

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

Síntesis de Máquinas Inteligentes
Ing. civil Mecánica, Universidad de Chile

2 de julio de 2013

UNIVERSITY NAME (IN BLOCK CAPITALS)

Abstract

Faculty Name
Ing. civil Mecánica, Universidad de Chile

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

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Índice general

Abstract	I
Acknowledgements	II
List of Figures	V
List of Tables	VII
1. Robot	1
1.1. Introducción	1
1.2. Desarrollo tecnológico desde Chile	2
1.3. ¿Qué es un robot?	3
2. Swarm	5
2.1. Comportamiento colectivo en la naturaleza	5
2.2. Robots para construir un enjambre	6
2.2.1. Kilobot	6
2.2.2. Organismo Multibot	7
2.2.3. E-puck	7
2.2.4. 3pi Robot	7
2.3. Necesidades de mercado	8
2.4. Necesidades de mercado	9
2.4.1. Usos Académico	10
2.4.2. Usos Militar	10
2.4.3. Usos Doméstico	10
3. MODI	11
3.1. Setup	11
3.2. Construcción	12
3.2.1. Fabricación Digital vs Análoga	12
3.2.2. Software CAD	14
3.2.3. Impresora 3D	15
3.2.4. Diseño PCB	18
3.3. Componentes electrónicos	20
3.3.1. Arduino FIO	21

3.3.2. Motores DC	23
3.3.3. XBee	23
3.3.4. Lipo Rider Pro	25
3.3.5. Motor Driver 1A Dual TB6612FNG	26
3.3.6. LED RGB	27
3.4. Componentes mecánicos	27
3.5. Locomoción	30
3.6. Tracking 2D	33
3.7. Software	34
3.8. Bill Of Material	34
4. Aplicaciones	35
4.1. Auto modelamiento	35
4.2. Educacion	37
4.2.1. Subsection 2	37
5. Conclusiones	38
5.1. Mejoras futuras	38
5.1.1. Subsection 1	38
5.1.2. Subsection 2	38
5.2. Conclusiones	38
A. Appendix Title Here	39
Bibliografía	40

Índice de figuras

1.1. Digesting Duck	3
1.2. Roomba y KUKA	4
2.1. Bandada	5
2.2. 3pi	8
2.3. Tabla	9
2.4. nicho	9
3.1. modi	11
3.2. Setup	12
3.3. modireplicator	13
3.4. Fablabs	14
3.5. ModiSolidWorks	15
3.6. Render modi	15
3.7. 3Dprint	16
3.8. 3DTouch	17
3.9. uprint	17
3.10. Replicator1	18
3.11. cables	18
3.12. MDX20	19
3.13. pcbeagle	19
3.14. pcb	20
3.15. fio	21
3.16. fio	22
3.17. fio	22
3.18. MotorDC	23
3.19. xbee	24
3.20. lipopro1	26
3.21. Bateria	26
3.22. TB6612FNG	27
3.23. ledRGB	27
3.24. chasismodi	28
3.25. ruedamodi	29
3.26. accesorio	29
3.27. logomodi	30
3.28. Atlas	31
3.29. DCMotor	31
3.30. pwmChart	31

3.31. pwm	32
3.32. Fiducial	33
4.1. Automodelado	36
4.2. Automodelado	36
4.3. AutomodeladoLIPSON	37

Índice de tablas

3.1. Comparación impresoras 3D	16
3.2. Especificaciones Fio	22
3.3. Motor	23

Dedicado Pía, mi fiel compañera de aventuras...

Capítulo 1

Robot

1.1. Introducción

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 cuento 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 la curiosidad mía 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 creo que los que más he perseguido para que me enseñaran lo que hacían fueron Linus Casassa y Jaime Martínez.

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, 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.

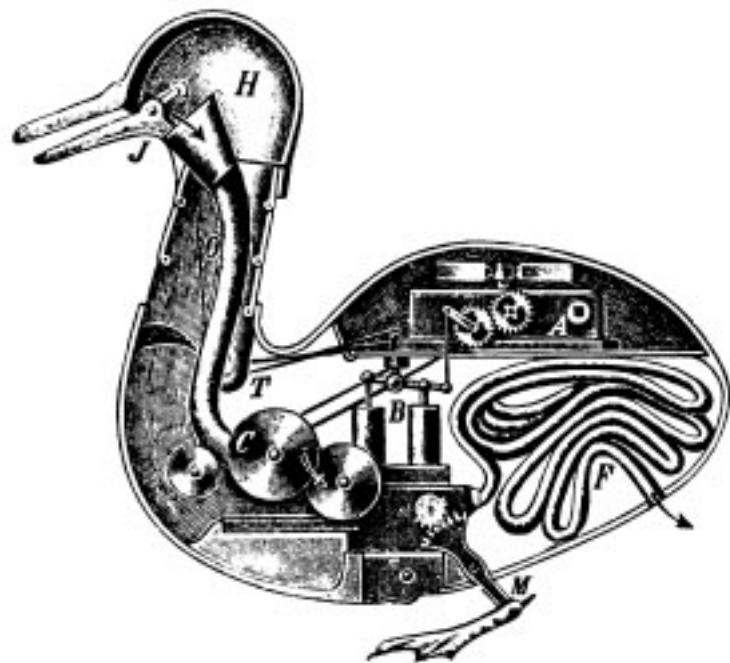


FIGURA 1.1: Digesting Duck, creado por Jacques de Vaucanson en 1739. Imagen extraída de Wikipedia.

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 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 1.2: **Izquierda:** Robot Roomba, primer Robot doméstico vendido en Chile, Imagen extraída de <http://www.irobotchile.cl>. **Derecha:** Robot industrial KUKA KR 1000 TITAN, Imagen extraída de <http://www.turbosquid.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, 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.

Capítulo 2

Swarm

2.1. Comportamiento colectivo en la naturaleza



FIGURA 2.1: Bandada de auklets, teniendo comportamiento de enjambre. Imagen extraída 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.

A continuación algunos robots que pueden ser utilizados para hacer un Robot Swarm junto con las información técnica disponible.

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, O. Kornienko, A. Nagarathinam y P. Levi., exploran el trabajo colaborativo en robots para un mejor rendimiento y mayor fiabilidad a nivel macroscopico. En este articulo 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]

2.2.3. E-puck

Uno de los robots más utilizados por los científicos en el mundo para estudios y publicaciones es el E-puck. 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 computador 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 Getronic 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, diseño el 3pi Robot. 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. Existen 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.

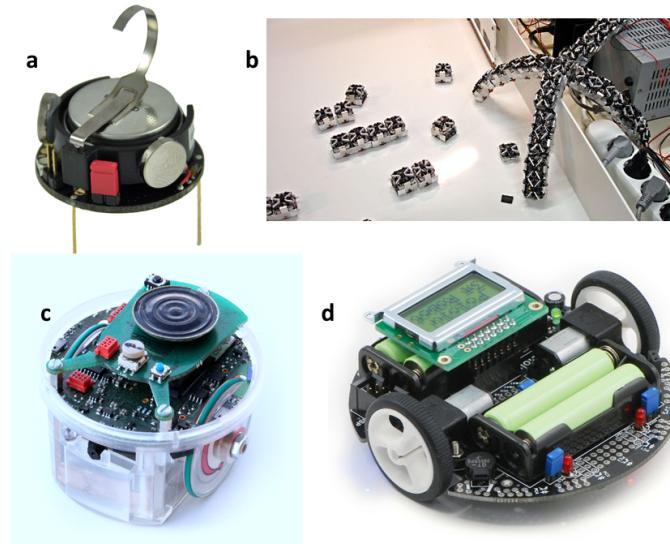


FIGURA 2.2: 3pi Robot, simple y económico pero no tiene comunicación inalámbrica.
Imagen extraída de <http://www.skpang.co.uk>

En la tabla 2.3 comparativa resume las principales características de los robots: Kilobot¹, E-puck² y 3pi³.

