

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

MEMORIA TITULACIÓN

**Diseño y construcción de plataforma
para estudios en enjambre de robots.**

Autor:

Supervisores:

*Memoria para optar al grado de Bachiller en Ciencias de la Ingeniería y
Licenciado en Ciencias de la Ingeniería Electrónica.*

en el

Departamento de Electrónica

11 de julio de 2013

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

Resumen

Departamento de Electrónica

Bachiller en Ciencias de la Ingeniería y Licenciado en Ciencias de la Ingeniería
Electrónica.

Diseño y construcción de plataforma para estudios en enjambre de robots.

por Sebastián SÁEZ

Desarrollo de una plataforma, de bajo costo, que permita estudiar el comportamiento de grupos de robots. Para que la plataforma funcione es necesario el diseño y construcción de los robots , implementar sistema de tracking 2D, armar la red de comunicación (XBee) y construir el software de control.

Acknowledgements

Este trabajo no habría podido realizarse sin el apoyo incondicional de mi profesora tutora, Dra. María José Escobar que ha tenido una paciencia infinita conmigo y siempre me ha ayudado en lo posible. También fue fundamental el apoyo del Dr. Juan Cristóbal Zagal que gentilmente me permitió trabajar en su innovador laboratorio de Síntesis de máquinas inteligentes. No menos importante fue el apoyo del Dr. Pablo Prieto que ha sido un guía para mí en el diseño del robot. Además de ellos conté con la ayuda de mucha gente que sea han dedicado a escuchar mis ideas y aconsejarme. Es por esto que quiero agradecer a Jaime Martínez por ayudarme a diseñar el PCB de MODI, Carlos Galáz por sus infinitos minutos telefónicos para discutir casi todos los aspectos de MODI, Fabián Rubilar por sus ideas de como mover los robots, Linus Casassa por su paciencia para explicarme cualquier cosa sobre electrónica, la gente del taller de electrónica por enseñarme a soldar y sacarme de apuros técnicos, Ricardo Pérez por las interminables discusiones sobre el robot, mis compañeros del laboratorio de Síntesis de máquinas inteligentes y a mi amada Pía que siempre ha estado ahí conmigo siendo un gran apoyo en mi vida.

Índice general

Abstract	I
Acknowledgements	II
Índice de Figuras y Tablas	v
1. Robot	1
1.1. Introducción	1
1.2. Desarrollo tecnológico desde Chile	3
1.3. ¿Qué es un robot?	3
2. Swarm	6
2.1. Comportamiento colectivo en la naturaleza	6
2.2. Robots para construir un enjambre	7
2.2.1. Kilobot	8
2.2.2. Organismo Multibot	8
2.2.3. e-puck	8
2.2.4. 3pi Robot	8
2.3. Necesidades de mercado	10
3. MODI	12
3.1. Diseño	12
3.2. Construcción	13
3.2.1. Fabricación Digital vs Análoga	13
3.2.2. Software CAD	16
3.2.3. Impresora 3D	16
3.2.4. Diseño PCB	21
3.3. Componentes electrónicos	22
3.3.1. Arduino FIO	24
3.3.2. Motores DC	24
3.3.3. XBee	24
3.3.4. Lipo Rider Pro	25
3.3.5. Motor Driver 1A Dual TB6612FNG	25
3.4. Componentes mecánicos	26
3.5. Locomoción	28

3.6. Implementación	30
3.6.1. Setup	30
3.6.2. Software	31
3.6.3. Tracking 2D	32
3.6.4. Bill Of Material	32
4. Aplicaciones	35
4.1. Auto modelamiento	35
4.2. Educación	37
4.3. Usos Militar	38
4.4. Usos Doméstico	38
5. Mejoras futuras y Conclusiones	39
5.1. Mejoras futuras	39
5.1.1. Encoders	39
5.1.2. Cables incrustados y Soporte motor	39
5.1.3. Una PCB para todo	40
5.1.4. Módulo de carga inalámbrica	41
5.2. Conclusiones	41
A.	43
Bibliografía	46

Índice de figuras

1.1. Robot Huggable	1
1.2. Robot Digesting Duck	4
1.3. Robots Roomba y KUKA	5
2.1. Bandada de auklets	6
2.2. Robots estudiados como alternativa para armar un enjambre de robots.	9
2.3. Tabla comparativa robots	10
2.4. Nicho de mercado	11
3.1. Robot MODI	12
3.2. Comparación de construcción análoga y digital	14
3.3. Mapa fab labs en el mundo.	15
3.4. Comparación entre el primer render realizado y el último	17
3.5. Modelo funcionamiento Impresora 3D	18
3.6. Tabla comparativa de impresoras 3D	19
3.7. Impresoras 3D	20
3.8. Diagrama de conexiones eléctricas necesarias para unir todos los componentes	21
3.9. Fabricación de PCB	22
3.10. Componentes Electrónicos	23
3.11. Tabla características mas importantes de los componentes Eléctronicos	26
3.12. Piezas 3D	27
3.13. Robot Atlas	28
3.14. Gráficos Motor DC	29
3.15. Señal pwm	30
3.16. Setup Enjambre MODI	31
3.17. Fiduciales usados como tag en reacTIVision	33
3.18. Bill Of Materials	34
4.1. Esquema algoritmo automodelado Hod Lipson	37
5.1. Cables incrustados y Soporte motor	40
A.1. Historia de construcción	44
A.2. Diagrama eléctrico de conexiones en PCB MODI	45

Pía, mi fiel compañera de aventuras...

Capítulo 1

Robot

1.1. Introducción

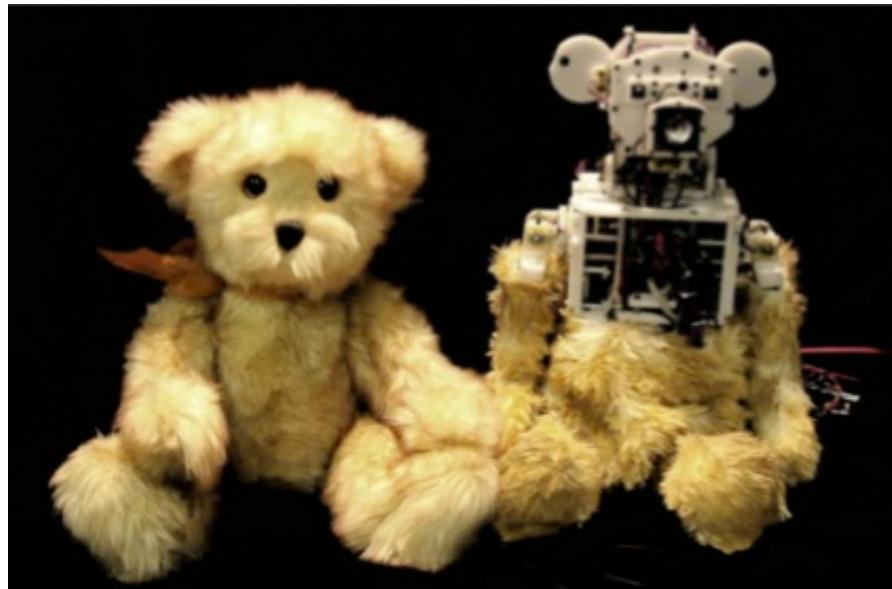


FIGURA 1.1: Robot creado por MIT Media Lab para cuidados personales. Imagen tomada de [1]

En la década de 1990, teniendo tan solo 6 años me encontraba escondido en la logia de la casa de mis padres, que estaba ubicada en pleno barrio Ñuñoa Santiago, desarmando cuanto artefacto cayera en mis manos. Segundo recuerdo, más lo que mi madre me cuenta, desarmé el único teléfono que había en la casa. Era uno de esos que ya no se ven, que hacen uso un dial con pulsos para marcar el número. Tal era mi curiosidad sobre esta máquina, que la desarme completamente solo para tratar de entender como se producía ese sonido sin igual que llamaba la atención de toda persona que estuviera en la casa. Por su puesto mi madre me retó porque no pude volver a armarlo, pero en secreto

siempre supe que ella y mi padre se divertían con mis pequeñas aventuras. En esta misma casa, mi padre intento montar una empresa de consultores con unos amigos, lo que significaba que al lado de mi pequeño taller clandestino tenía a mi disposición 3 o 4 computadores con sus flamantes pantallas Hércules donde pasé tardes completas jugando los primeros video juegos y sin darme cuenta mis primeros pasos en la computación. Algo más grande, cuando tenía 13 años, por el trabajo de mi padre nos fuimos a vivir a Concepción. Una ciudad bien particular, ya que el invierno da mucho tiempo para estar en el hogar. Ahí tuve mi primer gran acercamiento a la computación, recuerdo que mientras mis compañeros estaban en clases de inglés en la sala, yo me escabullía para ir directo a las salas de computación, donde el encargado me permitía ayudarle a instalar Windows 95 en los computadores. Disquete tras disquete íbamos intercambiándolos en cada computador hasta lograr que el famoso logo de la ventana saliera en la pantalla. Desde esa época que familiares y amigos me han pedido ayuda con sus maquinas. En la enseñanza media volvimos como familia a Santiago, volví a mi colegio de infancia, el Liceo San Agustín, donde junto a dos compañeros nos inscribimos en mi primer concurso, el concurso escolar de robótica de la UDP. Aquí alumnos de la universidad nos enseñaron como se podía construir y programar un robot, no fue fácil y luego de un par de meses entre tantos manuales y la simbología alienígena (diagramas eléctricos) tomamos la decisión de retirarnos. En el año 2005 ingresé a la UTFSM, llegué el 2006 a Valparaíso y lo primero que vi en el patio central fue un pequeño cartel que decía: "Taller del Centro de Robótica", junto a otro amigo que hice apenas ingresé a la USM entusiasmados con esta idea de hacer nuestros propios robots, no lo dudamos y nos inscribimos para ser parte de este grupo de alumnos. Una vez dentro del Centro de Robótica (CR), conocí mucha gente con intereses similares a los míos, con gran conocimiento y más que nada, una gran solidaridad para compartir este conocimiento. Tardes completas dedicadas a aprender las oscuras artes de la electrónica, luchando con la frustración de armar un circuito y que este no funcionara a la primera, fui paso a paso avanzando hasta poder construir mis primeros robots. Fue mucha la gente que me ha ayudado en el camino y espero este trabajo pueda inspirar a cualquiera que se interesen por la robótica y la tecnología.

1.2. Desarrollo tecnológico desde Chile

Hacer Robots en Valparaíso no es una tarea sencilla, la falta de lugares especializados para comprar dificulta contar con los componentes necesarios para construir una máquina.

Al comienzo de este proyecto se trabajó en Santiago de Chile, en conjunto con la Universidad de Chile (UChile) bajo la tutela del Doctor Juan Cristóbal Zagal en el Laboratorio

de Síntesis de Máquinas Inteligentes. Por parte de la Universidad Técnica Federico Santa María (USM), se trabajó con la Doctora María José Escobar, del Departamento de Electrónica y luego se incorporó el Doctor Pablo Prieto del Departamento de Diseño de Productos. El financiamiento para realizar este prototipo ha sido por parte de las dos Instituciones, Universidad Técnica Federico Santa María y Universidad de Chile.

Durante el proceso de desarrollo se tuvieron que hacer diversas compras de materiales, pero los lugares más recurrentes al momento de hacerlas fueron Olimex y Casa Royal. La primera es una empresa dedicada a traer productos para hacer prototipos y construir máquinas, la segunda cuenta con varios insumos básicos para trabajar en desarrollo de circuitos electrónicos. Ambas empresas se encuentran en Santiago, por lo que trabajar en esta ciudad es de gran ayuda para reducir los tiempos en desarrollo.

Luego de armar un primer robot básico funcional, con materiales disponibles en Santiago de Chile, se hizo una búsqueda de componentes en tiendas especializadas que permiten comprar en internet. Existen varias (ver anexo 1) pero las más importantes para este proyecto fueron: Sparkfun y Seeedstudio.

1.3. ¿Qué es un robot?

Un robot puede ser un software solamente o tener además una extensión física que le permita interactuar con la realidad y realizar tareas de forma autónoma. Etimológicamente, el término robot se le atribuye al dramaturgo checo Karel Čapek, que en su obra R.U.R en 1921 (Rossum's Universal Robots) utilizó la palabra *robotnik* para referirse a ayudantes artificiales. Luego fue el escritor Isaac Asimov (1920-1992) quien, gracias a su obra, difundió la palabra robótica haciendo referencia a la ciencia encargada de estudiar a los robots. La robótica contempla el estudio de al menos 6 áreas: La mecánica, la electrónica, la informática, el control automático, la física y la matemática.

En la historia hay varios intentos por construir estos ayudantes artificiales. A principios del siglo XVIII, Jacques de Vaucanson creó un autómata capaz de tocar la flauta, así como un pato mecánico que continuamente seguía su ciclo biológico.

