



Automatització per instruccions de veu d'una cadira de rodes elèctrica i geolocalització

Marc Hernández i Cort
2n Batxillerat Grup A

Tutora: Marta Riera Fanlo
Institut Santa Eugènia
15 d'octubre de 2021

“Un investigador comença la investigació en un camp nou amb fe, una idea brumosa, i alguns experiments salvatges. Finalment, la interacció dels resultats positius i negatius guien el treball.”

-Donald Cram

Abstract

Les cadires de rodes elèctriques, per algunes persones, són de molta utilitat i proporcionen llibertat a l'usuari però, s'ha de tenir en compte que sense la tecnologia res d'això hagués estat possible.

Els objectius del treball són per una part, substituir el comandament manual d'una cadira de rodes elèctrica per un sistema de comandament per veu i per altra part, geolocalitzar la cadira.

En tractar-se d'un projecte tecnològic, on la construcció és una de les parts essencials del treball, s'utilitza la metodologia pròpia de la tecnologia que és el procés tecnològic. Per tant, no es formulen hipòtesis sinó que es formulen uns requirements.

Els resultats de l'automatització de la cadira de rodes per veu no ha estat tan satisfactòria com s'esperava, no per la part mecànica, que ha funcionat amb una efectivitat del 100% sinó que el problema ha estat amb el mòdul de reconeixement de veu que s'ha mostrat irregular. Si bé assoleix un 65% d'encerts quan les ordres per veu es donen sense soroll de fons; quan hi ha soroll de fons, l'efectivitat cau a un 30%. Això es tradueix en poca fiabilitat del sistema a l'hora de poder ser comercialitzada.

D'altra banda, per l'altra requeriment que és la geolocalització, els resultats han estat molt fiables ja que s'ha assolit un 80% d'encerts.

Definitivament, la cadira de rodes automatitzada per veu no és fiable i els motius probables és que el micròfon que s'ha utilitzat no tenia filtres pel soroll de fons i que l'arduino UNO no té suficient capacitat per albergar aquest sistema.

Tanmateix, el sistema de geolocalització ha donat molt bons resultats i la fiabilitat és molt alta.

Finalment, tot i que sembli que aquest projecte ha anat rodat no és cert, ha tingut moltes dificultats durant el procés i en definitiva, el principal topall ha estat la limitació en el hardware.

Paraules clau: Cadira de rodes elèctrica, comandament per veu, geolocalització.

Abstract

Electric wheelchairs, for some people, are very useful and provide freedom to the user, but it must be borne in mind that without technology none of this would have been possible.

The objectives of the work are, on the one hand, to replace the manual control of an electric wheelchair with a voice control system and, on the other hand, to geolocate the chair.

As this is a technological project, where construction is one of the essential parts of the work, the technology's methodology, which is the technological process, is used. Therefore, no hypotheses are formulated but requirements are formulated.

The results of the automation of the wheelchair by voice has not been as satisfactory as expected, not for the mechanical part, which has worked with 100% effectiveness but the problem has been with the module of recognition of sees that it has been shown irregular. While it achieves 65% accuracy when voice commands are given without background noise; when there is background noise, the effectiveness drops to 30%. This translates into the unreliability of the system when it comes to marketing.

On the other hand, due to the other requirement which is geolocation, the results have been very reliable as 80% of the hits have been achieved.

The automated wheelchair by voice is unreliable and the probable reasons are that the microphone used did not have background noise filters and that the Arduino UNO does not have enough capacity to house this system.

However, the geolocation system has given very good results and the reliability is very high.

Finally, although it seems that this project has been shot is not true, it has had many difficulties during the process and ultimately, the main hurdle has been the limitation in the hardware.

Keywords: Electric wheelchair, voice control, geolocation.

INDEX

1.Introducció	6
1.1 Justificació del tema escollit.....	6
1.2 Descripció de la metodologia emprada.....	7
2. Marc teòric	8
2.1 Introducció.....	8
2.2. Cadira de rodes elèctrica.....	8
2.2.1 Introducció.....	8
2.2.2 Tipus de cadires de rodes elèctriques.....	9
2.2.3 Parts de la cadira de rodes elèctrica.....	9
2.2.4 Principi de funcionament.....	9
2.3 Dibuix assistit per ordinador (DAO): FreeCAD i Cura.....	9
2.3.1 FreeCAD.....	10
2.3.2. Cura.....	12
2.3.3 Procés de fabricació de peces amb la impressora 3D.....	13
2.4 Electrònica i electricitat.....	13
2.4.1 Arduino.....	13
2.4.2 Mòdul:ELECHOUSE Voice recognition module V3.1.....	16
2.4.3 Placa NodeMCU.....	17
2.5 Programació:	19
2.5.1 API Geolocation.....	19
2.5.2 IDE arduino	19
2.5.3 FIREBASE.....	20
2.5.3.1 Hosting Firebase.....	20
2.5.3.2 Base de dades FIREBASE	21
2.5.3.2 Base de dades SQL FIREBASE	21
2.5.3.2.1 Base de dades noSQL FIREBASE	22
2.5.4 APP inventor 2.....	23
3. Desenvolupament del Projecte	25
3.1 Descripció del problema i requeriments.....	25
3.2 Disseny	25
3.2.1 Modificació del comandament manual.....	25

3.2.2 Mòdul del control direccional per veu.....	28
3.2.3 Mòdul de Geocalització.....	31
3.2.3.1 Elecció de la placa.....	32
3.2.3.2 Obtenir les credencial API Geolocation.....	33
3.2.3.3 NodeMCU:conexio i obtenció de la latitud i la longitud.....	33
3.2.3.4 Firebase Hosting i Realtime Database.....	36
3.2.3.5 Emmagatzematge a Firebase Database desde el nodeMCU.....	39
3.2.3.6 Lectura de dades de la base d'una una pàgina web.....	39
3.2.3.7 Situació de les coordenades sobre un mapa interactiu.....	40
3.2.3.8 App locachair.....	40
3.3 Planificació.....	41
3.3.1 Pressupost.....	41
3.4 Realització.....	43
3.4.1 Construcció.....	43
3.4.2 Programació.....	44
3.4.2.1 Control de la cadira per veu.....	44
3.4.2.2 NodeMCU.....	49
3.4.2.3 Pàgina web, visualització de latitud i longitud en temps real.....	53
5. Avaluació i conclusions.....	56
6 Agraïments.....	57
7. Webgrafia.....	58

1. Introducció

Portar a terme un treball de recerca suposa feina i moltes hores de dedicació. És evident, doncs, que amb aquests supòsits valgui la pena triar un tema que, a més d'obligar-te a buscar informació, seleccionar-la i donar-li forma, t'aporti nous coneixements, et permeti un enriquiment personal, t'obligui a superar dificultats i a trobar solucions als problemes que es poden presentar mentre s'està desenvolupant el treball.

1.1 Justificació del tema escollit

El títol del meu treball, com heu pogut veure, s'anomena "Automatització per instruccions de veu d'una cadira de rodes elèctrica i geolocalització", però trobar el treball i el títol adequat no va ser fàcil.

El projecte inicial que volia desenvolupar estava relacionat amb la COVID-19, la idea era elaborar un sistema per automatitzar les obertures de les finestres de les classes de l'institut a través de la sensòrica per concentració CO₂. A l'estiu de 2021 vaig fer recerca sobre el tema però, a mitjans d'estiu vaig observar que el ritme de vacunació s'havia disparat i per tant, començava remetre la pandèmia i semblava que perdia sentit aquest treball inicial. Això, i que a principis d'agost, una de les meves àvies que té mobilitat reduïda, va venir de vacances amb nosaltres. Té les articulacions afectades, tant de les mans com la dels peus, però pot parlar perfectament. Per tant, vaig decidir implementar l'automatització d'una cadira de rodes elèctrica per instruccions de veu. D'altra banda, vaig observar que de nit li costava una mica orientar-se i per tant, la cadira hauria de tenir un sistema de localització per poder monitoritzar-la a través d'una aplicació.

L'objectiu d'aquest treball és beneficiar persones, com la meva àvia, que tinguin una discapacitat però que puguin comunicar-se bé amb la veu.

En aquell moment se'm va encendre la llumeta i vaig preguntar-me quatre coses:

- Podem interactuar sobre el mòdul de comandament per joystick d'una cadira de rodes elèctrica i modificar-lo per què obeeixi les ordres per veu?
- Hi ha algun mòdul per poder gravar la veu?
- Es podria universalitzar? O sigui, el projecte serviria per qualsevol motor i cadira del mercat?
- Es podria Geolocalitzar i situar en un mapa amb una aplicació mòbil o web?

Finalment, després d'uns dies donant-li voltes, va haver-hi dos fets que, junts, van animar-me a continuar treballant la solució:

D'una banda, contactar amb una associació anomenada MultiCapacitats. Aquesta associació integra i dinamitza molts projectes joves i engrescadors, i el seu principal objectiu és millorar la qualitat de vida de les persones amb diverses discapacitats. Aquesta associació em va donar l'oportunitat d'observar el funcionament de dos tipus de cadires de rodes. La primera era un motor que s'acopla a qualsevol cadira de rodes, és a dir, universal.

D'altra banda, aconseguir el préstec d'una cadira de l'escola d'educació especial Palau de Girona i gràcies al director Miquel Barris Ruset.

En conclusió, si pogués canviar el comandament manual i substituir-lo per un comandament de veu, o bé que hi hagués la possibilitat de treballar paral·lelament.

En tractar-se d'un projecte tecnològic, on la construcció és una de les parts essencials del treball, i donat que, utilitzaré la metodologia del procés tecnològic per portar-lo a terme; les hipòtesis del treball les formulo a partir dels requeriments referenciats a l'apartat 3.1.

1.2 Descripció de la metodologia emprada

Per desenvolupar el treball utilitzaré l'eina pròpia de la tecnologia, que és el procés tecnològic.

Les fases del procés tecnològic són:

- **Plantejament del problema:** principalment consisteix en plantejar-te un problema observat, i el tipus de condicions que hagi de complir.
- **Recerca d'Informació:** un cop pensat el problema, s'ha de buscar informació a tots els mitjans possibles: internet, llibres, parlar amb experts en l'àmbit que es està treballant...
- **Disseny:** abans de començar a pensar en els materials amb què es construirà l'objecte o disseny, s'han de crear els plànols
- **Planificació:** un cop s'ha escollit la idea i s'han dissenyat els plànols, s'han d'organitzar, indicar els materials i les eines utilitzades, s'ha de portar un ordre en les etapes que s'han de seguir, calcular el pressupost, entre moltes altres coses.
- **Realització:** s'ha de seguir pas a pas la planificació, anotant les incidències i problemes que hi pugui sorgir i altres coses relacionades amb el disseny, programació, creació i resolució del problema de l'objecte.

- **Avaluació i conclusions:** després de l'etapa anterior, cal experimentar si l'objecte creat compleix la seva funció. Ens podem trobar dos casos: que l'objecte funcioni adequadament (anirem directament a la següent etapa), o que l'objecte no funcioni (es tornaran a revisar els punts anteriors per veure on existeix l'error).
- **Divulgació:** l'objectiu de la tecnologia consisteix en satisfer i facilitar les necessitats de l'ésser humà.
Si es fabrica un objecte o giny que ho compleix, és lògic difondre'l perquè la societat en tregui benefici. Per això fem ús de la "Memòria Tècnica" i la publicació en premsa i revistes especialitzades.

Durant qualssevol de les fases que formen el procés és molt possible que apareguin dubtes o inconvenients que s'hauran de solucionar tornant a la fase de recerca d'informació.

2. Marc teòric

2.1 Introducció

Aquest projecte requereix l'especialització en alguns continguts de dos àmbits de coneixement, el científic i el tecnològic, d'aquí la seva interdisciplinarietat i la seva complexitat.

Concretament hi ha cinc camps de coneixement sobre els quals hem de fer recerca i aplicar-ne els coneixements al treball.

- Impressió 3D
- Dibuix assistit per ordinador (DAO): Freecad i CURA
- Electrònica i electricitat
- Programació: API Geolocation, IDE arduino, Firebase i APP inventor

2.2 Cadira de rodes elèctrica

2.2.1 Introducció

Una cadira de rodes elèctrica és una màquina que serveix per facilitar i millorar la qualitat de vida a persones amb problemes de mobilitat.

2.2.2 Tipus de cadires de rodes elèctriques

Actualment hi ha molts tipus de models comercials i es poden classificar de maneres diferents però, la manera més acceptada és la classificació proposada per la Norma ISO 9999. Es divideix en propulsió manual (ISO 9999 12.22) i de propulsió elèctrica (ISO 9999 12.23) i després es fa una subclassificació en detall.

Al nostre cas farem l'estudi amb la següent subclassificació de cadira elèctrica:

- De tracció al davant.
- De tracció al darrera
- Cadira de rodes manual amb acoblament de motor auxiliar.

2.2.3 Parts de la cadira de rodes elèctrica:

L'estructura és bàsicament de metall que acomoda una cadira per seure i sota d'aquest seient té dues bateries de 12 V connectades en sèrie per tant, treballa a 24 V. Dos motors de corrent contínua amb reductor d'engranatges connectats directament a la roda motriu de la cadira.

D'altra banda, tenim la placa controladora que es basa en una part de circuiteria de potència i una altra de control. La placa de control detecta el moviment del joystick i emet el senyal de sortida adequat cap als motors a través del circuit de potència.

2.2.4 Principi de funcionament

Es tracta de descomposar el moviment bidimensional dels dos eixos de coordenades clàssics perpendiculars entre ells (x, y) per tal que a través del Joystick es puguin governar aquestes direccions i sentits. El moviment d'aquests eixos coordinats fa que es mogui la cadira en els moviments (davant, darrera, dreta i esquerra).

2.3 DAO: Dibuix Assistit per Ordinador

Utilitzarem la impressora 3D per elaborar elements auxiliars que incorporarem al projecte. El modelatge de les peces el realitzarem amb el següent programari:

2.3.1 FreeCAD

El FreeCAD és un modelador CAD 3D paramètric de software lliure, orientat al disseny industrial, l'enginyeria i l'arquitectura. En el meu projecte s'han de dissenyar alguns elements a mida com per exemple una carcassa que tancarà el Joystick de la cadira de rodes i la diadema on tindrem situat el micròfon per l'usuari.

