



FACULTAT  
DE CIÈNCIES I TECNOLOGIA

UVIC | UVIC·UCC

UNIVERSITAT DE VIC

TREBALL FINAL DE GRAU  
GRAU EN ENGINYERIA MECATRÒNICA

---

# Disseny, construcció i control d'un robot auto equilibrat

---

Autor: Jordi Soldevila Casas

Tutor: Dr. Moisès Serra Serra

Juny de 2021

# Agraïments

Dono les gràcies al professor Moisès Serra per portar la tutoria d'aquest projecte i estar en tot moment involucrat en el seu desenvolupament. El seu interès i el seu coneixement han estat importants, igual que la seva motivació per l'ensenyament, l'electrònica, la programació i la robòtica.

Agraeixo també les ajudes, opinions i consells del professor de taller Jordi Serra, qui sempre ha estat predisposat a dedicar-me estones del seu temps quan ho he necessitat.

Altrament donar les gràcies a la meva família per fer-me costat al llarg d'aquest quatre anys, i fer menys feixuga la tasca que suposa compaginar feina i estudi.

Per últim, donar les gràcies als meus companys i companyes del grau d'enginyeria mecatrònica, en especial al més pròxims amb els quals hem treballat colze a colze i ens hem ajudat mútuament en moltes ocasions al llarg d'aquests anys.

# Resum

El robot auto equilibrat busca solucionar el comportament del pèndol invertit. Problema habitualment plantejat en el camp del control, ja que s'aconsegueix establir un sistema inestable gràcies a l'acció generada després de la lectura dels seus sensors d'entrada.

El present projecte presenta el disseny, la fabricació i el control d'un robot mòbil capaç de mantenir l'equilibri sobre les seves dues rodes, mitjançant un control digital LQR a través d'un microcontrolador. El sistema compensador LQR és un algoritme matemàtic format per equacions d'estat, basades en les propietats dinàmiques i cinemàtiques del sistema.

El procés comença per comprendre el principi de funcionament d'aquest tipus de robot. S'estudia la seva cinemàtica i dinàmica, i es determinen quins tipus de sensors i actuadors seran necessaris per implementar el sistema. Seguidament es realitza el disseny CAD amb el programa Inventor. Les plaques que formen l'estructura es mecanitzen en una fresadora CNC. Per la part elèctrica, es fa un disseny del circuit utilitzant el programa Fritzing, i es fabrica una placa on es muntan els diferents dispositius i sensors. Es compra el material necessari i es construeix el robot. Finalment, amb Arduino es programa el codi que permet l'auto equilibratge a través del seu control LQR.

Els resultats obtinguts han complert els objectius plantejats a l'inici del projecte. Per la part mecànica s'ha aconseguit un disseny simple, robust, lleuger, i fàcil de muntar i desmuntar. La part electrònica permet la comunicació dels diferents dispositius i la correcta alimentació del sistema. Pel que fa el control, s'ha implementat el sistema LQR aconseguint l'auto equilibrat del robot.

# Abstract

The self balancing robot seeks to solve the behavior of the inverted pendulum. A problem commonly used in the field of control, as it manages to stabilize an unstable system thanks to the action generated after reading its input sensors.

This project presents the design, manufacture and control of a mobile robot capable of maintaining balance on its two wheels by a digital LQR control through a microcontroller. The LQR compensation system is a mathematical algorithm formed by equations of state, based on the system dynamic and kinematic properties.

The process begins with understanding the principle of operation of this type of robot. Its kinematics and dynamics are studied, and it is determined what types of sensors and actuators will be needed to implement the system. The CAD design is done with the Inventor program. The manufacture of the plates that form the structure are machined in a CNC milling machine. On the electrical side, a circuit design is made using the Fritzing program, and a board is made where the different devices and sensors are mounted. The necessary material is bought and the robot is built. Finally, the software is implemented through Teensyduino, which allows self balance through its LQR control.

The results obtained are the ones proposed at the beginning of the project. On the mechanical side, a simple, robust, and lightweight design has been achieved, and it's easy to assemble and disassemble. The electronic part allows the communication of the different devices and the correct feeding of the system. In terms of control, the LQR system has been successfully implemented, achieving the self balancing of the robot.



# Índex

<b>Glossari</b>	<b>9</b>
<b>Acrònims</b>	<b>10</b>
<b>Capítol 1</b>	
<b>Introducció</b>	<b>11</b>
1.1 Història	12
1.2 Estat de l'art	13
1.3 Objectius	13
1.4 Estructura de la memòria	14
<b>Capítol 2</b>	
<b>Modelització matemàtica</b>	<b>15</b>
2.1 Propietats del pèndol invertit	15
2.2 Cinemàtica del sistema	17
2.2 Dinàmica del sistema	19
<b>Capítol 3</b>	
<b>Disseny mecatrònic</b>	<b>24</b>
3.1 Disseny mecànic	24
3.1.1 Bases de l'estructura	25
3.1.2 Rodes	26
3.1.3 Separadors	26
3.2 Disseny electrònic	27
3.2.1 Teensy 4.0	28
3.2.2 MPU-6050	29
3.2.3 Motors DC i encoder	30
3.2.4 Controlador TB6612FNG	31
3.2.5 Bluetooth HM10	32
3.2.6 Ultrasons HC-SR04	33
3.2.7 Reguladors de tensió	34
3.2.8 Bateria	34
3.2.9 Connexions dels dispositius	35
3.3 Sistema de control	38
3.3.1 Regulador quadràtic lineal (LQR)	38
3.3.2 Principis matemàtics del LQR	38
3.3.3 Càlcul dels valors de K	39
3.4 Programació	41

<b>Capítol 4</b>	
<b>Cost econòmic del prototip</b> .....	<b>42</b>
4.1 Desglos dels costos .....	42
<b>Capítol 5</b>	
<b>Resultats</b> .....	<b>44</b>
<b>Capítol 6</b>	
<b>Conclusions</b> .....	<b>46</b>
<b>Bibliografia</b> .....	<b>47</b>
<b>Apèndixs</b>	
A. Programa .....	48
B. Plànols .....	55
C. Esquema elèctric .....	60

# Índex de figures

Figura 1.1: Robot creat per Kazuo Yamafuji l'any 1986.....	12
Figura 2.1: Esquema bàsic de qualsevol sistema .....	15
Figura 2.2: Esquema del pèndol invertit .....	16
Figura 2.3: Gràfic del model cinemàtic .....	18
Figura 2.4: Variables del pèndol invertit .....	19
Figura 3.1: Disseny CAD .....	24
Figura 3.2: Mecanització de la placa base del robot .....	25
Figura 3.3: Rodes del robot .....	26
Figura 3.4: Separadors de llautó .....	26
Figura 3.5: Esquema bàsic del sistema electrònic.....	27
Figura 3.6: Teensy 4.0 .....	28
Figura 3.7: MPU-6050.....	29
Figura 3.8: Motor reductor DC de 12V amb encoder. ....	30
Figura 3.9: Senyals A i B d'un encoder.....	30
Figura 3.10: Driver TB6612FNG .....	31
Figura 3.11: Mòdul Bluetooth HM-10.....	32
Figura 3.12: Sensor d'ultrasons HC-SR04 .....	33
Figura 3.13: Regulador de voltatge .....	34
Figura 3.14: Bateria Lipo de la marca Ovonic .....	34
Figura 3.15: Esquema de connexions .....	37
Figura 3.16: Esquema sistema LQR.....	38
Figura 3.17: Diagrama de funcionament del programa .....	41
Figura 5.1: Prototip .....	44

# Índex de taules

Taula 2.1: Forces i paràmetres d'un robot auto equilibrat .....	20
Taula 3.1: Descripció de les connexions dels dispositius .....	36
Taula 3.2: Taula de paràmetres per calcular els valors de K .....	39
Taula 4.1: Costos del prototip.....	43

# Glossari

**Cinemàtica.** Branca de la física que descriu els moviments sense tenir en compte les forces externes. 3, 5, 15, 17, 18

**Dinàmica.** Branca de la física que descriu els moviments a partir de les forces externes. 3, 5, 11, 15, 16, 19

**Drift.** els valors no són constants en el temps sinó que aquests tendeixen a patir derives. 29

**Compensador.** Aparell que compensa una diferència o variació. 3, 10, 15, 29

**Driver.** Expressió anglesa per referir-se a un sistema destinat a controlar-ne un altre. 7, 27, 31, 34 - 36

**Microcontrolador.** Circuit integrat programable, que inclou blocs funcionals que compleixen una funció en específic. 3, 12, 14, 27, 28, 31, 32, 34, 41

**Self balancing robot.** Robot auto equilibrat. 11

**Pont H.** Circuit electrònic que serveix per controlar motors de corrent continua. 31

**Inventor.** Programa d'Autodesk de dibuix 3D. 3, 24

**Fritzing.** Programa de disseny electrònic. 3, 37

**Control.** Camp interdisciplinari entre l'enginyeria i les matemàtiques que estudia el comportament de sistemes dinàmics. 6, 13, 24 - 26, 30

# Acrònims

**DC** Corrent contínua. 5, 7, 9, 10, 30, 31, 41

**CAD** Disseny assistit per ordinador. 7, 13, 24

**CPU** Unitat central de control. 28

**PID** Compensador proporcional, integral i derivatiu. 13, 38

**LQR** Regulador quadràtic lineal. 3, 13, 14, 29, 38

**IMU** Unitat de mesura inercial. 27, 29, 35

**DMP** Processador de moviment digital. 29

**MOSFET** Transistor d'efecte de camp. 31

**BJT** Transistor bipolar. 31

**GND** Terra. 35, 36

**PWM** Modulació per ample de polsos. 31, 35, 36

**RST** Controlador per assignació de pols. 38

# Capítol 1

## Introducció

Els robots són màquines autònomes que integren components electrònics i mecànics, programats per a realitzar una tasca determinada. Els robots auto equilibrats, coneguts en anglès com a *self balancing robots*, són robots mòbils capaços de mantenir l'equilibri sobre dues rodes i girar in situ. Aquesta maniobrabilitat addicional permet una fàcil navegació per diversos terrenys, girar cantonades pronunciades i recórrer petits passos o vorades, capacitats que tenen el potencial de resoldre diversos reptes de la indústria i la societat. La singularitat i l'àmplia aplicació de la tecnologia derivada d'aquest sistema inestable ha atret l'interès de moltes investigacions i entusiastes de la robòtica de tot el món

Tot i que els robots humanoides superen els obstacles, són més complexos de dissenyar i controlar a causa del major nombre de graus de llibertat. Els robots amb rodes són més eficients energèticament, i solen tenir una dinàmica i una estructura mecànica més senzilla en comparació amb la que requereixen els robots humanoides per entrar en contacte amb el terra i proporcionar una força motriu. Els robots amb almenys tres rodes poden aconseguir estabilitat estàtica, simplificant encara més la dinàmica.

Pel que fa als robots auto equilibrats, es basen en el principi del pèndol invertit, el qual consisteix en un pèndol o vareta que gira lliurement sobre un dels seus extrems. Aquest extrem està unit a una articulació situada en un carro que es pot moure de manera horitzontal. El sistema del pèndol invertit, a diferència d'altres sistemes de control, és naturalment inestable. Per tant, s'ha de controlar el sistema per assolir l'estabilitat en aquest estat inestable. Per aconseguir-ho, cal dissenyar i implementar un controlador al robot per equilibrar el pèndol. La teoria del control és una disciplina que tracta del comportament de sistemes inestables, on la sortida desitjada d'un sistema s'anomena referència. Per la seva implementació, s'ha d'identificar quin tipus de dades interessa recopilar i amb quina finalitat. És quan un sistema amb entrades múltiples necessita un controlador que actuï sobre ell, de manera que sigui possible obtenir la sortida desitjada del sistema. En aquest cas, ens interessa mesurar la inclinació del pèndol ja que ens indicarà si el robot està caient i cap a quin costat.

Aquest projecte en particular consisteix en el disseny, fabricació i implementació del sistema de control d'un robot diferencial de dues rodes auto equilibrat. Per desenvolupar-lo, se segueixen les directrius dels robots auto equilibrats que es

troben al mercat, i que estan enfocats particularment a l'aprenentatge i a l'entreteniment.

## 1.1 Història

L'origen d'aquests robots es remunta l'any 1986 a la universitat d'Electro-Comunicacions de Chōfu, Japó, on el professor Kazuo Yamafuji va crear un robot capaç de simular el comportament d'un pèndol invertit. El disseny del robot inclou un eix proveït de dues rodes i un carro que contenia el dispositiu d'estabilització i control.

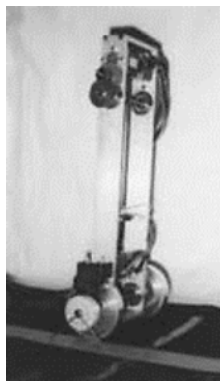


Figura 1.1: Robot creat per Kazuo Yamafuji l'any 1986

Un altre important exemple pot trobar-se l'any 2000. Aquest va ser creat als laboratoris de l'Escola Politècnica Federal de Suïssa, en aquest cas van afegir pesos a la vareta per simular el pes d'un ésser humà. Aquest prototip anomenat *Joe* es podria dir que va ser la base dels Segway.

A l'actualitat els robots auto-equilibrats han tingut un increment de popularitat, això ha estat degut principalment al gran desenvolupament que hi ha hagut de plataformes de baix cost per al desenvolupament de sistemes elèctrics, així com també la fàcil disponibilitat i baix cost dels components electrònics. Actualment, es poden trobar en el mercat robots basats en microcontroladors *PIC* o *AVR* amb diferents funcions i característiques a uns preus molt assequibles. Tot i això, la principal aplicació comercial d'aquest tipus de robots és el Segway, un vehicle de transport giroscòpic elèctric de dues rodes. Creat el 2001 per Dean Kamen, és el primer dispositiu auto equilibrat pensat pel transport, el treball, la seguretat i l'oci. El futur d'aquest tipus de robots auto equilibrats sembla que està destinat a ser robots que es mantenen en equilibri sobre una bola. Per a això s'ha de dissenyar un sistema de control de tres motors diferents, ja que el moviment és omnidireccional. A més, aquest tipus de desplaçament necessita un tipus de roda especial que permeti el moviment en més d'una direcció.



## 1.2 Estat de l'art

Després de fer una recerca, s'ha vist que els robots equilibristes que resolen el problema del pèndol invertit tenen una forma i dimensions similars. Pel que fa als elements que els componen, salvant les diferències depenent del model, es pot veure com també comparteixen certa semblança. Per aquest motiu es pot observar elements comuns a tots ells:

- Rodes: tenen un diàmetre generalment  $1/3$  de l'altura de el robot.
- Motors: tot i que també hi ha models que utilitzen motors pas a pas, els més comuns són els de corrent contínua, majoritàriament de 12V.
- Disseny: es creen estructures altes amb un centre de gravetat baix.
- Cos: s'utilitza un tipus de disseny apilat. Això ajuda a variar la posició del centre de massa i també ajuda a programar el punt de vista, ja que el bot s'ha d'afinar moltes vegades.
- Pes: aproximadament 1kg.
- Centre de massa: la massa es distribueix de manera que la posició del centre de massa sigui el més baixa possible. Si la posició és massa alta, és estable a les pertorbacions, però el requisit de parell serà molt alt i el rang d'angle de recuperació és inferior.
- Unitat de mesura inercial: sensor capaç d'obtenir la mesura de la inclinació del robot. Comunament s'empra la combinació d'un acceleròmetre i un giroscopi. Es col·loca a la pila inferior al llarg de l'eix del centre de massa i paral·lel a l'eix de la roda.
- Control: s'encarrega de proporcionar una acció correctora de l'error produït entre una variable a mesurar i la referència que hauria de seguir. Es pot emprar diferents tipus de control, ja sigui analògic, digital o lògica difusa. Els més habituals són el control PID (acrònim de Proporcional, Integrador i Derivatiu) o LQR (regulador quadràtic lineal).
- Bateria: s'utilitzen bateries de tipus Lipo i també bateries de piles.