2.3. Necesidades de mercado

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: Tabla comparativa robots

2.4. Necesidades de mercado

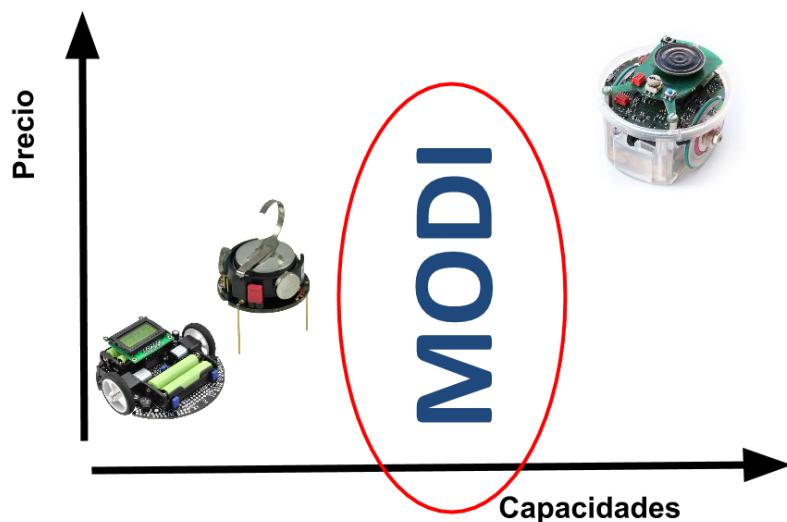


FIGURA 2.4: Nicho de mercado

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 capaz de controlarse de forma inalámbrica, simple de construir y fácil de usar. Además debe ser económico para poder construir varios. Se pretende hacer uso de tecnologías como impresoras 3D y de desarrollo como el Open Hardware.

2.4.1. Usos Académico

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 ayudar a niños con trastornos como el Asperger a practicar sus habilidades para reconocer estos mismo comportamientos.

2.4.2. Usos Militar

2.4.3. Usos Doméstico

Capítulo 3

MODI

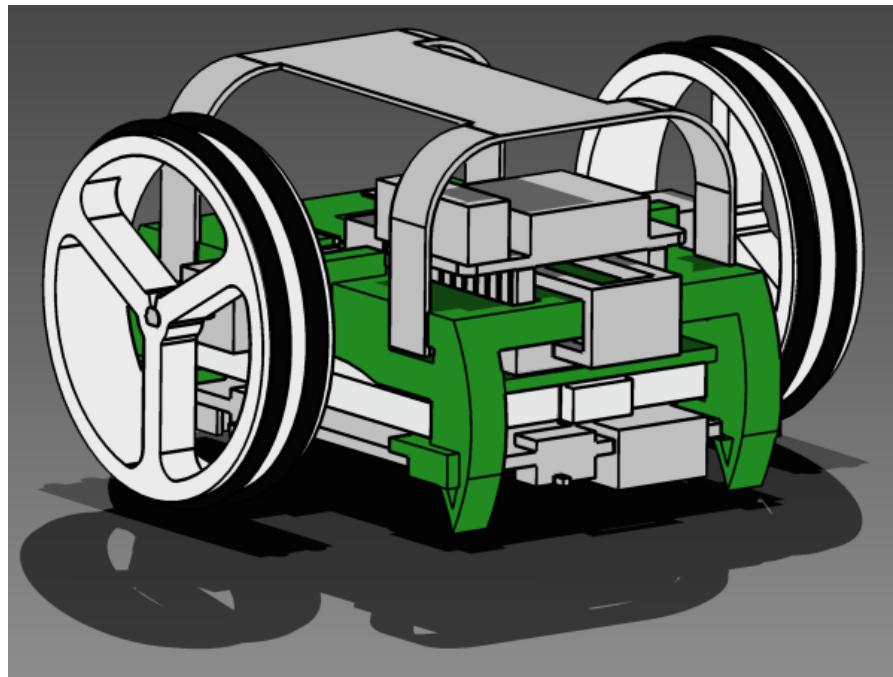


FIGURA 3.1: robot MODI (sigla para Modular Intelligence)

3.1. Setup

Se desea realizar una investigación sobre el comportamiento colectivo de un grupo de robots, en el mundo real sin hacer uso de simuladores. Por esto 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.

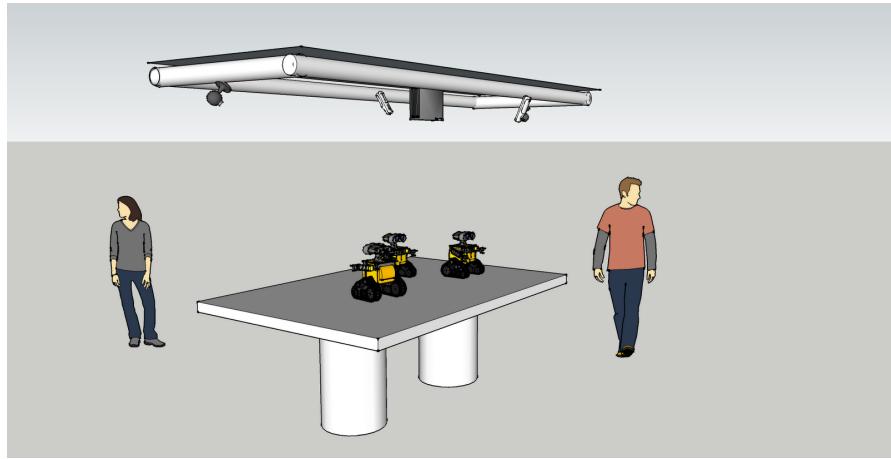


FIGURA 3.2: Setup a montarse para hacer estudios de grupos de robots.

La función principal de MODI es ser una plataforma móvil de fácil acceso. Existe un repositorio en GitHub donde se tienen los códigos actualizados para controlar y construir robots MODI. Para descargar códigos y piezas de construcción se debe ir a github.com/FabLabUCh/modi

Las funciones principales que se desarrollaron

- 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

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 hecho en un computador, obtener un objeto real 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.

Utilizando una impresora 3D Makerbot Replicator 1 se hizo esta primera versión con piezas plásticas.

