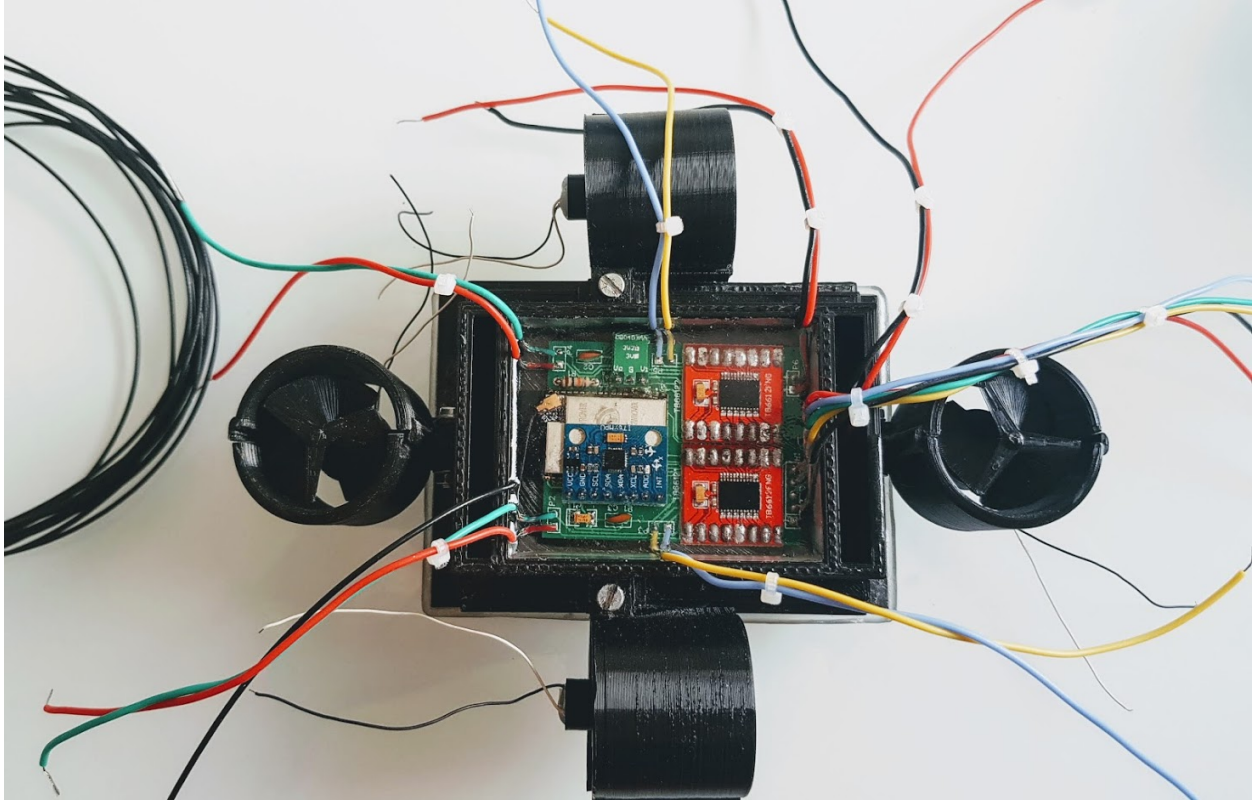
12.2 5.7.2 Muntatge

Si estem en aquest punt vol dir que tenim totes les parts anteriors apunt. Primer agafarem els propulsors i els cargolarem a la base. Per això caldrà fer passar els cables d'alimentació del xassís d'alimentació pel forat que hi ha al xassís de control.