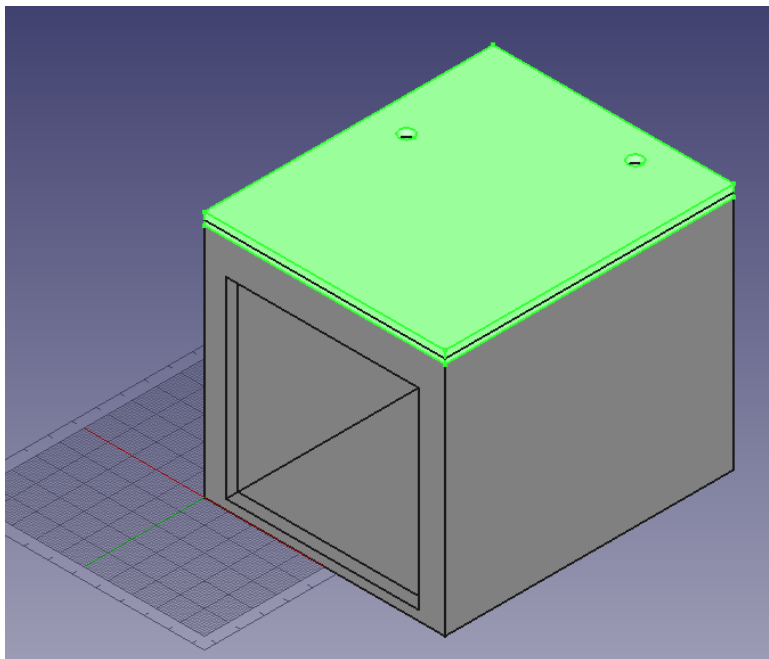


Fig: Disseny de la carcassa per el comandament de la cadira

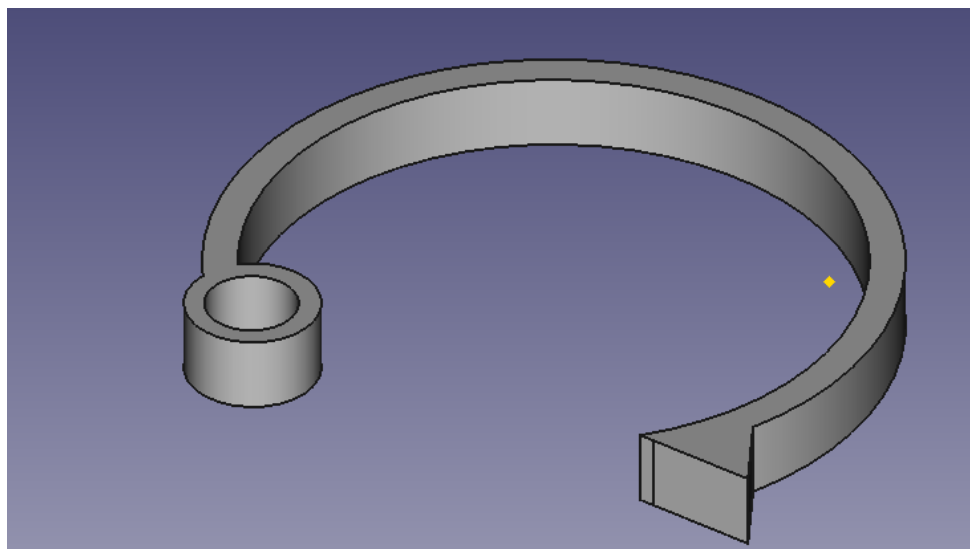
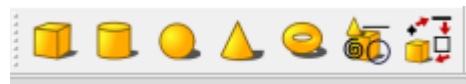


Fig: Disseny de diadema

Per realitzar els dissenys anteriors he necessitat conèixer algunes eines bàsiques del programa FreeCAD que esmento a continuació de manera simplificada:

- ❖ Utilitzar la modalitat “**Part**” per insertar figures bàsiques 3d sòlides (prismes, cilindres, piràmides, cons,...)



- ❖ Operacions boleanes entre dues formes: tall i unió de figures tridimensionals (escollir una figura i amb el control clicat escollir l'altra figura i procedir amb el tall o unió)





- ❖ Amagar les figures o tornar a mostrar clicant l'espai.

- ❖ Utilitzar la modalitat “**part design**” i  crear nou croquis.

- ❖ El croquis pot ser sobre plànols naturals o sobre una de les cares d'una figura.

- ❖ 

- ❖ Dibuixos 2D sobre cares o plànols que extrudim per transformar-les en 3D o bé realitzem un “pocket” per retallar sobre un figura 3D. 

- ❖ Cossos de revolució:  realitzem el croquis i revolucionem en un dels eixos.

Els arxius resultants s'exporten amb stl (stereolithography); és un format estàndard per a tecnologies de fabricació per addició que utilitza una malla de triangles tancats per definir l'objecte. Com més resolució menys es veuen les irregularitats a la impressió.

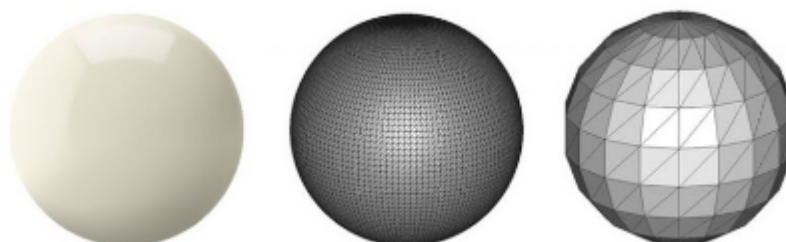


Fig: Objectes stl d'esquerra a dreta de més resolució a menys

2.3.2 Cura

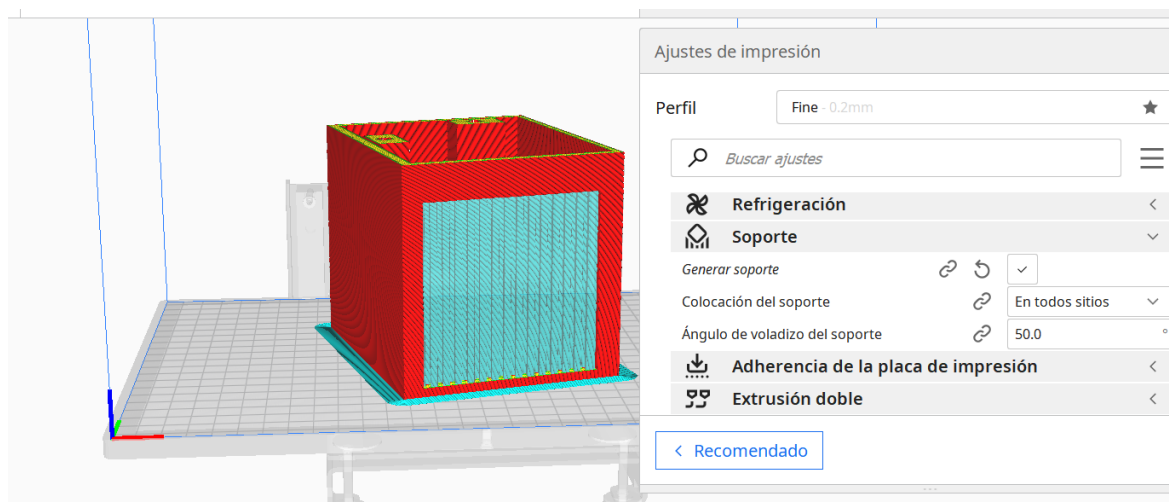


Fig: Ultimaker Cura

La funció del programa CURA és preparar el model que surt de FreeCAD perquè sigui reconegut per la impressora.

Aquest programa és un programa de laminació. Dit d'un altra manera, es trosseja la figura seguint l'eix vertical en fines làmines que són les capes de material plàstic que dipositarà la nostra impressora per l'extrusor. Per tant, aquest procés és l'encarregat de transformar l'arxiu stl a un arxiu gcode que és el propi de control numèric (CNC) i l'únic que reconeix la impressora.



D'altra banda, el programa CURA té infinitat de propietats que podem definir, però, al nostre cas hem deixat les opcions per defecte. L'únic que hem afegit és l'opció de **generar suport**. Aquesta opció crea unes estructures auxiliars que fan possible la impressió dels voladissos dels suports de motors i l'obertura frontal de la carcassa del comandament.

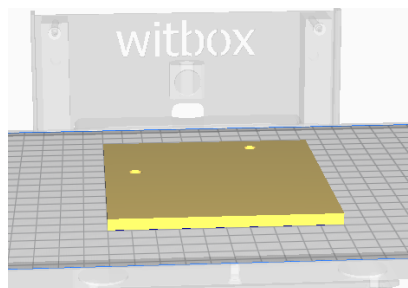


Fig: Visió de la tapa de la carcassa del comandament

2.3.3 Procés de fabricació de peces amb l'impressora 3D.

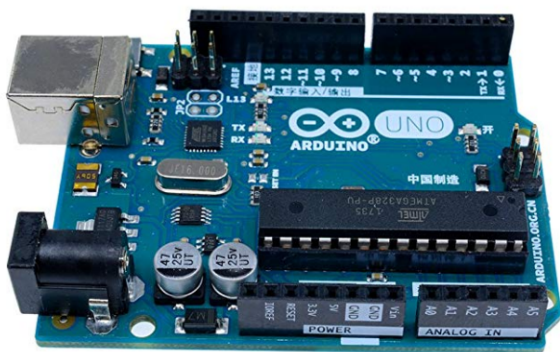
1- Hi ha contenidors web amb figures fetes que podem baixar directament com per exemple thingivers.com, o bé amb podem utilitzar el programa FreeCAD(local) o tinkercad(on-line) que fem el disseny de la peça; ens la desa amb el format .FCStd i l'exportem en format .stl.

2- Amb el programa CURA passem del format .stl a gcode. Gcode és l'extensió de de control numèric CNC que és l'únic llenguatge que entén la impressora i és un seguit d'instruccions de trajectòria que el que fan és laminar la nostra peça.

Dins el programa Cura, si volem multiplicar objectes i imprimir-ne més d'un a la vegada, escrivim el número que en volem. També podem moure les peces distribuint-les sobre la superfície al nostre gust.

2.4 Electrònica i electricitat

2.4.1 Arduino:



L'*Arduino* és un autòmat del tipus microcontrolador que té un Circuit Integrat Digital d'alt nivell d'integració de components.

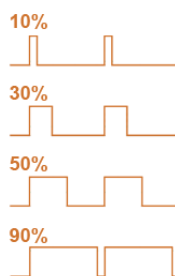
Un autòmat programable és un accessori que disposa, bàsicament, d'unes entrades, unes sortides, que fan que es comuniqui amb l'exterior, i un xip que és programable. El seu funcionament, és

bàsicament, la gestió de l'estat de les sortides depenent de les condicions de les entrades d'acord amb el programa introduït al xip.

Parts:

- La unitat central de processament o CPU (Central Processing Unit) és el principal component de l'*Arduino*, es tracta del xip ATmega328 que s'encarrega de processar les dades un cop a interpretat les instruccions dels programes.
- El cristall de quars és l'element que genera el senyal de "clock" del sistema de 16 Mhz.

- Totes les entrades mesuren tensions, de fet, l'*Arduino* només mesura tensions.
- Les entrades i sortides digitals tenen dos possibles estats, 1 o 0; són del tipus "tot i res". També podem dir que estan activades o desactivades, dit d'una altra manera, si hi ha voltatge o no.
- Les entrades analògiques mesuren tensions d'entrada dins un rang de 0 a 5V amb una resolució de 10 bits, el que suposa poder llegir el senyal en 1024 nivells de tensió amb una precisió de 2,44 mV.
- Les sortides analògiques són del tipus PWN (Pulse Width Modulation), és a dir, no són valors reals constants sinó que el que es fa és enviar un pols d'un temps determinat cada cicle de temps. D'aquesta manera, si el pols és repetitiu i perllongat en el temps, la sortida analògica tindrà un valor elevat, perquè la mitjana de valors de tensió al cicle serà elevat, i si és poc repetitiu al llarg del temps correspondrà a un valor baix.



- Memòria EEPROM: emmagatzema el programa de manera permanent tot i que hi hagi talls d'alimentació.
- Memòria RAM: guarda els valors dades provisionals quan el sistema està alimentat i en funcionament.

He escollit la placa *Arduino UNO*: té les característiques suficients per tal de que funcioni el sistema correctament.

Microcontrolador	Atmega168
Voltatge operatiu	5 volts
Voltatge d'entrada (recomanat)	7-12 volts
Voltatge d'entrada (límit)	6-20 volts
Pins d'entrada/sortida digital	14 (6 proporcionen PWM)
Pins d'entrada analògics	6
DC pels pins d'entrada/sortida	40mA
Memòria Flash	16KB (2KB pel bootloader)
SRAM	1KB
EEPROM	512 bytes
Velocitat de rellotge	16 Mhz

Característiques *ARDUINO-UNO* font: www.wikipedia.org

Així doncs, connectem els diferents sensors - entrades (NTC,...) i els actuadors a les sortides (servomotors, leds, mòdul de veu,...)

D'altra banda, hi ha un actuator que farà servir per al projecte, el servomotor. És un element que està format per un motor elèctric, una reducció per engranatges i una part electrònica que mesura l'angle de gir de l'aspa.

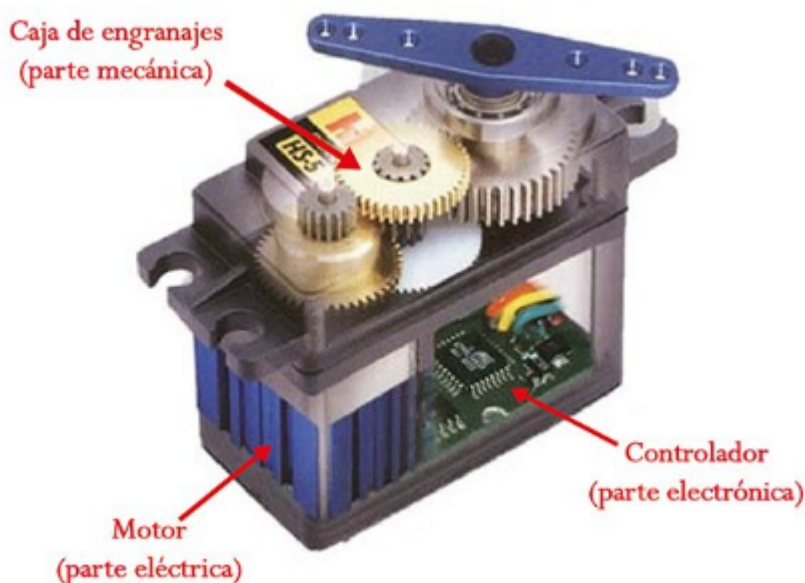


Fig. Servomotor

El servomotor permet controlar la posició de l'eix entre un angle d'entre 0 i 180 i mantenir-lo rígid en aquesta posició.

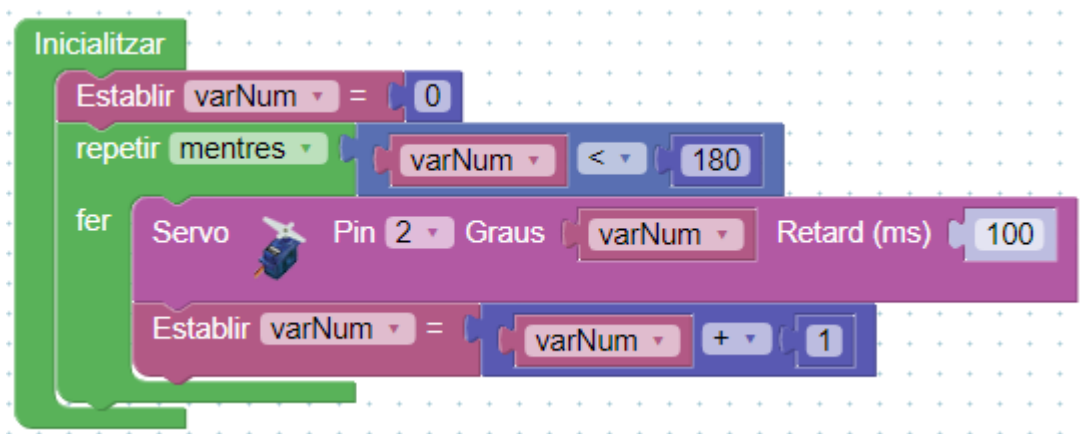


Fig. Exemple del gir progressiu entre 0 i 179° amb una interrupció entre angle i angle de 200 milisegons.