## 1.3 Objectius

Amb aquest projecte es vol:

- Fer un disseny mecatrònic on es tindran en compte les diferents parts:
  - Un model CAD previ a la fabricació.
  - Una part elèctrica i electrònica que implicarà una selecció de components elèctrics d'acord amb els requeriments del sistema, i el disseny .
- Fabricar i construir el prototip a partir del disseny.

- Resoldre les equacions que representa el model, en el seu balanceig i el seu gir.
- Fer un anàlisi del model per entendre la teoria de control i dissenyar un compensador LQR per tal d'estabilitzar el balanceig el sistema i trobar la manera de controlar el gir.
- Desenvolupar un programa pel microcontrolador que sigui capaç d'estabilitzar el sistema a través de les dades que rep dels diferents sensors.
- Analitzar com afecten la variació dels paràmetres del sistema.

El prototip d'aquest projecte ha de tenir com a mínim els següents requeriments:

- Tenir un disseny simple, amb poques peces, robust i fàcil de muntar.
- Que tingui un cost econòmic total baix, i que en cas d'iniciar un procés de fabricació a major escala, pogués ser un model competitiu dins el mercat d'aquest tipus de robots.
- Possibilitat de ser controlat per bluetooth a través d'un smartphone.

## 1.4 Estructura de la memòria

La memòria d'aquest projecte segueix un ordre concret, començant per l'anàlisi teòric i acabant per la part pràctica. Cada capítol engloba una part del projecte:

- El capítol 2 explica la modelització matemàtica dels sistema.
- El capítol 3 mostra com s'ha desenvolupat el disseny mecatrònic del prototip.
- El capítol 4 descriu els costos econòmics del projecte.
- El capítol 5 expressa els resultats obtinguts.
- El capítol 6 exposa les conclusions que s'han tret.

A més aquest document es complementa amb un seguit d'apèndixs:

- L'apèndix A mostra el programa realitzat.
- L'apèndix B consta dels plànols del conjunt del prototip i els de les peces que s'han fabricat.
- L'apèndix C mostra l'esquema elèctric del circuit del robot.

## Capítol 2

### Modelització matemàtica

Tots els sistemes mecànics i elèctrics es poden modelitzar matemàticament, la finalitat és trobar una funció de transferència que sigui capaç de substituir el model real. Aquesta tècnica s'utilitza en teoria de control per poder simular com el sistema es comporta amb i sense compensador.

Per desenvolupar aquest projecte, no és indispensable dissenyar un sistema que simuli el prototip i trobar la seva funció de transferència, ja que disposarem del robot físic i ens podrem basar en la reacció d'aquest davant dels canvis de paràmetres que s'apliquin. Per tant, la finalitat d'aquest apartat és determinar els principis matemàtics d'aquests robots auto equilibrats, explorant les propietats del seu principi físic, i la seva cinemàtica i dinàmica.

Per controlar qualsevol sistema s'ha de tenir clar quines seran les entrades i les sortides, és a dir, cal saber quins fenòmens actuaran sobre ell i quines forces generarà per contrarestar-les, esquemàticament ho representariem de la següent manera:

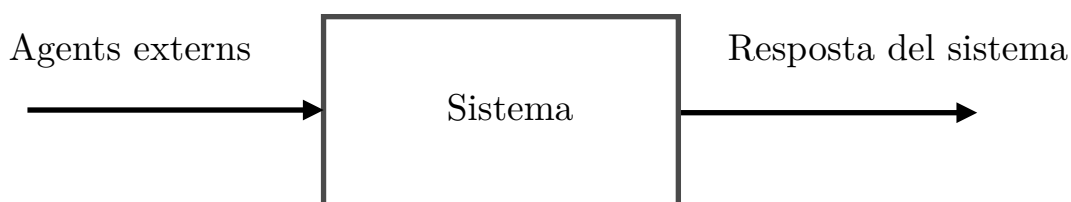


Figura 2.1: Esquema bàsic de qualsevol sistema

### 2.1 Propietats del pèndol invertit

Com ja s'ha comentat, aquest sistema es comporta com un pèndol invertit. Si es vol establir un control per corregir aquest comportament, caldrà començar per comprendre quina és la dinàmica que regeix el moviment de l'pèndol invertit. Un cop coneguda la dinàmica, es pot determinar quin tipus de sensors i actuadors seran necessaris per implementar el control.

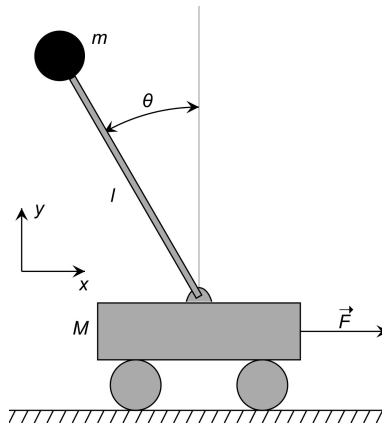


Figura 2.2: Esquema del pèndol invertit

Les equacions de moviment del pèndol invertit, depenen de les condicions inicials del pèndol. Suposant una configuració inicial en la qual és fixa el punt de pivot en l'espai, la dinàmica del sistema és similar a la d'un pèndol no invertit. Si no assumeix cap fricció, i el sistema està restringit a un moviment en dues dimensions, l'equació que regeix el moviment és:

$$\alpha - \frac{g}{L} \cdot \sin(\theta) = 0 \quad (2.1)$$

on:

- $\alpha$ : Acceleració angular del sistema.
- $g$ : Acceleració de la gravetat ( $9.81 \text{ m/s}^2$ ).
- $L$ : Longitud del pèndol.
- $\theta$ : Angle d'inclinació respecte la vertical.

Així, el pèndol guanya acceleració conforme més gran sigui la distància entre el pèndol i la posició vertical d'equilibri. Aquesta equació de moviment del pèndol invertit es pot obtenir mitjançant el parell i el moment d'inèrcia. El pèndol es suposa com una massa  $m$  puntual fixada a una vareta, la qual està fixada a un punt de pivot en l'altre extrem. El parell net del sistema és igual a el moment d'inèrcia per l'acceleració angular:

$$\tau_{net} = I \cdot \alpha \quad (2.2)$$

El parell que proporciona la gravetat correspon al parell net del sistema:

$$\tau_{net} = m \cdot g \cdot L \cdot \sin(\theta) \quad (2.3)$$

Ajuntant les equacions anteriors:

$$I \cdot \alpha = m \cdot g \cdot L \cdot \sin(\theta) \quad (2.4)$$

El moment d'inèrcia del sistema  $I$ , ve donat per:

$$I = m \cdot L^2 \quad (2.5)$$

Substituint a l'equació anterior:

$$m \cdot L^2 \cdot \alpha = m \cdot g \cdot L \cdot \sin(\theta) \quad (2.6)$$

I simplificant l'expressió:

$$\alpha = \frac{g}{L} \cdot \sin(\theta) \quad (2.7)$$

Aquestes són les expressions matemàtiques que regeixen el sistema físic del pèndol invertit, de manera que ja es coneix quina variable s'ha de controlar per mantenir l'equilibri el robot.

## 2.2 Cinemàtica del sistema

La navegació dels robots mòbils consisteix en el model de la configuració d'actuadors que permeten al robot moure's per l'entorn, és a dir el seu model cinemàtic. Obtenir una posició precisa de robot sol ser una tasca extremadament difícil. Pel cas del prototip del robot auto equilibrat, conté dues rodes que contribueixen al moviment i mantenen el seu equilibri. Pel moviment, la seva configuració és a partir de la tracció diferencial, ja que el direccionament es genera per la diferència de velocitats de les rodes. Per tant, per a formular el model cinemàtic, se simplifica amb alguns supòsits, els quals permeten generalitzar el comportament del robot:

1. El robot ha de ser considerat com un cos rígid, sòlid i sense parts flexibles.
2. S'ha de tenir en compte les restriccions no holonòmiques del sistema, això fa referència al fet que el robot pot moure's cap endavant o enrere, però no desplaçar-se cap als costats, per fer aquesta classe de desplaçaments s'ha de moure per parts.
3. El pla d'operació ha de ser tractat com horitzontal, ja que s'ha dissenyat el robot per treballar sobre una superfície plana sense cap tipus d'inclinació.
4. A diferència d'un robot diferencial que depèn d'una roda addicional per mantenir l'equilibri, el prototip de robot auto equilibrat ja manté l'equilibri sense necessitat d'algun dispositiu addicional.
5. El moviment del robot té com a base el balanç sense lliscament entre les rodes i el terra.
6. Els eixos de direcció han de ser ortogonals a la superfície.
7. Denotant aquests supòsits, per a realitzar el model és necessari conèixer les condicions físiques de sistema o paràmetres cinemàtics. S'ha de conèixer els radis de la roda esquerra ( $r_L$ ), de la roda dreta ( $r_R$ ) i la distància entre les rodes  $D$ . A la figura 2.2, es representa els dos rectangles negres com les dues rodes de el robot, entre ells tenen un punt central, aquest es pren com a referència de la ubicació de l'robot.

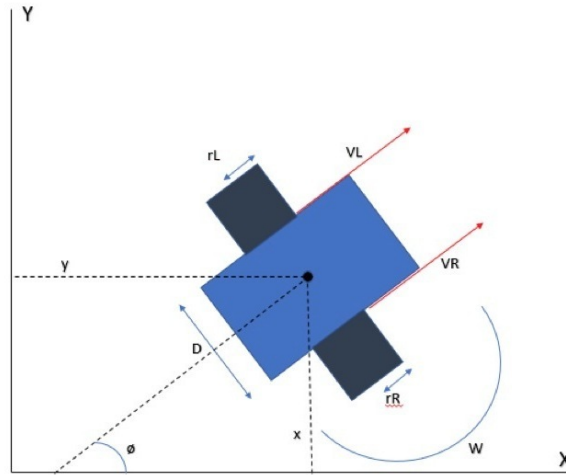


Figura 2.3: Gràfic del model cinemàtic

Les incògnites que té el prototip són la posició i l'orientació dins d'un sistema global, de tal manera que per tenir un moviment controlat, s'ha de poder dominar la velocitat de les rodes ( $VR$  i  $VL$ ). Així, amb aquest plantejament es pot partir amb equacions senzilles que permetran completar la cinemàtica completa.

Velocitats lineals a les rodes:

$$VR = r_R \cdot \omega_R \quad (2.8)$$

$$VL = r_L \cdot \omega_L \quad (2.9)$$

Les dues velocitats anteriors produeixen instantàniament una velocitat lineal:

$$V = \frac{VR + VL}{2} \quad (2.10)$$

Com a aproximació, fent ús de les velocitats lineals de les rodes per una primera posició del robot, aquest es mou segons:

$$x' = x + V \cdot \cos(\phi) \cdot t \quad (2.11)$$

$$y' = y + V \cdot \sin(\phi) \cdot t \quad (2.12)$$

Quan el robot té moviment de rotació sobre el seu centre de massa, tindrà una velocitat angular, això significa que hi haurà una diferència entre la velocitat de les seves rodes respecte la longitud que hi ha entre elles:

$$\omega = \frac{VR - VL}{d} \quad (2.13)$$

D'aquesta forma al fer una rotació donaria una nova orientació, que es representaria de la següent manera:

$$\phi' = \phi + \omega \cdot t \quad (2.14)$$

En resum, s'obtenen tres equacions a partir de les velocitats angulars de les rodes, i generant les velocitats lineals en les mateixes rodes s'obtindria les noves posicions i la nova orientació, així que l'equació final consta de dues matrius, la primera és la matriu de rotació i la segona és la matriu de velocitat:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix} \quad (2.15)$$

En aquest cas el radi de les rodes és igual, per tant, al resoldre la última equació el resultat del model és:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r \frac{VR + VL}{2} \\ r \frac{VR - VL}{D} \end{bmatrix} \quad (2.16)$$

Multiplicant aquestes dues matrius s'obté:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} r * \cos(\phi) * \frac{VR + VL}{2} \\ r * \sin(\phi) * \frac{VR + VL}{2} \\ r \frac{VR - VL}{D} \end{bmatrix} \quad (2.17)$$

Per tenir com entrades al sistema les velocitats de les rodes, la conversió quedaria:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r * \cos(\phi)}{2} & \frac{r * \cos(\phi)}{2} \\ r * \sin(\phi) & r * \sin(\phi) \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} VR \\ VL \end{bmatrix} \quad (2.18)$$

## 2.2 Dinàmica del sistema

Per l'anàlisi teòric que es planteja, existeix informació amb la qual hi ha una base tant de contingut web com de publicacions de diversos articles científics en bases de dades. Hi ha una diversa quantitat de solucions per modelar aquest sistema, totes aquestes solucions són encertades, ja que coincideixen en els punts clau.

La massa del pèndol es reparteix uniformement al llarg d'una vareta de longitud determinada, però sense amplada ni profunditat, i està connectada a un dels extrems a la base del sistema. Al realitzar la modelització es contempen inèrcies relacionades amb la dinàmica de la vareta.

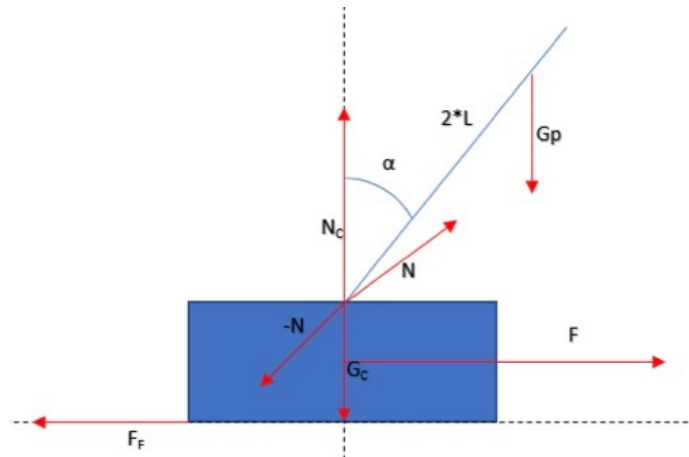


Figura 2.4: Variables del pèndol invertit

Per determinar el model de sistema, es divideix el robot en dues seccions: el pèndol i la base en la qual es desplaça (carro). Es descriuen les forces i paràmetres que intervenen, agrupades en la següent taula:

Variable	Descripció
N	Força normal de reacció del carro sobre el pèndol.
-N	Força normal de reacció del pèndol sobre el carro.
N <sub>C</sub>	Força normal de la superfície del carro.
F	Força aplicada per moure el carro.
F <sub>F</sub>	Força de fricció entre el carro i la superfície sobre la que es desplaça.
G <sub>C</sub>	Força de la gravetat sobre el carro.
G <sub>P</sub>	Força de la gravetat sobre el pèndol.
α	Angle de desviació del pèndol respecte la posició d'equilibri.
ω	Velocitat angular del pèndol.
ε	Acceleració angular del pèndol.
M <sub>C</sub>	Massa del carro.
M <sub>P</sub>	Massa del pèndol.
a <sub>C</sub>	Acceleració lineal del carro.
a <sub>P</sub>	Acceleració del pèndol.
L	Distància des de la unió del carro i el pèndol fins el centre de masses del pèndol.
g	Gravetat.
μ <sub>C</sub>	Coefficient de fricció entre la superfície i el carro.
I	Moment d'inèrcia del pèndol.
M	Sumatori dels parells no inercials que actuen en el pèndol respecte a l'articulació que l'uneix amb el carro.