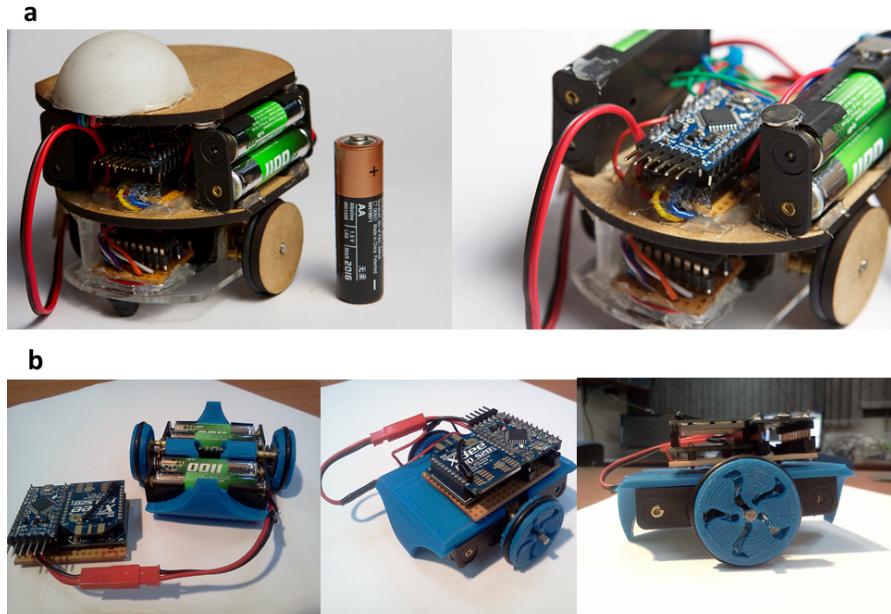


FIGURA 3.3: Primera versión de MODI usando técnica de *Fabricación Digital* con chasis de plástico construido con una MakerBot Replicator 1

Esta versión se dio de baja ya que 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 maquinas 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 3.4. 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 3.4: Mapa actual de lugares en el mundo que cuentan con un Fab Lab. En Chile a la fecha existen 3. Imagen obtenida desde 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.

Según Wikipedia, ”... 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 3.2.

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 logró este modelo, Figura 3.6 que tiene cinco piezas.

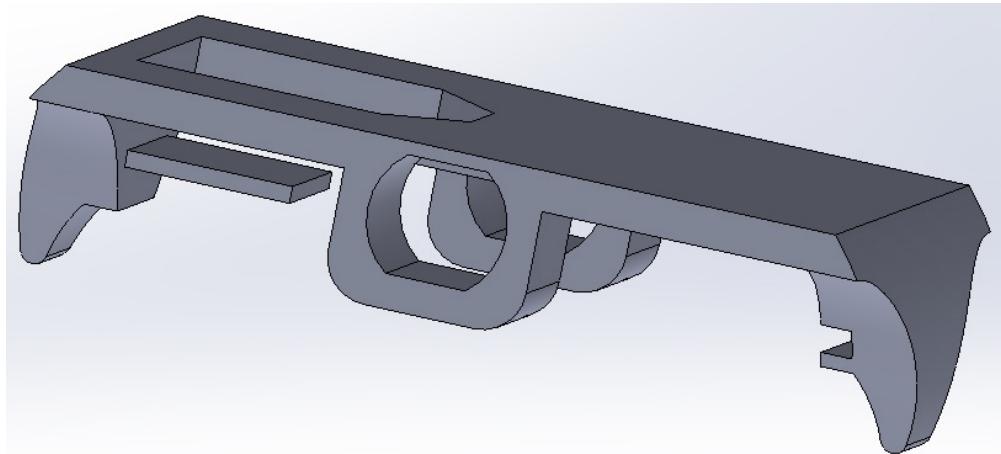


FIGURA 3.5: Primer Chasis de MODI, realizado con SolidWorks 2012. Esta versión sirve solo como prueba de concepto ya que no tiene espacio para las conexiones eléctricas necesarias

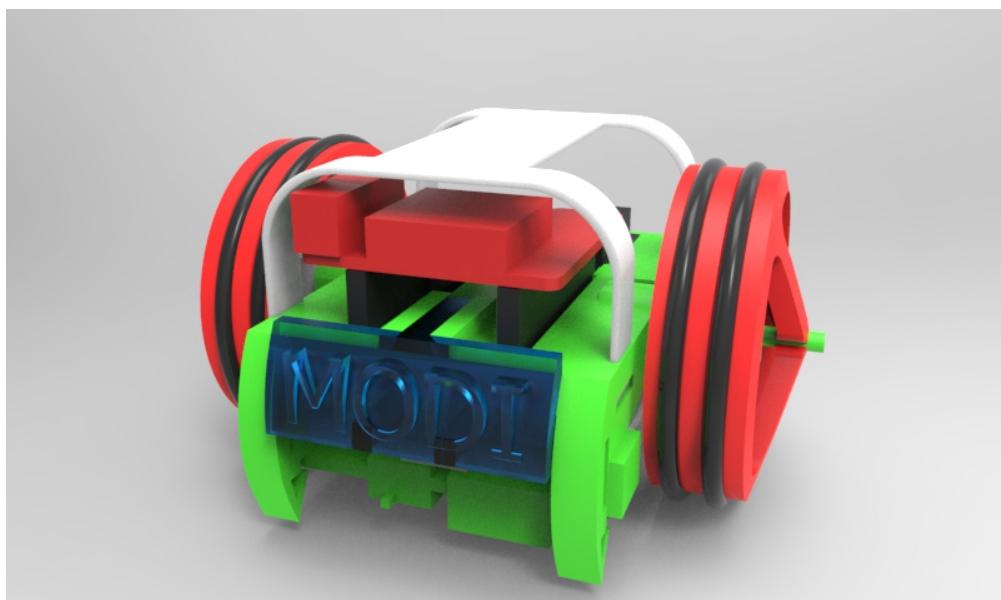


FIGURA 3.6: Modelos 3D de MODI realizado con Autodesk Inventor 2013 y renderizado con KeyShot4.

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.

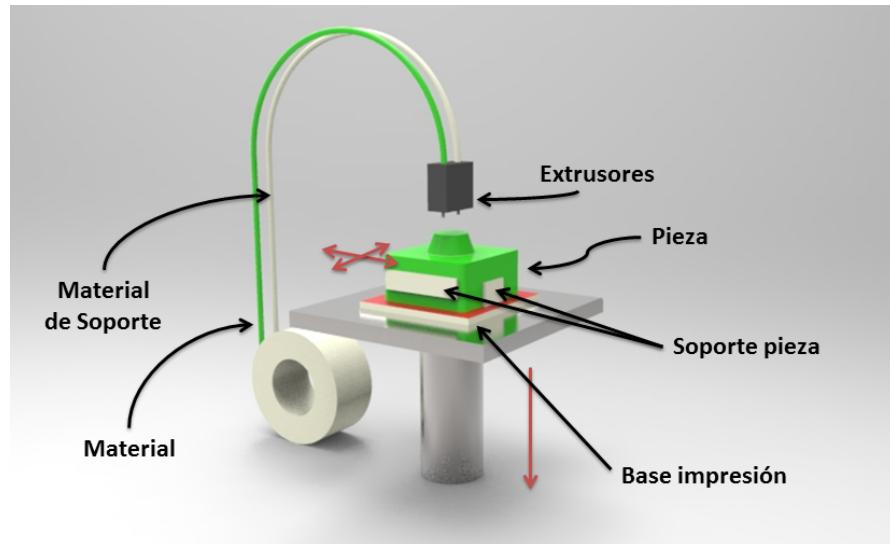


FIGURA 3.7: Modelo de funcionamiento para impresoras 3D MDF. Tiene dos materiales, uno para construir la pieza y otro que funciona como soporte estructural para la impresión, además existe una base que se calienta para sujetar la pieza mientras se construye.

