

INSTITUT JAUME CALLÍS

Departament de tecnologia

Disseny i desenvolupament d'un robot gestor d'aparcaments

Autor: **Genís Puntí Ricart**

Tutor: **Joan Miarons Serra**

2n de batxillerat C

2025-2026



Resum

Les ciutats intel·ligents busquen optimitzar els serveis urbans amb l'objectiu de millorar la qualitat de vida i augmentar l'eficiència i la sostenibilitat. L'objectiu d'aquest treball de recerca serà desenvolupar una proposta per millorar la gestió dels aparcaments utilitzant les característiques de les ciutats intel·ligents. Es construirà i programarà un prototip amb la funció d'aparcar els cotxes dins d'un aparcament de la forma òptima. Per això, caldrà fer recerca dels diferents components electrònics i mecànics que seran necessaris per aconseguir crear aquest prototip. Al mateix temps, es farà ús de llenguatges de programació per fer-lo funcionar i també de la impressió 3D per fabricar-ne alguns components. A banda d'això, s'introduiran alguns algorismes que m'ajudaran a generar la disposició de places més eficient dins d'un aparcament i a trobar el camí més ràpid per aparcar un vehicle. Finalment, s'avaluarà la precisió i les capacitats del producte final i es buscaran millores per optimitzar la seva funció.

Abstract

Smart cities seek to optimise urban services with the aim of improving quality of life and increasing efficiency and sustainability. The objective of this research project will be to develop a proposal to improve car park management using smart city features. A prototype will be built and programmed with the function of parking cars within a car park in an optimal manner. To do this, it will be necessary to research the different electronic and mechanical components that will be needed to create this prototype. At the same time, programming languages will be used to make it work, and 3D printing will be used to manufacture some components. In addition, some algorithms will be introduced to help me generate the most efficient layout of parking spaces within a car park and find the fastest route to park a vehicle. Finally, the accuracy and capabilities of the final product will be evaluated and improvements will be sought to optimise its performance.

Agraïments

M'agradaria donar les gràcies, primer de tot, al tutor del meu treball de recerca, Joan Miarons Serra, per l'orientació i els consells que m'ha donat. També voldria mostrar el meu agraïment a Salvador Puntí Vila, que treballa a Bonpreu, per haver-me donat l'oportunitat de visitar les seves instal·lacions; a Bob Bao i Dom Qin, que treballen a EasyEDA, pel seu suport i col·laboració amb material d'electrònica; i a Dmitry Borisov, pel seu model *Mecanum Wheel*. De la mateixa manera, agrair a l'Institut de Gurb i a l'Institut Jaume Callís per haver-me deixat utilitzar les seves instal·lacions i eines al llarg del treball. Igualment, estic agraït als professors de matemàtiques, tecnologia i informàtica de l'institut per ensenyar-me i fer-me interessar en les matemàtiques i la programació. Així mateix, m'agradaria agrair a la meva família pel suport que m'han donat i per repassar el treball, especialment, al meu avi Salvador per la seva ajuda en tot moment. I finalment, a tota la gent que de manera directa o indirecta han fet possible aquest treball, i a tothom que s'hagi interessat per ell.

Índex

Índex	3
Presentació	6
I Recerca teòrica	7
1 Smart cities	7
1.1 Visita a les instal·lacions de Bonpreu Esclat	8
2 Idea inicial	10
3 Recerca sobre maquinària (Hardware)	13
3.1 Actuadors	13
3.1.1 Motor amb caixa reductora	13
3.1.2 Motor pas a pas 28BYJ-48	15
3.1.3 Motor servo MG90S	17
3.2 Sensors	20
3.2.1 Mòdul de comunicacions BU01	20
3.2.2 Sensor de distància VL53L0X	22
3.2.3 Acceleròmetre i giroscopi GY-512	23
3.3 Microcontrolador - ESP32	28
3.4 Rodes Mecanum	29
4 Disseny del prototip físic del robot	34
4.1 Impressió 3D	34
4.2 Disseny 3D	35
4.3 Disseny 3D del prototip	36
4.3.1 Versió 1	36
4.3.2 Versió 2	38
4.3.3 Versió 2.1	39
4.3.4 Versió 3	40
4.3.5 Versió 3.1	42
4.4 Disseny 3D de les rodes	43
4.5 PCB	43
4.6 Disseny PCB	44
4.7 Prototips PCB	44

4.7.1	Versió 1	45
4.7.2	Versió 2	46
5	Recerca sobre programari (Software)	47
5.1	Proposta de tecnologies	47
5.2	Arxius DXF	47
5.3	Python	47
5.3.1	Python bàsic	48
5.3.2	pygame	49
5.3.3	ezdxf	50
5.3.4	Google Cloud Firestore	51
5.3.5	requests	52
5.3.6	flask	52
5.3.7	Llibreries built-in	53
5.4	ESP32	54
5.5	Algorismes	54
5.5.1	Algorisme de cerca A*	55
5.5.2	Algorisme de generació d'aparcaments	58
II	Implementació pràctica	63
6	Construcció	63
6.1	Determinació velocitat del robot	66
7	Programació	68
7.1	Programació per generar el mapa de l'aparcament	68
7.2	Programació per simular l'aparcament	68
7.3	Programació del robot	69
7.4	Programació de l'aplicació mòbil	70
7.5	Resum sobre la programació	70
8	Cost del prototip	71
III	Tancament i propostes	72
9	Resultats	72
9.1	Producte obtingut	72
9.2	Precisió del prototip	74

9.3 Possibles millores	76
10 Conclusions	77
Fonts d'informació	79
 IV Annexos	 83
A Eines emprades en la realització del treball	83
A.1 Sistema operatiu	83
A.2 Confecció del document escrit	83
A.2.1 L ^A T _E X i TeXworks	83
A.2.2 Microsoft Paint	84
A.3 Desenvolupament del programari	84
A.3.1 Visual Studio Code	84
A.3.2 Git i GitHub	84
B Esquemàtics de les PCB	85
C Fitxers de dades (Datasheets)	86

Presentació

El concepte *Smart City* o, en català, ciutat intel·ligent, és un terme que últimament és molt popular i utilitzat en el món de la tecnologia i fins i tot en política. Segons la Generalitat de Catalunya, aquest terme denota una ciutat dotada de mecanismes basats en tecnologies de la informació i comunicació i enfocats a millorar tant la gestió dels diferents serveis de la ciutat com la qualitat de vida dels seus habitants. Per tant, consisteix en una ciutat que millora tots els aspectes que pot fent ús de la tecnologia.

Un dels màxims inconvenients de les grans ciutats és la mobilitat. Les persones es poden desplaçar utilitzant el transport públic o el transport privat. En aquest treball, em proposo millorar la gestió del transport privat, concretament la gestió de l'aparcament dels vehicles privats. Aquesta proposta buscarà complir amb les característiques de les ciutats intel·ligents i optimitzar, tant en temps com en espai, l'ús dels aparcaments tant públics com privats.

Ja de petit he gaudit la robòtica, la programació i les matemàtiques. Això em va portar a fer diverses extraescolars i diferents concursos per portar a la pràctica els meus coneixements i satisfer la meva curiositat. A partir d'aquest punt vaig començar a aprendre diferents llenguatges de programació i eines tant del món de les matemàtiques com de la robòtica de forma autodidàctica. Totes aquestes experiències m'han portat fins aquí, a fer un treball de recerca que gira al voltant d'aquests tres mons i que ara tinc la possibilitat d'aplicar en una situació real.

Part I

Recerca teòrica

1 Smart cities

[41] Per trobar l'origen d'aquest concepte hem d'anar enrere en el temps fins al segle XV quan Leonardo da Vinci va introduir el terme de planificació urbana, inspirat en el fet que les malalties feien més vulnerables les ciutats que no pas el camp. Va ser per aquest motiu que va fer els plànols per un dels seus molts invents, la ciutat planificada, dissenyada per ser higiènica i habitable.

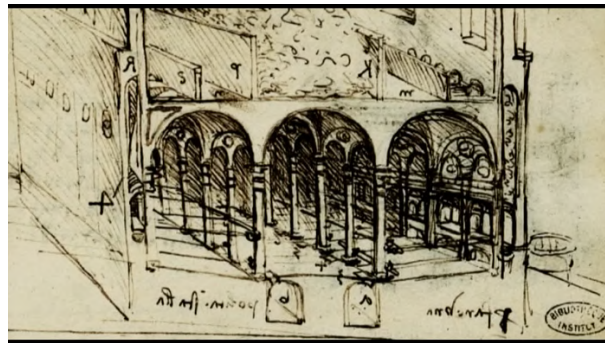


Figura 1: Un esbos de la ciutat ideal per Da Vinci. Font: [4]

Entre 1913 i 1914 es van plantejar esbossos i projectes de la ciutat utòpica coneguda com a Nova Ciutat a través del manifest futurista escrit per Filio Tommaso Marientti el 1908. [42] Anys més tard, entre el 1960 i el 1970, van aparèixer les primeres aplicacions informàtiques a la gestió urbana, ciutats com Los Angeles feien servir anàlisis computacionals per mapar dades socials i dirigir les polítiques locals. [40] Entre el 1980 i el 1990 van sorgir les primeres estratègies nacionals i infraestructures digitals, exemples com la creació del National Computer Board de Singapur mostren l'interès estatal per la computarització i la planificació a escala nacional. [8] Durant la dècada dels 2000 apareixen les primeres definicions acadèmiques i les primeres polítiques “smart”. Alguns articles acadèmics ofereixen la definició que una ciutat és “smart” quan les inversions en capital humà i infraestructures impulsen el creixement sostenible i la qualitat de vida.



Figura 2: Poster de la campanya publicitària IBM Smarter Planet. Font: [23]

[24] Entre els anys 2008 i 2011 hi va haver una gran difusió comercial, discursos i campanyes com “IBM Smarter Planet” i altres programes que van popularitzar el concepte i van convertir empreses tecnològiques en actors centrals de l’agenda “smart”. [43] [21] Durant la dècada del 2010 van aparèixer els primers projectes emblemàtics com Songdo a Corea del Sud i les polítiques “smart” de Barcelona.

Tots aquests moments històrics ens han portat al que són avui en dia les “smart cities”. La seva definició segons diverses fonts és: són ciutats que integren tecnologies digitals amb polítiques públiques i participació ciutadana per optimitzar els serveis urbans —mobilitat, energia, aigua, residus, salut, seguretat— amb l’objectiu de millorar la qualitat de vida, augmentar l’eficiència i la sostenibilitat, i reforçar la transparència i la governança democràtica.

Per tant, durant el desenvolupament de la meva proposta s’haurà de tenir en compte la definició anterior, o sigui, la qualitat de vida, l’eficiència i la sostenibilitat. Aquest treball se centrarà sobretot en l’eficiència i l’optimització.

1.1 Visita a les instal·lacions de Bonpreu Esclat

Avui en dia, el món de l’empresa i la tecnologia estan molt relacionats. Les empreses, de manera similar a les ciutats intel·ligents, utilitzen la tecnologia per millorar la qualitat de vida dels seus treballadors i augmentar l’eficiència per obtenir una millor productivitat. És per això que vaig fer una visita als magatzems robotitzats de l’empresa catalana Bonpreu, per descobrir com utilitzen diversos sistemes de control, sensors, rutes optimitzades i vehicles connectats per a gestionar el gran volum d’articles que han de controlar.

Les instal·lacions que vaig visitar varen ser els magatzems ubicats a Balenyà amb unes dimensions totals de 58.500 m^2 .



Figura 3: Imatge aèria de les instal·lacions de Bonpreu a Balenyà. Font: [5]

Les idees principals que vaig poder extreure de la visita van ser:

- Tecnologia per optimitzar la mobilitat: L'ús de sensors, GPS¹ i dades en temps real permet planificar rutes més eficients i reduir temps i emissions.
- Gestió intel·ligent de dades: Les decisions es basen en informació actualitzada, millorant la planificació del transport i la logística urbana.
- Automatització i reducció d'errors: Els processos automàtics minimitzen l'error humà i augmenten la seguretat i l'eficiència operativa.

Així doncs, amb aquest concepte definit i els nostres objectius fixats, l'eficiència i la productivitat, és hora de portar a terme la meva idea.

¹Sigles en anglès de Global Positioning System (sistema de posicionament global).

2 Idea inicial

Introducció

La meua proposta per aplicar les característiques de les ciutats intel·ligents al transport privat consisteix en desenvolupar una solució automatitzada per a la gestió dels aparcaments de vehicles privats.

Descripció del projecte

Per dur-la a terme, pretenc dissenyar, construir i programar un robot capaç de moure vehicles dins d'un aparcament i organitzar-los de manera òptima i eficient, tant pel que fa a l'aprofitament de l'espai com al temps requerit per col·locar o retirar un cotxe. Aquest robot haurà d'incorporar diversos sistemes mecànics i electrònics que li permetin assolir aquesta funció de forma òptima, precisa i fiable.

Disseny preliminar

Amb aquesta idea inicial en ment, vaig elaborar una sèrie d'esquemes preliminars per visualitzar com podria ser el prototip físic del robot i així preveure quins components mecànics necessitaria explorar. Tot i que alguns dels primers dissenys —com els que es mostren a les figures 4 i 5— no han estat finalment implementats, van resultar útils durant les fases inicials del projecte, ja que em van permetre començar a definir i donar forma a la proposta.

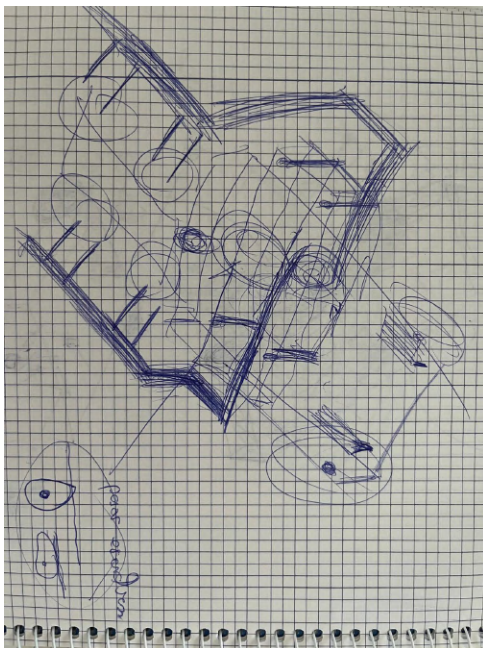


Figura 4: Primer esquema de la primera versió. Font: Elaboració pròpia



Figura 5: Segon esquema de la primera versió. Font: Elaboració pròpia

Programació i control

Paral·lelament, també vaig analitzar com hauria de ser la programació del robot per tal de garantir-ne el correcte funcionament. Això volia dir decidir quin llenguatge de programació i quins sistemes electrònics eren els més adequats per aconseguir les funcions que calien. El plantejament general del sistema es pot veure esquematitzat a la figura 6 i és explicat a continuació.

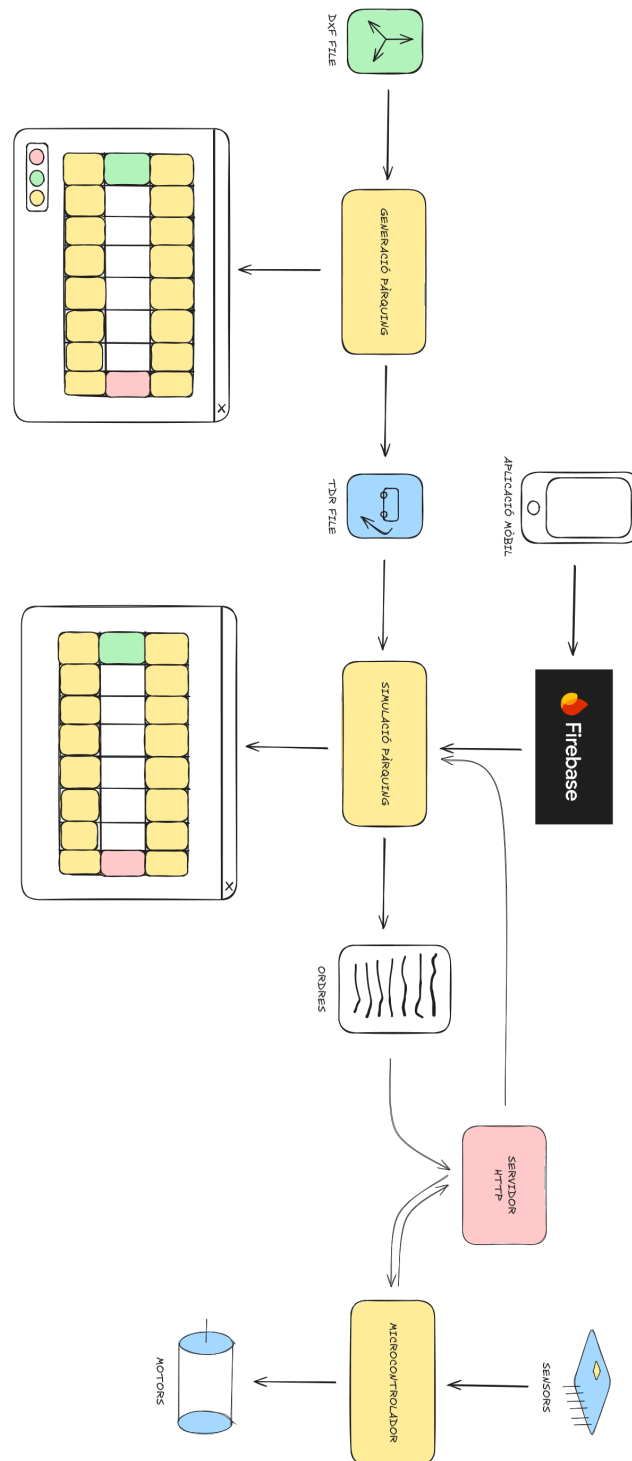


Figura 6: Esquema del sistema informàtic del projecte. Font: Elaboració pròpia

En termes de programació, es va desenvolupar una proposta inicial per al sistema de funcionament del robot. Per tal d'iniciar el procés, seria necessari disposar d'un arxiu amb les dimensions de l'aparcament i la ubicació dels diferents obstacles físics (com parets, columnes, etc.). Aquest arxiu podria tenir el format DXF², molt utilitzat en àmbits com l'enginyeria, el disseny industrial i l'arquitectura durant la fase de planificació de productes o espais. A partir d'aquest fitxer, el programa generaria automàticament un mapa de l'aparcament i proposaria la distribució òptima de les places, tenint en compte els límits físics i la maniobrabilitat del robot. Un cop definides les places, l'entrada de l'aparcament i el punt d'inici del robot, es generaria un nou fitxer amb un format personalitzat, que seria emprat pel següent programa: el simulador. Aquest segon programa, juntament amb un servidor HTTP³, s'encarregaria de gestionar les ordres que rep el robot, segons les sol·licituds dels usuaris. Aquestes peticions serien realitzades a través d'una aplicació mòbil senzilla, des de la qual l'usuari podria posar o treure el seu vehicle. L'aplicació es comunicaria amb el sistema de simulació mitjançant una base de dades allotjada al núvol, com per exemple Firestore, la qual actuaria com a intermediari entre el client i el simulador.

Recerca i desenvolupament

Per tant, en aquest treball, serà necessari dur a terme una recerca tant de components mecànics (hardware), per a la construcció física del robot, com d'eines de programació (software), per tal de desenvolupar el seu funcionament automatitzat de manera eficient i fiable. Com s'evidenciarà al llarg d'aquest treball, després de nombroses proves, errors i ajustos, he aconseguit desenvolupar un prototip funcional definitiu que compleix amb els objectius inicials plantejats. En els següents capítols, em centraré en la recerca i selecció dels components hardware i software necessaris per fer possible el desenvolupament d'aquest prototip.

²Sigles en anglès de Drawing Exchange Format (format d'intercanvi de dibuixos)

³Sigles en anglès de HyperText Transfer Protocol (protocol de transferència d'hipertext)

3 Recerca sobre maquinària (Hardware)

3.1 Actuadors

[18] Els actuadors són dispositius que converteixen senyals elèctrics o digitals en moviment físic. Els actuadors permeten al robot realitzar accions físiques com moure braços, rodes o qualsevol altra part del robot.

3.1.1 Motor amb caixa reductora

Els motors amb caixa reductora són motors molt utilitzats en projectes de robòtica educativa i sistemes automatitzats de baix cost. És un motor elèctric que funciona amb corrent continu (CC) i que porta incorporada una caixa reductora de velocitat, amb l'objectiu de reduir les revolucions per minut del motor augmentant-ne el parell motor (força). Útil en aplicacions que requereixen força, però no velocitat, com per exemple, robots, portes automàtiques o mecanismes petits.

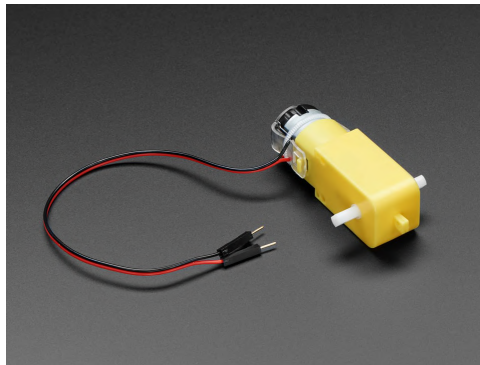


Figura 7: Imatge d'un motor amb caixa reductora. Font: [1]

Aplicació en el projecte

En aquest projecte s'utilitzaran 4 motors amb caixa reductora. Seran utilitzats pel desplaçament del robot i, per tant, formaran conjunt amb les quatre rodes del robot.