Taula 2.1: Forces i paràmetres d'un robot auto equilibrat

Definides les forces i els paràmetres del sistema, es determinen les equacions. En primer lloc, fent ús de la segona llei de Newton al moviment lineal, s'obté la suma de totes les forces:

$$\sum F = m * a \rightarrow F + F_f + G_C - N + N_C = m_C * a_C \quad (2.19)$$

Després, es descomposa cada força i acceleració de l'equació (2.19) en termes corresponents als eixos X i Y. Es té en compte que U<sub>x</sub> i U<sub>y</sub> són vectors unitaris.

$$\begin{aligned} F &= F \cdot U_x & F_f &= -F_f \cdot U_x & G_C &= m_C \cdot g \cdot U_Y \\ N &= N_x \cdot U_x - N_Y \cdot U_Y & N_C &= -N_C \cdot U_Y & a_C &= \ddot{x} \cdot U_x \end{aligned} \quad (2.20)$$



S'agrupen les equacions en dues, una per a cada eix:

$$\text{Eix X} \rightarrow F - F_f - N_x = m_c \cdot \ddot{x} \quad (2.21)$$

$$\text{Eix Y} \rightarrow m_c \cdot g + N_y - N_c = 0 \quad (2.22)$$

Es pot reescriure la força de fricció com descriu el model de Coulomb de fricció, tenint en compte que la superfície limita el moviment del carro cap amunt i cap avall, la força es descriuria així:

$$F_f = \mu_c \cdot |N_c| \cdot \text{sgn}(\dot{x}) = \mu_c \cdot N_c \cdot \text{sgn}(\dot{x} \cdot N_c) \quad (2.23)$$

S'ha d'aplicar la segona llei de Newton al moviment lineal del pèndol, obtenint així:

$$N + G_p = m_p \cdot a_p \quad (2.24)$$

A l'igual que amb la part del carro, s'han de descomposar les forces i l'acceleració en termes corresponents als eixos X i Y. Es descomposa la força de la gravetat exercida sobre el pèndol:

$$G_p = m_p \cdot g \cdot U_Y \quad (2.25)$$

Es descomposa l'acceleració del pèndol, s'ubica en el centre de massa del pèndol i es compon de:

- l'acceleració de el carro ja que està unit al pèndol:

$$a_c = \ddot{x} \cdot U_x \quad (2.26)$$

- la rotació del pèndol està subjecte a una velocitat angular:

$$\omega = \dot{\alpha} \cdot U_z \quad (2.27)$$

- l'acceleració angular del pèndol

$$\varepsilon = \ddot{\alpha} \cdot U_z \quad (2.28)$$

Es reescriu l'equació del pèndol de la següent manera:

$$a_p = a_c + \varepsilon \times r_p + \omega \times (\omega \times r_p) \quad (2.29)$$

on:

$$r_p = l \cdot (\sin(\alpha) \cdot U_x - \cos(\alpha) \cdot U_y) \quad (2.30)$$

$r_p$  és el vector que representa la posició de centre de massa del pèndol respecte l'articulació sobre la qual gira el pèndol. Substituint aquestes equacions sobre l'equació 2.25 s'obté:

$$a_p = \ddot{x} \cdot U_x + \ddot{\alpha} \cdot U_z \times l \cdot (\sin(\alpha) \cdot U_x - \cos(\alpha) \cdot U_y) + \dot{\alpha} \cdot U_z \times (\dot{\alpha} \cdot U_z \times l \cdot (\sin(\alpha) \cdot U_x - \cos(\alpha) \cdot U_y)) \quad (2.31)$$

Es realitza el producte vectorial dels vectors unitaris per simplificar l'equació:

$$U_z \times U_x = U_y \quad (2.32)$$

$$U_z \times U_y = -U_x \quad (2.33)$$

S'obté el següent resultat:

$$a_p = \ddot{x} \cdot U_x + \ddot{\alpha} \cdot l \cdot (\sin(\alpha) \cdot U_y + \cos(\alpha) \cdot U_x) + \dot{\alpha}^2 \cdot l \cdot (\sin(\alpha) \cdot U_x - \cos(\alpha) \cdot U_y) \quad (2.33)$$

Determinada l'acceleració del pèndol, se substitueix a l'equació 2.24 i d'aquesta manera es continua amb la descomposició de forces i acceleracions, en els termes dels eixos X i Y.

$$\text{Eix X} \rightarrow N_x = m_p \cdot (\ddot{x} + \ddot{\alpha} \cdot l \cdot \cos(\alpha)) - \dot{\alpha}^2 \cdot l \cdot \sin(\alpha) \quad (2.34)$$

$$\text{Eix Y} \rightarrow m_p \cdot g - N_y = m_p \cdot (\ddot{x} + \ddot{\alpha} \cdot l \cdot \cos(\alpha)) - \dot{\alpha}^2 \cdot l \cdot \sin(\alpha) \quad (2.35)$$

Aplicant la segona llei de Newton al moviment rotatori del pèndol voltant de l'articulació s'obté:

$$M = I \cdot \varepsilon + r_p \times a_c \quad (2.36)$$

De tal manera que l'equació 2.37, és el sumatori dels parells no inercials actuant en el pèndol respecte de l'articulació que l'uneix amb el carro.

$$M = r_p \times G_c - \mu_p \cdot \dot{\alpha} \cdot U_z \quad (2.37)$$

El moment d'inèrcia del pèndol respecte de l'articulació que l'uneix amb el carro és:

$$I = \left(\frac{4}{3}\right) \cdot m_p \cdot l^2 \quad (2.38)$$

Es pot interpretar l'equació 2.39 com el parell generat per la força inercial causat per l'acceleració del carro.

$$-r_p \times a_c \quad (2.39)$$

D'aquesta manera se substitueixen les equacions 2.37, 2.38 i 2.39 a la 2.36, i s'obté:

$$m_p \cdot g \cdot l \cdot \sin(\alpha) - \mu_p \cdot \dot{\alpha} = \left(\frac{4}{3}\right) \cdot m_p \cdot l^2 \cdot \ddot{\alpha} + m_p \cdot \ddot{x} \cdot l \cdot \cos(\alpha) \quad (2.40)$$

Realitzat aquest anàlisi, el següent pas és ordenar les equacions obtingudes, per a això es substitueix  $N_x$  de l'equació 2.34 en l'equació número 2.21, com es mostra a continuació:

$$\ddot{x} = \frac{F - F_f + m_p \cdot l \cdot (\ddot{\alpha} \cdot \cos(\alpha) - \dot{\alpha}^2 \cdot \sin(\alpha))}{m_c + m_p} \quad (2.41)$$

Per obtenir l'acceleració lineal, se substitueix l'equació 2.41 a l'equació 2.40:

$$\ddot{\alpha} = \frac{g \cdot \sin(\alpha) + \cos(\alpha) \cdot \left[ \frac{-F - m_p \cdot l \cdot \dot{\alpha}^2 \cdot \sin(\alpha) + F_f}{m_c + m_p} \right] - \frac{\mu_p \cdot \dot{\alpha}}{m_p \cdot l}}{l \cdot \left[ \frac{4}{3} \cdot \frac{m_p \cdot \cos(\alpha)^2}{m_c + m_p} \right]} \quad (2.42)$$

Ara s'ha d'aïllar  $N_y$  de l'equació 2.35 i se substitueix a l'equació 2.22:

$$N_c = (m_c + m_p) \cdot g - m_p \cdot l \cdot (\ddot{\alpha} \cdot \sin(\alpha) + \dot{\alpha}^2 \cdot \cos(\alpha)) \quad (2.43)$$

Se substitueix  $N_c$  de l'equació 2.43 i se substitueix a l'equació 2.23:

$$F_f = \mu_c \cdot [(m_c + m_p) \cdot g - m_p \cdot l \cdot (\ddot{\alpha} \cdot \sin(\alpha) + \dot{\alpha}^2 \cdot \cos(\alpha))] \cdot \text{sgn}(\dot{x} \cdot (m_c + m_p)) \cdot (g - m_p \cdot l \cdot (\ddot{\alpha} \cdot \sin(\alpha) + \dot{\alpha}^2 \cdot \cos(\alpha))) \quad (2.44)$$

Per acabar, s'introdueix l'expressió de la força de fricció en les equacions 2.41 i 2.42:

$$\ddot{x} = \frac{F - \mu_c \cdot [(m_c + m_p) \cdot g - m_p \cdot l \cdot (\ddot{\alpha} \cdot \sin(\alpha) + \dot{\alpha}^2 \cdot \cos(\alpha))] \cdot \text{sgn}(\dot{x} \cdot N_c) + m_p \cdot l \cdot (\ddot{\alpha} \cdot \cos(\alpha) - \dot{\alpha}^2 \cdot \sin(\alpha))}{m_c + m_p} \quad (2.45)$$

L'equació 2.45 representa l'acceleració lineal de sistema, i permet obtenir el moviment lineal, que posteriorment es pot integrar per determinar la velocitat lineal, i així mateix integrar la velocitat per trobar la posició de l'robot.

L'equació 2.46 equació permet obtenir el moviment angular del pèndol, ja que es pot determinar l'acceleració angular. Igual que en el cas anterior, es pot integrar per determinar la velocitat angular i així mateix integrar la velocitat per trobar l'angle de desviació respecte el punt d'equilibri.

Per a més simplificació, es poden anul·lar els termes de fricció, obtenint:

$$\ddot{\alpha} = \frac{g \cdot \sin(\alpha) + \cos(\alpha) \cdot \left[ \frac{-F - m_p \cdot l \cdot \dot{\alpha}^2 \sin(\alpha)}{m_c + m_p} \right]}{l \cdot \left[ \frac{4}{3} - \frac{m_p \cdot \cos(\alpha)^2}{m_c + m_p} \right]} \quad (2.46)$$

$$\ddot{x} = \frac{F + m_p \cdot l \cdot (\ddot{\alpha} \cdot \cos(\alpha) - \dot{\alpha}^2 \cdot \sin(\alpha))}{m_c + m_p} \quad (2.47)$$

Per poder tenir control sobre l'angle de posició, només es tindrà en compte l'equació que determina l'acceleració angular, per la qual cosa és necessari tenir identificades la massa del carro, i la massa i la longitud del pèndol.

# Capítol 3

## Disseny mecatrònic

Desenvolupar un projecte d'aquestes característiques és un repte mecatrònic on hi intervé l'enginyeria mecànica, electrònica, de control i de programari. En aquest apartat es mostra com s'ha abordat cadascun d'aquests camps.

### 3.1 Disseny mecànic

Ha sigut el procés de donar forma, dimensions, i materials perquè el sistema compleixi les determinades funcions. S'han tingut en compte aspectes com les tecnologies de fabricació a utilitzar, les característiques dels materials, i la disponibilitat, el preu i els terminis d'entrega dels productes a comprar. Com que el robot en si no és un conjunt molt pesat ni tampoc estarà sotmès a grans esforços, es desestima fer un anàlisi d'esforços i tensions de l'estructura. La Figura 3.1 mostra el disseny realitzat amb Inventor:

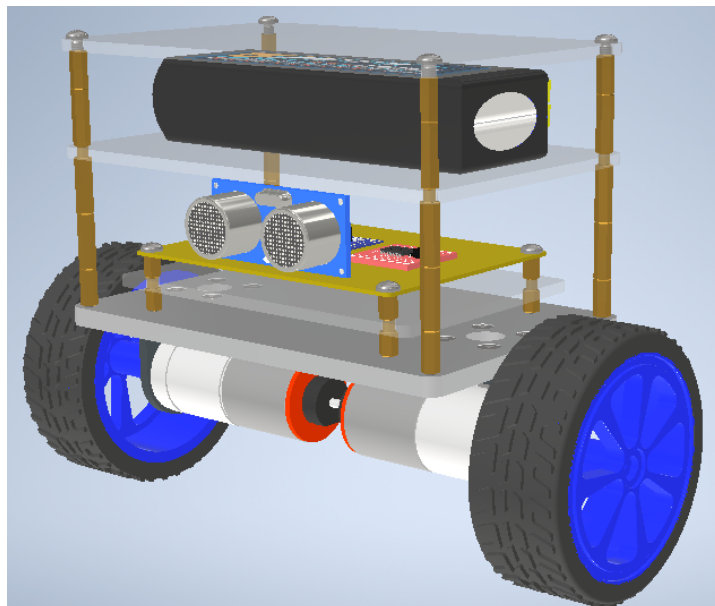


Figura 3.1: Disseny CAD

Com es pot observar, el disseny del robot és un cos rectangular sobre dues rodes paral·leles entre si. Aquest cos es troba sobre dels motors i en mig de les dues rodes, i es compona per tres bases unides per separadors a cascun dels seus vèrtex.

Aquest disseny mecànic no s'ha dut a terme de forma arbitrària:

- S'ha col·locat el component de major pes a la base del mig, lluny de l'eix de gir, per aconseguir un centre de gravetat el més alt possible, intentant mantenir el patró estètic desitjat.
- També s'ha buscat obtenir una estructura rígida, que fos lleugera i econòmica.
- Pel que fa al muntatge els objectius han estat que el robot fos fàcil de muntar i desmuntar, i que tingui un fàcil accés a qualsevol de les seves parts.

El cos rectangular del robot té unes dimensions de 125x80x76 mm i la distància entre rodes és de 134 mm. L'alçada des de l'eix de gir fins la part superior és de 97 mm. La distribució de massa, també ha estat un aspecte a tenir en compte. El moment d'inèrcia es designa com la massa d'un cos pel quadrat de la distància, en aquest cas a l'eix de gir, és a dir que la inèrcia d'un cos quan gira, no només depèn de la seva massa, sinó que també depèn de com estigui distribuïda aquesta massa respecte a l'eix de gir. Com més allunyada estigui la massa de l'eix de gir, major inèrcia presentarà i més costarà el seu moviment circular. Per aquesta raó, la bateria, al ser l'element més pesat de tots, s'ha situat a la base més llunyana a l'eix de gir situat a les rodes de l'robot.

Un cop explicat el disseny mecànic amb una descripció de l'estructura del robot, les seves mesures i la seva distribució principal, s'explica els components mecànics que el formen.

### 3.1.1 Bases de l'estructura

La base inferior és d'alumini, i és la base del robot. Al seu damunt conté la resta de l'estructura, i també una altre base on van muntats els components electrònics i el cablejat que els uneix. Per sota s'hi subjecten els suports que aguanten els motors. La base intermitja serveix per subjectar-hi la bateria que alimenta el sistema i la superior simplement fa de sostre i tanca l'estructura. Aquestes dues bases i la que suporta la placa electrònica són de metacrilat.