En la actualidad las empresas KUKA, Honda y Sony, entre otras, construyen robots especialmente diseñados para la industria. Los robots que se utilizan en la industria, y los pocos que han llegado al hogar, son controlados por un algoritmo. Este es parte de un software que escribe una persona, donde se detalla la tarea que el robot debe realizar; tiene un modelo de los motores, partes y piezas para que así la máquina tenga información de como es, y pueda ejecutar la tarea para la cual se le programó. Si se interfiere con el entorno del robot, por ejemplo moviendo 1 [cm], fuera del rango de

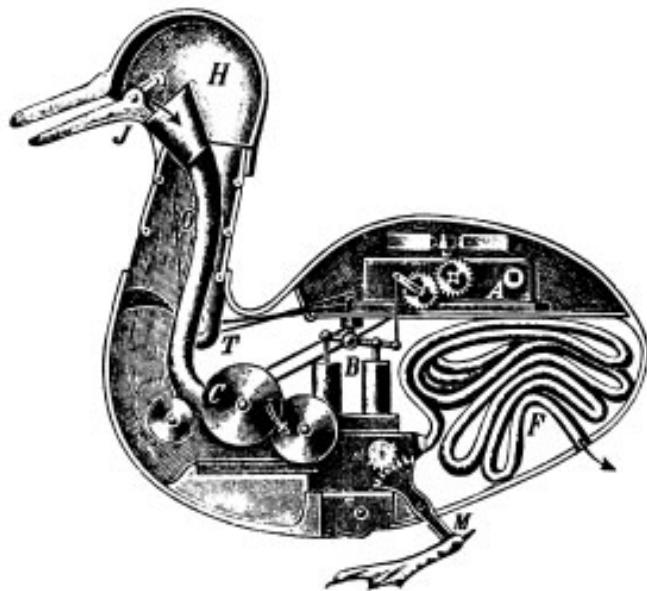


FIGURA 1.2: Digesting Duck, creado por Jacques de Vaucanson en 1739. Imagen tomada de Wikipedia.

los sensores, el perno que debe apretar algún robot industrial que ensambla autos, este no podrá *encontrarlo*. Los robots comerciales que existen hoy en día no son capaces de adaptarse a cambios en el entorno y menos ser capaces de generar una imagen de sí mismos que les permita entender qué sucede y recuperarse de fallas.

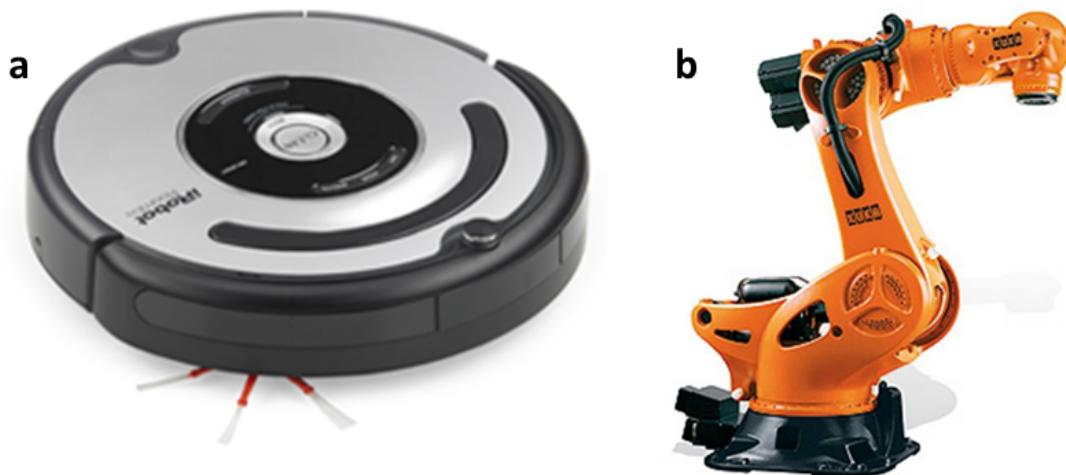


FIGURA 1.3: a. Robot Roomba, primer Robot doméstico vendido en chile. Imagen tomada de [2] b. Robot industrial KUKA KR 1000 TITAM. Imagen tomada de kukarobotics.com

Un algoritmo debe tener un modelo detallado de los motores y sensores que posee el robot, y es tarea del programador hacer la abstracción necesaria para poder darle sentido al movimiento del conjunto de motores. Si hablamos de un robot de 4 extremidades, con

3 grados de libertad en cada una, el programador debe ser capaz de indicar la secuencia de activación de cada motor para así primero hacer que el robot mueva una extremidad y luego con la suma de las 4 lograr desplazarse. Estamos hablando de 12 motores que pueden moverse de forma independiente, lo cual genera infinitas soluciones y no todas posibles debido a las restricciones físicas de la construcción misma del robot.

Capítulo 2

Swarm

2.1. Comportamiento colectivo en la naturaleza



FIGURA 2.1: Bandada de auklets teniendo comportamiento de enjambre. Imagen tomada de Wikipedia.

Existen casos de enjambres, los más típicos son las hormigas y abejas, pero los hay en peces, aves e incluso los mamíferos. Son sistemas donde nadie está a cargo y aún así ejecutan una tarea grupal. Las hormigas son un gran ejemplo. Al momento de construir su nido, no tienen un arquitecto o ingeniero estructural que esté dando órdenes, simplemente cada una sabe que tiene que hacer. No hay un director orquestando la construcción desde lo alto, en vez de esto lo que ocurre es un comportamiento emergente. También conocido como inteligencia de enjambre.

Otro tipo de comportamiento colectivo son las migraciones, desplazamientos periódicos que efectúan aves, peces, langostas y mamíferos de un hábitat a otro. Cada individuo activo en la migración sigue al grupo, los más pequeños como el plancton o anfibios aprovechan las corrientes de aire o agua, y las aves, más grandes, aprovechan los vientos y corrientes ascendentes. Hay diversas finalidades detrás de la migración, algunas especies lo hacen para escaparse de los crudos inviernos o secos veranos; mientras que otras, como las tortugas marinas, por una necesidad reproductiva emprenden un largo viaje de más de 10.000 millas, a lo largo de todo el Atlántico Norte.

Lograr que un enjambre de robots tenga un comportamiento emergente como el de las colonias de abejas es la piedra filosofal de los investigadores de esta área. Uno de los más destacados investigadores del área, James McLurkin, experto en robótica del Massachusetts Institute of Technology (MIT) dice que para lograrlo es necesario un software que ejecute tareas individuales y que de alguna forma se cumpla con una tarea grupal. He aquí una importante razón para desarrollar estudios sobre enjambres de robots, ya que aún no está claro cómo se coordina la naturaleza para llevar a cabo tales tareas.

2.2. Robots para construir un enjambre

Imaginemos una situación hipotética donde un edificio es destruido, la búsqueda de sobrevivientes no es una tarea fácil, implica que rescatistas ingresen al lugar corriendo grave peligro, usualmente buscando a las víctimas en condiciones de poca visibilidad. Esto mismo podría ser ejecutado por un enjambre robótico que esté programado para buscar gente y que de manera colectiva recorra un área mucho mayor que 2 o 3 personas. Incluso un robot del mismo enjambre puede fallar, pero al ser un sistema distribuido el enjambre continúa funcionando, es un sistema muy robusto.

Algunos robots que pueden ser utilizados para hacer un Robot Swarm junto con las información técnica disponible, son listados en la sub secciones siguientes.

2.2.1. Kilobot

El proyecto Kilobot, es un sistema de bajo costo escalable para demostrar comportamientos colectivos. Actualmente existen varios grupos que están investigando algoritmos para enjambres de robots, por esto que diseñaron Kilobot que es un robot de bajo costo, accesible, que permite hacer pruebas en cientos o miles de robots.

2.2.2. Organismo Multibot

S. Kornienko et al. [3], exploran el trabajo colaborativo en robots para un mejor rendimiento y mayor fiabilidad a nivel macroscópico. En este artículo demuestran sus últimos trabajos en sistemas colectivos y lo más sorprendente es que logran la agregación y desagregación autónoma para así obtener un organismo multibot.

2.2.3. e-puck

Uno de los robots más utilizados por los científicos en el mundo para estudios y publicaciones es el e-puck [4]. Este robot es compacto, tiene forma de cilindro con un diámetro de 7 [cm] y para moverse hace uso de sus dos ruedas, dejándolo en la categoría de robot con desplazamiento diferencial. Originalmente fue diseñado para educar en el área de la micro ingeniería por Michael Bonani y Francesco Mondala en el laboratorio ASL del Profesor Roland Siegwart en Escuela Politécnica Federal de Lausana (EPFL) en Suiza. El e-puck es open hardware, software es de código abierto, lo construyen y venden varias empresas. Para comunicarse con una computadora incorporan un módulo Bluetooth conectado a uno de sus dos puertos serie. Existen varios tipos de accesorios, entre los que destacan un Zigbee para comunicaciones, un módulo con varias cámaras y LEDs RGB como sistema de comunicación visual. Su precio a la fecha en Gctronic es 912 USD. Para comprarlo hay que encargarlo desde Suiza.

2.2.4. 3pi Robot

Pololu, la misma marca que tiene desarrollo de varios tipos de motores para robótica y PCB para controlarlos, diseña el 3pi Robot [5]. Es un robot bastante más económico que el e-puck, cuyo valor es 99.95USD ref. Sparkfun. También tiene dos ruedas para desplazarse de forma diferencial, 5 sensores de reflectancia, un LCD de 8x2 caracteres, un buzzer y tres botones para que el usuario pueda programarlos. Todos estos dispositivos están conectados a un microcontrolador ATmega328. Su velocidad es de 90 cm/s.

El 3pi fue diseñado especialmente como un robot seguidor de líneas y solucionador de laberintos. Existe varios videos que muestran la asombrosa velocidad de estos robots para solucionar un laberinto. Se programa en C, pero como posee un microcontrolador ATmega es posible hacer uso del bootloader Arduino y programarlo con ese IDE. Usa 4 baterías AAA y trae 4 LEDs.

Para el modelo que se planteó en este trabajo hace falta un módulo de comunicación inalámbrica, y el 3pi no tiene. Para usar esta alternativa es necesario sumarle a su precio 26USD, que es el precio del módulo XBee 1[w] serie 2 que se vende en Sparkfun.

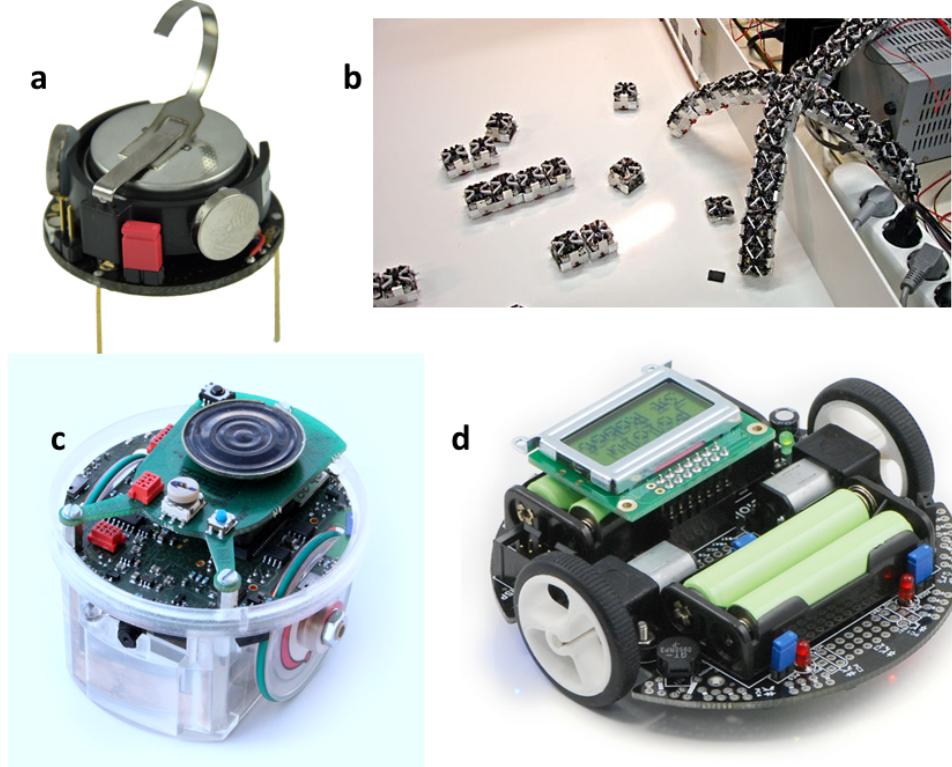


FIGURA 2.2: **a.** Kilobot, el más simple de todos y pequeño. Imagen tomada de [6]. **b.** Organismo Multibot, permite armar estructuras con los robots. Imagen tomada de [3]. **c.** e-puck, robot altamente difundido en investigaciones, caro y complejo. Imagen tomada de [4]. **d.** 3pi Robot, simple y económico pero no tiene comunicación inalámbrica. Imagen tomada de [5]

En la Figura 2.3 hay una tabla comparativa que resume las principales características de los robots: Kilobot¹, e-puck² y 3pi³.

¹<http://www.k-team.com/mobile-robotics-products/kilobot/specifications>

²<http://www.getronic.com/doc/index.php/E-Puck>

³<http://www.pololu.com/catalog/product/975/specs>

	Kilobot	E-puck	3pi
Procesador	ATmega 328 @ 8MHz	dsPIC 30 CPU @ 30 MHz	ATMega328
Memoria	32 KB Flash	114 KB Flash	32 KB Flash
Autonomía	3 meses (modo sleep)	3 horas	45 minutos
Cargador	opcional	opcional	opcional
Comunicación	Infrarrojo(IR)	Bluetooth	opcional
Sensores	Potencia recibida IR	Proximidad, cámara, otros	5 Reflectancia
Movimiento	2 motores vibración	2 motores diferencial	2 motores diferencial
Output	1 (RGB) LED	8 (RGB) LED + parlante	LCD 8x2 + buzzer
Diámetro	33 [mm]	70 [mm]	95[mm]
Alto	34[mm]	50 [mm]	32[mm]
Software	Básico	Complejo	Básico
Programación	WinAVR	MPLAB	Arduino
Precio USD	1.161 (10 pack)	900	100

FIGURA 2.3: En esta tabla se comparan las características principales de los robots: kilobot, e-puck y 3pi. No se incluyó el robot utilizado para el organismo multibot ya que no se comercializa.