[1] Característiques tècniques

- Tensió de funcionament: de 3 V a 6 V CC
- Velocitat nominal (a 6V): aproximadament 200 revolucions per minut (RPM)

- Tipus de caixa reductora: plàstica, de relació aproximada 1:48
- Parell de sortida: apte per a aplicacions lleugeres com petits robots o vehicles
- Consum de corrent: entre 100 mA i 250 mA en funcionament normal; pot augmentar amb càrrega

[17] Components i principi de funcionament d'un motor de corrent continu

Un motor de corrent continu funciona convertint energia elèctrica en moviment rotatori. Els motors de corrent continu estan formats per dues parts: l'estator i el rotor. L'estator és la part fixa del motor i conté el camp magnètic (imants o electroimants fixos). El rotor és la part que gira i conté les bobines per on circula el corrent elèctric. Quan aquest corrent passa per les bobines genera un camp magnètic. La interacció entre els dos camps magnètics genera una força que fa girar el rotor. Per tant, el funcionament d'aquest element es basa en la llei de Lorentz que descriu com un corrent experimenta una força quan és en un camp magnètic. L'equació fonamental és:

$$\vec{F} = q \cdot (\vec{v} \times \vec{B}) \quad (1)$$

On $\vec{F}$ és la força, q és la càrrega, $\vec{v}$ és la velocitat i $\vec{B}$ és el camp magnètic.

Avantatges i limitacions

Avantatges	Limitacions
Preu molt assequible	No és apte per a aplicacions amb altes càrregues o precisió
Fàcil de controlar mitjançant ponts H ⁴ o mòduls de control de motors	Caixa reductora de plàstic (pot fallar amb l'ús continu o excessiu)
Compatible amb moltes plataformes educatives	Velocitat limitada per a aplicacions de moviment ràpid
Disponibilitat d'accessoris com encoders i rodes	

Taula 1: Taula d'avantatges i limitacions del motor amb caixa reductora.

⁴Circuit electrònic que canvia la polaritat d'una tensió aplicada a una càrrega.

Resum

El motor amb caixa reductora és una opció molt pràctica per a projectes inicials i prototips on es necessita moviment simple, fiable i econòmic. És una peça clau en l'aprenentatge de l'electrònica aplicada i el disseny de sistemes mecatrònics bàsics.

3.1.2 Motor pas a pas 28BYJ-48

[31] El motor pas a pas és un tipus de motor elèctric que es caracteritza per la seva precisió a partir de petits passos angulars. Cada pas correspon a un senyal elèctric enviat al motor que permet controlar amb gran precisió la posició, la velocitat i el sentit de gir. Per fer-ho consta d'una sèrie d'electroimants que s'activen en una seqüència específica per tal de dur a terme un moviment concret. El motor pas a pas 28BYJ-48 és un motor unipolar, o sigui que tan sols té un conjunt de bobines i requereix un controlador.

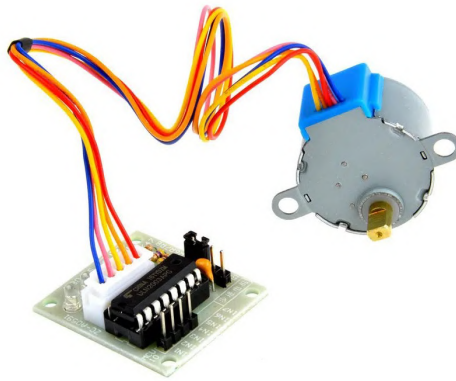


Figura 8: Imatge del motor pas a pas 28BYJ-48. Font: [31]

Aplicació en el projecte

Inicialment, en aquest projecte el motor pas a pas 28BYJ-48 es va plantejar per a dues funcions diferents, que serien realitzades per dos motors diferents. El primer seria l'encarregat d'accionar el sistema que agafaria els cotxes i el segon seria l'encarregat d'eleva tot aquest primer sistema.

[31] Característiques tècniques

- Motor pas a pas unipolar: control senzill, tan sols 4 cables
- Reductora integrada: alta precisió ($0,088^\circ$ per pas) i el parell de sortida ($3N \cdot cm$)
- Baix consum: 83 mA (model 5 V) o 32 mA (model 12 V)
- Alimentació: 5 V o 12 V (segons model)
- Cost reduït

[33] Components i principi de funcionament d'un motor pas a pas

Els motors pas a pas són formats, igual que els motors de corrent continu, de dues parts principals: el rotor i l'estator. Però en el cas d'aquest tipus de motors, el rotor té una forma de roda dentada. Quan s'aplica corrent a una bobina, es crea un camp magnètic que atrau el rotor. Les bobines s'activen una per una en una seqüència determinada fent que el rotor giri un pas cada vegada, cosa que permet girar d'una forma precisa. A part, utilitzant una reductora encara es pot augmentar més la seva precisió, ja que, per exemple, si tenim un motor que fa 8 passos per gir, però té una reductora 1:64, ara cada pas serà format per 64 passos, i per tant el seu total passarà de 8 a 512 passos.

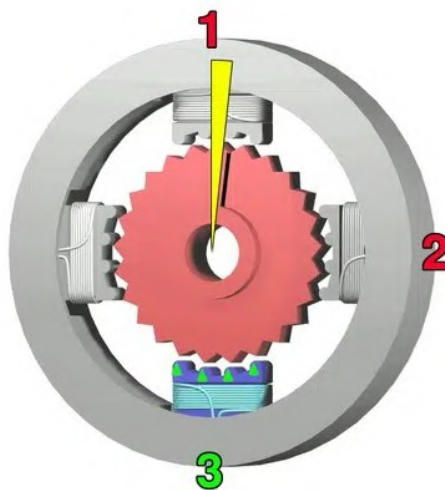


Figura 9: Imatge de l'interior d'un motor pas a pas. Font: [33]

[31] Avantatges i limitacions

Avantatges	Limitacions
Preu molt assequible	Velocitat limitada: la caixa reductora limita la velocitat del motor, cosa que pot ser un problema en aplicacions que requereixen moviments ràpids
Precisió: gràcies a la seva reductora, té una precisió de $0,087^\circ$ per pas (en mode mig pas)	Parell limitat: encara que la reductora el millori, el parell del motor segueix sent bastant dèbil en comparació amb altres models més grans
Fàcil d'utilitzar: es pot controlar fàcilment amb un microcontrolador i el controlador ULN2003	

Taula 2: Taula d'avantatges i limitacions del motor pas a pas 28BYJ-48

Resum

El motor pas a pas 28BYJ-48 és una opció molt recomanable pels petits projectes d'electrònica. El seu baix cost i facilitat a l'hora d'usar-lo fan que sigui una de les millors opcions del mercat. A més a més, encara que no tingui gaire parell, la seva precisió és més que suficient per a la majoria de projectes.

3.1.3 Motor servo MG90S

[34] El motor servo, com el motor pas a pas, és un tipus de motor elèctric que es caracteritza per la seva precisió. Les grans diferències amb els motors pas a pas és que utilitzen un bucle de control amb sensors (feedback) que ajuda a corregir errors i que ofereixen un moviment continu gràcies a l'electrònica de control. Tot això fa que siguin un directe substitut als motors pas a pas.



Figura 10: Imatge del motor servo MG90S. Font: [12]

Aplicació en el projecte

En aquest projecte el servomotor MG90S acaba essent el substitut del motor pas a pas 28BYJ-48, ja que aquest últim no acabarà tenint la força necessària per al projecte.

[12] Característiques tècniques

- Motor servo: control senzill, tan sols 1 cable PWM⁵
- Velocitat: 0,1 s/60°
- Parell de sortida: 1,8 kg/cm o Parell màxim: 2,2 kg/cm
- Alimentació: 4,8 V - 6 V (Normalment 5 V)
- Cost reduït

[15][16] Components i principi de funcionament d'un motor servo

Els motors servo tenen tres components diferents: un motor de corrent continu, engranatges i un sistema de retroalimentació (feedback). Bàsicament, el servo rep un senyal de control que indica la posició desitjada, seguidament es compara la posició actual amb la desitjada, si hi ha una diferència s'envia un senyal al motor perquè ajusti la seva posició. El sistema de retroalimentació proporciona constantment informació per assegurar que el motor aconsegueixi la seva posició desitjada. Els engranatges s'utilitzen per reduir la velocitat del motor a una velocitat més senzilla de controlar i amb un parell major. En la

⁵Sigles en anglès de Pulse-Width Modulation (modulació per amplada de polsos)

següent imatge es poden observar els diversos components d'un motor servo: el sistema de retroalimentació (Potentiometer i Control Unit), els engranatges (Gear Train) i el motor de corrent continu (DC Motor).

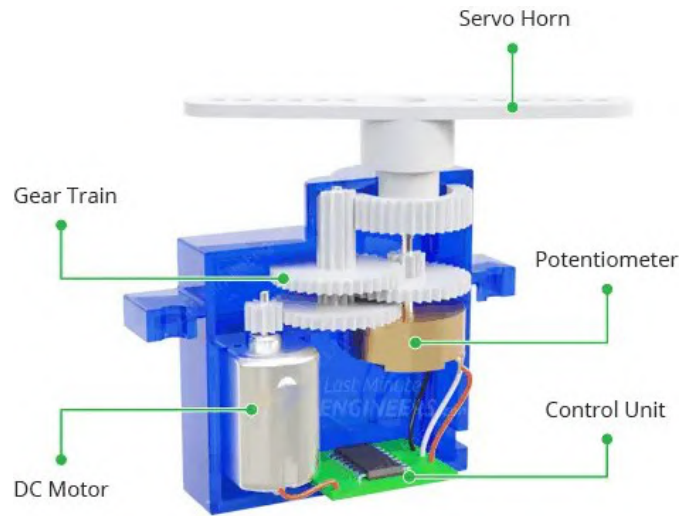


Figura 11: Imatge de l'interior d'un motor servo. Font: [30]

[12] Avantatges i limitacions

Avantatges	Limitacions
Compacte i lleuger: ideal per a petits projectes i aplicacions	Rotació limitada (0° - 180°): No apte per rotació contínua
Engranatges metàl·lics: més resistent que els servos de plàstic	Potència limitada: no ideal per càrregues pesades
Bon parell per la seva mida (fins $2,2 \text{ kg} \cdot \text{cm}$)	Sense retroalimentació (feedback) externa: menys precisió comparat amb els servos professionals

Taula 3: Taula d'avantatges i limitacions del motor servo MG90S

Resum

El servo MG90S és una opció molt recomanable per a projectes petits de robòtica i vehicles controlats a distància. El seu baix pes i mida compacta, juntament amb els engranatges metàl·lics i el parell motor suficient per a moltes aplicacions de robòtica educativa, el fan molt pràctic. A més, la seva facilitat de control el converteix en una de les millors opcions del mercat per a petits projectes.

3.2 Sensors

[18] Els sensors són dispositius que detecten i mesuren canvis en l'entorn o en el mateix robot i converteixen aquesta informació en un senyal que pot ser processat per un controlador.

3.2.1 Mòdul de comunicacions BU01

[3] El BU01 és un mòdul UWB o Ultra-Wideband (banda ultraampla), bàsicament, un mòdul de comunicacions basat en el xip DW1000 de DecaWave. Malgrat que pot ser utilitzat per a comunicacions, la seva utilitat real en aquest projecte és de localització i, per tant, és considerat un sensor.

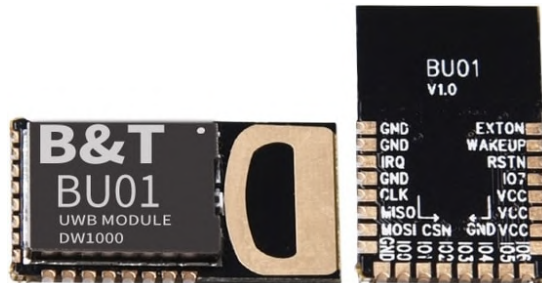


Figura 12: Imatge del mòdul BU01. Font: [3]

Aplicació en el projecte

Tal com he mencionat anteriorment, aquest mòdul busca ser utilitzat com a sistema de localització per tal de millorar la precisió de la totalitat del sistema. Finalment, aquest mòdul no s'utilitzarà en el projecte sinó que tan sols es proposa en les versions inicials.

[3] Característiques tècniques

- Fàcil d'integrar sense necessitat de disseny de radiofreqüència complex
- Pot gestionar molts dispositius a l'entorn sense problemes
- Compatible amb l'estàndard UWB, assegurant interoperabilitat
- Permet localització precisa amb mètodes de mesura de temps

- Ofereix diferents velocitats de transmissió segons les necessitats

[3] Principi de funcionament del mòdul BU01

Per tal de determinar la seva posició respecte d'un altre mòdul, el BU01 utilitza el que és conegut com a ToF⁶ (Temps de vol), és a dir, determina el temps que ha trigat el senyal transmès fins que l'altre mòdul la rep. Per fer-ho, el temps de vol es determina mitjançant la següent equació:

$$ToF = \frac{T_{loop} - T_{reply}}{2} \quad (2)$$

On ToF és el temps de vol, T_{loop} és el temps que ha passat entre que el mòdul ha enviat el senyal fins que ha rebut resposta i T_{reply} és una constant: el temps que el segon mòdul ha tardat entre rebre el senyal i enviar-ne una.

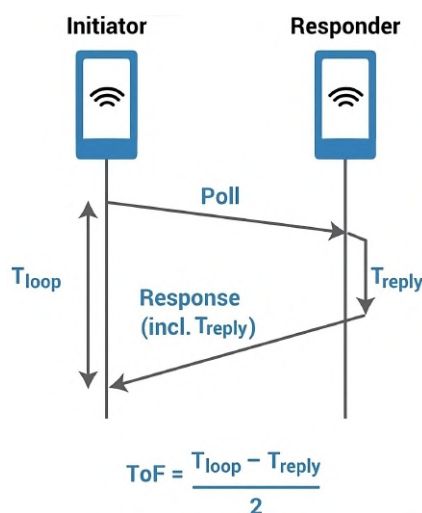


Figura 13: Esquema de funcionament d'un mòdul BU01. Font: [3]

Resum

El mòdul UWB BU01 és una opció molt recomanable per a projectes amb ESP32⁷ que necessiten localització precisa i mesura de distàncies. El seu baix consum, facilitat d'integració i suport per a mesurar distàncies en temps real el fan molt pràctic.

⁶Sigles en anglès de Time of Flight, és a dir, la mesura del temps que es triga a recórrer una distància determinada en un medi donat.

⁷Vegeu capítol 3.3

3.2.2 Sensor de distancia VL53L0X

[36] El VL53L0X és un sensor de distància d'alta precisió, de format petit i amb un cost baix, i per tant un sensor molt utilitzat en projectes de robòtica educativa i sistemes automatitzats de baix cost.

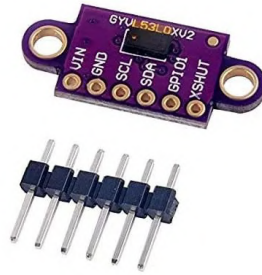


Figura 14: Imatge del VL53L0X. Font: [36]

Aplicació en el projecte

Aquest sensor vol ser utilitzat per tal de saber la distància entre el robot i les rodes del cotxe que hagi de transportar per tal d'elevat el cotxe pel punt correcte.

[35] Característiques tècniques

- Abast de mesurament: 50 mm a 1.200 mm (mode estàndard) i fins a 2.000 mm (mode ampliat).
- Tecnologia ToF: Funciona basant-se en el temps de vol d'un feix làser.
- Tensió: Opera amb un rang de 2,6 V a 5 V, compatible amb molts microcontroladors.
- Alta precisió: Desviació mínima de l'1% en distàncies fins a 1 metre.

[36] Principi de funcionament del VL53L0X

Igual que el mòdul BU01 3.2.1, el VL53L0X utilitza el ToF per determinar la distància fins a l'obstacle més pròxim. La gran diferència és que no hi ha un altre mòdul esperant un senyal, sinó que aquest fa rebotar un feix de llum a un obstacle per saber-ne la posició. L'equació que determina el temps de vol en aquest cas és la següent:

$$T_{oF} = \frac{T_{flight}}{2} \quad (3)$$

On ToF és el temps de vol i T_{flight} és el temps que ha passat entre que s'ha emès el feix de llum i s'ha rebut el rebot.

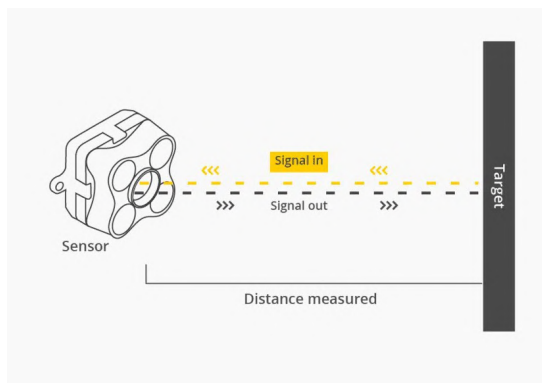


Figura 15: Esquema de funcionament del VL53L0X. Font: [36]

Resum

El VL53L0X és un sensor de distàncies compacte, barat i d'alta precisió basat en la tecnologia ToF amb làser infraroig. Pot mesurar distàncies entre 5 cm i 200 cm, amb una precisió molt superior als sensors ultrasònics o d'infraroigs (IR).

3.2.3 Acceleròmetre i giroscopi GY-512

[32] El GY-512 és un mòdul molt utilitzat en projectes que impliquen moviment i orientació, ja que integra un acceleròmetre i un giroscopi en el mateix dispositiu. Les dades que ens aporta són l'acceleració dels tres eixos X, Y i Z i també la velocitat angular de cada un.

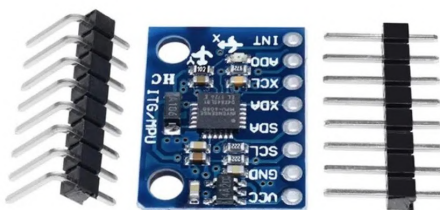


Figura 16: Imatge del GY-512. Font: [32]

Aplicació en el projecte

En aquest cas, el mòdul GY-512 es vol usar per a conèixer els angles respecte a la seva posició inicial i poder corregir errors de moviment en cas que n'hi hagi.

[32] Característiques tècniques

- Voltatge d'operació: pot funcionar amb 3.3V o 5V gràcies al regulador integrat.
- Connexió: comunicació per I²C⁸, fàcil d'integrar amb Arduino⁹ i altres plataformes.
- Acceleròmetre: rang ajustable de $\pm 2g$ a $\pm 16g$, permet escollir precisió segons el projecte.
- Giroscopi: rang ajustable de ± 250 a ± 2000 °/s, adaptable a diferents necessitats.

[25][38][10] Principi de funcionament d'un acceleròmetre MEMS

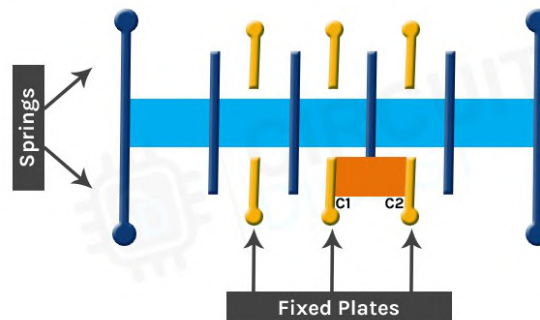


Figura 17: Esquema de funcionament de l'acceleròmetre GY-512. Font: [25]

La funció principal d'un acceleròmetre és mesurar el moviment lineal, sigui un moviment, un xoc o vibració, això sí, sense una referència fixa. Per fer-ho dins del xip hi ha una massa diminuta (la part de color blau clar de la figura 17) que té a cada banda unes plaques fixes conductores que formen condensadors variables (les parts blau fosc de la figura 17, excepte en els extrems) i tota aquesta part (la massa i les plaques) es pot moure gràcies a les molles dels extrems. Quan el dispositiu accelera, la massa es desplaça i, per tant, les plaques també, això provoca que un dels condensadors (per exemple, C1 de la figura 17) augmenti la seva capacítància¹⁰ i que l'altre condensador (en aquest cas C2 de la figura

⁸I²C és un bus de comunicacions en sèrie.

⁹Plataforma de hardware lliure basada en microcontroladors per crear projectes electrònics interactius.

¹⁰Magnitud física que defineix la facultat d'un cos per emmagatzemar càrrega elèctrica.