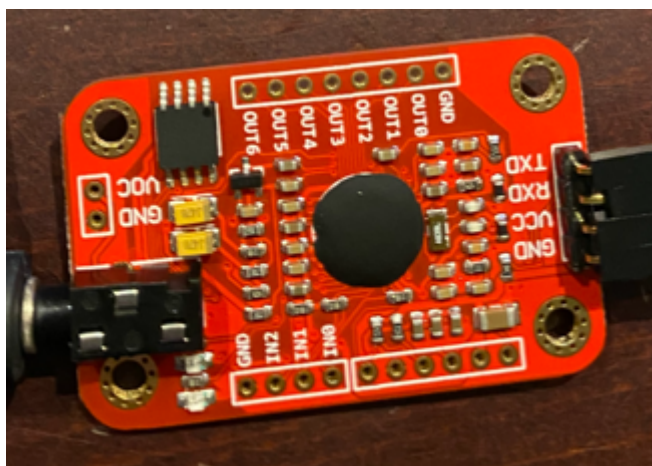
2.4.2 Mòdul : ELECHOUSE Voice recognition module V3.1

El mòdul de reconeixement de veu ELECHOUSE és un producte que emmagatzema paraules a través de l'entrenament per veu.

Admet emmagatzemar fins 80 comandes de veu en total i obeir les ordres d'un màxim de set d'aquestes a la vegada.

El control de la placa es pot fer de dues maneres:

- Port sèrie
- Programant directament a través dels pins d'entrada generals.



M'he decantat per l'opció de programació pel port sèrie, per què et permet monitoritzar dades i és més simple la transmissió al estar connectat a l'arduino.

El muntatge és molt simple i es tracta de connectar, com es pot apreciar a la figura, les sortides TXR i RXD del mòdul de reconeixement de veu amb els pins 3 i 2 de l'arduino respectivament. i el VCC i GND a 5 volts i massa respectivament:

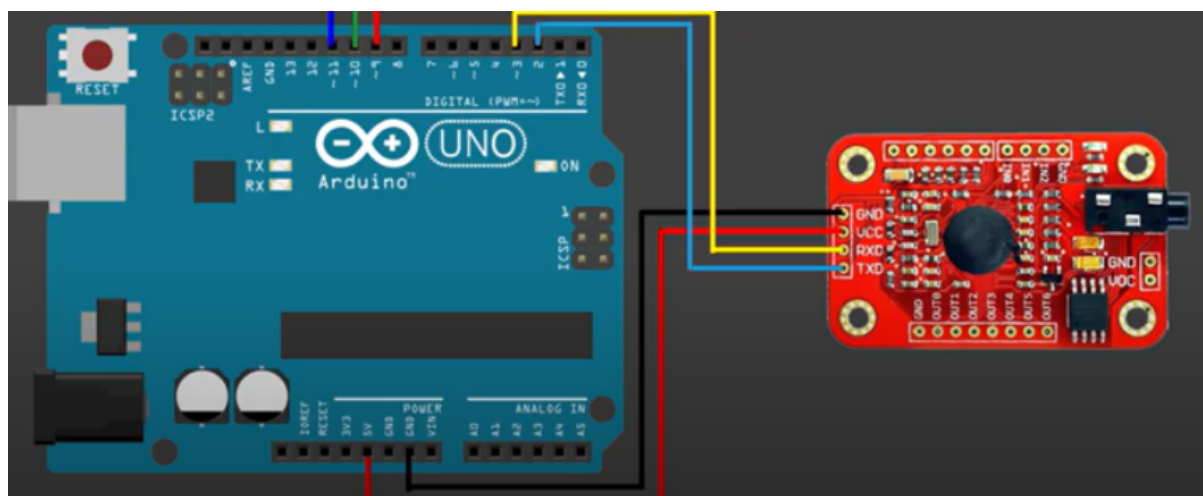


Fig: Esquema del muntatge

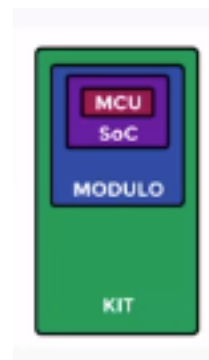
2.4.3 Mòdul: Placa NodeMCU v2 per l'accés a internet



El nodeMCU és una plataforma de desenvolupament semblant a la d'Arduino orientada a l'Internet de les coses (IoT Internet of things).

Conceptes preliminars:

- **SoM:** Sistema en mòduls: és l'apilament de varis components.
- **SoC:** sistema en un xip o el que és el mateix un sistema integrat amb múltiples components.
- **MCU:** És la unitat de control d'entrades, sortides i on s'emmagatzema i s'interpreta el programa propi de control.
- **Kit de desenvolupament:** afegeix nous components

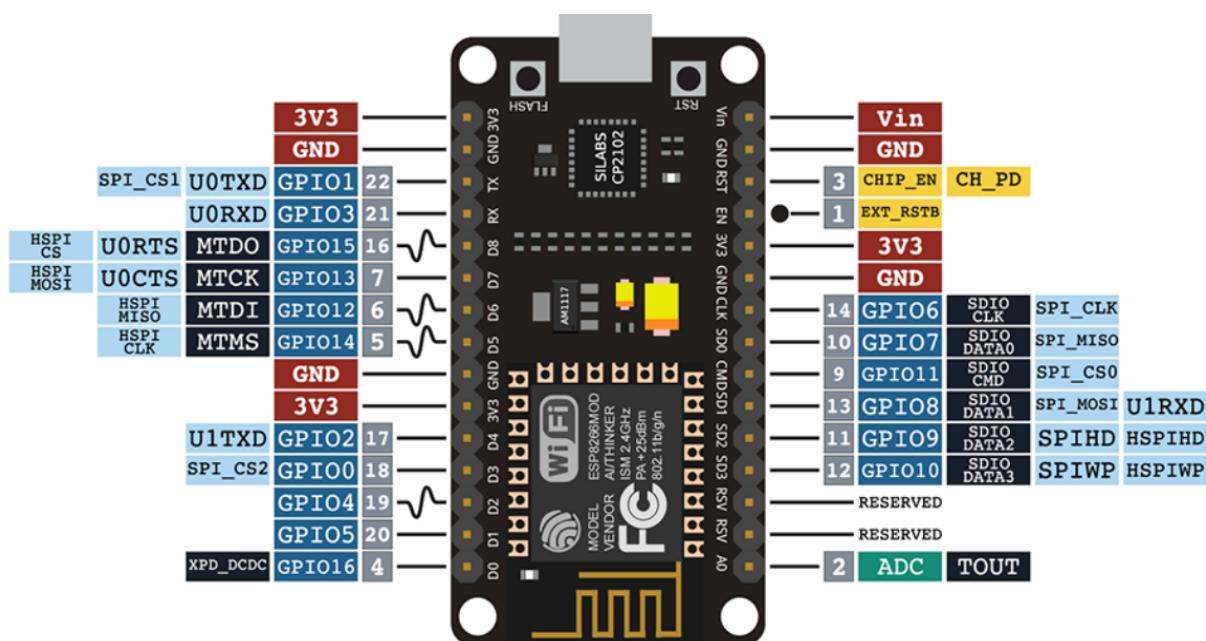


Per tant, la placa nodeMCU és bàsicament un autòmat amb arquitectura tipus SoM. Aquest sistema en mòduls pot ser ESP-12E o ESP-12F i consta de el SoC ESP8266, memòria FLASH, cristall oscilador per les funcions de rellotge i antena



WiFi.

- Mòdul SoC Wifi anomenat ESP8266 és un xip integrat com un miniordinador que podríem programar i accedir als seus pins d'entrada i sortida. Aquest és especial perquè permet accedir a internet i té els següents components a dins:
 - Microcontrolador (MCU) integrat al SoC. És el tensílica L106 de 32 Bits i velocitat de rellotge entre 80 i 120 MHz.
 - Wifi a 2,4 GHz compatible amb qualsevol router.
 - Memòria RAM de 50 kB.
 - Entrada analògica al pin 10 (ADC-Analògic Digital Converter)
 - 17 GPIO (Pins d'entrada sortida de propòsit general).
- El Kit o placa de desenvolupament NodeMCU és el nivell superior on hi ha els components que ens ajuden a programar i connectar el mòdul als nostres circuits.
 - Conversor Sèrie-USB per poder programar y alimentar a través d'USB. (TTL CP2102)
 - Fàcil accés als pins.
 - LEDs per indicar l'estat.
 - Botó de reset.
 - Botó de reset per la memòria Flash.

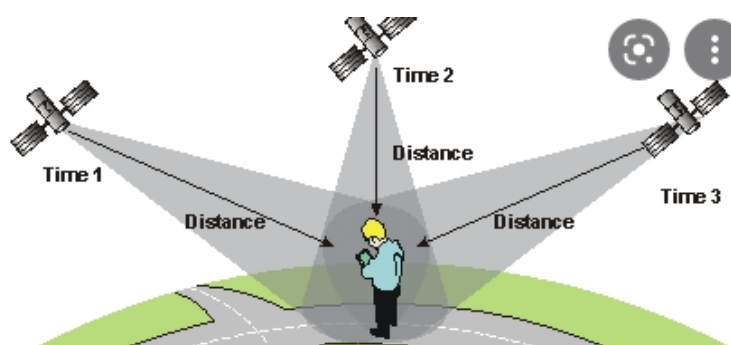


2.5 Programació: API Geolocation, IDE arduino, Base de dades Firebase, Hosting firebase i APP inventor

2.5.1 API Geolocation

Els sistema de geolocalització terrestre es pot fer a través de:

1. GPS(sistema de posicionament global) i es basa en trobar les coordenades mitjançant la triangulació de la posició entre tres satèl·lits diferents.



2. Geolocalització via WiFi es fa mitjançant una connexió a internet i amb el connector Google API Geolocation. Aprofita les antenes de telefonia i els WiFi's de la zona per obtenir les coordenades. Aquest sistema és menys precís que l'anterior però a les proves realitzades, m'ha situat molt bé la cadira de rodes..

2.5.2 IDE arduino.

L'Arduino IDE és un software que permet connexió amb la placa arduino.

Les sigles IDE volen dir entorn desenvolupament integrat i concretament conté l'editor de textos i el compilador per programar i transferir el contingut de les instruccions a la placa arduino amb codi màquina.

El llenguatge de programació utilitzat és l'anomenat Processing.



2.5.3 FIREBASE:

Firebase és una plataforma integral de desenvolupament d'APP's per qualsevol plataforma (IOS, Android, Web, Unity, etc) que integra hosting o allotjament web gratuït, mètode d'autenticació d'usuaris, emmagatzematge d'arxius (storage) i Base de dades .

Tot això ho trobem de manera gratuïta amb qualsevol compte de correu gmail.

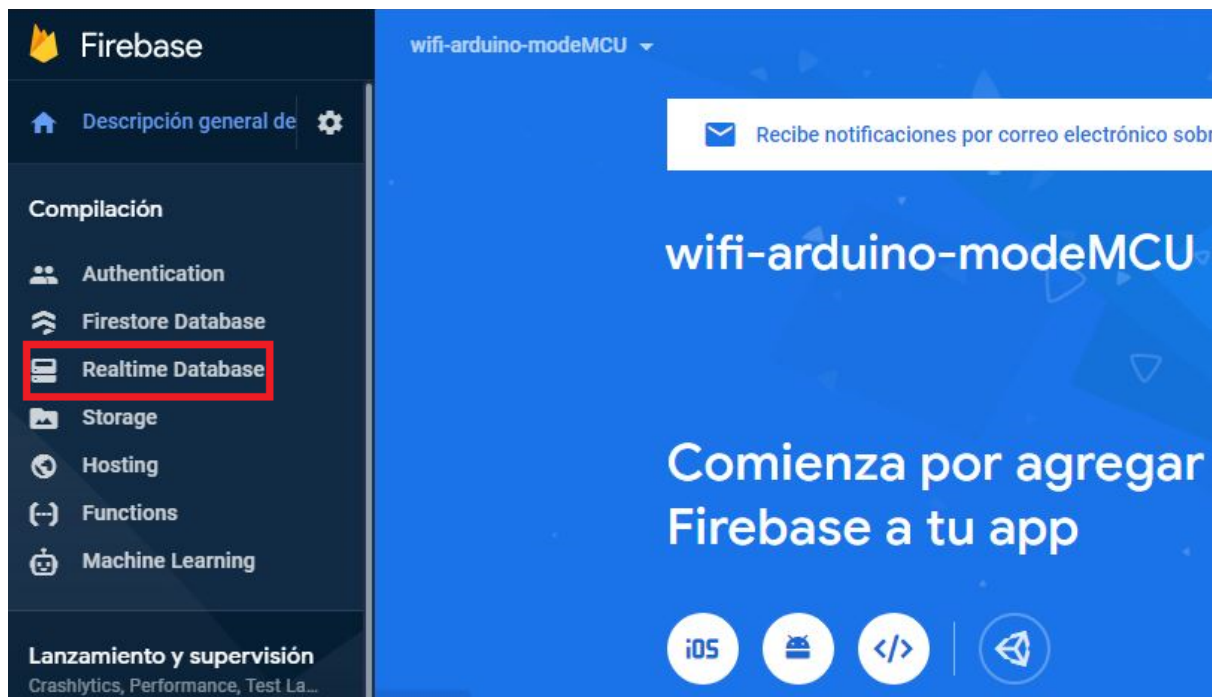
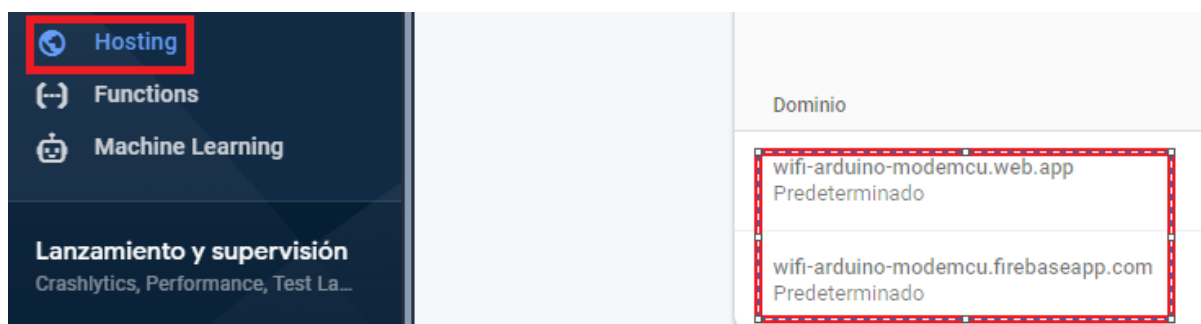


Fig. Consola d'administració (www.firebase.com)

Les dues funcionalitats que farem servir pel nostre projecte serà el hosting i la base de dades.

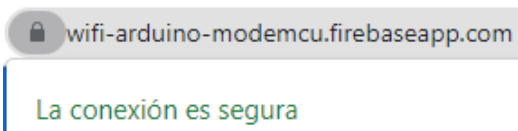
2.5.3.1 Hosting

Un Hosting és un servidor web que proporciona dos dominis, o sigui, dues adreces web.