2.3. Necesidades de mercado

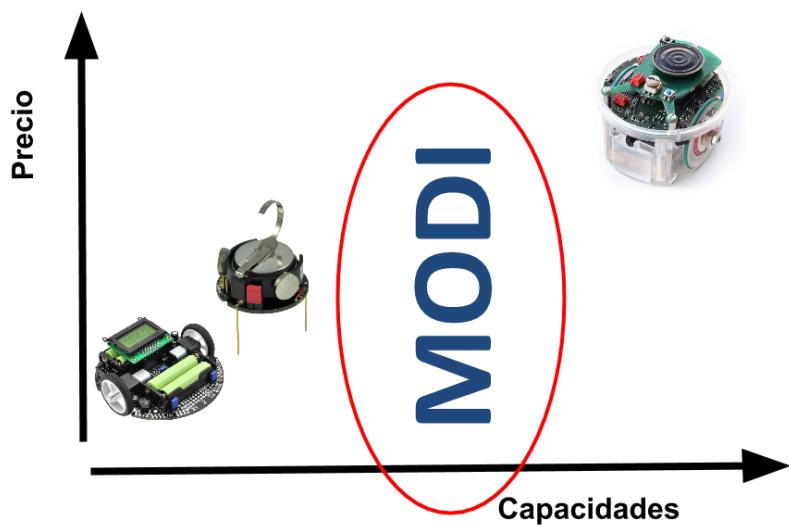


FIGURA 2.4: Nicho de mercado donde puede ingresar MODI

De los robots estudiados destaca en sus prestaciones el e-puck, pero este tiene dos grandes problemas para ser usado por gente que no es especialista en robots. Uno es que tiene demasiado hardware, lo que tiende a confundir y aumentar costos. Dos, que para su

comunicación inalámbrica hace uso de Bluetooth, protocolo que no soporta las redes Mesh para hacer de manera simple el control de muchos dispositivos en una red. Hace falta un robot que no sobrepase los 200 USD, capaz de controlarse en forma inalámbrica, simple de construir y fácil de usar. Se requiere hacer uso de tecnologías como impresoras 3D y diseño Open Hardware para formar el enjambre.

Capítulo 3

MODI

3.1. Diseño

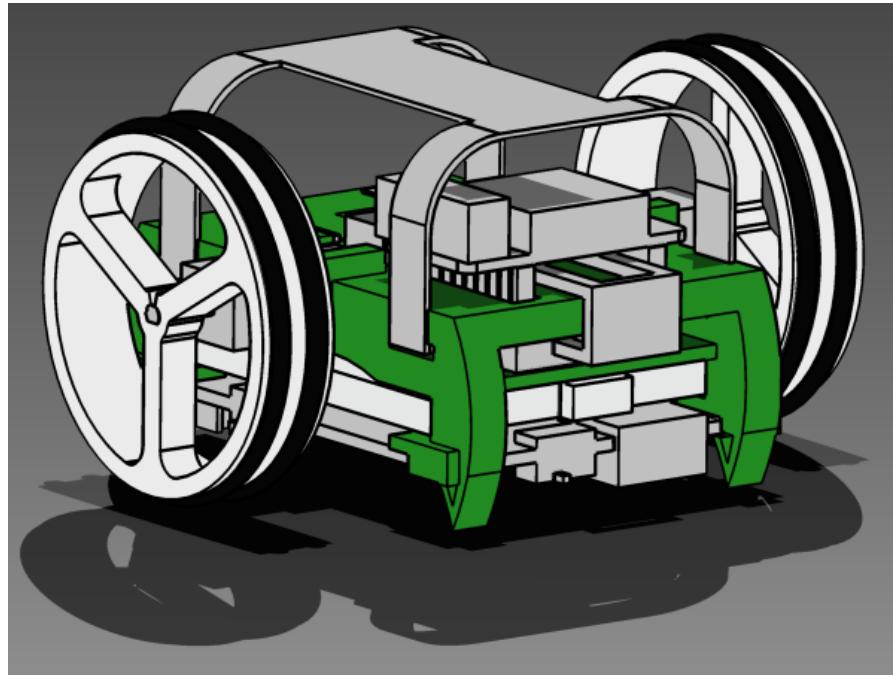


FIGURA 3.1: robot MODI (sigla para Modular Intelligence)

La función principal de MODI es ser una plataforma móvil de fácil acceso. Existe un repositorio en GitHub¹ donde se tienen los códigos actualizados para controlar y construir robots MODI. Una lista con componentes extras para comprar se encuentra en la Figura 3.18.

¹<https://github.com/FabLabUChile/modi>

Requerimientos de desarrollo

- Simple de fabricar
- Open Source
- Fabricable en un Fab Lab
- Componentes extras necesarios, de fácil acceso
- Minimizar la cantidad de componentes mecánicos extras necesarios
- Carga Autónoma con celda solar
- Seguimiento de grupo
- Control individual del color de cada MODI
- Movimiento simple de cada robot de forma independiente

3.2. Construcción

Construir un robot implica el diseño en computador de varios componentes. Estos pueden ser circuitos electrónicos (PCB), software de control y piezas mecánicas. Uno de los requerimientos principales del desarrollo es que sea simple de fabricar, es por esto que en el resultado final se reflejan las horas de diseño.

3.2.1. Fabricación Digital vs Análoga

Cuando se quiere pasar una idea al mundo real es necesario un proceso de fabricación. Dependiendo de la cantidad de herramientas que se tenga es más o menos fácil la tarea. Desde el comienzo hasta hace un par de años, quienes se dedican a construir robots, debían construir de manera *artesanal* donde es imposible que las piezas queden todas iguales y el tiempo empleado era bastante. Hoy en día existe una gran alternativa que surge como un nuevo paradigma, la Fabricación Digital. Las impresoras 3D, que no son más que un extrusor montado en un sistema con ejes que le dan 3 grados de libertad, permiten desde un modelo en un computador, obtener un objeto real hecho en plástico. También existe otro tipo de máquinas que permite hacer diseños en 2D, estas son las cortadoras LASER. La primera versión de MODI fue construido usando planchas de madera MDF y acrílico, cortados en LASER.

La primera versión se contruyó utilizando técnicas tradicionales, Figura 3.2 a y utilizando una impresora 3D Makerbot Replicator 1 se hizo la primera versión con piezas plásticas, Figura 3.2 b .

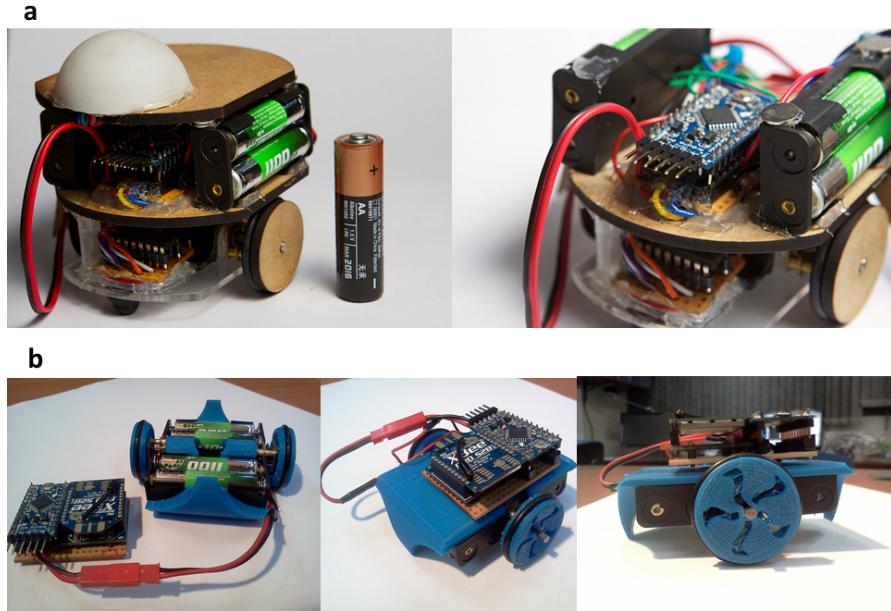


FIGURA 3.2: a. Primera versión MODI construido con MDF y bastante silicona, b. MODI usando técnica de *Fabricación Digital* con chasis de plástico construido con una MakerBot Replicator 1

Esta versión presentaba el problema de involucrar demasiadas partes que debían ser hechas por una persona. El microcontrolador, junto con la radio inalámbrica se colocaron en una placa electrónica para prototipado, más adelante se puede ver que fueron reemplazados por una PCB llamada Arduino FIO, que es una plataforma de desarrollo Arduino junto con un socket XBee. Otro factor clave para descartar esta versión es que utiliza 4 pilas AAA que necesitan ser removidas para poder ser recargadas, lo que impide que en futuras versiones exista la posibilidad de una carga autónoma por parte de los Robots. La versión actual de MODI permite su carga por medio de un puerto Mini USB, panel solar y de forma inalámbrica.

Aunque han bajado los precios de las máquinas para prototipado rápido, aún no están al alcance de todas las personas. Es por esto que existen los Fab Labs (acrónimo del inglés Fabrication Laboratory), que según Wikipedia es, *“un espacio de producción de objetos físicos a escala personal o local que agrupa máquinas controladas por ordenadores. Su particularidad reside en su tamaño y en su fuerte vinculación con la sociedad.”* Los Fab Labs están por todo el mundo, Figura 3.3. MODI fue concebido como un proyecto del Fab Lab de la Universidad de Chile y por esto es posible reproducirlo en cualquier Fab

Lab. En Chile, además del Fab Lab de la Universidad de Chile están: Design LAB UAI, Fab Lab Santiago y Stgo MakerSpace.



FIGURA 3.3: Mapa actual de lugares en el mundo que cuentan con un Fab Lab. En Chile a la fecha existen 3. Imagen tomada de fablabamersfoort.nl/fablabs/

3.2.2. Software CAD

CAD viene de sus siglas del Inglés, Computer-aided design, y se refiere a un diseño asistido por herramientas computacionales. Profesionales como ingenieros, arquitectos y del área del diseño por lo general son los que hacen más uso de estas herramientas.

Parte de la definición de Wikipedia es, “... se pueden dividir básicamente en programas de dibujo en dos dimensiones (2D) y modeladores en tres dimensiones (3D). Las herramientas de dibujo en 2D se basan en entidades geométricas vectoriales como puntos, líneas, arcos y polígonos, con las que se puede operar a través de una interfaz gráfica....”

Durante el transcurso del proyecto se trabajó con varios softwares CAD. El primero fué SketchUp 8 de Google, que permite fácilmente hacer bocetos de lugares y cuenta con una importante biblioteca de modelos para incluir en el diseño. Rápidamente se pudo hacer un sketch utilizando modelos descargados de Internet , Figura 3.16.

El diseño del chasis junto con las ruedas y demás partes plásticas, se hizo en un comienzo con SolidWorks 2012 y luego por ser más simple de usar, Inventor 2013 de Autodesk. Ambos softwares permiten generar modelos en 3D para luego exportar el diseño al

formato STL que es estándar para prototipar en plástico. Luego de varias iteraciones se lográ el modelo final, Figura 3.4 b, que tiene cinco piezas.

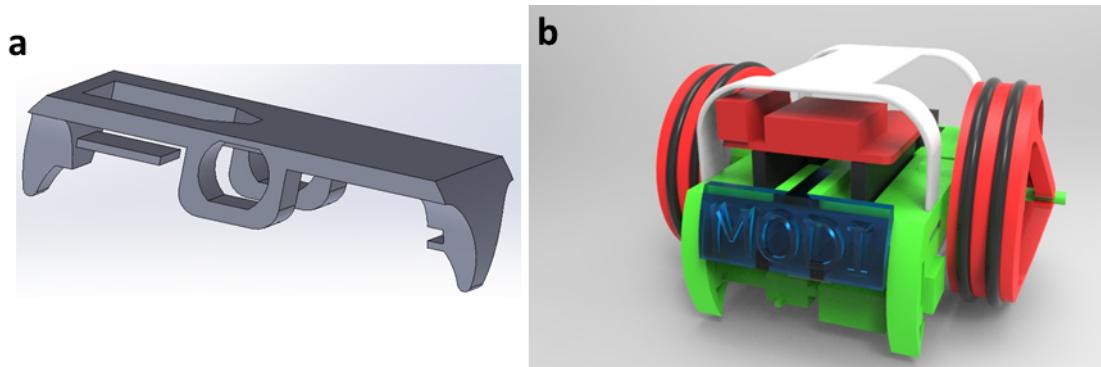


FIGURA 3.4: **a.** Primer Chasis de MODI, realizado con SolidWorks 2012. Esta versión sirve solo como prueba de concepto ya que no tiene espacio para las conexiones eléctricas necesarias. **b.** Última versión de MODI a la fecha, el diseño se realizó en Inventor 2013 y el render se obtuvo con KeyShot 4.

3.2.3. Impresora 3D

Parte importante de este trabajo se realizó con impresoras 3D. Estas existen desde los años 80' y hasta hace algunos años por su precio y tamaño eran de difícil acceso. Utilizan distintas tecnologías, las que se utilizaron en el proceso de este trabajo son las FDM, sigla del inglés "Fused deposition modeling" donde un cabezal extrusor de plástico se mueve en el plano xy depositando capa por capa de material mientras avanza en el eje z.

