

Campeonato de Robótica Educativa

Robocampeones 2005



Campeonato de Robótica Educativa

Robocampeones 2005



Financiado por el Ministerio de Educación y Ciencia
(675963C2/CCT001-04-00121)

Vicente Matellán Olivera
José María Cañas Plaza
Carlos Enrique Agüero Durán
Víctor Manuel Gómez Gómez
Francisco Martín Rico
Pablo Barrera González

Índice general

Índice de figuras	12
Índice de tablas	12
1. El Software Libre en la Robótica	1
1.1. Introducción	1
1.2. Software libre en la industria robótica	3
1.2.1. El robot Cyb de Probotics	3
1.2.2. El robot Pioneer robot de Activmedia	4
1.2.3. El proyecto OrocOS	5
1.3. Software libre en la enseñanza de la robótica	7
1.3.1. LEGO Mindstorms	7
1.3.2. El lenguaje Not Quite C (NQC)	8
1.3.3. El sistema operativo BrickOS	8
1.4. Software libre en la investigación en robótica	9
1.4.1. La competición de la Robocup	9
1.4.2. La plataforma Player/Stage	15
1.4.3. JDE	16
1.5. Conclusiones	18
2. Reglamento	19
2.1. Objetivos	19

2.2.	Participantes	19
2.3.	Descripción de las pruebas	20
2.3.1.	Prueba 1: El pañuelo curvo	20
2.3.2.	Prueba 2: Limpiadores de latas	21
2.4.	Reglamento General	23
2.4.1.	Artículo 1. Organización y arbitraje	23
2.4.2.	Artículo 2. Participación	23
2.4.3.	Artículo 3. Características del robot	23
2.4.4.	Artículo 4. Preparación	24
2.4.5.	Artículo 5. Condiciones de seguridad	24
2.4.6.	Artículo 6. Formato de la competición	24
2.5.	Fechas de inscripción y condiciones	25
2.6.	Premios	26
3.	Descripción de los Robots participantes	27
3.1.	Gran Capitán	27
3.1.1.	Datos del Instituto	27
3.1.2.	Participantes	27
3.1.3.	Descripción del robot	28
3.2.	Los Olivos (I)	29
3.2.1.	Datos del Instituto	29
3.2.2.	Participantes	29
3.2.3.	Descripción del robot	29
3.3.	Los Olivos (II)	31
3.3.1.	Datos del Instituto	31
3.3.2.	Participantes	31
3.3.3.	Descripción del robot	31
3.4.	Palas Atenea	33
3.4.1.	Datos del Instituto	33
3.4.2.	Participantes	33
3.4.3.	Descripción del robot	33
3.5.	Numancia	35
3.5.1.	Datos del Instituto	35

3.5.2.	Participantes	35
3.5.3.	Descripción del robot	35
3.5.4.	Descripción del programa	35
3.6.	Sefarad (I)	37
3.6.1.	Datos del Instituto	37
3.6.2.	Participantes	37
3.6.3.	Descripción del robot	37
3.7.	Sefarad (II)	38
3.7.1.	Datos del Instituto	38
3.7.2.	Participantes	38
3.7.3.	Descripción del robot	38
3.8.	Octavio Paz (I)	39
3.8.1.	Datos del Instituto	39
3.8.2.	Participantes	39
3.8.3.	Descripción del robot	39
3.9.	Octavio Paz (II)	42
3.9.1.	Datos del Instituto	42
3.9.2.	Participantes	42
3.9.3.	Descripción del robot	42
3.10.	Gaspar Melchor de Jovellanos (I)	44
3.10.1.	Datos del Instituto	44
3.10.2.	Participantes	44
3.10.3.	Descripción del robot	44
3.11.	Gaspar Melchor de Jovellanos (II)	46
3.11.1.	Datos del Instituto	46
3.11.2.	Participantes	46
3.11.3.	Descripción del robot	46
3.12.	Miguel Hernandez (I)	48
3.12.1.	Datos del Colegio	48
3.12.2.	Participantes	48
3.12.3.	Descripción del robot	48
3.13.	Miguel Hernandez (II)	51
3.13.1.	Datos del Instituto	51