17) la redueixi. La diferència de capacitància és proporcional al desplaçament de la massa, i per tant a l'acceleració com podem comprovar amb les següents equacions:

$$C = \frac{\epsilon A}{d} \quad (4)$$

On C és la capacitància, ϵ és la permitivitat de l'aïllant utilitzat, A és l'àrea de cada placa i d és la separació entre les plaques. Per tant, la capacitància i la separació de les plaques són inversament proporcionals, si la separació és major llavors la capacitància és menor, en canvi, si la separació és menor llavors la capacitància és major. Per altra banda, la 2a llei de Newton i la llei de Hooke exposen que:

$$F = m \cdot a \quad (5)$$

On F és la força, m és la massa i a és l'acceleració.

$$F = k \cdot x \quad (6)$$

On F és la força, k és la constant elàstica i x és el desplaçament. Si igualem les dues expressions:

$$m \cdot a = k \cdot x \quad (7)$$

d'on:

$$x = \frac{m}{k} \cdot a \quad (8)$$

Quan la massa es desplaça, la separació efectiva passa a ser $d \pm x$, de manera que podem expressar la capacitància amb la següent equació:

$$C(x) = \epsilon \cdot \frac{A}{d \pm x} = \epsilon A \cdot \frac{1}{d \pm x} \quad (9)$$

El problema és que aquesta relació no és lineal en x . Això complica els càlculs si volem expressar directament ΔC en funció d' a . En el cas que el desplaçament sigui molt petit respecte a la separació ($x \ll d$), podem fer una expansió de primer ordre:

$$\frac{1}{d \pm x} \approx \frac{1}{d} \mp \frac{x}{d^2} \quad (10)$$

Substituïm a $C(x)$:

$$C(x) = \epsilon A \cdot \frac{1}{d \pm x} \approx \epsilon A \left(\frac{1}{d} \mp \frac{x}{d^2} \right) = \frac{\epsilon A}{d} \left(1 \mp \frac{x}{d} \right) \quad (11)$$

$$\Delta C \propto x \propto a \quad (12)$$

Gràcies a l'equació (8), que relaciona el desplaçament amb l'acceleració, i a l'equació (11), que relaciona el desplaçament amb la capacítància, podem concloure que la variació de capacítància ΔC és proporcional al desplaçament x i, per tant, a l'acceleració a .

[25][38][7] Principi de funcionament d'un giroscopi MEMS

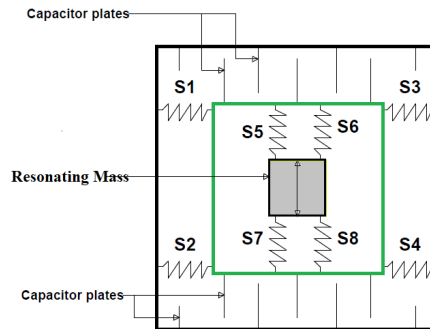


Figura 18: Esquema de funcionament del giroscopi GY-512. Font: [38]

A diferència d'un acceleròmetre, que detecta moviment lineal, el giroscopi detecta rotacions. És a dir, mesura velocitats angulars, com gira de ràpid un objecte al voltant d'un eix. Per fer-ho, igual que l'acceleròmetre, el giroscopi té una massa mòbil suspesa dins del xip amb molles, aquesta massa té unes plaques (indicades com a "Capacitor plates" a la figura 18) que formen condensadors variables i imitant a l'acceleròmetre, detectant els canvis de capacítància provocats pel moviment de la massa es poden mesurar les velocitats angulars. Per poder determinar aquestes velocitats la massa està dissenyada per

girar quan apareix una força de Coriolis. El principi físic de la força de Coriolis defineix que quan una massa té una velocitat de vibració i una velocitat angular, apareix una força:

$$\vec{F}_C = -2m \cdot (\vec{\omega} \times \vec{v}) \quad (13)$$

On $\vec{F}_C$ és la força de Coriolis, m és la massa, $\vec{\omega}$ és la velocitat angular i $\vec{v}$ és la velocitat de vibració. Al mateix temps, la massa està subjecta a molles, així que qualsevol força aplicada la desplaça segons la llei de Hooke exposada ja a l'equació (6). Si igualement la força de Coriolis amb la força de la molla queda que:

$$k \cdot x = -2m(\vec{\omega} \times \vec{v}) \quad (14)$$

d'on:

$$x = -\frac{2m}{k}(\vec{\omega} \times \vec{v}) \quad (15)$$

Igual que a l'acceleròmetre i tal com s'ha demostrat a l'equació (10), la capacitància és proporcional al desplaçament, i aquest alhora és proporcional a la velocitat angular:

$$\Delta C \propto x \propto \omega \quad (16)$$

En definitiva, amb l'ajuda de l'equació (15), que relaciona el desplaçament amb la velocitat angular, i de l'equació (11), que relaciona el desplaçament amb la capacitància, podem concloure que la variació de capacitància ΔC és proporcional al desplaçament x i, a causa d'això, a la velocitat angular ω .

Resum

El GY-521 és un mòdul molt utilitzat per mesurar moviment i orientació, ja que integra un acceleròmetre i a la vegada, un giroscopi de tres eixos. Funciona tant a 3,3 V com a 5 V i es comunica fàcilment mitjançant I²C. Gràcies als seus rangs ajustables i alta precisió, és ideal per a projectes amb Arduino, robòtica i drons.

3.3 Microcontrolador - ESP32

L'ESP32 és un microcontrolador molt popular desenvolupat per Espressif Systems. Destaca per la seva connectivitat integrada (Wi-Fi i Bluetooth), el baix consum energètic i la seva gran potència de càlcul respecte a altres plataformes més convencionals com Arduino. Gràcies a aquestes característiques, és àmpliament utilitzat en projectes d'IoT o Internet of Things (Internet de les coses), robòtica i aplicacions que requereixen comunicació sense fils.

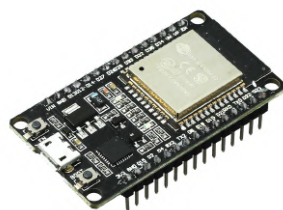


Figura 19: Imatge d'una placa ESP32 DevKit v1 DOIT

Un dels grans avantatges de l'ESP32 és la seva flexibilitat: existeixen nombroses versions i plaques de desenvolupament, amb diferent nombre de pins, grandàries i prestacions. Això permet escollir el model que millor s'adapti a les necessitats de cada projecte.

[39] Principals opcions d'ESP32

- **ESP32-WROOM-32:** És el mòdul més habitual, amb Wi-Fi i Bluetooth integrats i fins a 34 GPIOs¹¹ disponibles. Es troba en múltiples formats de plaques, amb 30, 36 o fins i tot més pins.
- **ESP32-WROVER:** Versió millorada amb més memòria RAM¹² (4 MB SRAM¹³ addicional) i sovint amb memòria PSRAM¹⁴. S'utilitza en projectes que requereixen càlcul intensiu o gestió de dades més complexes, com càmeres o aplicacions multimèdia.
- **Placa ESP32 DevKit (30 o 36 pins):** Les més habituals en entorns educatius i de prototipatge. Inclouen el mòdul ESP32-WROOM-32 muntat sobre una placa amb regulador de voltatge i connexió USB, la qual cosa en facilita molt la programació i ús amb Arduino IDE.

¹¹Sigles en anglès de General Purpose Input/Output (entrada/sortida per a un ús general)

¹²Sigles en anglès de Random Access Memory (memòria d'accès aleatori)

¹³Sigles en anglès de Static Random Access Memory (memòria d'accès aleatori estàtica)

¹⁴Sigles en anglès de Pseudo Static Random Access Memory (memòria d'accès aleatori pseudo estàtica)

Elecció: ESP32 DevKit v1 DOIT

En el marc d'aquest projecte s'ha escollit l'ESP32 DevKit v1 DOIT de 30 pins perquè:

- Ofereix prou GPIOs (30) per connectar tots els sensors i actuadors necessaris.
- La seva compatibilitat amb Arduino IDE¹⁵ permet un desenvolupament ràpid i accessible.
- Té gran disponibilitat al mercat i una comunitat activa que facilita trobar solucions i exemples.
- La mida compacta i la facilitat de muntatge en una placa de proves la fan ideal per a prototipatge i proves.

Resum

El DevKit v1 de DOIT és una opció robusta, fiable i econòmica per desenvolupar projectes que requereixen connectivitat sense fils i capacitat de càlcul, alhora que garanteix facilitat d'ús i suport tècnic gràcies a la seva popularitat.

3.4 Rodes Mecanum

[37] Per tal que el robot sigui capaç de moure's, necessita rodes. En aquest projecte s'utilitzen un tipus de rodes anomenades Mecanum. Aquestes rodes permeten moure el vehicle en qualsevol direcció sense necessitat de girar el xassís.



Figura 20: Imatge d'una roda Mecanum. Font: [37]

¹⁵Sigles en anglès de Integrated Development Environment (entorn integrat de desenvolupament)

[37][11] Principi de funcionament d'una roda Mecanum

Aquest tipus de rodes permet realitzar una sèrie de moviments complexos, tal com es mostra a la figura 21. El seu disseny és particular: la roda està formada per una sèrie de rodets disposats amb un angle de 45° respecte al pla de la roda, la qual cosa permet generar una força motriu en aquesta direcció, com es visualitza a la figura 22.

Per aconseguir que les combinacions de les components d'aquesta força motriu permetin efectuar determinats moviments, tal com s'observa a la figura 21, s'utilitzen dues variants de la roda Mecanum: una en què els rodets estan orientats a 45° respecte a l'eix d'abscisses o eix x i una altra en què els rodets estan orientats a 135° .

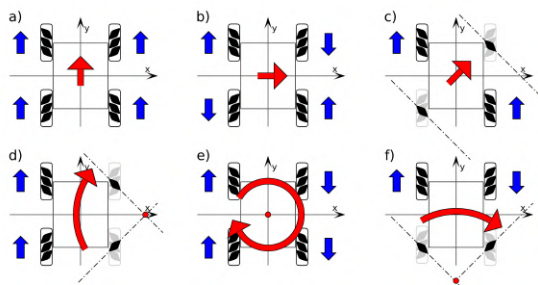


Figura 21: Possibles moviments gràcies a les rodes Mecanum. Font: [13]



Figura 22: Força motriu d'una roda Mecanum. Font: [11]

Gràcies al fet distintiu de les rodes, la força motriu en una direcció de 45 graus, i a les dues variacions d'aquesta, combinant les components de la força motriu podem aconseguir les diverses direccions. Ara farem un breu estudi del cas a) i b) de la figura 21:

Cas a):

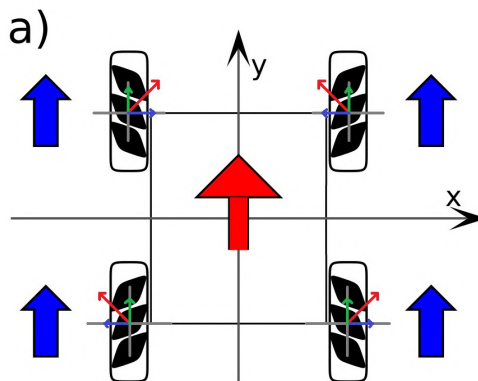


Figura 23: Esquema de forces de les rodes en el cas A. Font: Elaboració pròpia

Tal com ja observem gràcies a la figura 23, si resolguéssim la suma de les components de les forces de forma gràfica el resultat seria que no hi ha força resultant en l'eix d'abscisses o eix x, en canvi, en l'eix d'ordenades o eix y hi ha una força resultant significativa. De forma analítica, considerant que cada roda realitza una força d'1 Newton en una direcció de 45 o 135 graus:

$$\begin{aligned} F^{FR} &= 1N, & F^{FL} &= 1N \\ F^{RR} &= 1N, & F^{RL} &= 1N \end{aligned} \quad (17)$$

$$\begin{aligned} F_x^{FR} &= 1N \cdot \cos 135^\circ, & F_y^{FR} &= 1N \cdot \sin 135^\circ \\ F_x^{FL} &= 1N \cdot \cos 45^\circ, & F_y^{FL} &= 1N \cdot \sin 45^\circ \\ F_x^{RR} &= 1N \cdot \cos 45^\circ, & F_y^{RR} &= 1N \cdot \sin 45^\circ \\ F_x^{RL} &= 1N \cdot \cos 135^\circ, & F_y^{RL} &= 1N \cdot \sin 135^\circ \end{aligned} \quad (18)$$

$$\begin{aligned} F_x &= F_x^{FR} + F_x^{FL} + F_x^{RR} + F_x^{RL} \\ F_y &= F_y^{FR} + F_y^{FL} + F_y^{RR} + F_y^{RL} \end{aligned} \quad (19)$$

$$\begin{aligned} F_x &= 1 \cdot \cos 135^\circ + 1 \cdot \cos 45^\circ + 1 \cdot \cos 45^\circ + 1 \cdot \cos 135^\circ \\ &= -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} \\ &= 0 \\ F_y &= 1 \cdot \sin 135^\circ + 1 \cdot \sin 45^\circ + 1 \cdot \sin 45^\circ + 1 \cdot \sin 135^\circ \\ &= \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \\ &= 2\sqrt{2} \approx 2.828 \text{ N} \end{aligned} \quad (20)$$

Tal com podem veure doncs la força resultant en l'eix d'abscisses o eix x (F_x) és 0, per tant, no hi ha desplaçament en aquest eix, en canvi, en l'eix d'ordenades o eix y hi ha una força resultant (F_y) que no és 0, per tant, hi ha desplaçament en el cas d'aquest eix. Cal mencionar que els noms de les forces d'aquest breu estudi són definits per les posicions de les rodes en angles: Front right o FR (Davant dreta), Front left o FL (Davant esquerra), Rear right o RR (Darrera dreta) i Rear left o RL (Darrera esquerra).

Cas b):

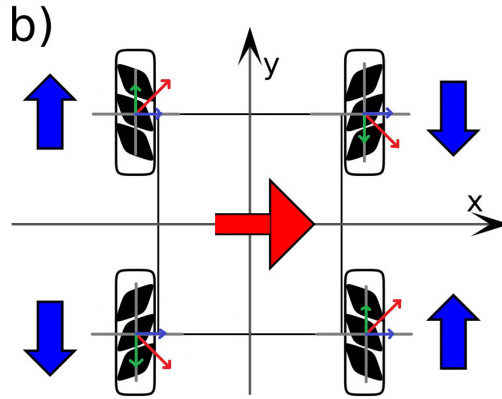


Figura 24: Esquema de forces de les rodes en el cas B. Font: Elaboració pròpia

Igual que amb l'estudi del cas A, gràcies a l'esquema ja podem veure que la força resultant en l'eix d'ordenades o eix y serà 0 i no hi haurà moviment, en canvi, en l'eix d'abscisses o eix x hi haurà una força resultant i, per tant, moviment. Malgrat que ho podem resoldre de forma gràfica també es pot fer de forma analítica:

$$\begin{aligned} F^{FR} &= 1N, & F^{FL} &= 1N \\ F^{RR} &= 1N, & F^{RL} &= 1N \end{aligned} \quad (21)$$

$$\begin{aligned} F_x^{FR} &= 1N \cdot \cos 315^\circ, & F_y^{FR} &= 1N \cdot \sin 315^\circ \\ F_x^{FL} &= 1N \cdot \cos 45^\circ, & F_y^{FL} &= 1N \cdot \sin 45^\circ \\ F_x^{RR} &= 1N \cdot \cos 45^\circ, & F_y^{RR} &= 1N \cdot \sin 45^\circ \\ F_x^{RL} &= 1N \cdot \cos 315^\circ, & F_y^{RL} &= 1N \cdot \sin 315^\circ \end{aligned} \quad (22)$$

$$\begin{aligned} F_x &= F_x^{FR} + F_x^{FL} + F_x^{RR} + F_x^{RL} \\ F_y &= F_y^{FR} + F_y^{FL} + F_y^{RR} + F_y^{RL} \end{aligned} \quad (23)$$

$$\begin{aligned}
 F_x &= 1 \cdot \cos 315^\circ + 1 \cdot \cos 45^\circ + 1 \cdot \cos 45^\circ + 1 \cdot \cos 315^\circ \\
 &= \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} \\
 &= 2\sqrt{2} \approx 2.828 \text{ N} \\
 F_y &= 1 \cdot \sin 315^\circ + 1 \cdot \sin 45^\circ + 1 \cdot \sin 45^\circ + 1 \cdot \sin 315^\circ \\
 &= -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} \\
 &= 0
 \end{aligned} \tag{24}$$

Tal com podem veure doncs la força resultant en l'eix d'abscisses o eix x (F_x) no és 0, per tant, hi ha desplaçament en aquest eix, en canvi, en l'eix d'ordenades o eix y, la força resultant (F_y) és 0, per tant, no hi ha desplaçament en el cas d'aquest eix. Cal mencionar que s'han mantingut els mateixos noms per les forces que en l'estudi pel cas A.

[22] Model cinemàtic d'un robot amb quatre rodes Mecanum

Tal com hem vist amb el principi de funcionament d'aquestes rodes, es tracta d'unes rodes que ofereixen moltes utilitats pel que fa al moviment del robot. Un dels problemes que tenim és que podem realitzar aquests moviments, però no calcular la seva velocitat lineal, per tant, no podrem saber on és ben bé el robot. Per resoldre aquest problema, he fet recerca bibliografia, i gràcies al model cinemàtic plantejat a l'article [22], podem determinar que el model cinemàtic pel nostre robot és el següent:

$$\begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} & -\frac{1}{l_x+l_y} & \frac{1}{l_x+l_y} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{25}$$

$$v_x(t) = \frac{r}{4} (\omega_1 + \omega_2 + \omega_3 + \omega_4) \tag{26}$$

$$v_y(t) = \frac{r}{4} (-\omega_1 + \omega_2 + \omega_3 - \omega_4) \tag{27}$$

Les equacions 26 i 27 permeten determinar la velocitat lineal del robot per tal de controlar-lo durant la gestió d'un aparcament.

4 Disseny del prototip físic del robot

Un cop determinats els components necessaris per a construir el robot, cal definir els materials i el programari que es faran servir per dissenyar tant el robot com la placa de circuit imprès (en anglès Printed Circuit Board, PCB). Aquestes han estat les opcions escollides:

4.1 Impressió 3D

En aquest projecte utilitzarem la impressió 3D per a fer el xassís del robot i també altres peces necessàries. Aquesta tècnica ofereix tres característiques clau: ens permet dissenyar components amb les mesures desitjades, ens ofereix un temps de producció molt reduït respecte als processos tradicionals i tot això amb un cost reduït.



Figura 25: Imatge d'una impressió feta amb PLA (Àcid Polilàctic). Font: [29]

La impressió 3D ofereix diversos materials de construcció, els més comuns són PLA (Àcid Polilàctic), PETG (Polietilè Tereftalat Glicolat) i l'ABS (Acrilonitril Butadiè Estirè). Les principals característiques de cada material són:

- PLA: fàcil d'imprimir i barat, però fràgil i deformable amb la calor.
- ABS: resistent i amb millor tolerància tèrmica, però difícil d'imprimir (es deforma fàcilment)
- PETG: un punt entremig, relativament fàcil d'imprimir i més resistent que el PLA

Finalment, es va decidir utilitzar PLA, ja que és fàcil d'imprimir, econòmic i ofereix una rigidesa suficient per a un prototip funcional. Per tal d'imprimir aquest material hem d'utilitzar una impressora 3D, en aquest cas utilitzarem una Bambu Lab A1.

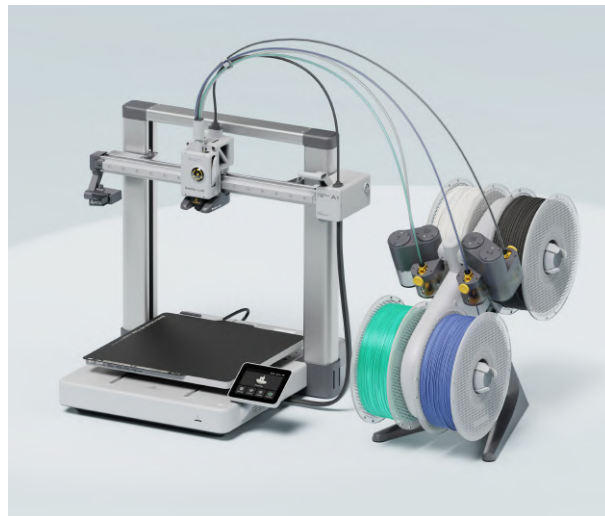


Figura 26: Imatge d'una impressora Bambu Lab A1. Font: [28]

Aquesta impressora és una de les impressores 3D més populars de tot el mercat i, per tant, compta amb una gran comunitat d'usuaris i suport documental d'on podré obtenir ajuda si és necessària. Amb el seu baix preu, ofereix unes característiques interessants, com el volum d'impressió de $256*256*256\text{ mm}^3$ o també el seu calibratge total automàtic, per tot això va ser la màquina escollida pel projecte.

4.2 Disseny 3D

Per al disseny de les peces del robot utilitzarem Fusion 360, un programari CAD¹⁶ desenvolupat per Autodesk. Aquest programari permet modelar components tridimensionals amb gran precisió, així com simular el seu comportament mecànic abans de la fabricació.



Figura 27: Logotip del programari Fusion360. Font: [20]

¹⁶Sigles en anglès de Computer-Aided Design (disseny assistit per ordinador).

Fusion 360 ofereix eines per a:

- **Assemblatges:** Possibilitat de combinar diverses peces per comprovar encaixos, col·lisions i rangs de moviment del robot abans de la seva construcció.
- **Simulació mecànica:** Inclou anàlisis de tensió, deformació i moviment, cosa que ajuda a identificar punts febles de les peces dissenyades.
- **Exportació per impressió 3D:** Els models es poden exportar directament en format `.stl` o `.obj` per a ser processats amb un slicer i impresos en 3D.
- **Documentació tècnica:** Permet generar plànols tècnics i llistes de materials, útils per al control del projecte i la fabricació.