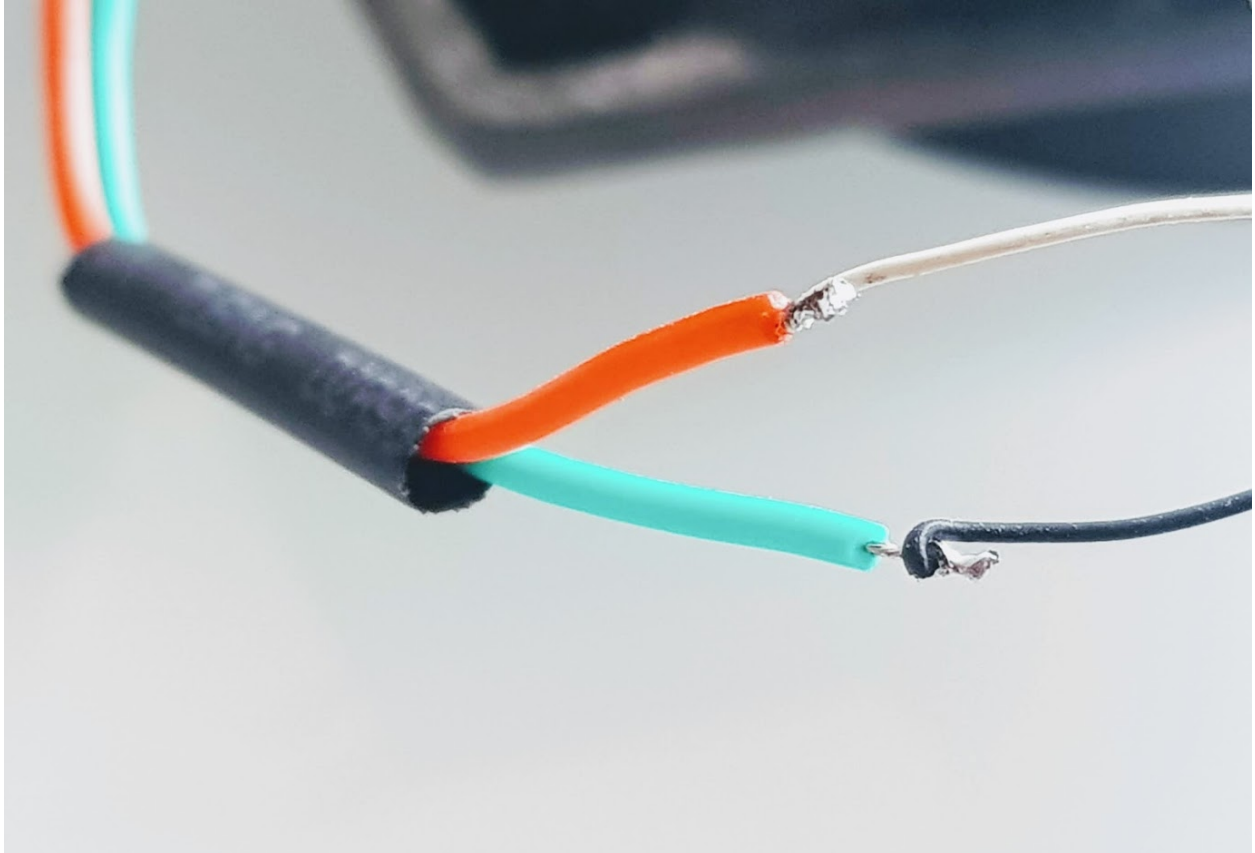
Si ho hem fet tot correctament ens quedarà l'R2B2-nano de la següent manera.