3.13.2. Participantes	51
3.13.3. Descripción del robot	51
3.14. Virgen de la Paz	53
3.14.1. Datos del Instituto	53
3.14.2. Participantes	53
3.14.3. Descripción del robot	53
3.14.4. Descripción del programa	54
3.15. Juan Carlos I	56
3.15.1. Datos del Instituto	56
3.15.2. Participantes	56
3.15.3. Descripción del robot	56
3.16. Isabel la Católica	57
3.16.1. Datos del Instituto	57
3.16.2. Participantes	57
3.16.3. Descripción del robot	57
3.17. Carmen Martín Gaité (I)	59
3.17.1. Datos del Instituto	59
3.17.2. Participantes	59
3.17.3. Descripción del robot	59
3.18. Carmen Martín Gaité (II)	60
3.18.1. Datos del Instituto	60
3.18.2. Participantes	60
3.18.3. Descripción del robot	60
3.19. Antonio de Nebrija (I)	61
3.19.1. Datos del Instituto	61
3.19.2. Participantes	61
3.19.3. Descripción del robot	61
3.20. Antonio de Nebrija (II)	62
3.20.1. Datos del Instituto	62
3.20.2. Participantes	62
3.20.3. Descripción del robot	62
3.21. Enrique Tierno Galvan (I)	63
3.21.1. Datos del Instituto	63

3.21.2. Participantes	63
3.21.3. Descripción del robot	63
3.22. Enrique Tierno Galvan (II)	64
3.22.1. Datos del Instituto	64
3.22.2. Participantes	64
3.22.3. Descripción del robot	64
3.23. Luis Vives	65
3.23.1. Datos del Instituto	65
3.23.2. Participantes	65
3.23.3. Descripción del robot	65

Índice de figuras

1.1.	Robot Cye equipado con bolsa recoge polvo	4
1.2.	Robots pioneer	5
1.3.	Robot Lego Mindstorm	8
1.4.	Partido de <i>small size</i> en Osaka 2005	10
1.5.	Proyección de partido simulado (Robocup 2005)	11
1.6.	Final de la Robocup 2005	14
1.7.	Captura de programa de Stage	16
1.8.	Interfaz gráfico de JDE	17
2.1.	Pista ejemplo para la prueba del pañuelo	21
3.1.	Robots Mario y Tamara	28
3.2.	Robot Cangrejillo	30
3.3.	Robot Willy	30
3.4.	Robot Hunter	32
3.5.	Robot Toro	32
3.6.	Robot Manolebot	34
3.7.	Robot Campeón	36
3.8.	Robot Respirit	40
3.9.	Robot Ocassionity	41
3.10.	Robot Game Over	43
3.11.	Robot No Fufona	43

3.12. Robot Zampalatas	44
3.13. Robot Trotalneas	45
3.14. Robot La Bestia	46
3.15. Robot Husmeador	47
3.16. Robot Scorpy	49
3.17. Robot No-Se	50
3.18. Robot Whatapanashi	52
3.19. Robot SVC-23	52
3.20. Robot atrapalatas	54
3.21. Robot Torete2	55
3.22. Robot Carpanta	56
3.23. Montaje de los robots Don Quijote y Sancho Panza	58

Capítulo 1

El Software Libre en la Robótica

El software es uno de los principales componentes de los robots. En particular, opinamos que es el principal problema para la extensión de la robótica en nuestra vida diaria. Uno de los factores que creemos que puede ayudar a paliar este problema es el fenómeno del software libre, que ha crecido de manera extraordinaria en el campo de la robótica. En este artículo analizamos la situación del software libre en tres subáreas de la robótica: industria, enseñanza e investigación.

Las herramientas libres han sido muy populares entre la comunidad docente e investigadora en el campo de la robótica. Incluso compañías cuyo principal modelo de negocio es la venta de robots, han creído conveniente compartir el software con el fin de promover comunidades alrededor de sus productos.

Todos estos casos muestran que el software libre puede actuar como un catalizador en la industria robótica, un sector aún en fase temprana de desarrollo, en el mundo investigador e industrial.

1.1. Introducción

El software libre ha alcanzado gran notoriedad gracias a proyectos muy difundidos como son el servidor Apache, el kernel de Linux o el paquete ofimático OpenOffice. Igualmente, disponemos

de múltiples herramientas de programación libres para varios lenguajes de programación como C Perl o Python. Pero más allá de esas aplicaciones, existe una amplia actividad en otros campos, quizá menos conocidos, pero también fuertemente influenciados por el software libre, sus modelos de desarrollo y herramientas. La robótica es claramente uno de estos campos.

