

Treball de recerca

Robot controlat des del mòbil



Pseudònim: Voyager

Curs: 2018 / 2019

Grup: 2n de Batxillerat

RESUM INICIAL

La finalitat del treball de recerca és dissenyar, construir i programar un robot que pugui ser controlat mitjançant un mòbil i que pugui executar les funcions de seguir una línia, esquivar obstacles, controlar el gir d'una càmera i poder-lo desplaçar endavant, endarrere, girar a l'esquerra i a la dreta. Aquest és el meu primer contacte en el món de la programació i del disseny gestionant-ho jo mateix.

Paraules clau: *robòtica, Arduino, App Inventor, components electrònics, programació*

RESUMEN INICIAL

La finalidad del proyecto es diseñar, construir i programar un robot que pueda ser controlado mediante un móvil y que pueda ejecutar las funciones de seguir una línea, esquivar obstáculos, controlar el giro de una cámara i poderlo desplazar adelante, atrás, girar a la izquierda y a la derecha. Este es mi primer contacto en el mundo de la programación y del diseño gestionándolo yo mismo.

Palabras clave: *robótica, Arduino, App Inventor, componentes electrónicos, programación*

ABSTRACT

The aim of the research project is to design, build and programme a robot controlled by a mobile phone. It must be able to follow different functions such as following a line, avoid obstacles, control a camera and move the vehicle forward, backward, turn left and right. This is my first contact with programming and design by myself.

Key words: *robotics, Arduino, App Invetor, electronic components, programming*

ÍNDIX

1. INTRODUCCIÓ	1
2. QUÈ ÉS UN ROBOT?	3
3. DESCRIPCIÓ GENERAL DEL VEHICLE VOYAGER	9
4. DESCRIPCIÓ DELS COMPONENTS ELECTRÒNICS.....	13
4.1. Arduino UNO r3 marca Elegoo	13
4.2. Bluetooth HC-05	15
4.3. Emissor d'ultrasons HC-SR04	17
4.4. Sensor d'infrarojos	19
4.5. Plaques de connexió.....	20
5. DESCRIPCIÓ DE LA MECÀNICA, SISTEMA ENERGÈTIC I ACCESSORIS DEL ROBOT.....	23
5.1. Motors.....	23
5.2. Servomotor	24
5.3. Motor pas a pas	26
5.4. Sistema de gestió de l'energia.....	29
6. DESCRIPCIÓ DEL SOFTWARE DESENVOLUPAT	31
6.1. Programació de l'Arduino.....	31
6.2. Programació de l'aplicació	33
7. ANÀLISI I CONCLUSIONS	37
8. BIBLIOGRAFIA I WEBGRAFIA	41
9. AGRAÏMENTS.....	43
10. ÍNDIX D'IL·LUSTRACIONS	44
11. LLISTAT D'ABREVIACIONS I SÍMBOLS	47
12. ANNEXOS.....	49

1. INTRODUCCIÓ

Per què he escollit un treball relacionat amb la robòtica? Bé, hi ha moltes raons per les quals he decidit realitzar aquest treball, però la principal és perquè considero que els robots comencen a ser el present i seran el futur de la societat. Els robots són creats per facilitar les accions humanes disminuint-ne el temps de treball i els esforços i problemes que poden causar. Per tant, trobant-nos en una societat a la qual li agraden les facilitats, els robots adquireixen un paper fonamental que permetrà que es realitzin projectes i investigacions per tal de treure al mercat els millors robots.

Un altre motiu és perquè va més enllà del paper. La base del treball no consisteix en una part escrita sinó que és pràctica i aquest fet em genera més interès i més ganes per realitzar el treball. A més a més, l'estiu passat vaig realitzar un curs a la URV, dins del programa *EstiuURV*, relacionat amb el tema i em va generar interès per aprendre'n més i, penso, que aquest era el moment idoni per fer-ho.

Un altre argument és perquè el treball no consisteix en una simple investigació, sinó que hauré d'adquirir nous coneixements per poder plantejar uns objectius de com vull que sigui el robot que construiré i m'hauré d'afrontar a diversos reptes quan el robot no funcioni tal com vulgui. Aquest camí d'obstacles que hauré de seguir serà el que em permetrà continuar tenint ganes de dedicar temps al treball i d'intentar obtenir un bon resultat.

El principal objectiu en elaborar aquest treball no és construir un robot d'unes característiques molt avançades i que es pugui llançar al mercat, sinó construir un robot en el qual pugui aplicar els nous coneixements adquirits, i que tot el procés sigui una oportunitat per anar superant les dificultats i els problemes que aniran sorgint.

El resultat del treball ha estat dissenyar, construir i programar un robot en forma de vehicle que he anomenat *Voyager*.

Les característiques principals del robot *Voyager* són:

- Disseny de l'estructura fet amb Tinkercad.
- Programació amb el compilador d'Arduino. El robot pot realitzar les següents accions: seguir una línia negra, esquivar obstacles, controlar la càmera que permet gravar per on es desplaça i controlar el robot endavant, endarrere, gir a l'esquerra o gir a la dreta.
- Control des del mòbil a partir d'una aplicació creada amb App Inventor.

El treball es basa a explicar com s'ha anat construint i programant el vehicle per arribar al resultat final. S'inicia amb una introducció a la robòtica, per passar a fer una descripció general del robot *Voyager*. Tot seguit, s'endinsa en explicar el funcionament dels components que constitueixen el vehicle i s'exposa el software utilitzat per permetre el moviment del robot. A les conclusions es fa una anàlisi del treball realitzat, es proposen possibles millores i s'expliquen els problemes que hi ha hagut en l'elaboració.

Es finalitza el treball amb els agraïments i els annexos.

2. QUÈ ÉS UN ROBOT?

La paraula “robot” prové del txec *robota* i significa “esclavitud” fent referència als “treballadors llogats” que van viure en l’Imperi Austrohongarès fins a 1848. Així i tot, qui va utilitzar la paraula robot amb el significat tal com el coneixem actualment va ser Karel Čapek a la seva obra teatral *Robots Universals de Rossum* juntament amb el seu germà Josef. Tot i que a l’obra es refereix com a robots a uns humans artificials orgànics, actualment el mot robot s’utilitza per a tota mena de mecanisme automatitzat que no ha de tenir necessàriament forma d’humà. Als anys 20, la seva obra, R.U.R., va arribar als escenaris londinencs i va permetre la difusió del terme.

Tot i que molts cops s’utilitzen com a sinònims, cal establir una diferència entre el terme “autòmat” i el terme “robot”. Tots dos termes fan referència a màquines automàtiques però es diferencien en el fet que els robots són màquines electròniques mentre que els autòmats són màquines mecàniques, és a dir, autopropulsades per mitjans hidràulics, pneumàtics, elèctrics, etc. D’aquesta manera, els autòmats són anteriors als robots perquè els mètodes per fabricar-los van ser descoberts abans, ja que no requereixen tant enginy com per fabricar un robot. Així doncs, el primer robot va ser creat l’any 1948 pel neuròleg William Grey Walter. Va construir la primera xarxa neuronal artificial que permetia que un robot de tres rodes es pogués moure en funció dels estímuls que captaven els sensors. El primer robot industrial va ser inventat el 1954 per George Devol. Aquest rebia el nom d’*Unimate* i tenia la funció d’aixecar peces calentes de metall i apilar-les.

Des de finals de segle XX fins a principis de segle XXI s’han creat una gran varietat de robots que ha permès que els puguem classificar tenint en compte molts factors. La classificació més útil i més coneguda és en funció del servei que ens aporta el robot:

- **Robots industrials:** Utilitzats en els processos de fabricació industrial. Majoritàriament es tracta de braços robòtics que tenen la funció de

moure objectes de lloc o realitzar alguna modificació sobre un producte. El principal objectiu d'aquests robots és que el procés de fabricació sigui més ràpid i reduir la mà d'obra.



Imatge 1: Robot industrial que realitza soldadures

Autor: Phasmatisnox

- **Robots de servei:** Acostumen a ser mòbils i, normalment, consten d'un o diversos braços mecànics i realitzen labors no industrials. En general, s'utilitzen per reemplaçar a l'ésser humà en entorns hostils.



Imatge 2: Robot de servei que neteja recintes tancats autònomament

Autor: Guzugi

- **Robots militars:** Tenen una funció militar específica i poden ser de moltes formes diferents en funció de la seva utilitat.



Imatge 3: Robot militar

Autor: Army Images

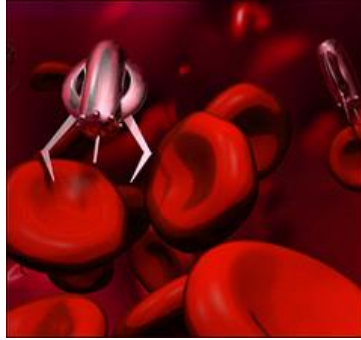
- **Robots mèdics:** S'utilitzen principalment en el camp de la cirurgia. Aporten una alta precisió mitjançant uns braços robòtics que faciliten la feina al cirurgià.



Imatge 4: Cirurgians de Keesles realitzen la primera cirurgia robòtica

Autor: Jenay Randolph

- **Nano robots:** Molt petits, fabricats amb nano tecnologia, s'incorporen a l'interior del cos humà amb l'objectiu de combatre determinades malalties.



Imatge 5: Nano robots
Autor: Nanotecnología Médica

- **Robots educatius:** Utilitzats principalment en escoles i instituts, el seu principal propòsit és educatiu. La línia de robots educatius més coneguda és LEGO Mindstorms.



Imatge 6: Robot educatiu LEGO Mindstorms
Autor: Bernard Goldbach

- **Joguines robòtiques:** Robots que poden adoptar formes molt diverses que tenen l'única funció d'entretenir.



Imatge 7: Joguina robòtica

Autor: Alexas_Fotos

L'any 2010, el Science and Engineering Research Council (SERC) va reunir a un grup d'experts en tecnologia, indústria, art, lleis i ciències socials, per elaborar uns principis relacionats amb els robots. Aquests principis van ser redactats perquè consideren que les tres lleis d'Asimov (un robot no ha de fer mal a un humà, ha d'obeir les ordres que l'humà li ha donat i ha de protegir la seva pròpia existència) no podrien ser aplicades estrictament, ja que podrien ser contradictòries i no s'adapten al tipus de robot actual. A més a més, consideren que la responsabilitat ètica de fabricar robots i introduir-los a la nostra societat no només ha de recaure sobre les màquines sinó també sobre aquells que l'han fabricat. Així doncs els 5 principis acordats per SERC són els següents:

1. Els robots no han de ser dissenyats exclusivament per perjudicar la integritat física dels éssers humans.
2. Els éssers humans són els responsables, no els robots. Els robots són eines dissenyades per complir amb els objectius dels humans.
3. Els robots han de ser dissenyats de tal manera que assegurin la seva protecció i seguretat.
4. Els robots són objectes, no han de ser dissenyats per evocar una resposta emocional. Sempre ha de ser possible diferenciar a un robot d'un ésser humà.

5. Sempre ha de ser possible saber qui és el responsable i propietari legal del robot.

L'existència dels robots a la nostra societat ha provocat que el nostre estil de vida canviï. Els robots van ser creats per substituir la mà d'obra humana en tasques perilloses, difícils o estressants o facilitar determinades feines als humans. Al sector industrial és on es va notar amb més força la seva aparició en la nostra vida. La utilització dels robots a les indústries ha causat un fort avantatge en la fabricació de productes accelerant el procés, reduint la mà d'obra i permetent l'abaratiment dels productes finals. En un principi aquest aspecte es considera positiu, però també cal tenir en compte que genera un problema per aquelles persones que treballaven a les fàbriques, que havien estudiat per això i que, a causa de l'automatització dels processos industrials, han perdut el seu lloc de treball i han hagut de buscar altres feines que no els genera els mateixos beneficis.

Un altre dels sectors on són molt útils els robots és el de la sanitat. En aquest sector la utilització de màquines electròniques es considera tot un avantatge, ja que permet ocupar llocs de treball que, a causa de falta de personal, estaven desocupats i facilita molta feina als cirurgians pel que fa al cansament i a la precisió.

