UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

IMPLEMENTACIÓ D'UN SISTEMA MUSICAL INTERACTIU AL *SOCIAL PET ROBOT* CASPER

Memòria

Autor: Albert Tibau Font
Director: Sebastián Tornil
Ponent: Cecilio Angulo

Convocatòria: Gener 2018

Resum

En aquesta memòria es presenta el treball de disseny i implementació d'una interfície lúdica musical que s'ha desenvolupat per al robot CASPER (**C**ognitive **A**ssistive **S**ocial **PE**t **R**obot), un autòmat d'assistència cognitiva creat per facilitar l'aprenentatge i millorar la capacitat de socialització de nens amb Desordre de l'Espectre Autista.

Es mostra tant la part hardware com software que compon el robot, que ha estat desenvolupat en llenguatge Arduino, molt similar al C++. S'han programat dos diferents modalitats de joc: un amb l'objectiu de guiar el nen per reproduir una melodia predeterminada i l'altre que li permetrà d'una manera lliure crear-ne una de pròpia. Aquests han estat desenvolupats amb l'objectiu de treure el màxim profit del hardware amb el qual ja comptava CASPER: s'ha fet ús de quatre servomotors Dynamixel i de dos microcontroladors Arduino (un màster i un esclau, comunicats entre ells mitjançant protocol I2C). A més, s'han implementat al robot sensors tàctils, leds i un altaveu, que aporten al projecte un aspecte més lúdic.

La música servirà d'intermediari entre el robot i el nen, i alhora podria suposar un punt de contacte inicial entre el terapeuta i el pacient, el qual generalment sol ignorar o fins i tot rebutjar la comunicació humana.

Resumen

En esta memoria se presenta el trabajo de diseño e implementación de una interfaz lúdica musical que se ha desarrollado para el robot CASPER (**C**ognitive **A**ssistive **S**ocial **PE**t **R**obot), un autómata de asistencia cognitiva creado para facilitar el aprendizaje y mejorar la capacidad de socialización de niños con Desorden del Espectro Autista.

Se muestra tanto la parte hardware como software que compone el robot, que ha sido desarrollado en lenguaje Arduino, muy similar al C++. Se han programado dos distintas modalidades de juego: uno con el objetivo de guiar al niño para reproducir una melodía predeterminada y el otro que le permitirá de una manera libre crear una melodía propia. Estos juegos han estado desarrollados con el objetivo de sacar el máximo provecho del hardware con el que ya contaba CASPER: se han usado cuatro servomotores Dynamixel y dos microcontroladores Arduino (un máster y un esclavo, comunicados entre ellos mediante protocolo I2C). Además, se han implementado al robot sensores táctiles, leds y un altavoz, que aportan al proyecto un aspecto más lúdico.

La música servirá de intermediario entre el robot y el niño, y a la vez podría suponer un punto de contacto inicial entre el terapeuta y el paciente, el cual generalmente suele ignorar o hasta rechazar la comunicación humana.

Abstract

This paper presents the work of design and implementation of a musical playful interface that has been developed for CASPER robot (**C**ognitive **A**ssistive Social **PE**t **R**obot), an automaton of cognitive assistance created to facilitate learning and improve the socialization capacity of children with Autism Spectrum Disorder (ASD).

It shows both hardware and software parts that makes up the robot, which has been developed in Arduino language, very similar to C++. Two different game modes have been programmed: one with the objective of guiding the child to play a predetermined melody and the other will allow him to create one of his own in a free way. These games have been developed with the aim of making the most of the hardware that CASPER already had: four Dynamixel servomotors and two Arduino microcontrollers (a master and a slave, communicated with each other using I2C protocol) has been used. In addition, tactile sensors, LEDs and a loudspeaker have been implemented to the robot, which provide a more playful point to the project.

The music will serve as an intermediary between the robot and the child, and at the same time could be an initial point of contact therapist-patient, which usually ignores or even rejects human communication.

Agraïments

Vull agrair l'ajuda desinteressada de tota la gent que ha treballat prèviament en el projecte CASPER que m'han ofert informació i consell quan m'hi he posat en contacte, especialment Daniel Capelo i Thomas Herpoel. També vull agrair al tutor Sebastián Tornil i al professor Cecilio Angulo, director del projecte, la seva predisposició, tant en el procés de preparació d'aquest document com en la part més tècnica del treball.

Gràcies també a la família i amics pel suport i la confiança incondicional.

Glossari

ASD: Autism Spectrum Disorder

CAN: Controller Area Network

CASPER: Cognitive Assistive Social Pet Robot

CLK: clock

CS: chip select

EEPROM: Electronically Erasable Programmable Read Only Memory

GPIO: General Purpose Input/Output

HMI: Human Machine Interface

I2C: Inter Integrated Circuit

ISR: Interruption Service Routine

MISO: Master Input Slave Output

MOSI: Master Output Slave Input

ROS: Robot Operating System

SCL: serial clock line

SDA: serial data

SPI: Serial Peripherical Interface

UART: Universal Asynchronous Receiver-Transmitter

Índex

1. Prefaci

L'autisme és un trastorn psicològic que afecta més d'1 de cada 100 adults a Europa, una taxa que va en augment. Aquest trastorn es manifesta, en la majoria de casos, durant els 3 primers anys després del naixement. Tot això fa que sigui molt important poder tractar aquests nens a una edat primerenca, i poder millorar així les seves capacitats socials de manera més senzilla. Altres trastorns similars com la síndrome d'Asperger o el trastorn generalitzat del desenvolupament no especificat tenen trets força semblants, dels quals se'n destaca la dificultat per reconèixer les emocions alienes, la baixa capacitat de socialització i la reciprocitat emocional.

Un robot pot ser l'aliat clau d'un terapeuta per poder entrar en contacte amb un nen que hagi estat diagnosticat amb un trastorn d'aquest tipus, o fins i tot pot ajudar un metge en la fase de diagnosi d'aquestes malalties [1]. El nen veu el robot d'una forma molt diferent a com veu un metge o un familiar, amb maneres de fer i interactuar amb l'exterior completament diferents de la seva. L'infant amb un trastorn autista pot veure el robot com un igual, amb unes capacitats d'actuació previsibles i repetitives, més similars a la seva pròpia.

1.2. Origen del treball

El projecte CASPER neix per satisfer les necessitats d'un nen amb característiques autístiques com les que s'han anomenat anteriorment o amb nens amb dèficit d'atenció.

A títol personal, la proposta de col·laborar en aquest projecte em va arribar quan em vaig posar en contacte amb el doctor Jose Luis de la Rosa, professor i investigador de la UdG. Quan li vaig comentar que era estudiant d'electrònica industrial de la UPC i estava interessat en la robòtica, em va recomanar el doctor Cecilio Angulo. Ell em va presentar diferents projectes sobre robòtica que estaven actius en el departament de l'ESAII, d'entre els quals em va cridar especialment l'atenció el robot CASPER.

1.3. Motivació

Actualment l'automatització i la robòtica estan a l'ordre del dia, i formen part d'un camp que no para de créixer: l'electrònica. Estem en contacte permanent amb dispositius electrònics, ja sigui per comunicar-nos a la feina, per aprendre, divertir-nos... Però també en els moments més difícils l'electrònica representa un punt de suport bàsic: la medicina utilitza cada cop més equips electrònics i robots per realitzar operacions quirúrgiques, monitoritzar pacients per tal de portar-ne un seguiment, o per implantar pròtesis biòniques.

Personalment, penso que no hi ha res més satisfactori que aplicar els coneixements d'aquesta branca de l'enginyeria per a millorar la qualitat de la vida humana. Una d'aquestes branques potser no tan coneguda és la de la robòtica aplicada a la psicologia infantil o al tractament de trastorns de tipus autista. És per això que el robot CASPER em va semblar un projecte realment interessant. La finalitat més humana i clínica del robot em va entusiasmar des d'un principi, sobretot pel fet que s'utilitzés en hospitals i que en un futur servís per prendre constants vitals en infants o simplement per fer-los interactuar i entretenir, amb el clar objectiu de millorar les seves habilitats socials i que això suposés un augment de la seva qualitat de vida.

.

2. Introducció

2.1. Objectius del treball

A grans trets, l'objectiu del treball és desenvolupar una plataforma musical interactiva per al prototip de robot CASPER. Es pretén, doncs, a més de desenvolupar el propi treball, implementar-lo dins d'un projecte més gran com és el de CASPER. Per tal de dur a terme aquesta integració, s'ha d'entendre el propòsit del robot i la seva base de funcionament, fins i tot abans de fer el projecte en si mateix.

Primerament, doncs, s'ha d'estudiar el treball que ja s'ha fet en aquest robot, tant des d'un punt de vista estructural (hardware i disseny físic) com des d'una visió més interna (software). Per tant, es té com a punt de partida l'objectiu de conèixer a fons el projecte, veure què s'hi ha desenvolupat, què pot faltar per aportar-hi que pugui ésser d'utilitat i de quina manera es pot fer. També s'ha d'entendre l'objectiu final del robot, és a dir, la funció que es vol que aquest realitzi, i com la idea d'un joc musical hi pot encaixar.

Com a objectiu principal, em proposo desenvolupar una interfície amb la qual els nens puguin interactuar mitjançant la música. D'aquesta manera hi ha una interacció amb l'entorn a través d'una via lúdica, que a llarg termini hauria de poder permetre millorar les capacitats socials i cognitives als nens amb trastorns autístics.

Des d'un punt de vista més tècnic, també pretenc entendre i millorar el coneixement en programació Arduino-RaspberryPi, bàsic en el disseny de molts robots, ja siguin androides o de servei. També analitzaré els diferents protocols de comunicació entre microcontroladors.

Un objectiu molt important és el de permetre que aquest treball pugui ser la base per a la continuació del projecte CASPER, aclarint-ne al màxim el funcionament amb aquesta memòria o facilitant la implementació d'altres funcionalitats dins del mateix codi. És a dir, cal estructurar el programa de manera que sigui fàcil encabir-hi altres petits programes, i al mateix temps, que aquest sigui fàcilment implementable dins del codi definitiu del CASPER.

3. Estat de l'art

El fet que un robot pugui ajudar a un infant amb dificultats de caire social pot semblar contradictori al principi, però al llarg dels últims anys nombrosos estudis han demostrat la influència positiva que un autòmat pot tenir sobre un nen[2], i el més important, el tipus d'influència que hi pot tenir, molt diferent de la dels metges o dels propis pares. Enginyers, neurocientífics, pediatres, psicòlegs i fins i tot educadors, treballen conjuntament en el desenvolupament d'idees que puguin ajudar a persones amb aquest tipus de trastorns.

Els robots assistencials, tant els humanoides com altres amb dissenys que puguin ésser més atractius per a nens (com en el cas del projecte CASPER, animals), compten generalment amb tipus molt variats de sensors i dispositius que permeten la comunicació robot-pacient, ja sigui amb càmeres, sensors tàctils, motors que permeten el moviment del robot, llums, micròfons i altaveus, etc.

En el present capítol es posarà de manifest aquest fet tot fent referència a diferents articles científics que han estudiat aquest camp, en el qual encara queda molt per descobrir. A més, es veuran diversos exemples de robots d'assistència, ja siguin prototips o en comercialització, que compten amb alguns dels dispositius anomenats anteriorment.

3.1. Robots socials assistencials

S'entén per robòtica assistencial l'àrea de la robòtica que s'especialitza en el disseny i desenvolupament d'equips que interactuen directament amb l'individu per facilitar-li alguna tasca, tot i que generalment s'aplica en el camp de la salut.

En aquest camp, hi ha robots assistencials que s'utilitzen de manera regular per ajudar la gent gran, persones amb algun tipus de demència o amb problemes de mobilitat. Fins i tot s'empren en tractaments de rehabilitació fisioterapèutica. Aquests robots fa temps que s'utilitzen i es comercialitzen en un mercat que ja es podria considerar estabilitzat.

Els robots assistencials enfocats a tractar trastorns com l'autisme no tenen encara un mercat gaire definit o estès. Tot i que alguns fa temps que es comercialitzen com a robots interactius a manera de joguina i actualment també s'utilitzen en aquesta vessant clínica, no hi ha gaires robots especialitzats en aquest camp que es comercialitzin d'una manera regulada. El concepte de robot d'assistència associat a nens amb discapacitats o dificultats de comunicació i atenció ha anat agafant forma al llarg dels últims anys, i això fa que sigui un camp on encara hi ha molt per fer. Universitats d'arreu proposen

diferents projectes amb funcionalitats molt diverses per tal d'estudiar l'efecte que el robot pot causar a l'infant. Algunes empreses han presentat:

3.1.1. KASPAR

Kaspar és un robot social humanoide desenvolupat per la Universitat de Hertfordshire, que destaca per la capacitat de mostrar expressions facials que poden ser reconegudes pels nens i d'aquesta manera els permeten identificar una emoció o sentiment amb un gest o una expressió. El fet que ofereixi una forma de comunicació més predictible i repetitiva, ajuda a simplificar la interacció i a fer-la més còmoda per a l'infant, que en general veu la interacció social com quelcom aclaparador, imprevisible i fins i tot aterridor.

FIgura 3. 1 El Robot KASPAR també incorpora un gadget musical (Font: [10])

3.1.2. NAO

El 2008 la companyia francesa Aldebaran Robotics va llançar al mercat el robot NAO, un projecte que va començar el 2004. Aquest robot serveix a universitats i laboratoris amb finalitats d'investigació i educació, i té el gran avantatge d'ésser programable pel propi usuari (NAO acadèmic). Mitjançant la plataforma NAO i la seva eina gràfica de programació Aldebarán Choréographe, es permet manipular amb molta llibertat el robot per tal que es pugui acomodar a les necessitats de cada desenvolupador. A més, és compatible amb altres softwares de programació Microsoft i Linux.

Així doncs, el fet que sigui un robot programable ha afavorit la seva utilització en múltiples branques de la robòtica, des d'una vessant lúdica -per exemple, la RoboCup, un torneig de futbol robòtic- fins a la mèdica -com ara la del tractament de trastorns de l'espectre autista- [3].

El robot compta amb micròfons, altaveus, comunicació Wi-Fi, càmeres HD per reconeixement facial i de formes), i sensors d'ultrasò, acceleròmetres, giròmetres, sensors tàctils i de pressió. Ha estat programat amb diferents llenguatges a la vegada: C++, Python, Java, MATLAB.

Un exemple de com la plataforma NAO ha estat utilitzada en teràpies per a nens autistes és RENÉ. Els seus desenvolupadors, a la Universitat de Zagreb, l'han programat perquè el seu ús estigui enfocat principalment a diagnosticar aquest tipus de trastorns: mitjançant l'ús del robot en subjectes que ja han estat diagnosticats s'utilitza en nens per confirmar el pronòstic i d'aquesta manera poder ajudar els psiquiatres en la fase de diagnosi.

FIgura 3. 2 El robot RENÉ utilitza l'estructura del NAO per aplicar-la al tractament de nens autistes. (Font: [3])

3.1.3. PLEO rb

PLEO rb és un altre exemple d'una mascota-robot reprogramable amb l'aparença d'un dinosaure, amb una capacitat molt desenvolupada per imitar moviments naturals i expressions. El fet que fos programable va fer que es plantegés d'ésser utilitzat com a plataforma per desenvolupar el projecte CASPER, tot i que finalment es va optar per crear un disseny propi.

Després d'haver estudiat el mercat actual de robots socials, es va decidir crear CASPER, plataforma que serà utilitzada per desenvolupar un sistema musical interactiu, que és l'objecte d'aquest treball de fi de grau. Així doncs, s'entra ara a estudiar les característiques de CASPER.