Muchos de los kits robóticos que se venden actualmente incluyen algún tipo de software para controlar o interactuar con el robot. Estos programas puede adaptarse, mejorarse o incluso reemplazarse para añadir nuevas funciones que no fueron incluidas por el fabricante. Las comunidades que se forman alrededor de estos robots no comparten solo el interés por ellos, sino también mejoras, soluciones, código y nuevas ideas gracias a Internet. Los usuarios se han sentido atraídos por el modelo de desarrollo del software libre tanto por razones éticas [Stallman98] como practicas [Raymond98].

Como introducción a este libro sobre RoboCampeones 2005 nos ha parecido interesante describir la situación actual del software libre en el campo de la robótica. En particular, porque gran parte de los participantes han hecho uso de herramientas de software libre.

De esta forma, nuestro estudio ha dividido en tres partes el campo de la robótica: industria, educación e investigación. Para cada una hemos elegido algunos casos de estudio conocidos por los autores, donde el software libre ha sido la estrategia adoptada.

El resto del capítulo está organizado de la siguiente manera: en la siguiente sección analizaremos el estado de la robótica en la industria a través de tres ejemplos: el robot Cyb de Probotics, el robot Pioneer de Activmedia y el proyecto Orocós, financiado por la Comisión Europea. La tercera sección se centrará en la educación, donde encontramos el que es sin duda el kit robótico más extendido: el LEGO Mindstorms. La cuarta sección tratará sobre las iniciativas de investigación, mencionando la competición Robocup y las plataformas Player/Stage y JDE. Finalmente, se presentan una serie de conclusiones y algunas predicciones acerca de cuál creemos que será la evolución del fenómeno del software libre en el campo de la robótica.

1.2. Software libre en la industria robótica

Durante mucho tiempo se vienen utilizando robots en las aplicaciones industriales. Hoy en día no es raro encontrar robots en fábricas, soldando, transportando materiales, o en cualquier otra tarea repetitiva. En este mercado es difícil la entrada de nuevas ideas en él, y la cantidad de usuarios interesados en algo más que productos cerrados sigue siendo limitada.

A pesar de que las compañías que trabajan con robots caros todavía no han encontrado los alicientes suficientes para entrar a una dinámica dirigida por el uso de software libre, los nuevos mercados donde el número de robots vendidos es muy alto, se ha demostrado claramente que existen muchos buenos argumentos para impulsar a una compañía de robots a entrar en un modelo caracterizado por compartir el código fuente que hace funcionar a sus robots.

En esta sección veremos algunos ejemplos de estrategias de negocio sobre robots, poniendo atención en los robots Cyé y Pioneer. También hablaremos de la plataforma software Orocós, que pretende ser un estándar en las aplicaciones de la industria de la robótica.

1.2.1. El robot Cyé de Probotics

En los últimos años de la década de los 90, algunas compañías fabricantes de robots comenzaron a darse cuenta que había otras formas de gestionar el software que incluían con sus robots.

Un ejemplo es el robot Cyé¹ desarrollado y distribuido por Probotics Inc. Inicialmente esta compañía vendía su software *Map-N-Zap* independientemente de sus robots. Pronto se dieron cuenta que los robots no eran útiles sin ese software, así que decidieron incluirlo en el kit básico del robot. En enero del año 2000 finalmente decidieron distribuir el software bajo licencia GNU GPL (GNU General Public License). El anuncio de esta estrategia tenía como objetivo claro la comunidad de software libre²:

¹<http://personalrobots.com>

²el anuncio de la versión libre se puede encontrar en

“Esta decisión fue tomada en parte por la numerosa cantidad de peticiones desde la comunidad Linux de nuestro código fuente, y el hecho de que ellos, y desarrolladores de otras plataformas, tienen mucho que ofrecer a nuestra misión, que es hacer verdaderamente atractiva la tecnología de los robots.”



Figura 1.1: Robot Cye equipado con bolsa recoge polvo

1.2.2. El robot Pioneer robot de Activmedia

El robot Pioneer de Activmedia se vende como una herramienta de investigación, y se usa por una gran cantidad de universidades como principal plataforma de investigación en robótica móvil. En la figura 1.2 se puede ver unos equipados con un brazo robotizado, cámaras y sensores láser.

En el momento en que Activmedia comenzó a vender este robot existían otras alternativas en el mercado, las cuales tenían características similares y se había comprobado que eran lo

suficientemente buenas para fines investigadores. Actualmente Activmedia es conocido como el mayor vendedor de robots para fines investigadores.