Han sigut fabricades utilitzant un centre de mecanitzats. Utilitzant programació CNC, s'han fresat els contorns de les plaques amb un fresolí de  $\varnothing 6$ , i s'han fet els forats de  $\varnothing 3,2$  per passar-hi els cargols de M3. A la placa d'alumini s'han fet també dos forats de  $\varnothing 10$  per passar-hi el cablejat dels motors i s'han avellanat els forats  $\varnothing 3,2$  a  $45^\circ$  per amagar les cabotes dels cargols.

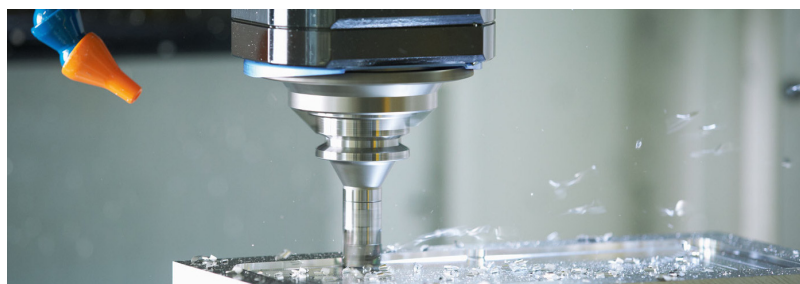


Figura 3.2: Mecanització de la placa base del robot

### 3.1.2 Rodes

Com més gran sigui el seu diàmetre, major serà la seva velocitat punta i tindrà un millor control. Com més petites siguin més lenta serà la velocitat lineal màxima ( $V = \omega \cdot R$ ), i per tant menys capç serà de recuperar-se de possibles caigudes, ja que el motor arribarà a la seva velocitat màxima amb més facilitat. En la majoria de casos que s'han estudiat per aquest projecte, utilitzen uns diàmetres d'entre 60 i 80 mm. Les adquirides per aquest projecte són de diàmetre 67 mm i tenen una amplada de 22 mm.



Figura 3.3: Rodes del robot

### 3.1.3 Separadors

Els separadors hexagonals que suporten l'estructura han sigut comprats. Són de llautó i la seva rosca és de M3. Els cargols i femelles són d'acer inoxidable. El paquet inclou diferents tipus de separadors tant mascles com femelles, tots ells amb una varietat de 6, 10, 15 i 20 mm de longitud. Aquesta varietat suposa un avantatge, fa versàtil l'estructura ja que permet variar alçades de forma senzilla i ràpida en cas de que es vulguin modificar.



Figura 3.4: Separadors de llautó



## 3.2 Disseny electrònic

El procés es pot dividir en tres parts que són:

- Determinar una sèrie de dispositius compatibles entre ells i que conjuntament permetin complir les expectatives del robot.
- Ubicar-los en el disseny mecànic.
- Dissenyar i realitzar les connexions

Els components principals del circuit del robot d'equilibri són: la unitat de mesura inercial (IMU), el microcontrolador i els motors de corrent contínua. L'IMU s'utilitza per al càlcul de l'angle d'inclinació, i està formada per un acceleròmetre i un giroscopi, de manera que es busca una forma d'entendre el seu funcionament i descobrir com extreure l'angle de les dades proporcionades. A través del microcontrolador s'implementa un sistema de control digital amb el qual s'intenta solucionar la inclinació de l'angle. Els motors són els encarregats del moviment del robot, de manera que hauran d'actuar en funció de la informació que s'obtingui del control. El disseny del sistema del *hardware* és important per unir el mecanisme i el programari. La figura 3.4 mostra el diagrama general del sistema electrònic del prototip.

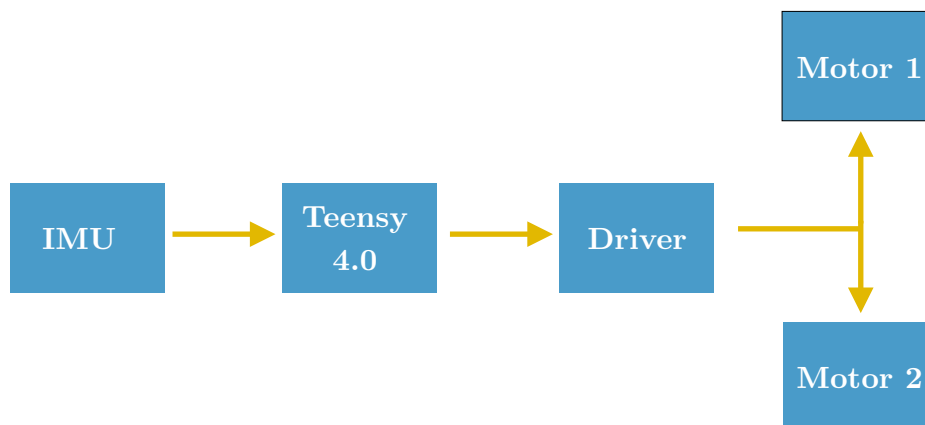


Figura 3.5: Esquema bàsic del sistema electrònic

En la implementació d'aquest sistema hi ha dos blocs importants:

- Bloc d'alimentació: es busca realitzar la correcta alimentació de tots els dispositius. Per a això, s'haurà de passar la tensió proporcionada per la bateria a 12V a totes les zones del circuit, i regular-la a tensions més baixes quan sigui necessari.
- Bloc de senyal: es distribuïran tots els elements del circuit entre els seus diferents pins de connexió, realitzant totes les connexions necessàries que permetin la correcta comunicació entre els sensors, actuadors i el microcontrolador.

A continuació es descriuen els dispositius utilitzats i es mostra el disseny del connexionat:

### 3.2.1 Teensy 4.0

Teensy 4.0 compta amb un processador *ARM Cortex-M7* a 600 MHz, amb un xip NXP i MXRT1062, el microcontrolador més ràpid disponible en l'actualitat. Teensy 4.0 té la mateixa mida i forma que Teensy 3.2 i manté la compatibilitat amb la majoria de les funcions dels pins del Teensy 3.2. Quan funciona a 600 MHz, Teensy 4.0 consumeix aproximadament 100 mA de corrent. Teensy 4.0 proporciona suport per a l'escala dinàmica del rellotge. A diferència dels microcontroladors tradicionals, on canviar la velocitat del rellotge causa velocitats de transmissió incorrectes i altres problemes, el maquinari Teensy 4.0 i el suport de programari de Teensyduino per a les funcions de sincronització Arduino estan dissenyats per permetre canvis dinàmics de velocitat. Les velocitats de transmissió en sèrie, les mostres de transmissió d'àudio i les funcions d'Arduino com *delay()* i *millis()*, i les extensions de Teensyduino com *IntervalTimer* i *elapsedMillis*, continuen funcionant correctament mentre la CPU canvia de velocitat. Teensy 4.0 també proporciona una funció d'apagat. En connectar un polsador al pin On / Off, la font d'alimentació de 3,3 V es pot desactivar completament mantenint premut el botó durant 5 segons i tornar-la a activar prement un breu botó.

L'ARM Cortex-M7 aporta moltes funcions de CPU potents a una autèntica plataforma de microcontroladors en temps real. Cortex-M7 és un processador de superescala i pot executar dues instruccions per cicle de rellotge a 600 MHz. Per descomptat, executar-ne dos simultàniament depèn de les instruccions i registres del compilador. Els paràmetres inicials han mostrat que el codi C++ compilat per Arduino tendeix a aconseguir dues instruccions aproximadament entre el 40% i el 50% del temps mentre realitza un treball numèricament intensiu mitjançant enters i punters. Cortex-M7 és el primer microcontrolador ARM que utilitza la branca de la predicció. La memòria acoblada és una característica especial, que permet l'accés ràpid a un cicle simple de Cortex-M7 a la memòria mitjançant un parell de busos d'amplada de 64 bits.

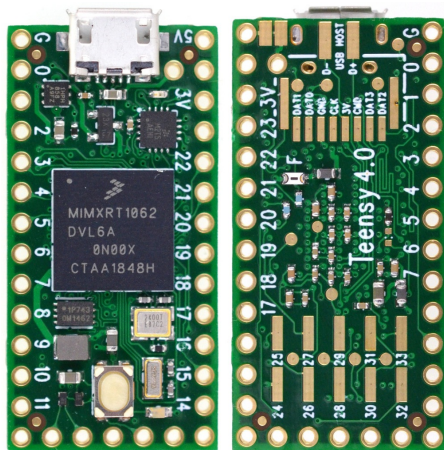


Figura 3.6: Teensy 4.0



### 3.2.2 MPU-6050

L'IMU s'utilitza per mesurar l'acceleració i la velocitat angular del robot. La sortida es processa de forma digital. Les entrades en brut de l'IMU es processen per obtenir l'angle d'inclinació del robot. Aquest angle d'inclinació s'introdueix a l'algorisme del controlador LQR per generar la velocitat adequada al motor de corrent continu per tal d'equilibrar el robot.

El xip MPU-6050 és una unitat de mesura inercial (IMU) de sis graus de llibertat, que combina un acceleròmetre i un giroscopi de 3 eixos cadascun. Un aspecte positiu és que ocupen poc espai, donen informació en tots els eixos i consumeixen molt poc. La comunicació amb el controlador pot realitzar-se a través de senyals analògics o amb bus de dades I2C, el qual redueix la complexitat del cablejat. La tensió d'alimentació és de baix voltatge entre 2.4 a 3.6V. Incorpora un processador intern (DMP Digital Motion Processor) que executa complexos algorismes de MotionFusion per combinar els mesuraments dels sensors interns. L'acceleròmetre retorna acceleracions en l'eix X, Y i Z, per tant es pot calcular l'angle en el qual es troba el sensor amb simple trigonometria. El giroscopi en canvi retorna la velocitat angular de cada un dels eixos.

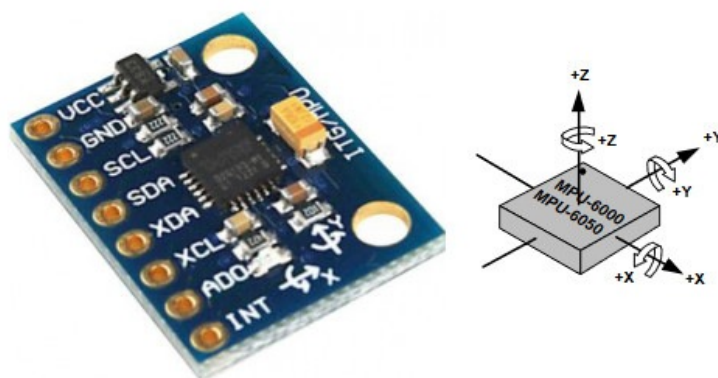


Figura 3.7: MPU-6050

És un dels “imus” més utilitzats per la seva gran qualitat i preu, i molt sovint s'incorporen en projectes d'electrònica i robòtica.

Cada un dels dos senyals té les seves característiques que cal tenir en compte:

- L'angle mesurat amb l'acceleròmetre és molt sorollós, s'ha d'aplicar algun filtre passa baixes per eliminar-lo, el soroll és un problema pel compensador.
- La velocitat angular mesurada amb el giroscopi pateix de *drift*, és a dir que els valors no són constants en el temps sinó que aquests tendeixen a patir derives.

Aquestes característiques es tenen en compte alhora de programar el control digital.

### 3.2.3 Motors DC i encoder

S'utilitzen motors reductors de corrent continua de 12V per generar un parell motor amb el qual el robot es pugui equilibrar i romandre en posició vertical. Aquesta, a més, ja disposen de codificadors incrementals acoblats. Utilitzar encoders per a la realització d'aquest projecte és una opció optativa, ja que no són necessaris per mantenir l'equilibri. Aquests es poden utilitzar per al càlcul de la velocitat del robot, de manera que permetin realitzar un doble llaç de control amb el que regular la velocitat del sistema. També es poden utilitzar per saber la direcció i la posició.

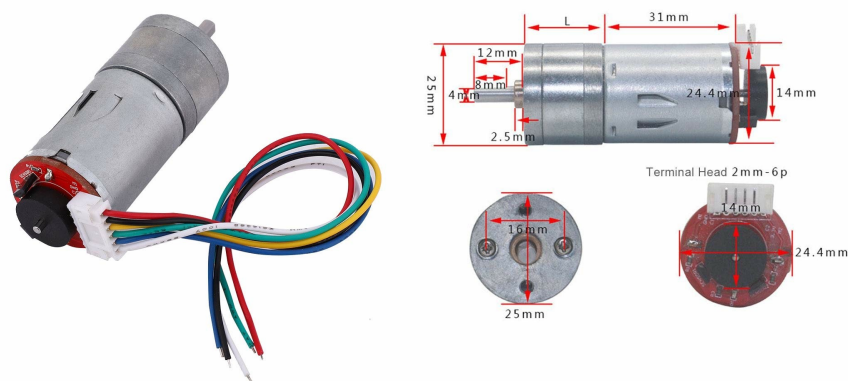


Figura 3.8: Motor reductor DC de 12V amb encoder.

El robot requereix una certa velocitat de rotació perquè pugui contrarestar l'error d'equilibri a una velocitat adequada. La falta de velocitat o un parell dels motors baix pot impossibilitar l'equilibri del robot. Els escollits per aquest projecte són de 130 rpm i el seu parell nominal és de 0.43 Nm. El parell mínim que requereix el robot dissenyat és el següent:

$$\tau = \frac{L \cdot m \cdot g \cdot \sin(40)}{2} = \frac{0.07 \cdot 0.85 \cdot 9.81 \cdot \sin(40)}{2} = 0.187 \text{ Nm} \quad (3.1)$$

On  $L$  és l'alçada del robot des del centre de massa,  $m$  és la seva massa,  $g$  és la gravetat i  $40^\circ$  és l'angle màxim d'inclinació. Es divideix per dos perquè el sistema té dos motors.

Pel que fa als encoders, la senyal del codificador està en quadratura. Segons el gir de l'eix del motor, es generen dues ones quadrades desfasades  $90^\circ$  elèctrics, una de cada sensor. Si arriba una ona desfasada abans o després, indicarà el sentit de gir del motor. Segons la freqüència de dites senyals, es podrà saber la velocitat de gir de l'eix. Comptant el número de flancs podem saber el número de voltes i quan les realitza el motor (comptant les dues senyals enviades pel codificador, és un total de quatre per període). Aquestes són les senyals de sortida A i B:

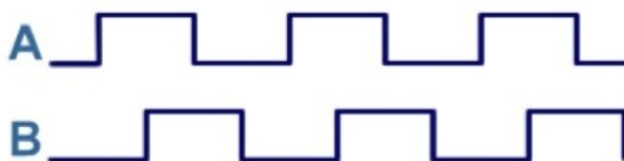


Figura 3.9: Senyals A i B d'un encoder

A més, incorporen un reductor que redueix la velocitat de l'eix de gir, a la vegada que s'augmenta el parell. El coeficient de reducció no venia donat pel fabricant, pel que ha sigut necessari realitzar proves experimentals per a la seva obtenció. La reducció obtinguda és de 1:45 entre l'eix del motor i l'eix de sortida. El *PPR* (polsos per revolució) és de 11 polsos per gir del motor.