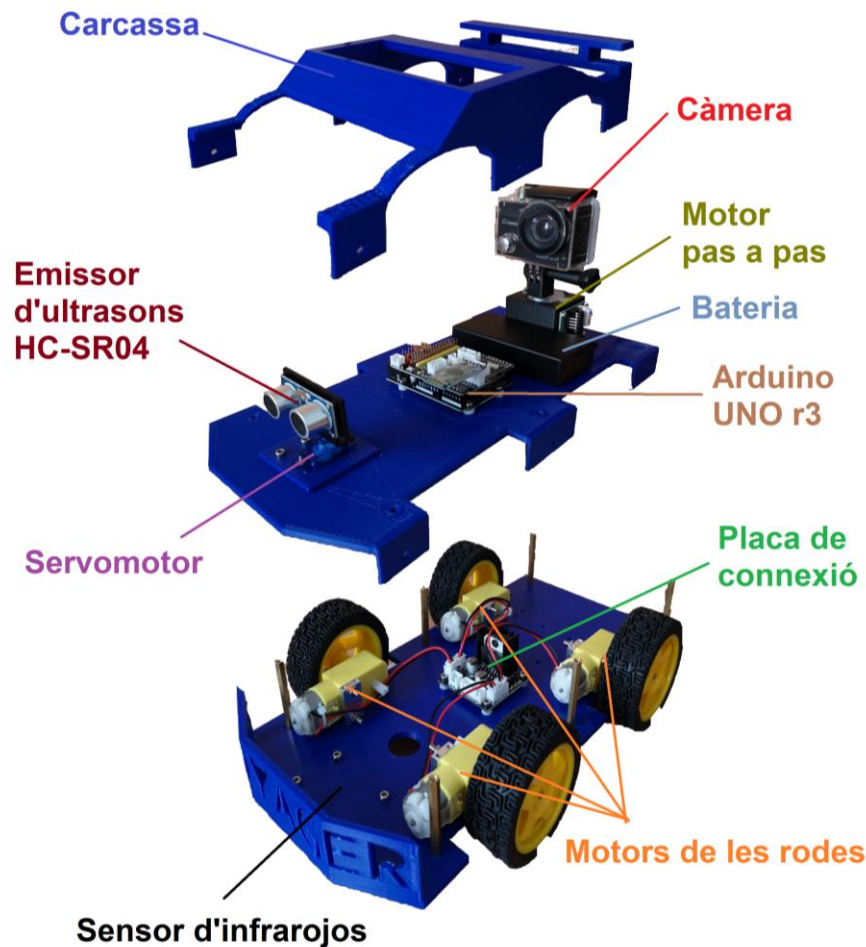
Altres factors en els quals també adquireix importància l'ús dels robots són el domèstic i el militar. En el cas del sector domèstic els robots es consideren un autèntic èxit, ja que són capaços de realitzar feines de casa, feines que a poques persones els agraden fer. D'altra banda el sector militar mostra la cara negativa de la utilització dels robots. Al llarg del temps s'han creat robots capaços de realitzar acrobàcies que pocs éssers humans poden fer i s'estan creant amb l'objectiu que puguin participar en guerres i matar a éssers humans. En aquest aspecte ja han sorgit diverses campanyes en contra en les que hi participa gent important relacionada amb el món de la tecnologia, com per exemple, el cofundador d'Apple, Steve Wozniak, o el fundador de PayPal, Elon Musk.

3. DESCRIPCIÓ GENERAL DEL VEHICLE VOYAGER

El robot que s'ha construït està inspirat en el robot que es pot muntar en comprar la caixa anomenada "Smart V3.0 Robot Car Kit" de la marca Elegoo. La compra d'aquesta caixa va permetre realitzar les primeres proves amb l'elaboració del programa. La caixa portava incorporat la majoria de components que componen el robot i, també, una aplicació senzilla per controlar el robot. Un cop es va crear el programa, es va desmuntar el robot i es van reorganitzar els components canviant-los de lloc i afegint-ne de nous per millorar-lo. A part, es va crear una nova aplicació i es va haver de dissenyar una nova estructura per sostenir els components.

El principal objectiu a l'hora de construir el robot era aprendre, per tant, es considera un robot educatiu. Per tal de complir aquest objectiu, el robot és capaç de realitzar tres funcions principals: ser controlat des d'un mòbil, seguir una línia negra i esquivar obstacles. Per tal de poder realitzar aquestes funcions s'ha escollit donar-li una forma molt semblant a la d'un cotxe. Al vehicle resultant se li ha posat de nom *Voyager*. El motiu pel qual s'ha elegit aquest nom és perquè *Voyager*, en anglès, vol dir viatger i es creu que aquest adjectiu defineix bastant al vehicle, ja que sempre s'està movent i mai para quiet. A continuació hi ha una descripció general de tots els components electrònics i els programes que s'han utilitzat per fer funcionar el robot.

Els components hardware que s'han utilitzat en muntar el vehicle *Voyager* són els següents:



Imatge 8: Estructura del vehicle Voyager

- **Arduino UNO r3** ⇒ És el cervell del robot que permetrà el seu funcionament. És el component on s'aboca el programa Arduino compilat i que serà l'encarregat d'enviar les determinades funcions a realitzar a cada component que forma el robot. Porta incorporat una placa de connexió per facilitar les connexions entre cables i components.
- **Bluetooth HC-05** ⇒ És el component que permet l'intercanvi de dades entre el robot i el mòbil, per tant, permetrà controlar el robot des del mòbil mitjançant una aplicació creada amb App Inventor.
- **Emissor d'ultrasons** ⇒ Es troba a sobre del servomotor. Permetrà que el robot es pugui moure de manera autònoma sense que xoqui amb cap

obstacle. Ho farà mitjançant l'emissió d'un ultrasò i calculant el temps que tarda a rebre'l.

- **Sensor d'infrarojos** ⇒ Està situat a la part inferior de la peça blava en la qual hi ha les rodes. És una placa que consta de tres sensors infrarojos que permetran que el robot segueixi una línia negra.
- **Placa de connexió** ⇒ A banda de la placa de connexió situada sobre la placa Arduino UNO r3, també n'hi ha una situada a la peça que suporta les rodes. La seva funcionalitat és reduir la quantitat de cablejat i facilitar les connexions.
- **Motors de les rodes** ⇒ El vehicle es mou mitjançant quatre rodes. Cada una està unida a un motor de corrent continu que en permet el seu moviment.
- **Servomotor** ⇒ És el component que permet el gir de l'emissor d'ultrasons. Pot girar un màxim de 180 graus i en el cas del robot *Voyager* podrà adoptar tres posicions fixes.
- **Motor pas a pas** ⇒ És un tipus de servomotor que ens permetrà girar la càmera. A diferència del servomotor, el motor pas a pas pot girar els 360 graus i pot adoptar fins a 8 posicions fixes. S'ha creat un suport dissenyat amb Tinkercad i imprès amb una impressora 3D. El suport permet mantenir el motor en vertical i adjuntar-hi la càmera. (Annex 1)
- **Bateria** ⇒ Està formada per dues bateries que tenen una forma semblant a la d'una pila. Cada pila aporta una tensió de 3'7 V, 7'4 V entre les dues.
- **Càmera** ⇒ Càmera de la marca Crosstour petita capacitada per realitzar vídeos i fotos. Consta de Wi-Fi que permet controlar-la a través d'una aplicació anomenada "iSmart DV" que es pot descarregar tant a la Play

Store com a l'Apple Store. Per tant, es necessita un altre dispositiu si es vol controlar la càmera a la mateixa vegada que es controla el vehicle.

- **Carcassa** ⇒ La carcassa del robot, juntament amb les dues peces blaves, ha estat dissenyada amb Tinkercad (programa per crear peces tridimensionals) i impresa mitjançant una impressora 3D. La funció de la carcassa és simplement decorativa mentre que les altres dues peces aguanten els components que formen el vehicle. (Annex 2)

A l'Annex 3 es troba una taula amb el pressupost que s'ha invertit per la construcció del vehicle.

Per tal que els components electrònics que formen el robot puguin funcionar de manera correcta i sincronitzada és necessari l'ús d'un software. En aquest projecte s'ha utilitzat el compilador d'Arduino per crear el programa que farà funcionar el robot i l'App Inventor per crear l'aplicació que permetrà dirigir des del mòbil al vehicle:

- **Compilador Arduino** ⇒ Permet crear un programa mitjançant el llenguatge C++. Entre altres coses, permet redefinir els objectes per ser més entenedors, incloure llibreries per realitzar determinades funcions i crear "voids" que generen funcions aïllades. El programa resultant es penja a l'Arduino UNO r3 mitjançant un cable USB.
- **App Inventor** ⇒ És un entorn de desenvolupament software que permet crear aplicacions per mòbils i tauletes mitjançant un llenguatge de blocs. L'aplicació resultant es pot instal·lar en format apk a qualsevol dispositiu Android.

4. DESCRIPCIÓ DELS COMPONENTS ELECTRÒNICS

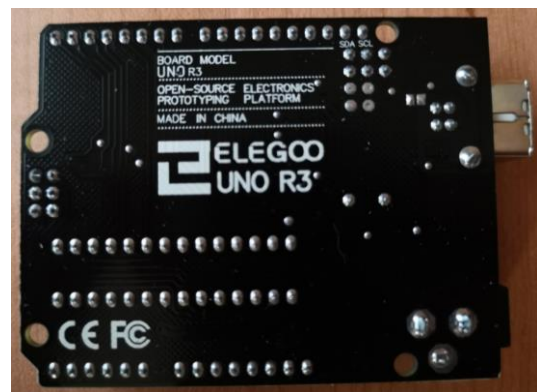
Els components electrònics que formen el robot són aquells que, mitjançant la programació, permeten que el robot pugui ser dirigit a través d'un mòbil, pugui seguir línies i que pugui esquivar obstacles. Així doncs, en aquest apartat es descriuran la placa Arduino UNO r3, el mòdul Bluetooth HC-05, l'emissor d'ultrasons HC-SR04 i el sensor d'infrarojos. També s'hi inclourà les plaques de connexió que són les que faciliten les connexions d'aquests components.

4.1. Arduino UNO r3 marca Elegoo

Arduino disposa d'una gran quantitat de plaques amb característiques diferents perquè s'esculli una que s'adapti millor a les circumstàncies necessàries. En aquest cas, s'ha decidit utilitzar la placa Arduino UNO r3 perquè és la que es recomana utilitzar quan es realitzen els primers projectes perquè disposa d'una quantitat d'entrades i sortides, tant analògiques com digitals, i d'una memòria interna suficient. La placa utilitzada a l'hora de muntar el robot no és de la marca Arduino sinó que és marca Elegoo. El motiu pel qual s'ha decidit utilitzar una placa de la marca Elegoo és simplement econòmic, però, així i tot, tenen la mateixa estructura, les mateixes característiques i utilitzen el mateix programa.



Imatge 9: Part superior de l'Arduino UNO r3

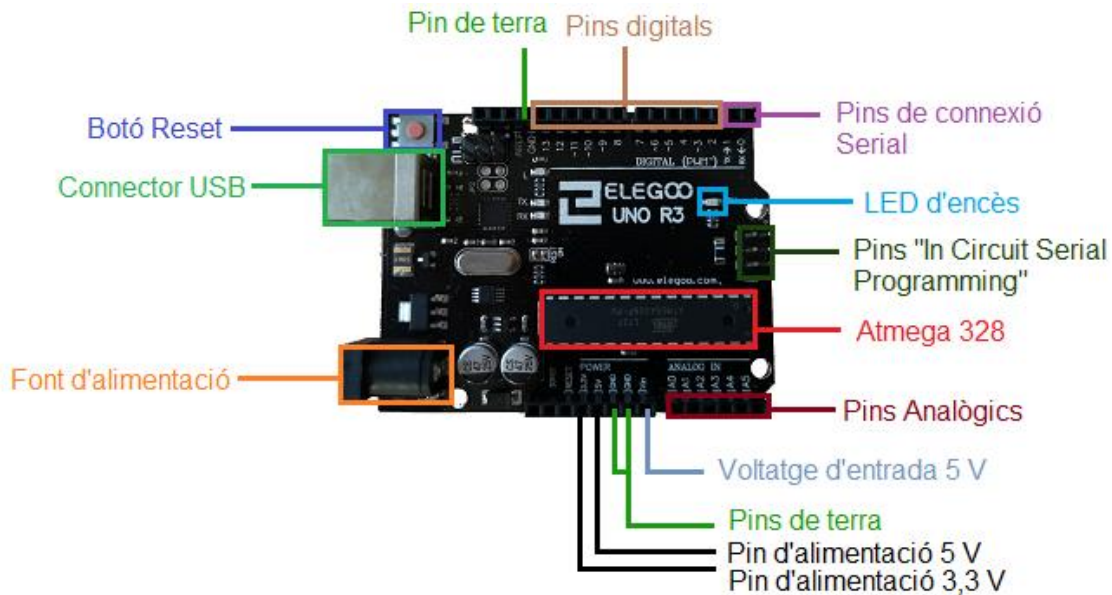


Imatge 10: Part inferior de l'Arduino UNO r3

La placa Arduino UNO r3 es basa en un microcontrolador Atmega328 on s'emmagatzemarà el programa. El voltatge d'entrada a la placa és d'entre 7 i

12 V; en el cas del vehicle *Voyager* és de 7'4 V. La placa està formada per 6 pins analògics i 14 pins digitals els quals suporten 40 mA i funcionen a 5 V.