Probablemente la diferencia determinante para esta evolución fue que Activmedia comenzó a distribuir su software de manera abierta mientras que las otras empresas eran más restrictivas. El software de control del robot, Aria³, era una nueva versión de su anterior plataforma de desarrollo Saphira, esta vez publicado bajo la licencia GPL. Una de las metas de este proyecto era ofrecer soporte a la plataforma anterior y a los clientes que estaban aún usándola. Aria ha sido también mejorada por contribuciones de usuarios, repercutiendo en la calidad y funcionamiento del sistema.



Figura 1.2: Robots pioneer

1.2.3. El proyecto Orocós

En la situación actual, en la que se pueden encontrar robots desde en plantas industriales hasta en oficinas, es importante tener un entorno común de desarrollo de aplicaciones para robots. Dichos entornos están presentes en otras tecnologías de software como el tratamiento de imágenes o la computación distribuida.

³<http://www.activrobots.com/SOFTWARE/aria.html>

El proyecto Orocos [Bruyninckx01] (Open RObot COntrol Software) presenta un entorno de desarrollo que ofrece funcionalidad genérica para los robots y sus aplicaciones. Este proyecto nació dentro de EURON (European Robotics Network) y fue patrocinado por la Comisión Europea (EC) en el año 2000. Actualmente el proyecto Orocos continúa con fondos del Centro de Tecnología Flanders Mechatronics, desarrollando el entorno de tiempo real y coordinando la integración de Orocos en máquinas industriales.

El objetivo principal del proyecto es desarrollar una plataforma software de control bajo una licencia libre, manteniendo la independencia sobre arquitecturas concretas, contribuir al desarrollo de interfaces de programación de arquitecturas de control y al desarrollo de material docente libre.

Además de sus raíces en el campo de la robótica, el entorno de desarrollo de aplicaciones en tiempo real de Orocos se ha centrado en el campo del control de maquinaria. Actualmente hay dos subproyectos principales dentro del marco de Orocos:

- **Servicios Abiertos de Control en Tiempo Real:** Este es un proyecto general, fuera del marco de la robótica, que desarrolla un kernel de tiempo real para todas las posibles arquitecturas de control.
- **Software Abierto de Control de Robots:** Este es un proyecto específico para robots, con la intención de desarrollar un entorno de desarrollo de aplicaciones, ofreciendo generación de movimientos e interpolación, cinemática y dinámica, algoritmos de control, estimación e identificación, etc.

Este proyecto puede ser considerado un hito en la industria robótica por su impacto en este campo: fue la primera vez que la Unión Europea aportaba fondos a un desarrollo de software libre en la industria; y segundo, porque ha sido el punto de partida para otros proyectos como Orca⁴, Ocean⁵, etc.

⁴<http://orca-robotics.sourceforge.net>

⁵http://www.fidia.it/english/research_ocean_fr.htm

1.3. Software libre en la enseñanza de la robótica

Además del uso práctico, en la industria, el software libre tradicionalmente ha sido muy empleado en la educación. En concreto, el uso del software libre es beneficioso para la enseñanza de ingenierías, como se ha reflejado ya [Barahona00]. Los cursos de robótica no son una excepción, como también han dejado claro [Kumar98] y [Ohara03].

En particular, el grupo de robótica de la URJC tiene experiencia en la enseñanza de los principios de la programación de control de robots a estudiantes de ingenierías informáticas. Además de buscar contenidos adecuados, es necesario elegir de manera adecuada el robot y las herramientas que deben usarse para su programación. En esta sección describiremos las herramientas que usamos para nuestros fines docentes, analizando las razones que hay detrás de nuestras decisiones.

1.3.1. LEGO Mindstorms

Los LEGO Mindstorms⁶ fueron originalmente concebidos como un producto para que los niños aprendieran, pero hoy en día es sin duda el kit para la construcción de robots más extendido en cualquier entorno docente. En la figura 1.3 se puede observar un robot realizado con piezas de Lego procedentes de kits Mindstorms que resuelve el famoso cubo de Rubik.

El éxito de este producto es, en parte, debido a la comunidad que se ha formado alrededor de él y que, por ejemplo, ha desarrollado por completo un nuevo sistema operativo (BrickOS[Lego]) y un lenguaje de programación (NQC [Baum99]), desplazando las herramientas cerradas proporcionadas por el fabricante. Los desarrollos de esta comunidad han sobrepasado los límites de los kits que vende LEGO, llevando Mindstorms a su posición privilegiada en el mercado.