3.2. CASPER

CASPER neix a partir de l'experiència prèvia obtinguda en els projectes PATRICIA i PLEO, amb la finalitat de millorar la qualitat de vida dels nens que visiten els hospitals de manera regular o que tenen necessitats especials. [4] Un dels objectius és desenvolupar una plataforma que permeti implementar eines tecnològiques comercials com tauletes o braçalets per millorar el seguiment del nen per part del metge, mitjançant la presa de constants per part del robot. Aquestes dades seran enviades, al seu torn, al metge.

Per altra banda, CASPER ha de tenir el paper d'ajudant i mediador social durant les sessions d'interacció social dels nens. Una manera d'afavorir aquesta interacció nen-robot és mitjançant activitats lúdiques que permetin el contacte directe entre ells. Així és com neix la idea del projecte d'implementació d'un sistema musical pel CASPER.

Els nens autistes, especialment en les primeres etapes, tendeixen a rebutjar o ignorar qualsevol tipus de contacte amb una persona, fins i tot amb el terapeuta: no obstant, un instrument musical pot servir d'intermediari efectiu entre el pacient i el metge, i ofereix a l'especialista un punt de contacte inicial.

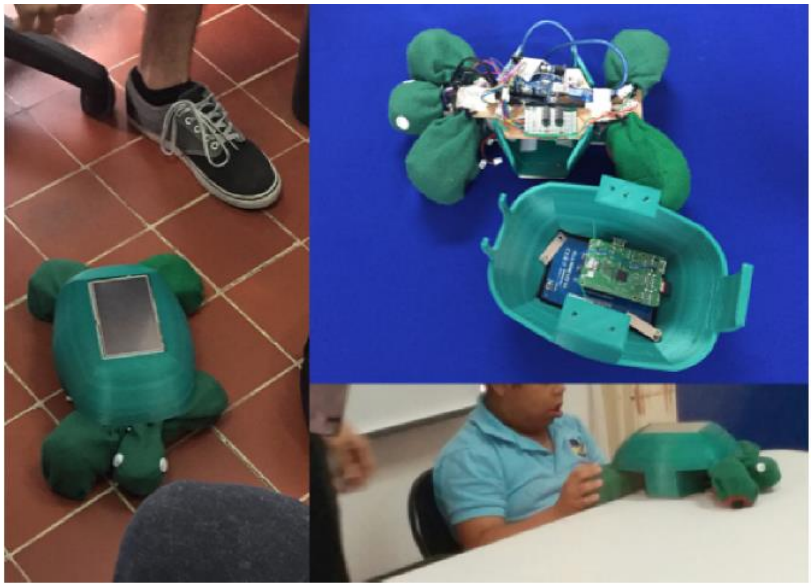


FIgura 3. 3Prototip de CASPER realitzat a Panamà interactua amb un nen autista (Font: [4])

Articles científics [5] descriuen com la musicoteràpia pot ser molt efectiva per reforçar i canviar el comportament social de l'autista. En l'àrea de la comunicació, aquesta teràpia facilita el procés de parla i vocalització, a més de regular el comportament sensitiu i motor, freqüentment alterat en el nen autista. A més, ofereix al pacient la llibertat per utilitzar un instrument musical i crear diferents sons d'una manera simplificada es cultiva aquesta vessant.

4. Arquitectura i components

4.1. Disseny físic

El disseny de exterior de la tortuga està basat en el treball de Beste Ozcan de l'Institute of Cognitive Sciences and Technologies de Roma. [6] Jonas Londschien, en unes pràctiques a la Universitat Politècnica de Catalunya, va adaptar aquest disseny i va desenvolupar la part mecànica que permet el moviment del robot. En la figura 4.1 es pot veure l'estructura de la tortuga impresa.

Figura 4. 1 Estructura de CASPER amb les cames en prototip (Font: [6])

Aquest disseny en particular, del qual per a aquesta versió de CASPER només es modifica la forma de les potes, està desenvolupat íntegrament mitjançant impressions 3D, utilitzant un material rígid per a la carcassa i un filament flexible per a les potes (la qual cosa permet a l'usuari jugar amb les potes sense trencar-les).

En la figura 4.2 s'observa la darrera modificació. Les potes prenen una forma més similar a la d'una tortuga, donant-li un aspecte més realista i atractiu per a l'usuari. De fet, es partirà d'aquesta estructura per desenvolupar aquest projecte, i únicament se n'aprofitarà l'estructura física per treballar com a punt de partida.

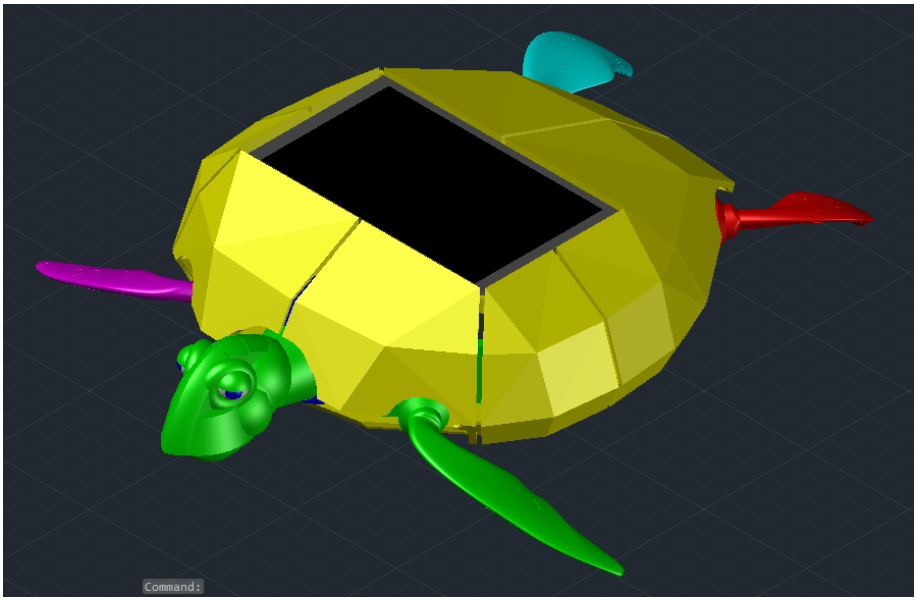


Figura 4. 2 Disseny de la tortuga en 3D (Font: [7])

4.2. Arquitectura interna de CASPER

El disseny intern de la plataforma es basa en un processador tipus Raspberry-Pi que fa d'intermediari entre les accions del robot i la transmissió d'aquestes accions al *cloud*. El procés es du a terme mitjançant una comunicació Wi-Fi, que emmagatzema la informació a una base de dades i alhora la transmet al terapeuta.

En la figura 4.3 es pot veure el funcionament de CASPER en la seva totalitat.

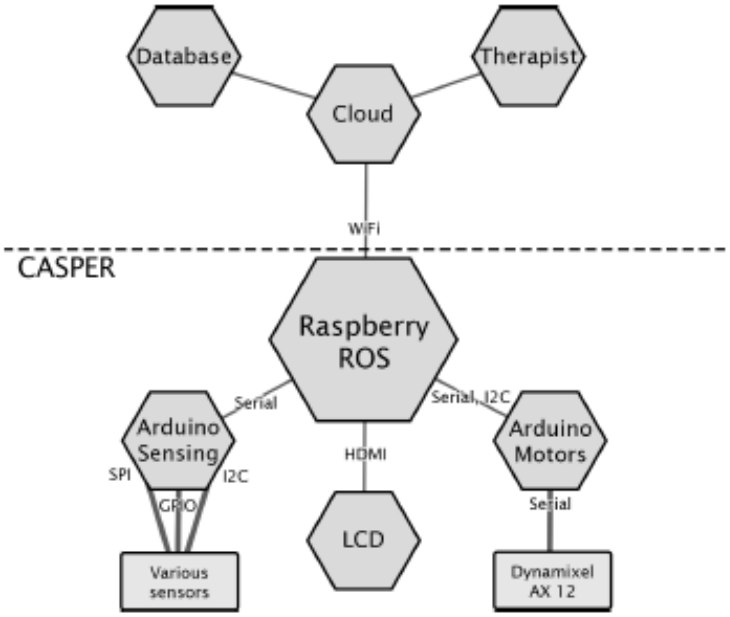


Figura 4. 3 Perspectiva general de la plataforma (Font: [8])

El processador tipus RaspberryPi 3 modificat per treballar en ROS (Robot Operating System) ens permet la seva programació amb el sistema operatiu Linux. Durant la seva col·laboració a CASPER per al seu Treball Final de Màster, Daniel Capelo va dissenyar l'interfície HMI (Human Machine Interface), mitjançant la qual es pot accedir al menú de CASPER. A través d'aquest menú s'accedeix també a la secció de *gaming* (del qual el sistema musical en formaria part). Una pantalla LCD es connecta mitjançant HDMI a la Raspberry Pi.

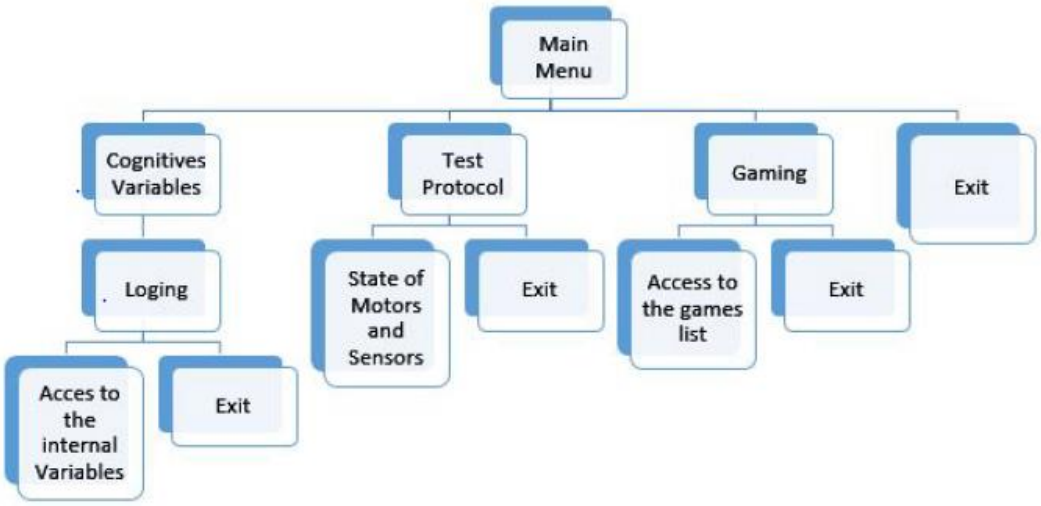


Figura 4. 4 Esquema dels menús de l'HMI (Font: [7])

4.3. Diagrama de blocs

A continuació s'explica el disseny utilitzat per al sistema musical de CASPER. Es fa ús dels dos Arduino UNO amb què compta teòricament CASPER, un per tal d'adquirir les mesures dels sensors tàctils que s'han dissenyat i l'altre per actuar sobre els motors segons les dades que s'obtinguin.

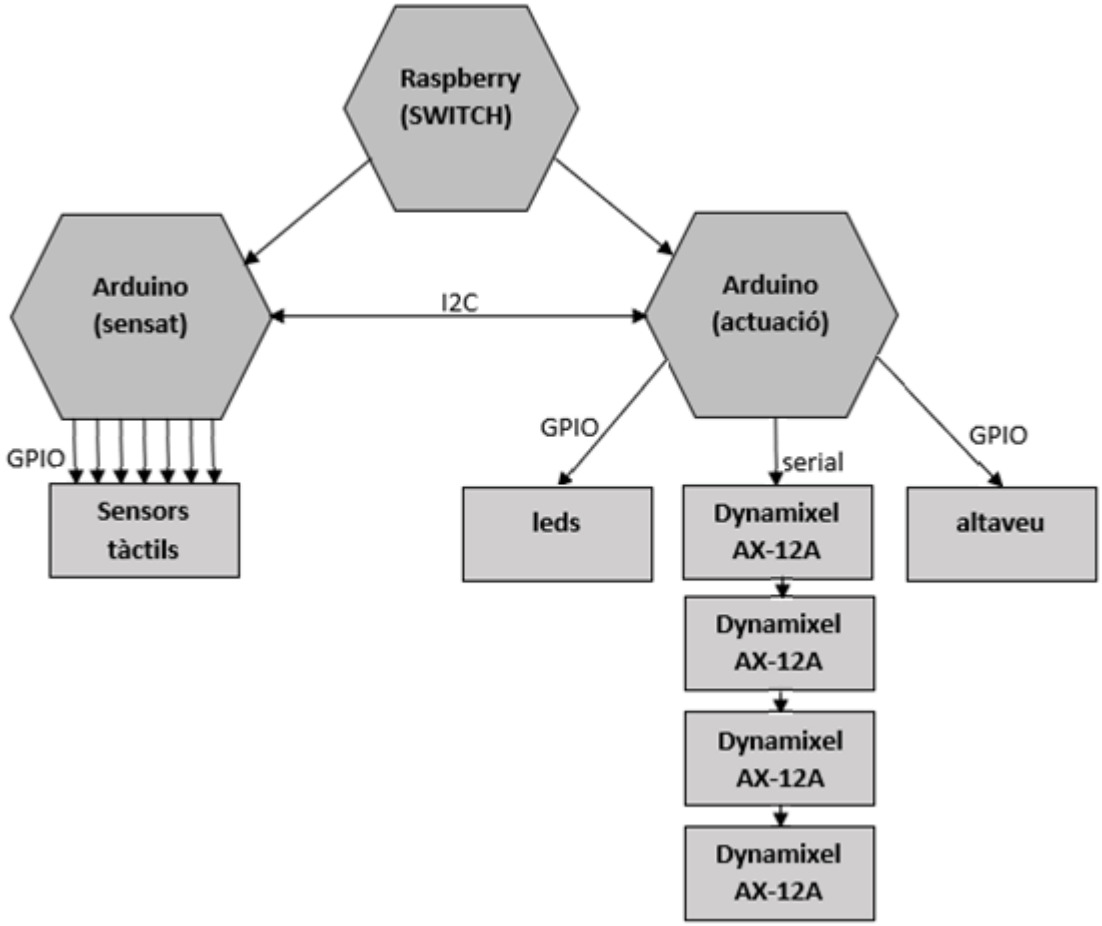


Figura 4. 5 Diagrama de blocs del sistema musical (Font: pròpia)

A la figura 4.5 es mostra l'esquema de la interfície musical, amb els processadors sensors i actuadors que la componen. L'Arduino encarregat d'adquirir i processar les dades dels sensors tàctils les transmet a l'Arduino dels actuadors mitjançant el protocol I2C. Aquest darrer té connectats en sèrie 4 servomotors Dynamixel AX-12A, a més de leds i un altaveu. Per al disseny del sistema musical, la RaspberryPi no hi té un paper com a tal, sinó que la funció de selecció de jocs es fa a través d'un *switch*.

4.4. Selecció i descripció de components

4.4.1. Placa arduino UNO

El projecte CASPER utilitza com a microprocessadors dues plaques de desenvolupament Arduino UNO amb un microprocessador ATMega328. Aquestes dues plaques estaven disponibles al laboratori de l'ESAII sense cap programa carregat a l'interior.