És important tenir en compte que hi ha dos tipus d'eix:

- Eix del motor: eix situat en el rotor del motor DC i del que es mesuren els polsos referents als girs d'aquest mitjançant el codificador.
- Eix de la roda: eix on s'acobla la roda. És l'eix de sortida de la reductora.

### 3.2.4 Controlador TB6612FNG

El TB6612FNG és un *driver* que disposa de dos canals, la qual cosa és possible controlar dos motors de corrent continu de forma independent, variant tant la velocitat com el sentit de gir. En el seu cas, els ponts H estan formats per transistors MOSFET, enlloc de transistors BJT. Això permet que el TB6612FNG tingui una millor eficiència i menors dimensions que altres models, i que pugui controlar intensitats de corrent superior, sent capaç de subministrar 1.2A per canal de forma contínua, i 3.2A de pic. Cada canal pot controlar el sentit de gir i la velocitat, el qual admet un senyal PWM de freqüència màxima de 100 kHz.

El TB6612FNG també incorpora un mode de Standby, que es desactiva per complet el controlador, entrant en un mode d'estalvi d'energia.

S'utilitza àmpliament en projectes electrònics i robòtics, per la seva senzillesa d'ús, baix cost, i bona qualitat. L'esquema de muntatge no és massa complicat. D'una banda, subministrem la tensió que alimentarà el motor des d'una font d'alimentació externa, mitjançant el pin VM. La tensió màxima és de 15V. A més, hem d'alimentar l'electrònica del mòdul mitjançant el pin VCC. El rang de tensió per VCC és 2.7 a 5.5V.

Els pins AIN1, AIN2 i PWMA controlen el canal A, mentre que els pins BIN1, BIN2, i PWMB controlen el canal B. Finalment, el pin STBY controla el mode *standby*. Hem de posar-lo en HIGH per activar el motor. El qual es pot connectar a un pin digital del microcontrolador.



Figura 3.10: Driver TB6612FNG

### 3.2.5 Bluetooth HM10

Com que es pretén controlar el robot a distància utilitzant un dispositiu *smartphone*, la solució més fàcil és la utilització d'un mòdul de comunicació bluetooth. Aquesta comunicació té avantatge d'estar integrat de fàbrica a la majoria de dispositius. Portàtils, tablets, i smartphones porten aquests sistema integrat. A més, el seu ús és independent de sistema operatiu (Windows, Linux, Mac o Android). Per això aquesta tecnologia és un dels millors mitjans per comunicar-nos sense fils amb microcontroladors.

A l'igual que mòduls serials anteriors, l'HM10 emula un port serial sobre el protocol bluetooth i és ideal per a comunicar-se amb dispositius mòbils tant d'Apple com Android.

En la comunicació sèrie, la informació es transmet emprant dos conductors principals, encara que poden existir altres com sincronització de rellotge:

- Rx: Recepció de dades
- Tx: Transmissió de dades

La comunicació es pot realitzar de dues maneres diferents:

- Fent servir les comandes AT per realitzar la configuració del dispositiu
- Des d'un terminal sèrie, que en el cas d'un dispositiu *smartphone* serà una aplicació de tercers descarregada des de Google Play o App Store amb la qual es poden enviar i rebre dades

En el cas d'aquest projecte es vol enviar dades al dispositiu, tot i que també es podria utilitzar, per exemple, per rebre mesures dels sensors.

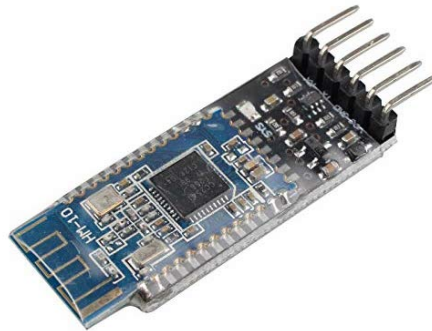


Figura 3.11: Mòdul Bluetooth HM-10

### 3.2.6 Ultrasons HC-SR04

Un sensor d'ultra sons és un dispositiu per mesurar distàncies. El seu funcionament es basa en l'enviament d'un pols d'alta freqüència, no audible per l'ésser humà. Aquest pols rebota en els objectes propers i és reflectit cap al sensor, que disposa d'un micròfon adequat per a aquesta freqüència. Mesurant el temps entre polsos i coneixent la velocitat del so, podem estimar la distància de l'objecte on impacte l'impuls d'ultrasons.

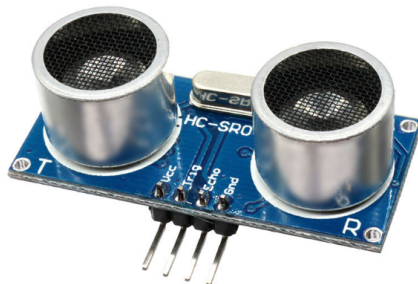


Figura 3.12: Sensor d'ultrasons HC-SR04

Els sensors d'ultrasons són sensors barats, i senzills d'utilitzar. El rang de mesurament teòric de l'sensor HC-SR04 és de 2 cm a 400 cm, amb una resolució de 0.3cm. L'orientació de la superfície a mesurar pot provocar que l'ona es reflecteixi, falsejant el mesurament. A més, no resulten adequats en entorns amb gran nombre d'objectes, ja que el so rebota en les superfícies generant ecos i falses mesuraments. Tampoc són apropiats per al funcionament a l'exterior i a l'aire lliure.

Malgrat aquesta baixa precisió, que impedeix conèixer amb precisió la distància a un objecte, els sensors d'ultrasons són àmpliament emprats. En robòtica és habitual muntar un o diversos d'aquests sensors, per exemple, per detectar obstacles, determinar la posició de l'robot, crear mapes d'entorn, o resoldre laberints.

En aplicacions en què es requereixi una precisió superior en el mesurament de la distància, solen acompanyar-se de mesuradors de distància per infrarojos i sensors òptics.

El sensor es basa simplement en mesurar el temps entre l'enviament i la recepció d'un pols sonor. Sabem que la velocitat del so és 343 m/s en condicions de temperatura 20°C, 50% d'humitat, pressió atmosfèrica a nivell de la mar. Transformant unitats resulta:

$$343 \frac{m}{s} \cdot 100 \frac{cm}{m} \cdot \frac{1}{1000000} \frac{s}{\mu s} = \frac{1}{29.2} \frac{cm}{\mu s} \quad (3.2)$$

És a dir, el so triga 29,2 microsegons a recórrer un centímetre. Per tant, podem obtenir la distància a partir de el temps entre l'emissió i recepció de l'pols mitjançant la següent equació:

$$Distància = \frac{Temps \cdot Velocitat}{2} \quad (3.3)$$

$$Distància(cm) = \frac{Temps(\mu s)}{29.2 \cdot 2} \quad (3.4)$$

El motiu de dividir per dos el temps és perquè hem mesurat el temps que triga el pols a anar i tornar, de manera que la distància recorreguda pel pols és el doble de la que volem mesurar.

Per aquest projecte es pretén utilitzar el sensor d'ultrasons per donar bàsicament dues funcions al robot: que s'aparti o segueixi un objecte o cos quan aquest s'he li acosti.

### 3.2.7 Reguladors de tensió

A causa de que no tots els dispositius necessiten la mateixa quantitat de voltatge, hi ha una varietat de reguladors al mercat que permeten adequar el voltatge. Per saber quin és l'ideal, cal consultar el consum de cada aparell electrònic.

Com s'ha vist en les anteriors descripcions, la bateria que alimenta el sistema és de 11.1V, però no totes les parts que s'han d'alimentar treballen a aquest voltatge. Els dos motors van alimentats directament per la *lipo*, mentre que per la resta d'elements, s'utilitzen dos reguladors de voltatge. Concretament s'utilitza el model LM7805 per passar de 12 a 5V per alimentar el sensor d'ultrasons, el mòdul bluetooth i el microcontrolador Teensy 4.0, i el model LD33 que passa de 5 a 3V per alimentar el driver i la unitat de mesura inercial.

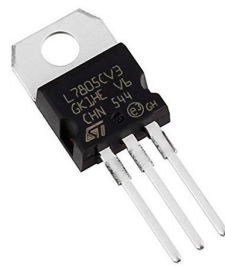


Figura 3.13: Regulador de voltatge

### 3.2.8 Bateria

Per l'autosuficiència energètica del robot, s'ha triat una bateria *lipo*. Tot i ser delicades pel que fa a la càrrega, tenen un bon rendiment, són relativament petites i fàcils d'encabir en dissenys de prototips. Aquests factors fan que sigui la millor opció. Concretament el model de bateria adquirit és de la marca Ovonics, és de 11.1V, 2.200mAh i és de 3 cel·les.



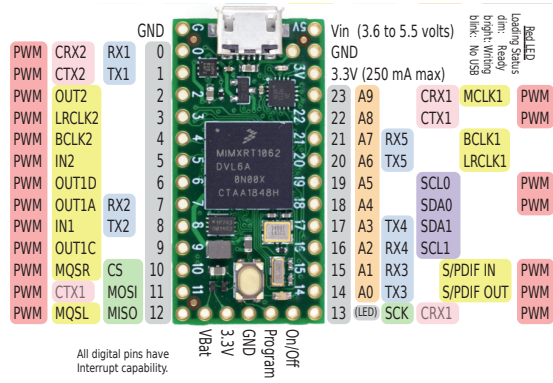
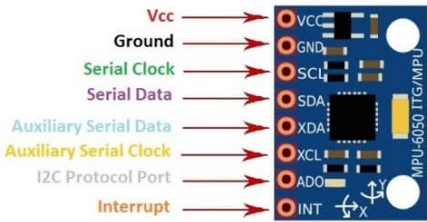
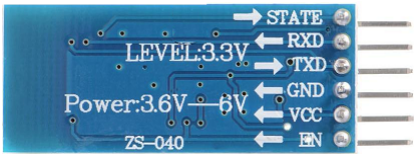

Figura 3.14: Bateria Lipo de la marca Ovonics

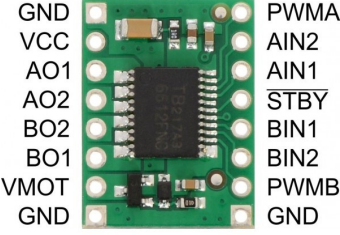

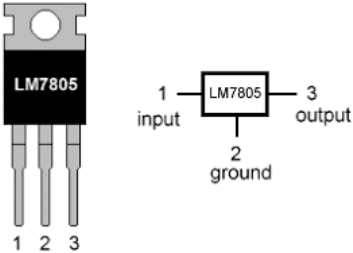
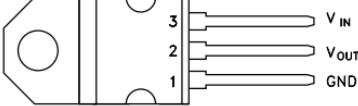



### 3.2.9 Connexions dels dispositius

És molt important realitzar un correcte connexionat entre els diferents dispositius descrits en els punts anteriors. Per aquest projecte, s'ha estudiat el *datasheet* de cada component, i s'han realitzat les connexions necessàries per garantir la comunicació entre ells. Com ja s'ha esmentat, l'alimentació del circuit es fa a través d'una bateria *lipo* d'11.1V. Aquest voltatge serveix per alimentar els motors, però s'ha de reduir per alimentar els altres dispositius que necessiten menys tensió.

La taula que es mostra a continuació resumeix les connexions entre els components:

Component	Figura	Connexions
Teensy 4.0	 <p>All digital pins have Interrupt capability.</p>	0 - TX (bluetooth) 1 - RX (bluetooth) 2 - AIN1 (driver) 3 - AIN2 (driver) 4 - BIN1 (driver) 5 - BIN2 (driver) 6 - PHA (motor1) 7 - PHB (motor1) 8 - PHA (motor2) 10 - PHB (motor2) 11 - INT (imu) 12 - Echo (ultrasons) 14 - Triger (ultrasons) 15 - EN (bluetooth) 18 - SDA (imu) 19 - SCL (imu) 22 - PWMA (driver) 23 - PWMB (driver)
MPU-6050		VCC - (3.3V) GND SCL - T. pin 19 SDA - T. pin 18 INT - T. pin 11
HM-10		VCC - (5V) RXD - T. pin 1 TXD - T. pin 0 EN - T. pin 15
HC-SR04		VCC - (5V) Echo - T. pin 12 Triger - T. pin 14 GND

<p>TB6612FNG</p>		<p>VCC - (5V)          AO1 - Red M1          AO2 - White M1          BO2 - White M2          BO1 - Red M2          VMOT - Bateria          PWMA - T. pin 22          AIN2- T. pin 4          AIN1 - T. pin 3          STBY - T. pin 2          BIN1 - T. pin 5          BIN2 - T. pin 6          PWMB - T. pin 23          GND</p>
<p>Motor 1</p>		<p>Red - AO1 (driver)          Black - GND          Yellow - T. pin 7          Green - T. pin 8          Blue - 3.3V          White - AO2 (driver)</p>
<p>Motor 2</p>	<ul style="list-style-type: none"> <li> Red-Motor +(positive and negative switching can control CW/CCW)</li> <li> Black- Encoder -(positive and negative can not be connected wrong, voltage range is 3.3-5V)</li> <li> Yellow-Encoder A Phase(The motor turns one turn output, 11 signals)</li> <li> Green-Encoder B Phase(The motor turns one turn output, 11 signals)</li> <li> Blue-Encoder+(positive and negative can not be connected wrong, voltage range is 3.3-5V)</li> <li> White-Motor-(positive and negative switching can control CW/CCW)</li> </ul>	<p>Red - BO1 (driver)          Black - GND          Yellow - T. pin 9          Green - T. pin 10          Blue - 3.3V          White - BO2 (driver)</p>
<p>LM7805</p>		<p>1 - Bateria (11.1V)          2 - GND          3 - Vout 5V</p>
<p>LD33</p>		<p>1 - GND          2 - Vout 3.3V          3 - LM7805 (5V)</p>
<p>Bateria</p>		<p>Cable vermell: Vout          Cable negre: GND</p>

Taula 3.1: Descripció de les connexions dels dispositius



Per dissenyar les connexions també s'ha utilitzat el programa Fritzing. Aquest programa fa accessible l'electrònica com a material creatiu per a tothom i fomenta un ecosistema creatiu que permet als usuaris documentar els seus prototips, compartir-los amb altres persones, ensenyar electrònica a l'aula i dissenyar i fabricar PCB professionals. La figura 3.15 mostra l'esquema que s'ha efectuat:

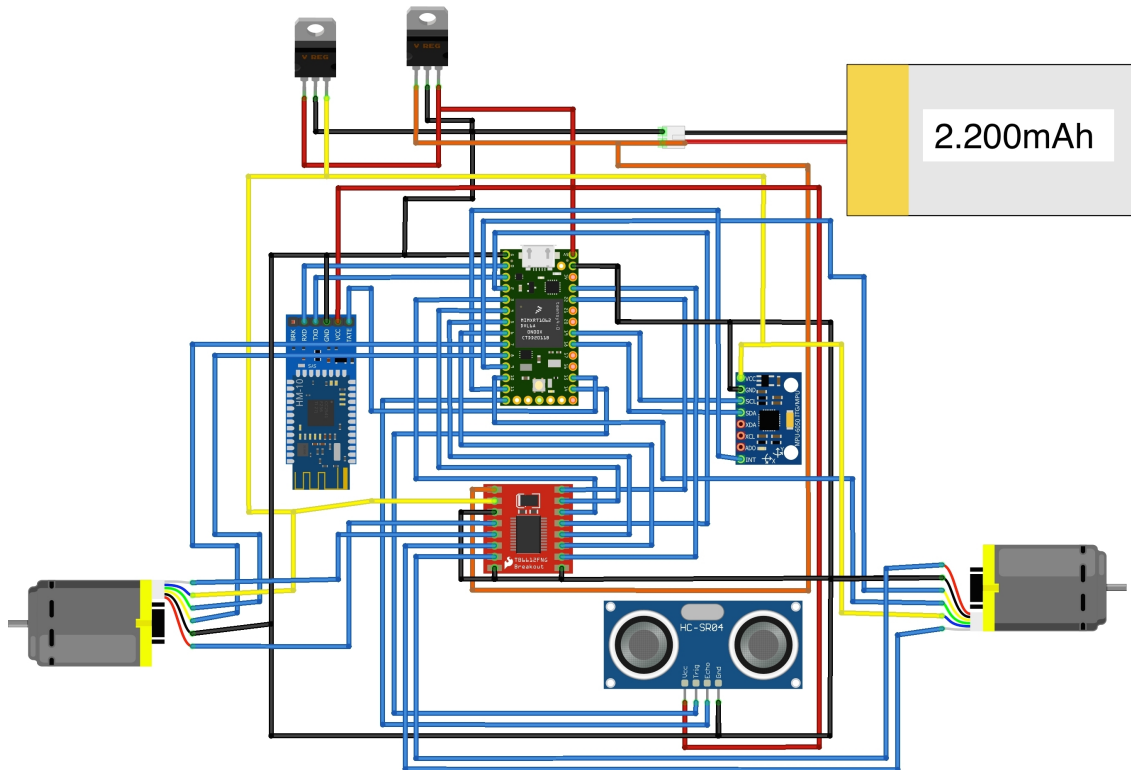


Figura 3.15: Esquema de connexions

### 3.3 Sistema de control

En aquest projecte s'implementa el sistema LQR. Aquest sistema ja ha estat implementat en altres projectes i per tant, no s'ha de fer investigació en crear el control sinó en aplicar-lo en el present robot.