⁶<http://www.legomindstorms.com>



Figura 1.3: Robot Lego Mindstorm

1.3.2. El lenguaje Not Quite C (NQC)

NQC [Baum99] son las siglas de Not Quite C, y es un lenguaje parecido a C que se puede usar para programar el ladrillo RCX de LEGO (del kit de Mindstorm). Es la alternativa textual sencilla al entorno visual que aporta Mindstorm.

NQC es software libre, publicado bajo la *Mozilla Public License (MPL)* pero, como usa el sistema operativo estándar desarrollado por LEGO, depende de un solución no libre.

1.3.3. El sistema operativo BrickOS

BrickOS⁷ es un sistema operativo embebido libre diseñado para el ladrillo de LEGO Mindstorms. Es una evolución de LegOS que fue principalmente diseñado por Markus Noga [Lego]. Comparado con el software estándar (el de LEGO), BrickOS ofrece extraordinarias mejoras de rendimiento y flexibilidad.

⁷<http://brickos.sourceforge.net>

1.4. Software libre en la investigación en robótica

El software libre ofrece a la comunidad científica la posibilidad de compartir soluciones a diferentes problemas comunes. Esto, por supuesto, es básico para la evolución de la ciencia; pero el software libre va más allá y hace que el proceso suceda de una manera asequible [Meyer03].

En esta sección hablaremos de cómo el software libre puede mejorar resultados de investigación en entornos competitivos, como es la Robocup. También veremos cómo los entornos de programación de software libre se usan para realizar tareas de programación de hardware heterogéneo usando plataformas abstractas de forma sencilla. Hay varias plataformas software y muchas de ellas son software libre, como por ejemplo Miro⁸, Marie⁹, Carmen¹⁰, Player/Stage¹¹ y JDE¹². Enfocaremos nuestra atención en las dos últimas. La primera se ha convertido en la plataforma de referencia en el mundo de la investigación y JDE es la plataforma desarrollada en nuestro grupo

1.4.1. La competición de la Robocup

Hemos decidido usar el caso de la competición de la Robocup [Kitano97], en el cual el grupo de robótica de la URJC lleva participando varios años, como ejemplo del uso del software libre en la investigación en robótica, por su popularidad y vistosidad y por haber sido utilizado como demostración en el intermedio de las pruebas de RoboCampeones 2005.

Robocup¹³ es el nombre “comercial” de una iniciativa internacional de investigación y divulgación. Su meta es fomentar

⁸<http://smart.informatik.uni-ulm.de/MIRO/content.html>

⁹<http://marie.sourceforge.net/>

¹⁰<http://www-2.cs.cmu.edu/~carmen/>

¹¹<http://playerstage.sourceforge.net>

¹²<http://gsync.escet.urjc.es/~jmplaza/software.html>

¹³<http://www.robocup.org>

la investigación en el campo de la inteligencia artificial y de la robótica, proporcionando un problema desafiante (principalmente jugar al fútbol), donde se pueden examinar e integrar una amplia gama de tecnologías.

La RoboCup está estructurada en varias categorías: los robots pequeños o F180 (180 mm de diámetro máximo) que usan una cámara cenital que envía imágenes a un PC que las analiza y envía las ordenes a los robots. Los robots medianos, donde los robots tienen que tener integrado todo su hardware, no hay visión, ni ordenador central. La liga de 4 patas donde el hardware es común, el robot Aibo de Sony y finalmente la liga de 2 patas (humanoides) que está todavía comenzando. Además existe una liga de robots simulados, una RoboCup Junior donde compiten alumnos de secundaria con robots Lego, y una liga de rescate.

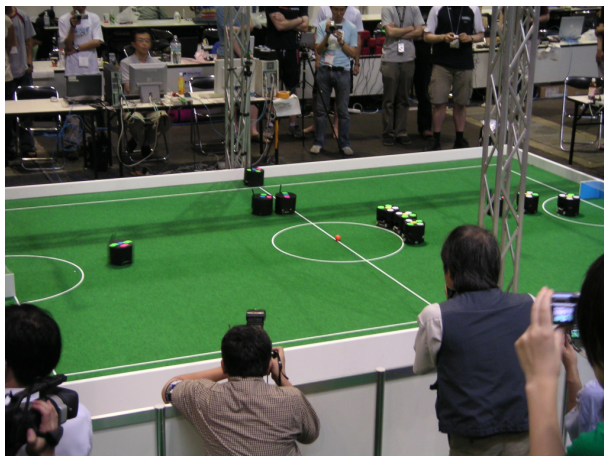


Figura 1.4: Partido de *small size* en Osaka 2005

Existen otras muchas competiciones internacionales entre robots, también nacionales, como el Hispabot (<http://www.hispabot.org>) para alumnos universitarios o el propio RoboCampeones para estudiantes de secundaria que describe este libro.