Les principals parts de l'Arduino UNO r3 són les següents:



Imatge 11: Parts de l'Arduino UNO r3

- **Atmega 328** ⇒ Microcontrolador de 32 Kbytes de memòria flash que emmagatzema el programa.
- **Connector USB** ⇒ Permet connectar l'Arduino UNO r3 a l'ordinador per poder pujar-hi el programa. També subministra energia per fer funcionar la placa.
- **Font d'alimentació** ⇒ És la principal font d'energia a l'Arduino quan està desconnectat de l'ordinador. Així i tot, no és l'única, ja que hi ha altres pins d'alimentació que també poden realitzar aquesta funció.
- **Botó reset** ⇒ Permet reiniciar el programa que l'Arduino té carregat al microcontrolador. No l'esborra, sinó que el fa començar des del principi.

- **Pins digitals** ⇒ Són els pins que únicament poden processar voltatges d'entre 5 i 0 volts, és a dir, només poden rebre o transmetre informació en dos estats: HIGH o LOW (1 o 0 respectivament).
- **Pins analògics** ⇒ Són els pins que poden processar diferents valors de voltatge compresos entre 0 i 5 volts, és a dir, poden rebre o transmetre informació de diversos valors.
- **Pins ICSP** ⇒ Els 6 pins "In Circuit Serial Programming" permeten poder afegir un segon microcontrolador extern a l'Arduino UNO en el cas que l'Atmega 328 no tingui més espai.
- **Pins de connexió Serial** ⇒ Permeten l'intercanvi d'informació entre dos dispositius. En el cas del vehicle *Voyager* permeten la connexió entre el mòdul Bluetooth HC-05 i l'Arduino UNO r3. Els pins Serial comparteixen pin amb les entrades o sortides digitals 0 i 1. Al pin 0 s'hi troba l'entrada RX que permet rebre informació. D'altra banda, al pin 1 s'hi troba la sortida TX que té la funció de transmetre informació. Quan els pins 0 i 1 s'utilitzen amb la funció connexió Serial, no poden exercir la funció de pins digitals.

4.2. Bluetooth HC-05

És el component electrònic que possibilita la transmissió de dades entre diferents dispositius mitjançant una radiofreqüència, és a dir, sense la utilització d'un cablejat. En aquest projecte, permetrà la connexió entre el mòbil i el robot, les ordres que s'enviaran des del mòbil les executarà el robot gràcies al mòdul Bluetooth. La radiofreqüència que utilitza és de 2'4 GHz.

Els models més coneguts de mòdul Bluetooth són el HC-05 i el HC-06. A l'hora de planificar la construcció del robot es va decidir utilitzar el HC-06 perquè només funciona en mode esclau a diferència del HC-05 que també pot funcionar en mode mestre. El fet que només pugui funcionar en mode esclau

significa que només pot rebre connexions des d'un ordinador o un telèfon intel·ligent i això en facilita la programació. D'altra banda, el mode mestre també permet enviar senyals. Així i tot, la facilitat per aconseguir un mòdul Bluetooth va fer que s'hagués d'utilitzar el mòdul HC-05. Si més no, aquest canvi de planificació no va provocar cap incidència greu, ja que el HC-05 ja va programat com a mode esclau des de fàbrica i, per tant, tampoc cal configurar-lo de cap manera especial.

Tot i que els mòduls Bluetooth més coneguts i més utilitzats actualment són el HC-05 i el HC-06, els darrers anys, els fabricants han tret al mercat dos nous models de mòduls Bluetooth: el HC-08 i el HC-10. Aquests dos mòduls també es coneixen per les sigles BLE (Bluetooth Low Energy), ja que la principal característica que els diferencia dels dos models anteriors és que consumeixen menys energia. Aquest fet també provoca que la distància de control entre el mòdul Bluetooth i el dispositiu que li envia els senyals disminueixi a uns 50 metres. Una altra característica és que permet el funcionament de dispositius iOS. A part d'aquestes diferències el mòdul Bluetooth HC-08 correspondria a una versió millorada del HC-06 i el mateix passa amb el HC-10 i el HC-05.

En connectar el mòdul Bluetooth a el corrent s'observa que s'encén un LED vermell que fa llum intermitent. Quan el Bluetooth es connecta al mòbil, la llum intermitent es transforma en una llum contínua. Sempre que es vulgui pujar un programa a l'Arduino UNO, s'haurà de treure el mòdul Bluetooth per evitar que es faci malbé.

El mòdul Bluetooth HC-05 està format per 4 pins:

RXD⇒ rep la informació

TXD⇒ transmet la informació

GND⇒ es connecta al punt de terra

VCC⇒ es connecta a la font d'alimentació de 3'3V



Imatge 12: Part inferior del mòdul Bluetooth



Imatge 13: Part superior del mòdul Bluetooth

4.3. Emissor d'ultrasons HC-SR04

És un component electrònic que s'utilitza per mesurar distàncies. El seu funcionament es basa en l'emissió i recepció d'ultrasons, un mètode molt semblant al que utilitzen els ratpenats per ubicar-se. Un ultrasò és una vibració que té una freqüència superior a 20 KHz, que es produeix en un medi elàstic. Aquestes vibracions són inaudibles pels éssers humans.

L'emissor d'ultrasons és el que permet que el robot es mogui esquivant els obstacles. Per assolir aquest objectiu l'emissor d'ultrasons està programat de la següent manera:

1. El vehicle *Voyager* avança a una velocitat inferior de la normal per assegurar-se que tindrà temps a detectar l'obstacle.
2. Quan el vehicle detecta un obstacle, s'atura. El servomotor al qual està unit l'emissor d'ultrasons fa girar l'emissor a la dreta.
3. L'emissor calcula la distància que hi ha a la dreta del vehicle i fa el mateix amb el costat esquerre.
4. L'emissor realitza un test de distàncies, on les compara i, depenent del resultat, actua de les següents maneres:
 - Si la distància al costat esquerre és superior a la del costat dret, el vehicle gira a l'esquerre.
 - Si la distància al costat dret és superior a la del costat esquerra, el vehicle gira a la dreta.
 - Si ambdues distàncies són inferiors a 30 cm, el vehicle retrocedeix i gira a la dreta.

5. Un cop s'ha realitzat la maniobra, si el vehicle ja no detecta cap obstacle, avança. Si no és així, el vehicle torna a realitzar el test de distàncies.

La distància es mesura mitjançant la següent fórmula:

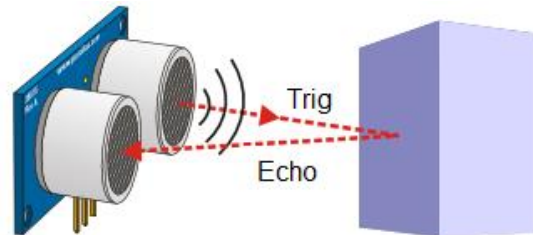
$$D = \frac{1}{2} \cdot v \cdot t$$

On:

$D \Rightarrow$ distància

$v \Rightarrow$ velocitat

$t \Rightarrow$ temps



Imatge 14: Representació del funcionament de l'emissor d'ultrasons

Autor: Daniel Arturo

En la construcció del robot s'ha utilitzat el model HC-SR04. Aquest model mesura distàncies entre 3 cm i 3 m. Està format per 4 pins: Vcc, Trig, Echo i GND. Trig i Echo són els utilitzats per mesurar les distàncies mentre que Vcc i GND són d'alimentació. L'emissor, Trig, emet un pols d'ultrasò que rebota en els objectes i torna al receptor, Echo, el qual emet un senyal d'entrada proporcional al temps que ha trigat a rebre el pols. D'altra banda, el pin Vcc es connecta a la font d'alimentació a 5 V i el pin GND es connecta al punt de terra.



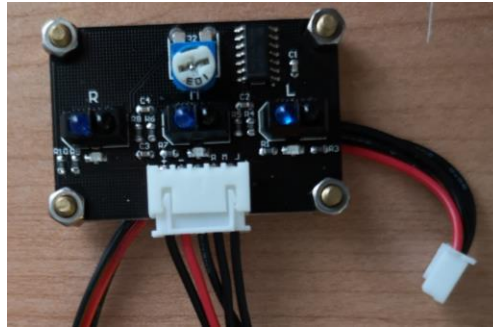
Imatge 15: Part davantera de l'emissor d'ultrasons



Imatge 16: Part del darrere de l'emissor d'ultrasons

4.4. Sensor d'infrarojos

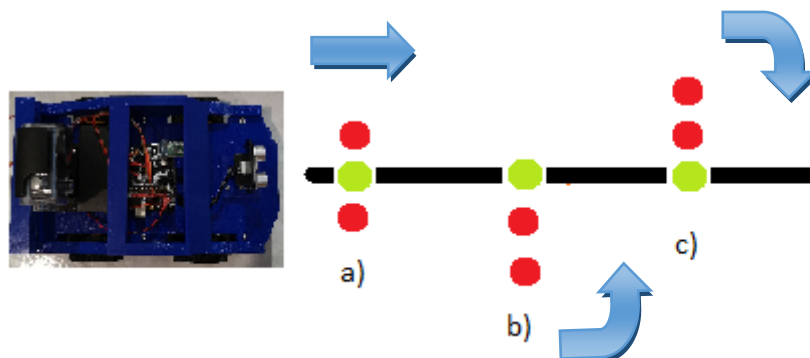
És el component electrònic que permet que el robot segueixi la línia negra. En el cas del robot, s'ha utilitzat una placa que conté 3 sensors d'infrarojos. La placa es troba a la part inferior davantera del vehicle i els sensors estan alineats paral·lelament a l'eix de les rodes.



Imatge 17: Placa que conté els tres sensors d'infrarojos

Cada sensor consta de dues parts: un tub emissor d'infraroig i un tub receptor d'infraroig. El tub emissor envia un infraroig que en rebotar a terra torna al receptor. La reflectància de la llum sobre una superfície negra és diferent de la d'una superfície blanca, d'aquesta manera el sensor sempre sap sobre quina superfície es troba. La placa inclou un potenciòmetre que permet regular la sensibilitat dels sensors.

El fet que la placa consti de tres sensors i no un permet que el moviment pugui ser rectilini. El seu funcionament és el següent:



Imatge 18: Representació gràfica del funcionament del mode seguidor de línies

- a) El sensor central detecta la línia recta i els sensors de la dreta i de l'esquerra no en detecten cap. El robot avança.
- b) El sensor de l'esquerra detecta la línia negra i els sensors de la dreta i el del centre no en detecten cap. El robot gira a l'esquerra.
- c) El sensor de la dreta detecta la línia negra i els sensors de l'esquerra i el del centre no en detecten cap. El robot gira a la dreta.

La placa consta de cinc pins. Els pins L, M i R envien la informació captada pels sensors de l'esquerra, el del centre i el de la dreta respectivament. El pin – es connecta al punt de terra mentre que el pin +5V es connecta a la font d'alimentació.