	Replicator 1	3D Touch	Uprint
Volumen impresión	5m 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 , 11 μm XY	0.1mm Z 0.2mm XY	
Material soporte	con herramienta	con herramienta	Necesita proceso extra
Precio insumo	60USD/Kg	100USD/Kg	400USD/Kg

TABLA 3.1: Comparación impresoras 3D

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]¹ cada capa en eje z. Luego por facilidad de uso y menor tiempo de impresión se usa una impresora MakerBot Replicator 1

¹teambastech.com/Store/index.php?route=product/product&product_id=205



FIGURA 3.8: Impresora 3D de Bit From Bytes, 3D Touch.

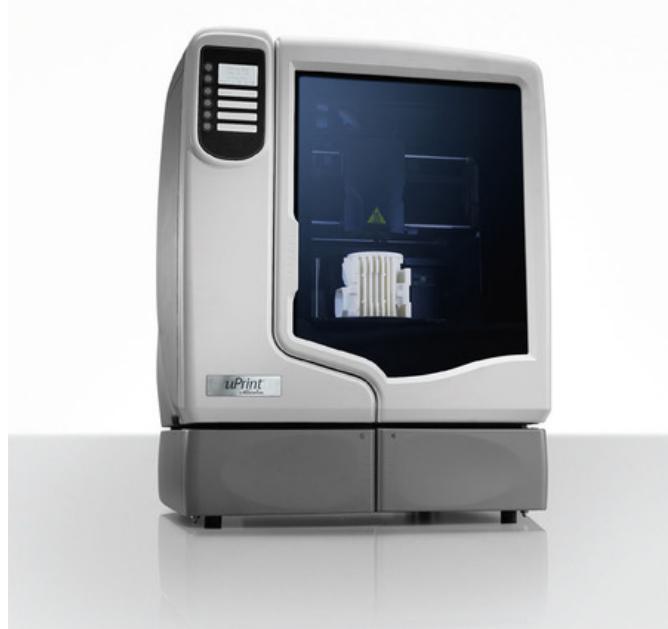


FIGURA 3.9: Impresora 3D de Stratasys, Uprint SE.

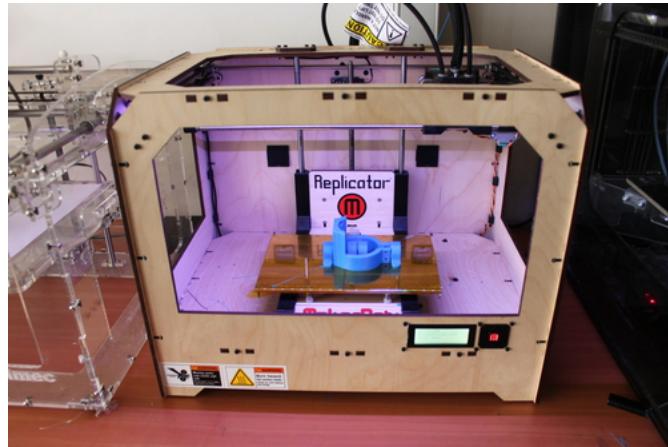


FIGURA 3.10: Replicator 1 de Makerbot, impresora 3D FDM utilizada para prototipar chasis y ruedas de robot MODI. En Laboratorio de Síntesis de maquinas inteligentes de la Universidad de Chile, junto a la Fab@Home y 3DTouch.

3.2.4. Diseño PCB

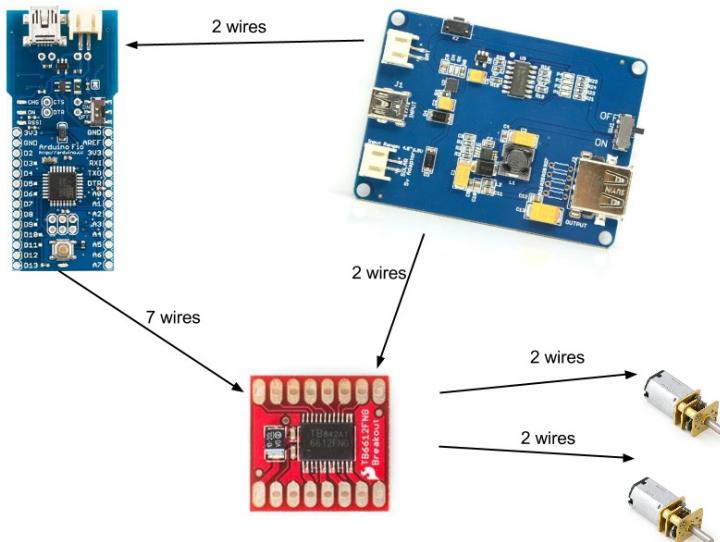


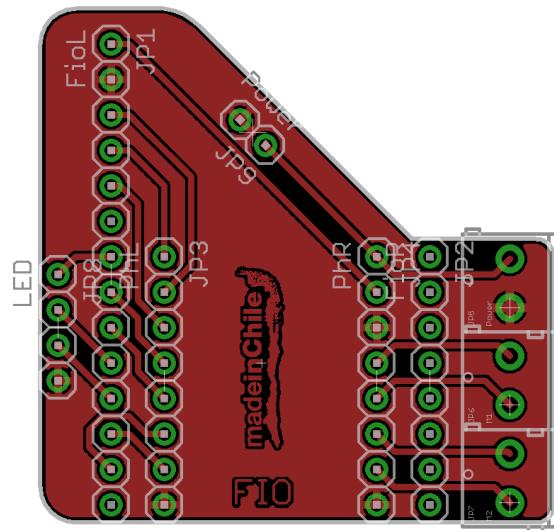
FIGURA 3.11: 15 conexiones que unen distintos componentes.

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 3.11. Por esto es necesario un PCB que permita reducir al mínimo el espacio necesario para energizar y comunicar los elementos. El requerimiento principal es poder fabricarse con una fresa CNC como la Roland Modela MDX-20, por

lo que 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 3.13.



FIGURA 3.12: Roland Modelo MDX-20 usada para hacer PCB y modelos 3D. Imagen tomada de roland.com



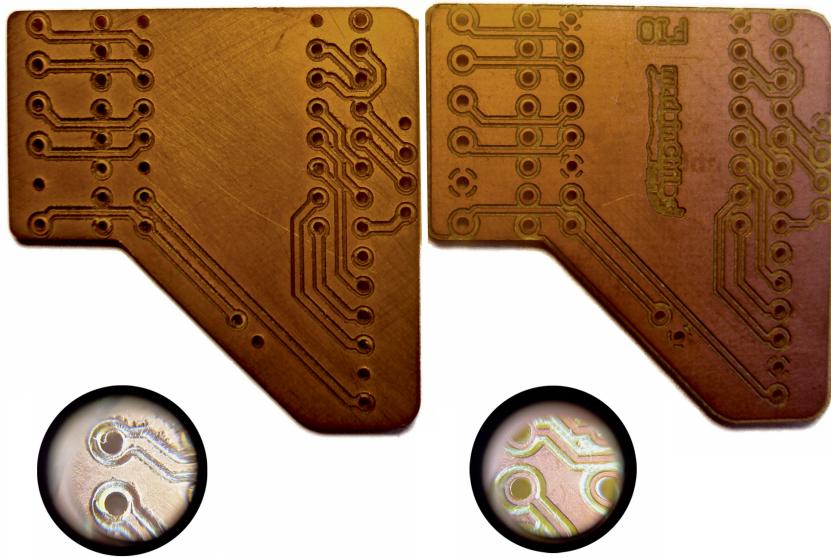


FIGURA 3.14: Misma PCB fresada en dos maquinas diferentes. Se puede apreciar en el detalle aumentado 60x, la importancia de contar con una buena resolución de fresado. En especial es necesario para tener un buen pad.