L'implementació d'aquest sistema és una mica complicat però, una vegada realitzat té algunes avantatges importants que val la pena aprofitar i que les cito a continuació:

- Es programa en local amb eines d'edició de text com el Notepad++ i es visualitzen els canvis gairebé a l'instant i directament a l'adreça web.
- El producte el va adquirir i desenvolupar l'empresa americana Google que assegura en les seves instruccions que compleix tots els protocols de seguretat via web.
- Es pot substituir el nom del domini per un de propi.

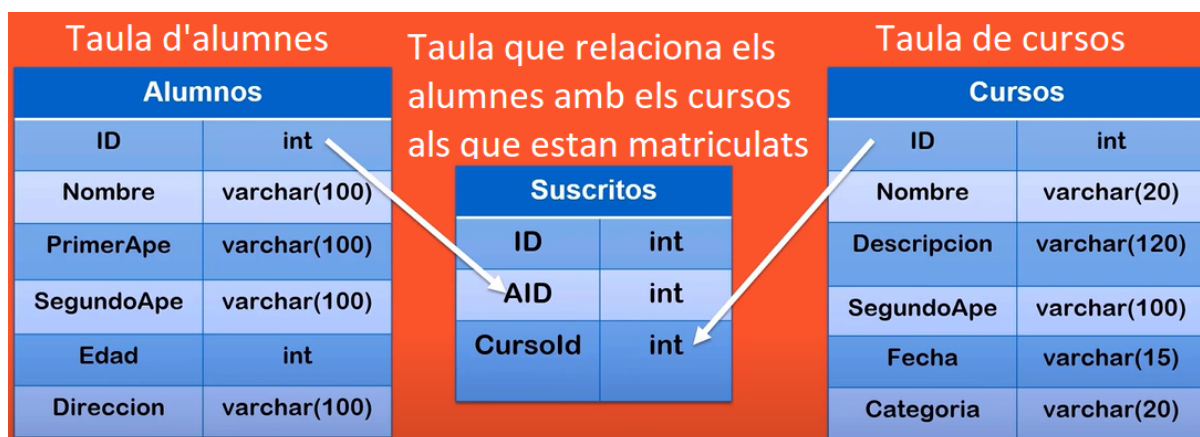


2.5.3.2 Base de dades

Al món de les bases de dades trobem les clàssiques que s'anomenen schema, relacionals o SQL i les no relacionals, no schema o noSQL.

2.5.3.2.1 Base de dades SQL

Les bases de dades SQL es basen en la relació entre diverses taules amb dades que es relacionen entre sí a través d'una altra taula. Es gestiona a través del llenguatge de consulta estructurat SQL (Structured Query Language).



Alguns tipus de base de dades tipus SQL són Oracle, MySQL, Microsoft SQL Server, Access, Ingres, etc. Aquestes són adequades quan el volum de dades creix però no de manera incontrolada, quan no necessiten més d'un servidor i quan no hi ha previst un pic d'accessos d'usuaris a l'hora.

El problema d'aquest sistema de base de dades radica en que un canvi en l'estructura d'una dada pot desestabilitzar la base de dades ja que està relacionada a través de les taules.

2.5.3.2.1 Base de dades noSQL: Firebase

La base de dades Firebase no estan sotmeses a una estructura tancada i es poden gestionar sense necessitat de tenir en compte com es relacionen les dades entre si.

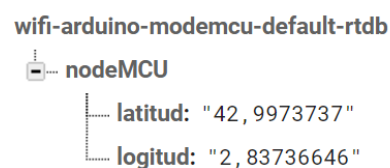
El format escollit és un format lleuger i de fàcil lectura i escriptura en forma d'arbre anomenat Json (JavaScript Object Notation) també anomenat sistema clau:valor, és a dir, s'assigna una clau amb un valor relacionat.

Exemple estructura de dades Json:

<pre>{ "Quantitat_alumnes":2, "Classe":"2nBatA", "Modalitat": "Tecnologia", "Alumnes": [{ "nom": "Pedro", "cognom": "Fernández" }, { "nom": "Valentina", "cognom": "Pujol" }] }</pre>	<p>Per exemple:</p> <p>la clau classe té el valor assignat de 2nBatA.</p> <p>i la clau alumnes té assignades les claus nom i cognom. Aquestes claus últimes contenen els valors de noms i cognoms de tots el alumne/as de la classe.</p>
---	--

Per tant, firebase adopta el tipus d'estructura de dades Json i té la peculiaritat de ser en temps real (Realtime Database), o sigui, que s'actualitza a cada instant. Aquesta característica la fa viable pel nostre projecte ja que hem d'anar emmagatzemant les coordenades de GPS a intervals de temps relativament curts per localitzar la cadira.

A tall d'exemple, aquesta base de dades és aconsellable en tipus d'aplicacions com WhatsApp on els missatges dels usuaris no han de estar relacionats entre ells i poden, en un

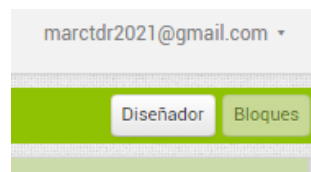


moment donat, assolir un volum de dades molt elevat sense que els servidors es colapsin.

2.5.4 MIT App inventor 2

App Inventor és un entorn de desenvolupament de software creat inicialment pel MIT (Institut Tecnològic de Massachusetts) i adquirit posteriorment per google.

El software té dues interfícies:



- Interfície de dissenyador: es realitza el disseny visual de l'APP



- Interfície de blocs: es programen les accions que volem que faci l'APP



Totes dues interfícies es programen amb el sistema drag and drop (arrossegar i deixar anar).

Exemple d'un programa que demana el nom i retorna una resposta per pantalla:

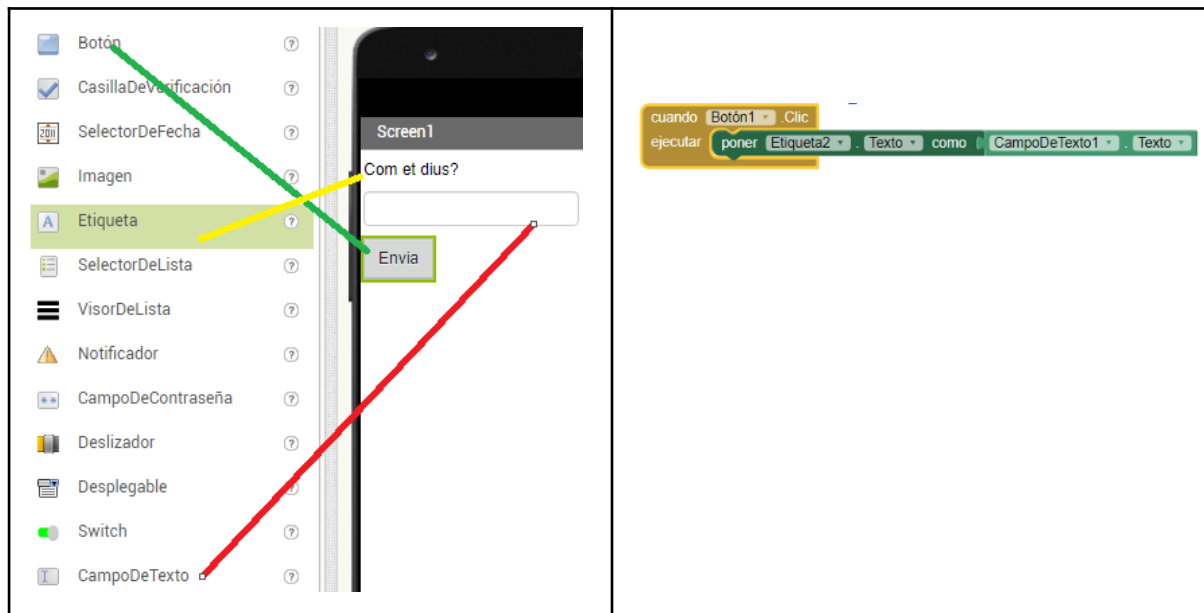
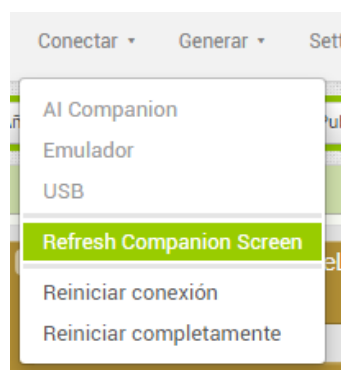
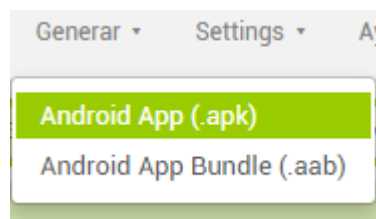


fig. Interfícies de programació: disseny i blocs



D'altra banda, APP inventor dóna la possibilitat provar els canvis a l'aplicació que estem creant en temps real. Això es realitza baixant el component AI2 Companion del l'App Store o Google Play depenent si es tracta del sistema operatiu IOS o Android respectivament .

D'altra banda, hi ha la possibilitat de generar un arxiu d'extensió apk per ser instal·lada a Android.



3. Desenvolupament del Projecte

3.1 Descripció del problema i requeriments

El problema principal del treball és substituir el comandament manual de la cadira per un comandament per veu i crear un sistema de localització i monitorització de la cadira amb aplicació de mòbil.

La màquina definitiva haurà de complir uns requeriments o unes condicions que són les següents:

1. Governar la cadira amb la veu de l'usuari.
2. Geolocalitzar la cadira.

3.2 Disseny

Definitivament, l'objectiu és treballar dos aspectes: la modificació del comandament de la cadira i la solució de geolocalització. Per tant, per dur a terme aquest projecte he hagut de dissenyar varis elements diferents amb l'impressora 3D, la carcassa per al comandament i la diadema pel micròfon.

3.2.1 Modificació del comandament manual:

Una vegada em van deixar la cadira elèctrica tenia dues possibilitats, una era realitzar una actuació no invasiva sobre el joystick, amb un sistema que l'accionés mecànicament des de l'exterior.

I l'altra, era utilitzar una metodologia invasiva, basada en desmuntar la placa de control i el comandament per intentar esbrinar quines senyals elèctriques dona el Joystick a la placa de control; per després tractar d'emular els resultats amb un microcontrolador.

Finalment, vaig decantar-me per una tècnica d'accionament extern per no haver de desmuntar el comandament.



La primera idea va ser realitzar una carcassa amb l'impressora 3D on amagar el comandament manual. El prototipus no va anar tan bé com s'esperava ja que va aparèixer un inconvenient important al collar-la i és que la força exercida pels servos provocava molts fimbresos.

Com que necessitava un material més rígid, vaig plantejar-me dues solucions, realitzar una carcasa similar a l'anterior amb fusta o realitzar un altra disseny diferent amb metall.

Una carcasa de fusta en una cadira rodes no sembla una solució molt elegant, tinguent en compte que està situada al final d'un dels recolzabraços.

Llavors, la solució adoptada per el disseny era molt més senzilla i eficaç ja que no consistia en fer una caixa, sinó que els servomotors estiguessin incorporats a dues menes de parets d'alumini (eix X,Y).

Com que aquest sistema el veia molt més a l'abast, vaig fer un predisseny fent els forats, plecs i talls amb un cartró perquè encaixes al controlador. Després, vaig traslladar aquest patró al perfil d'alumini.

Per tant, vaig escollir un passamà d'alumini de 3 cm de gruix, el vaig mecanitzar amb les següents operacions: tallar a una mida de 20 cm amb una serra de metalls, fer els forats amb ajuda d'un trepant amb broca del cinc, doblegar amb un angle de 90° amb ajuda d'un cargol de banc, un martell per metalls i una entalla al metall amb un formon per poder connectar la clavilla de connexió.



Pel disseny era important que la vora del perfil estigués alineat amb el centre del joystick del comandament.

D'altra banda, la peça que governa el desplaçament lateral dreta/esquerra sobresurt del comandament i podia ser un inconvenient si el muntatge es fa al possabraços esquerra de la cadira. Aleshores, vaig establir un requisit nou que va ser que el sistema es pogués muntar tant per dretans com per esquerrans només girant la peça i per això apareixen el quatre forats en una de les peces. Per muntar-la per esquerrans només cal fer una rotació de 180° a l'horitzontal i collar pels altres cargols.

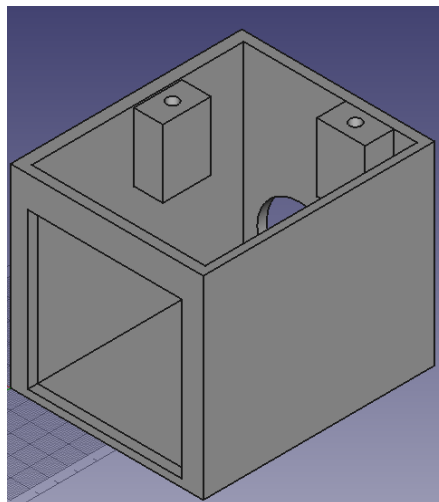




Fig. Solució de l'actuació sobre el comandament (versió dretans)

Finalment, falta la connexió dels servos amb el joystick del comandament l'he fet amb filferro de 3 mm de gruix que té prou rigidesa per impulsar el Joystick. Cal destacar, que té una petita dificultat a l'hora de fer la connexió entre el filferro i les aspes del servo ho hem fet amb una connexió tipus “ziga-zaga”, que s'utilitza en aeromodelisme.

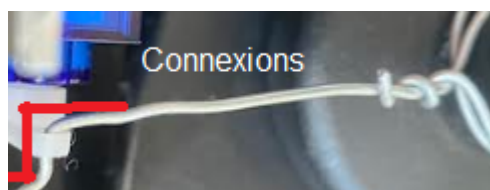
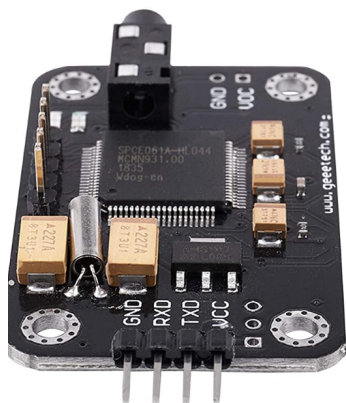


Fig. Comandament modificat

3.2.2 Mòdul de control direccional per veu



Els mòduls de control de veu són molt entretinguts i divertits d'entrenar però, el primer mòdul que vaig provar que va ser el TOOGOO voice recognition necessita un programa auxiliar anomenat Docklight V2.3, complicat d'entendre i de programar.

Aleshores, vaig fer l'adquisició de l'ELECHOUSE Voice recognition module V3 que va ser més fàcil d'entrenar perquè no requereix d'un programa auxiliar sinó que es realitza tot a través de l'arduino IDE. L'únic inconvenient és que és un 30 % més car.