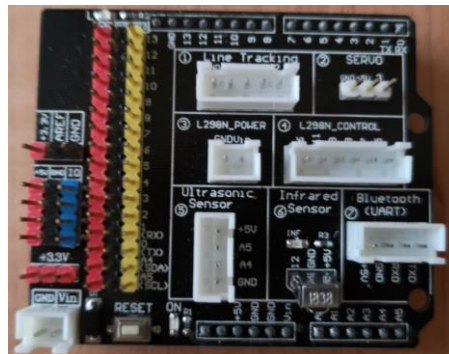
Imatge 19: Part del darrere de la placa que conté els sensors d'infrarojos

A l'hora de muntar el circuit amb la línia negra cal tenir en compte que l'amplada de la línia sigui suficientment gran perquè el sensor la pugui detectar i suficientment petita perquè no cobreixi dos sensors a la vegada. Pel mateix darrer motiu és necessari que les corbes no siguin molt tancades.

4.5. Plaques de connexió

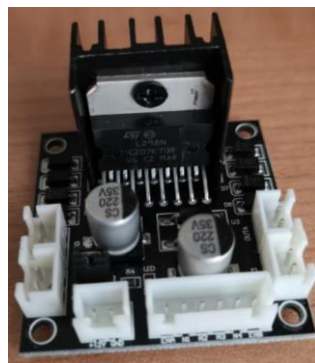
Amb l'objectiu de reduir el cablejat, facilitar les connexions i com a conseqüència obtenir un resultat més net, s'han utilitzat dues plaques de connexió. Una d'elles es troba a la planta alta del robot, a sobre de l'Arduino

UNO r3. El principal objectiu d'aquesta placa és facilitar les connexions amb la placa Arduino UNO r3.



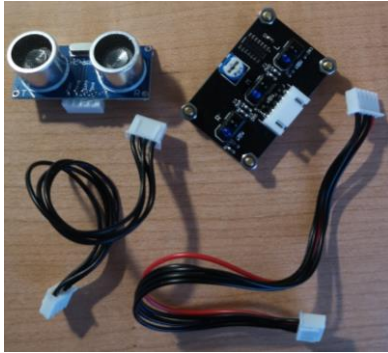
Imatge 20: Placa de connexió que va col·locada sobre l'Arduino UNO

L'altra placa es troba a la planta baixa a la zona central. Aquesta placa, a banda de realitzar la funció de reduir el cablejat, consta de 2 ponts H i d'un regulador de tensió. Els ponts H, formats per 4 transistors cadascun, permeten controlar la velocitat i el sentit dels 4 motors que mouen al vehicle. El regulador de voltatge permet reduir la tensió d'entrada que és de 7'4 V a 5 V, que és la tensió necessària per al correcte funcionament dels motors de corrent continu i de l'Arduino UNO.



Imatge 21: Placa de connexió que uneix el cablejat dels quatre motors

Per tal de facilitar les connexions és necessari que els cables i els diferents components electrònics disposin d'uns adaptadors.



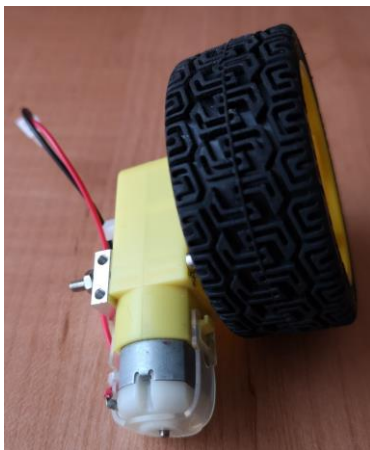
Imatge 22: Emissor d'ultrasons i sensor d'infrarojos amb els seus respectius cables que porten incorporats unes peces blanques per facilitar les connexions

5. DESCRIPCIÓ DE LA MECÀNICA, SISTEMA ENERGÈTIC I ACCESSORIS DEL ROBOT

A banda dels components electrònics, el robot consta d'altres sistemes que permeten que els components electrònics puguin realitzar correctament el seu funcionament. Aquests sistemes són els que formen els motors de les rodes, el servomotor de l'emissor d'ultrasons, el motor pas a pas de la càmera i la bateria.

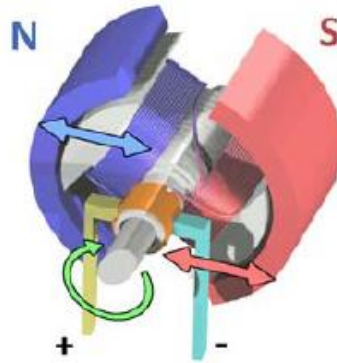
5.1. Motors

El robot es pot moure gràcies a quatre rodes de goma, dues en cada costat. Cada roda està unida a un motor de corrent continu que es troba a la planta de baix.



Imatges 23 i 24: Motor de corrent continu amb la roda de goma

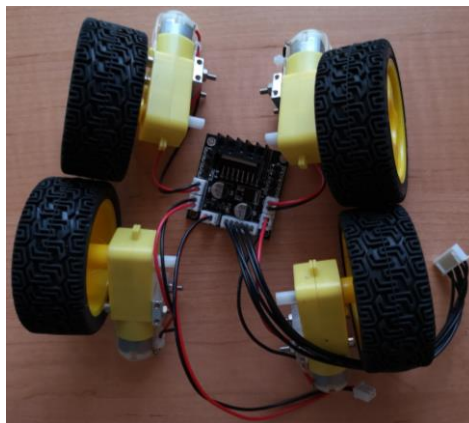
Aquests motors transformen l'energia elèctrica en energia mecànica. Aquesta transformació d'energia és possible gràcies a un imant i a un electroimant situats a l'interior del motor. L'electroimant es troba al mateix eix de la roda i exerceix la funció d'imant quan rep energia elèctrica. L'imant es troba en forma de circumferència al voltant de l'electroimant. Els pols de l'electroimant i els de l'imant coincideixen i, per tant, es repel·leixen, fet que provoca la rotació de l'electroimant, de l'eix i de la roda.



Imatge 25: Gràfic del funcionament d'un motor de corrent continu

Autor: Wapcaplet

Tot i tenir les rodes col·locades com un cotxe, el robot no es mou com un cotxe. La diferència es troba a l'hora de girar, ja que cada roda té el seu motor i no comparteixen el mateix eix. Així doncs, si es vol que el cotxe giri cap a l'esquerra, el parell de rodes situades a l'esquerra del vehicle giraran cap al darrere mentre que el parell de rodes situades a la dreta giraran cap al davant. Si es vol girar a la dreta, el gir de les rodes es produirà al revés de si es vol girar a l'esquerra.



Imatge 26: 4 motors del vehicle Voyager juntament amb les rodes

5.2. Servomotor

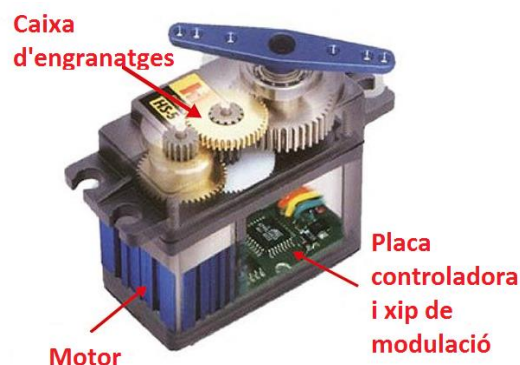
El servomotor és el component que permet el gir de l'emissor d'ultrasons. També és un motor de corrent continu però es diferencia en el fet que tan sols pot girar 180 graus i es pot mantenir estable en una determinada posició.

La connexió del servomotor es fa mitjançant tres cables: el marró es connecta al punt de terra, el vermell a l'alimentació de 5V i el taronja és el que permet enviar el senyal.



Imatge 27: Servomotor

El servomotor consta d'un xip de modulació de senyal, d'una placa controladora, d'un circuit intern, d'un motor i d'una caixa d'engranatges. La placa controladora subministra una tensió bàsica al xip de modulació. El circuit intern, en el qual hi ha un potenciòmetre que permet determinar l'angle actual del motor, també subministra una tensió estàndard al xip de modulació. Aquests dos voltatges es comparen entre ells i la diferència arribarà al motor. Depenent del voltatge d'arribada el motor girarà en un sentit o en un altre en un angle determinat.



Imatge 28: Parts d'un servomotor

Autor: PanamaHitek

Per controlar la rotació, d'entrada es necessari un pols de temps comprès entre 0'5 ms i 2'5 ms. Depenent de l'amplada del pols el motor girarà uns determinats graus:

0'5 ms	0 graus
1 ms	45 graus
1'5 ms	90 graus
2 ms	135 graus
2'5 ms	180 graus

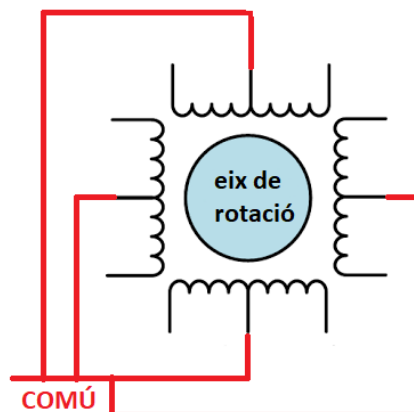
5.3. Motor pas a pas

És el component que permetrà el gir de la càmera. Es tracta del model 28BYJ-48 que porta incorporat un adaptador amb un microxip ULN2003A. Aquest adaptador es pot controlar amb un voltatge d'entre 5 i 12 V, fet que ens el permet connectar amb l'Arduino UNO.



Imatge 29: Motor pas a pas amb l'adaptador que conté el microxip ULN2003A

El motor pas a pas es caracteritza per poder regular amb molta precisió la posició de l'eix. Aquesta precisió és gràcies a les quatre bobines situades a l'interior del motor. El model 28BYJ-48 és un motor unipolar, és a dir, que les quatre bobines són alimentades per un mateix punt comú.



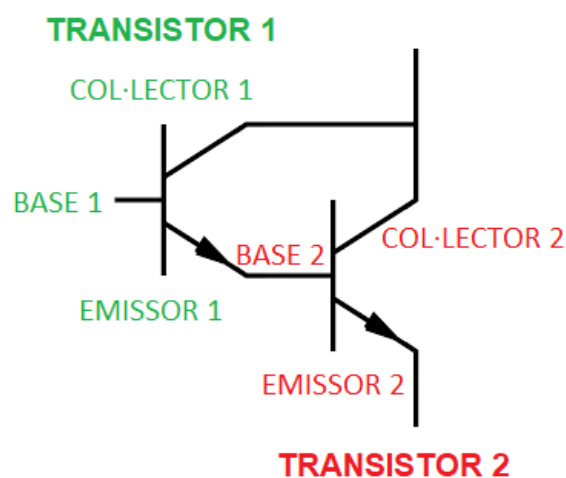
Imatge 30: Esquema de l'interior d'un motor pas a pas

D'altra banda, cada bobina del motor s'activa amb un mínim de 55 mA, per tant, no es poden connectar directament a un pin de l'Arduino UNO. Per aquest motiu, és necessària la utilització del xip ULN2003A incorporat a l'adaptador. Aquest xip consisteix en una matriu de transistors enllaçats mitjançant una connexió Darlington que permetrà augmentar la intensitat de sortida per tal que la placa Arduino UNO pugui alimentar el motor pas a pas. L'adaptador i el xip ULN2003A es connecten de la següent manera:



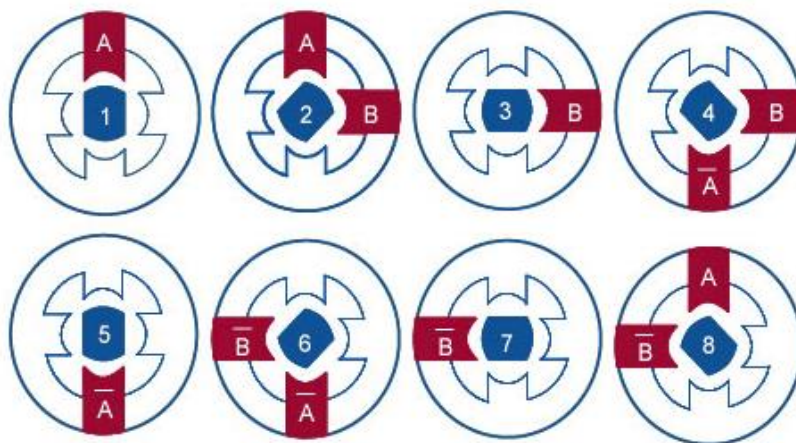
Imatge 31: Adaptador i microxip ULN2003A

Un transistor és un component electrònic que s'utilitza com a amplificador. Els transistors consten de tres terminals: col·lector, base i emissor. La connexió Darlington es realitza amb la unió de dos transistors i consisteix a unir l'emissor d'un primer transistor amb la base d'un segon transistor. La intensitat de sortida de l'emissor del segon transistor s'haurà multiplicat. Si més no, el voltatge haurà augmentat el doble.