4.4.1.1. Protocol de comunicació

Tal com reflecteix el capítol de disseny electrònic més endavant, les dues plaques Arduino es comuniquen a través del protocol I2C. Aquest protocol té diversos punts a favor respecte a altres protocols de comunicació tals com el CAN o l'SPI. En l'I2C, la fàcil addició de més dispositius al canal obre la possibilitat d'engrandir el projecte en un futur, i el fet que sigui un protocol *Multi-master* li dóna a I2C un gran avantatge, però vegem a continuació alguns trets distintius de cada tipus de comunicació:

	Baud rate	Cost	Ocupació de pins	Direccionalitat	Concepte	Dificultat d'implementació
SPI	25Mbps	baix	4 (MOSI, MISO, CLK, CS)	Full-duplex	Unimàster, multiesclau	Baixa
I2C	1Mbps	baix	2 (SDA, SCL)	Half-duplex	multimàster multiesclau	Alta
CAN	1Mbps	alt	2 (SDA, SCL)	Full-duplex	multimàster multiesclau	Alta

Taula 4. 1 Comparativa dels principals protocols de comunicación

El protocol SPI pot semblar d'entrada una opció força vàlida per al projecte: en destaquen una alta velocitat de transmissió i el fet que sigui full-duplex. Aprofundint una mica en la funcionalitat i les característiques de CASPER, ens n'adonem que aquests avantatges aparents no suposen cap benefici (o un benefici mínim) per a la comunicació.

Per una banda, la velocitat de transmissió tan alta no és útil, ja que s'ha comprovat que per a l'actual projecte l'Arduino no pot processar la informació a un ritme tan elevat. Per altra banda, el fet que sigui full-duplex tampoc ens proporciona un avantatge real perquè no hi ha necessitat de tenir dos canals de comunicació oberts a la vegada entre dos dispositius.

4.4.2. Dynamixel AX-12A

Els smart-servomotors escollits per al CASPER permeten la transmissió de dades del motor cap a l'Arduino. Així es pot saber en tot moment en quina posició es troba el servomotor. A més, mitjançant l'espai de memòria EEPROM del processador amb què compta l'AX-12 es pot establir una temperatura llindar de treball, una velocitat màxima a la qual poden actuar o fins i tot uns límits de posició. D'aquesta manera es poden evitar sobreescalfaments o trencaments de l'eix de rotació, punt molt important quan es tracta d'un robot social que interactua amb nens. El fet que sigui un hardware funcional amb senyals tipus UART (Universal Asynchronous Receiver-Transmitter) que permet regular la velocitat de transmissió de dades (fins a 1Mbps), és també un factor important, tot i que no s'ha utilitzat aquesta velocitat de transmissió per la dificultat de l'AtMega de processar la informació dels 4 motors a la vegada. Finalment el *baud rate* utilitzat ha estat de 115200bps.

Figura 4. 6 El Dynamixel Ax-12 (Font: [11])

Hi ha un conjunt de biblioteques amb diferents llibreries per Arduino disponibles per tal de poder programar els Dynamixels des del mateix Arduino, mitjançant la comunicació sèrie entre aquests i la placa. Per altra banda, s'ha de tenir en compte el fet que l'Arduino actuador ha de ser capaç de transmetre i llegir les dades que arriben del Dynamixel i a la vegada gestionar la informació que percep de l'altre Arduino a través de la comunicació I2C.

És per això que per a aquest treball s'ha utilitzat la biblioteca de SavageElectronics disponible al web SourceForge, en concret la llibreria "DynamixelSerial". Aquesta llibreria permet la comunicació sèrie Arduino-Dynamixel sense entorpir a la vegada la comunicació I2C Arduino-Arduino, de manera que l'Arduino actuador pot rebre i emmagatzemar dades de manera simultània.

4.4.3. Sensors tàctils

Com a sensors tàctils s'ha utilitzat una placa de coure que ha estat retallada per obtenir una mida adequada que encaixi amb el robot. La clau del sensat rau en el codi del programa, més que en el hardware en si. A grans trets el que es fa és calcular el temps que pren un impuls a enviar-se d'un pin i arribar a un altre. En l'esquema de la figura 4.7 es mostra el funcionament elèctric del sensor.

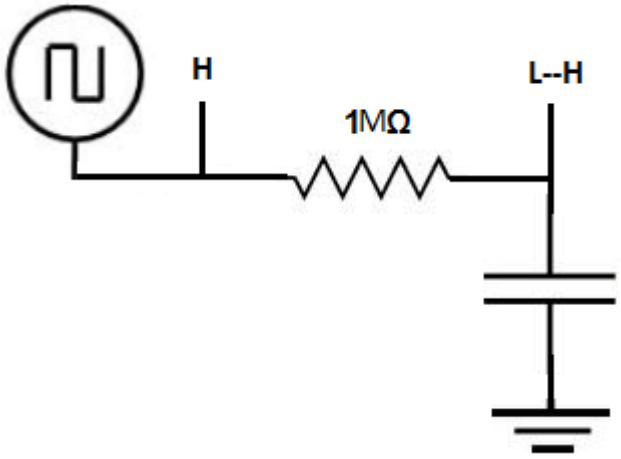


Figura 4. 7 Esquema elèctric representatiu del sensor tàctil (Font: pròpia)

El pin d'enviament passa a 1 i es porta un compte del nombre de cicles (mitjançant una variable que s'incrementa dintre d'un bucle *while*) que passen fins que el pin de rebuda passa de Low a *High*. Aquest *delay* es veu modificat per la capacitat en paral·lel, més concretament per una constant de temps definida com R*C, en què aquesta C seria la capacitat al pin de rebuda sumada a qualsevol altra capacitat (per exemple, una interacció amb el cos humà i la placa de coure). Així doncs, la resistència juga també un paper important: la sensibilitat del sensor augmenta proporcionalment amb el valor d'aquesta resistència. Si en lloc d'una R de 1MΩ, se n'agafa una de major valor (10MΩ), podem fins i tot obtenir un senyal sense necessitat d'entrar en contacte directe amb el sensor, cosa que podria ser molt útil en cas que es recobrís l'estructura de la tortuga amb alguna tela o silicona.

Quant a la llibreria utilitzada per a la part de programació, Arduino en proporciona una creada per Paul Badger anomenada "CapSense" a través de la qual es poden implementar tants sensors com pins estiguin disponibles.

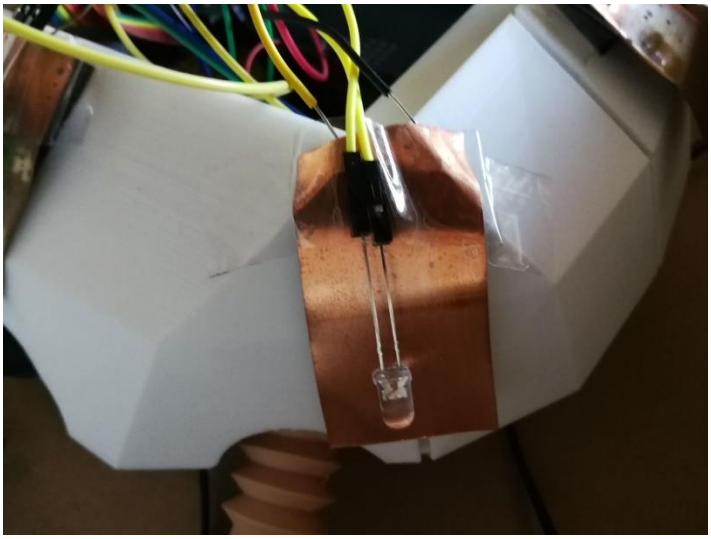


Figura 4. 8 Aparença física del sensor (Font: pròpia)

4.4.4. Altaveu

L'altaveu que s'ha utilitzat és un altaveu passiu ZJFL de 0.5 W i 8 Ω. D'aquesta manera aconseguim minimitzar el consum. S'ha programat perquè sonin les notes musicals que se li demanen a través de l'emissió de freqüències a diferents nivells mitjançant la llibreria "pitches". Així doncs, per tal d'emetre les notes musicals Do, Re, Mi, Fa, Sol, La, Si de la quarta octava s'utilitza la llibreria de la següent manera:

#define NOTE_C4 262 //C4 correspon al Do de la quarta octava (es troba a una freqüència de 262 Hz)

#define NOTE_D4 294 //D4 correspon al Re de la quarta octava (es troba a una freqüència de 294 Hz)

#define NOTE_E4 330 //E4 correspon al Mi de la quarta octava (es troba a una freqüència de 330 Hz)

#define NOTE_F4 349 //F4 correspon al Fa de la quarta octava (es troba a una freqüència de 349 Hz)

#define NOTE_G4 392 //G4 correspon al Sol de la quarta octava (es troba a una freqüència de 392 Hz)

#define NOTE_A4 440 //A4 correspon al La de la quarta octava (es troba a una freqüència de 440 Hz)

#define NOTE_B4 494 //B4 correspon al Si de la quarta octava (es troba a una freqüència de 494 Hz)

4.5. Disseny electrònic

El disseny electrònic es va testejar primerament en protoboard, i d'aquesta manera s'han corregit els errors que s'han trobat abans de fer-ne el prototip final. L'aspecte del disseny electrònic del sistema musical es troba representat en la figura 4.9.

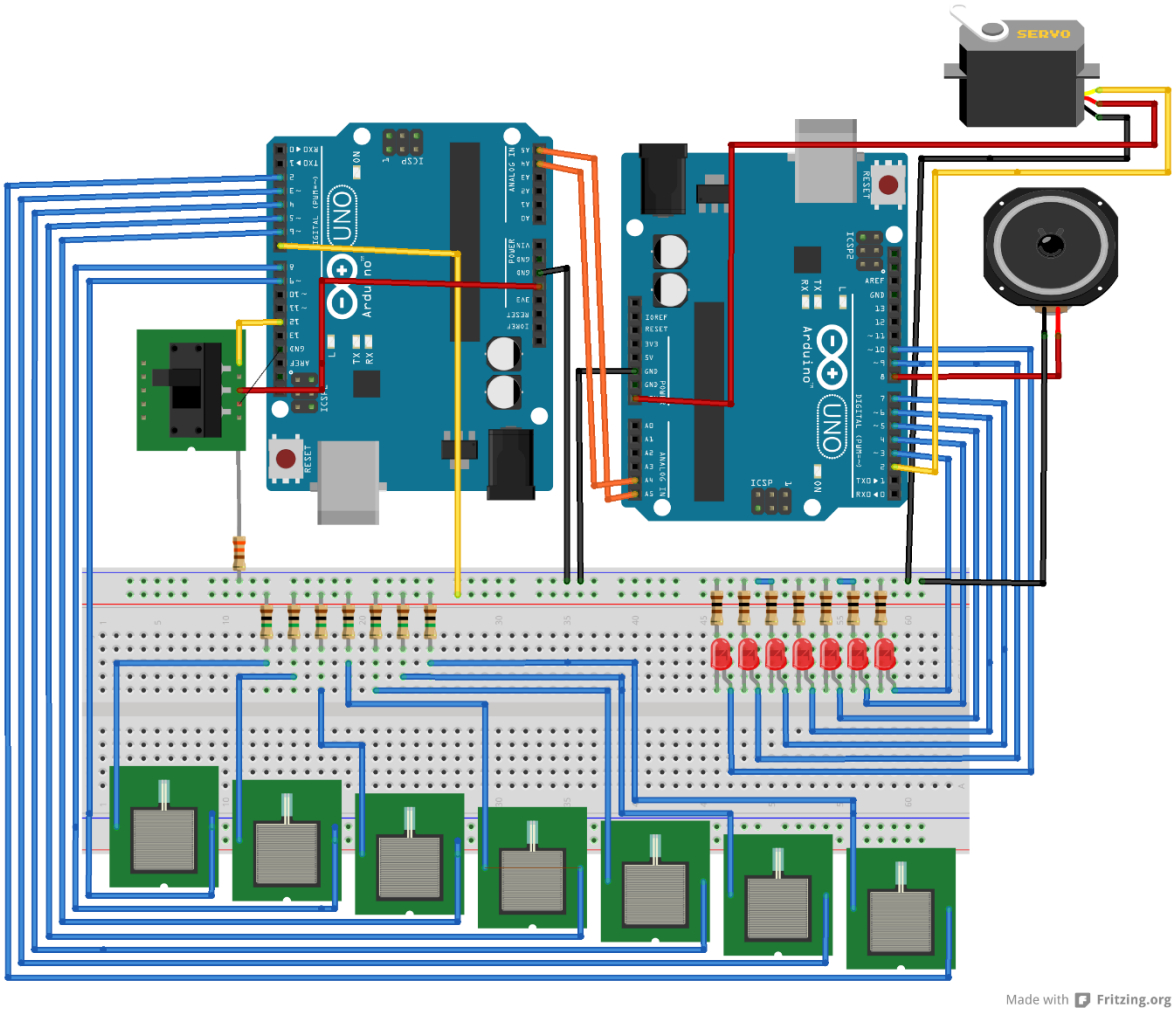


Figura 4. 9 Prototip del sistema musical en protoboard (Font: pròpia)

En la figura 4.10 s'observa l'esquema electrònic del sistema.

Un cop s'ha testejat el funcionament correcte del sistema s'ha passat a soldar en placa de forats el prototip definitiu, amb un aspecte similar al de la figura 4.11.

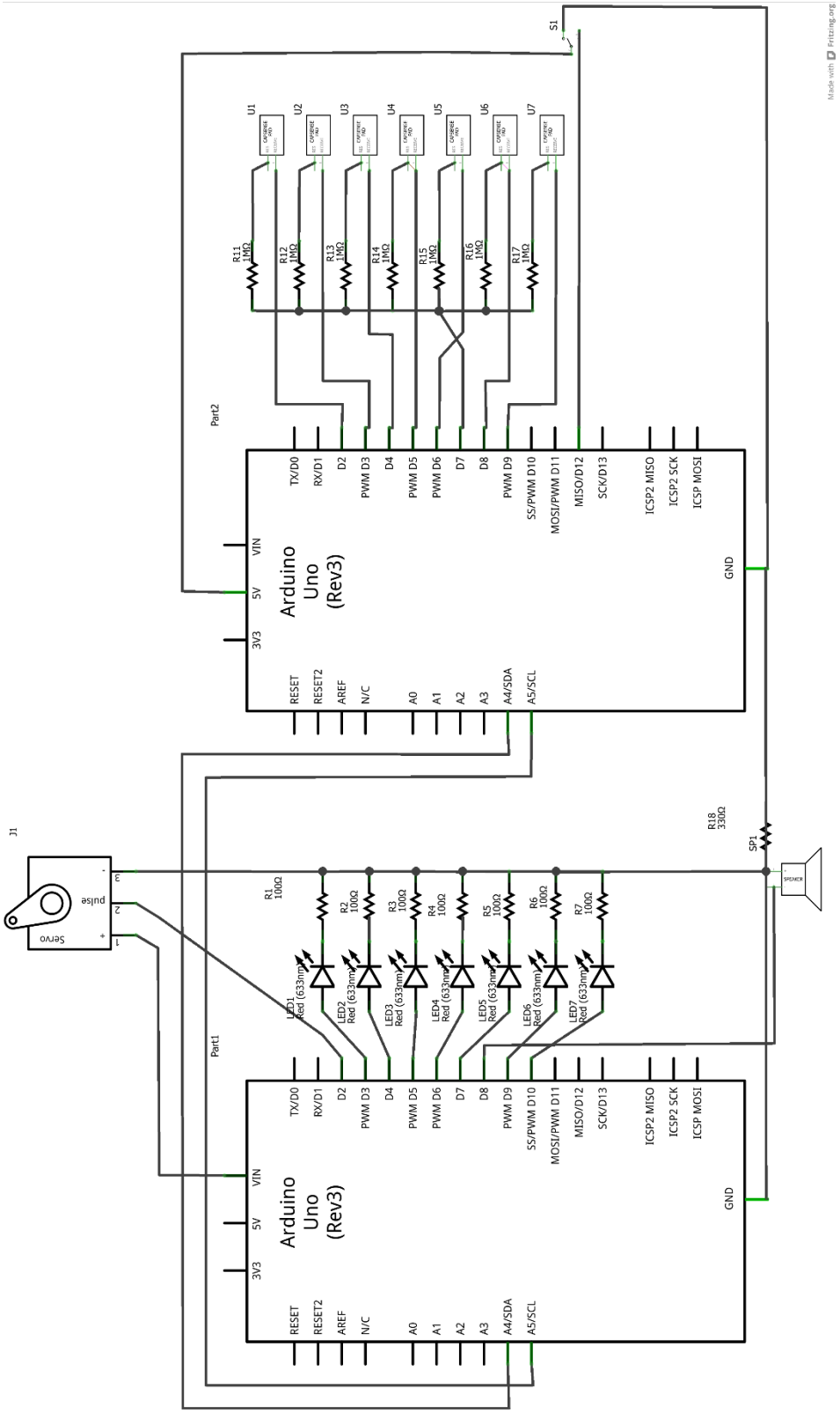


Figura 4. 10 Esquema electrònic del sistema musical (Font: pròpia)