Després de les primeres proves, l'objectiu va ser introduir els missatges de veu en alguns dels vuitanta canals de memòria que disposa el dispositiu. Els segments de memòria que té estan numerats del 0 al 79 i només vaig ocupar ocupar 7 amb les següents accions:

Memòries:	Acció	Instrucció
Memòria 0	Canya	Endavant ràpid
Memòria 1	Endavant	Endavant mitja velocitat
Memòria 2	lent	Endavant lent
Memòria 3	Endarrere	Endarrere
Memòria 4	Dreta	Dreta
Memòria 5	Esquerra	Esquerra
Memòria 6	Para (stop)	Frenar

La memorització d'aquest sons s'anomena entrenament, com he dit anteriorment, i és important tenir en compte que el mòdul reconeix la veu de l'usuari i que per cada persona s'hauria de fer l'entrenament inicial amb la seva veu.

Per fer l'entrenament es necessita realitzar els següents passos:

- ❖ Instal·lar la biblioteca VoiceRecognitionV3-master sobre l'arduino IDE
- ❖ Obrir l'exemple que porta la biblioteca per fer l'entrenament (vr_sample_train)
- ❖ Transferir el programa a l'arduino UNO on hem connectat pel port sèrie el mòdul ELECHOUSE voice recognition

- ❖ Visualitzar el monitor sèrie (Eines/Monitor Serie)
- ❖ Executar pel monitor sèrie la instrucció sigtrain seguit del lloc de memòria i el nom de l'acció. Per exemple, si volem memoritzar l'adreça de memòria número 78 amb l'acció aturada. teclejem sigtrain 78 aturada. El programa ens farà repetir la instrucció per veu fins que el memoritzi correctament.

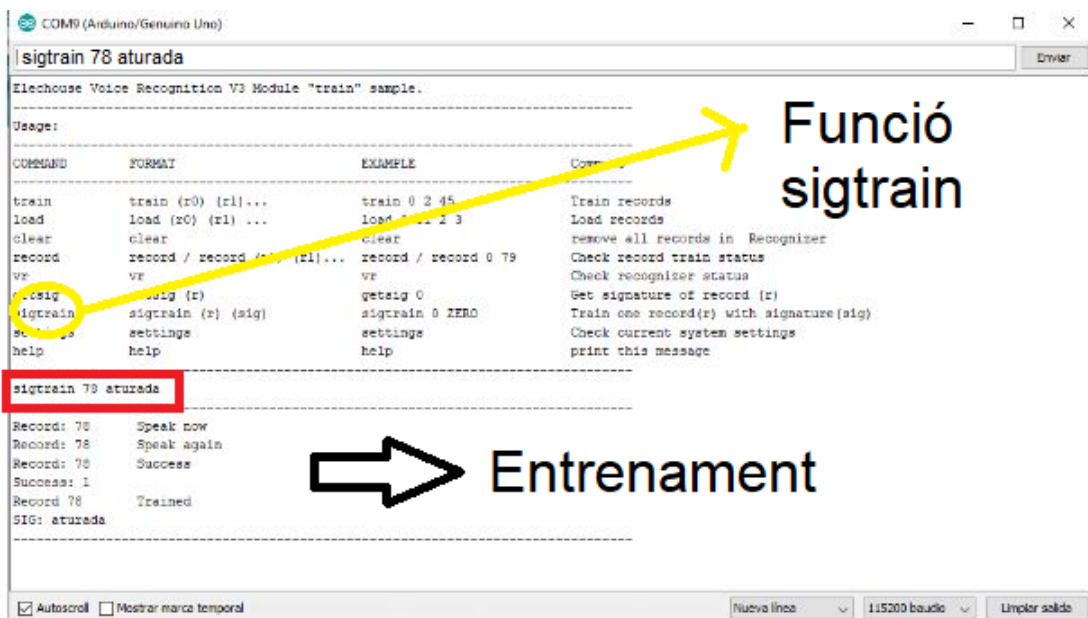


Fig: Monitor serial amb la llista d'instruccions per fer l'entrenament de veu i la memorització.

Una vegada l'ELECHOUSE està entrenat es procedeix amb la modificació de l'exemple (vr_sample_control_led) que ve amb la biblioteca VoiceRecognitionV3. En concret es fan dues modificacions:

- Monitoritzar si les memòries carreguen perfectament
- Introduir els moviments dels servomotors.

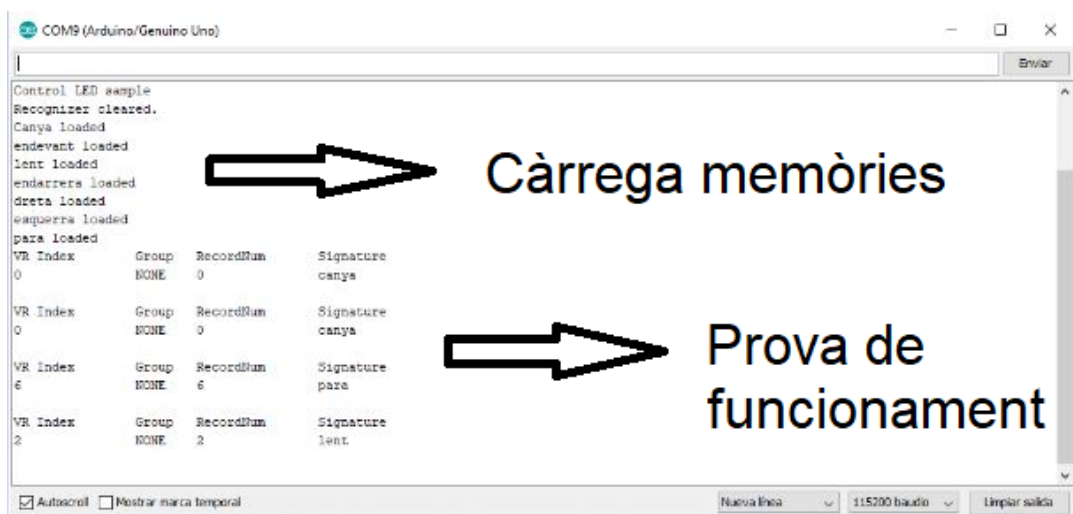


Fig. Monitorització funcionament global del programa reconeixement per veu

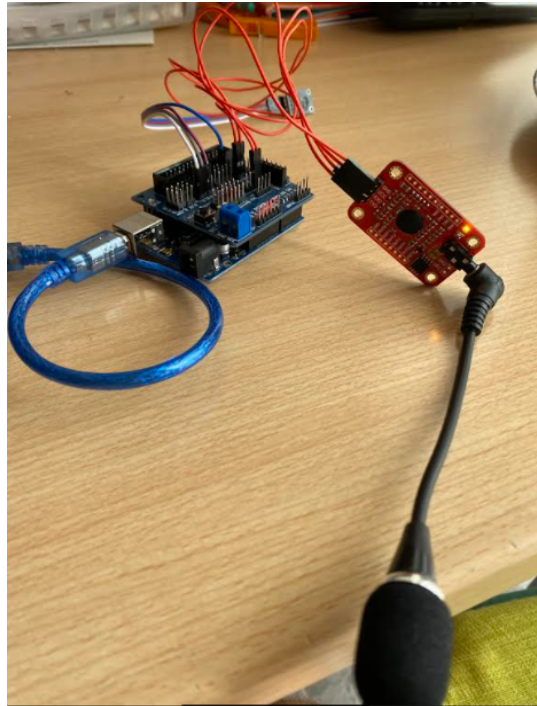


Fig. Visió global del mòdul Voice recognition V3 connectat amb l'arduino UNO

El programa de funcionament del dispositiu està desenvolupat a l'apartat programació 3.4.2.1.

3.2.3 Mòdul de Geolocalització

Abans de començar amb el desenvolupament de la solució, vull deixar clar que el l'objectiu és aconseguir la posició de la cadira en temps real. Aquesta posició es mostra en un mapa interactiu via web i/o APP per telèfon mòbil:



Fig. Web i APP de la localització de la cadira en temps real.

La primera dificultat va ser planificar i organitzar la seqüència ordenada d'accions que, van dur-se a terme per assolir l'objectiu. Llavors, la llista ordenada és la següent:

1. Es selecciona la placa o mòdul on es desenvolupa el projecte
2. S'obté les credencial API google Geolocation
3. Es realitza la connexió a internet del mòdul
4. Es realitza l'extracció via API Geolocation de les coordenades de la cadira (latitud i longitud)
5. S'envien les coordenades a Firebase Database
6. S'obté i es visualitza en un mapa via web les coordenades
7. Es crea una APP per a telefonia mòbil

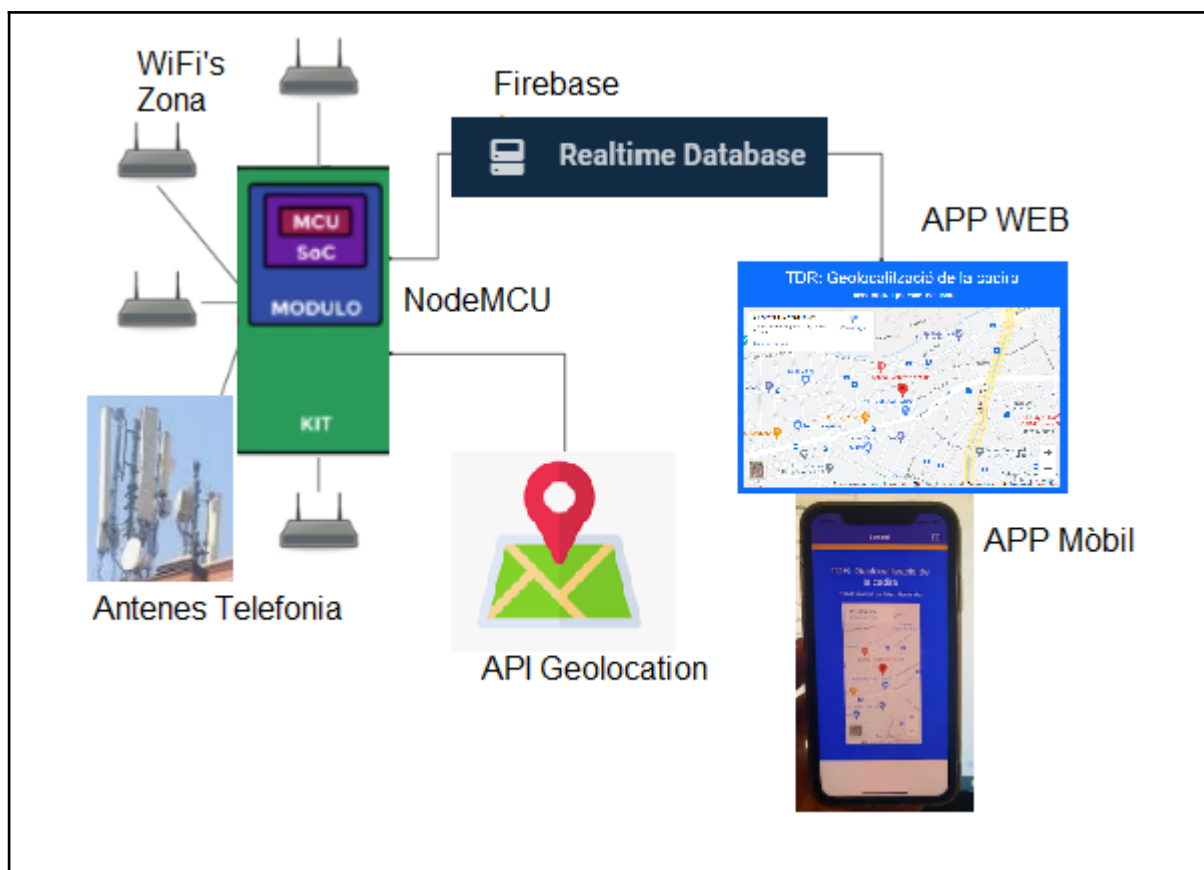


Fig: Esquema de funcionament de la Geolocalització
A partir de la planificació anterior es desenvolupa el projecte.

3.2.3.1 Elecció de la placa

L'elecció del tipus de placa ve condicionada per el propi projecte ja que es necessita connexió a internet a l'hora d'enviar la posició a la base de dades. Per tant, entre el debat de placa GPS i via WiFi, escollim WiFi, sempre i quan, els valors de posicionament siguin fiables.

Una vegada decidit el sistema per a la geolocalització, vaig cercar informació sobre les plaques que fan aquesta funció. Hi ha dues plaques amb aquestes característiques, la nodeMCU i la MKR1000. Totes dues tenen la funcionalitat WiFi però la diferència principal és que el nodeMCU es bloqueja quan està alimentat elèctricament i porta molta estona sense connectar-se a internet. Tanmateix, el nodeMCU té un cost molt inferior, entorn als sis euros, respecte als trenta-cinc del MKR1000. Evidentment, la diferència de preu es transforma en qualitat i més fiabilitat.

Al final s'ha decidit utilitzar la nodeMCU bàsicament per dos motius:

- La renovació de la connexió WiFi es farà cada quinze minuts. De fet, s'han

realitzat unes proves de fiabilitat que superen el 80 % d'èxit a les connexions en les citades condicions. És a dir, que de cada deu intents hi ha vuit que es realitzen amb èxit

- La relació qualitat preu

3.2.3.2 Obtenir les credencials API Geolocation

L'objectiu és aconseguir les credencials per poder enllaçar qualsevol APP amb el API de geolocalització. Aquesta credencial es diu API KEY i s'aconsegueix amb el següent procediment:

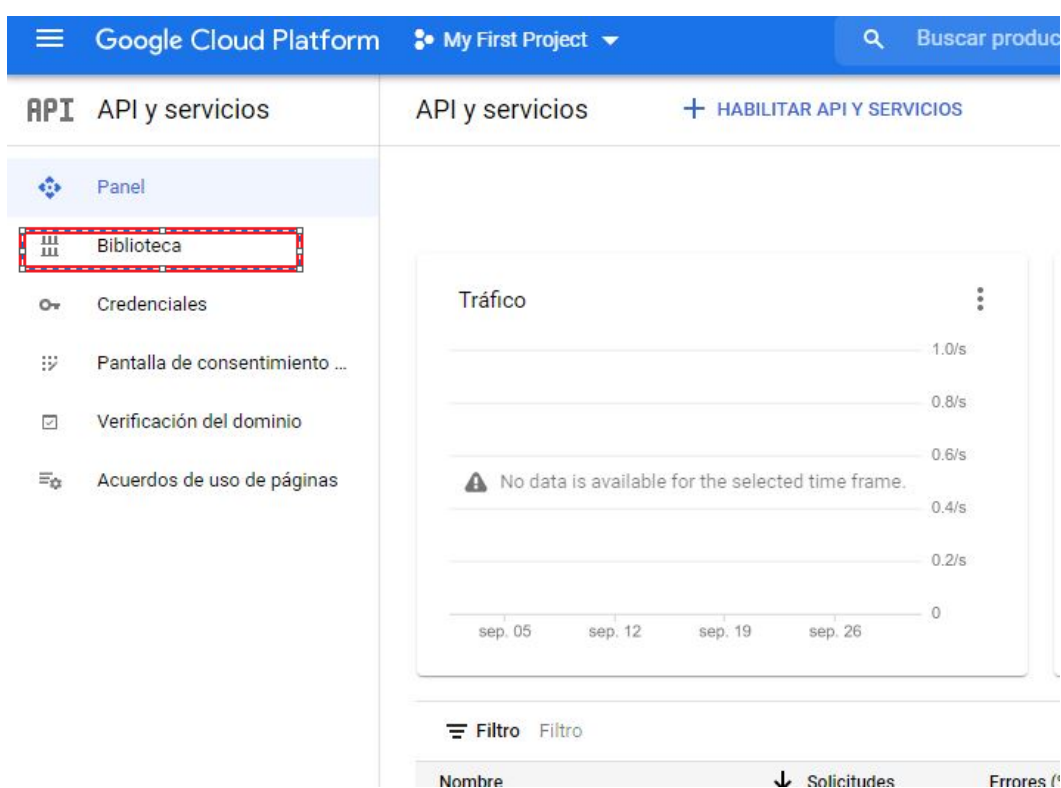


Fig: Biblioteca d'API's

A través de la consola d'administració de google (<https://console.cloud.google.com/>) i amb qualsevol compte gmail es pot habilitar les API's de la biblioteca.