Imatge 32: Gràfic de la connexió Darlington

En motor pas a pas és capaç de mantenir-se estable en vuit posicions determinades. Aquestes posicions són marcades mitjançant la combinació de l'activació de les quatre bobines. Aquestes són les posicions que pot adoptar l'eix del motor i quines bobines s'han d'activar en cada cas per assolir aquella posició:



Imatge 33: Posicions que pot adoptar l'eix d'un motor pas a pas

Autor: Enrique Crespo

Per tal de poder mantenir el motor pas a pas en vertical per poder aguantar la càmera s'ha dissenyat i imprès amb una impressora 3D un suport en forma de cub amb un forat rodó, on s'insereix el motor pas a pas. Al costat esquerre del cub s'hi ha afegit l'adaptador.

5.4. Sistema de gestió de l'energia

El robot consta d'un sistema de gestió de l'energia per tal que es pugui moure de forma autònoma, sense estar connectat a un ordinador mitjançant un cable. Aquest sistema està format per dues bateries de 3'7 V cada una connectades a l'Arduino UNO. Aquestes bateries es poden carregar. Per tal de facilitar la connexió de les bateries s'ha utilitzat una capsa que, fins i tot, disposa d'un botó per encendre i apagar.

Voyager



Robot controlat des del mòbil



Imatges 34 i 35: Capsa amb les dues bateries

6. DESCRIPCIÓ DEL SOFTWARE DESENVOLUPAT

Per tal que tots els components que constitueixen el robot puguin exercir la seva funció i, per tant, permetre el correcte funcionament del robot, és necessari la creació d'un software. En el cas d'aquest projecte, el software es divideix en dues parts: el compilador de l'Arduino i l'App Inventor.

Al compilador d'Arduino s'hi ha creat el programa on hi ha definides les funcions que han de realitzar els diferents components mentre que, amb l'App Inventor s'ha creat l'aplicació que permet controlar el robot des d'un mòbil.

6.1. Programació de l'Arduino

El programa creat amb l'Arduino és molt extens. En aquest apartat s'hi afegirà la part més important del codi, el "void loop", és a dir, les instruccions que es van realitzant de forma continuada, com un bucle. El programa complet amb els respectius comentaris es troba a l'Annex 4.

```
void loop() {  
  
  // Si tenim alguna cosa pel port Serial / Bluetooth  
  if (Serial.available() > 0) {  
  
    // Llegim el caràcter del port Serial / Bluetooth i l'emmagatzemem a la variable getstr  
    getstr = Serial.read();  
    Serial.write(getstr);  
  
    // Gestionem les accions que haurà de realitzar el cotxe mitjançant un switch case  
    switch (getstr) {  
      case 'a': avansa(0);  
        break;  
  
      case 'r': retrocedeix();  
        break;  
  
      case 'e': girEsquerra(0);  
        break;  
  
      case 'd': girDreta(0);  
        break;  
  
      case 'p': parar();  
        break;  
  
      case 'l': do {  
        flag = 1; modeSeguidorLinies();  
        if (Serial.available() > 0) {  
          flag = 0;  
          // Llegim el caràcter del port Serial / Bluetooth i l'emmagatzemem a la variable getstr  
          getstr = Serial.read();  
        }  
      } while (flag == 1);  
      break;  
  
      case 'u': do {  
        flag = 1; modeUltrasons();  
        if (Serial.available() > 0) {  
          flag = 0;  
          // Llegim el caràcter del port Serial / Bluetooth i l'emmagatzemem a la variable getstr  
          getstr = Serial.read();  
        }  
      } while (flag == 1);  
      break;  
  
      case 'y': cameraGirEsquerra(511);  
        break;  
  
      case 'x': cameraGirDreta(511);  
        break;  
  
      default: break;  
    }  
  }  
}
```

Imatge 36: Captura de pantalla del “void loop” del programa Arduino

Les primeres instruccions que trobem dins del “void loop” ordenen la lectura del senyal que s’envia des de l’aplicació. Tal com s’explica al següent apartat, aquest senyal arriba en forma de lletres. Per aquest motiu s’ha creat un “switch case” que forma l’altra part d’instruccions que hi ha al “void loop”.

Aquest “switch case” porta incorporat totes les lletres que es poden rebre des de l’aplicació. Cada lletra activa una funció, d’aquesta manera, quan el mòdul Bluetooth rebí la lletra “a”, s’iniciarà la funció que activa la lletra “a”. El mateix amb la resta de lletres.

Tal com es pot veure al “void loop”, hi ha dues lletres que activen dues funcions més extenses que les altres. És el cas de les lletres “l” i “u” que activen les instruccions necessàries que permeten el funcionament del mode seguidor de línies i del mode emissor d’ultrasons respectivament. El motiu pel qual tenen una funció més extensa és perquè es necessita que el bloc d’instruccions s’executi continuadament fins que es rebí una altre senyal, a diferència de la resta de funcions que només s’executen un cop. Així doncs, s’han incorporat uns “flags” que permeten que un bloc d’instruccions s’executi sempre que el valor rebut sigui 1 (la lletra que els acciona) i que es deixi d’executar quan el valor rebut sigui 0 (cap o qualsevol altre lletra que no sigui la que l’acciona)

6.2. Programació de l’aplicació

Per tal de poder dirigir el robot des del mòbil, s’ha creat una aplicació amb App Inventor, un entorn de desenvolupament que permet crear aplicacions mitjançant blocs.

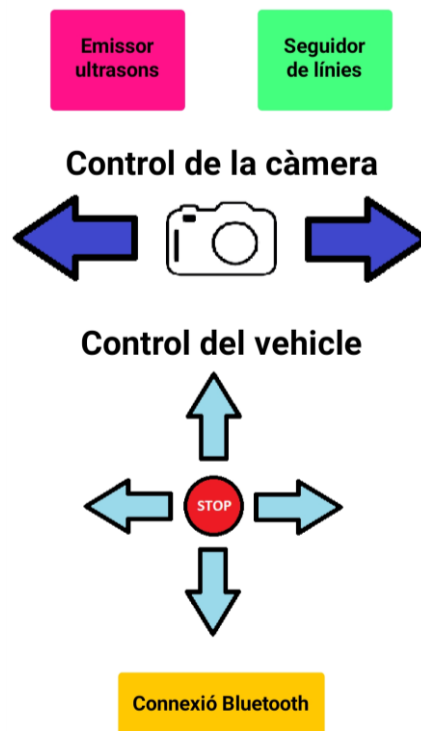
L’aplicació té el mateix nom que el vehicle i aquest és l’aspecte que té el logotip quan es veu amb totes les altres aplicacions del mòbil:



VOYAGER

Imatge 37: Logotip de l'aplicació

Un cop s'entra a l'aplicació apareix una pantalla que és la que permet dirigir el vehicle *Voyager*:



Imatge 38: Disseny de l'aplicació

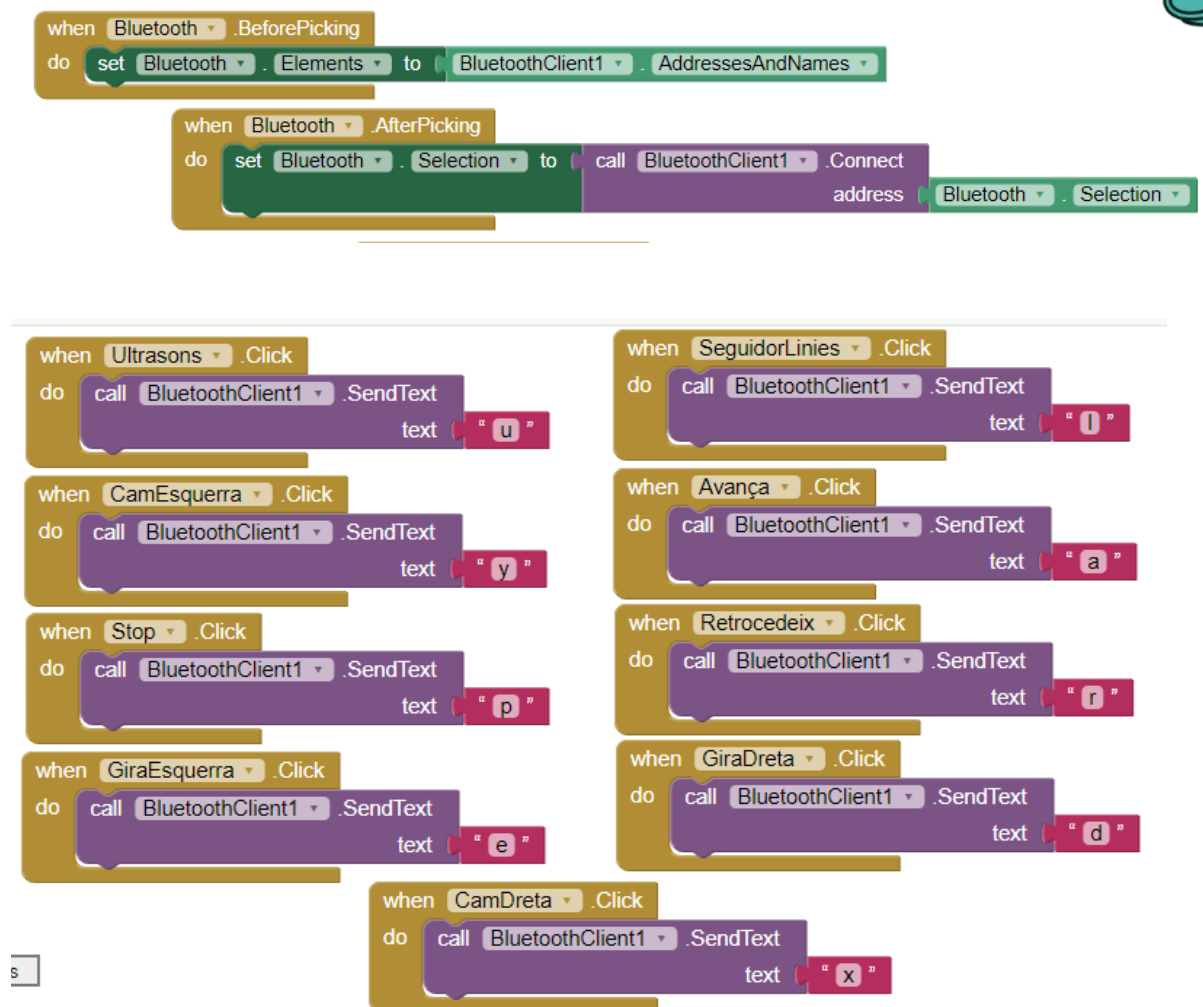
El primer que s'ha de fer a l'entrar a l'aplicació és enllaçar-ho al vehicle. Per fer-ho cal pressionar el botó "Connexió Bluetooth" i escollir el Bluetooth del robot. En aquest cas, el Bluetooth que cal escollir d'entre tots els Bluetooth que tenim enllaçats amb el mòbil és el que s'anomena HC-05.



Imatge 39: Llistat de Bluetooth connectats al mòbil

Un cop s'ha seleccionat el Bluetooth ja es poden enviar totes les ordres al robot pressionant els diferents botons. L'aplicació ens permet controlar el robot mitjançant les fletxes de color blau fluix, controlar la càmera mitjançant les fletxes de color blau fort i activar els modes seguidor de línies i emissor d'ultrasons a través del seu respectiu botó.