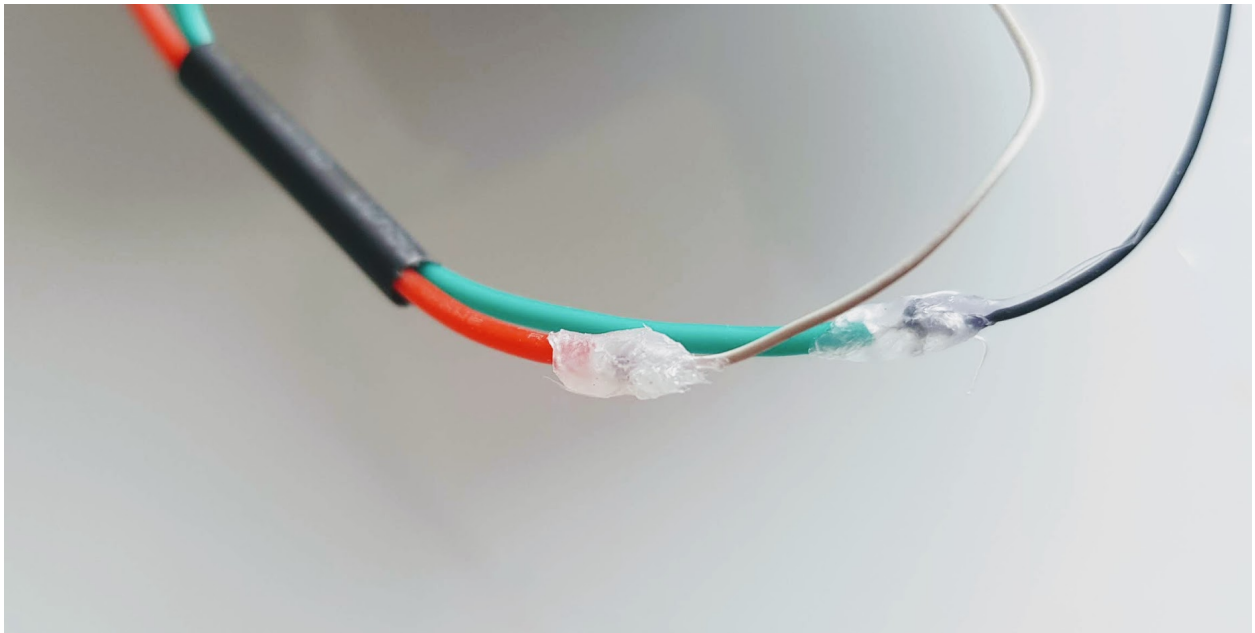
Amb brides petites enparellarem els cables que surten de la placa de control i els del xassís d'alimentació per posar una mica d'ordre i poder treballar millor.



Per cada parell de cables dels motors hi passarem una funda termoretràctil i llavors soldarem els dos cables als ports dels motors de la placa de control (v0.2: M1, M2, M3 i M4 o v0.1: P2, P3, P4 i P5).



Per assegurar-nos de que sigui impermeable i no es curtcircuiti cobrirem les soldadures amb cola tèrmica, llavors hi posarem la funda termoretràtil a sobre i hi aplicarem calor pq s'hi ajusti, però sense excedir-nos ja que podem fondre la cola tèrmica i creuar els cables del motor.