Primero se utilizó la 3D Touch Dual de la empresa Bit from Bytes, que permite hacer modelos de hasta 185 x 273 x 200[mm], con una buena resolución de 0.125[mm]² cada capa en eje z. Luego por facilidad de uso y menor tiempo de impresión se usa una impresora MakerBot Replicator 1

²teambastech.com/Store/index.php?route=product/product&product_id=205

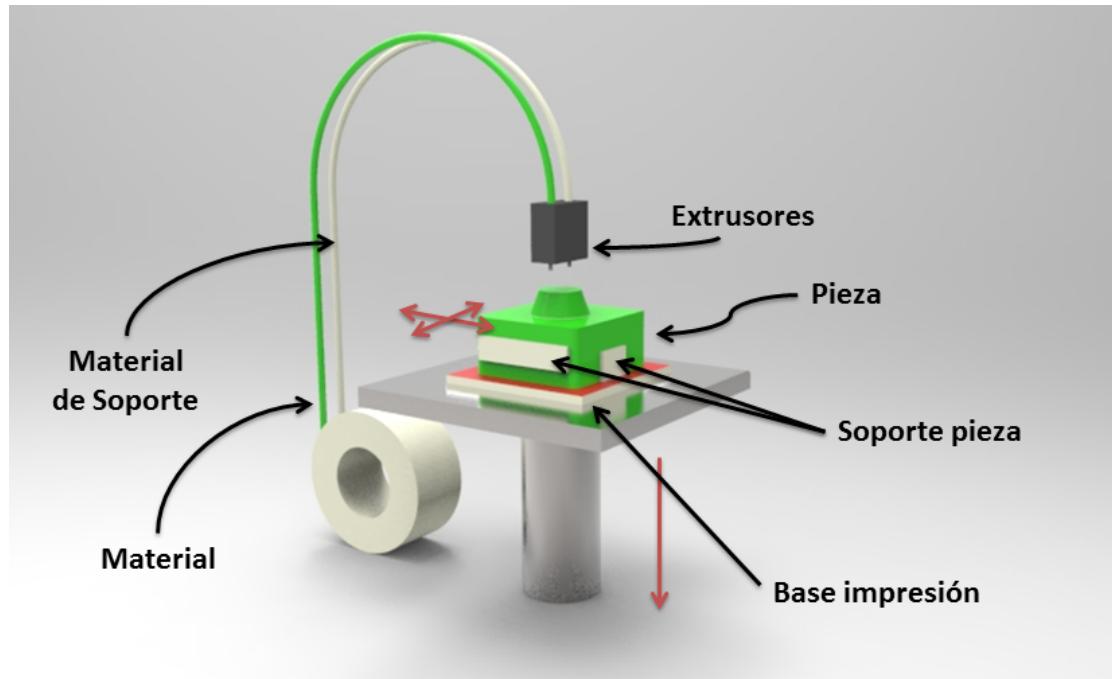


FIGURA 3.5: Modelo de funcionamiento impresoras 3D MDF. Tiene dos materiales, uno para construir la pieza y otro que funciona como soporte estructural para la impresión, además existe una base que se calienta para sujetar la pieza mientras se construye.

	Replicator 1	3D Touch	Uprint
Volumen impresión	5 m ³	10 m ³	4.7 m ³
Espesor capa	0.3 mm	0.125mm	0.254 mm
Velocidad	40 mm/s	6 mm/s	lenta
Precisión	2.5 µm Z , 11µm XY	0.1mm Z 0.2mm XY	~
Material soporte	con herramienta	con herramienta	Necesita proceso extra
Precio insumo	60 USD/Kg	100 USD/Kg	400 USD/Kg
Precio impresora	1,750 USD	3,999 USD	2,799 USD

FIGURA 3.6: Comparación de las características mas importantes de impresoras 3D, Replicator 1, 3D Touch y Uprint

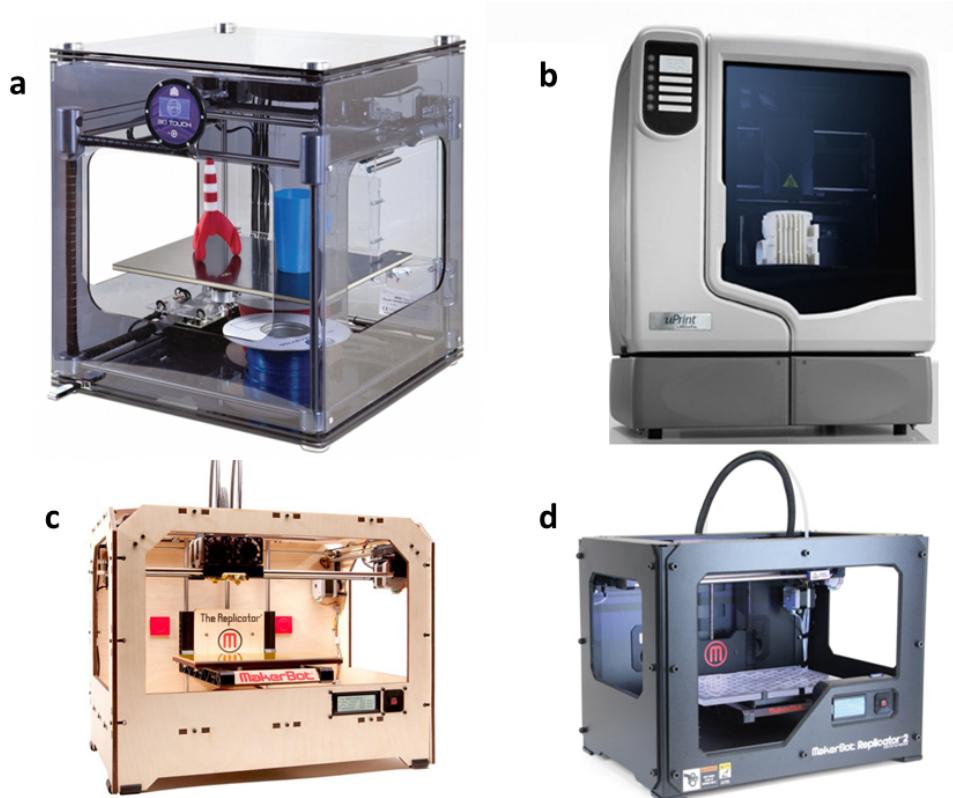


FIGURA 3.7: **a.** 3D Touch Dual de la empresa Bit from Bytes, **b.** Uprint SE de Stratasys, **c.** Replicator 1 de Makerbot, impresora 3D FDM utilizada para prototipar chasis y ruedas de robot MODI, **d.** Replicator 2 de MAkerbot, tiene mejor resolución que la versión 1.

3.2.4. Diseño PCB

En los primeros prototipos un problema que se repetía constantemente son las conexiones entre los sistemas que componen al robot. Son 15 conexiones que unen distintos componentes, Figura 3.8. Por esto se requiere un PCB que permita reducir al mínimo el espacio necesario para energizar y comunicar los componentes. El requerimiento principal es poder fabricarlo con una fresa CNC como la Roland Modela MDX-20, por lo que para facilitar el proceso de construcción es necesario diseñar un PCB de una sola capa. Se utilizó Eagle como software para el diseño del diagrama eléctrico y PCB, el resultado se puede ver en la Figura 3.9.

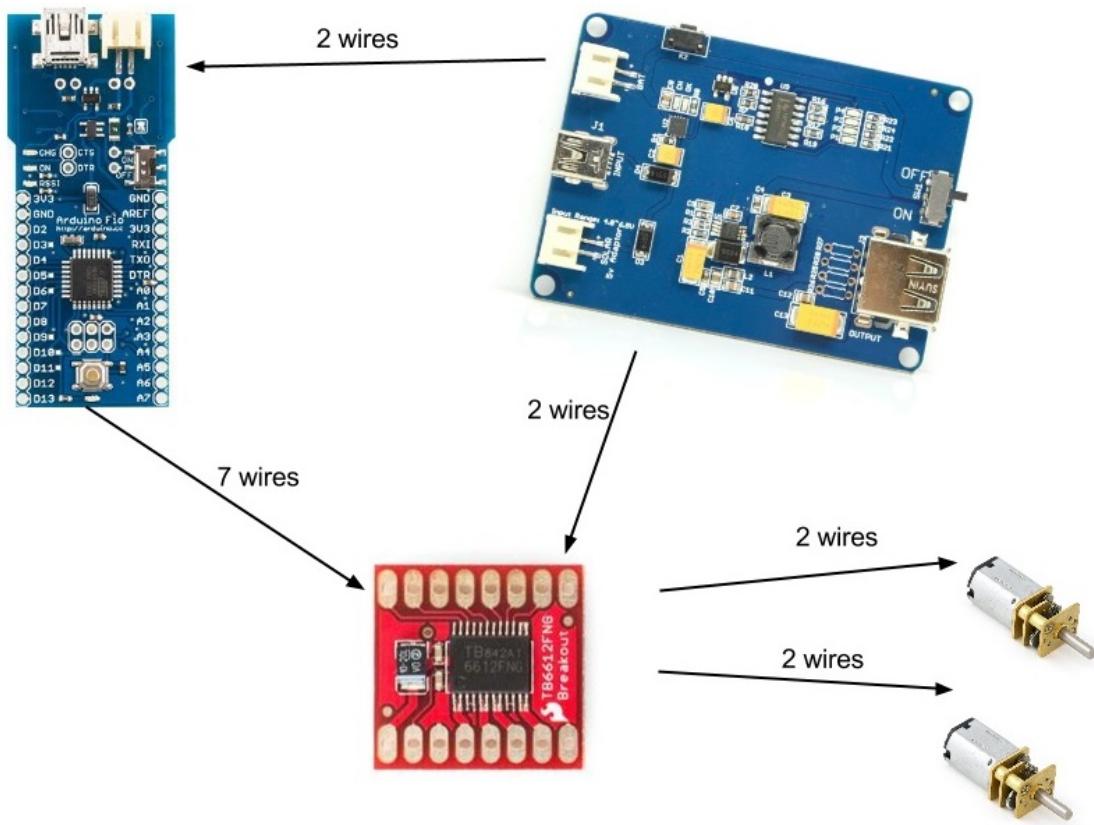


FIGURA 3.8: Son 15 conexiones que unen distintos los componentes.

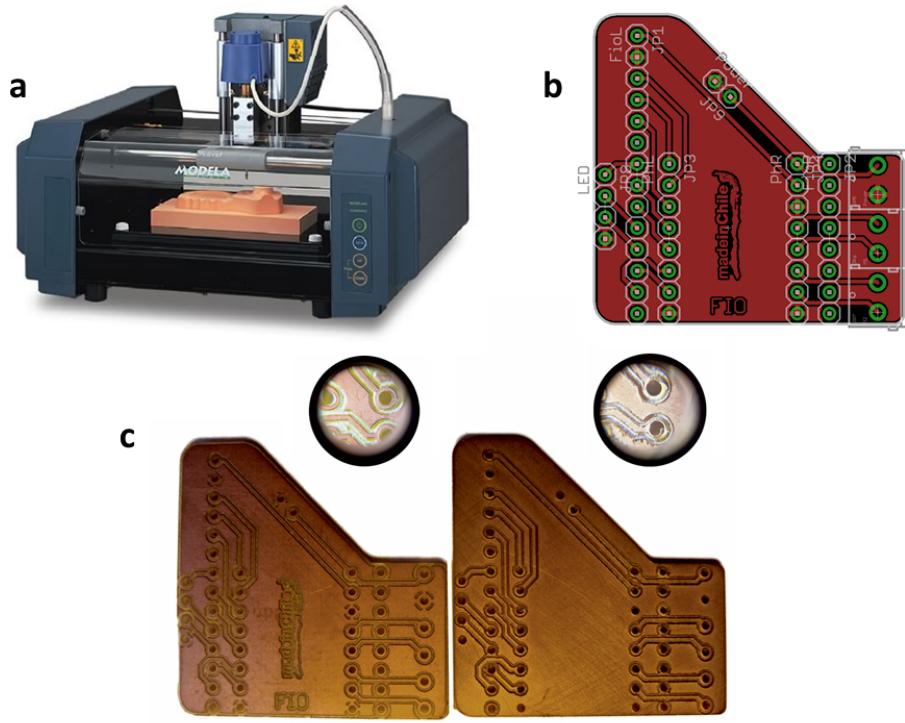


FIGURA 3.9: a. Roland Modelo MDX-20 usada para hacer PCB y modelos 3D b. MODIBoard es una PCB diseñada para poder conectar de forma simple los diversos componentes electrónicos. Conecta: Arduino Fio, Lipo Rider Pro, Motor Driver, LED RGB y Motores, c. PCB fresada en dos maquinas diferentes. Se puede apreciar en el detalle aumentado 60x, la importancia de contar con una buena resolución de fresado.

En especial es necesario tener un buen pad para facilitar el soldado.

3.3. Componentes electrónicos

MODI está compuesto por varios componentes electrónicos que implementan tecnologías nuevas. De forma sencilla el usuario puede hacer uso de todas estas, que por ser Open Source tienen una amplia documentación. Su uso en colegios, talleres y universidades, por ser de bajo costo, y sin partes externas extremadamente frágiles, es una excelente herramienta para iniciar a alumnos en temas como la Programación, Robótica, Comunicación Inalámbrica y Visión Artificial, entre otros, sin los miedos asociados a utilizar un equipo costoso y delicado.