Per tal que l'aplicació funcionés, s'ha hagut de crear el següent programa:



Imatges 40 i 41: Blocs que programen l'aplicació

Tal com es veu, els dos primers blocs, situats a la part superior de la imatge, són els que permeten seleccionar el Bluetooth. La resta de blocs fan referència a cada botó que forma l'aplicació. La funció de cada bloc és que, en prémer el corresponent botó, mitjançant la connexió Bluetooth s'envia una determinada lletra a l'Arduino UNO r3 del vehicle perquè es realitzi una determinada funció que conté el programa Arduino emmagatzemat. Per exemple, en prémer el botó "Seguidor de línies" s'enviarà la lletra "l" i el programa Arduino activarà la funció "modeSeguidorLinies".

7. ANÀLISI I CONCLUSIONS

El resultat del projecte ha estat el que desitjava. He construït un robot que m'ha permès aprendre a programar per realitzar determinades funcions, com per exemple, controlar un motor pas a pas, un sensor d'infrarojos, un servomotor, un emissor d'ultrasons, els quatre motors de les rodes, etc. Per tal d'assolir aquest resultat he hagut de fer front a diversos problemes que m'han anat sorgint mentre elaborava el treball:

- **Motor pas a pas** ⇒ Un cop vaig programar el motor pas a pas, la programació del robot ja estava acabada. Vaig realitzar la programació i el resultat va ser favorable tot i que hi havia vegades que el motor no responia adequadament. En posar en funcionament el robot per gravar-lo, el motor pas a pas no funcionava. Havia sorgit un problema i era necessari detectar-lo per saber si s'havia d'actuar al programa o als components electrònics. El problema el vaig detectar als components electrònics, ja que amb el programa realitzat, hi va haver moments que havia funcionat. D'aquesta manera vaig comprar un motor pas a pas nou però el resultat va ser negatiu. Després d'hores buscant el problema vaig arribar a la conclusió que el culpable era l'estat dels cables. Així doncs, vaig comprar nous cables i, afortunadament, el resultat va ser positiu.
- **El “void loop” del programa Arduino** ⇒ Quan s'entra al compilador d'Arduino hi apareixen dos “voids”: el “void setup”, on es configuren les funcions dels pins, i el “void loop”, on s'elabora el programa que es vol realitzar en forma de bucle, és a dir, constantment. Si més no, el programa necessari per controlar el robot necessita diferents subprogrames per realitzar cadascuna de les funcions que pot executar el robot, és a dir, si es vol activar el mode seguidor de línies només s'ha d'executar en forma de bucle el subprograma del mòdul seguidor de línies i no tot el programa que té emmagatzemat l'Arduino UNO. Aquest fet em va suposar més feina, ja que l'Arduino UNO només pot emmagatzemar un sol programa i aquest programa només pot tenir un

“void loop”. D’aquesta manera vaig haver de buscar altres recursos com és el cas de “flags”.

- **El mòdul Bluetooth** ⇒ La caixa a partir de la qual vaig realitzar modificacions per obtenir el treball resultant, portava incorporada el mòdul Bluetooth HC-08. En intentar connectar aquest mòdul amb l’aplicació creada amb App Inventor, el robot no responia. Després d’investigar, vaig veure que aquest mòdul anava codificat perquè només pogués rebre ordres d’una aplicació que també portava incorporada la caixa anomenada “ELEGOO BLE Tool”. D’aquesta manera vaig haver de comprar un nou mòdul Bluetooth, en aquest cas el HC-05. A aquest mòdul l’hi vaig haver d’afegir una peça blanca que permet la connexió amb la placa. En instal·lar aquest mòdul a l’Arduino UNO, el robot ja podia ser controlat per l’aplicació creada amb App Inventor.

Tot i que el resultat del projecte és positiu, hi ha una sèrie de millores que a causa d’una falta de recursos i de temps, permetrien millorar el rendiment del vehicle. Aquestes millores es poden realitzar de cara al futur i són les següents:

- **El mode d’esquivar els obstacles** ⇒ Actualment el robot consta d’un emissor d’ultrasons que es troba a una alçada d’uns deu centímetres del terra. La millora consistiria a afegir-hi més emissors d’ultrasons i sincronitzar-los. D’aquesta manera el vehicle no xocaria mai (independentment de si l’obstacle té forma corba o no queda perpendicular a la direcció del vehicle). Una altra possible millora, en aquest sentit, seria baixar l’alçada en la qual es troba l’emissor perquè, actualment, tots aquells obstacles inferiors als deu centímetres no són detectables.
- **El mode seguidor de línies** ⇒ El gran avantatge que té el vehicle a l’hora de seguir línies és que consta de tres sensors d’infrarojos per detectar les línies. Aquest fet permet que el moviment del robot en línies rectes sigui més fluid, sense fer ziga-zaga. Si més no, aquest avantatge

també suposa un inconvenient: a l'hora de fer les corbes, si dos sensors queden sobre la línia negra, el robot s'atura. La solució consistiria a augmentar la separació entre els sensors d'infrarojos.

- **El control del vehicle** ⇒ Actualment, si es vol controlar el vehicle i la càmera a la vegada, són necessaris dos dispositius. La millora consistiria a adjuntar el control de la càmera juntament amb una pantalla on es vegi el que es grava a l'aplicació creada per controlar el robot. Aquesta millora requeriria la sincronització del mòdul Bluetooth del vehicle i del Wi-Fi de la càmera per poder-ho connectar a un sol dispositiu. D'altra banda, si es vol incorporar una pantalla on es vegi tot el que es grava, seria necessària la utilització d'un programa diferent de l'App Inventor que tingui més recursos.

L'elaboració d'aquest treball m'ha permès posar en pràctica coneixements apresos anteriorment, tant a l'ESO com al batxillerat, i aprendre'n de nous. Entre els coneixements que ja havia adquirit abans de realitzar el treball i que he aplicat destaquen els següents:

- **Tinkercad** ⇒ El funcionament d'aquesta eina me la van explicar a l'assignatura de dibuix tècnic de primer de batxillerat i m'ha permès tenir més agilitat a l'hora de fer els dissenys en 3D. La impressora que he utilitzat és el model SIGMAX de l'empresa BCN3D.
- **App Inventor** ⇒ Aquest entorn de desenvolupament software me'l van introduir a l'assignatura d'informàtica de quart d'ESO. Juntament amb els coneixements adquirits al curs EstiURV, m'ha permès que el procés de creació de l'aplicació hagi sigut més ràpid i el resultat més bo.
- **GIMP** ⇒ El funcionament d'aquest programa me'l van explicar a l'assignatura d'informàtica de quart d'ESO. Aquest programa disposa de moltes eines per poder editar imatges i l'he utilitzat en moltes de les imatges que es troben al cos del treball.

- **El funcionament d'alguns components electrònics** ⇒ Com és el cas de l'emissor d'ultrasons i el sensor d'infrarojos que els vaig estudiar a quart d'ESO a l'assignatura de tecnologia.

Entre els coneixements nous adquirits destaquen els següents:

- **Arduino** ⇒ Tot i tenir una petita noció apresada al curs EstiURV sobre aquest programa, l'elaboració d'aquest treball m'ha permès especialitzar-me en el funcionament de la placa Arduino UNO i, sobretot, aprendre l'ús del llenguatge Arduino per realitzar determinats programes.
- **El funcionament d'alguns components electrònics** ⇒ Com és el cas del mòdul Bluetooth, el servomotor i el motor pas a pas.
- **Com detectar un problema** ⇒ El fet que el robot l'hagi hagut de programar i muntar jo mateix, ha potenciat que davant d'una incidència hagués de detectar i distingir l'arrel del problema, ja que es podia tractar d'un problema electrònic o del programa. Un cop ja detectava el problema, ja sabia on i com havia d'actuar.

8. BIBLIOGRAFIA I WEBGRAFIA

Amazon (1996-2018) [en línia] <<https://www.amazon.com>> [consultat el dia 21/06/2018]

Arduino (2018) [en línia] <<https://www.arduino.cc>> [consultat el dia 21/06/2018]

Arduino.cl [en línia] <<http://arduino.cl>> [consultat el dia 24/07/2018]

ARMESTO, Leopoldo (2017) *Diseña, fabrica y programa tu propio robot* [en línia] <<http://mooc.es/course/disena-fabrica-y-programa-tu-propio-robot>> [consultat el dia 01/05/2018]

CRESPO, Enrique (2018) *Aprendiendo Arduino. Aprendiendo a manejar Arduino profundamente* [en línia] <<https://aprendiendoarduino.wordpress.com>> [consultat el dia 01/05/2018] CC BY NC SA 4.0

CUÉ, Carlos i REBOSSIO, Alejandro (28 juliol 2015). Científicos contra robots armados. *El País*. [consultat el dia 4/09/2018]
<https://elpais.com/internacional/2015/07/28/actualidad/1438078885_287962.html>

DE LA FUENTE, Santiago (2016) Partes de un Arduino Uno [en línia] <<https://www.slideshare.net/SantiagodelaFuenteBl/partes-de-un-arduino-uno>> [consultat el dia 14/10/2018]

DEL VALLE, Luis (2018) *Programarfacil* [en línia] <<https://programarfacil.com>> [consultat el dia 21/06/2018]

Diotronic S.A. Componentes Electrónicos [en línia] <<https://diotronic.com>> [consultat el dia 5/07/2018]

ELEGOO (2011-2018) [en línia] <<https://www.elegoo.com>> [consultat el dia 5/07/2018]

GONZÁLEZ, David *Blanchard Space* [en línia]
<<https://blanchardspace.wordpress.com>> [consultat el dia 10/10/2018]

Guille *paperblog* [en línia] <<https://es.paperblog.com/arduino-tutorial-22-control-de-un-motor-paso-a-paso-28-byj-48-4072450/>> [consultat el dia 14/09/2018]

MACHO, Juan Carlos (2018) *Prometec* [en línia] <<https://www.prometec.net>>
[consultat el dia 01/05/2018]

Naylamp. Mechatronics [en línia] <<https://naylampmechatronics.com>> [consultat el dia 24/10/2018]

Polarity Reus (2018) [en línia] <<http://polarityreus.com>> [consultat el dia 21/06/2018]

UPV *Wiki de Robótica* [en línia] <<http://wiki.robotica.webs.upv.es>> [consultat el dia 3/09/2018]

Viquipèdia. L'enciclopèdia lliure (2018) [en línia] <<https://ca.wikipedia.org>>
[consultat el dia 26/07/2018] CC BY SA

Vix (2015 - 2018) [en línia] <<https://www.vix.com>> [consultat el dia 26/07/2018]

Xataka [en línia]<<https://www.xataka.com>> [consultat el dia 24/07/2018]

9. AGRAÏMENTS

M'agradaria donar les gràcies a totes aquelles persones que m'han ajudat, m'han aconsellat i m'han resolt dubtes al llarg del treball, ja que sense elles, no hagués estat possible la construcció i programació del robot *Voyager*.

- A la meva tutora del treball de recerca, pels consells que m'ha donat al llarg del treball.
- A l'institut, per haver-me deixat material Arduino per iniciar-me al món de la programació.
- Al meu tiet, per haver-me deixat la impressora 3D i per haver-me introduït el món de la programació.
- Als meus pares, pels seus consells i guiatge.