En el caso de la RoboCup se trata de una competición de carácter mundial y se celebra anualmente. Este año fue en Osaka (Japón) del 13 al 19 de Julio, el pasado año fue en Lisboa (Portugal)

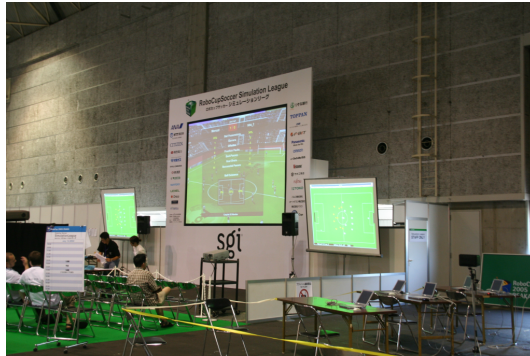


Figura 1.5: Proyección de partido simulado (Robocup 2005)

coincidiendo con el campeonato de fútbol entre humanos, y en el 2006 será en Alemania de nuevo coincidiendo con el campeonato de fútbol. Además existen unos eventos previos regionales: Open USA, Europeo, etc.

Alguno pensará que cuál es el objetivo de semejante ocurrencia. ¿Para qué queremos robots que jueguen al fútbol? Pues realmente para nada, es simplemente un dominio de prueba de las investigaciones en robótica y más concretamente en inteligencia artificial. Durante algún tiempo el problema estrella fue el ajedrez, sin embargo ya hace mucho que los ordenadores superan a los humanos sin problema. El ajedrez, aún siendo un dominio de búsqueda exponencial, es un dominio muy acotado, muy abstracto. El fútbol es un entorno mucho más realista: hay que tratar con un entorno muy dinámico, hay que lidiar con el mundo real, las luces, otros robots... Es decir, el objetivo no es el fútbol, es una excusa para promover el desarrollo de este campo.

Lo que realmente se “investiga” aquí es en software, en crear comportamientos en los robots, o lo que es lo mismo, se trata de escribir software, y no pocas líneas. Por ejemplo, el equipo ganador en la categoría de las cuatro patas en el año 2004 fue el equipo alemán, mejor dicho, uno de los equipos alemanes llamado el *German-Team* y su código son unas 300.000 líneas de código. El de nuestro equipo, mucho más modesto, tiene unas 50.000.

¿Cómo sé el número de líneas de código del rival? Pues porque una regla de esta competición es que es obligatorio publicar el código así como una descripción del equipo. En concreto el caso del *German-Team* la descripción y documentación de su código es realmente ejemplar. Les obliga su naturaleza germánica supongo, y su propia organización. Este equipo está formado por 4 universidades distintas que compiten individualmente en el campeonato europeo (se ha llamado tradicionalmente German Open, aunque el año que viene se llamará European) y que luego integran las mejores ideas en un equipo común. Eso les obliga a realizar un código muy modular y muy bien estructurado.

A la hora de publicar el código cada equipo puede elegir que tipo de licencia le otorga, aunque lo usual es publicarlo con licencias libres tipo GPL. Esto entronca completamente con la tradición científica y tecnológica de la publicación y compartición de resultados. De hecho el progreso de la ciencia se basa en la comprobación de los mismos. No vale de nada asegurar haber realizado un descubrimiento si éste no es reproducible, es decir, hay que publicar los resultados y la forma en que se han obtenido para que otros puedan verificar que realmente los resultados son que se indican, que son inventados o resultado de una mala realización del experimento (el famoso caso de la fusión fría).

En el caso del campo de la informática la única forma de poder hacer esta comprobación a mi modo de ver es disponiendo del código fuente. ¿Cómo se pueden comprobar las cosas? Alguien podría decir que con el código propietario también se puede hacer, te dan el binario, lo pones en los robots y compruebas que te ganan, luego es mejor. Desde luego, esa aproximación es válida, pero repito que el objetivo no es ganar al fútbol, es demostrar la valía de los algoritmos propuestos, hay que poder comprobar que realmente se está usando lo que se dice.

Aceptando que la motivación científica fuese suficiente, aún nos queda otro escollo: a primera vista cuadra mal lo de competición y software libre. Si se trata de una competición nadie quiere que los rivales conozcan el software, ni desde luego que lo usen...