El robot MODI es pequeño y de fácil construcción con las herramientas de Fabricación Digital que se puede encontrar en un FabLab. Tiene un pequeño microcontrolador **Arduino** que controla dos motores, y un **XBee** para la comunicación inalámbrica con un computador central que usando cualquier lenguaje de programación con comunicación serial puede controlar uno o más robots. La posición de cada robot se obtiene con tags **Fidenciales**, que permiten por medio de una cámara cenital saber la coordenada y rotación de cada uno. En esta primera aplicación es necesario contar con autonomía

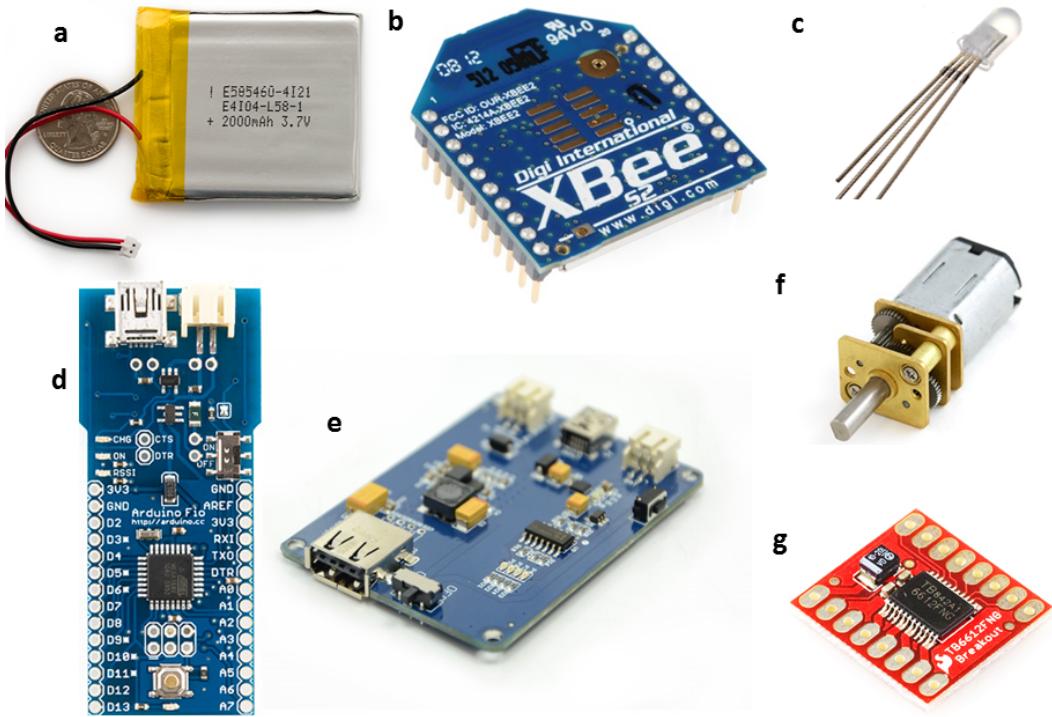


FIGURA 3.10: Todos los componentes electrónicos que forman MODI. a. Batería LIPO 2,000 mAh, b. XBee, c. LED Rgb difuso, de esta manera se obtiene un ángulo de visión mucho más amplio, d. Arduino Fio, e. Lipo Rider Pro, f. Motor DC 30:1 g. Motor Driver 1A Dual TB6612FNG.

energética por lo que se incluye una batería lipo de 2000[mAh] junto a un módulo que carga la batería y eleva el voltaje de salida a 5[v]. Este el módulo de carga, **Lipo Rider Pro**, tiene además la ventaja de permitir distintos tipos de carga, como una celda de carga solar o un sistema de carga inalámbrica por medio de bobinas. Tener la posibilidad de cargar una batería con luz solar o por medio de forma inalámbrica hace posible el hacer experimentos que necesiten estar funcionando semanas sin la intervención de un humano para cargar cada uno.

3.3.1. Arduino FIO

El Arduino FIO, Figura 3.10 d, ha sido diseñado por Shigeru Kobayashi, es una placa para microcontrolador basada en el ATmega328P. Funciona a 3.3V y 8 MHz. Tiene 14 pines de I/O digitales (de los cuales 6 pueden usarse como salidas PWM), 8 entradas analógicas, un oscilador en placa, un botón de reinicio (reset), y agujeros para montar conectores de pines. Tiene conexiones para una batería de polímero de Litio e incluye un

circuito de carga a través de USB. En el reverso de la placa tiene disponible un zócalo para módulos XBee.

El Arduino FIO está diseñado para aplicaciones inalámbricas. El usuario puede subir sus sketches con un cable FTDI o una placa adicional adaptadora Sparkfun. Además, si utiliza un adaptador de USB a XBee modificado (como el USB Explorador de XBee), el usuario puede subir sketches de forma inalámbrica. La tarjeta viene sin conectores pre-montados, permitiendo el uso de diversos tipos de conectores o la soldadura directa de los cables. La Figura ?? tiene una tabla que resume sus características principales.

3.3.2. Motores DC

Este motor Pololu, Figura 3.10 f es muy pequeño (largo: 9.27 mm), por lo mismo usa muy poca corriente y tiene una caja de reducción metálica de 30:1. El eje tiene forma de D.

Pese a no ser la alternativa más económica, estos motores han sido muy difundidos para su uso en robótica por su bajo consumo y alto torque. La Figura 3.11 tiene una tabla que resume sus características principales.

3.3.3. XBee

Los dispositivos con los cuales interactuamos a diario poseen acceso a distintos tipos de sistemas de comunicación. Los *Smartphone* cuentan con diversas antenas para conectarse a WIFI, Bluetooth, red celular, por nombrar algunas. Estamos en un momento en que las comunicaciones son una de las claves tecnológicas que nos permiten cumplir con tareas que para nuestros abuelos eran imposibles de hacer en un día. Es claro que un sistema con varios o cientos de robots también necesitan comunicarse. Y en el caso particular de MODI, que por el momento no tiene sensores, es fundamental la información que se transmite para controlarlo.

Para comunicar un robot con otro pueden usarse diversas técnicas. Dependiendo del área en particular que sea del interés del científico, los robot pueden comunicarse por medio de sensores de proximidad o tacto, para saber que hay otro cerca, pueden tener un sistema de visión artificial con una cámara a bordo que le permita “ver” el entorno al robot, con parlantes y micrófonos pueden emitir algún ruido que sea interpretado por su entorno, o lo mismo con sensores de otros tipos.

El robot que se construyó para hacer las pruebas cuenta con un sistema externo de visión artificial que permite obtener posición y orientación de cada robot. Esto es necesario ya

que como se mencionó, no existe ningún sensor interno del robot que le pueda ayudar a ubicarse en el espacio. Esta información es transmitida a cada robot por medio del protocolo *zigbee* implementado en los chip XBee de la empresa Digi, Figura 3.10 b. Esto permite comunicar un alto número de robots, con muy poco consumo energético y de manera relativamente simple. Se escogió usar visión artificial sobre el sistema para bajar los costos individuales de los robots y XBee por ser el estándar en la industria en redes de sensores inalámbricos. Cabe destacar que los robots que existen en el mercado no incluyen XBee, y los que lo traen lo hacen como accesorio.

3.3.4. Lipo Rider Pro

El LiPo Rider Pro, Figura 3.10 a, es una versión mejorada del LiPo Rider, que es una fuente de poder para alimentar distintos gadget con energía verde. Esta placa permite alimentar con 5V distintos dispositivos. Puede obtener energía del sol o por medio de inducción magnética. También puede cargar un SmartPhone.

La Lipo Rider carga una batería de Litio Polímero. Posee una alta densidad de carga, 2000mAh a 3.7V. Figura 3.10 a.

3.3.5. Motor Driver 1A Dual TB6612FNG

El TB6612FNG motor driver puede controlar hasta dos motores DC, a una corriente constante de 1.2A (3.2A peak). Dos señales de entrada (IN1 y IN2) puede ser usadas para controlar el motor en una de cuatro modos de funcionamiento - CW, CCW, short-brake, y stop. Cada una de las salidas hacia los motores (A y B) se puede controlar separadamente, la velocidad de cada motor es controlada vía PWM con una frecuencia de hasta 100khz. El pin STBY debe colocarse en High para que el motor salga del estado Standby. Figura 3.10 g.

Arduino FIO	
Microcontrolador	ATmega328P
Voltaje operación	3.3 V
Input Voltaje	3.35 - 12 V
Input Voltaje carga batería	3.7 - 7 V
Pins Digital I/O	14 (6 salidas PWM)
Input Pins Análogos	8
Corriente DC en cada Pin I/O	40 mA
Memoria Flash	32 KB (2KB Bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock	8 MHz
Motores DC Pololu	
Velocidad	440 RPM @6 V, 40 mA
Torque	3 Kg-mm
Reducción	30 : 1
Consumo trancado	0.36 A
XBee	
Voltaje operación	3.3V @ 295 mA
Tasa de datos máximo	250kbps
Potencia output	50mW (+17dBm)
Rango distancia	1,600m
Tipo antena	Interna
Certificación	FCC
ADC	4 pins de entrada de 10-bit
Digital I/O	10 pins
Seguridad	Encriptación en 128 bits
Set comandos	AT y API
LED RGB	
Voltaje operación	(RGB):(2.0, 3.2, 3.2)V
Corriente máxima	(RGB):(20,20,20)mA
Luminocidad	(RGB):(2,800, 6,500, 1,200)
Motor Driver 1A Dual TB6612FNG	
Voltaje operación	VM=15V max, VCC=2.7-5.5V
Corriente máxima	Iout=1.2A(average) / 3.2A (peak)
Señal control	Standby para bajo consumo
Seguridad	Built-in thermal shutdown circuit y low voltage detecting circuit
Batería LiPo	
Voltaje operación	3.7 V
Capacidad	2000 mAh
Conector	2-pin JST-PH - 2mm
Lipo Rider Pro	
Voltaje operación	5 V @ 1 A load output
Inputs	Puerto USB, panel solar y fuente inalámbrica
Carga	Algoritmo de carga/descarga built en el chip
Tipo bateria	Lithium Polymer
Indicadores	4 LEDs de nivel batería

FIGURA 3.11: Resumen de las características más importantes de los componentes electrónicos que tiene MODI.

3.4. Componentes mecánicos

Como se comentó en la sección [3.2.3](#), las piezas del robot fueron diseñadas para ser construidas por una maquina de prototipado rápido del tipo FDM y se tener en cuenta la tolerancia mecánica de las piezas. En especial hay dos piezas: la rueda y el chasis, que se unen al motor. Esta unión debe quedar con un ajuste mecánico Forzado Medio ³, para que estas uniones queden fijas y las ruedas esten en el eje correspondiente.

Está formado por cuatro partes:

- Chasis, debe cumplir con la tarea de contener toda la electrónica y motores
- Ruedas, permiten el desplazamiento y poseen dos canales para poner o-ring y así aumentar el roce con la superficie.
- Accesorio, tiene un sistema de anclaje al chasis que permite tener distintas carcasa y sensores.
- Logo, sujeta parte de la electrónica contenida en el Chasis y un LED RGB.

La pieza que más demora en construirse es el Chasis, que son 108 minutos en una Replicator 1 de MakerBot.

³<http://campusurico.ualca.cl/~fespinoz/Ajustes%20y%20tolerancias%20mecanicas.pdf>

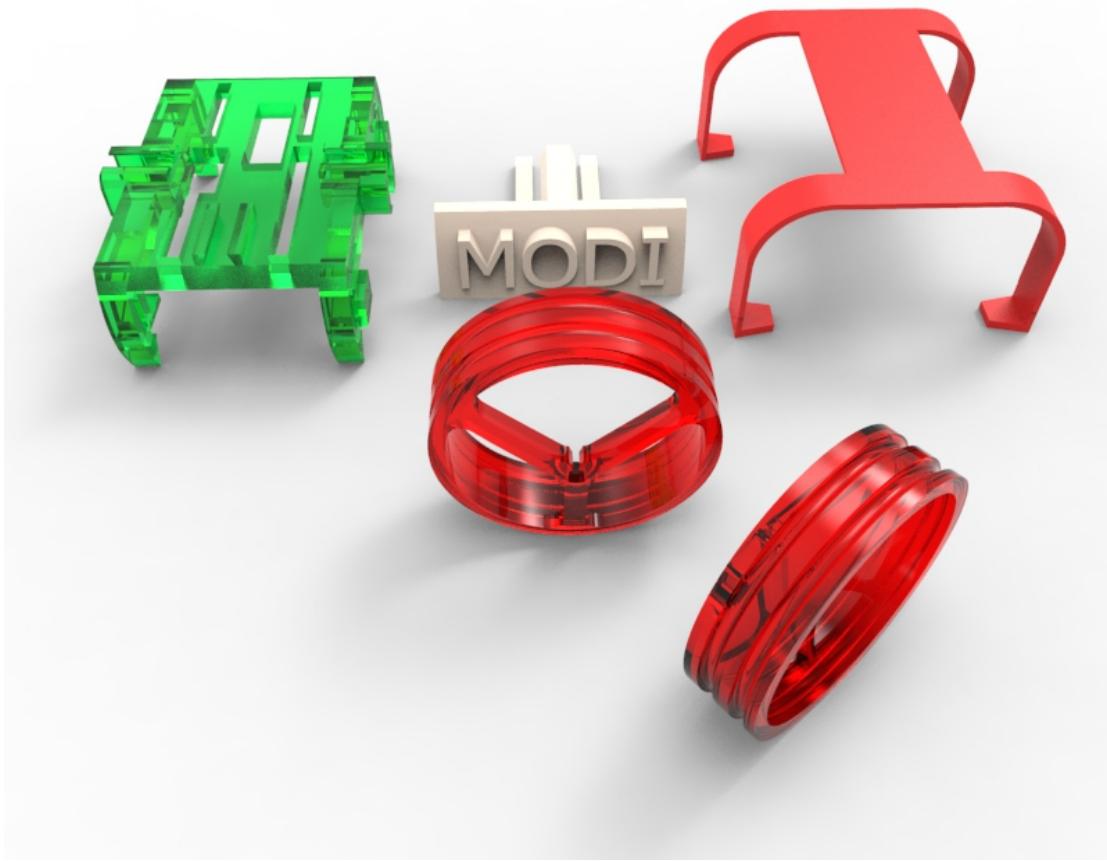


FIGURA 3.12: Las cinco piezas que componen el hardware mecánico de MODI, estas son: el chasis, las dos ruedas, el accesorio y logo.

3.5. Locomoción

Existen distintas áreas donde se utilizan robots, dependiendo de la tarea a realizar es la movilidad que este debe tener. Algunos tienen complejos sistemas como la plataforma del proyecto Atlas (The Agile Anthropomorphic Robot) de Boston Dynamics, Figura 3.13. En nuestro caso, solo es necesario desplazarse sobre un superficie plana y por esto, en vez de tener costosos actuadores neumáticos, la solución más simple es contar con dos motores para poder tener un movimiento diferencial. Si se desea tener más información sobre los tipos de locomoción en robots con ruedas, ver esta pagina web⁴.

Para moverse MODI tiene dos motores DC ???. Estos tienen una caja de reducción 30:1 que permiten aumentar el torque y disminuir la velocidad. La Figura 3.14 muestra la característica de estos.