10. ÍNDEX D'IL·LUSTRACIONS

Imatge 1 (pàg 4): Robot industrial que realitza soldadures. Autor: Phasmatisnox

Imatge 2 (pàg 4): Robot de servei que neteja recintes tancats autònomament.
Autor: Guzugi

Imatge 3 (pàg 5): Robot militar. Autor: Army Images

Imatge 4 (pàg 5): Cirurgians de Keesles realitzen la primera cirurgia robòtica.
Autor: Jenay Randolph

Imatge 5 (pàg 6): Nano robots. Autor: Nanotecnología Médica

Imatge 6 (pàg 6): Robot educatiu LEGO Mindstorms. Bernard Goldbach

Imatge 7 (pàg 7): Joguina robòtica. Alexas_Fotos

Imatge 8 (pàg 10): Estructura del vehicle *Voyager*

Imatge 9 (pàg 13): Part superior de l'Arduino UNO r3

Imatge 10 (pàg 13): Part inferior de l'Arduino UNO r3

Imatge 11 (pàg 14): Parts de l'Arduino UNO r3

Imatge 12 (pàg 16): Part inferior del mòdul Bluetooth

Imatge 13 (pàg 16): Part superior del mòdul Bluetooth

Imatge 14 (pàg 18): Representació del funcionament de l'emissor d'ultrasons.
Autor: Daniel Arturo

Imatge 15 (pàg 18): Part davantera de l'emissor d'ultrasons

Imatge 16 (pàg 18): Part del darrere de l'emissor d'ultrasons

Imatge 17 (pàg 19): Placa que conté els tres sensors d'infrarojos

Imatge 18 (pàg 19): Representació gràfica del funcionament del mode seguidor de línies

Imatge 19 (pàg 20): Part del darrere de la placa que conté els sensors d'infrarojos

Imatge 20 (pàg 21): Placa de connexió que va col·locada sobre l'Arduino UNO

Imatge 21 (pàg 21): Placa de connexió que uneix el cablejat dels quatre motors

Imatge 22 (pàg 22): Emissor d'ultrasons i sensor d'infrarojos amb els seus respectius cables que porten incorporats unes peces blanques per facilitar les connexions

Imatges 23 i 24 (pàg 23): Motor de corrent continu amb la roda de goma

Imatge 25 (pàg 24): Gràfic del funcionament d'un motor de corrent continu.
Autor: Wapcaplet

Imatge 26 (pàg 24): 4 motors del vehicle *Voyager* juntament amb les rodes

Imatge 27(pàg 25): Servomotor

Imatge 28 (pàg 25): Parts d'un servomotor. Autor: Panama Hitek

Imatge 29 (pàg 27): Motor pas a pas amb l'adaptador que conté el microxip ULN2003A

Imatge 30 (pàg 27): Esquema de l'interior d'un motor pas a pas

Imatge 31 (pàg 28): Adaptador i microxip ULN2003A

Imatge 32 (pàg 28): Gràfic de la connexió Darlington

Imatge 33 (pàg 29): Posicions que pot adoptar l'eix d'un motor pas a pas. Autor: Enrique Crespo

Imatges 34 i 35 (pàg 30): Capsa amb les dues bateries

Imatge 36 (pàg 32): Captura de pantalla del "void loop" del programa Arduino

Imatge 37 (pàg 34): Logotip de l'aplicació

Imatge 38 (pàg 34): Disseny de l'aplicació

Imatge 39 (pàg 35): Llistat de Bluetooth connectats al mòbil

Imatges 40 i 41 (pàg 36): Blocs que programen l'aplicació

Imatge 42 (annex I): Disseny amb Tinkercad dels suports del motor pas a pas

Imatge 43 (annex II): Disseny amb Tinkercad del xassís del vehicle

Imatges 44-55 (annex IV): Captures de pantalla del programa Arduino

11. LLISTAT D'ABREVIACIONS I SÍMBOLS

.apk ⇒ Android Application Package

3D ⇒ Tres dimensions

BLE ⇒ Bluetooth Low Energy

cm ⇒ Centímetre

ESO ⇒ Educació Secundària Obligatoria

GHz ⇒ Gigahertz

GND ⇒ Ground

ICSP ⇒ In Circuit Serial Programming

Kbyte ⇒ Quilobyte

KHz ⇒ Quiloherz

L ⇒ Left

LED ⇒ Light Emitting Diode

m ⇒ Metre

M ⇒ Middle

mA ⇒ Mil·liAmperes

ms ⇒ Mil·lisegon

Voyager

Robot controlat des del mòbil

R ⇨ Right

R.U.R. ⇨ Robots Universals de Rossum

RXD ⇨ Recepció de Dades

SERC ⇨ Science and Engineering Research Council

TXD ⇨ Transmissió de Dades

URV ⇨ Universitat Rovira i Virgili

USB ⇨ Universal Serial Bus

V ⇨ Volt

Vcc ⇨ Voltatge corrent continu

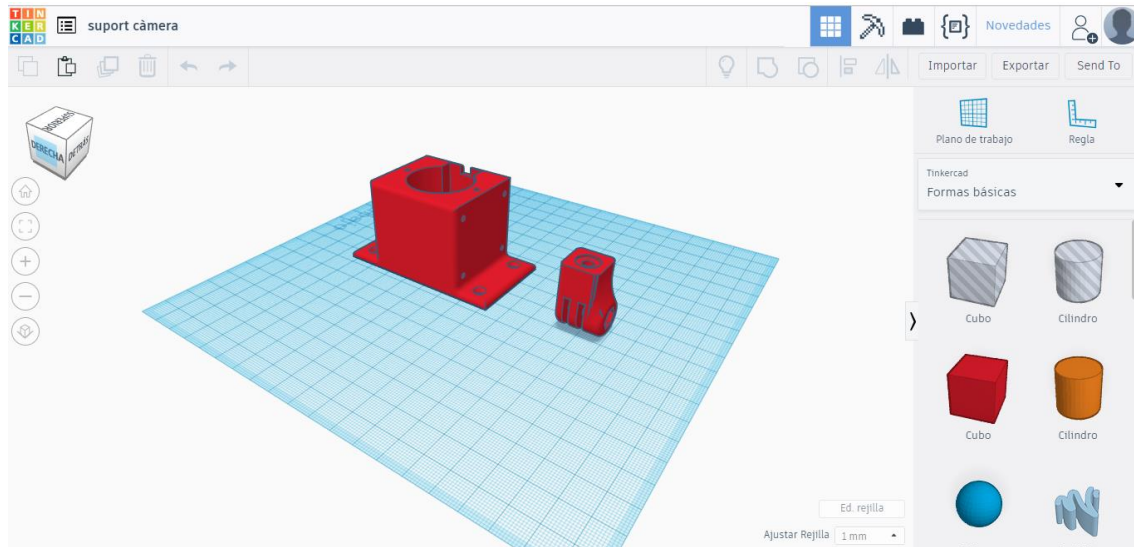
Wi-Fi ⇨ Wireless Fidelity

12. ANNEXOS

ÍNDIX D'ANNEXOS

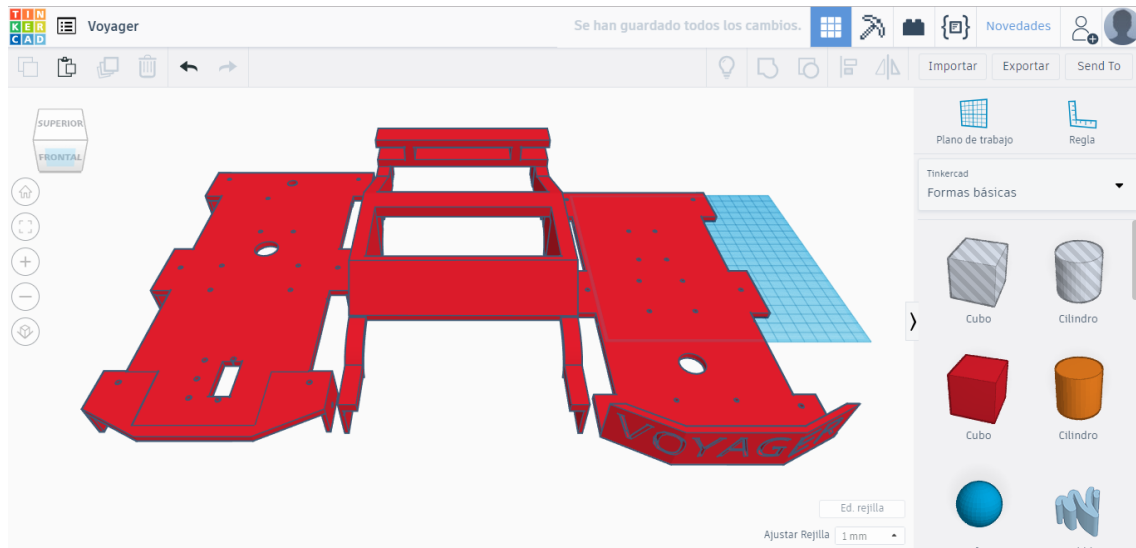
<i>Annex 1 Suport del motor pas a pas</i>	I
<i>Annex 2 Xassís del vehicle</i>	II
<i>Annex 3 Pressupost</i>	III
<i>Annex 4 Codi Arduino</i>	IV

Annex 1: Suport del motor pas a pas



Imatge 42: Disseny amb Tinkercad dels suports del motor pas a pas

Annex 2: Xassís del vehicle



Imatge 43: Disseny amb Tinkercad del xassís del vehicle

Annex 3: Pressupost

En aquesta taula s'hi reflecteix el preu de tot el que s'ha hagut de comprar per fer possible la construcció del robot:

Producte	Quantitat	Preu
Smart V3.0 ROBOT CAR kit	1	86'13 €
Càmera Crosstour	1	32'32 €
Cables Elegoo	120	7'59 €
Motor pas a pas	5	11'99 €
Mòdul Bluetooth HC-05	1	8 €
Impressió 3D	5 peces	10 €
Total		156'03 €

Annex 4: Codi Arduino

Aquest és el codi complet que permet el funcionament del vehicle *Voyager*:

```
//-----//
//  ARDUINO  //
//-----//
//  ÍNDEX  //
//  //  //
//    --> BIBLIOTECA  //
//    --> CONFIGURACIÓ DELS PINS  //
//    --> VARIABLES  //
//    --> FUNCIÓ SETUP  //
//    --> FUNCIÓ LOOP  //
//    --> FUNCIONS DECLARADES PEL PROGRAMADOR  //
//  //  //
//-----//

//-----//
//    --> BIBLIOTECA  //
//-----//

// @ SERVO

#include <Servo.h>

//-----//
//    --> CONFIGURACIÓ DELS PINS  //
//-----//

// Definim els pins per les velocitats pels parells de motors

#define vel_A 5
#define vel_B 6

// Definim els pins pels motors

#define motor1 7
#define motor2 8
#define motor3 9
#define motor4 11

// Definim els pins necessaris per l'ús del mòdul seguidor de línies

#define LT_R !digitalRead(10)
#define LT_M !digitalRead(4)
#define LT_L !digitalRead(2)

// Definim els pins necessaris per l'ús del mòdul ultrasons

int Echo = A4;
int Trig = A5;

// Definim els pins necessaris per l'ús del motor pas a pas que fa girar la càmera

#define eCoil_1 14
#define eCoil_2 15
#define eCoil_3 16
#define eCoil_4 17
```

```

//-----//
//      --> VARIABLES                                     //
//-----//

// Creem l'objecte myservo per controlar el servomotor de l'ultrasons

    Servo myservo;

// Creem les variables necessàries per les distàncies llegides per l'ultrasons

    int rightDistance = 0, leftDistance = 0, middleDistance = 0;

// Creem Variable getstr per guardar el caràcter llegit pel port Serial / Bluetooth

    char getstr;

// Moviment del motor pas a pas
//
//      PAS  BOBINA A  BOBINA B  BOBINA C  BOBINA D
//      1    HIGH    LOW      LOW      LOW
//      2    HIGH    HIGH     LOW      LOW
//      3    LOW     HIGH     LOW      LOW
//      4    LOW     HIGH     HIGH     LOW
//      5    LOW     LOW      HIGH     LOW
//      6    LOW     LOW      HIGH     HIGH
//      7    LOW     LOW      LOW      HIGH
//      8    HIGH    LOW      LOW      HIGH

// A partir d'aquesta taula, creem la seqüència del motor pas a pas
int sequenceSteps [8][4] =
{
    {1, 0, 0, 0},
    {1, 1, 0, 0},
    {0, 1, 0, 0},
    {0, 1, 1, 0},
    {0, 0, 1, 0},
    {0, 0, 1, 1},
    {0, 0, 0, 1},
    {1, 0, 0, 1}
};

// Variables necessàries per utilitzar el motor pas a pas

boolean Direction = true;

int Steps = 0; // Indica el pas actual de la seqüència

int steps_left = 511;

// Flag que ens permetrà executar els modes Ultrasons i Seguidor de línies dins del Switch-Case

    int flag = 0;

// Variable pels bucles for

    int m;