A continuació s'explica el sistema LQR, s'exposen els principis matemàtics i finalment es mostra com s'ha aconseguit els valors de  $K$  per aplicar el regulador al codi del microcontrolador.

#### 3.3.1 Regulador quadràtic lineal (LQR)

Els controladors d'espai d'estat lineal com el controlador de col·locació de pols (RST) i els reguladors quadràtics lineals (LQR) són els dos sistemes de control més populars. La implementació d'aquests controladors va demostrar que el controlador LQR funcionava millor que el controlador de col·locació de pols en equilibrar un pèndol invertit muntat en un braç de rotació. Això es deu al fet que el controlador LQR ofereix un control òptim sobre l'entrada del sistema, tenint en compte els estats del sistema i l'entrada de control. Aquest control òptim busca moure el sistema mitjançant l'entrada, intentant que l'energia necessària sigui la mínima. També millora el procés de prova i error d'altres controladors clàssics com el PID.

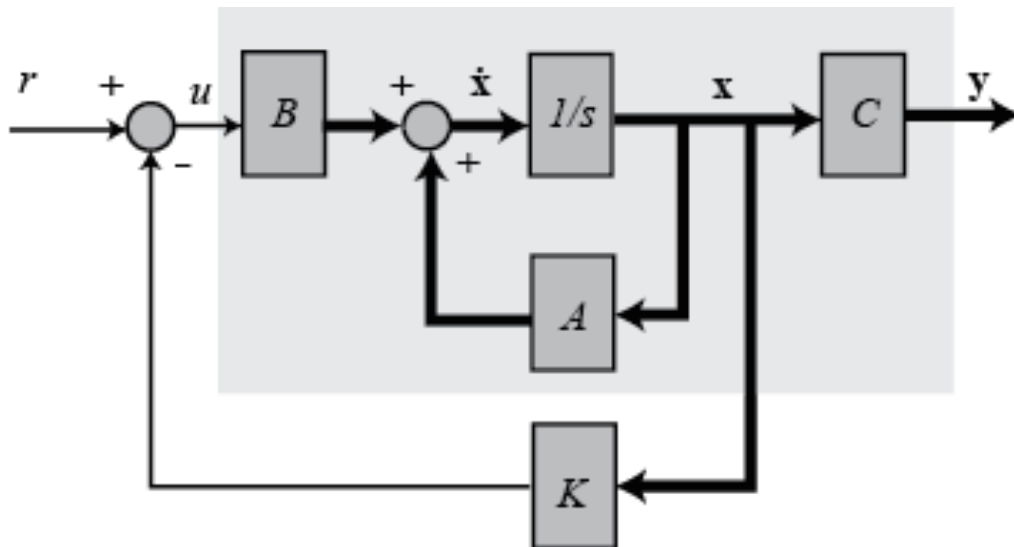


Figura 3.16: Esquema sistema LQR

#### 3.3.2 Principis matemàtics del LQR

Tenint el sistema dinàmic a controlar en l'espai d'estats, el control busca minimitzar l'índex de rendiment  $J$ :

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (3.5)$$

On les matrius  $Q$  i  $R$  són matrius de cost d'acció dels estats i del control respectivament. Són simètriques constants i definides positives:

$$Q = \begin{bmatrix} q_{11} & 0 & \dots & 0 \\ 0 & q_{22} & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \ddots \end{bmatrix}, R = \begin{bmatrix} r_{11} & 0 & \dots & 0 \\ 0 & r_{22} & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \ddots \end{bmatrix} \quad (3.6)$$

L'estat és el vector  $x$  i la acció de control és el vector  $u$ . Això és vàlid per a un sistema de la forma:

$$\dot{x} = Ax + Bu \quad (3.7)$$

On l'acció de control és una llei de guanys  $K$  fixa en el temps per al problema estacionari tal que:

$$u = -Kx \quad (3.8)$$

El vector de guanys  $K$  s'obté de resoldre l'equació algebraica de Riccati, on  $S$  és la matriu incògnita de dimensió  $n \times n$  (on  $n$  és l'ordre del sistema a controlar):

$$A^T S + SA - (SB)R^{-1}(B^T S) + Q = 0 \quad (3.9)$$

$$K = R^{-1}(B^T S) \quad (3.10)$$

A partir d'una combinació inicial de matriu  $Q$  i  $R$ , s'obté per aquest mètode un vector de guanys inicial  $K$ . Aquest vector de guanys inicial resulta un punt de partida per a l'ajust experimental posterior.

### 3.3.3 Càlcul dels valors de $K$

Per aplicar el procés descrit en el punt anterior i obtenir els valors de  $K$ , s'ha utilitzat el codi `lqr_Medbot` (Matlab). La taula 3.2 mostra els paràmetres necessaris que necessita el codi per calcular la  $K$ . Tal i com s'ha mencionat anteriorment, els valors de  $K$  que s'obtenen són orientatius i s'hauran d'ajustant de forma experimental. Tot i això, per garantir un bon punt de partida, és important afinar al màxim els valors de la taula que es mostra. El codi també permet variar els costos de les matrius  $Q$  i  $R$ , afectant directament els valors de  $K$  que s'obtenen.

Paràmetre	Valor	Unitats	Descripció
$g$	9.81	[m/sec <sup>2</sup> ]	Acceleració de la gravetat
$m$	0.05	[kg]	Massa d'una roda
$R$	0.0335	[m]	Radi de la roda
$J_w$	$5.61 \times 10^6$	[kg m <sup>2</sup> ]	Moment d'inèrcia de la roda
$M$	0.85	[kg]	Pes del cos del robot
$W$	0.125 - 0.195	[m]	Amplada

D	0.08	[m]	Profunditat
H	0.100 - 0.135	[m]	Alçada
L	0.0323	[m]	Dist. del centre de massa a l'eix de les rodes
R <sub>m</sub>	7.15	[Ω]	Resistència del motor
n	45:1	-	Relació de transmissió
$f\pi$	0.8 - 1	-	Coefficient de fricció entre les rodes i el terra

Taula 3.2: Taula de paràmetres per calcular els valors de K

Els resultats obtinguts, mantenint un cost mig tant a Q com a R són:

$$K = \{ 0.7, 20, 0.7, 0.8 \}.$$

## 3.4 Programació

El microcontrolador Teensy 4.0 és programable amb entorn Arduino. Això permet la utilització de totes les funcions pròpies d'Arduino. A més, disposa d'un ampli ventall de llibreries que faciliten la programació dels dispositius.

El programa realitzat implementa les connexions descrites a la taula 3.1. Pel que fa al microcontrolador, destacar connexions com: PWM (drivers), SCL i SDA (connexió I2C) i Rx i TX (connexió bluetooth).

També s'inclouen llibreries com:

- *Software Serial.h*
- *encoders.h*
- *elapsedMillis.h*
- *MPU6050\_6Axis\_MotionApps\_moi.h*
- *SparkFun\_TB6612.h*

El programa que permet l'auto equilibrat del robot es mostra a l'apèndix A d'aquest document. A continuació, el diagrama de la figura 3.17 resumeix el funcionament seqüencial del programa:

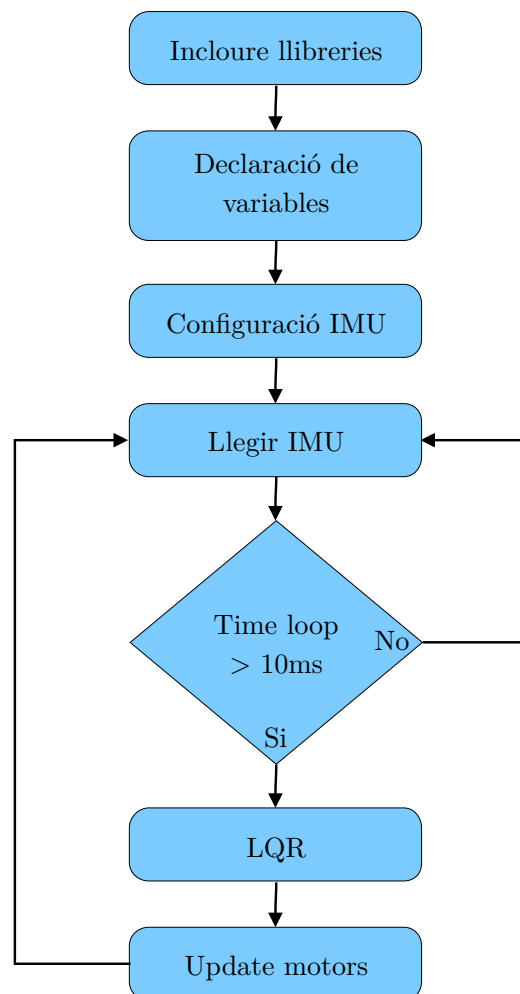


Figura 3.17: Diagrama de funcionament del programa

# Capítol 4

## Cost econòmic del prototip

Minimitzar el preu total del robot ha sigut un objectiu més d'aquest projecte, per tant el factor econòmic s'ha tingut sempre present. Pel que fa a costos materials, s'han requerit elements relativament diversos, els quals s'han hagut de comprar en diferents proveïdors. Això a suposat que es tingués cura, en la mesura del possible, de no acumular costos d'enviaments. A pesar de tot, s'han hagut de fer compres de petit import, on el transport ha sigut més car que el producte en si. En alguna ocasió també s'ha hagut de comprar més quantitat de la que es necessitava. Tot i això, les compres realitzades han estat a preus competitiu. Principalment s'ha comprat de forma online a través de Amazon i BricoGeek.

### 4.1 Desglos dels costos

A continuació es desglossa el cost total que ha suposat l'adquisició del material necessari pel prototip:

Component	Quantitat	Preu
Microcontrolador Teensy 4.0	1	21.95 €
ARCELI GY-521 MPU6050	1	6.85 €
Infitronic - 10 x Brides de "velcro". (16x160 mm, blau)	1	4.90 €
OVONIC Lipo RC bateria 11.1V 2200mAh 50C 3S	1	25.99 €
Carregador Lipo de 80W 6A I-MAX B6	1	34.0 €
15 peces de 40 pins mascle connexió femella + 15 peces de 40 pins mascle connexió mascle.	1	7.99 €
Mòdul Bluetooth BLE 4.0, AT-09 - CC2541 (HM-10)	1	3.50 €
Controlador de motors TB6612FNG	1	4.35 €
DC 12V DIY Encoder Motor reductor (130RPM) + suport motor + rodes de diàmetre 65 mm.	2 (x 12.10€)	24.20 €
Mòdul HC-SR04 sensor d'ultrasons	1	2.70 €

Laserplast - Metacrilat transparent 3x200x200 mm.	1	9.50 €
Placa alumini 10x100x100 mm.	1	8.50 €
Joc de 200 separadores hexagonals de llautó i cargols d'acer inoxidable.	1	9.85 €
Regulador de tensió 3.3V - LD33	1	2.10 €
Regulador de tensió 5V - LM7805	1	2.10 €
Material divers utilitzat en el muntatge electrònic (cables elèctrics, connectors, tub retràctil, etc)	-	10.0 €
Costos d'enviament Bricoggek	-	6.05 €
	<b><i>TOTAL</i></b>	184.53 €

Taula 4.1: Costos del prototip

# Capítol 5

## Resultats

En aquest capítol es fa una reflexió dels objectius plantejats a l'inici d'aquest document envers els resultats aconseguits. Es valoren tant els punts forts com els febles, i també els aspectes a millorar.

Seguint la llista d'objectiu inicials, s'observa que:

- S'ha realitzat un disseny mecànic que ha assolit les perspectives desitjades. S'ha aconseguit un disseny simple, robust, lleuger, i fàcil de muntar i desmuntar.
- El disseny de la part electrònica també ha assolit complir els requeriments del sistema i a través un clar i senzill connexionat dels diferents elements.
- S'ha fet una correcta selecció de dispositius electrònics, que han permès que el robot tingui la funcionalitat esperada.
- S'han comprès les equacions matemàtiques que representen el balanceig i el gir del model.
- S'ha fet un anàlisi de diferents projectes de control basats en LQR, per trobar la forma d'auto equilibrar el robot i aplicar el control del seu gir.
- S'ha dissenyat un programa per el microcontrolador capaç d'enllaçar tots els sensors i actuadors. S'ha aconseguit aplicar l'algoritme del control LQR i satisfer l'objectiu de l'auto equilibrat del robot.