⁴en.wikibooks.org/wiki/Robotics/Types_of_Robots/Wheeled

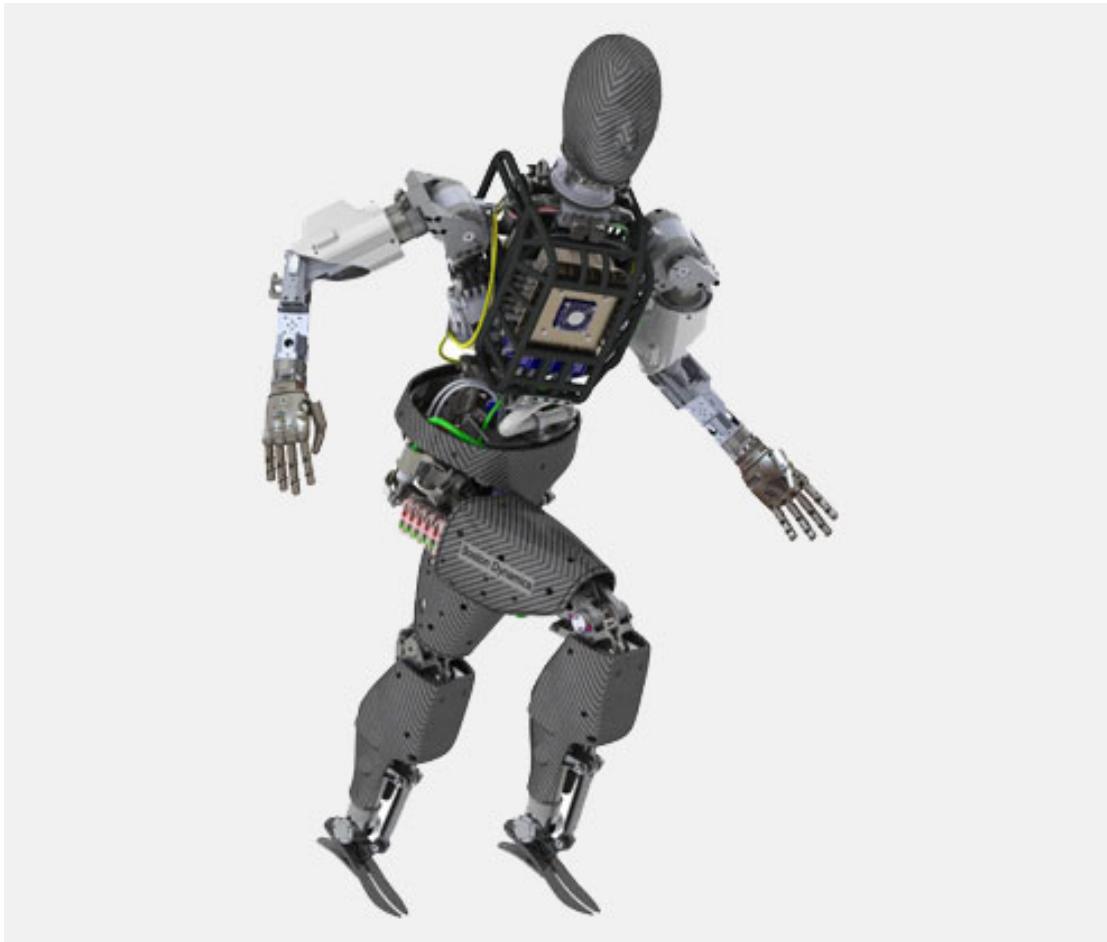


FIGURA 3.13: Atlas es un humanoide con gran movilidad, diseñado para poder usar herramientas humanas. Posee 28 grados de libertad con actuadores hidráulicos y la energía se le entrega por medio de un cable flexible. Imagen tomada de bostondynamics.com.

El control de velocidad de los motores se hace con PWM, que es una señal periódica a la cual se le modifica el ciclo de trabajo. El sentido de giro es controlado con el chip TB6612FNG [3.14](#) a. Los valores que genera el Arduino para control de los motores van desde -255 a 255, y para los distintos casos se puede ver la Figura [3.15](#) que indica los 3 movimientos que son: avanzar, avanzar hacia a la izquierda y girar en el eje. Aquí los numeros representan el periodo del ciclo de trabajo de la señal enviada por el Arduino al controlador de los motores. Se puede realizar un control mas fino de la posición usando la información de posicion del robot en un PID, se recomienda ver este video tutorial [5](#). Para más información sobre PWM y este chip ver esta pagina web [6](#).

⁵<http://www.youtube.com/watch?v=aE7RQNhwnPQ>

⁶<http://www.pololu.com/docs/0J21/5.c>

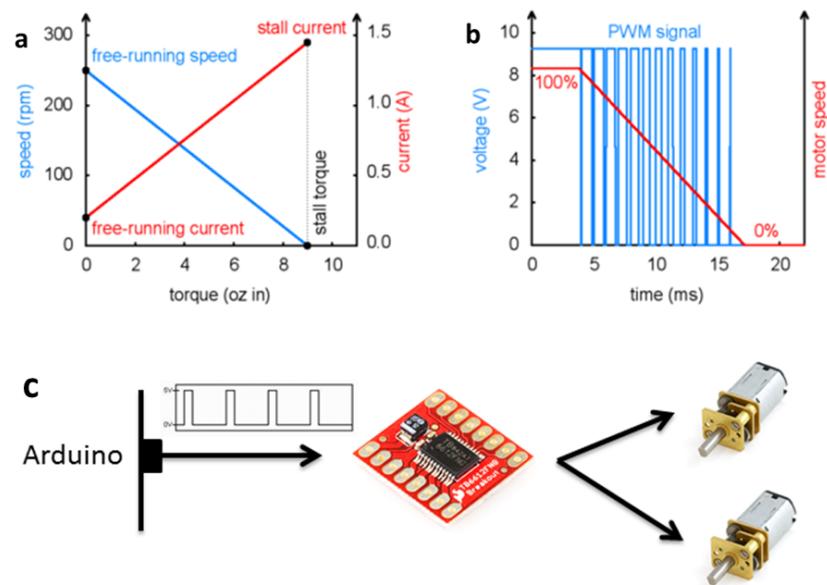


FIGURA 3.14: a. Operación motores: Corriente, velocidad y torque. Imagen extraída de pololu.com, b. Control de velocidad PWM, se observa como disminuye la velocidad según el ancho del pulso pwm.

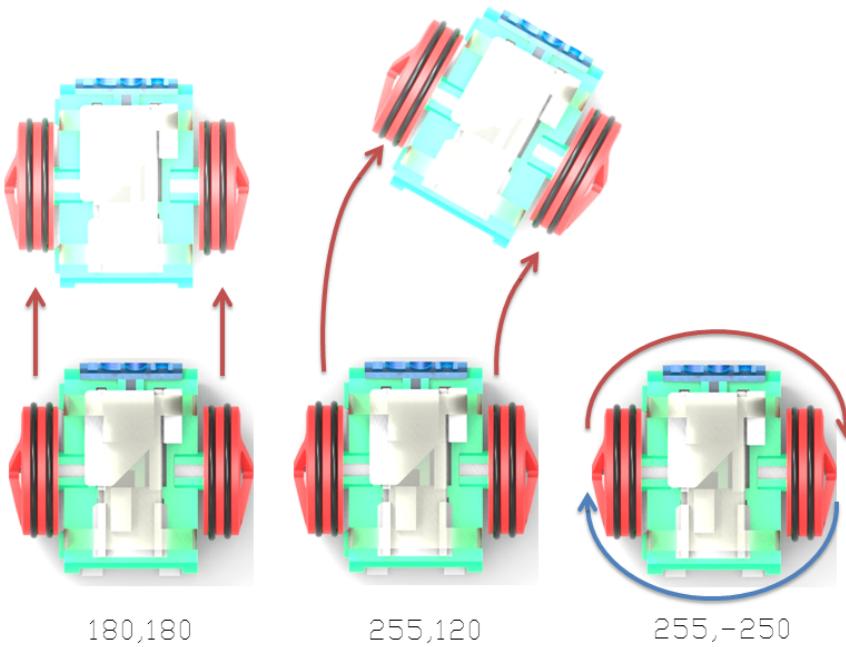


FIGURA 3.15: Movimiento diferencial. A diferencia de un auto, MODI tiene dos motores que giran de forma independiente. Las distintas velocidades y sentido de giro determinan el angulo para girar. Si se necesita solamente rotar los motores deben girar en sentidos opuestos.

3.6. Implementación

La construcción de MODI fue motivada por una aplicación concreta, que es tener un grupo de robots controlables de forma independiente para hacer estudios del comportamiento del grupo. Para esto fue necesario montar un sistema simple con una cámara cenital que vigila la posición y orientación de cada uno.

3.6.1. Setup

Para investigar el comportamiento colectivo de un grupo de robots, en el mundo real sin hacer uso de simuladores, es necesario contar con un lugar físico donde poder activar los robots. Además para simplificar cada uno de los robots, estos no tienen sensores internos por lo que hay una cámara montada sobre el plano de movimiento de estos, para hacer Seguimiento Visual. Esta información de coordenadas llega a un computador servidor que le indica a cada robot como moverse.

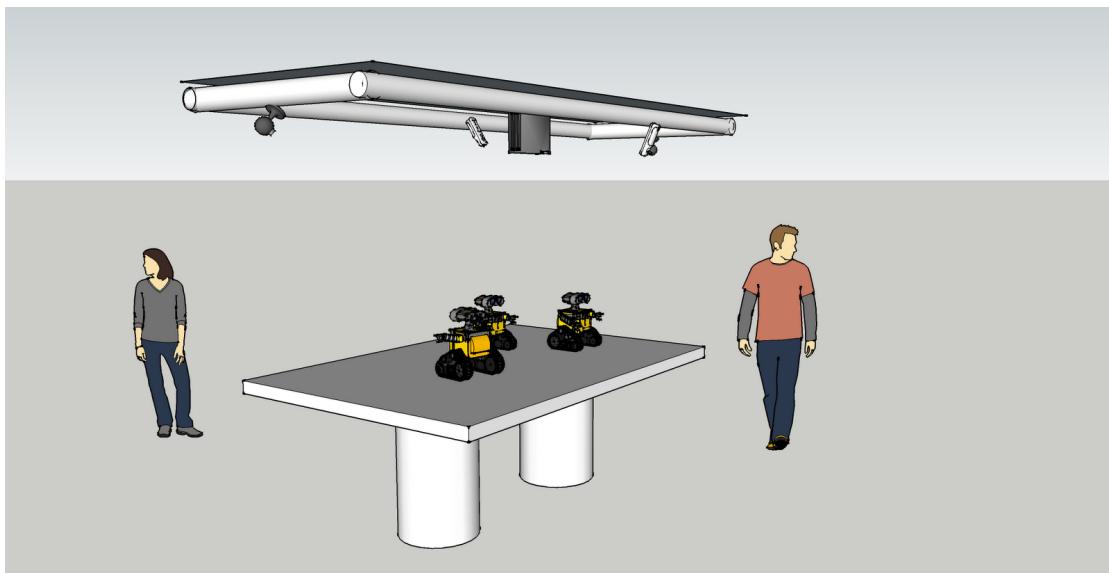


FIGURA 3.16: Setup a montarse para hacer estudios de grupos de robots.

3.6.2. Software

Software, el cerebro del robot. Gracias a un puerto serial escuchando las instrucciones para mover los motores del robot, es bastante sencillo controlar a MODI desde el computador usando cualquier lenguaje que tenga una biblioteca para uso de puertos seriales.

A continuación algunos ejemplos de como controlar a MODI. El puerto serie en MODI tiene configurada una velocidad de 115200 y esta esperando los caracteres ‘w’, ‘a’, ‘d’ y ‘s’, como Adelante, Izquierda, Derecha y Atrás respectivamente. El código de MODI, el Firmware, se encuentra en el Git de MODI en Github.

3.6.3. Tracking 2D

Al igual que los seres vivos un robot necesita *sentidos* o algún sistema sensorial para poder interactuar con su entorno. Estos pueden ser sensores IR para detectar objetos cercanos, cámaras o LASER para hacer un mapa del entorno o simplemente un pulsador que sea presionado cada vez que el robot colisiona. Para bajar costos, MODI actualmente no cuenta con sensores “onBoard”. Cada robot es comandado por un sistema que le dice donde donde está él, dónde están los demás y los límites del área de trabajo. Se diseñó de esta manera para simplificar la programación de cada robot.

reactTIVision [7] es un framework open source, cross-platform, para realizar tracking rápido y robusto de tags fiduciales pegadas a objetos físicos. Ha sido diseñado como un toolkit para el desarrollo rápido multi-touch interactive surfaces. Este Framework fue desarrollado por Martin Kaltenbrunner y Ross Bencina en el Music Technology Group de la universidad de Pompeu Fabra en Barcelona, España. reactTIVision fue diseñado como sensor principal de la Reactable, un sintetizador modular tangible que ha marcado los standards en lo que a aplicaciones multitouch se refiere. Esta alternativa debe ir acompañada de otros elementos, como la cámara y el computador en el cual se ejecuta el software.