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 **Fiduciales**, 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 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

Microcontrolador El Arduino FIO 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 tabla 3.2 resume sus características principales.

Arduino FIO. Similar al arduino mini pro pero además incluye conexión para módulo XBee. Imagen tomada de arduino.cc

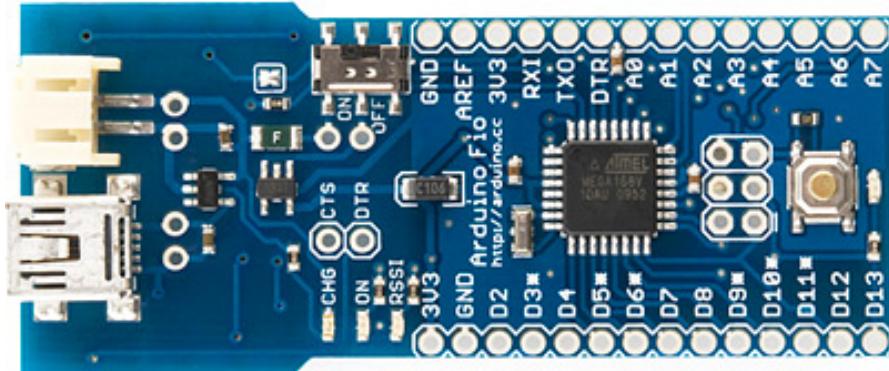


FIGURA 3.15: Arduino FIO. Similar al arduino mini pro pero además incluye conexión para módulo XBee. Imagen tomada de arduino.cc

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 (con 6 salidas PWM)
Input Pins Analogos	8
Corriente DC en cada Pin I/O	40 mA
Memoria Flash	32 KB (2KB son usados para Bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock	8 MHz

TABLA 3.2: Especificacion tecnica de Arduiono 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

FIGURA 3.16: Arduino FIO. Similar al arduino mini pro pero además incluye conexión para módulo XBee. Imagen tomada de arduino.cc

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

FIGURA 3.17: Arduino FIO. Similar al arduino mini pro pero además incluye conexión para módulo XBee. Imagen tomada de arduino.cc

Velocidad	440 RPM @6 V, 40 mA
Torque	3 Kg-mm
Reducción	1:30
Consumo trancado	0.36 A

TABLA 3.3: Especificación técnica de Motor DC Pololu

3.3.2. Motores DC

Este motor Pololu es muy pequeño (largo: 9.27 mm), por lo mismo usa muy poca corriente. Además 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 tabla 3.3 resume sus características principales.

FIGURA 3.18: Motor pololu con reducción 1:30. Imagen tomada de pololu.com

3.3.3. XBee

Los dispositivos con los cuales interactuamos a diario poseen acceso a distintos tipos de sistemas de comunicación. Los llamados “Smartphone” cuentan con diversas antenas para conectarse, 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.

Para comunicar un robot con otro pueden usarse diversas técnicas. Dependiendo del área en particular que sea del interés del científico, los robots 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. Esta información es transmitida a cada robot por medio del protocolo zigbee implementado en los chip XBee de la empresa Digi, Figura 3.19. 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.

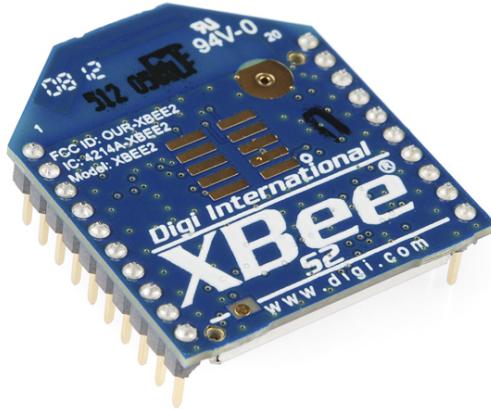


FIGURA 3.19: XBee es un producto de la empresa Digi basado en el conocido protocolo de comunicación inalámbrica ZigBee. El módulo XBee XBP24-AWI-001 funciona a 2.4GHz. La serie Pro tiene el mismo pinout y set de instrucciones que la serie básica más un incremento de la potencia de salida a 60mW. Estos módulos permiten extender la comunicación serial de forma inalámbrica. Con XBee se pueden hacer redes punto a punto y multipunto de forma fácil.

3.3V @ 295 mA. Tasa de datos máximo de 250kbps. 50mW de salida (+17dBm). 1600m de rango. Antena interna. Certificado total FCC. 4 pines de entrada de 10-bit ADC. 10 pines digitales de Entrada/Salida. Encriptación en 128 bit. Configuración local o ‘en el aire’. Set de comandos AT o API.

3.3.4. Lipo Rider Pro

El LiPo Rider Pro 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. Figura ??



FIGURA 3.20: LiPo Rider Pro, carga de batería y boost.

La Lipo Rider carga una batería de Litio Polímero. Posee una alta densidad de carga, 2000mAh a 3.7V. Figura 3.21

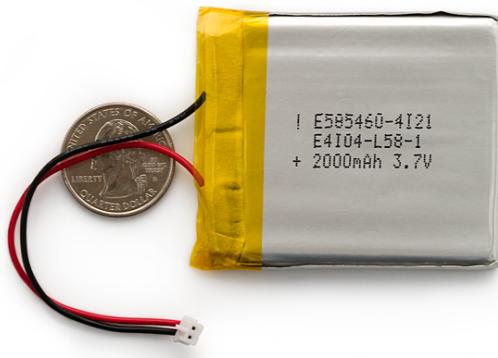


FIGURA 3.21: Batería lipo 2000mAh a 3.7V

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 ??

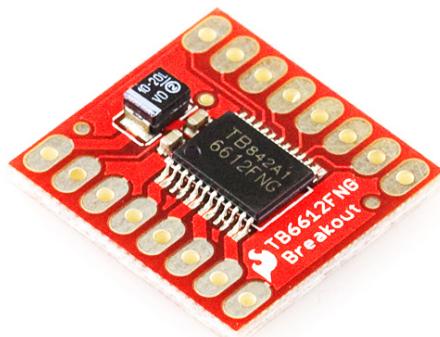


FIGURA 3.22: Driver motores TB6612FNG.

3.3.6. LED RGB

LED RGB difuso, de esta manera se obtiene un ángulo de visión mucho más amplio.