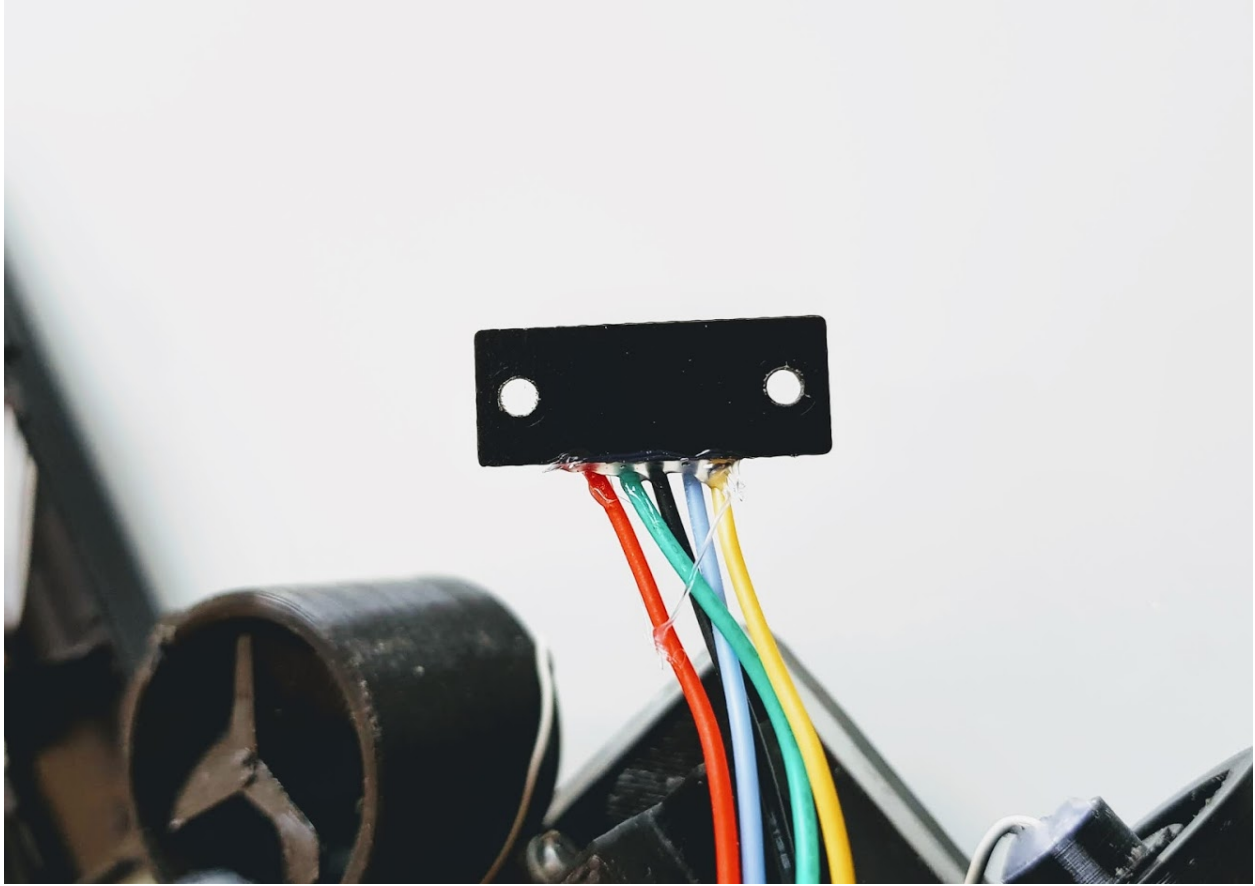
Farem el mateix procediment per unir el cable de massa que surt de la base d'alimentació amb el cable de massa del xassís de control. Primer passarem la funda tèrmica, soldarem, recobrirem amb cola tèrmica i taparem amb la funda termoretràtil.



Per el cable d'alimentació restant hi muntarem un terminal DuPont mascle i un femella. Així podrem fer servir-lo com a interruptor.



Ara els únics cables que ens queden a l'aire són els cables pel port de programació. Com que pràcticament no farem servir aquest port, només si tenim algun problema amb el WiFi o Firmware, l'encapsularem dins el protector del port.