L'ús de Fusion 360 permet una modificació fàcil i ràpida en el disseny: si alguna peça no s'ajusta correctament o necessita millores, es pot modificar digitalment i tornar a exportar per a impressió, reduint el temps i cost de prototipat.

4.3 Disseny 3D del prototip

En aquest apartat tractarem les diverses versions del prototip 3D i analitzarem la seva evolució, determinant les raons per als diversos canvis.

4.3.1 Versió 1

Aquesta primera versió es va basar en els primers esquemes 4 i 5.

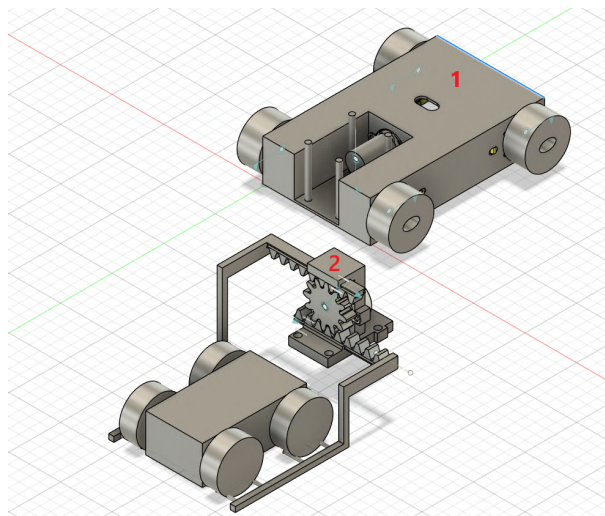


Figura 28: Imatge de la versió 1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

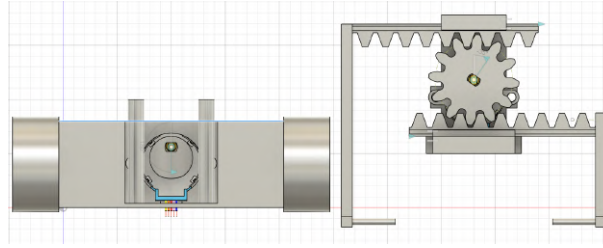


Figura 29: Imatge frontal de la versió 1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

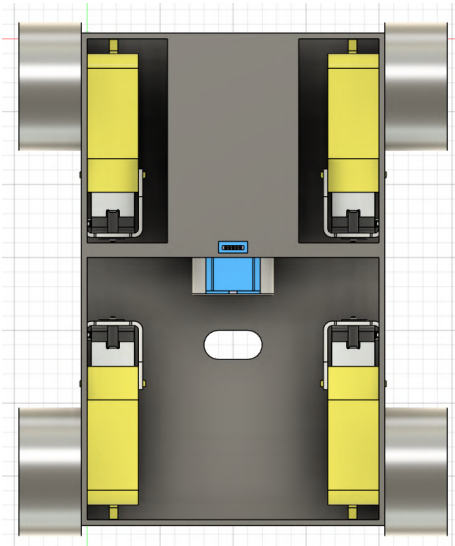


Figura 30: Imatge inferior de la versió 1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

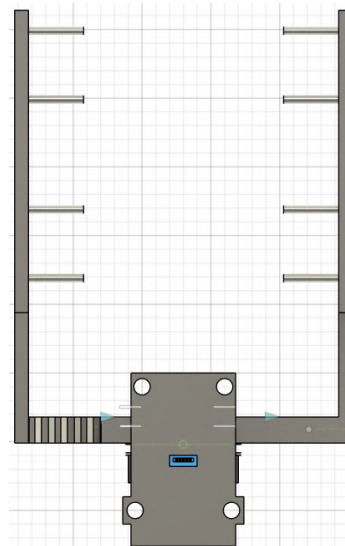


Figura 31: Imatge inferior de la versió 1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

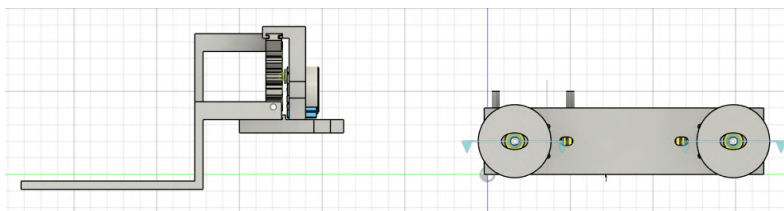


Figura 32: Imatge lateral de la versió 1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

Aquesta primera versió, com es pot veure a la figura 28, constava de dues parts principals: una primera part (número 1 a la figura 28) encarregada de la mobilitat del robot i d'elevat la segona part, i una segona part (número 2 a la figura 28) encarregada d'agafar el cotxe per les rodes. Aquesta segona part, el “gadget”, agafaria el cotxe gràcies a dos braços formats per dues cremalleres que roscarien amb un engranatge mogut per un motor pas a pas 28BYJ-48. Aquests braços són elevats per la primera part gràcies a una excèntrica (un disc amb l'eix descentrat). Aquest prototip hauria pogut ser funcional, però va ser descartat degut a la seva grandària i pel seu disseny poc refinat.

4.3.2 Versió 2

Aquesta segona versió es va basar en el següent esquema 33.

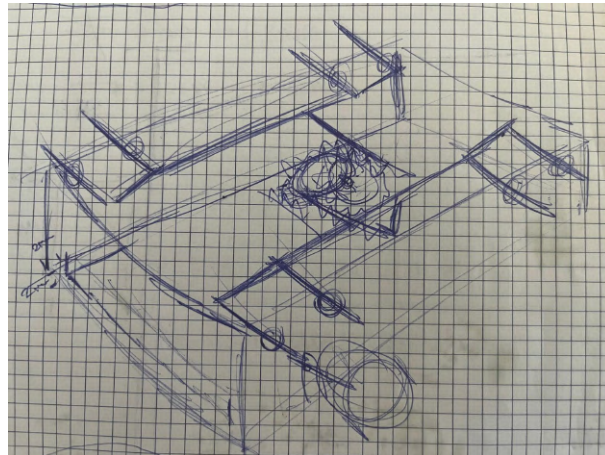


Figura 33: Esquema de la segona versió. Font: Elaboració pròpia

A partir d'aquest esquema va sorgir el següent disseny:

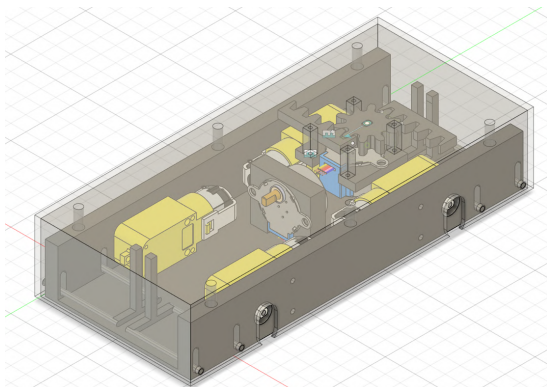


Figura 34: Imatge de la versió 2 del prototip en el programari Fusion 360. Font: Elaboració pròpia



Figura 35: Imatge superior de la versió 2 del prototip en el programari Fusion 360. Font: Elaboració pròpia

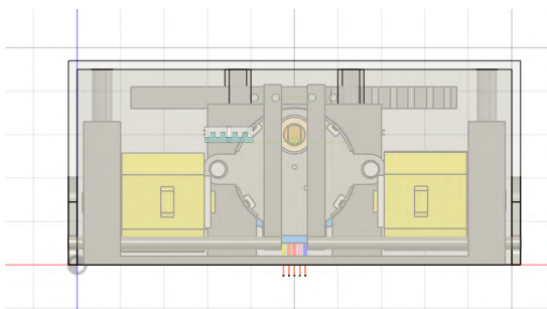


Figura 36: Imatge frontal de la versió 2 del prototip en el programari Fusion 360. Font: Elaboració pròpia

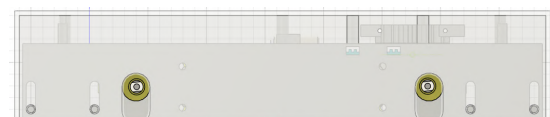


Figura 37: Imatge lateral de la versió 2 del prototip en el programari Fusion 360. Font: Elaboració pròpia

Amb aquest disseny es va crear un prototip més refinat, compacte i de menor dimensió. No va poder ser el prototip definitiu perquè van aparèixer nous problemes, tant mecànics com d'espai. El problema mecànic més important va ser que amb una sola cremallera i un sol motor pas a pas no era suficient per fer moure les quatre peces encarregades d'agafar les rodes del cotxe.

4.3.3 Versió 2.1

En aquest nou disseny s'han tingut en compte els problemes de l'anterior mantenint la mateixa idea general:

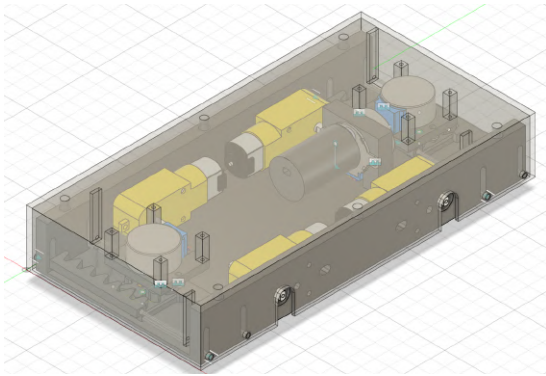


Figura 38: Imatge de la versió 2.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

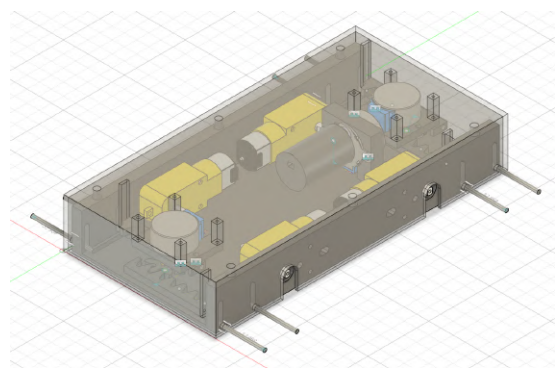


Figura 39: Imatge de la versió 2.1 del prototip en el programari Fusion 360 amb el sistema desplegat. Font: Elaboració pròpia

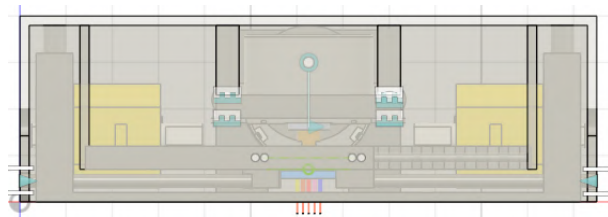


Figura 40: Imatge frontal de la versió 2.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia



Figura 41: Imatge lateral de la versió 2.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

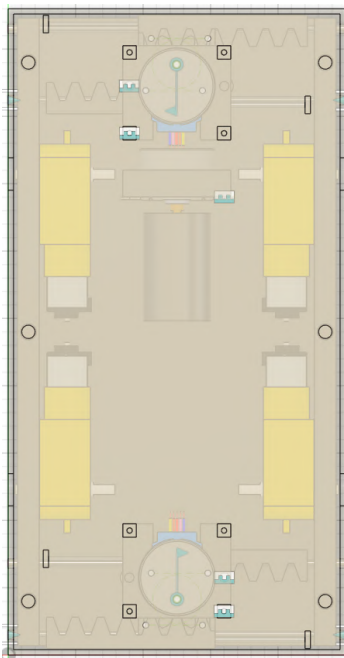


Figura 42: Imatge superior amb tots els motors de la versió 2.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

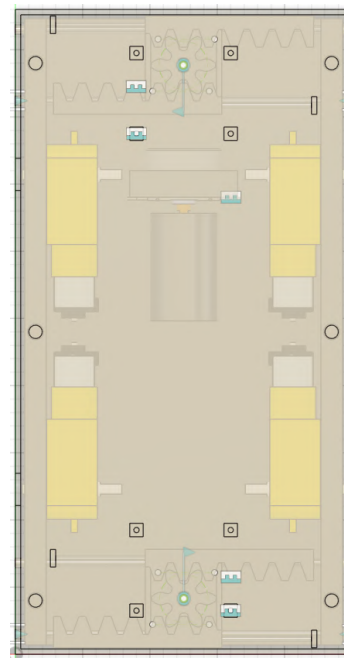


Figura 43: Imatge superior de la versió 2.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

La principal modificació que ha fet falta per millorar el model de la versió 2 ha estat incorporar un segon mecanisme independent de cremallera - pinyó alimentat per un motor pas a pas. Malgrat que aquest disseny hauria pogut ser el definitiu perquè era funcional, es va descartar per la seva complexitat mecànica que queda reflectida a les figures 38 - 43, on es pot veure la necessitat d'utilitzar molts components (4 motors amb caixa reductora, 3 motors pas a pas, 2 sistemes cremallera - pinyó i 1 excèntrica).

4.3.4 Versió 3

Aquesta versió s'aproxima a ser la definitiva. Amb aquesta versió vaig buscar la senzillesa mecànica conservant la funcionalitat. Per complir aquest objectiu vaig adonar-me que la forma més senzilla per elevar un cotxe és per sota, aprofitant els seus punts de suport i no per les rodes. Amb aquest canvi de concepte es crea aquest disseny:

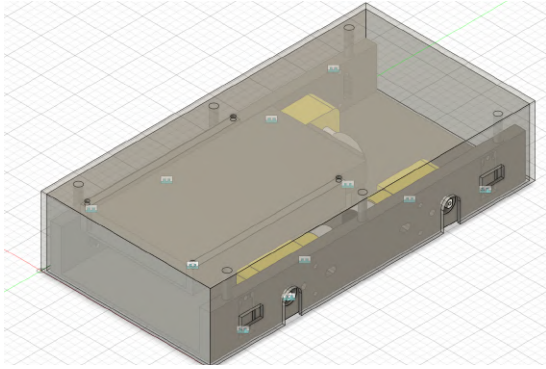


Figura 44: Imatge de la versió 3 del prototip en el programari Fusion 360 amb la powerbank. Font: Elaboració pròpia

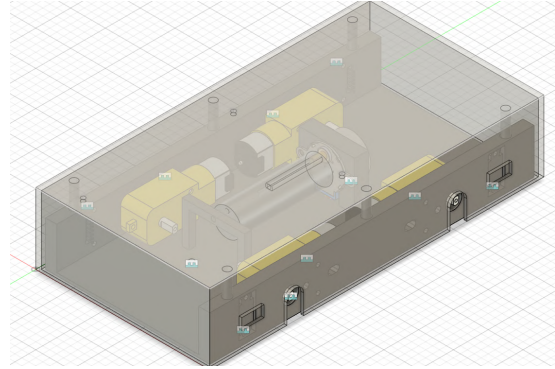


Figura 45: Imatge de la versió 3 del prototip en el programari Fusion 360 sense la powerbank. Font: Elaboració pròpia

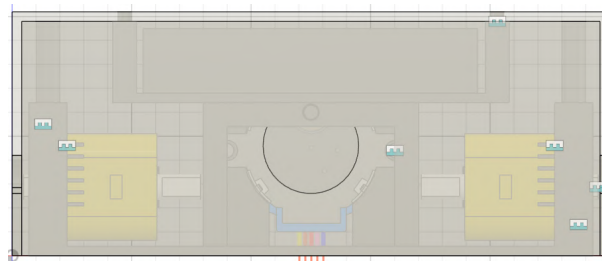


Figura 46: Imatge frontal de la versió 3 del prototip en el programari Fusion 360. Font: Elaboració pròpia

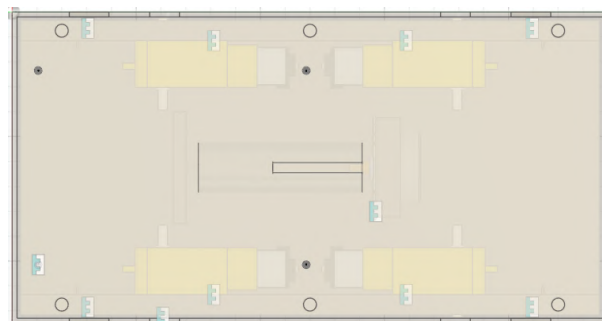


Figura 47: Imatge superior de la versió 3 del prototip en el programari Fusion 360. Font: Elaboració pròpia



Figura 48: Imatge lateral de la versió 3 del prototip en el programari Fusion 360. Font: Elaboració pròpia

En aquest model ja no és necessari el sistema cremallera - pinyó i això em permet guanyar espai i simplicitat. En aquesta versió, el cotxe s'eleva mitjançant la carcassa del robot: el robot es col·loca sota el cotxe, una excèntrica aixeca la seva carcassa, i aquesta, al seu torn,

eleva el cotxe. Pel seu funcionament continuem necessitant 4 motors amb caixa reductora, però podem reduir a un sol motor pas a pas i mantenir l'excèntrica. En finalitzar el disseny d'aquesta versió es va detectar un greu error i és que el motor pas a pas 28BYJ-48 no tindria prou força, i per tant es va plantejar la millora d'aquesta versió utilitzant dos servomotors MG90S.

4.3.5 Versió 3.1

Com ja he introduït anteriorment, per millorar la versió del prototip, en aquest model es canvia el motor pas a pas per dos servomotors MG90S per aconseguir la força necessària per elevar la carcassa del robot.

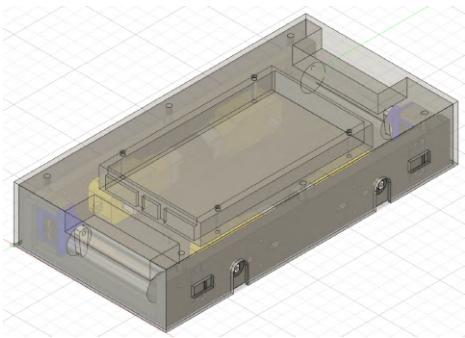


Figura 49: Imatge de la versió 3.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

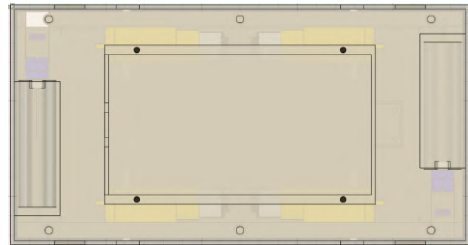


Figura 50: Imatge superior de la versió 3.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

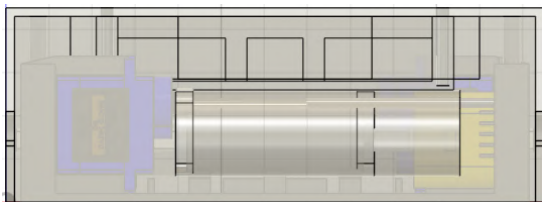


Figura 51: Imatge frontal de la versió 3.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia



Figura 52: Imatge lateral de la versió 2.1 del prototip en el programari Fusion 360. Font: Elaboració pròpia

Aquesta última versió 3.1 ha estat escollida com a la definitiva gràcies a la seva senzillesa mecànica i a la vegada, la seva funcionalitat. Aquest prototip utilitza quatre motors amb caixa reductora i incorpora dos servomotors MG90S que accionaran dues excèntriques independents. A més a més, inclou un mòdul GY-512 (acceleròmetre i giroscopi) i dos sensors VL53L0X per mesurar distàncies. Per alimentar tot el sistema té un espai reservat per a una bateria externa (powerbank).

4.4 Disseny 3D de les rodes

Les rodes mecanum, escollides per aconseguir el moviment del prototip, tenen un disseny molt particular i difícil de realitzar. Per aquest motiu es va decidir cercar un model ja dissenyat i fer-ne les modificacions necessàries

Per a buscar el disseny es va utilitzar una plataforma de models 3D, Thingiverse. Després d'una recerca i l'anàlisi de diverses opcions, es va acabar escollint el model *Mecanum Wheel*[6] de Dmitry Borisov. Els arxius d'aquest model estaven en format .IPT¹⁷, fet que complicava la utilització del disseny amb el programari Fusion 360. Per tal de solucionar aquest problema es va utilitzar l'eina 3D-Tool[26] per convertir el format .IPT a un format compatible amb el programari Fusion 360.

Un cop importat el disseny escollit, van fer falta algunes modificacions, com ajustar les dimensions de les diferents peces de les rodes i adaptar la forma de l'arbre, per tal que encaixessin amb el motor del prototip. Aquest és el disseny final de les rodes:

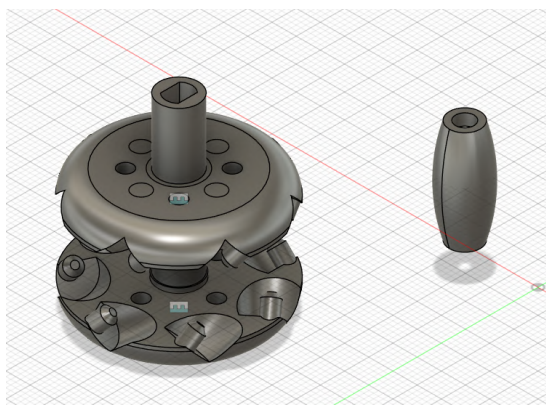


Figura 53: Imatge del disseny d'una roda Mecanum en el programari Fusion 360. Font: Elaboració pròpia

4.5 PCB

Un cop acabat el disseny de la carcassa i dels mecanismes del robot, calia començar a dissenyar la part electrònica. A causa de la certa complexitat de l'electrònica i per a una millor gestió es va escollir dissenyar una PCB. Una PCB és la base física sobre la qual es munten i connecten els diversos components electrònics d'un dispositiu. Aquestes ens ajuden a substituir els antics sistemes de cablejat manuals oferint-nos una solució compacta i fiable.