Características

Forward Voltage (RGB): (2.0, 3.2, 3.2)V

Max Forward Current (RGB): (20, 20, 20)mA

Max Luminosity (RGB): (2800, 6500, 1200)mcd

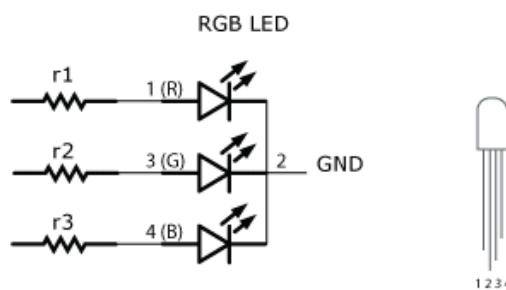


FIGURA 3.23: Diagrama conexión LED RGB y resistencias por color

3.4. Componentes mecánicos

Como se comentó en la sección 3.2.3, las piezas del robot fueron diseñadas para ser construidas por una máquina 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 ², para que estas uniones queden fijas y las ruedas esten en el eje correspondiente.

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

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

A continuación hay imágenes de todas las piezas renderizadas diseñadas.

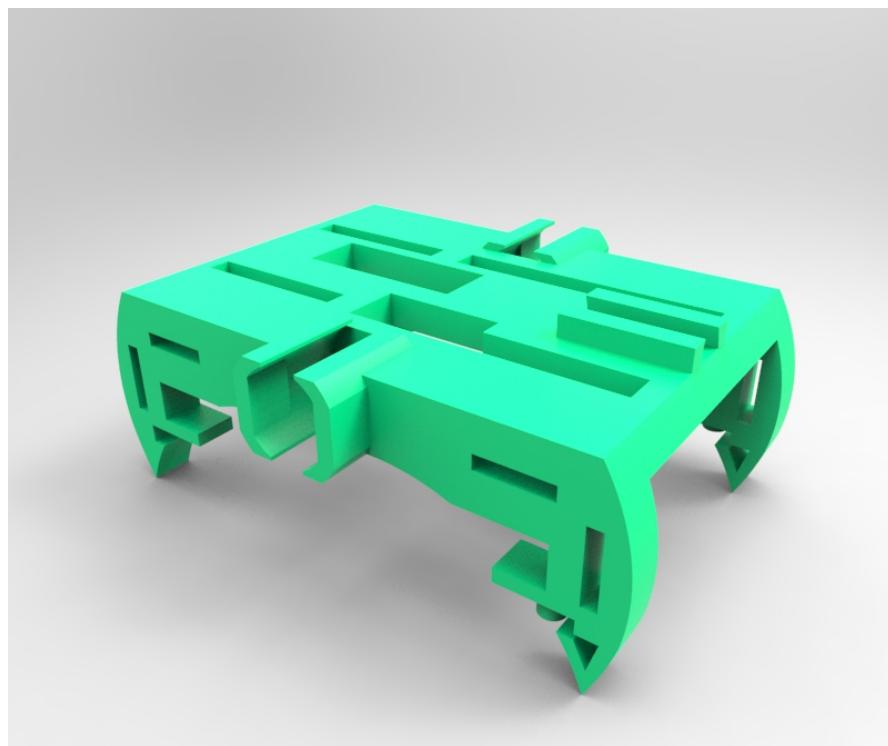


FIGURA 3.24: Chasis modi

²<http://campusurico.ualca.cl/~fespinoz/Ajustes%20y%20tolerancias%20mecanicas.pdf>



FIGURA 3.25: Rueda modi

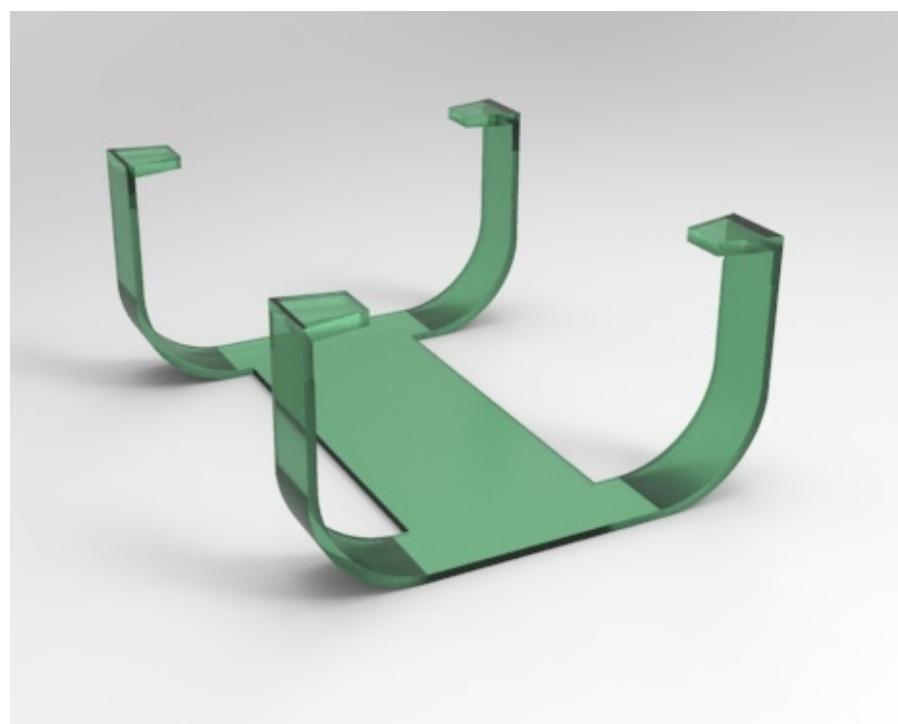


FIGURA 3.26: Accesorio modi

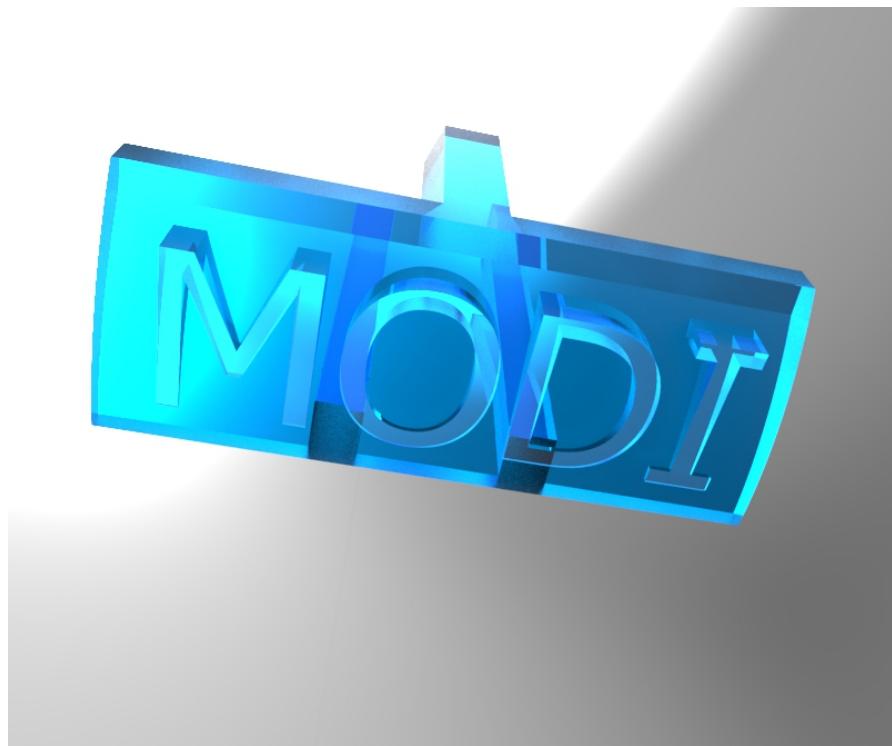


FIGURA 3.27: Logo modi

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 3.28. 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 web³.