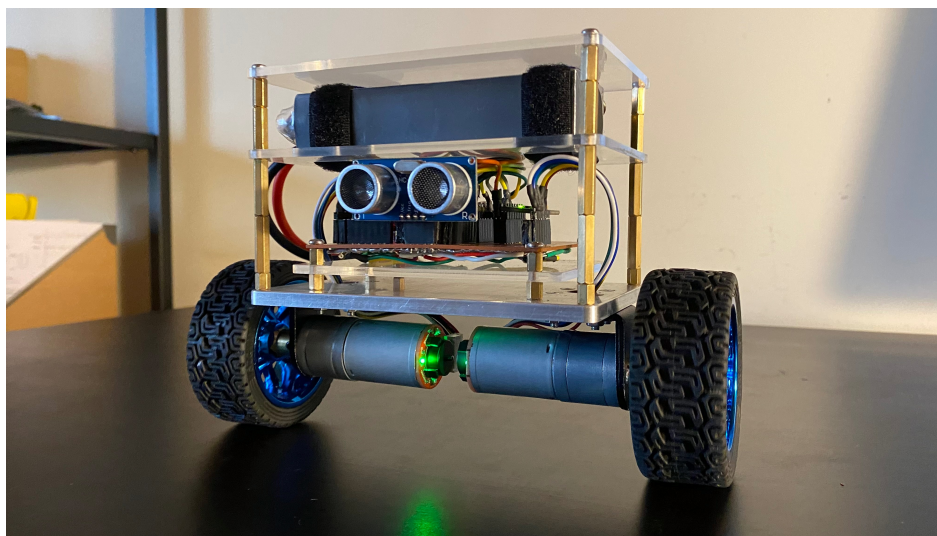


Figura 5.1: Prototip



Pel que fa els aspecte a millorar, es pot dir que:

- Queda pendent instal·lar un interruptor on/off físic en el robot.
- Aplicar el gir del robot en l'algorisme de control.
- Ampliar les funcions del robot pel que fa als seus sensors de bluetooth i ultrasons. Aconseguint que el robot tingui diferents modes de funcionament.

# Capítol 6

## Conclusions

Després d'introduir-me en el món de la robòtica mòbil a través d'aquest projecte, puc dir que els robots auto equilibrats són un tipus de robot dignes de ser estudiats. Les seves característiques són idònies per posar en pràctica els conceptes estudiats durant el grau d'enginyeria mecatrònica i també per plantejar-ne de nous. Enllaçar les parts mecatròniques d'un projecte és un repte que requereix una visió global: la connexió de les seves parts mecàniques, elèctriques, de control i de programació suposa un repte amb moltes possibles solucions.

Pel que fa els resultats, tant el disseny mecànic com la part electrònica compleixen les característiques descrites en els objectius. Per l'estabilització del robot, s'ha comprovat que el sistema LQR és eficient i s'ha aconseguit l'auto equilibrat del robot. Tot i això aquesta és la part amb més possibilitats i falta experimentar més per implementar el control en el gir.

Considero assolits els objectius plantejats a l'inici del projecte, però no finalitzats. Es pot ampliar la part del gir i també les funcions del robot. D'altre banda, el prototip construït pot servir de base per experimentar noves idees, aplicant diferents sistemes de control i formes de programar el codi del microcontrolador.

Personalment, estic satisfet de finalitzar el grau amb la realització d'un projecte d'aquestes característiques, és una motivació més per seguir endavant, aprenent dels encerts, i també dels errors. Una nova experiència que segur que servirà per futurs projectes.

# Bibliografia

- [1] Contingut d'informació sobre el funcionament del microcontrolador i exemples d'ús. URL: <https://www.pjrc.com/store/teensy40.html>
- [2] Article de recerca: “*The LQR Controller Design of Two-Wheeled Self-Balancing Robot Based on the Particle Swarm Optimization Algorithm*”. Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2014, Article ID 729095, 6 pages. URL: <http://dx.doi.org/10.1155/2014/729095>
- [3] Contingut referent al món Arduino. URL: <https://www.luisllamas.es/>
- [4] Article: Eyantra “Robotic Competition 2019-2020”. Implementació del control LQR.
- [5] Memòria projecte: “Diseño y construcción de un robot balancin mediante Arduino” Universidad pública de Navarra.
- [6] Memòria projecte: “Disseny i construcció d'un segway” Universitat de Vic.

# Apèndixs

## A. Programa

```
/*=====
                Treball de fi de grau
                CONTROL D'UN ROBOT AUTO EQUILIBRAT
                Autor: Jordi Soldevila Casas
                Tutor: Moisès Serra Serra
                =====*/

//Incloure llibreries
#include <SoftwareSerial.h> //bluetooth
#include "MPU6050_6Axis_MotionApps_moi.h" //MPU
#include <Encoder.h> //encoders
#include <SparkFun_TB6612.h> //driver TB6612FNG

//Valors de K (control LQR)
float k[4]={0.7,20 ,0.7,0.8};
//k[0]: valor de k per posició
//k[1]: per velocitat
//k[2]: per pos. angular
//[3]: per vel. angular
//Declaració de variables:
float v; //Tensió necessària per equilibrar el robot
float vsat; //Tensió v en valors PWM
float averagePos; //La mitja dels dos encoders (posició)
float previous_averagePos; //Posició anterior
float averageWpos; //posició angular
float dT; //variable epl control del temps
elapsedMillis timerLoop; //"timers" elapsed
elapsedMillis one_by_dT;
float RollOffset = -0.03; //Desfasament angular inicial
//Variables de referència per la funció LQR
float refAveragePos = -1;
float refAverageWpos = 0;
float refTheta = 0;
float refWtheta = 0;

//Variables encoders
float pos1 = 0; //posició inicial del encoders
float pos2 = 0;
float encPos1 = 0; //variables funció readEncoders()
```

```

float encPos2 = 0;
long oldPosition1 = -999;
long oldPosition2 = -999;

//Driver
#define AIN1 3 //motor1
#define BIN1 5
#define AIN2 4 //motor2
#define BIN2 6
#define STBY 2 //stanby TB6612FNG
#define PWMA 22
#define PWMB 23

//Motors
//Aquestes constants s'utilitzen per configurar el motor
//El valor poden ser 1 o -1
const int offsetA = -1;
const int offsetB = 1;
//Definir motors (SparkFun_TB6612.h)
Motor motor1 = Motor(AIN1, AIN2, PWMA, offsetA, STBY);
Motor motor2 = Motor(BIN1, BIN2, PWMB, offsetB, STBY);

//Definir encoders (Encoders.h)
Encoder myEnc1(7, 8);
Encoder myEnc2(9, 10);

//Definir mòdul bluetooth (SoftwareSerial.h)
SoftwareSerial HM10(0, 1); //RX=0, TX=1
//variables char i string per la funció listenBLE()
char appData;
String inData = "";

//MPU
MPU6050 mpu;
//Variables
float thetaX;
float thetaZ;
float dthetaX;
float dthetaZ;
uint32_t Ts = 10;
float roll, pitch, yaw;
float AngularVx, AngularVy, AngularVz;
const byte interruptPin = 11;

//configuració

```

```

volatile bool mpuInterrupt = false;
void dmpDataReady() {mpuInterrupt = true;}; //interrupció, 10ms
//MPU control d'estat
bool dmpReady = false; //defineix true si l'inici de DMP és ok
uint8_t mpuIntStatus; //manté el byte d'estat d'interrupció real
uint8_t devStatus; //retorn de l'estat després de cada operació del
dispositiu (0=ok,!0=error)
uint16_t packetSize; //mida prevista del paquet DMP
uint16_t fifoCount; //recompte de tots els bytes a FIFO
uint8_t fifoBuffer[64]; //Memòria intermèdia FIFO
//orientació / moviment
Quaternion q; //[w, x, y, z] quatèrnió
VectorInt16 aa; //[x, y, z] mesures del sensor d'accel.
VectorInt16 aaReal; //[x, y, z] mesures sensor d'accel. sense
gravetat
VectorInt16 aaWorld; //[x, y, z] mesures del sensor d'accel.
fotograma mundial
VectorInt16 gyro; //[x, y, z] valors giroscopi
VectorFloat gravity; //[x, y, z] vector de gravetat
float ypr[3]; //[yaw, pitch, roll] yaw/pitch/roll

//-----
void setup(){
  Serial.begin(1000000); //configurar serial baud rate
  HM10.begin(9600); //HM10 serial a 9600
  setupIMU(); //configuració MPU
  //!durant el segons que dura el setup, s'ha de mantenir el
  //robot fixe i en posició vertical
}
void loop(){
  pos1 = readEncoder1(); //lectura encoder (motor1)
  pos2 = readEncoder2(); //lectura encoder (motor2)
  averagePos = 0.5 * (pos1 + pos2); //posició mitja del robot
  dT = one_by_dT; //es guarda el temps transcorregut
  one_by_dT = 0; //es posa la variable elapsed a 0
  //es calcula la velocitat del robot
  averageWpos = ((averagePos - previous_averagePos) * 1000 / dT);
  previous_averagePos = averagePos; //guardar posició actual
  readIMU(); //lectura del MPU
  thetaX = roll + RollOffset; //roll offset (ajust offset)
  dthetaX = AngularVx; //es guarda la lectura del giroscopi
  if (timerLoop > Ts){ //el sistema actua cada 10ms
    timerLoop = 0;
    LQR(); //funció LQR
    updateMotors();} //actualització dels motors
}
//-----

```

```

void LQR() { //funció LQR
    v = 0;
    //algoritme LQR
    v -= k[0] * (refAveragePos - averagePos);
    v -= k[1] * (refTheta - thetaX);
    v -= k[2] * (refAverageWpos - averageWpos);
    v -= k[3] * (refWtheta - dthetaX);
    vsat = v * 255 / 11.2; //conversió de voltatge a senyal PWM
    if (vsat > 255)vsat = 255; //es restringeix vsat màxim
    if (vsat < -255)vsat = -255; //es restringeix vsat mínim
}

void updateMotors() { //Act. motors (llibreria SparkFun_TB6612)
    if (vsat < 0){back(motor1,motor2, vsat);} //si vsat és
    negatiu -> motors "back"
    else{forward(motor1, motor2, vsat);} //si vsat és positiu ->
    motors "front"
}

float readEncoder1() { //funcio per llegir l'encoder 1
    long newPosition1 = myEnc1.read();
    //es multiplica per -1 perquè el robot té els motors colocats
    simetricament
    newPosition1 = newPosition1 * (-1);
    if (newPosition1 != oldPosition1){
        oldPosition1 = newPosition1;}
    encPos1 = TicksToRad(newPosition1); //TicksToRad per passar la
    lectura a radians
    return encPos1;}

float readEncoder2() { //funcio per llegir l'encoder 2
    long newPosition2 = myEnc2.read();
    if (newPosition2 != oldPosition2) {
        oldPosition2 = newPosition2;}
    encPos2 = TicksToRad(newPosition2); //TicksToRad per passar
    la lectura a radians
    return encPos2;}

float TicksToRad (int x){ //per passar les lectures a rad.
    float rad = x * ((2 * PI) / 1980); //1980 constant encoders
    return rad; }

String listenBLE(){ //Per llegir les dades rebudes al bluetooth
    HM10.listen(); //listen port HM10
}

```

```

while (HM10.available() > 0) { //si HM10 fa algun eviament:
  llegeix
  appData = HM10.read();
  inData = String(appData);} //es guarda la dada en format
  string
  return inData;
}
//-----

//ReadIMU parameters, 10ms
void readIMU() {
  //si la programació ha fallat, no intenteu fer res
  //espereu la interrupció del MPU o els paquets addicionals
  disponibles
  while (!mpuInterrupt && fifoCount < packetSize) {}
  mpuInterrupt = false; //restablir el senyal d'interrupció i
  obtenir INT_STATUS byte
  mpuIntStatus = mpu.getIntStatus();
  fifoCount = mpu.getFIFOCount(); //obtenir el recompte FIFO actual
  //comproveu si hi ha desbordament
  if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
    mpu.resetFIFO(); //restableix perquè pugui continuar
    Serial.println(F("FIFO overflow!")); }
  else if (mpuIntStatus & 0x02) {
    //espereu la longitud correcta de les dades disponibles
    while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();
    mpu.getFIFOBytes(fifoBuffer, packetSize); //llegir paquet FIFO
    //fa un seguiment del recompte FIFO
    fifoCount -= packetSize;
    //angle i velocitat angular són obtinguts en radians
    //la suma del valor és l'offset per tal d'acabar
    d'ajustar-lo a 0
    int16_t gyro[3];
    mpu.dmpGetGyro(gyro, fifoBuffer);
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    AngularVx = (gyro[0] * 250 / 32768); //reducció per la relació
    de sensibilitat del giroscopi
    AngularVy = (gyro[1] * 250 / 32768);
    AngularVz = (gyro[2] * 250 / 32768);
    yaw = ypr[0]; //variables yaw, pitch & roll
    pitch = ypr[1];
    roll = ypr[2];
    return;
  }
}