¹⁷Sigles en anglès de Autodesk Inventor Part (part d'Autodesk Inventor)

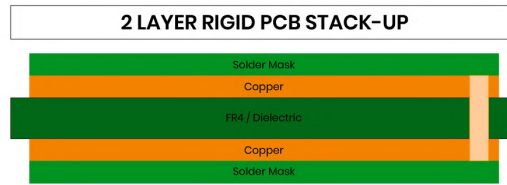


Figura 54: Estructura d'una placa de circuit imprès de 2 capes. Font: [9]

[9] Una placa de circuit imprès (PCB) està formada per una o diverses capes de coure adherides sobre un substrat aïllant (normalment material FR-4, una fibra de vidre epoxídica) i les pistes de coure fan de conductors que uneixen els diferents components. Segons la seva complexitat poden ser monocapa, de doble capa o multicapa, en el nostre cas utilitzarem una placa de circuit imprès de dues capes, ja que ens ofereix simplicitat i per un preu reduït.

4.6 Disseny PCB

Per a dissenyar aquest tipus de component s'ha d'utilitzar programari especialitzat, en aquest cas s'utilitzarà EasyEDA Pro, un programa basat en web que facilita el disseny d'esquemàtics i l'exportació automàtica dels fitxers per a fabricació. El que em va fer decantar per aquest programari és que ofereix l'accés als dissenys via navegador sense necessitat d'instal·lació, també compta amb una extensa biblioteca de components amb la possibilitat d'afegir components personalitzats i compta també amb una gran comunitat d'usuaris i un munt de documentació.



Figura 55: Logotip del programari EasyEDA. Font: [14]

4.7 Prototips PCB

En aquest apartat tractarem les diverses versions del prototip PCB i analitzarem la seva evolució, determinant les raons per als diversos canvis.

4.7.1 Versió 1

Aquest primer disseny de la placa de circuit imprès es va realitzar per ser utilitzada en la versió 1 del prototip 3D del robot, exposat a l'apartat 4.3.1. En aquest disseny es van utilitzar els següents components:

- 1* ESP32 de 30 pins: Microcontrolador encarregat de la computació.
- 2* SN74HC595: Registre de desplaçament que permet controlar múltiples sortides digitals utilitzant poques línies de control.
- 2* ULN2003A: Controlador de transistors que permet commutar càrregues de potència com motors.
- 2* L293D: Pont en H doble que permet controlar el sentit i la velocitat de dos motors de correct continu.
- 1* BU01: Mòdul de comunicació d'ultra banda ampla (UWB).
- 1* L7805CV: Regulador lineal de tensió que subministra una sortida estable de 5V.
- Diversos components electrònics com resistències, condensadors o connectors.

Durant el disseny d'aquesta placa es van utilitzar diverses fitxes de dades (datasheets) incloses en l'annex C. El resultat final és exposat per les següents figures 56, 57 i l'esquemàtic inclòs a l'annex B.

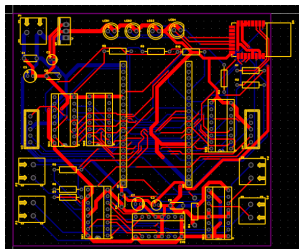


Figura 56: Imatge del disseny de la PCB versió 1 en el programari EasyEDA Pro. Font: Elaboració pròpia

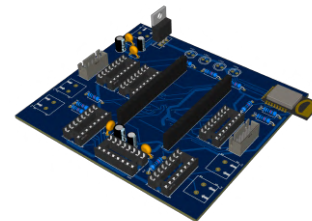


Figura 57: Imatge 3D del disseny de la PCB versió 1 en el programari EasyEDA Pro. Font: Elaboració pròpia

Aquest disseny es va acabar descartant perquè el prototip 3D pel qual havia estat pensat va ser descartat. A partir d'aquest punt no es varen tornar a dissenyar més PCB fins al disseny final del robot.

4.7.2 Versió 2

Aquesta segona versió de la placa de circuit imprès es va començar a dissenyar quan es va determinar el prototip 3D final, per tant, quan es va acabar el disseny de la versió 3.1, explicada a l'apartat 4.3.5. En aquest segon disseny es van utilitzar els següents components:

- 1* ESP32 de 30 pins: Microcontrolador encarregat de la computació.
- 2* SN74HC595: Registre de desplaçament que permet controlar múltiples sortides digitals utilitzant poques línies de control.
- 2* L293D: Pont en H doble que permet controlar el sentit i la velocitat de dos motors de correct continu.
- 4* VL53L0X: Sensor de làser de temps de vol que mesura distàncies amb alta precisió.
- 1* GY512: Mòdul que incorpora un acceleròmetre i un giroscopi.
- Diversos components electrònics com resistències, condensadors o connectors.

Durant el disseny d'aquesta placa es van utilitzar diverses fitxes de dades (datasheets) incloses en l'annex C. El resultat final és exposat per les següents figures 58, 59 i l'esquemàtic inclòs a l'annex B.

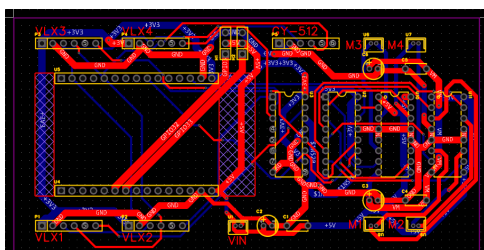


Figura 58: Imatge del disseny de la PCB versió 2 en el programari EasyEDA Pro. Font: Elaboració pròpia

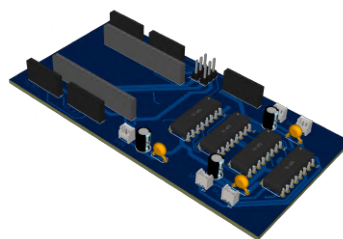


Figura 59: Imatge 3D del disseny de la PCB versió 2 en el programari EasyEDA Pro. Font: Elaboració pròpia

Aquesta versió va ser escollida com a definitiva per la seva senzillesa, mesures i funcionalitat. La placa és capaç de gestionar tots els components inclosos en el disseny 3.1 del prototip del robot i, per tant, perfecte en tots els sentits.

5 Recerca sobre programari (Software)

5.1 Proposta de tecnologies

Com ja es varen mencionar a l'apartat 2 i podem observar a la figura 6 per a portar a terme per complet aquest projecte es volen utilitzar diverses tecnologies, aquestes són:

- Arxius DXF
- Python
- Firestore
- HTTP
- C++

Cada una d'aquestes tecnologies tindrà una funció específica en el projecte explicada en els següents apartats 5.2, 5.3 i 5.4.

5.2 Arxius DXF

[2] Els arxius DXF són arxius vectorials. Aquests van ser llançats al mercat l'any 1982 per a ajudar els equips de disseny a compartir dibuixos en les diverses aplicacions CAD. Aquest nou format va ser un èxit gràcies al fet que es tracta d'un format de codi obert, independent dels diversos programes CAD i amb una alta precisió. Com que aquest format d'arxiu està implementat en els sectors de l'enginyeria, el disseny i l'arquitectura es va escollir el format DXF per a poder importar les mesures i diversos obstacles de l'aparcament. Per a tractar les dades en termes de programació, utilitzarem una llibreria de Python anomenada ezdxf, exposada a l'apartat 5.3.3.

5.3 Python

Python és un dels llenguatges més populars i usats actualment a escala mundial. Això és degut al fet que és un dels llenguatges més simples d'utilitzar gràcies a les seves intuïtives ordres en anglès. A causa d'aquesta simplicitat i en ser un dels més usats a escala mundial en podem trobar molta documentació, per això ha estat escollit com el llenguatge de programació principal del projecte.



Figura 60: Logotip de Python. Font: [19]

5.3.1 Python bàsic

Com s'ha comentat Python és conegut per la seva gran llegibilitat en el codi, és per això que com podem veure les seves ordres més bàsiques són fàcilment comprensibles:

- `print()` — Mostra un missatge o el valor d'una variable per pantalla.
- `input()` — Llegeix una entrada de l'usuari i la retorna com a text.
- `type()` — Indica el tipus de dada d'una variable (per exemple: `int`, `str`, `float`).
- `len()` — Dona la longitud (nombre d'elements) d'una cadena, llista o altre objecte iterable.
- `int()`, `float()`, `str()` — Converteixen valors a enter, nombre decimal o text respectivament.
- `for ... in ... :` — Bucle que recorre elements d'una seqüència (com una llista o un rang).
- `if ... :` — Condició que permet executar codi només si es compleix una expressió lògica.
- `else:`, `elif ... :` — Alternatives dins d'una estructura condicional.
- `while ... :` — Bucle que s'executa mentre es compleixi una condició.
- `def nom_funcio(...):` — Defineix una funció que agrupa instruccions reutilitzables.
- `import ...` — Permet utilitzar mòduls externs de Python (per exemple, `import math`).

A banda de totes aquestes ordres bàsiques n'hi ha moltes més, i en el cas que es ne-

cessiti una ordre molt específica es poden utilitzar llibreries com les que es descriuen a continuació.

5.3.2 pygame

pygame és una llibreria de Python que com descriu el seu nom en anglès s'utilitza per a crear videojocs, en el cas d'aquest projecte s'utilitza per a crear la interfície gràfica dels diversos programes que s'han d'utilitzar com per exemple, la generació del mapa de l'aparcament o la simulació de l'aparcament. La característica principal de Pygame és que proporciona totes les eines bàsiques necessàries per a construir un joc: gestió d'esdeveniments (com el teclat o el ratolí), dibuix d'objectes a la pantalla, control del temps d'execució i càrrega de recursos multimèdia (imatges i sons). Algunes ordres bàsiques d'aquesta llibreria són:

- `import pygame` — Importa la llibreria de Pygame.
- `pygame.init()` — Inicialitza tots els mòduls necessaris de Pygame.
- `pygame.display.set_mode((ample, alt))` — Crea una finestra de joc amb les dimensions donades.
- `pygame.display.set_caption("Títol")` — Assigna un títol a la finestra del joc.
- `pygame.event.get()` — Obté tots els esdeveniments (tecles, clics de ratolí, etc.).
- `pygame.key.get_pressed()` — Comprova quines tecles estan premudes en un instant donat.
- `screen.fill((R,G,B))` — Omple la pantalla amb un color donat (RGB¹⁸).
- `pygame.draw.rect(screen, color, (x,y,ample,alt))` — Dibuixa un rectangle.
- `pygame.draw.circle(screen, color, (x,y), radi)` — Dibuixa un cercle.
- `pygame.display.update()` — Actualitza el contingut de la finestra.
- `clock = pygame.time.Clock()` — Crea un rellotge per controlar els fotogrames per segon.

¹⁸Sigles en anglès de Red, Green, Blue (vermell, verd, blau)

- `clock.tick(fps)` — Controla la velocitat d'execució del bucle principal del joc.
- `pygame.quit()` — Tanca la finestra i surt correctament de Pygame.

Tots els jocs i programes creats amb `pygame` segueixen una mateixa estructura bàsica anomenada *gameloop* o bucle de joc. Aquest bucle és l'encarregat de mantenir el programa actiu i es divideix en tres fases:

1. **Gestió d'esdeveniments:** es comproven les accions de l'usuari.
2. **Actualització de lògica:** es calcula l'estat del programa.
3. **Dibuix:** es neteja la pantalla i es tornen a dibuixar tots els elements visuals.

5.3.3 ezdx

`ezdx` és una llibreria de Python que ens ajuda a extreure informació dels fitxers amb format DXF, en el meu cas extreure les dades sobre les mesures i obstacles de l'aparcament que hi ha en aquest fitxer. La llibreria és especialment útil, ja que si no es disposés d'aquesta eina, s'hauria d'estudiar tota l'estructura del format DXF per saber com extreure'n les dades. Amb aquesta llibreria tot això ho podem fer amb unes simples ordres. Les ordres més bàsiques són:

- `import ezdx` — Importa la llibreria de `ezdx`.
- `ezdx.readfile(ruta_fitxer)` — Llegeix el fitxer DXF indicat per la ruta del sistema.
- `fitxer.modelspace()` — Retorna l'espai vectorial del fitxer amb tots els seus elements.
- `mspOriginalMap.query(TIPUS_ELEMENT)` — Retorna una llista amb tots els elements d'un tipus de l'espai vectorial.
- `doc.saveas("nom.dxf")` — Guarda el fitxer DXF amb el nom indicat.

5.3.4 Google Cloud Firestore

Google Cloud Firestore és una base de dades gestionada per Google, pensada per aplicacions mòbils, web i servidors. Ens permet emmagatzemar dades en documents organitzats en col·leccions, amb sincronització en temps real, seguretat integrada i de forma gratuïta. La llibreria oficial de Python per accedir a Firestore és `google-cloud-firestore` i, en aquest cas, sempre necessitarem un arxiu amb les credencials per a accedir a les dades. Les ordres que utilitzarem en el projecte són:

- `from google.cloud import firestore` — Importa el client per connectar amb Firestore.
- `from google.oauth2 import service_account` — Permet gestionar credencials de servei.
- `firestore.Client(credentials=CREDS)` — Ens permet crear un client per accedir a les dades.
- `db.collection(collection)` — Extreu una col·lecció concreta de la base de dades.
- `collection.document(document)` — Extreu un document concret d'una col·lecció.
- `doc_ref.get()` — Extreu les dades d'un document.
- `doc_ref.update(data)` — Actualitza les dades d'un document.
- `doc_ref.set(data)` — Estableix les dades d'un document.

Els passos a seguir per realitzar una connexió exitosa amb la base de dades són:

1. **Importar les llibreries:** s'importen les diverses ordres a Python.
2. **Definició de credencials:** es defineixen les credencials de la base de dades.
3. **Creació d'un client:** es crea un client per fer les diverses sol·licituds.

En el meu projecte aquesta eina s'ha utilitzat per transmetre la informació de les peticions d'aparcament dels usuaris des dels seus telèfons mòbils fins al programa de simulació d'aparcament.

5.3.5 requests

La llibreria **requests** és una de les més utilitzades en Python per realitzar peticions HTTP de forma senzilla i eficient. Permet enviar sol·licituds a servidors web i rebre respostes, facilitant la interacció amb les interfícies de programació d'aplicacions (en anglès Application Programming Interface, API) i pàgines web. Les ordres més bàsiques són:

- `import requests` — Importa la llibreria de requests.
- `requests.get(url)` — Realitza una petició GET a la URL especificada.
- `requests.post(url, data={})` — Realitza una petició POST amb dades.
- `response.status_code` — Retorna el codi de resposta HTTP.
- `response.text` — Retorna el contingut de la resposta en format text.
- `response.json()` — Retorna el contingut de la resposta en format JSON¹⁹.

Aquesta llibreria serà utilitzada per a realitzar les connexions entre el simulador de l'aparcament i el servidor HTTP i ens permet actualitzar les instruccions en temps real que rep el robot.

5.3.6 flask

flask és un microframework i no una llibreria de Python per al desenvolupament d'aplicacions web. La diferència entre una llibreria i un microframework és que les llibreries són un conjunt de funcions i classes, en canvi, un microframework és una estructura lleugera que ja té una base i unes directrius per desenvolupar una aplicació. S'ha escollit **flask** perquè és lleuger, flexible i fàcil d'utilitzar, ideal per projectes petits i mitjans, així com per prototips ràpids. A part, proporciona les eines bàsiques per crear aplicacions web, com ara el maneig de rutes, peticions HTTP i plantilles HTML²⁰, i es pot ampliar amb extensions. Algunes ordres i conceptes bàsics amb **flask** són:

- `app = Flask(__name__)` — Crea una nova aplicació Flask.
- `@app.route("/ruta")` — Decorador per definir una ruta URL²¹.

¹⁹Sigles en anglès de JavaScript Object Notation (notació d'objectes de JavaScript)

²⁰Sigles en anglès de HyperText Markup Language (llenguatge de marques d'hipertext)

²¹Sigles en anglès de Uniform Resource Locator (localitzador uniforme de recursos)

- `def funcio():` — Funció que s'executa quan es visita la ruta.
- `return "text"` o `return render_template("fitxer.html")` — Retorna resposta HTML o text.
- `app.run(debug=True)` — Executa l'aplicació en mode debug.

En aquest treball, `flask` busca ser utilitzat com a servidor HTTP per a la comunicació entre la simulació i el robot.

5.3.7 Llibreries built-in

Python inclou una sèrie de llibreries integrades, conegudes com a **builtin modules**, que permeten realitzar moltes tasques sense necessitat d'instal·lar llibreries addicionals. Aquestes llibreries cobreixen funcions com manipulació de fitxers, gestió de dades, operacions matemàtiques, manipulació de text, etc. Les utilitzades en aquest projecte han estat:

- `ast` — Permet analitzar i manipular el codi Python com un arbre de sintaxi abstracta.
- `os` — Permet interactuar amb el sistema operatiu (gestió de fitxers, directoris, variables d'entorn, etc.).
- `threading` — Permet crear i gestionar fils (*threads*) per a execució concurrent.
- `time` — Funcions relacionades amb el temps i pauses en l'execució.
- `uuid` — Generació d'identificadors únics universals.
- `copy` — Permet crear còpies superficials (*shallow*) o profundes (*deep*) d'objectes.
- `random` — Generació de valors aleatoris i selecció aleatòria d'elements.
- `heapq` — Implementació d'una estructura de dades de tipus heap (munt de dades).
- `tkinter` — Permet crear interfícies gràfiques d'usuari (en anglès Graphical User Interface, GUI).

5.4 ESP32

Per programar l'ESP32 s'ha utilitzat el framework d'Arduino, que permet escriure codi en C++²² amb una estructura senzilla i llibreries predefinides. Aquest framework proporciona una gran varietat de funcions i llibreries per gestionar sensors, comunicacions, actuadors i altres components de manera eficient. La programació amb Arduino facilita el desenvolupament ràpid de projectes IoT i permet una gran compatibilitat amb components electrònics. Les llibreries utilitzades en el desenvolupament del programari del robot han estat:

- "Adafruit_VL53L0X.h" — Llibreria per gestionar el sensor de distància VL53L0X basat en tecnologia ToF (Time of Flight). Permet mesurar distàncies amb alta precisió.
- <MPU6050_light.h> — Llibreria simplificada per utilitzar el sensor de moviment MPU6050, que combina un giroscopi i un acceleròmetre.
- <Wire.h> — Llibreria per utilitzar el protocol I²C, utilitzat per comunicar-se amb sensors i dispositius compatibles.
- <ArduinoJson.h> — Llibreria per treballar amb dades en format JSON, útil per comunicar-se amb servidors o dispositius IoT.
- <WiFi.h> — Llibreria per gestionar connexions WiFi amb l'ESP32, incloent connexió, desconnexió i obtenció d'estat de xarxa.
- <HTTPClient.h> — Llibreria per realitzar peticions HTTP (GET, POST, etc.), essencial per la comunicació amb APIs i serveis web.
- <ESP32Servo.h> — Llibreria per controlar servomotors amb precisió utilitzant l'ESP32.

5.5 Algorismes

Durant el desenvolupament s'han utilitzat dos algorismes diferents per a dues tasques:

1. **Buscar el camí més ràpid:** algorisme de cerca A* (a-estrella o a-star)
2. **Generar l'aparcament òptim:** algorisme propi fent ús de l'algorisme de cerca A*

²²Llenguatge de programació que fou creat el 1983 per Bjarne Stroustrup.

5.5.1 Algorisme de cerca A*

[27] L'algorisme A* és un algoritme de cerca heurística utilitzat per trobar el camí més curt entre un punt inicial i un punt final dins d'un espai de cerca. És àmpliament utilitzat en robòtica, videojocs, sistemes de navegació i d'intel·ligència artificial. A* combina els avantatges funcionals de l'algorisme Dijkstra (el seu predecessor) amb una funció heurística que permet accelerar la cerca. L'heurística és una funció que ordena les diferents opcions que té un algorisme de cerca en cada pas a partir d'informació extra (en aquest cas, l'estimació del cost restant) amb la finalitat de trobar més ràpidament l'objectiu final. En el cas d'aquest projecte, s'utilitzarà l'algorisme per dotar al robot de les capacitats necessàries per determinar el camí més ràpid fins a l'objectiu (plaça de l'aparcament).

Funcionament:

Per aconseguir-ho l'algorisme combina dues dades: el cost real del recorregut des de l'origen fins a un punt concret i una estimació del cost restant fins la destinació. En el nostre cas, l'algorisme considera l'aparcament com una graella on el fet de recórrer una cel·la té el cost d'1 i l'estimació del cost restant és definida per la suma de la distància restant fins al punt concret on el vol arribar tant en l'eix d'abscisses o eix x com en l'eix d'ordenades o eix y. L'algorisme funciona utilitzant aquestes dades en bucle fins a trobar la solució òptima, a cada pas del bucle, l'algorisme selecciona la cel·la amb el menor valor de la funció de cost total:

$$f(n) = g(n) + h(n) \quad (28)$$

on $g(n)$ és el cost acumulat fins a la cel·la n i $h(n)$ és l'estimació del cost restant. El procés continua avançant i expandint-se per les cel·les amb el menor cost total fins que assoleix la destinació final o no queden camins possibles.