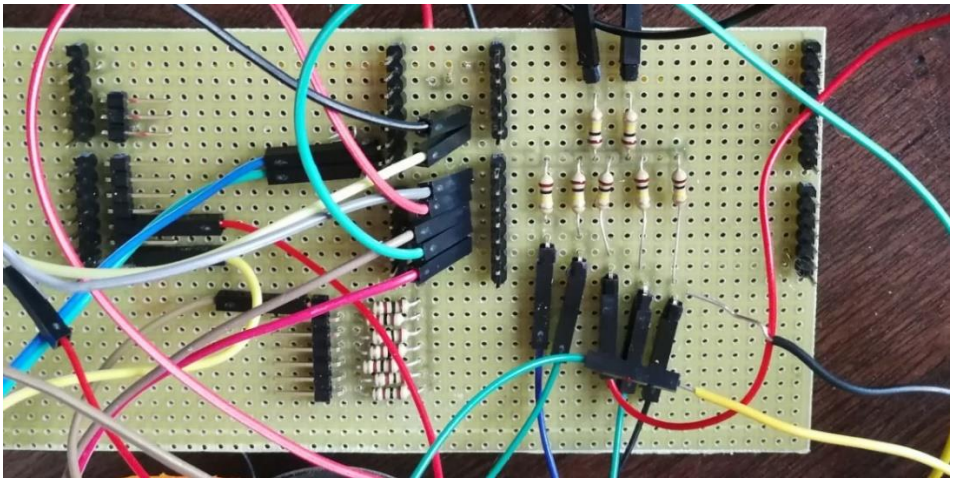


Figura 4. 11 Disseny electrònic parcial en placa de forats (Font: pròpia)

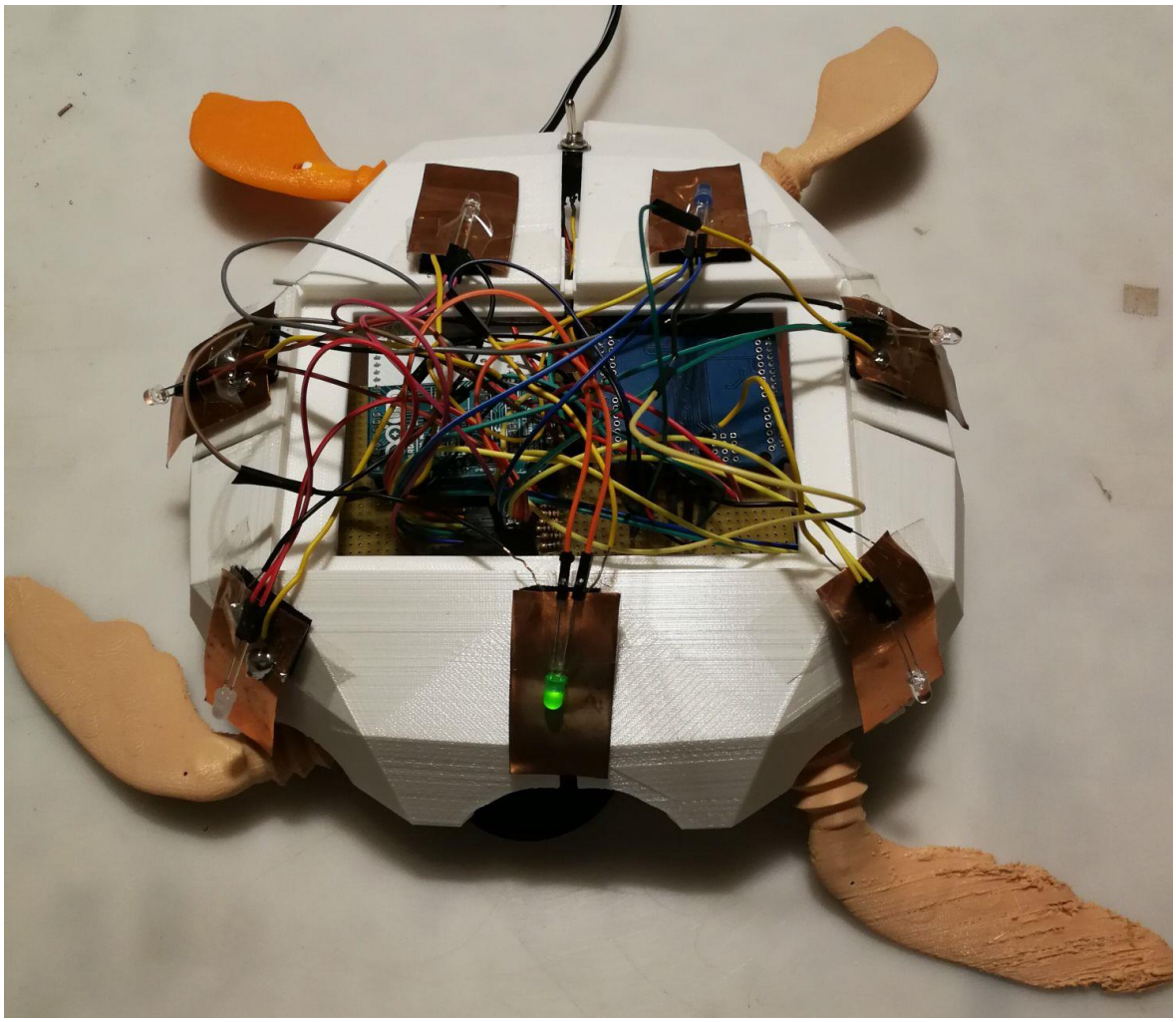


Figura 4. 12 CASPER amb placa i motors implementats (Font: pròpia)

5. Programació

5.1. Ordinograma i descripció del funcionament

Quant al sistema musical de CASPER que s'ha implementat, a grans trets hi ha dos modes de funcionament: un d'ells és el de seguiment d'una melodia, i l'altre és el mode "lliure" en el qual es poden tocar les notes que es vulguin per tal de crear una melodia. Aquests dos modes estan implementats en ambdós Arduino: és important remarcar que cada Arduino té funcions als dos modes de joc, un fent de sensor i l'altre d'actuador. Vegem a continuació l'ordinograma del funcionament:

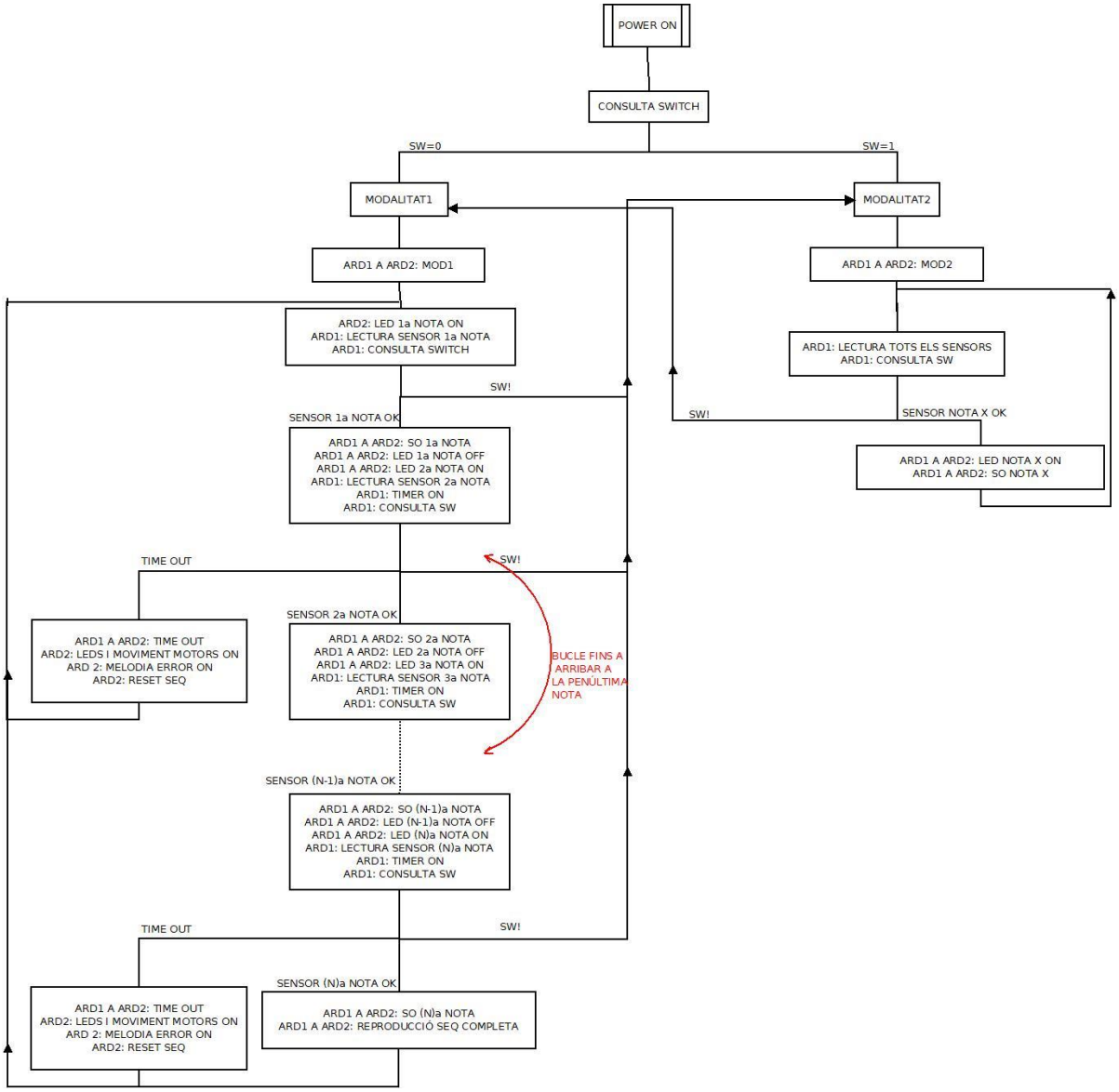


Figura 4. 13 Ordinograma del funcionament del programa (Font: pròpia)

5.1.1. Mode de seguiment

En aquest mode el nen ha de tocar les parts de la tortuga que s'il·luminen per fer sonar una melodia concreta. Hi ha 7 sensors tàctils, un per cada nota de l'escala musical natural (do, re, mi, fa, sol, la, si), i 7 llums led situats en cada un dels sensors. Cada cop que es toca la part de la tortuga il·luminada, sona la nota corresponent al sensor. Seguidament el led del sensor que s'ha tocat s'apaga i s'il·lumina el led adjacent al sensor de la següent nota de la cançó. D'aquesta manera, el nen reprodueix un conjunt de sons que acaben formant una melodia. Ara bé, si passats 5 segons després que s'il·lumini el led, el nen no toca el sensor, el progrés es perd. Sona una curta melodia i el robot es mou (el so de la melodia es relaciona clarament amb un error, i el moviment del robot representa aquesta disconformitat amb el fet que no s'hagi aconseguit completar el joc) i es torna a començar la cançó des del principi.

Periòdicament es consulta l'estat del *switch* (emulador d'una funció superior que vindria donada pel menú selectiu de la RaspberryPi), per tal de poder canviar de modalitat de joc.

5.1.2. Mode de creació

En aquest mode el nen pot tocar qualsevol sensor de la tortuga per tal de fer sonar la nota que vulgui. Cada sensor correspon a una nota, i així pot crear la seva pròpia melodia. Així doncs, quan es toca el sensor corresponent al Do sonarà aquesta nota, i s'il·luminarà el led corresponent al sensor. Quan es deixa de tocar el sensor, automàticament l'Arduino sensor (a l'ordinograma, ARD1) envia un senyal d'stop a l'Arduino actuador (a l'ordinograma, ARD2) per tal d'aturar el so de l'altaveu.

També en aquesta modalitat es verifica d'una manera constant l'estat del *switch* per fer el canvi de modalitat.

5.2. Codificació del protocol de comunicació

Per tal de comunicar els 2 Arduinos s'ha desenvolupat una codificació pròpia. L'Arduino sensor envia una seqüència de números segons les dades que llegeix que l'Arduino actuador interpreta d'una manera concreta i actua en conseqüència. Tot seguit es mostren aquestes seqüències i s'explica de quina manera respondrà l'Arduino actuador.

LECTURA ARD1	SEQÜÈNCIA ENVIADA	ACTUACIÓ ARD2
Switch en posició 0 (lliure)	100	S'entra en modalitat lliure
Sensor 1 detecta contacte	11	Sona un Do. LED sensor 1 ON
Sensor 2 detecta contacte	12	Sona un Re. LED sensor 2 ON
Sensor 3 detecta contacte	13	Sona un Mi. LED sensor 3 ON
Sensor 4 detecta contacte	14	Sona un Fa. LED sensor 4 ON
Sensor 5 detecta contacte	15	Sona un Sol. LED sensor 5 ON
Sensor 6 detecta contacte	16	Sona un La. LED sensor 6 ON
Sensor 7 detecta contacte	17	Sona un Si. LED sensor 7 ON
No es detecta cap contacte	22	Altaveu OFF. LED OFF.
LECTURA ARD1	**SEQÜÈNCIA ENVIADA**	**ACTUACIÓ ARD2**
Switch en posició 1 (seguiment)	101	S'entra en modalitat seguiment
S'ha tocat el sensor adequat	1	Sona nota pertinent
El temps s'ha esgotat	123	Sona seqüència d'error

Taula 5. 2 Codificació de la comunicació entre Arduinos (Font: pròpia)

5.3. Codi Arduino Sensor

Aquest codi correspon al programa que s'ha carregat a l'Arduino encarregat de llegir les dades dels sensors capacitius, així com de l'interruptor que permet canviar la modalitat de joc. A grans trets, es troben dues funcions diferents segons l'estat d'aquest interruptor mitjançant la variable "opció".

En la primera opció, la de modalitat lliure, es porta una lectura constant de tots els sensors mitjançant un bucle *for*, que avalua un sensor per cada iteració, i envia una ordre a l'Arduino actuador assenyalant a quin sensor s'ha detectat un contacte.