Para moverse MODI tiene dos motores DC 3.18. Estos tienen una caja de reducción 30:1 que permiten aumentar el torque y disminuir la velocidad. La Figura 3.29 muestra la característica de estos.

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 3.22. Los valores van desde -255 a 255. Para más información sobre pwm y este chip ver esta pagina web⁴.

³en.wikibooks.org/wiki/Robotics/Types_of_Robots/Wheeled

⁴<http://www.pololu.com/docs/0J21/5.c>

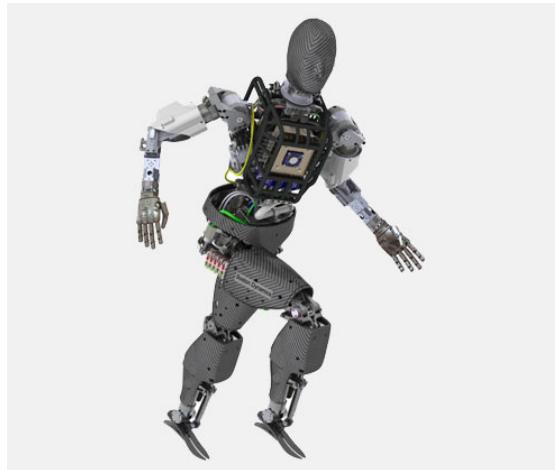


FIGURA 3.28: Atlas es un humanoide con gran movilidad, diseñado para poder usar herramientas humanas. Posee 28 grados de libertad con actuadores hidráulicos y la energía se le entrega por medio de un cable flexible. Imagen extraída de bostondynamics.com

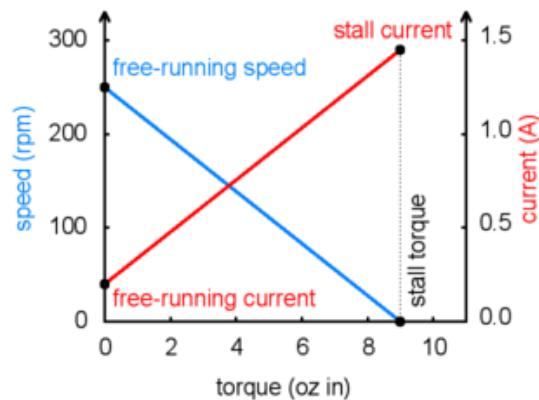


FIGURA 3.29: Operación motores: Corriente, velocidad y torque. Imagen extraída de pololu.com

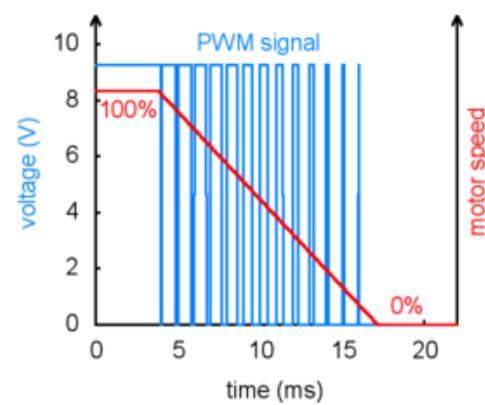


FIGURA 3.30: Control de velocidad PWM, se observa como disminuye la velocidad según el ancho del pulso pwm.

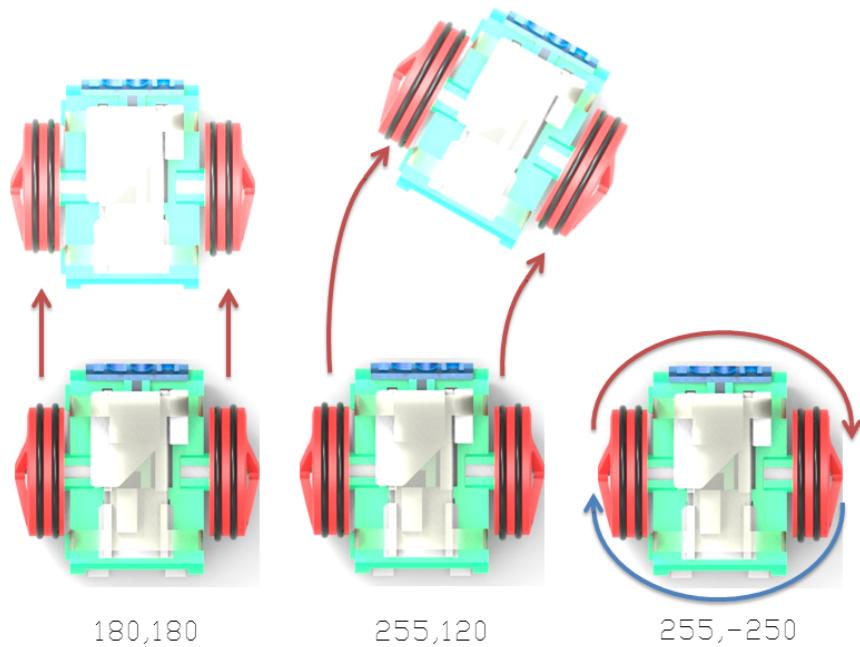


FIGURA 3.31: Movimiento diferencial. A diferencia de un auto, MODI tiene dos motores que giran de forma independiente. Las distintas velocidades y sentido de giro determinan el angulo para girar. Si se necesita solamente rotar los motores deben girar en sentidos opuestos.

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

3.6. Tracking 2D

Al igual que los seres vivos un robot necesita “sentidos” para poder tener 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 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.

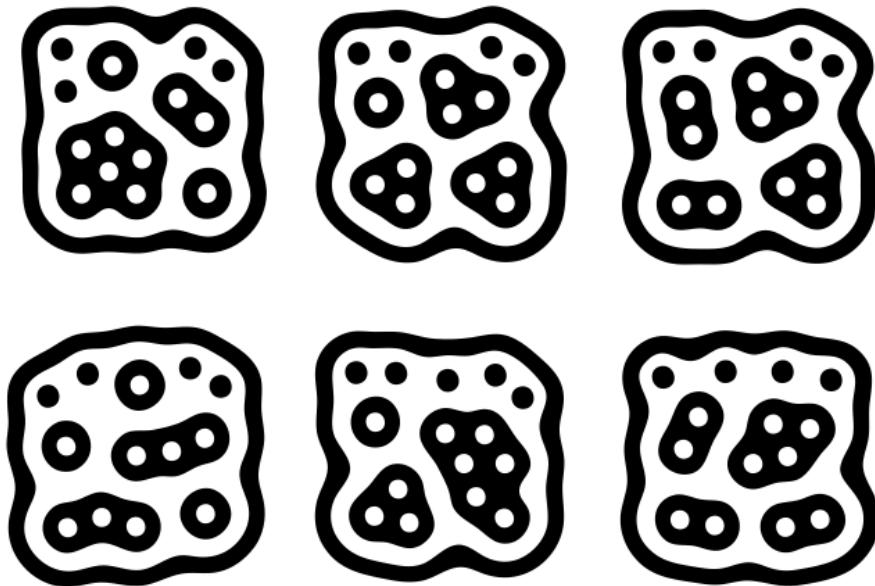


FIGURA 3.32: Códigos fiduciales utilizados en reactTIVision. Estos códigos aunque parecen algo extraños para los seres humanos están diseñados de tal forma que es muy fácil diferenciar unos de otros y además obtener su orientación y posición en un software de análisis de imágenes. Se tiene un total de 216 configuraciones posibles de códigos.