Exemple d'ús de l'algorisme

Per a exemplificar aquest algorisme, calcularem el camí més ràpid, per tant, el més curt, des de la cel·la verda fins a la cel·la vermella de la figura 61. A la figura el negre significa obstacle o paret, el verd significa inici, el vermell és la destinació.

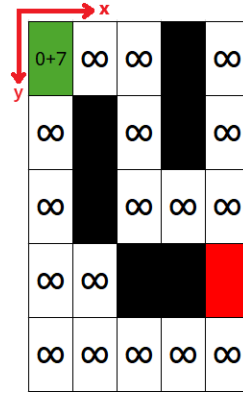


Figura 61: Graella a descobrir el camí. Pas 1. Font: Elaboració pròpia

Per començar amb l'anàlisi hem de deixar clar que durant l'exemple es considera el punt (0,0) l'extrem superior esquerre i el punt (4,4) l'extrem inferior dret. A banda, també hem de tenir en compte que la suma dels nombres de cada cel·la és el valor de la funció de cost total de cada cel·la; el primer nombre fa referència a $g(n)$ i el segon nombre a $h(n)$. Per tant, com podem veure, la primera cel·la seria 0+7, ja que ens hem desplaçat 0 cel·les i si sumem la distància restant en l'eix y de 3, i la distància restant en l'eix x de 4, tenim una estimació del cost restant de 7.

Quan executéssim l'algorisme el primer que faria aquest és començar a l'inici i definir-ne el cost, a les altres cel·les les hi assignaria un cost d'infinit. Seguidament, comprovaria els seus veïns i en determinaria que pels dos veïns de l'inici el valor és 1+6, una cel·la de desplaçament i 6 fins a la destinació.

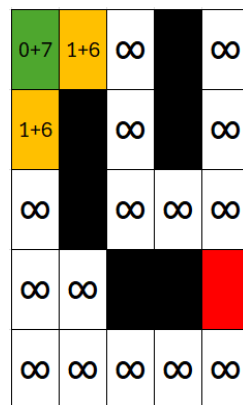


Figura 62: Graella a descobrir el camí. Pas 2. Font: Elaboració pròpia

Durant els següents tres passos realitzaria el mateix, sempre analitzant els veïns de les cel·les amb la funció de cost total menor, com es representa en les següents figures 63, 64 i 65.

0+7	1+6	2+5		∞
1+6		∞		∞
2+5		∞	∞	∞
∞	∞			
∞	∞	∞	∞	∞

Figura 63: Graella a descobrir el camí. Pas 3. Font: Elaboració pròpia

0+7	1+6	2+5		∞
1+6		3+4		∞
2+5		∞	∞	∞
3+4	∞			
∞	∞	∞	∞	∞

Figura 64: Graella a descobrir el camí. Pas 4. Font: Elaboració pròpia

0+7	1+6	2+5		∞
1+6		3+4		∞
2+5		4+3	∞	∞
3+4	4+3			
4+5	∞	∞	∞	∞

Figura 65: Graella a descobrir el camí. Pas 5. Font: Elaboració pròpia

En aquest punt, com podem veure ja no totes les cel·les tindrien el mateix valor, sinó que tenim dues cel·les amb valor 7 i una amb valor 9, per tant, l'algorisme descartaria la cel·la amb el valor 9 i continuaria cercant el camí més ràpid amb les altres dues cel·les. Tot això es representa en la següent figura 66 on el lila significa que es descarta la cel·la.

0+7	1+6	2+5		∞
1+6		3+4		∞
2+5		4+3	5+2	∞
3+4	4+3			
4+5	5+4	∞	∞	∞

Figura 66: Graella a descobrir el camí. Pas 6. Font: Elaboració pròpia

0+7	1+6	2+5		∞
1+6		3+4		∞
2+5		4+3	5+2	6+1
3+4	4+3			
4+5	5+4	∞	∞	∞

Figura 67: Graella a descobrir el camí. Pas 7. Font: Elaboració pròpia

Com podem observar, en el pas 7 es descarta una segona cel·la i ja tan sols se n'ha d'analitzar una.

0+7	1+6	2+5		∞
1+6		3+4		7+2
2+5		4+3	5+2	6+1
3+4	4+3			7+0
4+5	5+4	∞	∞	∞

Figura 68: Graella a descobrir el camí. Pas 8. Font: Elaboració pròpia

Aquest vuitè pas seria l'últim, ja que l'algoritme ja s'ha topat amb la destinació i, per tant, es conclou el final del bucle. Per tant, el resultat en aquest cas seria que el millor camí π és:

$$\pi = (0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (3, 2), (4, 2), (4, 3) \quad (29)$$

5.5.2 Algorisme de generació d'aparcaments

Un dels objectius d'aquest projecte, a part de crear un sistema automàtic per a la gestió d'un aparcament, és també l'optimització de l'espai d'aquest i, per tant, un punt a tenir en compte és la disposició de les places. Per tal d'acomplir aquesta tasca, el programari necessita un algorisme per a avaluar com és d'efectiva una determinada disposició de les places d'un aparcament.

Funcionament

L'algorisme considera l'aparcament com una graella d'espais. En aquesta graella cada cel·la té un cost de pas segons la seva naturalesa, és a dir, segons si és una cel·la és definida com una plaça, un obstacle o un passadís. Llavors per cada disposició de places L i una entrada E , es calcula amb l'algorisme A^* la distància mínima des d' E fins a cada plaça. Això permet obtenir la distància d'accés mínima $d(s)$ per a cada plaça s . Finalment, realitzant unes mètriques globals que seguidament desenvoluparé es determina el cost final de la disposició de places que a posteriori es podrà comparar amb altres disposicions de places. Per a poder trobar la millor disposició per a cada cas, es generen una sèrie de disposicions de places d'aparcaments aleatòries i llavors s'avaluen seguint l'algorisme.

Model matemàtic del cost de l'aparcament

1. Definició del domini

L : disposició de l'aparcament (graella de cel·les)

$\mathcal{C}$: conjunt de cel·les interiors de l'aparcament

$\mathcal{S} \subset \mathcal{C}$: conjunt de places d'aparcament

$E \in \mathcal{C}$: cel·la d'entrada

Cada cel·la $c \in \mathcal{C}$ té un cost de pas $w(c)$ definit com:

$$w(c) = \begin{cases} 1, & \text{si } c \text{ és passadís,} \\ k, & \text{si } c \in \mathcal{S} \text{ (plaça),} \\ \infty, & \text{si } c \text{ és obstacle o paret,} \end{cases} \quad \text{amb } k > 1. \quad (30)$$

2. Càlcul de distància d'accés mínima

Per a cada plaça $s \in \mathcal{S}$:

$$d(s) = \min_{\pi \in \mathcal{P}(E,s)} \sum_{c \in \pi} w(c), \quad (31)$$

on $\mathcal{P}(E, s)$ és el conjunt de camins viables des d'E fins a s , π és cada camí del conjunt, c és cada cel·la de cada camí i $w(c)$ el cost de pas per cada cel·la.

3. Mètrica global d'accessibilitat

$$C_{\text{access}}(L) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} d(s), \quad (32)$$

Aquesta fórmula ens ajuda a determinar el cost d'accés total de l'aparcament, $C_{\text{access}}(L)$, per fer-ho es divideix el sumatori de les distàncies d'accés de cada cel·la $\sum_{s \in \mathcal{S}} d(s)$, entre el nombre de cel·les, $|\mathcal{S}|$.

4. Densitat de places

$$\text{densitat}(L) = \frac{|\mathcal{S}|}{|\mathcal{C}|}. \quad (33)$$

5. Funció objectiu global

Finalment, aquesta fórmula ens ajuda a determinar la puntuació final de l'aparcament.

$$J(L) = \alpha C_{\text{access}}(L) - \beta \text{densitat}(L), \quad (34)$$

on $\alpha, \beta > 0$ ponderen la importància relativa de capacitat i accessibilitat (p. ex. $\alpha = 1, \beta = 3$).

6. Objectiu final

$$L^* = \arg \min_L J(L). \quad (35)$$

És a dir, es busca la disposició L que minimitza $J(L)$, equilibrant accessibilitat i densitat.

Exemple d'ús de l'algorisme

Per a exemplificar aquest algorisme, calcularem el cost de la disposició de la figura 69 on el color gris significa obstacle o paret, el verd significa entrada, el blanc significa passadís i el taronja significa plaça.

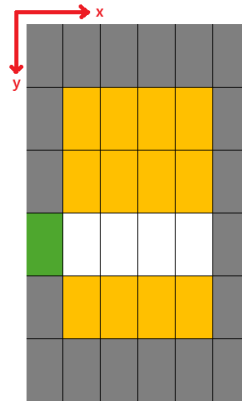


Figura 69: Possible disposició per a un aparcament. Font: Elaboració pròpia

Per començar amb l'anàlisi hem de fer algunes definicions. Aquestes definicions consideren el punt $(0,0)$ l'extrem superior esquerre i el punt $(5,5)$ l'extrem inferior dret.

$$k = 3$$

$$\alpha = 1$$

$$\beta = 3$$

$$\mathcal{C} = \{(1, 1), (1, 2), (1, 3), (1, 4),$$

$$(2, 1), (2, 2), (2, 3), (2, 4),$$

$$(3, 1), (3, 2), (3, 3), (3, 4),$$

$$(4, 1), (4, 2), (4, 3), (4, 4)\},$$

$$\mathcal{S} = \{(1, 1), (1, 2), (1, 4), (2, 1)$$

$$(2, 2), (2, 4), (3, 1), (3, 2),$$

$$(3, 4), (4, 1), (4, 2), (4, 4)\},$$

$$E = (0, 3)$$

Cel·la	Cost	Cel·la	Cost	Cel·la	Cost	Cel·la	Cost
(1,1)	k	(2,1)	k	(3,1)	k	(4,1)	k
(1,2)	k	(2,2)	k	(3,2)	k	(4,2)	k
(1,3)	1	(2,3)	1	(3,3)	1	(4,3)	1
(1,4)	k	(2,4)	k	(3,4)	k	(4,4)	k

Taula 4: Pesos de cel·les: 1 = passadís, k = plaça, ∞ = obstacle.

Ara per a cada plaça $s \in \mathcal{S}$ hem de calcular-ne la distància d'accés. Per exemple, en el cas de la plaça $s = (1, 1)$, el camí π que té més sentit és:

$$\pi = (1, 3), (1, 2), (1, 1) \tag{36}$$

$$d(s) = \sum_{c \in \pi} w(c) = 1 + k + k = 1 + 3 + 3 = 7 \text{ up}^{23} \tag{37}$$

²³Unitats de pas, és el cost per passar.

En el cas de la plaça $s = (4, 4)$, el camí π que té més sentit és:

$$\pi = (1, 3), (2, 3), (3, 3), (4, 3), (4, 4) \quad (38)$$

$$d(s) = \sum_{c \in \pi} w(c) = 1 + 1 + 1 + 1 + k = 1 + 1 + 1 + 1 + 3 = 7 \text{ up} \quad (39)$$

Si fem aixó per a cada plaça s , els resultats són els següents:

Cel·la	Cost	Cel·la	Cost	Cel·la	Cost
(1,1)	7	(2,2)	5	(3,4)	6
(1,2)	4	(2,4)	5	(4,1)	10
(1,4)	4	(3,1)	9	(4,2)	7
(2,1)	8	(3,2)	6	(4,4)	7

Taula 5: Distància d'accés de cada cel·la.

En aquest punt, és hora de calcular la mètrica global d'accessibilitat i la densitat de places:

$$C_{\text{access}}(L) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} d(s) = \frac{1}{12} \cdot (7+4+4+8+5+5+9+6+6+10+7+7) = \frac{78}{12} = 6,5 \text{ up} \quad (40)$$

$$\text{densitat}(L) = \frac{|\mathcal{S}|}{|\mathcal{C}|} = \frac{12}{16} = 0,75 \text{ places/cel} \cdot \text{la} \quad (41)$$

Finalment hem de calcular la funció objectiu global, el cost de la disposició:

$$J(L) = \alpha C_{\text{access}}(L) - \beta \text{densitat}(L) = 1 \cdot 6,5 - 3 \cdot 0,75 = 4,25 \text{ punts} \quad (42)$$

Per tant, la puntuació d'aquesta disposició seria de 4,25 i utilitzariem aquest nombre per a comparar-la amb d'altres disposicions i escollir la millor per al nostre aparcament.

Part II

Implementació pràctica

6 Construcció

En aquest punt del projecte era l'hora de començar amb la construcció del prototip. Els primers passos varen ser imprimir les diferents peces necessàries per la construcció del robot:

- Base inferior del robot: peça on s'ancoren els diversos components mecànics com els motors i servomotors
- Base superior del robot: peça superior del robot que s'eleva i on s'ancora la bateria externa.
- Suport per la bateria externa: peça dissenyada per contenir la bateria externa
- 28 rodets per muntar les rodes Mecanum
- 8 peces per muntar les rodes Mecanum
- 4 arbres: transmeten el moviment dels motors a les rodes
- 2 excèntriques: encarregades d'eleva la base superior del robot
- 2 guies per les excèntriques: encarregades que les excèntriques no escapin la seva posició

Segons el programari utilitzat durant la impressió 3D, el temps total d'impressió per totes les peces va ser d'unes 13 hores utilitzant 129 metres de filament, i per tant uns 400 grams de filament. Un cop totes les peces van estar impreses, es van muntar i ajuntar utilitzant els següents caragols autoroscants:

- 8 caragols M3 de 25 mm pels 4 motors amb caixa reductora
- 4 caragols M3 de 16 mm pel suport per la bateria externa

- 4 caragols M3 de 12 mm pels 2 servomotors
- 8 caragols M3 de 8 mm pels 4 sensors VL53L0X

La resta de components que no s'han esmentat a la llista varen ser muntats utilitzant encaixos a pressió i encaixos per clip. En aquest instant del treball, en termes de construcció ja tan sols faltava la fabricació de les plaques de circuit imprès o PCB. Per la fabricació de la PCB, vaig utilitzar els serveis de l'empresa JLCPCB, i amb l'ajuda i el suport d'en Dom Qin i en Bob Bao, dos treballadors d'EasyEDA (empresa que ofereix programari de disseny de plaques de circuit imprès) vaig tenir la possibilitat d'utilitzar aquests serveis de fabricació de forma gratuïta. Abans d'enviar les PCB es va comprovar el seu circuit utilitzant plaques de prova, com podem observar a la figura 70, i en ser aquesta prova exitosa es van encarregar les plaques de circuit imprès.

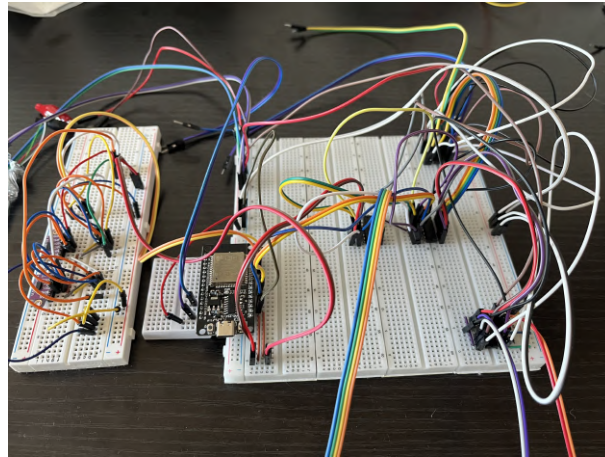


Figura 70: Fotografia de les plaques de prova utilitzades. Font: Elaboració pròpia

Un cop van arribar les PCB sense els components soldats, figura 71, utilitzant un soldador i estany, figura 72, es van muntar tots els components a la placa donant com a resultat la placa de la figura 73.

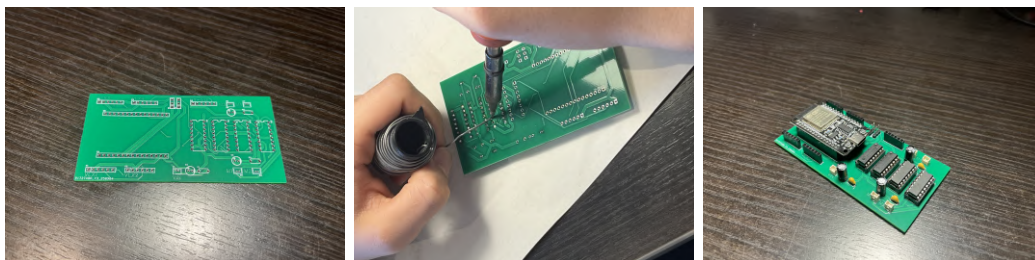


Figura 71: Fotografia de les PCB sense components. Font: Elaboració pròpia
Figura 72: Fotografia del procès de soldadura. Font: Elaboració pròpia
Figura 73: Fotografia de les PCB amb components. Font: Elaboració pròpia

Amb les plaques muntades, es va provar el circuit de nou i es va detectar un error. Es tractava d'un error de connexió entre dues malles. Em vaig adonar que durant el disseny de les connexions de la placa no es van connectar dues malles (línies de coure) i, per tant, hi havia una malla per la qual no circulava tensió. Això es va arreglar molt fàcilment fent una soldadura amb cable de coure rígid, com podem veure encerclat a la següent figura 74.

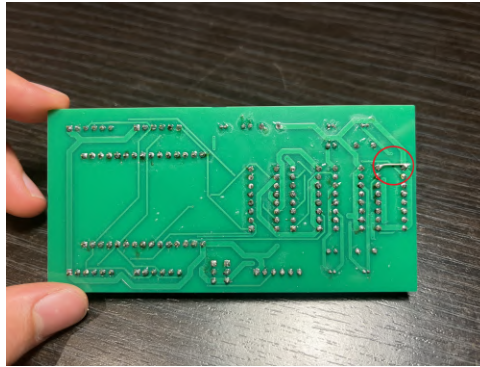


Figura 74: Fotografia de la soldadura per solucionar l'error. Font: Elaboració pròpia

Aquest és el resultat final de la construcció del robot amb tots els seus components tant mecànics com elèctrics:

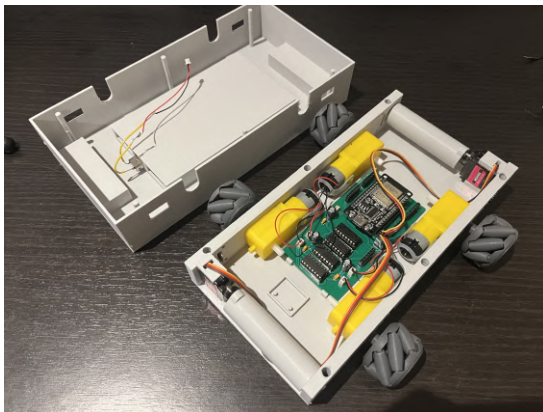


Figura 75: Fotografia del robot obert acabat. Font: Elaboració pròpia



Figura 76: Fotografia del robot tancat acabat. Font: Elaboració pròpia

Encara que aquest prototip del robot va equipat amb un mòdul GY-512, acceleròmetre i giroscopi, i dos mòduls VL53L0X, per mesurar distàncies, cap d'aquests sensors és utilitzat realment, sinó que han estat equipats al robot com a proposta de millora pel futur.

6.1 Determinació velocitat del robot

Seguint el model cinemàtic del robot, definit a l'apartat 3.4, necessitem la velocitat angular dels motors per obtenir la velocitat lineal del robot. Per determinar la velocitat angular dels motors, s'ha utilitzat el programari Tracker, una eina de processament de vídeo que permet analitzar el moviment a partir d'imatges enregistrades.

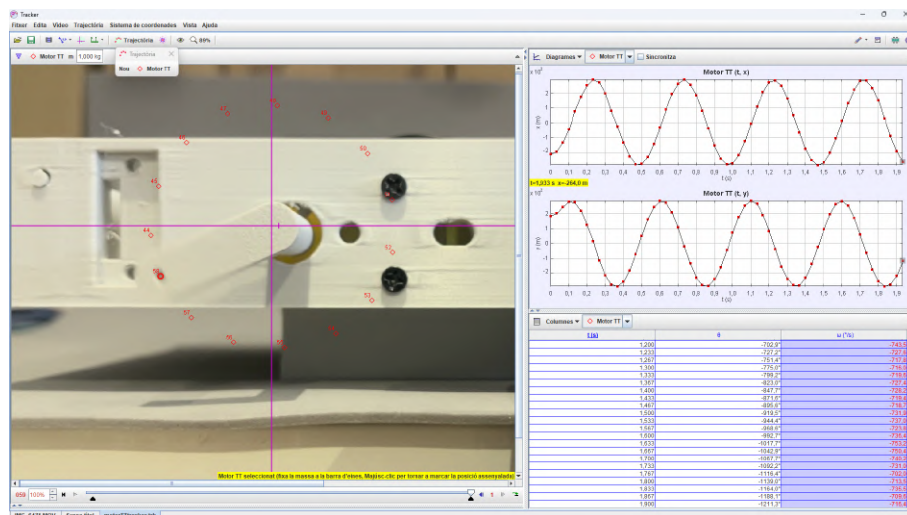


Figura 77: Captura de pantalla del programari Tracker. Font: Elaboració pròpia

Després de fer la mitjana de les diverses velocitats angulars que determina el Tracker, obtenim que:

$$\omega = 12,4912 \text{ rad/s} \quad (43)$$

En aquest punt utilitzant les fórmules del model cinemàtic i coneixent el radi de les rodes podem trobar les velocitats, recordem que:

$$v_x(t) = \frac{r}{4} (\omega_1 + \omega_2 + \omega_3 + \omega_4) \quad (44)$$

$$v_y(t) = \frac{r}{4} (-\omega_1 + \omega_2 + \omega_3 - \omega_4) \quad (45)$$

En el cas d'anar cap endavant o enrere les velocitats són:

$$v_x(t) = \frac{0,0185}{4} (12,4912 + 12,4912 + 12,4912 + 12,4912) = 0,2311 \text{ m/s} \quad (46)$$

$$v_y(t) = \frac{0,0185}{4} (-12,4912 + 12,4912 + 12,4912 - 12,4912) = 0 \text{ m/s} \quad (47)$$

En el cas d'anar cap a la dreta o esquerra les velocitats són:

$$v_x(t) = \frac{0,0185}{4} (12,4912 + (-12,4912) + (-12,4912) + 12,4912) = 0 \text{ m/s} \quad (48)$$

$$v_y(t) = \frac{0,0185}{4} (-12,4912 + (-12,4912) + (-12,4912) - 12,4912) = 0,2311 \text{ m/s} \quad (49)$$

Així doncs, ara ja podem tenir en compte aquestes velocitats a l'hora de determinar les ordres per a les diverses accions del robot.