Després, es necessita activar la biblioteca Geolocation API i crear un projecte. Una vegada fet això, s'ha d'anar a l'apartat de credencials a la consola i clicar el botó de crear credencials.

Finalment, es segueix un procés amb dos passos, en aquest cas, es selecciona l'opció dades públiques per tal que no sigui necessari autenticació d'usuari per accedir a l'API.

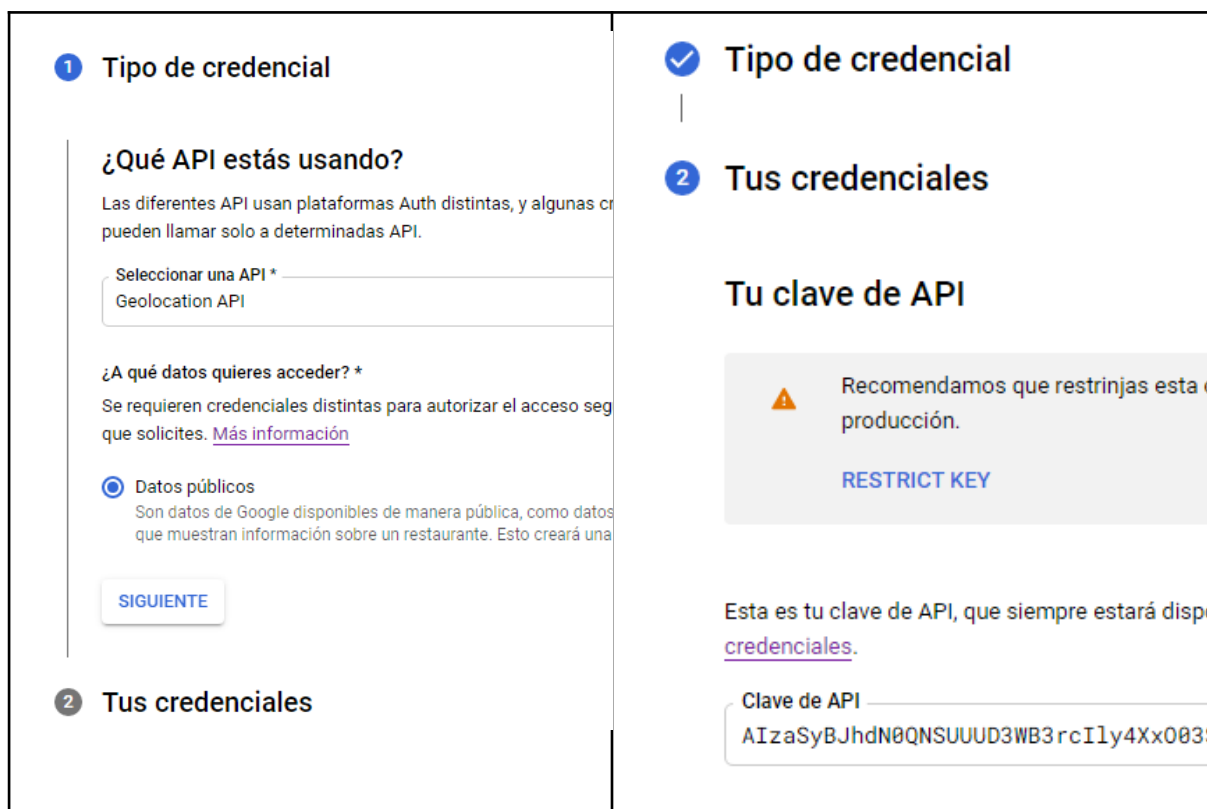
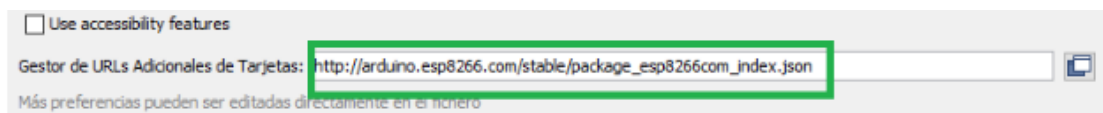


Fig: Credencial: API Key

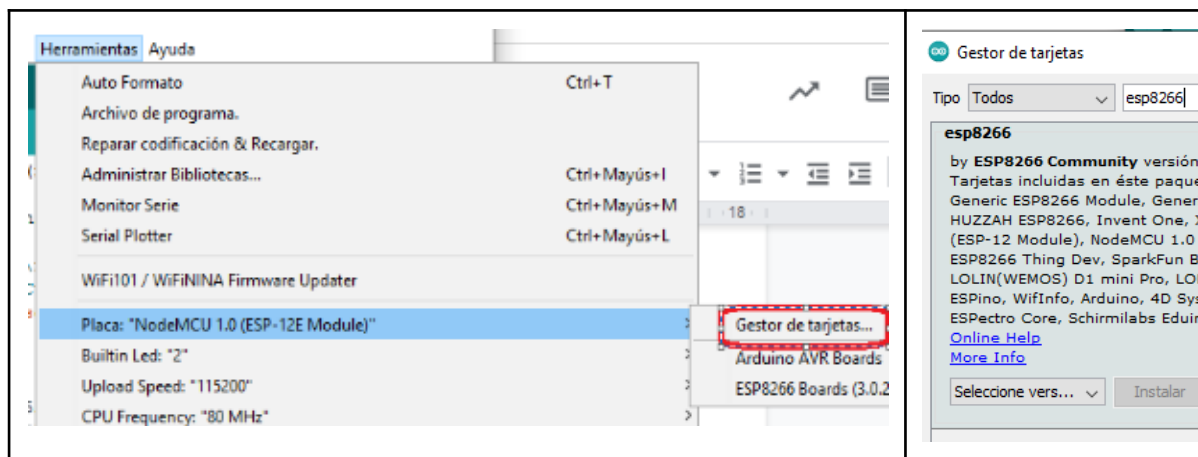
3.2.3.3 NodeMCU: internet i obtenció de les coordenades

El programa arduino IDE és el que faig servir per programar el mòdul nodeMCU, per tal que funcioni amb aquesta placa s'ha de configurar dues coses, una afegir el driver de la citada placa i afegir la biblioteca de WifiLocation. Aquestes dues accions requereixen els següents passos:

- ❖ Afegir una adreça URL amb els drivers de les diferents versions de la placa ESP8266. Per realitzar això anem al menú arxiu/preferències de l'arduino IDE i afegim a ajustaments:

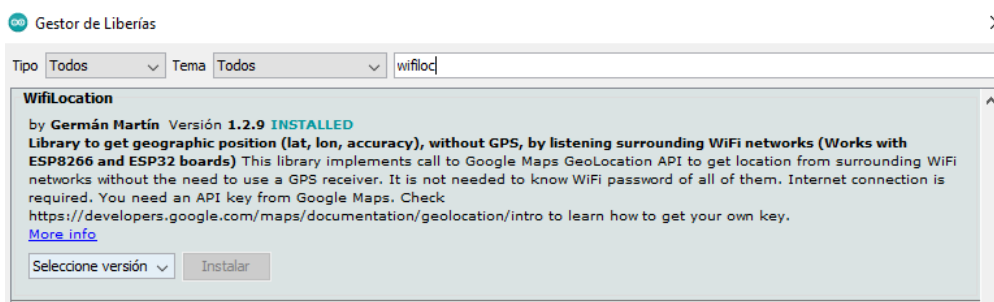


- ❖ Carregar i seleccionar la tarjeta ESP 8266 desde eines/gestor de targetes:



Al nostre cas serà la NodeMCU 1.0 (ESP-12E Modelo).

- ❖ Baixar la biblioteca Wifilocation des de el menú eines/Administrar biblioteques:



- ❖ A través de l'exemple googleLocation que proporciona la biblioteca WiFiLocation utilitzem el mòdul de geolocalització per wifi. S'ha d'afegir el API KEY, el SSID de la xarxa i contrasenya.

```
Location request data
[{"macAddress": "BE:DD:C2:5D:71:50", "signalStrength": -64, "channel": 1},
{"macAddress": "18:82:8C:B3:6D:ED", "signalStrength": -60, "channel": 1},
{"macAddress": "00:15:6D:9E:26:AE", "signalStrength": -73, "channel": 1},
{"macAddress": "84:AA:9C:08:80:D4", "signalStrength": -86, "channel": 6},
{"macAddress": "3E:55:64:65:56:3A", "signalStrength": -34, "channel": 6},
{"macAddress": "84:D8:1B:B8:85:B6", "signalStrength": -62, "channel": 4},
{"macAddress": "84:D8:1B:B8:5C:8E", "signalStrength": -72, "channel": 4},
{"macAddress": "84:AA:9C:9E:EA:6C", "signalStrength": -87, "channel": 6},
{"macAddress": "64:CC:22:4C:23:0C", "signalStrength": -76, "channel": 11},
{"macAddress": "94:6A:B0:1C:37:A2", "signalStrength": -82, "channel": 11}]

Location: 41.9758492,2.8056178
Accuracy: 18
Result: OK
```

Finalment, hem comprovat si les coordenades que proporciona el serial són correctes amb la web (<https://www.coordenadas-gps.com/>).

El desenvolupament del programa es troba a l'apartat 3.4.2.2.

3.2.3.4 Firebase Hosting i Realtime Database:

Implementació del hosting:

Per la implementació del hosting m'he passat moltes hores ja que les instruccions de la consola de firebase no funcionaven prou bé al meu ordinador. Els següents passos resumeixen la implemmentació del hosting:



❖ Instal·lació del node.js, aquest programa permet, entre altres coses, crear un enllaç amb uns repositoris on podem baixar les eines de Firebase.

A partir d'aquí el següents passos es fan desde la consola de comandes de windows amb les directrius de la consola d'administració de Firebase:

❖ Baixar i instal·lar els paquets necessaris per Firebase a través de la consola de comandes de windows. (npm install -g firebase-tools)

- ❖ Inicialitzar el hosting (firebase init)
 - seleccionar instal·lar aquests dos paquets:
 - Hosting
 - Database
- ❖ Entrar amb l'usuari (firebase login)
- ❖ Implementació de la base de dades i el hosting (firebase deploy --only hosting)

```
firebase deploy --only hosting
=== Deploying to 'wifi-arduino-modemcu'...
i deploying hosting
i hosting[wifi-arduino-modemcu]: beginning deploy...
i hosting[wifi-arduino-modemcu]: found 1 files in public
+ hosting[wifi-arduino-modemcu]: file upload complete
i hosting[wifi-arduino-modemcu]: finalizing version...
+ hosting[wifi-arduino-modemcu]: version finalized
i hosting[wifi-arduino-modemcu]: releasing new version...
+ hosting[wifi-arduino-modemcu]: release complete
+ Deploy complete!
Project Console: https://console.firebase.google.com/project/wifi-arduino-modemcu/overview
Hosting URL: https://wifi-arduino-modemcu.web.app
```

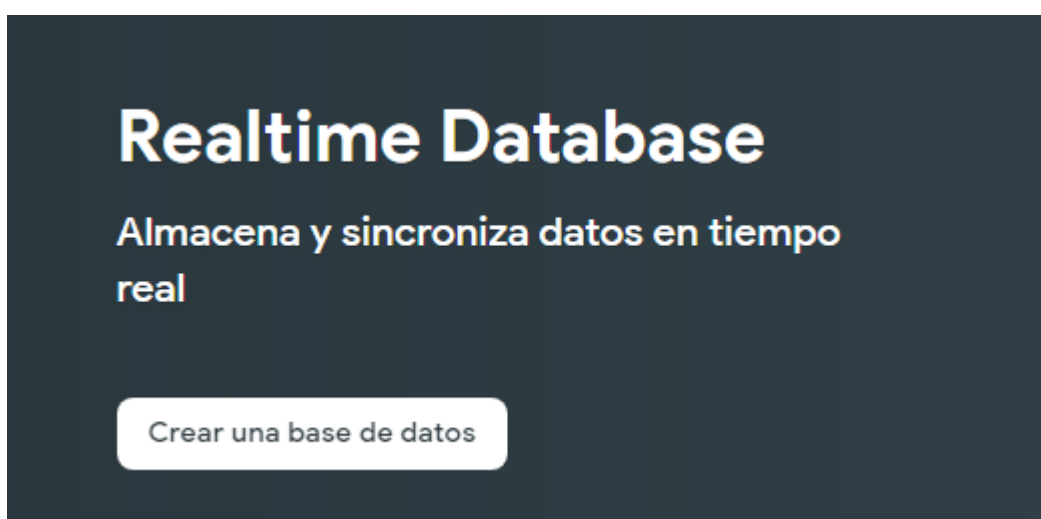
En realitat, amb aquest procés, s'aconsegueix fer visible per internet a través de l'adreça web (<https://wifi-arduino-modeMCU.web.app>), l'arxiu index.html, que està ubicat a la carpeta local anomenada public.

Finalment, destacar que l'arxiu index és el que editem per programar la web.

Implementació de Firebase Database:

En quant a configuració de la base de dades es realitza totalment amb la consola d'administració de Firebase (firebase.com). És força intuïtiu i es resumeix en dues parts:

- Crear la base de dades:



- Editar les regles de seguretat per al projecte :

He decidit que com que les dades que introduïrem són només latituds i longituds no caldrà posar cap contrasenya de seguretat per accedir a les dades.

Si calgués es podria modificar les regles fent que s'hagués d'entrar amb autenticació d'usuari o de contrasenya.

Una vegada arribat a aquest punt podem afegir dades des de la consola. Ara només cal implementar-la per tal que es pugui accedir desde el nostre domini d'internet. La implementació es fa des de la consola de comandes amb la instrucció `firebase deploy --only database`.

Amb el passos anteriors la base de dades està preparada per enviament i recepció de dades.

Les operacions bàsiques que es realitza sobre les dades en qualsevol base de dades són les altes (set), les baixes (remove), les modificacions (update) i les consultes (get).

Firestore, és una mica especial i apareixen dues noves funcionalitats; una és el push (emputjar) que seria semblant a set, o sigui introduir dades però, creant registre de cada interacció. L'altra funció, és snapshot que és una consulta de dades tipus get però, d'alguna manera, observa o vigila els canvis de valors a la base de dades per actualitzar l'App. Per entendre aquestes dues funcions, podem pensar en la famosa aplicació de missatgeria whatsapp; quan un usuari envia un missatge a un grup es genera un push, és a dir, es van acumulant els missatges; i els receptors reben els missatges d'altres usuaris en temps real (snapshot).

Al projecte he utilitzat dues d'aquestes funcionalitats set per l'escriptura de dades i snapshot per la lectura.