En la segona opció, la de modalitat de seguiment, també s'utilitza un bucle *for*, que itera en un *array* que conté la seqüència de sensors que cal polsar per tal de fer sonar la melodia correcta. A més, inclou un *timer* per tal de detectar quan ha passat un temps determinat i fer saltar l'alarma i fer un *reset* de la seqüència.

```
//s'inclou llibreria comunicació I2C
#include <WSWire.h>
//s'inclou llibreria de sensor capacitiu
#include <CapacitiveSensor.h>


// R de 10M entre pins 7 i 2
CapacitiveSensor   cs_7_2 = CapacitiveSensor(7, 2);
// R de 10M entre pins 7 i 3
CapacitiveSensor   cs_7_3 = CapacitiveSensor(7, 3);
CapacitiveSensor   cs_7_4 = CapacitiveSensor(7, 4);
CapacitiveSensor   cs_7_5 = CapacitiveSensor(7, 5);
CapacitiveSensor   cs_7_6 = CapacitiveSensor(7, 6);
CapacitiveSensor   cs_7_8 = CapacitiveSensor(7, 8);
CapacitiveSensor   cs_7_9 = CapacitiveSensor(7, 9);
int sensor0;
//definim sensors
int sensor1 =  cs_7_2.capacitiveSensor(30);
int sensor2 =  cs_7_3.capacitiveSensor(30);
int sensor3 =  cs_7_4.capacitiveSensor(30);
int sensor4 =  cs_7_5.capacitiveSensor(30);
int sensor5 =  cs_7_6.capacitiveSensor(30);
int sensor6 =  cs_7_8.capacitiveSensor(30);
int sensor7 =  cs_7_9.capacitiveSensor(30);
//definim pin de switch
int swPin = 12;
void setup() {
  Wire.begin();   //unió al bus I2C
  cs_7_2.set_CS_AutocaL_Millis(0xFFFFFFFF);
  //apaguem autocalibració del primer canal
  Serial.begin(9600);
  pinMode(swPin, INPUT);
}


void loop() {
  //periode que tardara a saltar l'alarma
  int period = 7000;
  //lectura switch
  int opcion = digitalRead(swPin);
  opcion = digitalRead(swPin);
```

```
  ////////////////////////////////////////////////////////////////////MODALITAT LLIURE
  if (opcion == 0) {
    //enviem 100 si estem en la modalitat lliure de joc
    Wire.beginTransmission(8);
    Wire.write(100);
    Wire.endTransmission();
    delay (100);
    //variable que itera en tots els sensors
    int  thisMeasure = 1;
    int measure[] = {sensor0, sensor1, sensor2, sensor3, sensor4, sensor5, sensor6,
sensor7};
    for (thisMeasure; thisMeasure < 8; thisMeasure) {
      //escollim quin data enviem segons quina posició estiguem de measure
      switch (thisMeasure) {

        case 1: //si s'activa el primer sensor
          sensor1 =  cs_7_2.capacitiveSensor(30);
          Serial.print("SENSOR1:");
          Serial.print(thisMeasure);
          if (sensor1 > 50) {
            Wire.beginTransmission(8);
            Wire.write(11); //11 és el codi del primer sensor
            Wire.endTransmission();
            delay(500);
          }
          break;

        case 2:
          sensor2 =  cs_7_3.capacitiveSensor(30);
          Serial.print("SENSOR2:");
          Serial.print(thisMeasure);
          if (sensor2 > 50) {
            Wire.beginTransmission(8);
            Wire.write(12);
            Wire.endTransmission();
            delay(50);
          }
          break;


        case 3:
          sensor3 =  cs_7_4.capacitiveSensor(30);
          Serial.print("SENSOR3:");
          Serial.print(sensor3);
          if (sensor3 > 50) {
            Wire.beginTransmission(8);
            Wire.write(13);
            Wire.endTransmission();
            delay(50);
          }
          break;


        case 4:
          sensor4 =  cs_7_5.capacitiveSensor(30);
          Serial.print("SENSOR4:");
          Serial.print(sensor4);
          if (sensor4 > 50) {
            Wire.beginTransmission(8);
            Wire.write(14);
            Wire.endTransmission();
            delay(50);
          }
          break;


        case 5:
          sensor5 =  cs_7_6.capacitiveSensor(30);
          Serial.print("SENSOR5:");
          Serial.print(sensor5);
          if (sensor5 > 50) {
```

```
          Wire.beginTransmission(8);
          Wire.write(15);
          Wire.endTransmission();
          delay(50);
        }
        break;


      case 6:
        sensor6 =  cs_7_8.capacitiveSensor(30);
        Serial.print("SENSOR6:");
        Serial.print(sensor6);
        if (sensor6 > 50) {
          Wire.beginTransmission(8);
          Wire.write(16);
          Wire.endTransmission();
          delay(50);
        }
        break;

      case 7:
        sensor7 =  cs_7_9.capacitiveSensor(30);
        Serial.print("SENSOR7:");
        Serial.println(sensor7);
        if (sensor7 > 50) {
          Wire.beginTransmission(8);
          Wire.write(17);
          Wire.endTransmission();
          delay(50);
        }
        break;
    }
    //si sa'arriba al setè sensor es repeteix el for loop
    if (thisMeasure == 7) thisMeasure = 0;
    //enviem senyal d'STOP
    Wire.beginTransmission(8);
    Wire.write(22);
    Wire.endTransmission();
    opcion = digitalRead(swPin);
    Serial.println(opcion);
    if (opcion == 1) break; //canviem de modalitat si canvi sw
    thisMeasure++;
  }
}

////////////////////////////////////////////////////////////////////MODALITAT SEGUIMENT
if (opcion == 1) {
  //enviem modalitat seguiment a ard2
  Wire.beginTransmission(8);
  Wire.write(101);
  Wire.endTransmission();
  int alarm;
  //comencem contador de milis per l'alarma
  int startMillis = millis();
  for (int thisPoint; thisPoint < 27; thisPoint) {
    opcion = digitalRead(swPin);
    Serial.println(opcion);
    //comprovem estat de sw
    if (opcion == 0) {
      break;
      thisPoint = 0;
    }
    //si s'ha activat l'alarma enviem senyal de rst a ard2
    if (alarm == 1) {
      startMillis = millis();  //initial start time
      Wire.beginTransmission(8);
      Wire.write(69);
      Wire.endTransmission();
      alarm = 0;
    }
    int sensoron = 0;
    int start = 0;
```

```
  while (start == 0) {
    //comprovem estat dels sensors
    sensor1 =  cs_7_2.capacitiveSensor(30);
    sensor2 =  cs_7_3.capacitiveSensor(30);
    sensor3 =  cs_7_4.capacitiveSensor(30);
    sensor4 =  cs_7_5.capacitiveSensor(30);
    sensor5 =  cs_7_6.capacitiveSensor(30);
    sensor6 =  cs_7_8.capacitiveSensor(30);
    sensor7 =  cs_7_9.capacitiveSensor(30);

    int sensors[] = {sensor3, sensor4, sensor5, sensor3, sensor1,
                     sensor4, sensor3, sensor2, sensor1, sensor2,
                     sensor2, sensor3, sensor1, sensor3, sensor4,
                     sensor5, sensor3, sensor1, sensor4, sensor3,
                     sensor2, sensor1, sensor2, sensor3, sensor1
                    };
    //limitem el valor dels sensors a 255 per no saturar el canal de comunicació
    if (sensors[thisPoint] >= 255) sensors[thisPoint] = 255;

    Serial.print("VALOR:");
    Serial.print(sensors[thisPoint]); // presenta sensor output 1
    Serial.print("\t");
    //si detectem sensor correcte activat
    if (sensors[thisPoint] >= 50) {
       //s'envia senyal a ard2
      Wire.beginTransmission(8);
      Wire.write(1);
      Wire.endTransmission();
      //sortim del bucle while per passar a la seg nota
      start = 1;
      thisPoint++;
      startMillis = millis();  //resetejem temps de alarma
    }

    int  currentMillis = millis();  //agafem milis actuals
    //si hem superat el temps i ja hem comencat a jugar
    if (currentMillis - startMillis >= period & thisPoint > 0) {
      thisPoint = 0;
      opcion = digitalRead(swPin);
      Serial.println(opcion);
      if (opcion == 0) {  //comprovem estat de sw
        break;
      }
      //si no hi ha hagut canvis enviem senyal alarma
      Wire.beginTransmission(8);
      Wire.write(123);
      Wire.endTransmission();
      Serial.println("time out");
      alarm = 1;
      currentMillis = 0;
      start = 1;
    }
    if (currentMillis - startMillis >= period) {
     //cada periode "period" verifiquem sw
      opcion = digitalRead(swPin);
      Serial.println(opcion);
      if (opcion == 0) {
        break;
        thisPoint = 0;
      }

    }
    if (thisPoint == 25) {
      //si hem arribat al punt 25 (final de melodia)
      start = 1;
      thisPoint = 0;

    }
    delay(300);
  }
}
delay(100);
```

```
}
```

5.4. Codi Arduino Actuador

Aquest codi correspon al codi que s'ha carregat a l'Arduino actuador. Aquí es llegeixen les dades que es reben de l'Arduino sensor i es duen a terme les actuacions dels servomotors, els LEDs i l'altaveu de manera conseqüent amb el que es llegeix a través del bus I2C. Aquesta lectura es du a terme de manera constant a través d'una rutina de tipus ISR. A través d'aquest mecanisme es permet associar una funció a l'ocurrència d'un determinat esdeveniment. És a dir, cada cop que arriben dades a través del bus es salta a la funció de lectura, i s'interromp així el curs d'execució actual per passar a executar el codi específic i tractar aquesta situació.

Igualment en aquest codi també es distingeix entre dos grans funcions segons la modalitat en què ens trobem. A diferència de l'Arduino sensor, aquí esperem a rebre l'estat del *switch* a través del bus de comunicació, ja que és aquest l'encarregat de llegir-ne l'estat, en qualitat d'Arduino sensor. Així doncs, es rebrà a través del bus un "100" o un "101" per decidir en quina funció de joc ens trobem.

En la primera opció que apareix en el codi, la de modalitat lliure, s'espera rebre una codificació des de "11" fins a "17", que indicarà quin sensor ha estat polsat i així farà sonar la nota corresponent i n'encendrà el LED.

En la segona opció rebrem un "1" cada cop que el sensor que toca hagi estat polsat. Quan aquesta condició es doni, farem sonar la nota corresponent de l'*array* mitjançant un bucle for i s'encendrà el LED de la següent nota a polsar, també mitjançant una seqüència integrada en un *array* que itera en un *for loop*. En rebre un "123" sabrem que s'ha esgotat el temps i per tant s'entrarà en el bucle "thisFail", fent sonar la melodia d'error i movent les potes de la tortuga. D'altra banda, si s'arriba a la posició 25 de l'*array* que conté la melodia a seguir, s'haurà completat tota la cançó: es reprodueix automàticament la melodia completa i es torna a començar la seqüència.

```
//s'inclou llibreria per connexió amb els servos
#include <DynamixelSerial.h>
//s'inclou llibreria per comunicació entre arduino
#include <Wire.h>
//s'inclou llibreria per la freq de les notes
#include "Pitches.h"

int sensor = 0;
int thisNote;
int thisLed;
const byte ledPin3 = 3;
const byte ledPin4 = 4;
const byte ledPin5 = 5;
const byte ledPin6 = 6;
const byte ledPin7 = 7;
const byte ledPin9 = 9;
const byte ledPin10 = 10;
```

```
//array leds, mostra l'ordre que s'encendran
//els leds corresponents a cada sensor
int leds[] = {ledPin5, ledPin6, ledPin7, ledPin5, ledPin3,
              ledPin6, ledPin5, ledPin4, ledPin3, ledPin4,
              ledPin4, ledPin5, ledPin3, ledPin5, ledPin6,
              ledPin7, ledPin5, ledPin3, ledPin6, ledPin5,
              ledPin4, ledPin3, ledPin4, ledPin5, ledPin3
             };

//array de melodia, mostra l'ordre de les notes
//que sonaran quan es toqui el corresponent sensor
int melody[] = {NOTE_E4, NOTE_F4, NOTE_G4, NOTE_E4, NOTE_C4,
                NOTE_F4, NOTE_E4, NOTE_D4, NOTE_C4, NOTE_D4,
                NOTE_D4, NOTE_E4, NOTE_C4, NOTE_E4, NOTE_F4,
                NOTE_G4, NOTE_E4, NOTE_C4, NOTE_F4, NOTE_E4,
                NOTE_D4, NOTE_C4, NOTE_D4, NOTE_E4, NOTE_C4
               };
int startMillis;
int currentMillis;
// duració que ha de tenir cada nota segons la melodia
//2 representa un to sencer i 4 mig to
int noteDurations[] = {4, 4, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4,
                       4, 4, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4
                      };

void setup() {
  pinMode(ledPin3, OUTPUT); // configura el pin 3 com sortida
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, OUTPUT);
  pinMode(ledPin7, OUTPUT);
  pinMode(ledPin9, OUTPUT);
  pinMode(ledPin10, OUTPUT);
  Wire.begin(8);                //unió al bus I2C al dispositiu 8
  Wire.onReceive(receiveEvent); // què fer quan es reben dades
  Serial.begin(9600);
  Dynamixel.begin(115200);      //comencem comunicació amb el dynamixel
}

void loop() {

  ////////////////////////////////////////////////////////////////////////MODALITAT LLIURE
  if (opcion == 100) {

    if (sensor == 11) {       //al tocar el primer sensor sona un Do (C)
      tone(8, NOTE_C4, 200);  //C4 representa un Do en la 4a octava
      digitalWrite(ledPin3, HIGH);
      digitalWrite(ledPin4, LOW);
      digitalWrite(ledPin5, LOW);
      digitalWrite(ledPin6, LOW);
      digitalWrite(ledPin7, LOW);
      digitalWrite(ledPin9, LOW);
      digitalWrite(ledPin10, LOW);
    }
    if (sensor == 12) {       //al tocar el segon sensor sona un Re (D)
      tone(8, NOTE_D4, 200);
      digitalWrite(ledPin3, LOW);
      digitalWrite(ledPin4, HIGH);
      digitalWrite(ledPin5, LOW);
      digitalWrite(ledPin6, LOW);
      digitalWrite(ledPin7, LOW);
      digitalWrite(ledPin9, LOW);
      digitalWrite(ledPin10, LOW);
    }
    if (sensor == 13) {
      startMillis = 0;
      currentMillis = millis();
      tone(8, NOTE_E4, 200);
      digitalWrite(ledPin3, LOW);
      digitalWrite(ledPin4, LOW);
      digitalWrite(ledPin5, HIGH);
      digitalWrite(ledPin6, LOW);
```

```
      digitalWrite(ledPin7, LOW);
      digitalWrite(ledPin9, LOW);
      digitalWrite(ledPin10, LOW);
    }
    if (sensor == 14) {
      startMillis = 0;
      currentMillis = millis();
      tone(8, NOTE_F4, 200);
      digitalWrite(ledPin3, LOW);
      digitalWrite(ledPin4, LOW);
      digitalWrite(ledPin5, LOW);
      digitalWrite(ledPin6, HIGH);
      digitalWrite(ledPin7, LOW);
      digitalWrite(ledPin9, LOW);
      digitalWrite(ledPin10, LOW);
    }
    if (sensor == 15) {
      startMillis = 0;
      currentMillis = millis();
      tone(8, NOTE_G4, 200);
      digitalWrite(ledPin3, LOW);
      digitalWrite(ledPin4, LOW);
      digitalWrite(ledPin5, LOW);
      digitalWrite(ledPin6, LOW);
      digitalWrite(ledPin7, HIGH);
      digitalWrite(ledPin9, LOW);
      digitalWrite(ledPin10, LOW);
    }
    if (sensor == 16) {
      startMillis = 0;
      currentMillis = millis();
      tone(8, NOTE_A4, 200);
      digitalWrite(ledPin3, LOW);
      digitalWrite(ledPin4, LOW);
      digitalWrite(ledPin5, LOW);
      digitalWrite(ledPin6, LOW);
      digitalWrite(ledPin7, LOW);
      digitalWrite(ledPin9, HIGH);
      digitalWrite(ledPin10, LOW);
    }
    if (sensor == 17) {
      startMillis = 0;
      currentMillis = millis();
      tone(8, NOTE_B4, 200);
      digitalWrite(ledPin3, LOW);
      digitalWrite(ledPin4, LOW);
      digitalWrite(ledPin5, LOW);
      digitalWrite(ledPin6, LOW);
      digitalWrite(ledPin7, LOW);
      digitalWrite(ledPin9, LOW);
      digitalWrite(ledPin10, HIGH);
    }
  }
  ////////////////////////////////////////////////////////////////////MODALITAT SEGUIMENT
  if (opcion == 101) {
    digitalWrite(ledPin3, LOW);
    digitalWrite(ledPin4, LOW);
    digitalWrite(ledPin5, LOW);
    digitalWrite(ledPin6, LOW);
    digitalWrite(ledPin7, LOW);
    digitalWrite(ledPin9, LOW);
    digitalWrite(ledPin10, LOW);
    //iteració de notes que han de sonar
    for (thisNote = 0; thisNote < 26; thisNote) {
      if (opcion == 100) {  //consulta switch
        break;
        thisLed = 0;
        thisNote = 0;
      }
      int start = 0;
      thisLed = 0;
      while (start == 0) {
```

```
digitalWrite(leds[thisLed], HIGH); //encenem el led corresponent
int noteDuration = 1000 / noteDurations[thisNote]; //definim la duració de la nota
delay(100);

if (sensor == 1) {   //si rebem "1" de l'arduino de sensat, fem sonar nota
  sensor = 0;
  digitalWrite(leds[thisLed], LOW); //apaguem LED corresponent
  tone(8, melody[thisNote], noteDuration); //fem sonar la nota corresponent
  thisNote++;
  thisLed = thisLed + 1; //passem a la seg posició darray i s'encen el seg led

}
int ledState = 1;

if (sensor == 123) {
  int melodyFail[] = {NOTE_AS5, NOTE_A5, NOTE_GS5, NOTE_G5};
  int failDurations[] = {300, 400, 500, 600};

  for (int thisFail = 0; thisFail < 6; thisFail++) {
    Dynamixel.setEndless (1 , ON );
    Dynamixel.setEndless (2 , ON );
    Dynamixel.setEndless (18 , ON );
    Dynamixel.setEndless (19 , ON );
    digitalWrite(ledPin3, ledState);
    digitalWrite(ledPin4, ledState);
    digitalWrite(ledPin5, ledState);
    digitalWrite(ledPin6, ledState);
    digitalWrite(ledPin7, ledState);
    digitalWrite(ledPin9, ledState);
    digitalWrite(ledPin10, ledState);
    Dynamixel.turn(1, LEFT, 500);
    Dynamixel.turn(2, LEFT, 500);
    Dynamixel.turn(18, RIGTH, 500);
    Dynamixel.turn(19, RIGTH, 500);
    if (thisFail <= 3) tone(8, melodyFail[thisFail], failDurations[thisFail]);
    Dynamixel.turn(1, RIGTH, 500);
    Dynamixel.turn(2, RIGTH, 500);
    Dynamixel.turn(18, LEFT, 500);
    Dynamixel.turn(19, LEFT, 500);
    ledState = !ledState;
    if (thisFail >= 5) {
      Dynamixel.setEndless (1 , OFF );
      Dynamixel.setEndless (2 , OFF );
      Dynamixel.setEndless (18 , OFF );
      Dynamixel.setEndless (19 , OFF );
    }
  }
  thisNote = 0;
  thisLed = 0;
}

if (thisNote == 25) {  //si completem la melodia...
  thisLed = 0;
  for (int thisNote = 0; thisNote < 25; thisNote++) {
     //sona la melodia completa
    digitalWrite(leds[thisLed], HIGH);
    int noteDuration = 500 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    digitalWrite(leds[thisLed], LOW);
    thisLed = thisLed + 1;

  }
  thisNote = 0;
  thisLed = 0;
}
Serial.print("SENSORON:");
Serial.print(thisNote);        // presenta sensor output 1
Serial.println("\t");
if (opcion == 100) {
  break;
```

```
          thisNote = 0;
          thisLed = 0;
        }
      }
    }
  }
}

void receiveEvent (int HowMany) {

  sensor = Wire.read(); //llegim el que arriba de l'arduino de sensat
  Serial.println(sensor);
  if (sensor == 100) opcion = 100;
  if (sensor == 101) opcion = 101;

}
```