3.6.4. Bill Of Material

Con un precio total de 180 USD, figura 3.18, MODI se propone como una alternativa económica para construir un robot. Pensando en comprar desde Chile, son tres las tiendas en que se cotizaron los componentes. Es posible comprar todo desde internet. Para más detalles sobre la Lista de materiales ver ⁷

⁷<http://kitbom.com/otrab/modi>

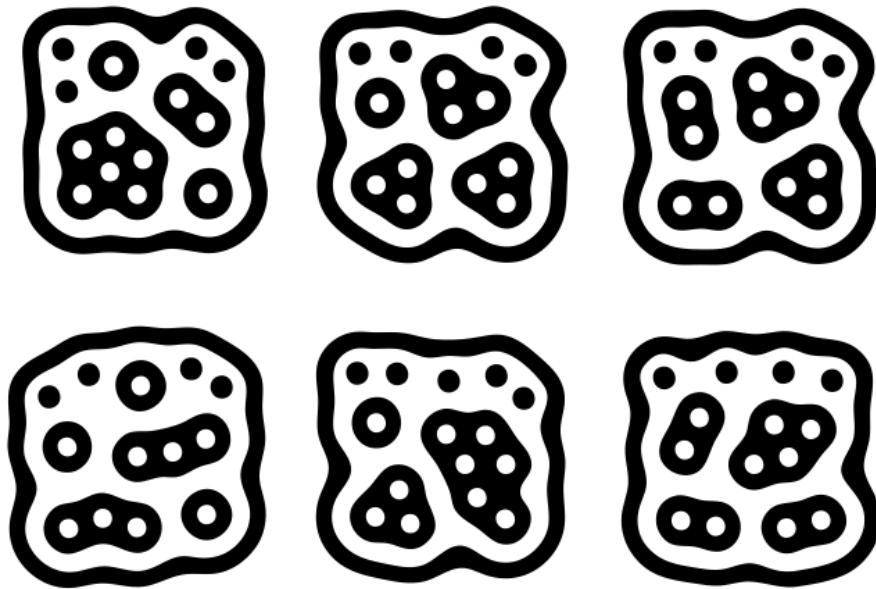


FIGURA 3.17: Códigos fiduciales utilizados en reacTIVision. Estos códigos aunque parecen algo extraños para los seres humanos están diseñados de tal forma que es muy fácil diferenciar unos de otros y además obtener su orientación y posición en un software de análisis de imágenes. Se tiene un total de 216 configuraciones posibles de códigos.

Bill of Materials		Build View	Purchase View	
Part (# Needed)		Recommended Purchase (# to Buy)		Total Price
Olimex				
Arduino Fio (1)		Olimex MCI-TDD-01292 (1)		\$29.73 (1 @ \$29.73)
Arduino Stackable Header - 8 Pines (1)		Olimex MCI-PRT-00671 (1)		\$0.51 (1 @ \$0.51)
Terminales de tornillos de 3.5mm (2 Pines) (3)		Olimex MCI-PRT-00781 (3)		\$1.95 (3 @ \$0.65)
Female Header 1x8 Pines - altura 8.5mm (Arduino) (2)		Olimex MCI-PRT-01118 (2)		\$1.40 (2 @ \$0.70)
Female Header 1x6 Pines - altura 8.5mm (Arduino) (1)		Olimex MCI-PRT-01117 (1)		\$0.58 (1 @ \$0.58)
Pin Header 40 pines (1)		Olimex MCI-PRT-00177 (1)		\$0.43 (1 @ \$0.43)
2 Pin .100" Header Connector - Single Row (1)		Olimex MCI-PRT-00451 (1)		\$0.25 (1 @ \$0.25)
Crimp Pin for Header Connector (2)		Olimex MCI-PRT-00453 (2)		\$0.20 (2 @ \$0.10)
Micro Metal Gearmotor 30:1 (2)		Olimex MCI-RBT-00548 (2)		\$46.56 (2 @ \$23.28)
XBee 2mW PCB Antenna - Series 2 (ZigBee Mesh) (1)		Olimex MCI-WIR-01109 (1)		\$41.12 (1 @ \$41.12)
Bateria de Lithium Ion - 2000mAh (1)		Olimex MCI-PRT-00146 (1)		\$24.00 (1 @ \$24.00)
LiPo Rider Pro (1)		Olimex MCI-PRT-01126 (1)		\$21.97 (1 @ \$21.97)
SparkFun Electronics				
Motor Driver 1A Dual TB6612FNG (1)		SparkFun Electronics ROB-09457 (1)		\$8.95 (1 @ \$8.95)
LED - RGB Diffused Common Cathode (1)		SparkFun Electronics COM-09264 (1)		\$1.95 (1 @ \$1.95)
Victronics				
RES 100R 1/8W 5% (2)		Victronics 55-80101_ANY (2)		\$0.06 (2 @ \$0.03)
RES 180R 1/8W 5% (1)		Victronics 55-80181_ANY (1)		\$0.03 (1 @ \$0.03)
				TOTAL PROJECT COST: \$179.69

FIGURA 3.18: Lista de materiales a comprar para construir un robot MODI, la versión online de este BOM se puede ver en <http://kitbom.com/otrab/modi>

Capítulo 4

Aplicaciones

MODI en esencia es un Arduino con ruedas y control inalámbrico. Esto significa que para programarlo y hacer uso de él en un proyecto la curva de aprendizaje es rápida. Existen un sin fin de tutoriales donde se puede aprender lo básico para programar un Arduino y en un día ya se puede controlar de forma simple el movimiento del robot. Es por esto que MODI es indicado para usarse por cualquier persona que desee experimentar con robots y no tenga los conocimientos técnicos para construir uno desde cero. En las secciones siguientes se describen algunas aplicaciones posibles.

4.1. Auto modelamiento

Uno de los fines de MODI es poder ser utilizado para investigar el Auto modelamiento en robots, que permite a un robot sintetizar un comportamiento en base a la exploración de los posibles modelos de el mismo. La importancia de esta área radica en que actualmente los robots más difundidos hacen uso de un software llamado controlador, que por ejemplo para mover un robot del punto a al punto b controla las señales de cada moto, en base a como se encuentran ubicados en el espacio los motores, pero si un motor falla o si el robot en vez de tener ruedas tiene patas, no logra adaptarse. Juan Cristobal Zagal junto a Hod Lipson [8] exploraron el comportamiento de un robot capaz de entrar en un proceso autoreflexivo. Ellos estudiaron a un robot al cual se le programaron dos controladores, uno primitivo (primitive controller) y uno reflexivo (reflective controller) que puede observar al primero.

El controlador reflexivo es capaz de determinar el control primitivo sin tener acceso directo a sus estados internos, haciendo uso de ingeniería inversa al leer los input/output de éste. Acá se implementa como comportamiento que el robot (e-puck) debe alejarse

de luces rojas y acercarse a azules. Se logra con una exploración mínima posible de hardware.

El diagrama del algoritmo que controla todo el proceso que se implementa como comportamiento para recuperarse de fallas puede verse en el anexo.

J. Bongard., V. Zykov y H. Lipson., logran que un robot se recupere de una falla inesperada de forma autónoma. Dicen que el robot puede recuperarse de manera autónoma haciendo uso de su (propio al robot) Self-modeling. Concretamente implementan sus algoritmos en un robot de 4 patas que utiliza una relación entre sus sensores y motores para indirectamente inferir su propia estructura.

Si se remueve una extremidad, el robot es capaz de generar una nueva forma de caminata que le permita cumplir con su misión que es avanzar. En la figura 4.1, se describe un esquema del algoritmo. El robot realiza una acción física (A), al azar, luego se ejecuta la mejor acción que se encuentre en (C). A continuación genera varios auto-modelos que coincidan con las lecturas de los sensores obtenidas en (B). Aún no sabe cual es el modelo, por lo que en (C) genera varias acciones posibles que acotan la búsqueda de modelos. Despues de varios ciclos de (A) a (C), el modelo obtenido se utiliza en (E) para generar la secuencia de locomoción. La mejor secuencia de locomoción es probada físicamente en el robot. Se refina el modelo volviendo al paso (B) y en (D) pueden crearse nuevos comportamientos.

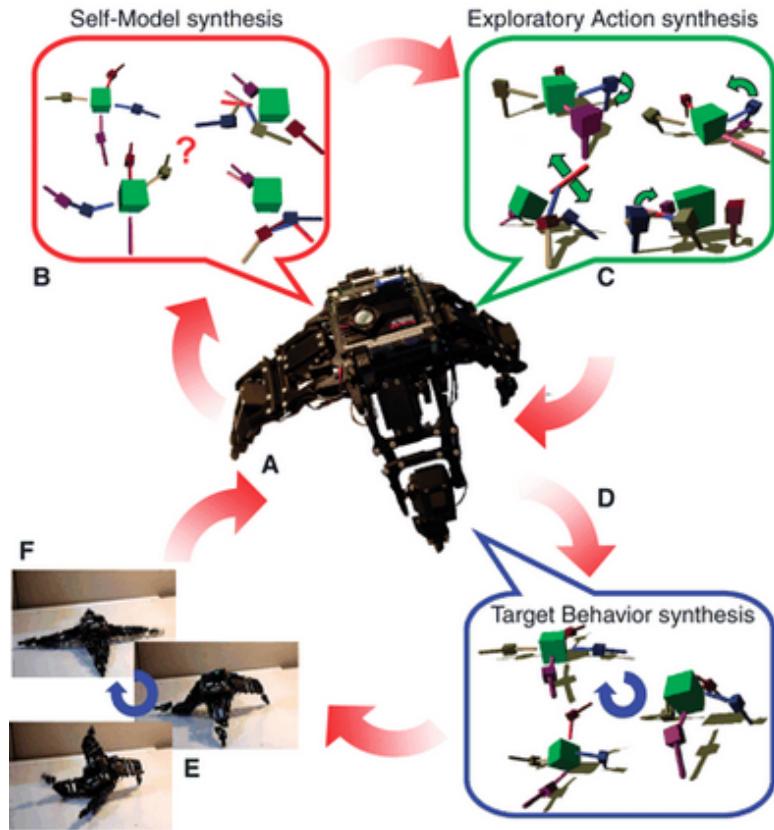


FIGURA 4.1: Esquema del algoritmo que puede usar un robot para desplazarse haciendo uso de Automodelos. Imagen tomada de [9]

4.2. Educación

Un enjambre de robots puede presentar muchas ventajas dentro del aula. Si se tiene un sistema de fácil uso para los alumnos, el profesor puede asignar una tarea a un grupo de estudiantes donde cada uno tiene la responsabilidad de controlar o programar un robot para que el conjunto logre una meta determinada como ordenar unos bloques o hacerse cargo de regar un pequeño huerto. Abusando un poco del concepto de la colectividad, incluso pueden generarse tareas donde cada colegio se especializa en un tipo de tareas para luego juntar los distintos robots y probar cómo interactúan.

Tener un setup con robots que demuestren un comportamiento colectivo puede ser muy ventajoso para promover el aprendizaje de roles sociales si la docente a cargo programa los robots para que tengan distintos roles como Líder y participante. Este mismo juego de roles puede ayudar a los niños a generar estrategias y herramientas para enfrentarse al mundo. En síntesis haciendo juegos para los niños con los robots se puede tener su atención para reforzar su educación y favorecer en el desarrollo de habilidades y competencias.

4.3. Usos Militar

Con un enjambre de robots, se puede simular una situación de catástrofe, donde es necesario poder desplegar un grupo de robots para hacer una tarea de reconocimiento y así poder buscar personas heridas o atrapadas.

4.4. Usos Doméstico

Existen las aspiradoras Roomba, que sin necesidad de un operario humano pueden aspirar nuestras casas. Ellas recorren nuestro hogar y gracias a sus sensores pueden auto generar un mapa del entorno. Este modelo del hogar puede hacerse mucho más rápido si en vez de tener un solo robot, se tienen 20. El mismo concepto se puede aplicar en seguridad del hogar, donde se pueden tener varios mini robots haciendo rondas en el perímetro de nuestra casa y en forma colectiva abarcan lo más posible. Todas estas situaciones se pueden simular con los robot MODI.

Capítulo 5

Mejoras futuras y Conclusiones

5.1. Mejoras futuras

Para poder tener una primera versión de MODI se tuvieron que dejar algunas cosas como mejoras futuras. En estas subsecciones se explican algunas de ellas y como se deben implementar.

5.1.1. Encoders

Una de las características claves de MODI es que hace uso de un tag fiducial, para poder tener información de la posición y orientación de cada robot. Con esto es suficiente para controlar la navegación del robot, pero si se necesita tener un control más fino sobre cada uno de los motores es fundamental incluir encoders en las ruedas. Por los motores que se tiene, se sugiere usar los encoders Pololu ¹.

5.1.2. Cables incrustados y Soporte motor

Es ideal que el mismo chasis del robot pueda contener las conexiones eléctricas y así reducir aún más la dificultad de ensamblado. Es por esto que se propuso un diseño con pequeños canales para incrustar los cables de los motores y que, además permite usar pins en los extremos que sirven para conectar eléctricamente los motores. Se puede ver en la figura 5.1 esta idea, junto con una versión mejorada del Soporte para los motores Pololu, que son motores ampliamente usados en robotica.