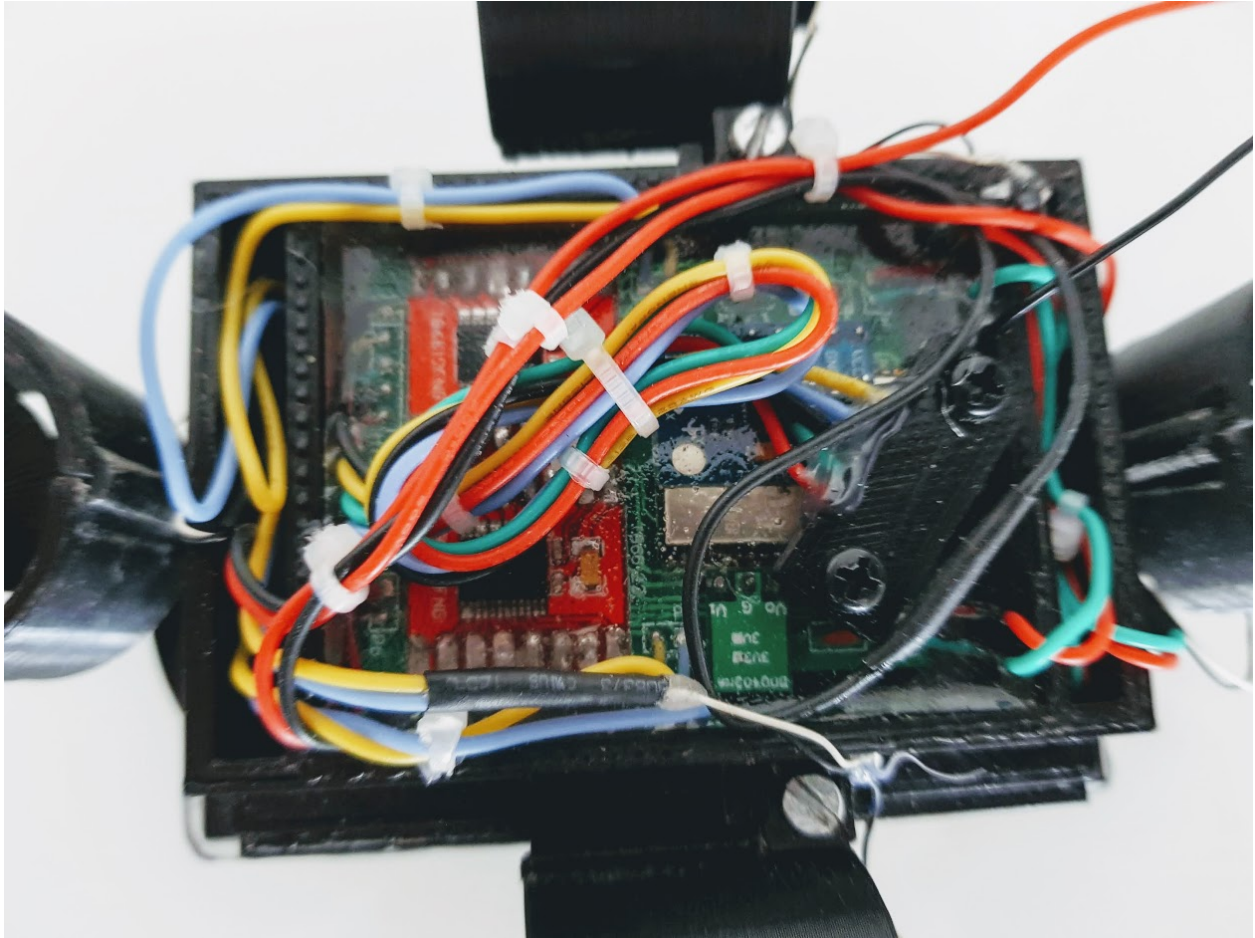
Agafarem una de les dues parts de protector del port, l'omplirem de vaselina i hi posarem els cables.