6. Anàlisi de l'impacte ambiental

Actualment és molt important valorar l'impacte ambiental de qualsevol projecte que es realitzi. A més, si es té en compte el fet que el projecte té possibles previsions futures de comercialització, aquest punt és encara més essencial. S'ha de prendre consciència de tot el que es genera i de quina influència té en el medi, així com el possible impacte en les generacions futures.

El fet que s'hagi creat un objecte físic evidentment genera un impacte més gran del que es podria generar si es duu a terme un treball final de grau més lligat a la recerca. Igualment, però, durant aquest treball s'han dedicat moltes hores a la recerca d'informació i per tant s'analitzarà l'impacte ambiental des de dos punts de vista: el del treball de recerca i el de la generació purament física del robot.

6.1. Impacte d'investigació, recerca i de desenvolupament del prototip

Per desenvolupar aquest treball final de grau s'ha hagut de començar dedicant inicialment moltes hores a la investigació, recerca i lectura d'altres projectes similars i de tècniques de desenvolupament robòtic. Això es tradueix en l'ús d'electricitat per al funcionament del PC, la il·luminació i climatització del laboratori, i en material bàsic d'oficina (paper, fotocòpies, llibres i material de suport...), així com el transport fins a la biblioteca o el laboratori de la universitat i l'ús de tinta d'impressora i de CDs. Tota aquesta llista pot semblar d'entrada quelcom fútil o sense importància, però aquest impacte, que es podria considerar bàsic o inevitable, és realment alt.

Tot i això, s'han intentat utilitzar tots aquests recursos de manera conscient i respectuosa amb el medi ambient.

6.2. Impacte de desenvolupament del robot

Quant a l'impacte del robot pròpiament dit, s'han de tenir en compte diversos factors: el fet de l'ús del material i el cost ambiental del seu desenvolupament, i el cost de l'assemblatge d'aquest material. L'estructura del robot és impresa en 3D, i això té l'avantatge d'evitar el cost ambiental de transport.

En canvi, els microprocessadors Arduino, tot i tenir l'etiqueta RoHS compliant, que prohibeixen l'ús de certs materials nocius tant per a l'home com per al medi, sí que ha generat despeses de transport i de fabricació que contaminen el medi ambient. També s'ha de tenir en compte l'ús de cable, d'estany i del soldador, així com la resta de hardware que compon CASPER.

Pel que fa a l'ús domèstic del robot, evidentment se'n derivaran també despeses energètiques per la càrrega de bateries i la connexió al corrent, així com les despeses associades de transport i comercialització del propi robot.

Si es du a terme la comercialització o la fabricació en sèrie d'aquest robot, s'hauria de tenir en compte sempre l'impacte que se'n generaria, minimitzant sempre les despeses energètiques tant d'electricitat com de combustible fòssil, i mirant de treballar d'una manera sostenible i respectuosa amb el medi ambient.

Conclusions

Un cop finalitzat el treball arriba el moment d'avaluar el projecte que s'ha fet. Personalment, penso que els resultats finals obtinguts són molt satisfactoris i s'han complert els objectius principals que s'havien plantejat a l'inici del projecte. El fet d'haver treballat en un projecte comú ha plantejat algunes dificultats que s'han superat o bé han forçat haver de prendre camins que no s'havien planejat en un principi.

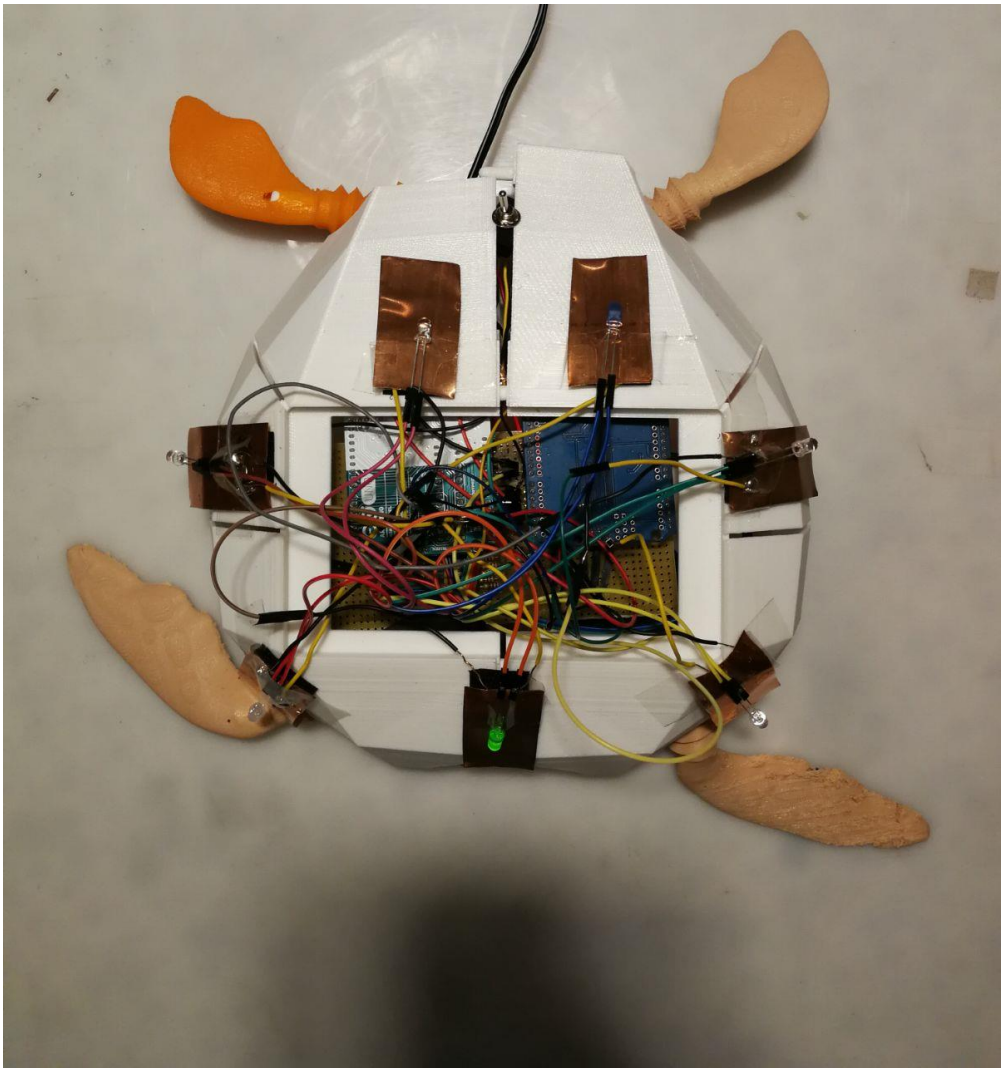
D'entrada, cal remarcar que s'ha aconseguit desenvolupar el sistema musical interactiu i implementar-lo a l'estructura física de CASPER adequadament, i per tant s'ha creat un robot completament autònom i funcional. El fet de no haver treballat amb la RaspberryPi ha facilitat la realització d'aquest treball, però a la vegada en dificulta la implementació al projecte general. Tot i que se n'ha fet l'emulació amb un *switch*, s'hauria d'aprofundir en la manera d'implementar la RaspberryPi per tal que fes de màster i decidís la modalitat de joc.

Seria realment interessant treballar en comú amb l'autor del treball final de màster que ha dissenyat la HMI controlada a través de la Raspberry (treball que es pot trobar citat a la bibliografia) per tal d'implementar de manera efectiva el sistema musical en aquesta interfície.

Per altra banda, el codi ha estat descrit de manera que sigui fàcil afegir més opcions de joc a les dues modalitats (o bé que se'n puguin crear de noves), i fins i tot s'ha facilitat el fet que es puguin utilitzar els sensors tàctils que enguany s'han implementat a CASPER per funcions que vagin més enllà del sistema musical.

Quant a l'acabat del robot, un aspecte a tenir en compte és la bateria: a mode de prototip se l'utilitza connectat a la corrent. Per la seva possible comercialització, seria convenient adquirir una font d'alimentació portàtil i encabir-la dins el robot per tal que aquest fos operatiu sense dependre d'una connexió a una presa de corrent altern domèstic, ja que el fet de no tenir autonomia en limita el seu ús.

El fet d'haver treballat des de l'inici amb un codi propi ha fet que s'hagués de treballar amb una llibreria genèrica per comunicar els Arduino. Per tant, la llibreria que havia estat desenvolupada expressament per al CASPER [8] no ha pogut ésser utilitzada. La impossibilitat de treballar amb aquesta darrera llibreria ha provocat haver de fer un treball doble. Tot i això, s'ha superat l'entrebanc i la comunicació entre microcontroladors funciona correctament: no tots els problemes tenen una única solució, i en aquest cas s'ha aconseguit arribar a l'objectiu final mitjançant una altra metodologia.



Il·lustració: El robot CASPER (Font: pròpia)

Finalment, m'agradaria dir que aquest treball final de grau m'ha donat peu per aprofundir en un camp que trobo personalment molt interessant, el de la robòtica. Espero que aquest projecte sigui el primer de molts altres, i que el treball fet pugui ser útil per continuar amb un projecte tant robòtic i alhora humà com el de CASPER.

Bibliografia

6.3. Referències bibliogràfiques

- [1] F. Petric, D. Miklić, M. Cepanec, P. Cvitanović and Z. Kovačić, "Functional imitation task in the context of robot-assisted Autism Spectrum Disorder diagnostics: Preliminary investigations," *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Lisbon, Portugal, 2017, pp. 1471-1478.

- [2] R. S. De Silva, K. Tadano, M. Higashi, A. Saito and S. G. Lambacher, "Therapeutic-assisted robot for children with autism," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, 2009, pp. 3561-3567.

- [3] H. Feng, A. Gutierrez, J. Zhang and M. H. Mahoor, "Can NAO Robot Improve Eye-Gaze Attention of Children with High Functioning Autism?," *2013 IEEE International Conference on Healthcare Informatics*, Philadelphia, PA, 2013, pp. 484-484.

- [4] Devyn Curley, Alex Barco, Sandra Pico, Pablo Gallego, Dimitris Zervas, Cecilio Angulo, Beste Ozcan, Julien Delvaux, Matthieu Lhoir, Jordi Albo-Canals, 2nd International Conference on Social Robots i Therapy and Education, "New Friends", 2016.

- [5] J. M. Beer, M. Boren and K. R. Liles, "Robot assisted music therapy a case study with children diagnosed with autism," *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Christchurch, 2016, pp. 419-420.

- [6] Jonas Londschien, *Mechanical parts of the CASPER robotic platform using a 3D printer.* Universitat Politècnica de Catalunya, 2016.

- [7] Daniel Capelo, *Implementation of Computer Vision Algorithms for the CASPER Robot.* Universitat Politècnica de Catalunya, 2017.

- [8] Thomas Herpoel, *CASPER social assistive robotics platform: top down System design and intracommunication protocol.* Haute École Louvain en Hainaut, 2016.