7 Programació

En aquest apartat es tracta de forma resumida utilitzant diagrames de flux els principals programes del projecte. És cert que en aquestes explicacions no s'ha pogut entrar molt en detall, ja que si sumem totes les línies de codi del projecte, estem tractant amb unes 1800 línies de codi i, per tant, impossibles d'explicar enterament. Tot el codi del treball es pot consultar en un repositori de GitHub, explicat a l'annex A.

7.1 Programació per generar el mapa de l'aparcament

Aquest primer programa és l'encarregat de transformar els arxius DXF en arxius amb els quals el robot pugui treballar i entendre per tal de moure's a través de l'aparcament. El seu diagrama de flux i la seva interfície gràfica són els següents:

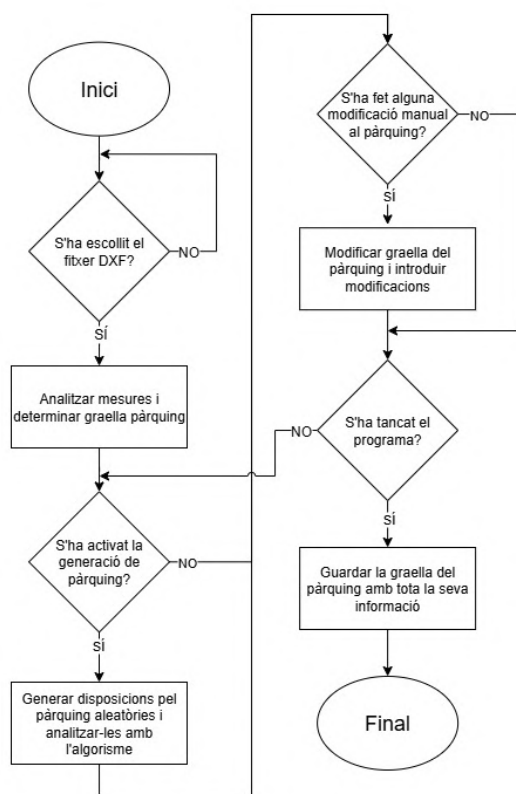


Figura 78: Diagrama de flux de l'aplicació per definir el mapa. Font: Elaboració pròpia

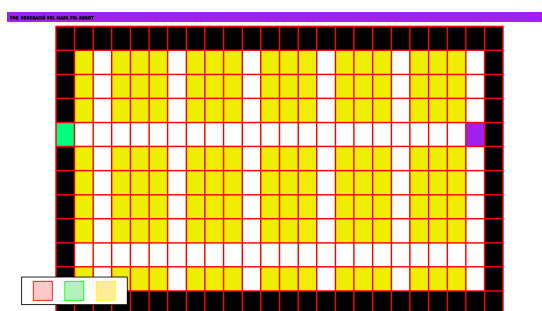


Figura 79: Captura de pantalla de l'aplicació per fer definir el mapa. Font: Elaboració pròpia

7.2 Programació per simular l'aparcament

Aquest segon programa és l'encarregat de rebre les dades de la base de dades, computar les ordres pel robot i enviar aquestes al servidor HTTP. El seu diagrama de flux i la seva interfície gràfica són els següents:

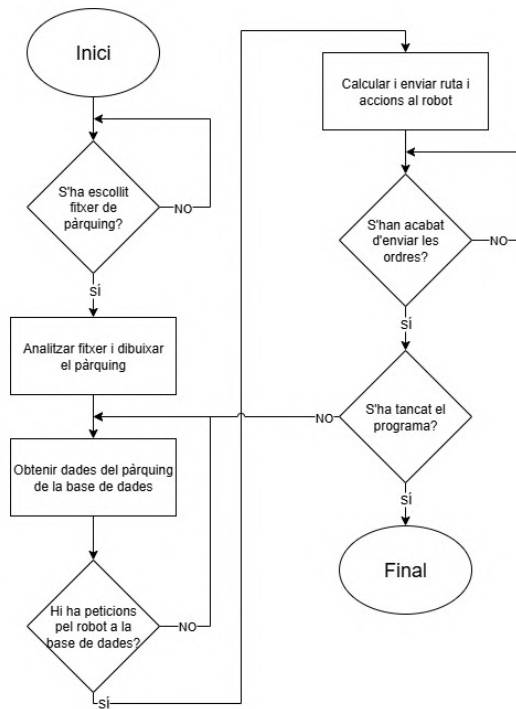


Figura 80: Diagrama de flux de l'aplicació per fer les simulacions. Font: Elaboració pròpia

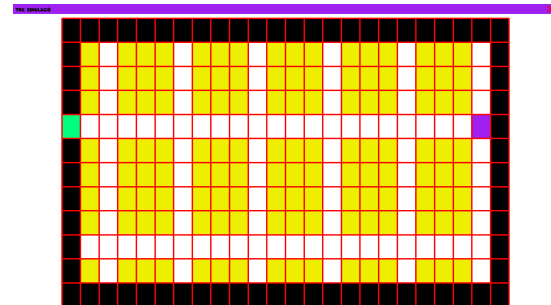


Figura 81: Captura de pantalla de l'aplicació per fer les simulacions. Font: Elaboració pròpia

7.3 Programació del robot

Aquest tercer programa és l'encarregat de rebre les ordres del servidor HTTP i accionar els diversos actuadors. El seu diagrama de flux és el següent:

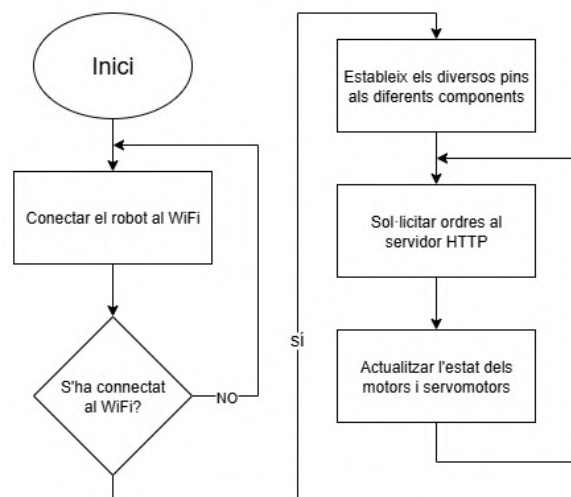


Figura 82: Diagrama de flux del robot. Font: Elaboració pròpia

7.4 Programació de l'aplicació mòbil

Aquest últim programa és l'encarregat d'actualitzar les dades de la base de dades i fer les peticions per posar o treure un cotxe. El seu diagrama de flux i la seva interfície gràfica són els següents:

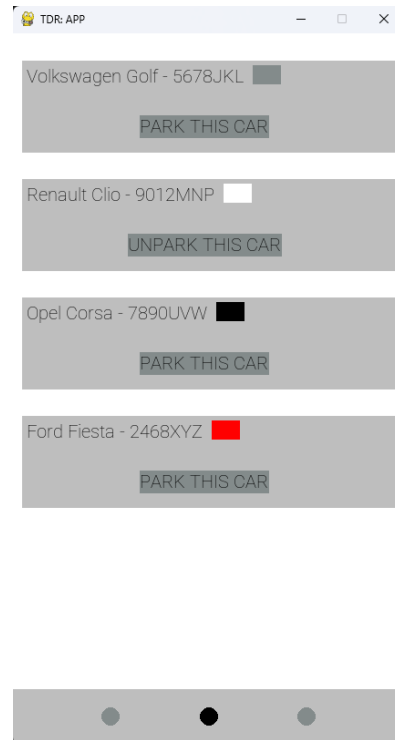
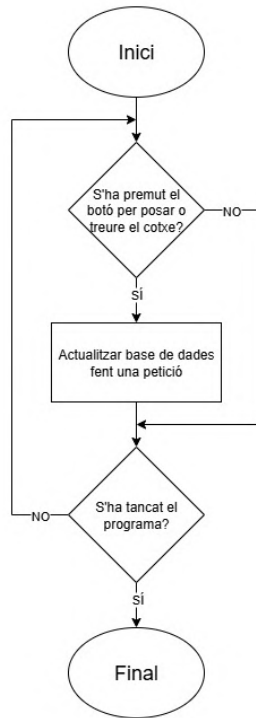


Figura 83: Diagrama de flux de l'aplicació mòbil. Figura 84: Captura de pantalla de l'aplicació mòbil. Font: Elaboració pròpia

7.5 Resum sobre la programació

En resum, si ajuntem totes les funcionalitats d'aquests quatre programes tenim el següent:

- Som capaços de transformar arxius DXF a un format llegible per la simulació.
- Podem mirar si hi ha peticions per posar o treure un cotxe i generar les ordres necessàries per al robot.
- També podem generar peticions per posar o treure els cotxes usant l'aplicació mòbil.
- En últim lloc, el robot és capaç d'aconseguir les ordres per part de la simulació i portar-les a terme.

8 Cost del prototip

Aquesta taula resumeix el cost del prototip final:

Producte	Unitats	Cost/unitat (€)	Cost total (€)
Impressions 3D	0,4 kg	23 €/kg	9,20 €
Caragols autoroscants	1	5,59 €	5,59 €
Motors amb caixa reductora	4	1,05 €	4,20 €
Servomotor MG90S	2	2 €	4,00 €
ESP32	1	4,15 €	4,15 €
Bateria externa INIU B41	1	42,08 €	42,08 €
Mòdul VL53L0X	5	1,128 €	6,09 €
Mòdul GY-512	1	1,30 €	1,30 €
Xips L293D	10	0,284 €	2,84 €
Xips SN74HC595	10	0,11 €	1,10 €
Condensadors electrolítics	20	0,1595 €	3,19 €
Condensadors ceràmics	100	0,0153 €	1,53 €
Altres components electrònics	1	11,14 €	11,14 €
PCB	5	0,00 €	0,00 €
Gestions enviament PCB	1	40,00 €	40,00 €
Total			136,41 €

Taula 6: Desglossament de costos del projecte

Part III

Tancament i propostes

9 Resultats

En primer lloc, en aquest apartat s'explica el funcionament elemental del robot prototip i, seguidament, s'exposa la precisió del robot en termes de moviment.

9.1 Producte obtingut

Finalment, després de tota la recerca, programació, disseny i feina feta el resultat obtingut ha estat el següent. S'ha aconseguit crear un prototip d'un robot funcional que és capaç d'elevat cotxes per sota i moure's en totes les direccions utilitzant les rodes Mecanum. Aquest prototip crec que aconsegueix de forma exitosa associar el terme “smart city”, és a dir, optimització, eficiència i productivitat al meu model d'aparcaments. És cert que encara es podrien aplicar un munt de millores al projecte, però la funcionalitat bàsica ha estat assolida amb aquest prototip. Seguidament, passo a explicar com funcionaria en un cas real.

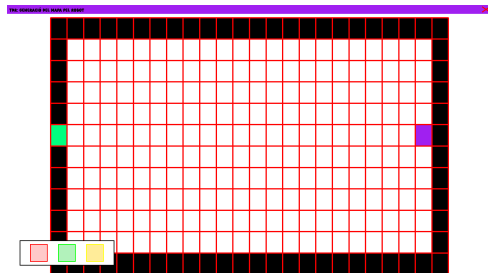


Figura 85: Captura de pantalla de l'aplicació per generar el mapa. Pas 1. Font: Elaboració pròpia

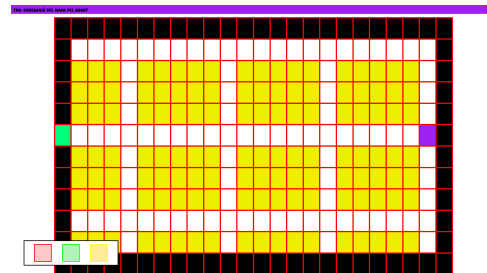


Figura 86: Captura de pantalla de l'aplicació per generar el mapa. Pas 2. Font: Elaboració pròpia

En la figura 85 es pot veure que tenim un mapa obert en el programa de generació d'aparcaments, això vol dir que ja s'hauria indicat quin arxiu DXF s'hauria d'utilitzar. Seguidament, es definiria l'entrada, de color verd, i la posició d'inici del robot indicada en lila. Després s'executaria l'algorisme de generació de l'aparcament i com a resultat s'obtindria la figura 86, on es pot veure que ja hi ha definides totes les places de l'aparcament en color groc. En aquest punt amb la disposició de l'aparcament ja

definida s'hauria acabat la feina en aquest programa i a partir d'ara s'hauria d'utilitzar el programa de simulació d'aparcament i fer ús de l'aplicació mòbil per fer peticions d'aparcament.

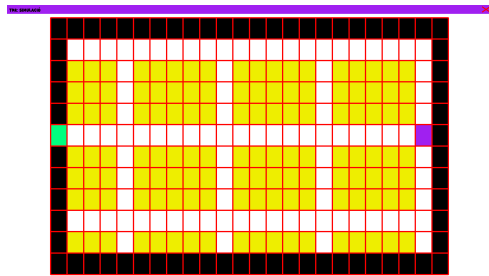


Figura 87: Captura de pantalla de l'aplicació de simulació. Pas 1. Font: Elaboració pròpia

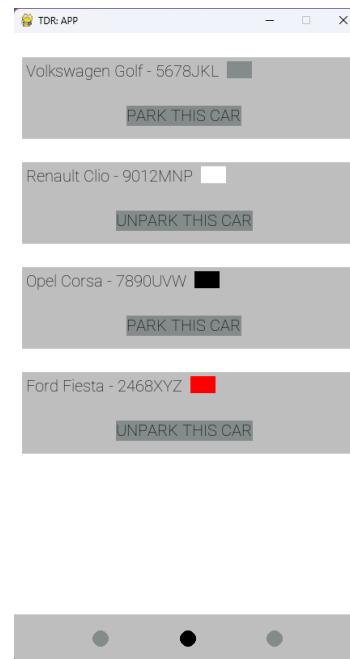


Figura 88: Captura de pantalla de l'aplicació mòbil. Font: Elaboració pròpia

En el moment en què alguna persona utilitzant l'aplicació mòbil fés una sol·licitud per aparcar el seu cotxe, aquesta, arribaria al programa de simulació que calcularia les ordres pel robot i les hi enviaria. Llavors, el robot faria efectiva l'acció ordenada. A més a més, a l'aplicació de simulació també es mostraria el moviment i s'actualitzaria el mapa.

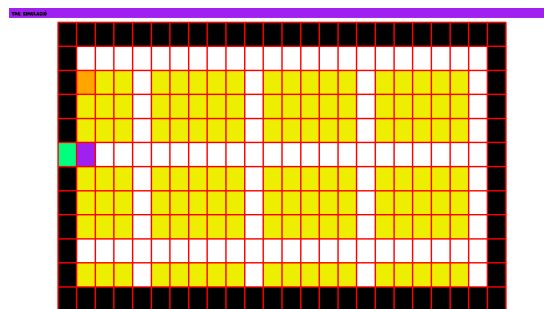


Figura 89: Captura de pantalla de l'aplicació de simulació. Pas 2. Font: Elaboració pròpia

El prototip obtingut es tracta d'un prototip funcional que es podria millorar si es tinguessin en compte els valors dels sensors o s'utilitzessin altres tipus de sensors que fossin capaços de donar més precisió.

9.2 Precisió del prototip

Per mesurar la precisió del robot es van fer dues proves, una per mesurar l'error en el moviment endavant - enrere, i una altra prova per mesurar l'error en el moviment dreta - esquerra. Per simular l'aparcament es va utilitzar una làmina de goma EVA on es va marcar una quadrícula d'1*1 cm utilitzant una talladora làser. Aquesta quadrícula em va facilitar la mesura de l'error de distància. L'experiment va consistir a col·locar el robot sobre la quadrícula i donar-li l'ordre de moure's 100 mm cap a la dreta / esquerra o endavant / enrere. Un cop fet el moviment podia mesurar la distància real que havia recorregut el robot, i la diferencia respecte als 100 mm és l'error de distància que pot fer el robot.

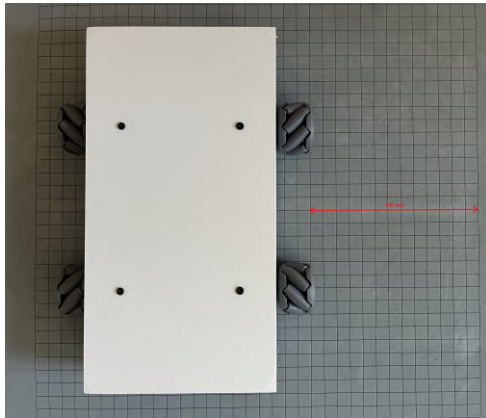


Figura 90: Fotografia del robot abans de realitzar el moviment. Font: Elaboració pròpia

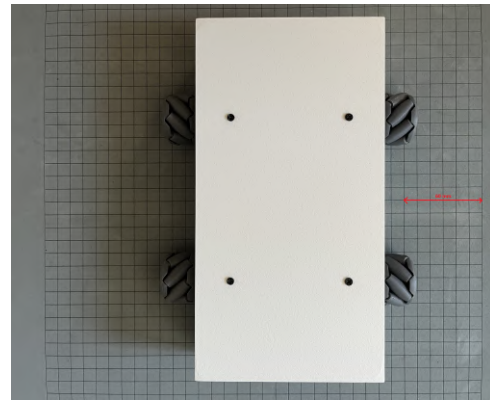


Figura 91: Fotografia del robot després del moviment. Font: Elaboració pròpia

En aquest cas, el de les figures 90 i 91, el moviment va ser de 70 mm, malgrat que el moviment desitjat era de 100 mm, per tant, la desviació en aquest cas va ser de 30 mm. Aquestes són les dades obtingudes amb les dues proves:

Distància desitjada	Distància obtinguda	Diferència
100	110	10
100	100	0
100	100	0
100	90	-10
100	100	0
100	80	-20
100	100	0
100	100	0
100	120	20
100	120	20
100	100	0
100	110	10
100	90	-10
100	150	50
100	100	0

Taula 7: Errors de moviment del robot — endavant / enrere

Distància desitjada	Distància obtinguda	Diferència
100	80	-20
100	80	-20
100	80	-20
100	100	0
100	80	-20
100	90	-10
100	190	90
100	90	-10
100	80	-20
100	70	-30
100	80	-20
100	70	-30
100	80	-20
100	70	-30
100	80	-20

Taula 8: Errors de moviment del robot — dreta / esquerra

Llavors si calculem l'error mitjà per cada sèrie de proves:

$$\begin{aligned}EM_1 &= \frac{10 + 0 + 0 - 10 + 0 - 20 + 0 + 0 + 20 + 20 + 0 + 10 - 10 + 50 + 0}{15} \\&= \frac{70}{15} = 4,667 \text{ mm} \\EM_2 &= \frac{-20 - 20 - 20 + 0 - 20 - 10 + 90 - 10 - 20 - 30 - 20 - 30 - 20 - 30 - 20}{15} \\&= \frac{-180}{15} = -12,0 \text{ mm}\end{aligned}\tag{50}$$

Com es pot veure, el robot no acaba de ser del tot precís, sinó que en el cas dels moviments endavant / enrere per cada 100 mm, de mitjana, té una desviació d'uns 4,667 mm, per tant, una desviació del 4,667%. Per altra banda, amb els moviments dreta / esquerra encara tenim una desviació més gran, per cada 100 mm, hi ha una desviació de 12 mm, o sigui que una desviació del 12%.