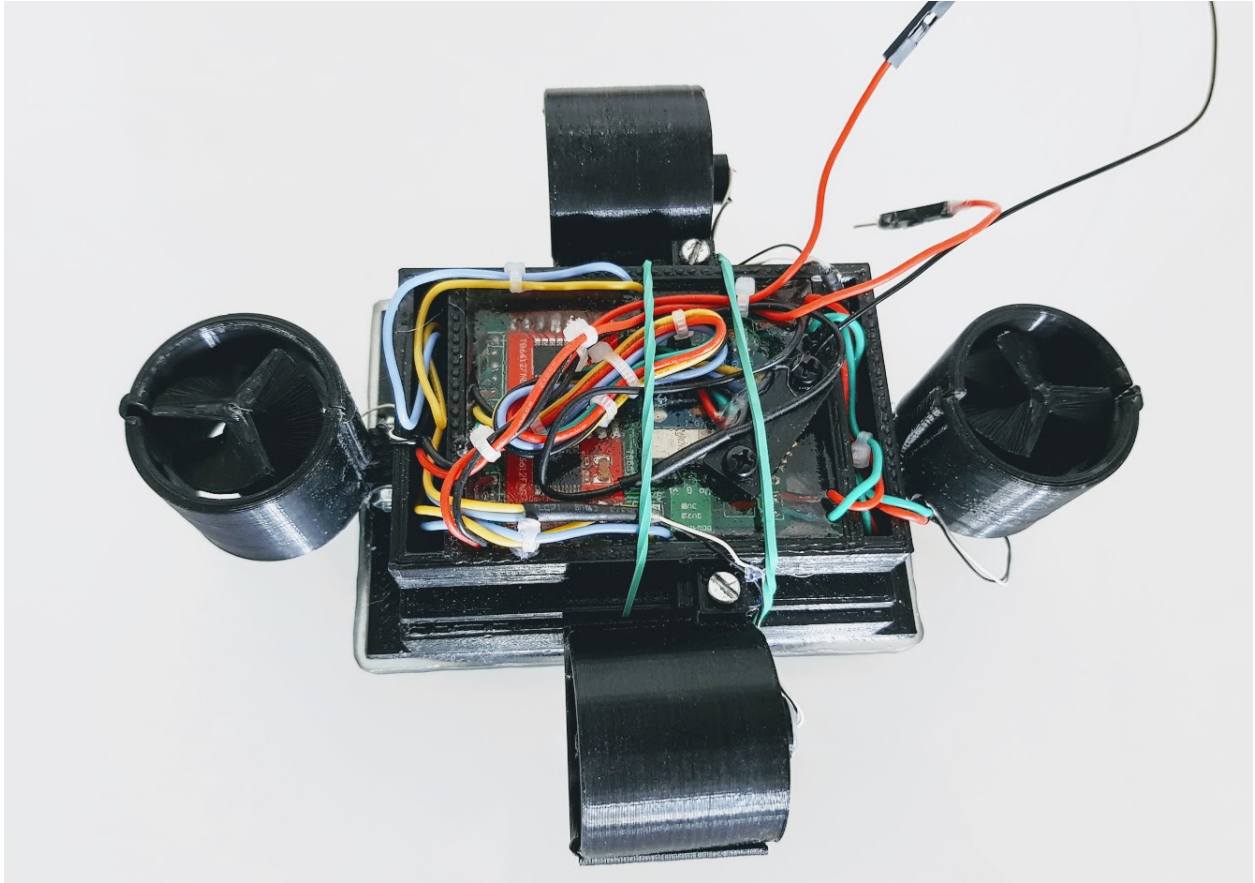
Ara agafarem l'altre part del protector i el cargolarem al damunt. Caldrà netejar amb un drap els excedents de vaselina que ens sortiran pels laterals del protector del port.



Un cop tenim tots els cables soldats i protegits amb ajuda de cola tèrmica podem anar col·locant tots els cables en ordre.



Podem fer servir una goma elàstica per subjectar tots els cables.



Ja podem posar la carcassa a sobre.



Per subjectar la carcassa podem fer servir una altre goma elàstica.



13.1 A.1.1. Cargols



Fig. 1: Cargols, femelles i arandeles M3.



Fig. 2: Cargol sense cap M2x3mm.







13.2 A.1.1. Cable Silicona 28AWG

13.3 A.1.1. Connectors Dupont

13.4 A.1.1. Cable Coaxial

13.5 A.1.1. Estany

13.6 A.1.1. Resina Polièster

13.7 A.1.1. Funda Termoretractil

13.8 A.1.1. Cola Tèmica









CAPÍTOL 14

A.2. Eines

14.1 A.2.1. Pistola de cola tèrmica

14.2 A.2.2. Soldador





CAPÍTOL 15

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)