En principio ese razonamiento es muy plausible, por lo que se ha añadido una regla que es que los equipos liberan su código *después* de terminada la competición. Es decir, ahora mismo podemos ver el código del equipo alemán del año pasado, no el que competirá este año.

Eso soluciona el problema de no jugar contra tus propios algoritmos, contra tu propio software. No arregla el que otros equipos obtengan ventaja de tu trabajo. Efectivamente es así, hay equipos que literalmente son una copia con pocas modificaciones de otros. Por ejemplo, las 300.000 líneas del equipo alemán han sido el punto de partida de muchos equipos. De hecho es algo que se pretende desde la organización, no se trata de una competición pura, se trata de usar una competición para ayudar al avance de la ciencia y la tecnología.

Obviamente, un equipo basado totalmente en el equipo alemán no puede reclamar mucho crédito, aunque gane a otro, en especial si este otro ha sido realmente programado por ellos mismos. ¿Dónde estaría el mérito, en ser capaces de copiarlo? En cualquier caso será del equipo alemán. La garantía de que el mérito se adjudique a los autores originales es la posibilidad del escrutinio público. Si sólo se utilizase el resultado del partido sería mucho más difícil saber si un equipo simplemente usa el código de otro.

Si bien el usar “todo” el código de otro equipo no está bien visto, si lo está el utilizar partes. En concreto partes completamente resueltas, o en las que un equipo determinado no está interesado pueden tomarse de otro equipo y obviamente atribuirles el crédito por esas partes. De eso se trata en el software libre.

Por ejemplo, nuestro equipo no está interesado en los *walking styles*, esto es en la locomoción del perro, trabajamos más en aspectos de localización, compartición de información, etc. Por ello usamos el código de locomoción del equipo francés, que sí está interesado en esos aspectos. Ellos por su parte utilizan nuestras rutinas de reconocimiento de la pelota, que todo hay que decirlo son bastante buenas, en especial ante cambios en la iluminación ambiente.

En la liga de 4 patas (o de Aibo) como ya he comentado lo más relevante es el software, pues el hardware es igual en todos los equipos, pero lo es también en muchas otras ligas, en especial en la de simulación donde todo es software. En este caso los participantes tienen que programar 11 jugadores que se comunican mediante un protocolo estándar con un servidor que realiza la simulación del partido. El servidor les envía información de los elementos que “ve” el jugador (bola, líneas del suelo, porterías, jugadores, contrarios, etc.), con ruido (a mayor distancia mayor error); y el jugador tiene que enviar comandos (giro, avance, chut, mensaje a los compañeros, etc.).



Figura 1.6: Final de la Robocup 2005

En la liga de simulación se ha llegado a convertir código escrito por los participantes en código común. En concreto el cálculo de localización, es decir, el que permite a un jugador conocer su posición en el campo a partir de la información visual, ha sido incorporado a la distribución básica. Se trata de unos cálculos relativamente complicados, pero que bien formulados no aportan nada novedoso. En vez de re-escribir continuamente ese código se ha elegido una versión exhaustivamente probada y se ha liberado a los programadores de esa tarea.

Los programadores del código reutilizado pueden enorgullecerse de su trabajo y citar a los equipos que lo utilizan como parte de ese mérito. De nuevo es equivalente a lo que se hace en el campo científico, donde la métrica que se considera mejor actualmente es el número de otros investigadores que referencian los trabajos de un investigador (frente al número de artículos publicados que se usaba tradicionalmente). A la vez, los programadores interesados en otros aspectos (colaboración p.e.) pueden centrarse en su campo y olvidar aspectos de bajo nivel.

Además, los que trabajan en otros campos ahora no pueden “excusarse” si el rendimiento de su equipo no es bueno, en que ellos no trabajan en esos campos de bajo nivel. Está resuelto.

Esta misma idea es la que en el fondo anima a muchas empresas a liberar su código incluso siendo líderes de su mercado: por una parte consiguen aún más visibilidad, para observadores externos “se pueden permitir el lujo de liberar su software”; y por otra se aseguran de liderar el mercado, pues muy probablemente muchas otras empresas usarán ese software de que son creadores y que por tanto conocen mejor que nadie. Es decir, están liderando su mercado desde el punto de vista de los clientes y de los competidores. Los competidores en muchos casos no pueden hacer más que seguir al líder.