6.4. Consultes bibliogràfiques

- [10] www.herts.ac.uk/kaspar (accés:5 agost, 2017)

- [11] https://www.tedxbarcelona.com/2016/05/09/pleo-el-uso-de-un-robot-dinosaurio-en-el-resumen-del-evento-tedxbarcelona/ (accés: 10 agost, 2017)

- [12] https://www.robotlab.com/store/ask-nao-autism-solution-for-kids (accés: 14 agost, 2017)

- [13] https://www.arduino.cc/en/Tutorial/toneMelody (accés: 5 setembre, 2017)

- [14] https://playground.arduino.cc/Main/CapacitiveSensor?from=Main.CapSense (accés: 6 setembre, 2017)

- [15] https://github.com/steamfire/WSWireLib (accés, 13 setembre, 2017)

- [16]http://support.robotis.com/en/product/actuator/dynamixel/ax_series/dxl_ax_actuator.htm (accés: 25 octubre, 2017)

- [17]http://www.savageelectronics.blogspot.com.es/2011/01/arduino-y-dynamixel-ax-12.html (accés: 26 octubre, 2017)

- [18] https://learn.sparkfun.com/tutorials/i2c (accés: 4 novembre, 2017)

- [19] http://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/ (accés: 6 novembre, 2017)

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

IMPLEMENTACIÓ D'UN SISTEMA MUSICAL INTERACTIU AL *SOCIAL PET ROBOT* CASPER

Pressupost i anàlisi econòmica

Autor: Albert Tibau Font
Director: Sebastián Tornil
Ponent: Cecilio Angulo
Convocatòria: Gener 2018

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

Pressupost i anàlisi econòmica

En aquest capítol s'analitzarà quin cost econòmic ha tingut el procés de realització del treball i del prototip de robot CASPER.

Primerament es mostra quina ha estat la despesa en materials i components necessaris per al desenvolupament del robot.

Component	Preu/unitat	Quantitat	TOTAL €
Material oficina	20€	/	20
Protoboard	15€	2	30
Placa forats	10€	1	10
Cable	5€	/	5
Arduino	20€	2	40
Estany	10€	/	10
Resistències	0,1€	40	4
LEDs	0,5€	10	5
Altaveu	5€	1	5
Servos	45€	4	180
Placa coure sensors	7€	1	7
Switch	2€	1	2
SUBTOTAL			318€

Taula 7. 1Pressupost de components i materials (Font: pròpia)

Quant al cost dels recursos humans:

Descripció	Preu/hora	Quantitat	TOTAL €
Transport	20€	20	400
Estudiant	20€	650	13.000
SUBTOTAL		670h	13.400€

Taula 7. 2. Cost de recursos humans (Font: pròpia)

A continuació es mostra una taula amb el desglossament de les hores dedicades al treball, i en què han estat invertides:

Descripció	Preu/hora	Hores	TOTAL €
Estudi de l'objecte del treball	20	100	2.000
Disseny de hardware	20	100	2.000
Desenvolupament en *protoboard*	20	60	1.200
Assemblatge de hardware	20	10	200
Desenvolupament de software	20	150	3.000
Software *debugging*	20	100	2.000
Proves funcionals	20	30	600
Documentació	20	100	2.000
SUBTOTAL		650h	13.000€

Taula 7. 3. Desglossament d'hores invertides en el projecte (Font: pròpia)

Finalment, s'obté el cost total del projecte, tenint en compte els recursos humans emprats i el material que ha estat necessari:

Descripció	Preu	Quantitat	TOTAL €
Recursos humans	20€/h	670h	13.400
Materials i components	318€		318
TOTAL			**13.718€**

Taula 7. 4. Costos totals relatius al projecte (Font: pròpia)

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

IMPLEMENTACIÓ D'UN SISTEMA MUSICAL INTERACTIU AL *SOCIAL PET ROBOT* CASPER

Annexos

Autor: Albert Tibau Font
Director: Sebastián Tornil
Ponent: Cecilio Angulo
Convocatòria: Gener 2018

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

Annex A

A1. Especificacions del servomotor Dynamixel AX-12A

H/W Specification

- Weight : 53.5g (AX-12/AX-12+), 54.6g (AX-12A)
- Dimension : 32mm * 50mm * 40mm
- Resolution : 0.29°
- Gear Reduction Ratio : 254 : 1
- Stall Torque : 1.5N.m (at 12.0V, 1.5A)
- No load speed : 59rpm (at 12V)
- Running Degree
  - 0° ~ 300°
  - Endless Turn
- Running Temperature : -5℃ ~ +70℃
- Voltage : 9 ~ 12V (Recommended Voltage 11.1V)
- Command Signal : Digital Packet
- Protocol Type : Half duplex Asynchronous Serial Communication (8bit,1stop,No Parity)
- Link (Physical) : TTL Level Multi Drop (daisy chain type Connector)
- ID : 254 ID (0~253)
- Communication Speed : 7343bps ~ 1 Mbps
- Feedback : Position, Temperature, Load, Input Voltage, etc.
- Material : Engineering Plastic

Stall torque is the maximum instantaneous and static torque
Stable motions are possible with robots designed for loads with 1/5 or less of the stall torque

Control Table

Control Table consists of data regarding the current status and operation, which exists inside of Dynamixel. The user can control Dynamixel by changing data of Control Table via Instruction Packet.

EEPROM and RAM

Data in RAM area is reset to the initial value whenever the power is turned on while data in EEPROM area is kept once the value is set even if the power is turned off.

Address

It represents the location of data. To read from or write data to Control Table, the user should assign the correct address in the Instruction Packet.

Access

Dynamixel has two kinds of data: Read-only data, which is mainly used for sensing, and Read-and-Write data, which is used for driving.

Initial Value

In case of data in the EEPROM Area, the initial values on the right side of the below Control Table are the factory default settings. In case of data in the RAM Area, the initial values on the right side of the above Control Tables are the ones when the power is turned on.

Highest/Lowest Byte

In the Control table, some data share the same name, but they are attached with (L) or (H) at the end of each name to distinguish the address. This data requires 16bit, but it is divided into 8bit each for the addresses (low) and (high). These two addresses should be written with one Instruction Packet at the same time.

Address Function Help

EEPROM Area

Model Number

It represents the Model Number.

Firmware Version

It represents the firmware version.

ID

It is a unique number to identify Dynamixel.

The range from 0 to 252 (0xFC) can be used, and, especially, 254(0xFE) is used as the Broadcast ID. If the Broadcast ID is used to transmit Instruction Packet, we can command to all Dynamixels.

> Please be cautious not to have the same IDs for the connected dynamixels. You may face communication issues or may not be able to search when IDs overlap.

Baud Rate

It represents the communication speed. 0 to 254 (0xFE) can be used for it.

This speed is calculated by using the below formula.

Speed(BPS) = 2000000/(Data+1)

Data	Set BPS	Target BPS	Tolerance
1	1000000.0	1000000.0	0.000 %
3	500000.0	500000.0	0.000 %
4	400000.0	400000.0	0.000 %
7	250000.0	250000.0	0.000 %
9	200000.0	200000.0	0.000 %
16	117647.1	115200.0	-2.124 %
34	57142.9	57600.0	0.794 %
103	19230.8	19200.0	-0.160 %
207	9615.4	9600.0	-0.160 %

> Note : Maximum Baud Rate error of 3% is within the tolerance of UART communication.

Return Delay Time

It is the delay time per data value that takes from the transmission of Instruction Packet until the return of Status Packet.

0 to 254 (0xFE) can be used, and the delay time per data value is 2 usec.

That is to say, if the data value is 10, 20 usec is delayed. The initial value is 250 (0xFA) (i.e., 0.5 msec).

CW/CCW Angle Limit

The angle limit allows the motion to be restrained.

The range and the unit of the value is the same as Goal Position(Address 30, 31).

- CW Angle Limit: the minimum value of Goal Position(Address 30, 31)
- CCW Angle Limit: the maximum value of Goal Position(Address 30, 31)

The following two modes can be set pursuant to the value of CW and CCW.

Operation Type	CW / CCW
Wheel Mode	both are 0
Joint Mode	neither at 0

The wheel mode can be used to wheel-type operation robots since motors of the robots spin infinitely. The joint mode can be used to multi-joints robot since the robots can be controlled with specific angles.

A2. Llista de material emprat

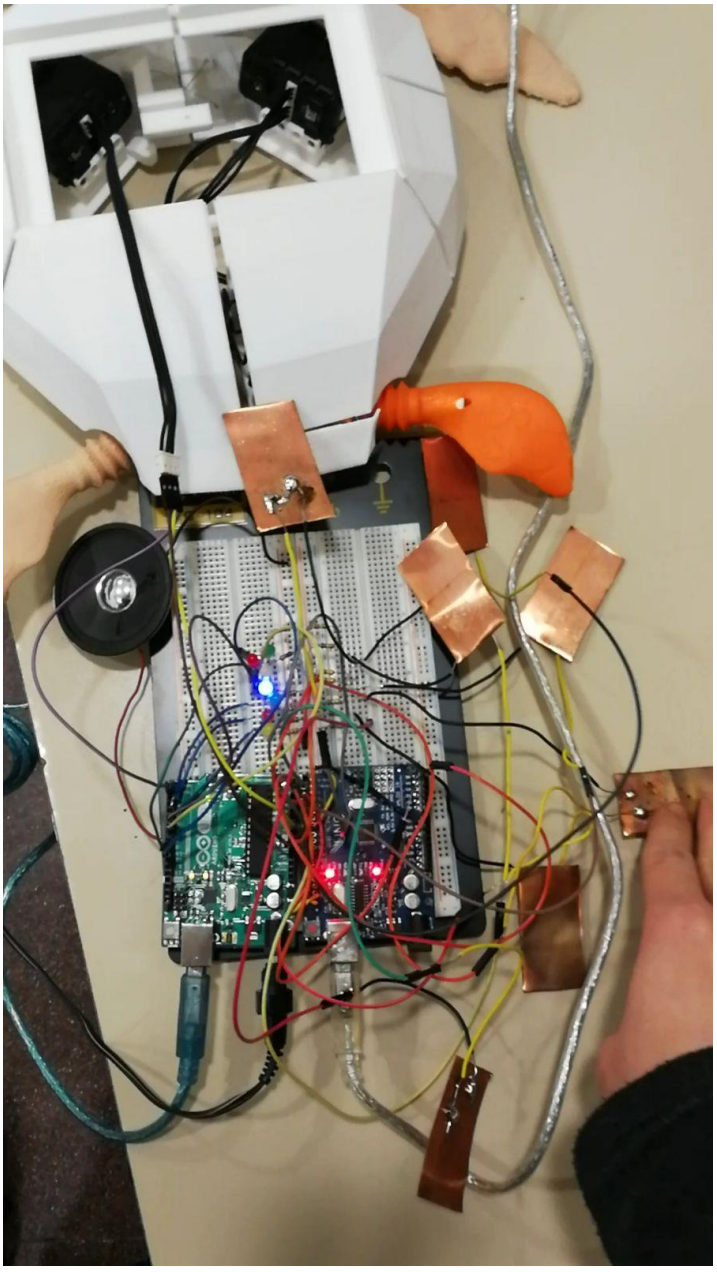
Assembly List

Label	Part Type	Properties
J1	Dynamixel AX-12A	4 servomotors in serial
LED1	Red LED - 3mm	package 3 mm [THT]; leg yes; color Red (633nm)
LED2	Red LED - 3mm	package 3 mm [THT]; leg yes; color Red (633nm)
LED3	Red LED - 3mm	package 3 mm [THT]; leg yes; color Red (633nm)
LED4	Red LED - 3mm	package 3 mm [THT]; leg yes; color Red (633nm)
LED5	Red LED - 3mm	package 3 mm [THT]; leg yes; color Red (633nm)
LED6	Red LED - 3mm	package 3 mm [THT]; leg yes; color Red (633nm)
LED7	Red LED - 3mm	package 3 mm [THT]; leg yes; color Red (633nm)
Part1	Arduino Uno (Rev3)	type Arduino UNO (Rev3)
Part2	Arduino Uno (Rev3)	type Arduino UNO (Rev3)
R1	100 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 100Ω; pin spacing 400 mil
R2	100 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 100Ω; pin spacing 400 mil
R3	100 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 100Ω; pin spacing 400 mil
R4	100 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 100Ω; pin spacing 400 mil

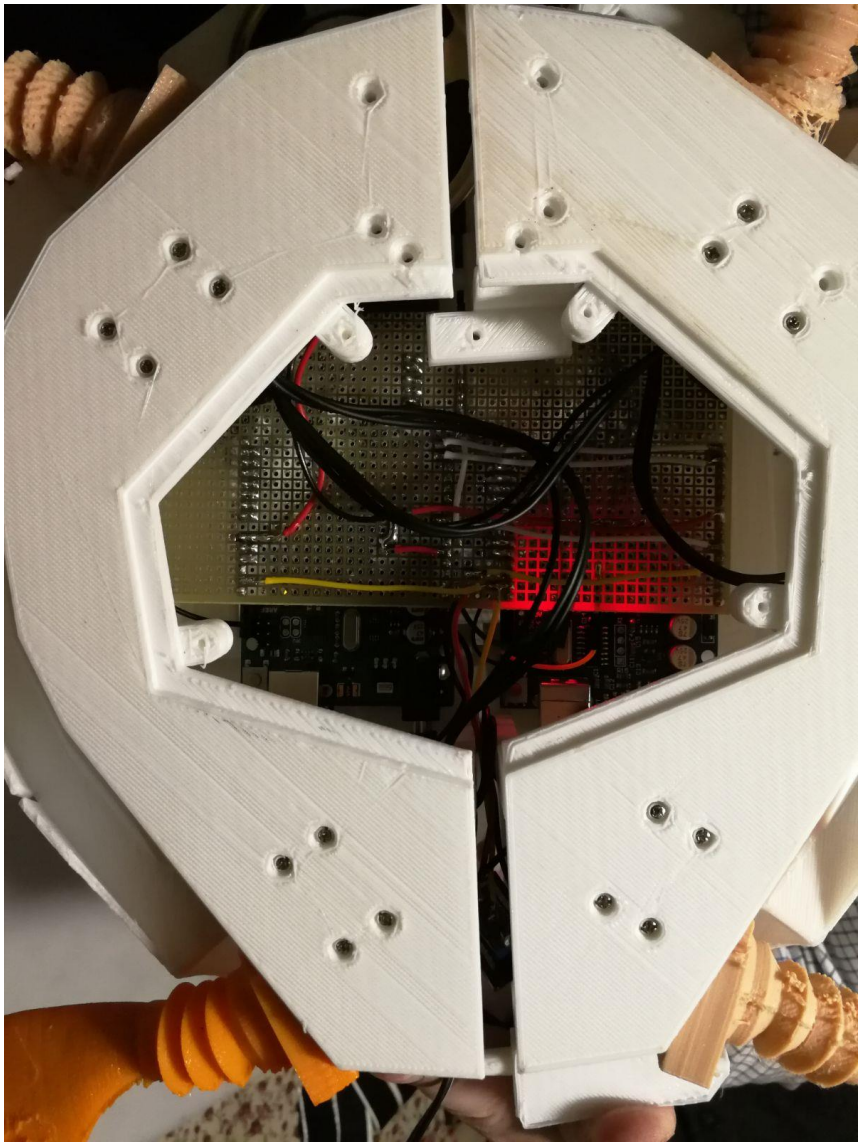
Label	Part Type	Properties
R5	100 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 100Ω; pin spacing 400 mil
R6	100 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 100Ω; pin spacing 400 mil
R7	100 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 100Ω; pin spacing 400 mil
R11	1M Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 1MΩ; pin spacing 400 mil
R12	1M Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 1MΩ; pin spacing 400 mil
R13	1M Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 1MΩ; pin spacing 400 mil
R14	1M Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 1MΩ; pin spacing 400 mil
R15	1M Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 1MΩ; pin spacing 400 mil
R16	1M Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 1MΩ; pin spacing 400 mil
R17	1M Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 1MΩ; pin spacing 400 mil
R18	330 Ω Resistor	package THT; tolerance ±5%; bands 4; resistance 330Ω; pin spacing 400 mil
S1	SPST Switch	package switche-dpdt; variant pth2
SP1	SPEAKER	package pcb_mount_speaker
U1	CAPSENSE_PAD	package capsense_pad
U2	CAPSENSE_PAD	package capsense_pad
U3	CAPSENSE_PAD	package capsense_pad