9.3 Possibles millores

Ara, amb els resultats obtinguts i el projecte finalitzat és hora de veure com i quines coses es podrien millorar per ser més precisos i eficients:

- Utilitzar el mòdul GY-512, equipat amb acceleròmetre i giroscopi, el mòdul VL53L0X, per mesurar distàncies, i el mòdul BU01, per conèixer la ubicació del robot. D'aquesta forma es milloraria la precisió fent diverses maniobres de correcció.
- Un altre problema que es va detectar va ser els retards entre el fet d'enviar les ordres al robot i que aquest les rebés. Per millorar-ho s'hauria de canviar el sistema informàtic i plantejar una solució de comunicacions diferent de l'actual. En el meu prototip actual les ordres es van donant una rere l'altra separades en el temps, i es podria plantejar enviar-les totes juntes i que el robot les anés seguint.
- En últim lloc, també intentaria millorar el sistema pels usuaris afegint més funcionalitats i eines. Per exemple, opcions sobre l'hora de recollida del cotxe o informació en temps real sobre l'estat del seu vehicle.

10 Conclusions

En aquest treball de recerca he pogut aprofundir en diversos temes sobre robòtica, programació i matemàtiques, que m'han estat útils per aconseguir el meu objectiu final. L'objectiu era desenvolupar una proposta per millorar la gestió dels aparcaments utilitzant les característiques que defineixen una ciutat intel·ligent. He començat buscant i definint què és una ciutat intel·ligent per poder aplicar el concepte a la meua idea. He descobert que al concepte de ciutat intel·ligent se li comença a donar forma des del segle XV quan da Vinci proposa la ciutat planificada i que avui en dia aquest terme ja no només s'aplica a les ciutats sinó que s'ha fet extensiu a molts àmbits entre ells al món empresarial.

Un cop plantejada una idea, cal portar-la a la realitat, i per això sovint fa falta un bon treball de recerca, és a dir, cercar la informació útil per fer efectiva la idea. En el meu cas, la idea per complir el meu objectiu ha estat la construcció d'un robot i he fet molta recerca sobre mecànica, electrònica i eines de programació. El procés de desenvolupament d'aquest treball no ha estat gens fàcil. Tot i tenir tota la informació que necessitava i tots els components que calien per construir el robot, m'ha fet falta crear cinc versions del robot i dues versions de la placa PCB per arribar al prototip funcional definitiu. Així doncs, he pogut comprovar com en l'evolució del disseny d'un prototip hi ha moltes fases de desenvolupament i experimentació abans no s'arriba a l'objectiu desitjat.

Pel que fa a la programació, he treballat principalment amb Python i C++. Amb Python, he pogut explorar les ordres més bàsiques i algunes llibreries externes que amplien les funcionalitats del llenguatge. En paral·lel, he fet servir C++ per programar el microcontrolador, que actua com a cervell del robot. A més de la programació, també he gaudit especialment aplicant conceptes matemàtics per resoldre reptes del projecte.

He fet ús de les matemàtiques per aplicar principalment dos algorismes: l'algorisme a-estrella i l'algorisme de collita pròpia per a la generació de les disposicions de places per als aparcaments. Gràcies a l'algorisme a-estrella, el robot és capaç de trobar la ruta òptima dins l'aparcament per anar a les diverses places. Durant la recerca d'aquest algorisme he tingut la sort d'endinsar-me en la teoria de grafs, la branca de les matemàtiques en la que es basa aquest algorisme, i descobrir-ne un munt de coses. A partir de l'algorisme a-estrella jo he desenvolupat un algorisme propi que avalua les diverses disposicions dels aparcaments per millorar l'eficiència i productivitat.

Després d'aconseguir la construcció del robot definitiu ha quedat clar que tota la recerca

ha valgut la pena i ha estat recompensada amb un prototip funcional que utilitza tot el que he après per portar les característiques de les ciutats intel·ligents cap als aparcaments. A més, he estat capaç de provar-lo, avaluar-ne la precisió i planificar-ne certes millores de cara al futur.

Finalment, a nivell personal, aquest treball m'ha aportat un munt de coneixement i experiències que em valdran per a tota la vida. El fet d'aconseguir l'objectiu final del projecte ha estat molt positiu i recompensador. A més a més, el fet de desenvolupar un algorisme propi mostra tot l'aprenentatge que m'ha aportat aquest projecte. Aquest treball de recerca m'ha confirmat que vull continuar estudiant per formar-me en el món de la tecnologia i les matemàtiques. M'ha agradat molt treballar amb algorismes i en la creació d'una PCB. Crec que aquest treball ha estat possible gràcies a la curiositat i interès que he tingut des de ben petit, cosa que vull mantenir fins que no sigui possible.

Fonts d'informació

- [1] Adafruit. *DC Gearbox Motor - "TT Motor" - 200RPM - 3 to 6VD*. 2018. URL: <https://www.adafruit.com/product/3777> (consultat el 10/06/2025).
- [2] Adobe. *Información sobre archivos DXF — Adobe*. 2025. URL: <https://www.adobe.com/es/creativecloud/file-types/image/vector/dxf-file.html> (consultat el 31/08/2025).
- [3] Mamtaz Alam. *Getting Started with ESP32 DW1000 UWB (Ultra Wideband) Module*. 2024. URL: <https://how2electronics.com/getting-started-with-esp32-dw1000-uwu-ultra-wideband-module/> (consultat el 31/08/2025).
- [4] Isabel Rubio Arroyo. *La "ciudad ideal" diseñada por Leonardo da Vinci*. 2022. URL: <https://sacyr.com/-/la-ciudad-ideal-disenada-por-leonardo-da-vinci> (consultat el 08/10/2025).
- [5] BonpreuEsclat. *Hem inaugurat el nou magatzem de productes refrigerats a Balenya*. 2017. URL: <https://www.bonpreuesclat.cat/blog/-/blogs/hem-inaugurat-el-nou-magatzem-de-productes-refrigerats-a-balenya> (consultat el 08/10/2025).
- [6] Dmitry Borisov. *Mecanum Wheel - 2188345*. 2017. URL: <https://www.thingiverse.com/thing:2188345> (consultat el 27/02/2026).
- [7] Blue Butterfly. *How Gyroscope Sensor Works ? — 3D Animated*. 2024. URL: <https://www.youtube.com/watch?v=REvp33SwwHE> (consultat el 01/09/2025).
- [8] Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. "Smart Cities in Europe". Dins de *Journal of Urban Technology* 18.2 (2011), pàgines 65–82. DOI: [10.1080/10630732.2011.601117](https://doi.org/10.1080/10630732.2011.601117). eprint: <https://doi.org/10.1080/10630732.2011.601117>. URL: <https://doi.org/10.1080/10630732.2011.601117> (consultat el 08/10/2025).
- [9] Alex Chen. *PCB Layers Explained: Everything You Need to Know*. 2024. URL: <https://www.pcbasic.com/blog/pcb-layers.html> (consultat el 05/10/2025).
- [10] CircuitBread. *How does an Accelerometer work? — 3D Animation*. 2023. URL: <https://www.youtube.com/watch?v=KuekQ-m9xpw> (consultat el 01/09/2025).
- [11] Sabin Ingeniería Civil. *¡La Brillante Ingeniería de las Ruedas Mecanum!* 2022. URL: <https://www.youtube.com/watch?v=PYLt2bCU5Ek> (consultat el 02/09/2025).
- [12] Components101. *MG90S Micro Servo Motor Datasheet, Wiring Diagram & Features*. 2019. URL: <https://components101.com/motors/mg90s-metal-gear-servo-motor> (consultat el 31/08/2025).
- [13] Wikipedia contributors. *Mecanum wheel - Wikipedia*. 2025. URL: https://en.wikipedia.org/wiki/Mecanum_wheel (consultat el 02/09/2025).

- [14] EasyEDA. *EasyEDA - Online PCB design & circuit simulator*. 2025. URL: <https://easyeda.com/> (consultat el 05/10/2025).
- [15] Matan Electricity and Magnetism. *Motores servo – Electricity – Magnetism*. 2023. URL: <https://www.electricity-magnetism.org/es/motores-servo/> (consultat el 31/08/2025).
- [16] Matan Electricity and Magnetism. *Motores servo – Electricity – Magnetism*. 2023. URL: <https://www.electricity-magnetism.org/es/motores-servo-2/> (consultat el 31/08/2025).
- [17] Matan Electricity and Magnetism. *¿Cuál es el principio de funcionamiento de un motor de corriente continua?* 2023. URL: <https://www.electricity-magnetism.org/es/cual-es-el-principio-de-funcionamiento-de-un-motor-de-corriente-continua/> (consultat el 19/06/2025).
- [18] Escuela formativa. *Tema-4-Sensores-y-Actuadores-en-Robotica.pdf*. 2024. URL: <https://escuelaformativa.com/wp-content/uploads/2024/08/Tema-4-Sensores-y-Actuadores-en-Robotica.pdf> (consultat el 10/06/2025).
- [19] Python Software Foundation. *Welcome to Python.org*. 2025. URL: <https://www.python.org/> (consultat el 03/10/2025).
- [20] freelogovectors.net. *Autodesk Fusion 360 Logo - PNG Logo Vector Brand Downloads (SVG, EPS)*. 2023. URL: <https://www.freelogovectors.net/autodesk-fusion-360-logo-2/> (consultat el 05/09/2025).
- [21] Mila Gascó-Hernandez. *Building a Smart City: Lessons from Barcelona*. 2018. URL: <https://cacm.acm.org/research/building-a-smart-city/> (consultat el 08/10/2025).
- [22] Nurallah Ghaeminezhad Hamid Taheri Bing Qiao. “Kinematic Model of a Four Mecanum Wheeled Mobile Robot”. Dins de *International Journal of Computer Applications* 113.3 (2015), pàgines 6–9. ISSN: 0975-8887. DOI: [10.5120/19804-1586](https://doi.org/10.5120/19804-1586). URL: <https://ijcaonline.org/archives/volume113/number3/19804-1586/> (consultat el 05/09/2025).
- [23] IBM. *IBM Smarter Planet Campaign*. 2008. URL: <https://www.commartts.com/project/17079/ibm-smarter-planet-campaign> (consultat el 08/10/2025).
- [24] IBM. *Sustainability — IBM*. 2025. URL: <https://www.ibm.com/history/sustainability> (consultat el 08/10/2025).
- [25] Jobit Joseph. *How Does the MPU6050 Accelerometer - Gyroscope Sensor Work and Interfacing It With Arduino*. 2022. URL: <https://circuitdigest.com/microcontroller-projects/interfacing-mpu6050-module-with-arduino> (consultat el 01/09/2025).

- [26] 3D-Tool GmbH & Co. KG. *Affordable Easy-To-Use CAD Viewer and Converter - 3D-Tool*. 2026. URL: <https://www.3d-tool.com/> (consultat el 27/02/2026).
- [27] Rajesh Kumar. *The A* Algorithm: A Complete Guide*. 2024. URL: <https://www.datacamp.com/tutorial/a-star-algorithm> (consultat el 03/10/2025).
- [28] Bambu Lab. *Bambu Lab A1 — Desktop 3D Printer — A1 series — Bambu Lab EU*. 2025. URL: <https://bambulab.com/en-eu/a1> (consultat el 05/09/2025).
- [29] Bambu Lab. *Ultimate Guide to PLA 3D Printing Filament — Bambu Lab EU*. 2025. URL: <https://bambulab.com/en-eu/filament/pla> (consultat el 05/09/2025).
- [30] LastMinuteEngineers.com. *How Servo Motor Works & Interface It With Arduino - Last Minute Engineers*. 2025. URL: <https://lastminuteengineers.com/servo-motor-arduino-tutorial/> (consultat el 31/08/2025).
- [31] Isaac Hardware libre. *28BYJ-48: todo lo que necesitas saber sobre este motor paso a paso*. 2024. URL: <https://www.hwlibre.com/28byj-48/> (consultat el 19/06/2025).
- [32] Isaac Hardware libre. *Cómo usar el módulo GY-521 con Arduino: Guía completa*. 2024. URL: <https://www.hwlibre.com/como-usar-el-modulo-gy-521-con-arduino-guia-completa/> (consultat el 01/09/2025).
- [33] Isaac Hardware libre. *Motor paso a paso: integración con Arduino*. 2019. URL: <https://www.hwlibre.com/motor-paso-a-paso/> (consultat el 19/06/2025).
- [34] Isaac Hardware libre. *Servo: cómo usar el servomotor con Arduino*. 2021. URL: <https://www.hwlibre.com/servo-arduino/> (consultat el 31/08/2025).
- [35] Isaac Hardware libre. *Todo sobre el VL53L0X: Sensor de distancia láser de alta precisión*. 2024. URL: <https://www.hwlibre.com/todo-sobre-el-vl53l0x-sensor-de-distancia-laser-de-alta-precision/> (consultat el 31/08/2025).
- [36] Isaac Hardware libre. *VL53L0X: sensor de distancia láser de alta precisión*. 2024. URL: <https://www.hwlibre.com/vl53l0x/> (consultat el 31/08/2025).
- [37] Slind Parts. *Ruedas Mecanum: ¿Qué son?, ventajas y aplicaciones*. 2024. URL: <https://www.slindparts.com/que-son-las-ruedas-mecanum/> (consultat el 02/09/2025).
- [38] Dilip Raja. *MEMS - Here is Everything you need to know about Micro Electro Mechanical Systems*. 2024. URL: <https://circuitdigest.com/tutorial/what-is-mems-various-mems-devices-and-applications> (consultat el 01/09/2025).
- [39] Espressif Systems. *ESP Modules — Espressif Systems*. 2025. URL: <https://www.espressif.com/en/products/modules> (consultat el 01/09/2025).
- [40] Rebecca Tan. *National Computer Board*. 2021. URL: <https://www.nlb.gov.sg/main/article-detail?cmsuuid=f77d8fbe-07c0-4180-9761-8e6b47539c0f> (consultat el 08/10/2025).

- [41] Josué Torrado. *HISTORIA DEL CONCEPTO DE SMART CITY*. 2016. URL: <https://inteligenciayconvivenciaciudadelfuturo.wordpress.com/2016/08/15/historia-del-conceto-de-smart-city-2/> (consultat el 08/10/2025).
- [42] Mark Vallianatos. *Uncovering the Early History of “Big Data” and the “Smart City” in Los Angeles*. 2015. URL: <https://boomcalifornia.org/2015/06/16/uncovering-the-early-history-of-big-data-and-the-smart-city-in-la/> (consultat el 08/10/2025).
- [43] Hongyang Zhou. *Songdo International Business District*. 2015. URL: <https://atlasofurbantech.org/cases/kor-incheon-songdo-ibd/> (consultat el 08/10/2025).

Part IV

Annexos

A Eines emprades en la realització del treball

Per dur a terme aquest treball he fet un ús intensiu de diverses eines informàtiques, que han estat essencials per al seu desenvolupament. Gràcies a aquestes eines he pogut elaborar tant la part escrita com la de disseny i implementació del prototip, facilitant-ne així el procés i millorant-ne el resultat final.

A.1 Sistema operatiu

Per escriure aquest treball i desenvolupar el prototip i el seu programari he utilitzat el sistema operatiu Windows, desenvolupat per Microsoft. Treballar en entorns Windows resulta pràctic i accessible, ja que ofereix una àmplia compatibilitat amb diferents programes i eines de desenvolupament. A més, disposa d'una gran varietat d'aplicacions i entorns de treball que faciliten la programació i la gestió de projectes informàtics.

A.2 Confecció del document escrit

A.2.1 $\text{\LaTeX}$ i TeXworks

Per redactar aquest treball he utilitzat el sistema $\text{\LaTeX}$ que a diferència d'altres programes com Microsoft Word o Google Documents, permet un major control del resultat final i facilita la inserció de fórmules matemàtiques i altres figures.

A banda de tot això també permet l'ús de paquets que encara faciliten més la redacció afegint més funcionalitats. En el cas d'aquest treball, s'han utilitzat els següents paquets: *geometry*, *amssymb*, *amsmath*, *graphicx*, *fancyhdr*, *titlesec*, *algorithm*, *algpseudocode*, *parskip*, *tcolorbox*, *caption*, *tocloft*, *hyperref*, *hypcap*, *booktabs*, *pdfpages*, *inputenc* i *biblatex*.

Quan s'acaba d'escriure el $\text{\LaTeX}$, es compila i es genera un PDF. Per fer aquesta tasca, he utilitzat el programari TeXworks, que permet veure com avança la redacció a mesura que s'escriu.

A.2.2 Microsoft Paint

Per editar algunes imatges i ajustar detalls gràfics he utilitzat Microsoft Paint. Aquest programa permet retocar colors, formes i afegir elements senzills de manera ràpida, cosa que ha facilitat la creació de figures clares i comprensibles per al lector. Tot i ser una eina senzilla, Paint ha estat útil per a la preparació de les imatges que acompanyen el treball, especialment aquelles que no requerien gràfics complexos ni disseny vectorial.

A.3 Desenvolupament del programari

A.3.1 Visual Studio Code

Per escriure el codi font del robot i tot el sistema vaig utilitzar principalment l'editor Visual Studio Code, que ofereix moltes funcionalitats útils per al desenvolupament de programari. Entre les més rellevants hi ha el ressaltat de sintaxi amb colors, la detecció d'errors mitjançant extensions, la formatació automàtica del codi, el suport per a depuració i la integració amb entorns de desenvolupament com Python.

Visual Studio Code és un programa gratuït desenvolupat per Microsoft, amb una gran comunitat de plugins que amplia les seves funcionalitats segons les necessitats del projecte.

A.3.2 Git i GitHub

Git és un sistema de control de versions que permet registrar i gestionar les diferents edicions dels fitxers d'un projecte al llarg del seu desenvolupament. Aquesta eina és especialment útil per revisar els canvis efectuats o per revertir modificacions si és necessari. En projectes col·laboratius, Git facilita la integració de treballs realitzats per diverses persones en un únic repositori central. És àmpliament utilitzat en nombrosos projectes de programari, i en el meu cas l'he emprat durant el desenvolupament de tot el programari del prototip.

D'altra banda, GitHub és un servei proporcionat per Microsoft que ofereix allotjament gratuït per a repositoris que utilitzen Git. A més de conservar el codi, GitHub permet compartir-lo amb la comunitat, fent-lo accessible a projectes de codi obert com tot aquest treball. El repositori del projecte es pot consultar públicament a través del següent enllaç: <https://github.com/Geni64/robotGestorAparcaments>.

B Esquemàtics de les PCB

Esquemàtic PCB v1

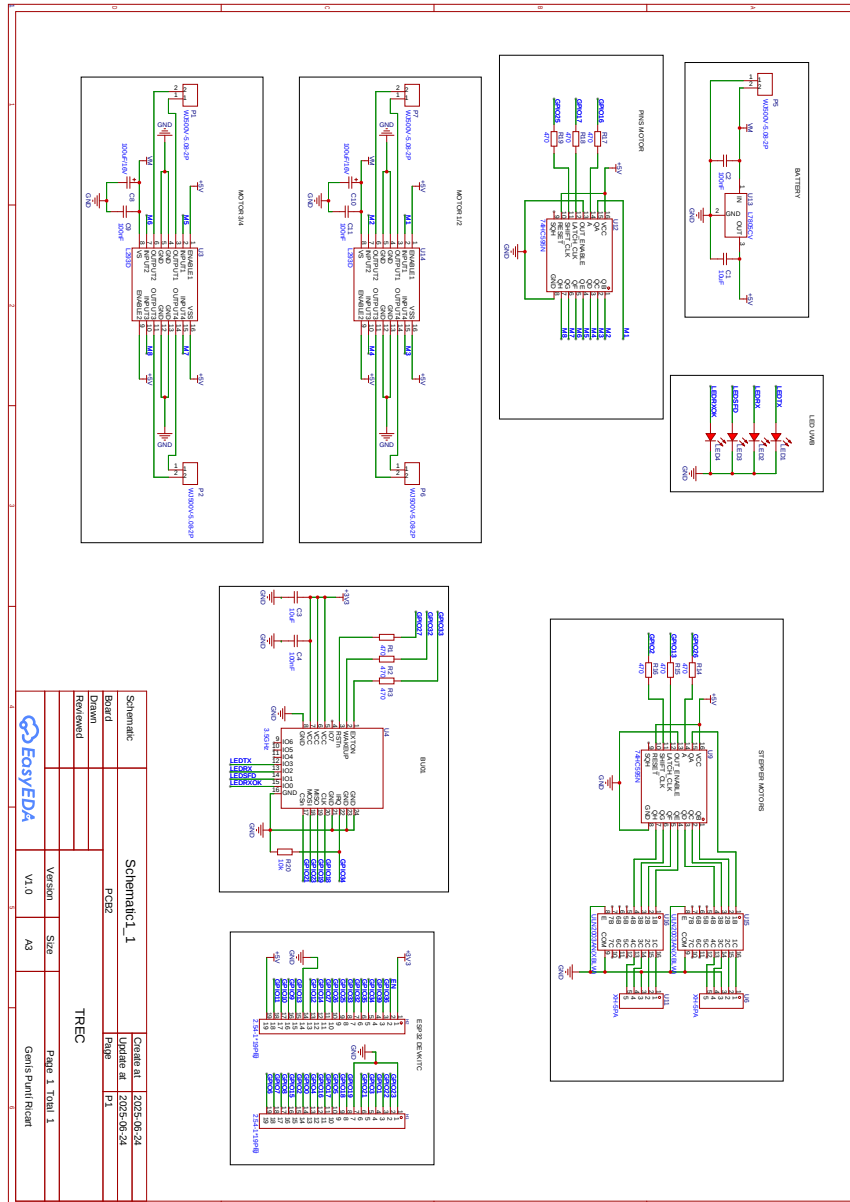


Figura 92: Esquemàtic de la placa de circuit imprès versió 1. Font: Elaboració pròpia