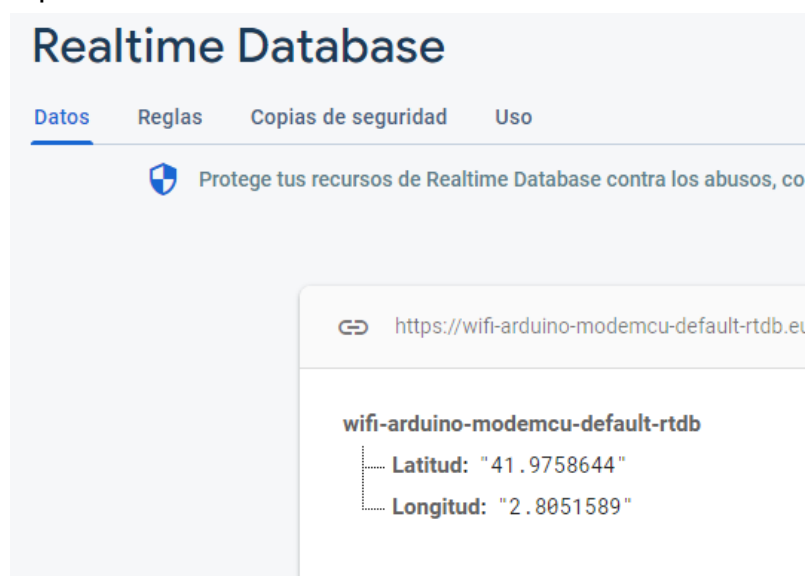
3.2.3.5 Emmagatzematge a Firebase Database desde el nodeMCU

L'emmagatzematge de les coordenades de latitud i longitud a la Firebase es realitza desde el nodeMCU amb la funció set generant dos Json:

```
firebase.setString(fbdo, "Longitud", String(loc.lon, 7));  
firebase.setString(fbdo, "Latitud", String (loc.lat, 7));
```

La funció SET cada vegada que rep noves dades de latitud i longitud elimina els valors anteriors. Si volguéssim guardar un històric utilitzaríem la funció push.

El resultat vist per la consola firebase és:



3.2.3.6 Lectura de dades de la base de dades mitjançant una pàgina web

Per extreure els valors de les coordenades de la base de dades s'utilitza la sentència snapshot d'aquesta manera quan hi hagi un canvi en els valors de les coordenades automàticament canviaran els valors de l'aplicació web.

```
dbRef.once("value")  
  .then(function(snapshot) {  
    var longitud = snapshot.child("Longitud").val();  
    var latitud= snapshot.child("Latitud").val();
```


3.2.3.7 Situació de les coordenades sobre un mapa interactiu

Es genera un marc centrat a la web on situem el link de google maps amb els valors de les coordenades extrems de la base de dades en temps real.

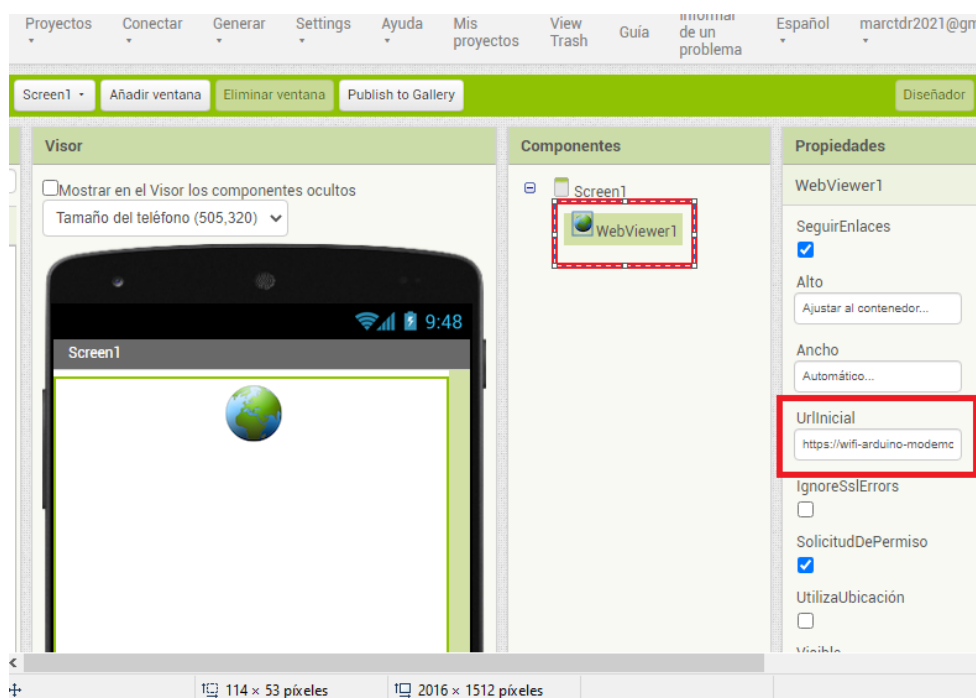
var link =

```
('https://www.google.com/maps/embed/v1/place?key=AIzaSyBJhdN0QNSUUUD3WB3rcIly4Xx003SAfE0&q='+latitud+', '+longitud)
```

```
var iframe = document.createElement('iframe');
iframe.frameBorder=0;
iframe.width="75%";
iframe.height="450px";
iframe.style="border:0";
iframe.loading="lazy";
iframe.setAttribute("src", link);
document.getElementById("demo").appendChild(iframe);
});
```

3.2.3.8 APP Locachair

L'aplicació per a telèfon mòbils es diu "Locachair" i és bàsicament la visualització, mitjançant el component webViewer de l'APP inventor de la web creada i explicada al punt 3.4.2.3.




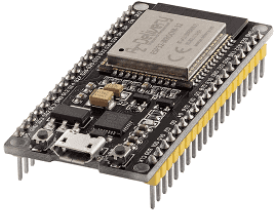
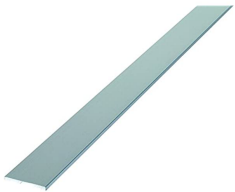
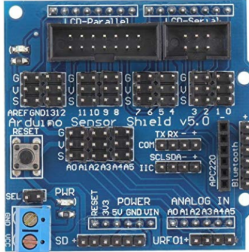



3.3 Planificació

Abans de començar a modificar i construir els elements auxiliar de la màquina, s'ha d'aconseguir tot el material de construcció. Tot el material usat per dur a terme aquest projecte ha estat encarregat a amazon.es (una empresa encarregada de la distribució d'una gran diversitat de productes).

Aquesta llista es troba a l'apartat següent juntament amb el pressupost.

3.3.1 Pressupost

OBJECTE	IMATGE	NOMBRE D'UNITATS	PREU UNITARI	COST TOTAL
Arduino Uno R3.		1	20.90€	20,90€
Elechouse Voice Recognition v3		1	32.99€	32.99€
Micro servo tower PRO SG90 9grs		2	4,24€	8,48€
Cable per a Arduino.		Pack 17	1.98€	1.98€

<p>AZDelivery ESP8266 ESP-12F NodeMCU</p>		<p>3</p>	<p>5.83€</p>	<p>5.83€</p>
<p>Barra d'alumini</p>		<p>1</p>	<p>8,99€</p>	<p>8,99€</p>
<p>Arduino targeta d'expansió sensor shield v5.0.</p>		<p>1</p>	<p>7.00€</p>	<p>7.00€</p>
<p>Cinta de doble cara</p>		<p>1</p>	<p>7,99</p>	<p>7,99</p>
<p>Led RGB per Arduino.</p>		<p>1</p>	<p>0.64€</p>	<p>0.64€</p>
<p>Cargoleria</p>		<p>1</p>	<p>1,00€</p>	<p>1.00€</p>
<p>TOTAL</p>				<p>86,81€</p>

3.4 Realització

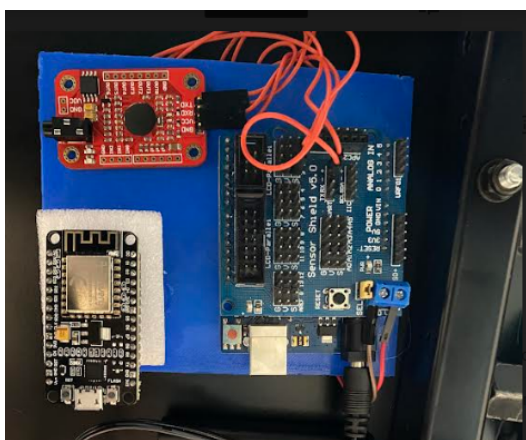
3.4.1 Construcció

El primer que vaig fer quan vaig aconseguir la cadira de rodes va ser provar el seu funcionament per comprovar les maniobres. A continuació, vaig desmuntar el comandament, el compartiment de les bateries i la caixa on hi ha la placa electrònica. Això em va servir per observar com ho havien muntat els fabricants i prendre mesures i fotografies dels detalls.

D'altra banda, vaig fixar-me en el xassís, els motors, les rodes i demés components que tenia la cadira.

Després de tot això, vaig treballar en les diferents solucions i una vegada preparats els components i fabricades les peces necessàries les vaig instal·lar a la cadira.

La modificació mecànica del comandament s'ha adaptat perfectament a la cadira de rodes i la seva funcionalitat és efectiva 100%.



Vaig aprofitar un compartiment a sota la cadira per poder collar una plataforma de plàstic on vaig situar les tres plaques electròniques (Arduino uno, Elechouse i nodeMCU) amb l'alimentació corresponent i vaig cablejar el sistema amb brides i canonada amagafils.

3.4.2 Programació

3.4.2.1 Control de la cadira per veu

A continuació, trobareu el programa que controla el comandament de la cadira a través dels dos servos i arduino UNO:

<pre> #include <SoftwareSerial.h> #include "VoiceRecognitionV3.h" #include <Servo.h> //biblioteca Servos Servo servoEndavant; Servo servoDretaEsq; /** Connection Arduino VoiceRecognitionModule 2 -----> TX 3 -----> RX */ VR myVR(2,3); uint8_t records[7]; // save record uint8_t buf[64]; #define canya (0) #define endavant (1) #define lent (2) #define endarrera (3) #define dreta (4) #define esquerra (5) #define para (6) void printSignature(uint8_t *buf, int len) { int i; </pre>	<p>-Aquestes tres primeres línies inclou les llibreries necessàries per fer funcionar el programari del port sèrie, el sensor VR3 i els actuadors servo. //definir el nom dels servos en aquest cas un que governa el moviment endavant i endarrera i l'altra esquerra dreta</p> <p>-Informació sobre els pins de connexió de l'arduino, o sigui s'escolleix el pin 2 per la transmissió i el 3 per la recepció de la informació en el port sèrie.</p> <p>Es defineix VoiceRecognition amb els pins que volguem, en aquest cas com hem dit escollim 2 i 3 però podem canviar-los..</p> <p>-Crea una variable de 8 posicions amb 10 compartiments de memòria a cadascuna de les posicions.</p> <p>-Definició de llocs de memòria utilitzats per les accions.</p> <p>Es genera una variable printSignature per a monitoritzar els llocs de memòria mitjançant punters.</p>
--	--

```

for(i=0; i<len; i++){
    if(buf[i]>0x19 && buf[i]<0x7F){
        Serial.write(buf[i]);
    }
    else{
        Serial.print("[");
        Serial.print(buf[i], HEX);
        Serial.print("]");
    }
}
}

/**
 * @brief Print signature, if the
 * character is invisible,
 * print hexible value instead.
 * @param buf --> VR module return
 * value when voice is recognized.
 * buf[0] --> Group
 * mode(FF: None Group, 0x8n: User,
 * 0x0n:System
 * buf[1] --> number of
 * record which is recognized.
 * buf[2] --> Recognizer
 * index(position) value of the recognized
 * record.
 * buf[3] --> Signature
 * length
 * buf[4]~buf[n] -->
 * Signature
 */
void printVR(uint8_t *buf)
{
    Serial.println("VR
    Index\tGroup\tRecordNum\tSignature");

    Serial.print(buf[2], DEC);
    Serial.print("\t\t");

```

<pre> if(buf[0] == 0xFF){ Serial.print("NONE"); } else if(buf[0]&0x80){ Serial.print("UG "); Serial.print(buf[0]&(~0x80), DEC); } else{ Serial.print("SG "); Serial.print(buf[0], DEC); } Serial.print("\t"); Serial.print(buf[1], DEC); Serial.print("\t\t"); if(buf[3]>0){ printSignature(buf+4, buf[3]); } else{ Serial.print("NONE"); } Serial.println("\r\n"); } void setup() { /** initialize */ myVR.begin(9600); Serial.begin(115200); Serial.println("Elechouse Voice Recognition V3 Module\r\nControl Cadira de Rodes"); servoEndavant.attach(9); servoDretaEsq.attach(10); servoEndavant.write(90); servoDretaEsq.write(90); </pre>	<p><u>-Inicialització de variables i paràmetres</u></p> <p>-Activa el Voice Recogniton</p> <p>-Estableix el port sèrie amb una velocitat de comunicació de 115200 bits/s</p> <p>-Defineix que els servos endavant/darrera i Dreta esquerra se li assignin el Pin 9 i 10 respectivament.</p> <p>-Coloca el servos amb un angle de 90°, o sigui posició de repòs per el que fa referència al comandament de la cadira.</p>
--	---

<pre>// digitalWrite(led, LOW); if(myVR.clear() == 0){ Serial.println("Recognizer cleared."); }else{ Serial.println("Not find VoiceRecognitionModule."); Serial.println("Please check connection and restart Arduino."); while(1); } if(myVR.load((uint8_t)canya) >= 0){ Serial.println("Canya loaded"); } if(myVR.load((uint8_t)endevant) >= 0){ Serial.println("endevant loaded"); } if(myVR.load((uint8_t)lent) >= 0){ Serial.println("lent loaded"); } if(myVR.load((uint8_t)endarrera) >= 0){ Serial.println("endarrera loaded"); } if(myVR.load((uint8_t)dreta) >= 0){ Serial.println("dreta loaded"); } if(myVR.load((uint8_t)esquerra) >= 0){ Serial.println("esquerra loaded"); } if(myVR.load((uint8_t)para) >= 0){ Serial.println("para loaded"); } }</pre>	<p>-Avisa al monitor sèrie si no troba instruccions emmagatzemades a la memòria.</p> <p>-Informa si no troba el VR3.</p> <p>-Informació per port sèrie de la càrrega correcta de les 7 accions memoritzades.</p>
--	--

<pre> void loop() { int ret; ret = myVR.recognize(buf, 50); if(ret>0){ switch(buf[1]){ case canya: servoDretaEsq.write(90); servoEndavant.write(180); break; case endavant: servoDretaEsq.write(90); servoEndavant.write(150); break; case lent: servoDretaEsq.write(90); servoEndavant.write(120); break; case endarrera: servoDretaEsq.write(90); servoEndavant.write(0); break; case dreta: servoEndavant.write(90); servoDretaEsq.write(180); break; case esquerra: servoEndavant.write(90); servoDretaEsq.write(0); break; case para: servoEndavant.write(90); servoDretaEsq.write(90); break; default: Serial.println("Record function undefined"); } } } </pre>	<p><u>Bucle de funcionament.</u></p> <p>-Variable ret que val 0 i canvia de valor quan detecta veu pel micròfon.</p> <p>Quan s'ha detectat veu es comprova si el so correspon amb algun dels emmagatzemats i es dóna la resposta en forma de moviment dels servos segons la maniobra associada.</p> <p>-L'acció canya posa la cadira a velocitat màxima capa davant.</p> <p>-L'acció endavant elimina els girs esquerra o dreta i posa la cadira a velocitat mitjana.</p> <p>-L'acció lent elimina els girs esquerra o dreta i posa la cadira a velocitat lenta.</p> <p>-Endarrera elimina els girs esquerra o dreta i posa la cadira marxa endarrera.</p> <p>-Para la marxa endavant o endarrera i gira a la dreta.</p> <p>-Para la marxa endavant o endarrera i gira a l'esquerra.</p> <p>-Posició de repòs.</p> <p>-Si no troba el coincidències dóna resposta de so no emmagatzemat o indefinit.</p>
---	---

<pre> break; } /** voice recognized */ printVR(buf); } } </pre>	<p>-Imprimeix el so que ha reconegut.</p>
---	---

3.4.2.2 NodeMCU

El programa fa realitat tres funcionalitats: la connexió a internet del nodeMCU, obtenció dades de geolocalització a través de les torres de telefonia i nodes WiFi i enviament de les coordenades GPS a la base de dades del núvol Firebase:

Aquest programa el separe en 4 parts per explicar-lo amb més claretat.