3.7. 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 github.com/FabLabUChile/modi/

<https://code.google.com/p/pytuio/> <http://www.tuio.org/?software> <http://pyserial.sourceforge.net/>
<http://www.aaronstannard.com/post/2012/08/17/How-to-Setup-a-Proper-Python-Environment-on-Windows.aspx>

<https://bugs.launchpad.net/python-uinput/+bug/752699>

3.8. Bill Of Material

<http://kitbom.com/otrab/modi/build-view>

Capítulo 4

Aplicaciones

4.1. Auto modelamiento

Juan Cristobal Zagal y Hod Lipson [1] exploraron el comportamiento de un robot capaz de entrar en un proceso de self-reflection. 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 si no que 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 4.3, 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

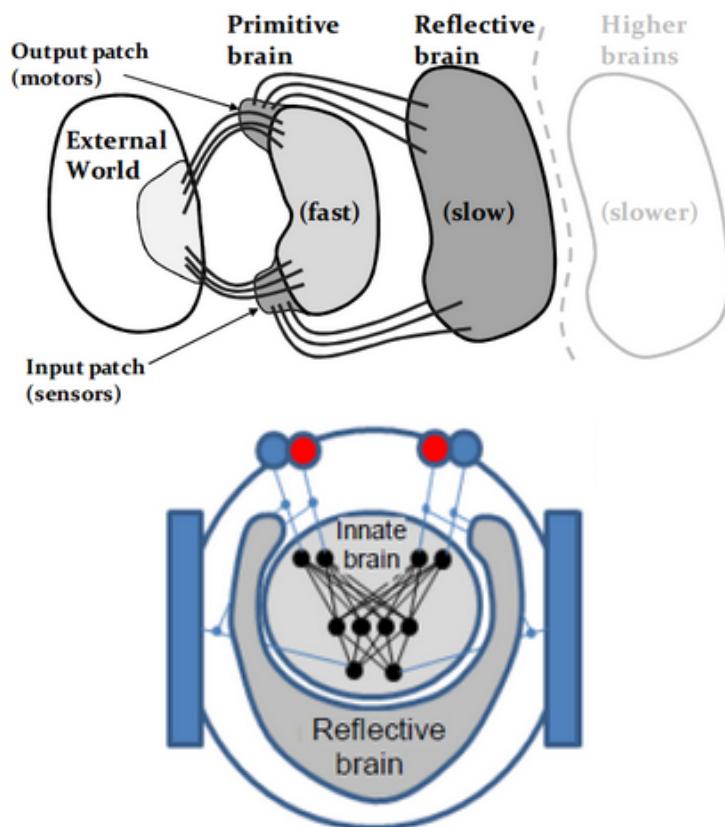


FIGURA 4.1: **Arriba:** Arquitectura de cerebros anidados usado en el trabajo de J.Zagal y H.Lipson, basado en modelos de M. Minsky. **Abajo:** Modelo de cerebros implementado en robot e-puck. Ambas imágenes pertenecen a Juan Cristóbal Zagal.

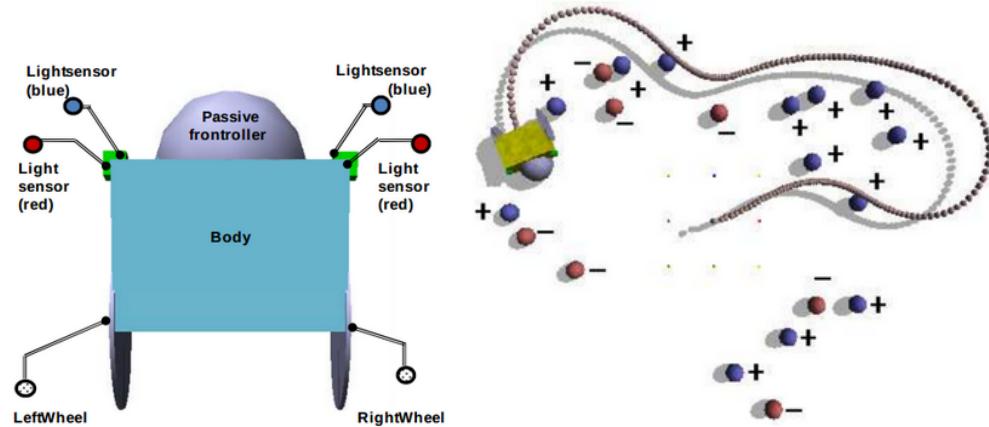


FIGURA 4.2: **Izquierdo:** Arquitectura de cerebros anidados usado en el trabajo de J.Zagal y H.Lipson, basado en modelos de M. Minsky. **Derecha:** Modelo de cerebros implementado en robot e-puck. Ambas imágenes pertenecen a Juan Cristóbal Zagal.

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.

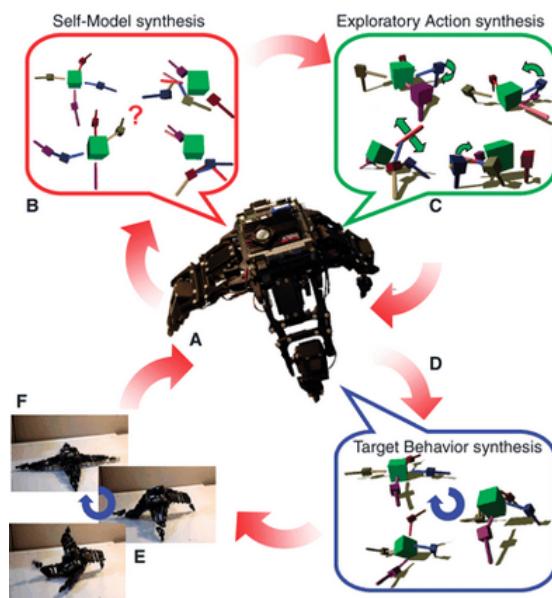


FIGURA 4.3: Esquema del algoritmo que puede usar un robot para desplazarse haciendo uso de Automodelos. Imagen obtenida desde <http://creativemachines.cornell.edu>

4.2. Educación

4.2.1. Subsection 2

Capítulo 5

Conclusiones

5.1. Mejoras futuras

5.1.1. Subsection 1

5.1.2. Subsection 2

5.2. Conclusiones

Apéndice A

Appendix Title Here

Write your Appendix content here.

Referencias

- [1] 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>.
- [2] 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.