```



```

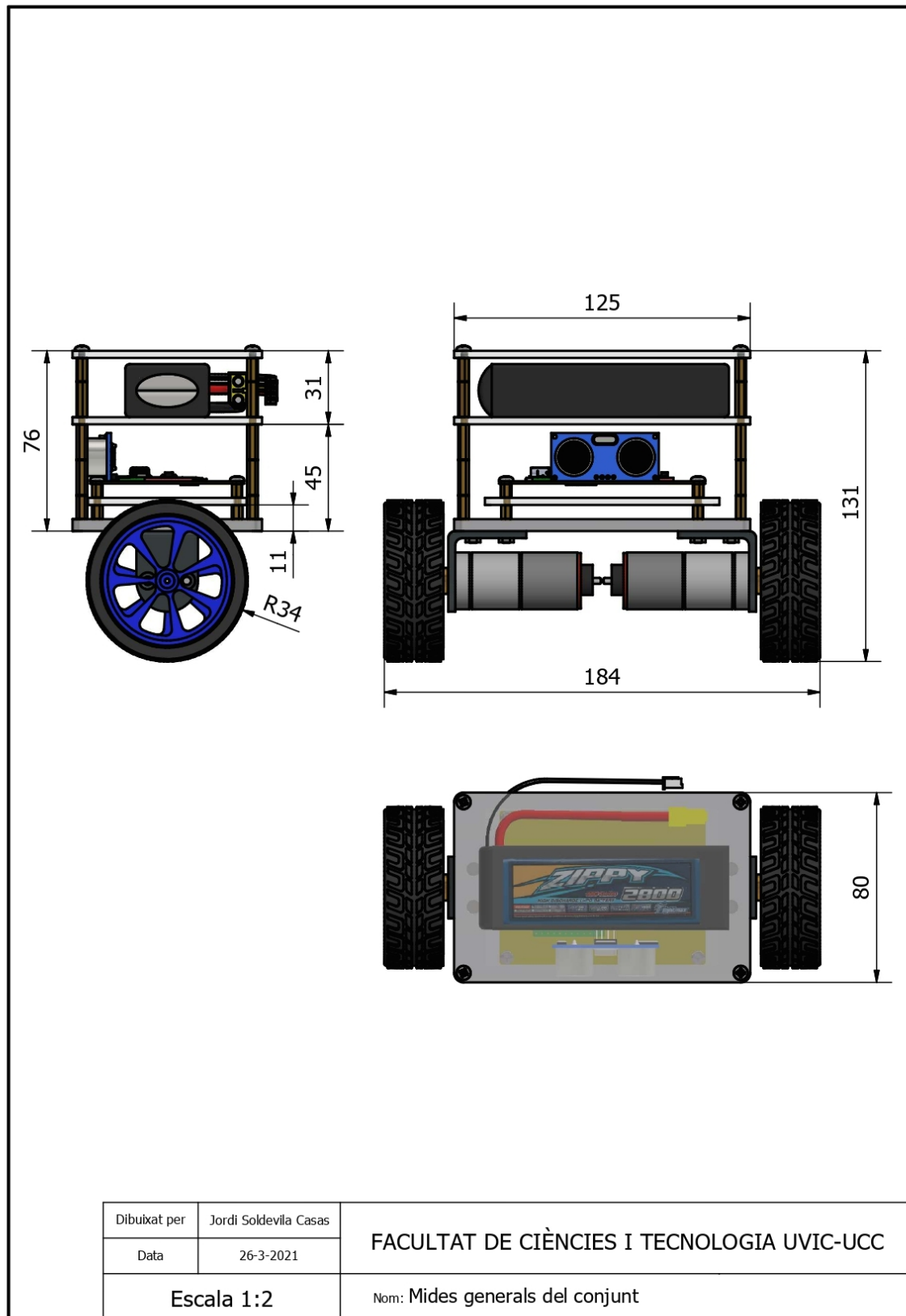
    }
    return;
}

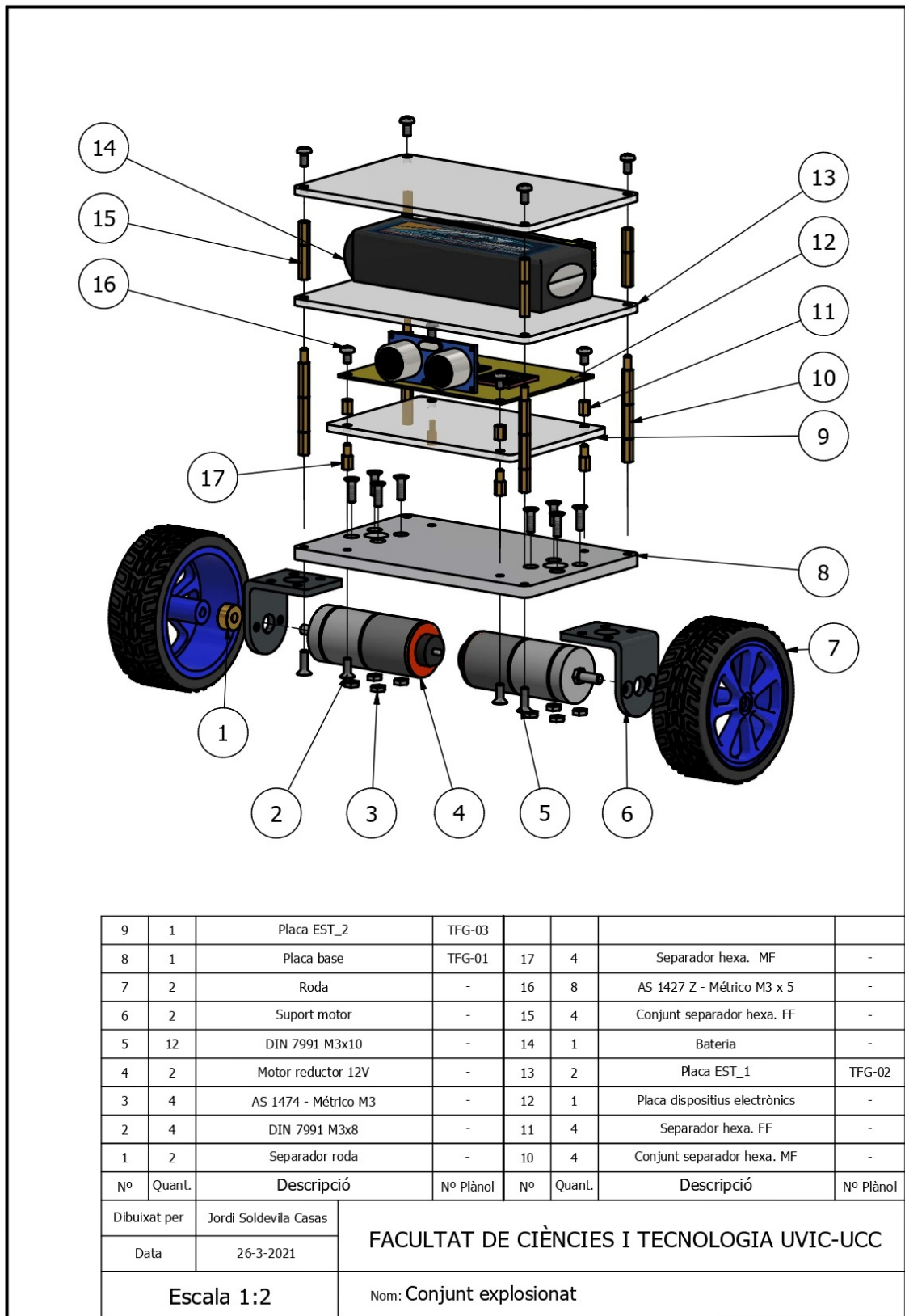
void setupIMU() {
    pinMode(18, INPUT_PULLUP); //entrada SDA i SCL del Teensy.
    pinMode(19, INPUT_PULLUP);
    pinMode(interruptPin, INPUT); //pin d'interruptió
    Wire.begin(); //coneexió
    mpu.initialize(); //Wire.setClock(40000);
    Serial.println(F("Testing device connections...")); //verificar
    connexió
    Serial.println(mpu.testConnection() ? F("MPU6050 connection
    successful") : F("MPU6050 connection failed"));
    Serial.println(F("Initializing DMP...")); //carregar i
    configurar el DMP
    devStatus = mpu.dmpInitialize();
    //introduir les pròpies compensacions      giroscòpiques
    mpu.setXGyroOffset(105);
    mpu.setYGyroOffset(-248); //ajustades per a una
    sensibilitat mínima.
    mpu.setZGyroOffset(34);
    mpu.setZAccelOffset(1050);
    if (devStatus == 0) { //Temps de calibratge: genera
    compensacions i calibra l'MPU6050
        delay(2000);
        //mpu.CalibrateAccel(20); //temps de calibració //
        mpu.CalibrateGyro(20);
        mpu.PrintActiveOffsets();
        Serial.println(F("Enabling DMP...")); // turn on the DMP,
        now that it's ready
        mpu.setDMPEnabled(true);
        //activa la detecció d'interrupcions d'Arduino
        Serial.print(F("Enabling interrupt detection (Arduino
        external interrupt "));
        Serial.print(digitalPinToInterrupt(interruptPin));
        Serial.println(F(")..."));
        attachInterrupt(digitalPinToInterrupt(interruptPin),
        dmpDataReady, RISING);
        mpuIntStatus = mpu.getIntStatus(); Serial.println(F("DMP ready!
        Waiting for first interrupt..."));
        dmpReady = true;
        //obtenir la mida esperada del DMP per a una comparació
        posterior
        packetSize = mpu.dmpGetFIFOpacketSize();

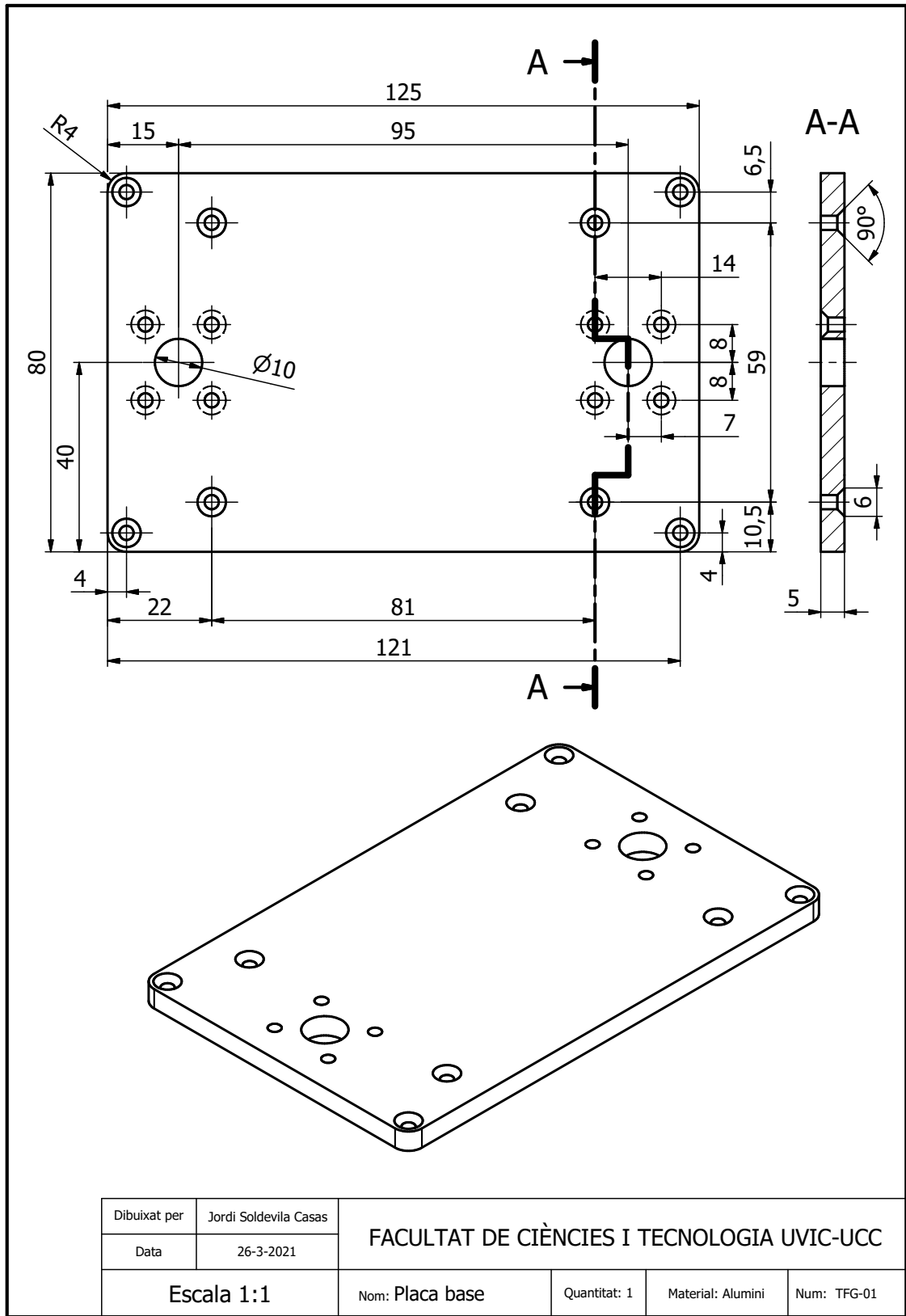
```

```
}  
else {  
    // ERROR!  
    // 1 = ha fallat la càrrega inicial de la memòria.  
    // 2 = Les actualitzacions de configuració de DMP han  
    fallat  
    Serial.print(F("DMP Initialization failed (code ")");  
    Serial.print(devStatus); Serial.println(F(")"));  
}  
}
```

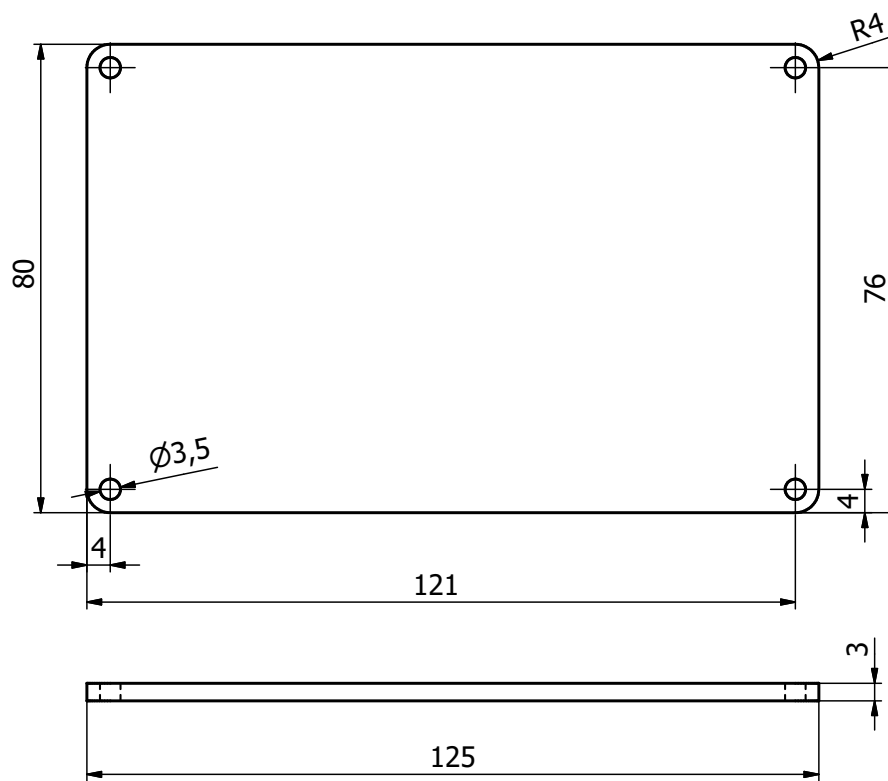
## B. Plànols



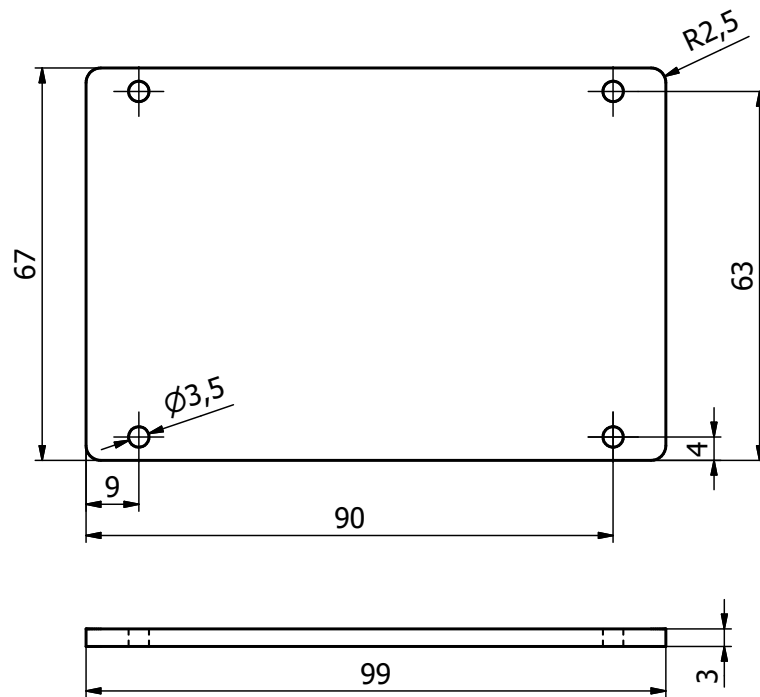




Dibuixat per	Jordi Soldevila Casas	FACULTAT DE CIÈNCIES I TECNOLOGIA UVIC-UCC			
Data	26-3-2021				
Escala 1:1		Nom: Placa base	Quantitat: 1	Material: Alumini	Num: TFG-01



Dibuixat per	Jordi Soldevila Casas	FACULTAT DE CIÈNCIES I TECNOLOGIA UVIC-UCC			
Data	26-3-2021				
Escala 1:1		Nom: Placa ESP_1	Quantitat: 2	Material: Metacrilat	Num: TFG-02



Dibuixat per	Jordi Soldevila Casas	FACULTAT DE CIÈNCIES I TECNOLOGIA UVIC-UCC			
Data	26-3-2021				
Escala 1:1		Nom: Placa ESP_2	Quantitat: 1	Material: Metacrilat	Num: TFG-03

# C. Esquema elèctric