Primera part: Afegir les biblioteques necessàries pel projecte.

<pre> #include <common.h> #include <Firebase.h> #include <FirebaseESP8266.h> #include <FirebaseFS.h> #include <Utils.h> #include <string> #include <Arduino.h> #include <WifiLocation.h> #include <ArduinoJson.h> #if defined ARDUINO_ARCH_ESP8266 #include <ESP8266WiFi.h> #elif defined ARDUINO_ARCH_ESP32 #include <WiFi.h> #else #error Wrong platform </pre>	<p>-S'inclou la llibreria Firebase ESP8266 client que consta de les llibreries que es detallen al costat.</p> <p>-Afegim la llibreria string per poder crear variables tipus cadena de text</p> <p>-Carreguem la llibreria genèrica d'arduino.</p> <p>-Inclou les llibreries necessàries per la geolocalització WiFi que aprofita els routers de la rodalia i antenes mòbils.</p> <p>-Llibreria per poder generar arxius Json amb arduino.</p> <p>- Aquesta part del programa comprova si els xip wifi és l'ESP8266 o un altra d'anterior que és el ESP32, arregant el component més adequat. Al nostre cas tenim el xip ESP8266 i per tant carregarà la llibreria ESP8266WiFi.</p>
--	---

<code>#endif</code>	
---------------------	--

La segona part és la generació de les variables i objectes necessaris:

<pre> //Variables api de geolocation google, lloc web o URL firebase, autenticació base de dades, precissió messura GPS: const char* googleApiKey = "AIzaSyCCH1VeU0qX0j9r17K1095XDko6NE83 N9w"; const char* FIREBASE_HOST= "wifi-arduino-modemcu-default-rtdb.eu rope-west1.firebaseio.com"; const char* FIREBASE_AUTH= "4LALMEJjnaRLC0A0g4ICm2G8May00ZUD0JGD kBxC"; const char* ssid = "ChairWIFI"; const char* passwd = "tdr_push"; #define LOC_PRECISION 7 WifiLocation location (googleApiKey); WiFiClientSecure client; FirebaseData fbdo; void setClock () { configTime (0, 0, "pool.ntp.org", "time.nist.gov"); Serial.print ("Waiting for NTP time sync: "); </pre>	<p>Variable API de Geolocalització que és la credencial per poder connectar amb el nostre projecte.</p> <p>-Nom de la base de dades Firebase a connectar.</p> <p>-Clau secreta per poder tenir accés a la base de dades. (ara no és tan secreta!)</p> <p>-Usuari SSID que és el nom de la xarxa wifi i la contrasenya per poder accedir a internet.</p> <p>- Precisió en quant al nombre de decimals del valor de latitud i longitud.</p> <p>-Objecte anomenat location de la classe WifiLocation per permet accedir a l'api de Google.</p> <p>-Crea un objecte client de la classe WiFiClientSecure.</p> <p>- Objecte fbdo de firebaseData on gravar les dades per transmetre a firebase</p> <p>-Amb el NTP sincronitzem l'hora del component (NnodeMCU) amb la del servidor. Sense ajustar aquest paràmetre no es produeix la resposta GPS via WiFi.</p>
--	--

<pre> time_t now = time (nullptr); while (now < 8 * 3600 * 2) { delay (500); Serial.print ("."); now = time (nullptr); } struct tm timeinfo; gmtime_r (&now, &timeinfo); Serial.print ("\n"); Serial.print ("Current time: "); Serial.print (asctime (&timeinfo)); } </pre>	
--	--

La tercera part és l'apartat de configuració on inicialitzem el port sèrie i es realitza l'accés a internet:

<pre> void setup() { Serial.begin(115200); // Connect to WPA/WPA2 network WiFi.mode(WIFI_STA); //mode d'accés STA WiFi.begin(ssid, passwd); //Inici de wifi amb usuari ssid i contrasenya while (WiFi.status() != WL_CONNECTED) { Serial.print("Intent de connexió WPA SSID: "); Serial.println(ssid); Serial.print("Status = "); Serial.println(WiFi.status()); //informació de l'estat de la connexió. </pre>	<p>-connexió pin sèrie a 115200 bauds</p> <p>-Accés a internet pel protocol de seguretat WPA/WPA2. Aquest protocol permet d'una banda, accedir a internet i de l'altra, els missatges que s'intercanvien són xifrats.</p> <p>Fins que no es connecti a internet dóna per pantalla missatges de l'estat de la connexió.</p> <p>Missatges d'estat de la connexió:</p> <pre> WL_NO_SHIELD = 255, // for compatibility with WiFi Shield Library WL_IDLE_STATUS = 0, WL_NO_SSID_AVAIL = 1, WL_SCAN_COMPLETED = 2, WL_CONNECTED = 3, WL_CONNECT_FAILED = 4, WL_CONNECTION_LOST = 5, </pre>
---	--

<pre> delay(500); } Serial.println ("Connectat"); setClock (); } </pre>	<pre> WL_DISCONNECTED = 6 </pre> <p>-Espera 5 segons per connectar-se per no saturar el nodeMCU amb molts intents seguits.</p>
---	--

Finalment, la quarta part és el bucle que reportarà les coordenades de GPS cada quart d'hora.

<pre> void loop() { Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); //connexió amb les nostres credencials. Firebase.reconnectWiFi(true); location_t loc = location.getGeoFromWiFi(); Serial.println("Dades de Geolocalització"); Serial.println(location.getSurroundingWiFiJson()+ "\n"); Serial.println ("Localització: " + String (loc.lat, 7) + "," + String (loc.lon, 7)); //Extreu la longitud i la latitud de log Serial.println ("Precissió: " + String (loc.accuracy)); Serial.println ("Resultat: " + location.wlStatusStr (location.getStatus ())); } </pre>	<p>-Connexió amb la base de dades.</p> <p>-Crea una variable loc que emmagatzema una Json amb les variables de la geolocalització que hem rebut.</p> <p>-Monitorització dels resultats de la geolocalització via Google API Geolocation. Aquest sistema aprofita les antenes de telefonia i els WiFi's de la zona per obtenir la geolocalització.</p>
--	---

<pre> Firebase.setString(fbdo, "Longitud", String(loc.lon, 7)); Firebase.setString(fbdo, "Latitud", String (loc.lat, 7)); delay(900000); } </pre>	<p>-Emmagatzemar els valors de la longitud i latitud a Firebase.</p> <p>-Pausa entre dada i dada de GPS de 15 minuts</p>
--	--

3.4.2.3 Pàgina web amb visualització de la latitud i longitud en temps real.

<pre> <!DOCTYPE html> <html> <head> <title>Geolocalització TDR Marc Hernandez</title> <script defer <script type="module"> const firebaseConfig = { apiKey: "AIzaSyD_3spHdM21nzR-vNifOjbd457NekkFwg", authDomain: "wifi-arduino-modemcu.firebaseio.com", databaseURL: "https://wifi-arduino-modemcu-default-rtdb.europe-west1.firebaseio.com", projectId: "wifi-arduino-modemcu", storageBucket: "wifi-arduino-modemcu.appspot.com", messagingSenderId: "997116213366", appId: "1:997116213366:web:4dcbb12659bd74af158757", measurementId: "G-R4Y770L0ZW" }; firebase.initializeApp(firebaseConfig); </pre>	<p>Paràmetres per poder connectar amb la base de dades.</p> <p>Inicialitza la base de dades</p>
---	---

<pre>const dbRef = firebase.database().ref(); dbRef.once("value") .then(function(snapshot) { var longitud = snapshot.child("Longitud").val(); var latitud= snapshot.child("Latitud").val(); console.log(longitud); console.log(latitud); var link = ('https://www.google.com/maps/embed/v1/place?key=AIzaSy BJhdN0QNSUUUD3WB3rcIly4Xx003SAfE0&q='+latitud+', '+longi tud) var iframe = document.createElement('iframe'); iframe.frameBorder=0; iframe.width="75%"; iframe.height="450px"; iframe.style="border:0"; iframe.loading="lazy"; iframe.setAttribute("src", link); document.getElementById("demo").appendChild(iframe); }); </script> <body> <h1>TDR: Geolocalització de la cadira</h1></pre>	<p>Crear una "còpia" o referència de la base de dades</p> <p>-Carrega valors de la base de dades.</p> <p>-Funció d'obtenir dades snapshot (cada vegada que un valor canvia a la base de dades es refresca a la web)</p> <p>-obtenim dins la variable local els valors corresponents.</p> <p>-Imprimeix a la consola aquests valors.</p> <p>-Visualitza el mapa amb la posició de geolocalització.</p> 
---	--

<pre><p>Treball realitzat per Marc Hernandez</p> <div id="demo"></div> </body> </html></pre>	<p>TDR: Geolocalització de la cadira <small>Treball realitzat per Marc Hernandez</small></p>
--	--

5. Avaluació i conclusions

Principalment, m'agradaria destacar que he aconseguit realitzar bastants dels els objectius plantejats inicialment, quan vaig triar el tema del treball.

En primer lloc, he aconseguit automatitzar la cadira elèctrica per veu, és a dir, he fet que la cadira es desplaci a partir de les instruccions de veu esmentades en el treball. Però cal dir, que la cadira de rodes automatitzada per veu no és fiable i els motius probables és que el micròfon que s'ha utilitzat no tenia filtres pel soroll de fons i que l'arduino UNO no té suficient capacitat per albergar aquest sistema.

D'una banda, els resultats de l'automatització de la cadira de rodes per veu no ha estat tan satisfactòria com s'esperava, no per la part mecànica, que ha funcionat amb una efectivitat del 100% sinó que el problema ha estat amb el mòdul de reconeixement de veu que s'ha mostrat irregular. Si bé assoleix un 65% d'encerts quan les ordres per veu es donen sense soroll de fons; quan hi ha soroll de fons, l'efectivitat cau a un 30%. Això es tradueix en poca fiabilitat del sistema a l'hora de poder ser comercialitzat.

Tanmateix, el sistema de geolocalització ha donat molt bons resultats i la fiabilitat és molt alta. Envia les coordenades i la localització en el mapa en temps real.

Per altra banda, probablement, el hardware utilitzat és senzill, barat i no és del tot adequat per la fita d'aquest projecte. Si plantejés el projecte actualment, compraria l'arduino MEGA, un mòdul amb filtres del soroll per al micro i una millor placa per realitzar la geocalització perquè tingues gairebé el 100% d'efectivitat en comptes del 80% que ha tingut.

He realitzat aquest treball perquè penso que podria ajudar a moltes persones. De fet, si el hardware utilitzat es millorés i es potenciés la cadira seria útil i per tant, podria comercialitzar-se.

Per acabar, estic bastant satisfet amb els resultats obtinguts en aquest treball, ja que el treball m'ha aportat nous coneixements i experiències, i l'oportunitat de dirigir un projecte.

Finalment, tot i que sembli que aquest projecte ha anat rodat no és cert, ha tingut moltes dificultats durant el procés i definitivament el principal topall d'aquest projecte ha estat la limitació en el hardware.



6. Agraïments

Primer de tot, voldria donar les gràcies a la meua tutora, Marta Riera, per donar-me suport en tot moment, ajudar-me en els aspectes que necessitava i per estar preocupada perquè el treball sortís el millor possible.

Per altra part, volia agrair la confiança del director de l'escola d'educació especial Palau de Girona per haver-me prestat la cadira. D'aquesta manera he pogut desenvolupar el projecte. També el suport que m'ha donat l'associació Multicapacitats, buscant diferents usuaris amb cadires diferents per poder estudiar-les.

Finalment, voldria agrair a la meua família el seu suport, tant anímic com econòmic.

7. WEBGRAFIA

LLAMAS, LUIS. Ingeniería, informática i diseño

www.luisllamas.es/maquina-de-estados-finitos-arduino/

WIKIPEDIA. La enciclopedia libre www.wikipedia.org

VICKIPEDIA. L'enciclopedia lliure www.vikipedia.org

R3ALD. Innovation for real 3D prints www.r3ald.com/que-es-un-fichero-stl

GONZÁLEZ, JUAN. Curso de Introducción del diseño 3D en FreeCad

<http://diwo.bq.com/course/curso-de-introduccion-a-freecad/>

PADI cnc. http://padicnc.com/index.php?link=G-Code/106_CorteSiluetaSimple

AMAZON. Distribuidor de productos <https://www.amazon.es/>

POLARIDAD.ES. Crear G-Code con Cura <https://polaridad.es/gcode-cura-imprimir-3d/>

DIDAC. Diccionari de català <http://www.dicdidac.cat/>

CLASSIFICACIÓ CADIRES DE RODES

<https://upcommons.upc.edu/bitstream/handle/2117/181445/3.Estat%20del%20Art%20%28Cadires%20de%20Rodes%2C%20Sistemes%20entrada%20Vehicle%2C%20Adaptacions%20Vehicle%29.pdf?sequence=4&isAllowed=y>

POWER PACK PLUS

<https://www.ortopediamimas.com/movilidad/motores-para-sillas-de-ruedas-manuales/4761-motor-auxiliar-para-silla-de-ruedas-power-pack-plus.html>

PÀGINA DEL FABRICANT DEL MÒDUL ELECHOUSE DE RECONEXIEMENT DE VEU

http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath&products_id=2254

MANUAL ELECHOUSE

https://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf

BIBLIOTECA PER ARDUINO IDE DE LOCALITZACIÓ WIFI

<https://github.com/gmag11/WifiLocation>

Característiques del MÒDUL nodeMCU

<https://naylorlampmechatronics.com/expressif-esp/153-nodemcu-v2-esp8266-wifi.html>

NodeMCU tutorial paso a paso desde cero

<https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

Bases de dades relacionals i no relacionals

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

Connexió nodeMCU amb firebase

<https://www.taloselectronics.com/blogs/tutoriales/como-conectar-nodemcu-con-firebase>

Biblioteca per arduino Firebase-ESP8266

<https://github.com/mobizt/Firebase-ESP8266>

Consola developers de google amb la documentació dels projectes (documentació google Maps en aquest cas)

https://developers.google.com/maps/documentation/embed/embedding-map#place_mode