```

```
//-----//
//      --> FUNCIO SETUP                                     //
//-----//

void setup() {

    Serial.begin(9600);

    // Configurem els pins dels motors com una sortida

    pinMode(motor1, OUTPUT);
    pinMode(motor2, OUTPUT);
    pinMode(motor3, OUTPUT);
    pinMode(motor4, OUTPUT);

    // Configurem els pins de les velocitats dels motors com una sortida

    pinMode(vel_A, OUTPUT);
    pinMode(vel_B, OUTPUT);

    // Executem la funció parar per assegurar-nos que quan entrem al LOOP les rodes estaran parades

    parar();

    // Configurem els pins dels sensors infrarojos del mòdul seguidor de línies com una entrada d'informació -> INPUT

    pinMode(LT_R, INPUT);
    pinMode(LT_M, INPUT);
    pinMode(LT_L, INPUT);

    // El mètode attach(pin) connecta una variable Servo al pin assignat.

    myservo.attach(3); // adjunta la variable servo al pin 3

    // Els pins Echo i Trig, els definirem un com entrada i l'altre com sortida,
    // ja que Trig serà l'encarregat d'emetre l'ultrasó i Echo l'encarregat de rebre'l

    // Configurem el pin Echo del sensor d'ultrasons com una entrada d'informació -> INPUT
    pinMode(Echo, INPUT);

    // Configurem el pin Trig del sensor d'ultrasons com una sortida d'informació -> OUTPUT
    pinMode(Trig, OUTPUT);

    // Configurem els pins de les bobines del motor pas a pas que fa girar la càmera com una sortida d'informació -> OUTPUT
    pinMode(eCoil_1, OUTPUT);
    pinMode(eCoil_2, OUTPUT);
    pinMode(eCoil_3, OUTPUT);
    pinMode(eCoil_4, OUTPUT);

}

//-----//
//      --> FUNCIO LOOP                                     //
//-----//

void loop() {

    // Si tenim alguna cosa pel port Serial / Bluetooth
```

```

if (Serial.available() > 0) {

    // Llegim el caràcter del port Serial / Bluetooth i l'emmagatzemem a la variable getstr
    getstr = Serial.read();
    Serial.write(getstr);

    // Gestionem les accions que haurà de realitzar el cotxe mitjançant un switch case
    switch (getstr) {
        case 'a': avansa(0);
            break;

        case 'r': retrocedeix();
            break;

        case 'e': girEsquerra(0);
            break;

        case 'd': girDreta(0);
            break;

        case 'p': parar();
            break;

        case 'l': do {
            flag = 1; modeSeguidorLinies();
            if (Serial.available() > 0) {
                flag = 0;
                // Llegim el caràcter del port Serial / Bluetooth i l'emmagatzemem a la variable getstr
                getstr = Serial.read();
            }
        } while (flag == 1);
        break;

        case 'u': do {
            flag = 1; modeUltrasons();
            if (Serial.available() > 0) {
                flag = 0;
                // Llegim el caràcter del port Serial / Bluetooth i l'emmagatzemem a la variable getstr
                getstr = Serial.read();
            }
        } while (flag == 1);
        break;

        case 'y': cameraGirEsquerra(511);
            break;

        case 'x': cameraGirDreta(511);
            break;

        default: break;
    }
}

//-----//
//      --> FUNCIONS DECLARADES PEL PROGRAMADOR                                //
//-----//

```

```
// --> avansa
// --> retrocedeix
// --> girEsquerra
// --> girDreta
// --> parar
//
// --> stepper
// --> setDirection
// --> càmeraGiroEsquerra
// --> càmeraGirDreta
//
// --> modeSeguidorLinies
//
// --> testDistància
// --> servoTestEsquerra
// --> servoTestDreta
// --> servoCentre
// --> modeUltrasons
//
// Direccions de les rodes i els seus moviments
// -----
// vel_A   vel_B   motor1   motor2   motor3   motor4
//
// HIGH    HIGH    HIGH    LOW    LOW    HIGH    ||-----> Cotxe Avansa
// HIGH    HIGH    LOW    HIGH    HIGH    LOW    ||-----> Cotxe Retrocedeix
// HIGH    HIGH    LOW    HIGH    LOW    HIGH    ||-----> Cotxe Gir Esquerra
// HIGH    HIGH    HIGH    LOW    HIGH    LOW    ||-----> Cotxe Gir Dreta
//
// HIGH    HIGH    LOW    LOW    LOW    LOW    ||-----> Cotxe Parat
// HIGH    HIGH    HIGH    HIGH    HIGH    HIGH    ||-----> Cotxe Parat
// LOW    LOW    N/A    N/A    N/A    N/A    ||-----> Cotxe Parat
//
//-----
//-----

/* Funció avansa() */

void avansa(int mode) {

    if (mode == 1)
    {
        analogWrite(vel_A, 100);
        analogWrite(vel_B, 100);
    }
    else {
        analogWrite(vel_A, 255);
        analogWrite(vel_B, 255);
    }
    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, LOW);
    digitalWrite(motor3, LOW);
    digitalWrite(motor4, HIGH);

}
//-----

/* Funció retrocedeix() */
```

```
void retrocedeix() {  
  
    analogWrite(vel_A, 255);  
    analogWrite(vel_B, 255);  
  
    digitalWrite(motor1, LOW);  
    digitalWrite(motor2, HIGH);  
    digitalWrite(motor3, HIGH);  
    digitalWrite(motor4, LOW);  
  
}  
//-----//  
  
/* Funció girEsquerra() */  
  
void girEsquerra(int gi) {  
  
    switch (gi) {  
  
        case 0: // Gir esquerra quotidiana  
  
            analogWrite(vel_A, 255);  
            analogWrite(vel_B, 255);  
  
            digitalWrite(motor1, LOW);  
            digitalWrite(motor2, HIGH);  
            digitalWrite(motor3, LOW);  
            digitalWrite(motor4, HIGH);  
  
            delay(400); parar();  
            break;  
  
        case 1: // Gir esquerra línia  
  
            analogWrite(vel_A, 150);  
            analogWrite(vel_B, 150);  
  
            digitalWrite(motor1, LOW);  
            digitalWrite(motor2, HIGH);  
            digitalWrite(motor3, LOW);  
            digitalWrite(motor4, HIGH);  
  
            delay(100); parar();  
            break;  
    }  
}  
  
//-----//  
  
/* Funció girDreta() */  
  
void girDreta(int gd) {  
    switch (gd) {  
  
        case 0: // Gir a la dreta quotidiana  
  
            analogWrite(vel_A, 255);  
            analogWrite(vel_B, 255);
```

```

    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, LOW);
    digitalWrite(motor3, HIGH);
    digitalWrite(motor4, LOW);

    delay(400); parar();
    break;

case 1: // Gir drete línia

    analogWrite(vel_A, 150);
    analogWrite(vel_B, 150);

    digitalWrite(motor1, HIGH);
    digitalWrite(motor2, LOW);
    digitalWrite(motor3, HIGH);
    digitalWrite(motor4, LOW);

    delay(100); parar();
    break;
}
}

//-----//

/* Funció parar() */

void parar() {
|
    digitalWrite(vel_A, LOW);
    digitalWrite(vel_B, LOW);
}

//-----//
//-----//

/* Funció stepper() */

void stepper() { // Avansa un pas cada cop que es invoca

    digitalWrite( eCoil_1, sequenceSteps[Steps][ 0] );
    digitalWrite( eCoil_2, sequenceSteps[Steps][ 1] );
    digitalWrite( eCoil_3, sequenceSteps[Steps][ 2] );
    digitalWrite( eCoil_4, sequenceSteps[Steps][ 3] );

    setDirection();
}

//-----//

/* Funció setDirection() */

void setDirection() { // Controla el sentit del gir i el valor de Steps

    if (Direction) {
        Steps++; // Incrementa un pas de steps
    }

    else {

```

```

    Steps--; // Disminueix un pas de steps
}

Steps = ( Steps + 8 ) % 8 ; // Permet passar de la primera fase de la seqüència a l'última i viceversa
}

//-----//

/* Funció càmeraGirEsquerra() */

void cameraGirEsquerra(int steps_left)
{
    Direction = false;

    for (m = steps_left ; m > 0; m--)
    {
        stepper() ;    // Avansa un pas
        delay(1);
    }
}

//-----//

/* Funció càmeraGirDreta() */

void cameraGirDreta(int steps_left)
{
    Direction = true;

    for (m = steps_left ; m > 0; m--)
    {
        stepper() ;    // Avansa un pas
        delay(1);
    }
}

//-----//
//-----//

/* Funció modeSeguidor Linies() */

void modeSeguidorLinies() {

    // Si el mòdul central detecta la línia negra -> avansa()
    if (LT_M) {
        avansa(1);
    }

    // Si no, si el mòdul dret detecta la línia negra -> girDreta() mentre que el mòdul segueixi detectant la línia
    else if (LT_R) {
        girDreta(1);
        while (LT_R);
    }

    // Si no, si el mòdul esquerra detecta la línia negra -> girEsquerra() mentre que el mòdul segueixi detectant la línia
    else if (LT_L) {
        girEsquerra(1);
    }
}

```



```

    while (LT_L);
  }
  else
  {
    avansa(1);
  }
}

//-----//

/**
Funció testDistància():
Serveix per realitzar una emissió i una recepció en el sensor d'ultrasons
i així poder calcular la distància a la que es troba un objecte del nostre cotxe

@ return (Fdistance)
*/

int testDistancia() {

  digitalWrite(Trig, LOW);           // Ens assegurem de que Trig no estigui emetén
  delayMicroseconds(2);              // Espqerem 2 mil·lisegons
  digitalWrite(Trig, HIGH);          // Emetem un ultrasò durant 20 mil·lisegons
  delayMicroseconds(20);
  digitalWrite(Trig, LOW);           // Parem d'emetre

  float Fdistance = pulseIn(Echo, HIGH); // Guardem a la variable Fdistance, el valor rebut del pin Echo
  Fdistance = Fdistance / 58;           // Apliquem la fórmula per calcular la distància respecte el temps que triga en enviar-se i rebre'l
  return (int)Fdistance;               // Retornem la variable Fdistance
}

//-----//

/* Funció servoTestDreta() */

void servoTestEsquerra() {

  for (m = 90; m < 180; m++)
  {
    myservo.write(m);
    delay(1);
  }
}

//-----//

/* Funció servoTestEsquerra() */

void servoTestDreta() {

  for (m = 90; m > 0; m--)
  {
    myservo.write(m);
    delay(1);
  }
}

```

```
//-----//

/* Funció servoCentre() */

void servoCentre() {

    myservo.write(90);
    delay(10);
}

//-----//

/* Funció modeUltrasons() */

void modeUltrasons() {
    avansa(1);
    myservo.write(90);
    delay(500);
    middleDistance = testDistancia();

    if (middleDistance <= 30) {

        parar();

//--- Gira dreta Servo Ultrasons per obtenir el valor d'un possible obstacle a la dreta

        delay(500);
        servoTestDreta();
        delay(1000);
        rightDistance = testDistancia();

//--- Gira Servo Ultrasons al valor central

        delay(500);
        myservo.write(90);
        delay(500);

//--- Gira esquerra Servo Ultrasons per obtenir el valor d'un possible obstacle a la dreta

        delay(1000);
        servoTestEsquerra();
        delay(1000);
        leftDistance = testDistancia();

//--- Gira Servo Ultrasons al valor central

        delay(500);
        myservo.write(90);
        delay(2000);

//--- Comparem les distàncies obtingudes entre elles per poder decidir quina funció de moviment executar depenent de la major distància obtinguda

        if ((rightDistance <= 30) && (leftDistance <= 30)) {
            retrocedeix();
            delay(360);
            girDreta(0);
            delay(360);
            girDreta(0);
        }
    }
}
```

```
    delay(360);  
  }  
  else if (rightDistance < leftDistance) {  
    girEsquerra(0);  
    delay(360);  
  }  
  else if (rightDistance > leftDistance) {  
    girDreta(0);  
    delay(360);  
  }  
  else {  
    avansa(1);  
  }  
}  
else {  
  avansa(1);  
}  
}
```

```
//-----//|
```

Imatges 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54 i 55: Captures de pantalla del programa Arduino