¹<http://www.pololu.com/catalog/product/1217>

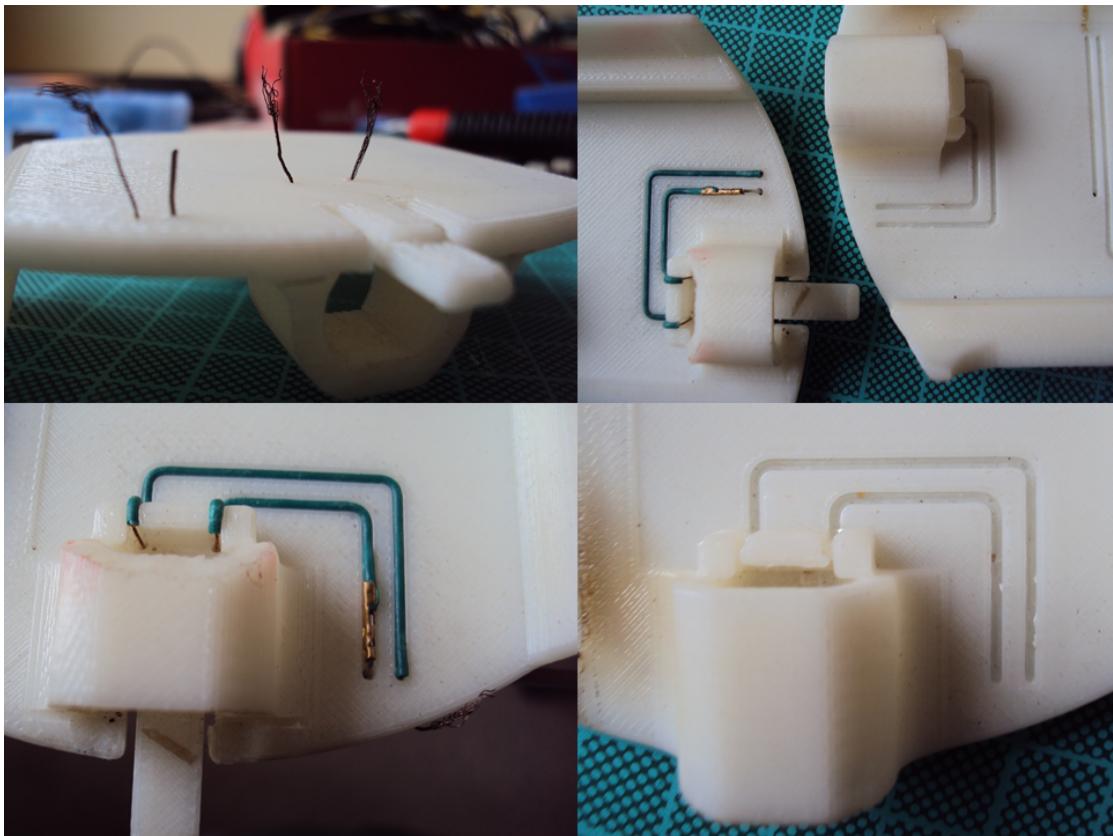


FIGURA 5.1: Diseño propuesto por Dr. Pablo Prieto para optimizar el cableado eléctrico en el chasis de MODI. También se puede ver otra versión del Soporte motor, que incluye una pestaña que afirma al motor.

5.1.3. Una PCB para todo

Para simplificar las conexiones del robots es necesario diseñar un nuevo PCB que incluya: Arduino, Driver motores y socket XBee, componentes electrónicos de MODI en Figura 3.10. Esto ya que demanda cierto tiempo tener que hacer los conectores necesarios para que se comuniquen entre si. Tener este PCB además implica bajar los costos de fabricación ya que es una sola PCB que se manda a fabricar, en vez de comprar tres en el mercado. Cabe destacar que esto debe ser un trabajo relativamente simple ya que todas las placas utilizadas son Open Source, lo que implica que sus archivos de diseño se encuentran completamente libre.

5.1.4. Módulo de carga inalámbrica

La elección del tener un Lipo Rider Pro en el robot en parte considera el uso de diversas formas de carga de energía. De forma simple se puede conectar un panel solar con conector JST 2.0, y en este mismo conector se puede poner un modulo de carga

inalámbrica como el que vende SeeedStudio². Estas alternativas permiten tener un setup de robots funcionando ininterrumpidamente ya que no es necesario intervenir los robots para cargar sus baterías.

5.2. Conclusiones

Diseñar un robot es una hermosa misión que implica desarrollar habilidades en varias áreas. Por lo mismo es necesario tener un equipo multidisciplinario que pueda abordar el problema desde distintas perspectivas. Para este trabajo fue fundamental contar con visiones expertas en robótica, diseño y biología-electrónica para hacer que MODI sea un gran proyecto que en un futuro pueda involucrar a mucha gente.

Con el equipo de trabajo armado los desafíos son adaptarse a la realidad nacional, donde se tienen menos recursos, tanto monetarios como de *materias primas electrónicas*. Por esto es necesario lograr hacer pruebas a pequeña escala para tomar decisiones en el diseño. Una herramienta fundamental para hacer pruebas a pequeña escala son las impresoras 3D que permiten en un día probar varios modelos distintos y hacer pruebas que no implican grandes pérdidas. Además de las impresoras 3D, con acceso a una Router CNC y un laboratorio básico de electrónica se puede hacer un producto que puede competir en el mercado.

Otras herramientas importantes en este trabajo son las tecnologías Open Source, que más bien es una filosofía donde se entrega de manera libre a la comunidad trabajo de gran calidad para que sea *utilizado, modificado y compartido*. Muchos de los elementos importantes de MODI son Open Source, lo que implica que MODI también es un proyecto Open. Gracias a que se incluyeron estas tecnologías se pudo reducir el tiempo de desarrollo drásticamente.

MODI como robot es simple, tan simple que si no se tienen aplicaciones concretas el proyecto puede diluirse. Es importante hacer difusión del proyecto y en especial trabajar con áreas que estén algo más alejadas de la electrónica, que son las más beneficiadas con MODI ya que al tener una curva de aprendizaje rápida, permite enfocarse sólo en la tarea fundamental y dejar de lado los aspectos técnicos que implican construir un robot. Se recomienda que gente del área de la zoología, biología, visión, psicología, educación, informática, por nombrar algunas, se les capacite y facilite el acceso a esta plataforma que puede ayudar a desentrañar aspectos claves de nuestra naturaleza.

Un ejemplo de aplicación en otras áreas es el uso de este robot en educación. Muchos de los elementos incluidos en MODI, ya sea software o hardware, son grandes herramientas

²<http://www.seeedstudio.com/depot/wireless-charging-module-p-1354.html>

que pueden aplicarse para resolver otros problemas y gracias a la simplicidad del proyecto quedan expuesto de manera clara y modular. Es por esto que MODI es ideal como robot para introducirse en la robótica, permitiendo al usuario de manera libre estudiar todas sus características en la medida que lo necesite. De lograrse esto, se tendría un gran impacto educacional en Chile ya que MODI podrá ser la puerta de entrada para la robótica.

Apéndice A

Algunas cosas interesantes que faltó incluir en el trabajo. Primero una imagen de algunas versiones de MODI, que fueron posibles por tener acceso a impresoras 3D ya que hacen muy simple el proceso de hacer cambios en los prototipos.

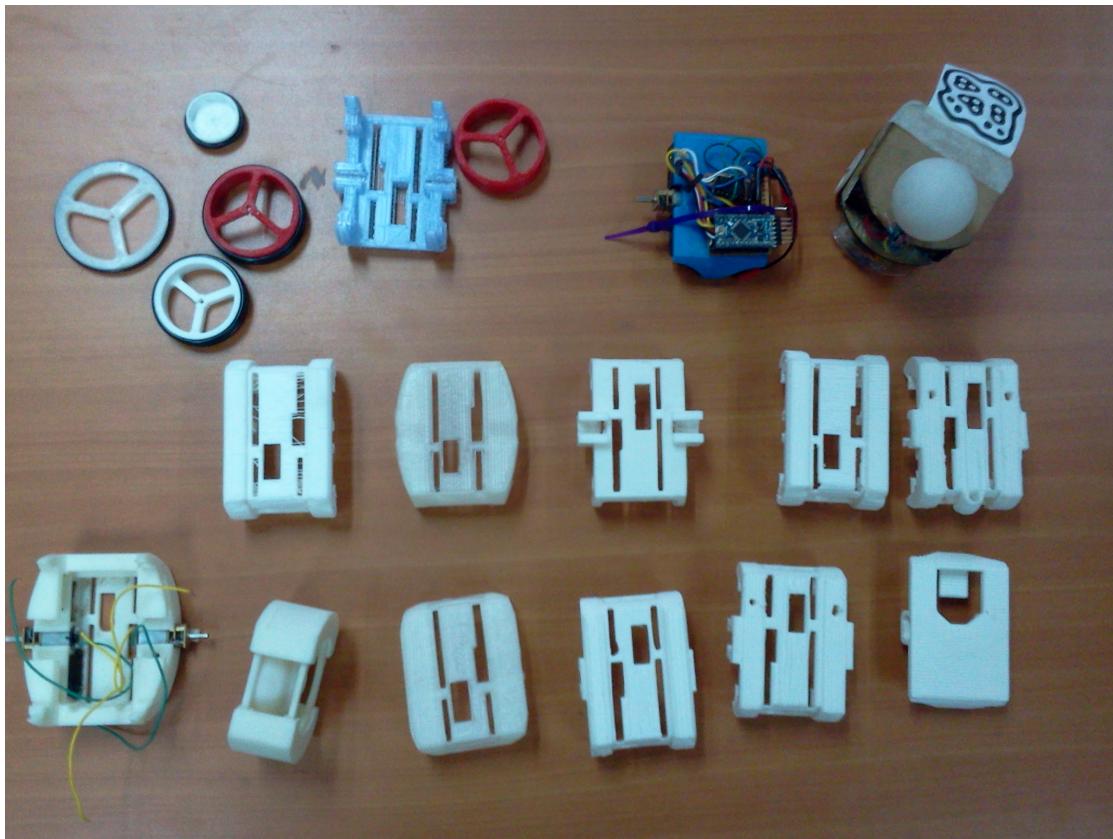


FIGURA A.1: Algunos de los modelos que se construyeron para llegar a la versión final de MODI.

Lo segundo es un diagrama con las conexiones eléctricas usadas en el PCB diseñado para MODI.

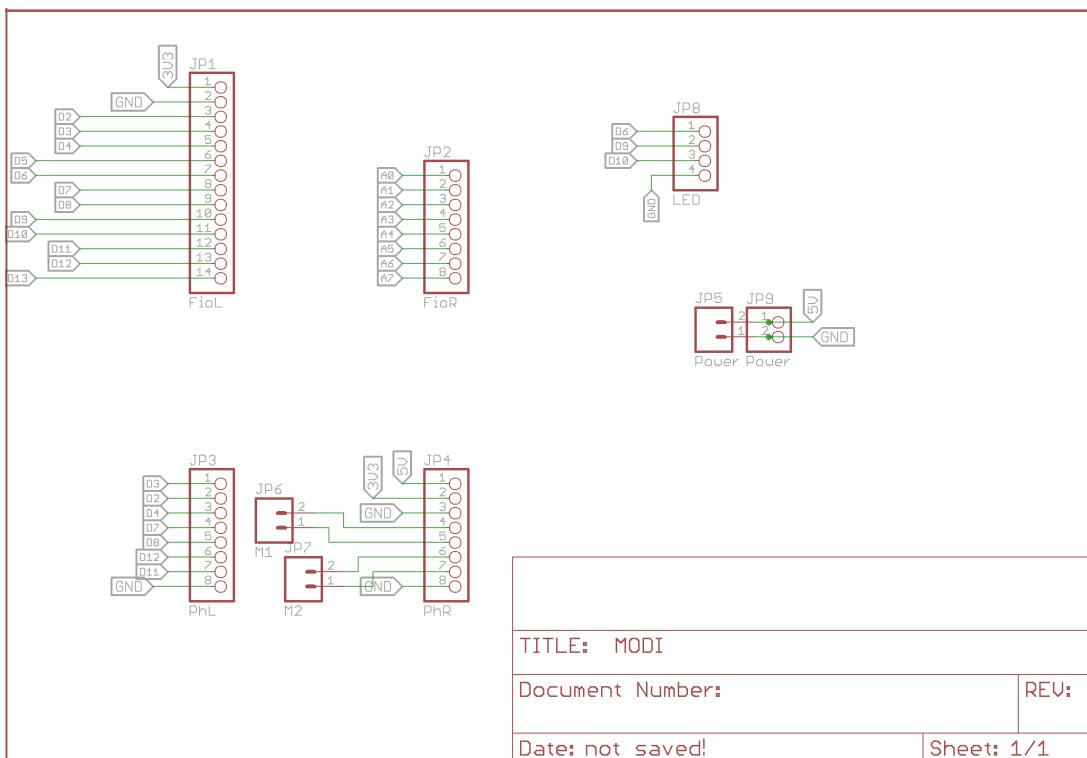


FIGURA A.2: Diagrama de conexiones eléctricas en PCB diseñada para MODI. Está hecho en Eagle 6.4.0.

Referencias

- [1] Walter Dan Stiehl, Cynthia Breazeal, Kuk-Hyun Han, Jeff Lieberman, Levi Lalla, Allan Maymin, Jonathan Salinas, Daniel Fuentes, Robert Toscano, Cheng Hau Tong, Aseem Kishore, Matt Berlin, and Jesse Gray. The huggable: a therapeutic robotic companion for relational, affective touch. In *ACM SIGGRAPH 2006 Emerging technologies*, SIGGRAPH '06, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6. doi: 10.1145/1179133.1179149. URL <http://doi.acm.org/10.1145/1179133.1179149>.
- [2] Jodi Forlizzi and Carl DiSalvo. Service robots in the domestic environment: a study of the roomba vacuum in the home. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, HRI '06, pages 258–265, New York, NY, USA, 2006. ACM. ISBN 1-59593-294-1. doi: 10.1145/1121241.1121286. URL <http://doi.acm.org/10.1145/1121241.1121286>.
- [3] S. Kernbach, P. Levi, E. Meister, F. Schlachter, and O. Kernbach. Towards self-adaptation of robot organisms with a high developmental plasticity. In *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:*, pages 180–187, 2009. doi: 10.1109/ComputationWorld.2009.11.
- [4] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65, 2009.
- [5] Branislav Thurský and Gabriel Gašpar. Using pololu's 3pi robot in the education process.
- [6] M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3293–3298, 2012. doi: 10.1109/ICRA.2012.6224638.

- [7] Martin Kaltenbrunner and Ross Bencina. reactivision: a computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 69–74. ACM, 2007.
- [8] Juan Cristóbal Zagal and Hod Lipson. Self-reflection in evolutionary robotics: resilient adaptation with a minimum of physical exploration. In Franz Rothlauf, editor, *GECCO (Companion)*, pages 2179–2188. ACM, 2009. ISBN 978-1-60558-505-5. URL <http://dblp.uni-trier.de/db/conf/gecco/gecco2009c.html#ZagalL09>.
- [9] Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006. doi: 10.1126/science.1133687. URL <http://www.sciencemag.org/content/314/5802/1118.abstract>.