Label	Part Type	Properties
U4	CAPSENSE_PAD	package capsense_pad
U5	CAPSENSE_PAD	package capsense_pad
U6	CAPSENSE_PAD	package capsense_pad
U7	CAPSENSE_PAD	package capsense_pad

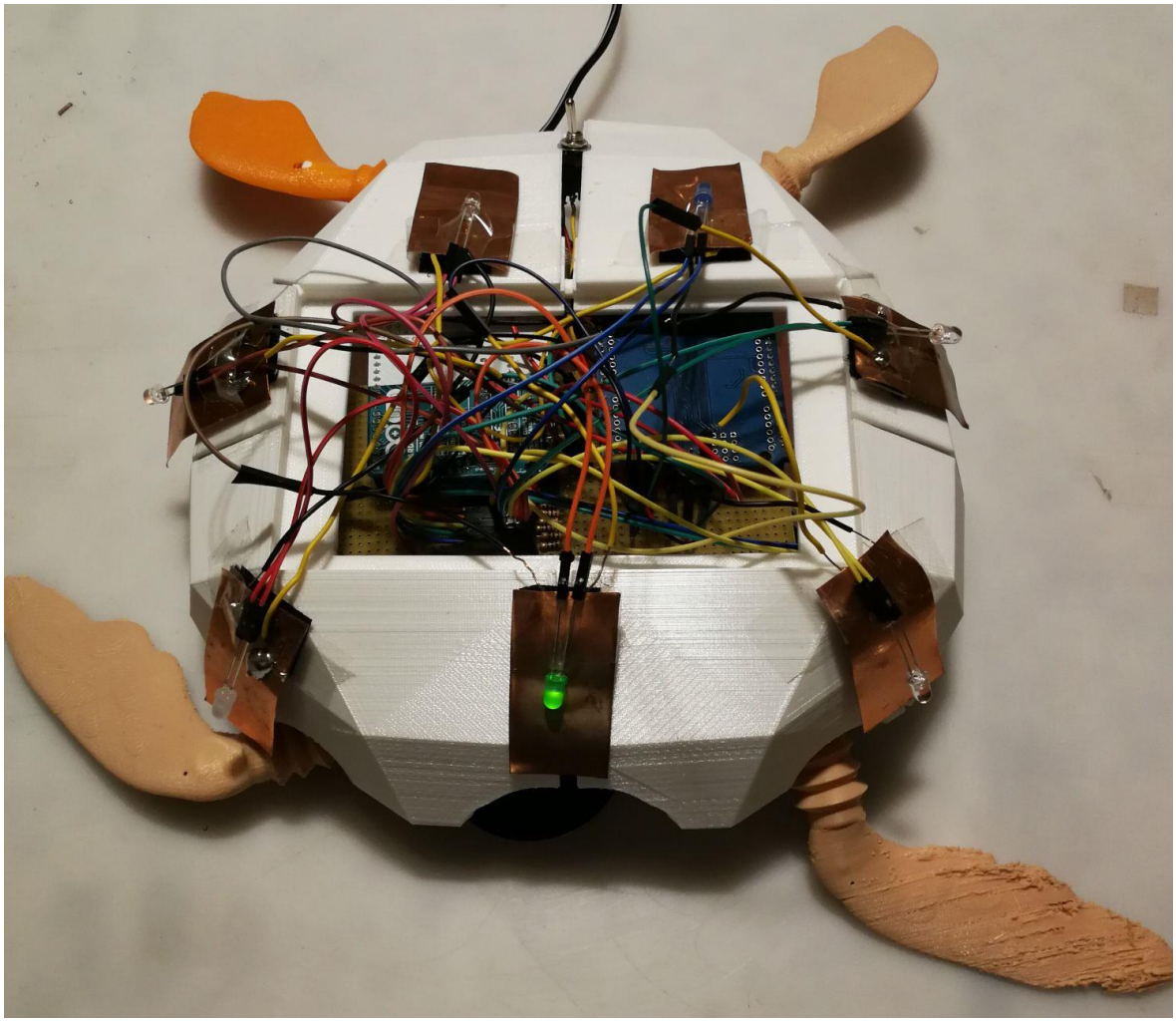
A3. Reportatge multimèdia



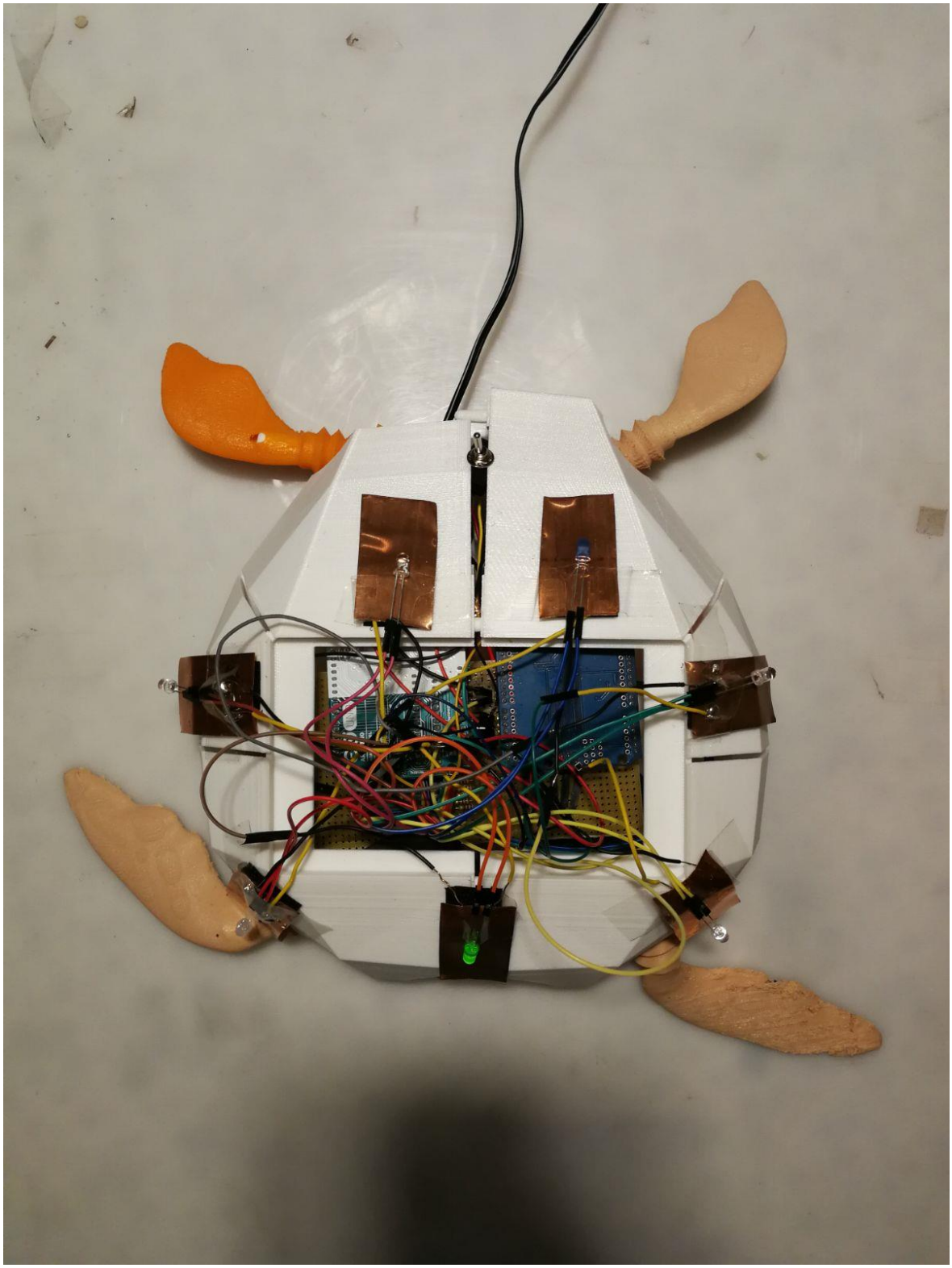
A3.1. 1 CASPER en test amb protoboard



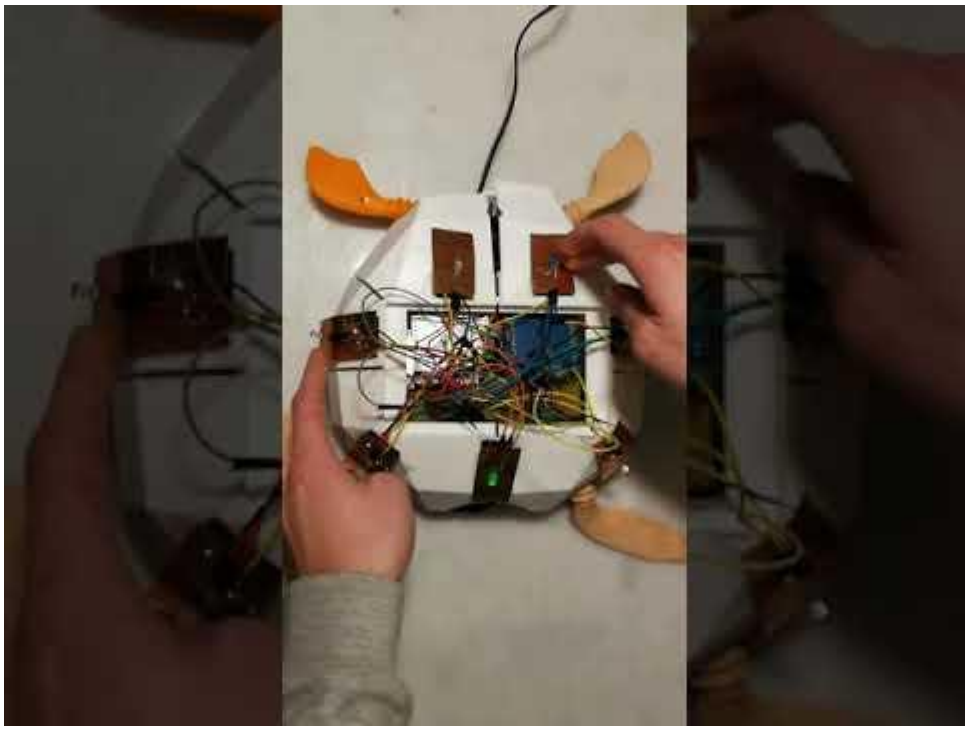
A3.1. 2. Vista inferior de casper amb el processador a l'interior



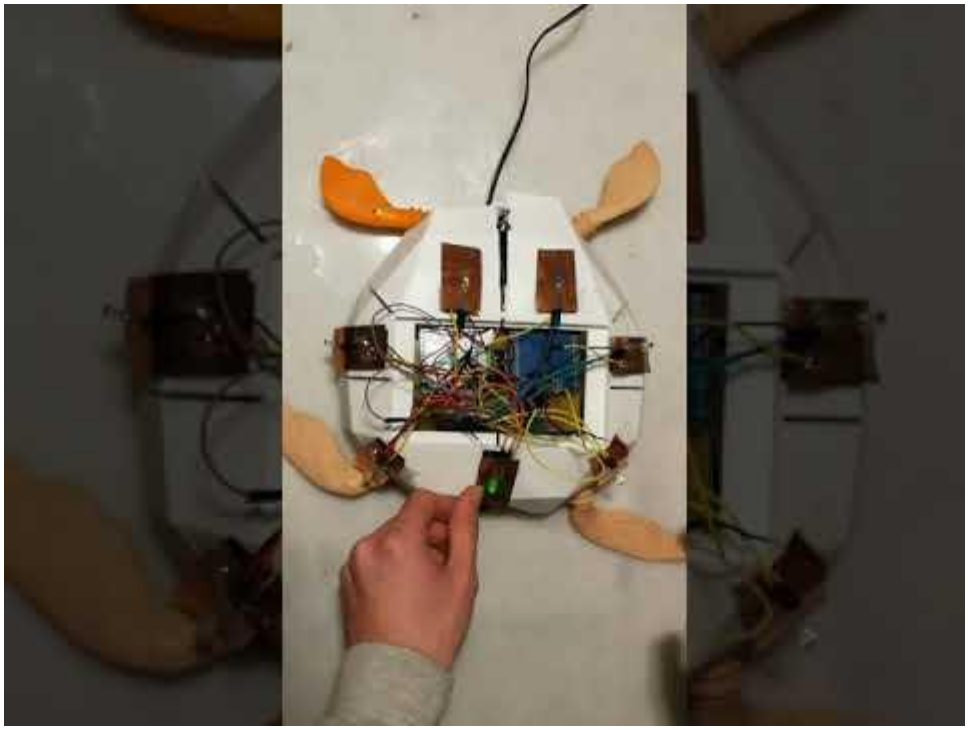
A3.1. 3. CASPER en funcionament



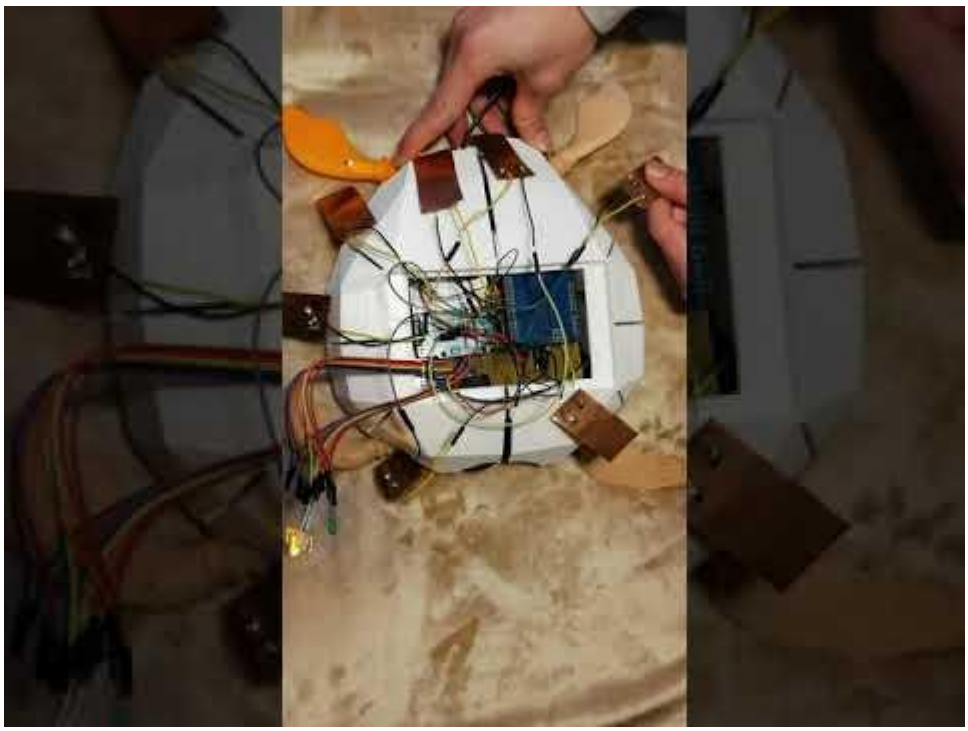
A3.1. 4. Vista superior de CASPER



A3.1. 5. Vídeo de CASPER en les dues modalitats (https://www.youtube.com/watch?v=kZUzTSvVUlo)



A3.1. 6. CASPER en modalitat de seguiment completa el joc i arriba al final de la melodia (https://www.youtube.com/watch?v=QO98rnM4daw)



A3.1. 7. CASPER no completa la melodia i no acaba el joc (https://www.youtube.com/watch?v=QFEDVlyAI8U)