En resumen, puede haber competición, en el sentido literal de la palabra, con software libre. El líder no pierde nada por liberar software, incluso gana porque marca tendencias que él lidera, además de conseguir mucha visibilidad.

1.4.2. La plataforma Player/Stage

Un componente importante del desarrollo con robots es la arquitectura software. Existen múltiples robots de diversos fabricantes que tienen hardware totalmente distinto y sus propios entornos del desarrollo. Con el fin de facilitar esta tarea han aparecido plataformas software que abstraen los detalles de más bajo nivel, proporcionando un API común para programar diferentes robots. El desarrollo de esta clase de herramientas no es la meta

principal de la investigación, no obstante pueden mejorar la calidad y el coste de los resultados. El software libre presenta también algunas ventajas aquí. Compartiendo sus herramientas, diferentes grupos introducen cambios pequeños en el sistema completo, beneficiándose del trabajo hecho por el resto de la comunidad. Como un ejemplo tenemos el proyecto Player/Stage [Gerkey03].



Figura 1.7: Captura de programa de Stage

Desde el principio el desarrollo ha estado abierto a cualquier contribución, ya que proporcionaban su software bajo licencia GPL. Los autores originales sabían que los usuarios de su software eran también desarrolladores que podrían contribuir con su trabajo para conseguir un producto mejor. Debido a esto, el proyecto fue utilizado por más investigadores por todo el mundo, construyendo una gran comunidad. El soporte para diferentes modelos de robots y múltiples lenguajes de programación ha convertido a Player/Stage en la plataforma estándar *de facto* para desarrollos de investigación.

La difusión de Player/Stage ha aumentado de manera que se ha convertido en la plataforma de referencia para muchos equipos de investigación. Por lo tanto, aparecen nuevas herramientas, como un visualizador para Player y arquitecturas de control construidas sobre el sistema completo.

1.4.3. JDE

Sobre las plataformas citadas anteriormente (Aria y Player/Stage), se construyen arquitecturas de más alto nivel.

1.4. SOFTWARE LIBRE EN LA INVESTIGACIÓN EN ROBÓTICA 17

Lo que se denominan arquitecturas *cognitivas*, que proporcionan las capacidades del robot. En el caso del grupo de robótica de la URJC se ha desarrollado la arquitectura JDE, iniciales de Jerarquía Dinámica de Esquemas, que también se distribuye como software libre.

JDE es una aproximación jerárquica a una arquitectura robótica de control, inspirada en la etología y basada en la activación selectiva de esquemas. Es decir, genera las jerarquías dinámicas que gobiernan el comportamiento del robot.

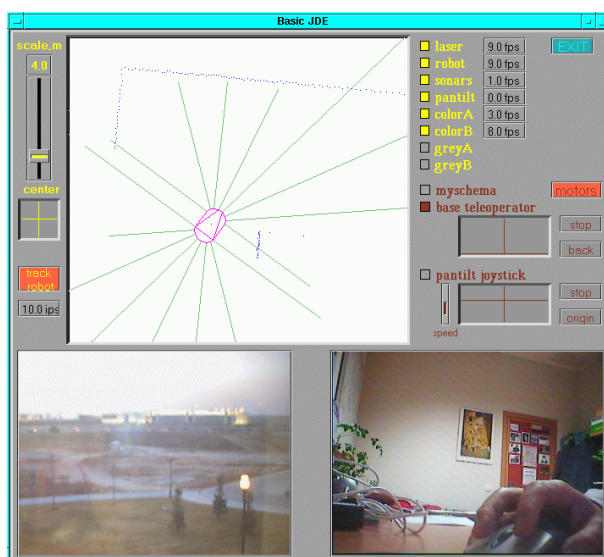


Figura 1.8: Interfaz gráfico de JDE

La implementación actual de JDE consiste en una infraestructura de software y una colección de esquemas específicos de comportamiento, que contienen técnicas particulares de control o de percepción. La infraestructura incluye los esqueletos de esquemas, los esquemas del servicio según el robots soportado, los mecanismos del software para la selección distribuida de la acción y la activación selectiva de esquemas, así como su ajuste. Además, algunas bibliotecas también se han escrito para manipulación de rejillas y control borroso, con el fin de hacer más fácil la generación

del comportamiento. Todo el código se ha desarrollado en el lenguaje C para máquinas GNU/Linux, y está liberado bajo licencia GPL.

1.5. Conclusiones

La robótica es un sector emergente de la industria de las computadoras que todavía se encuentra en etapas tempranas de investigación y uso. En este trabajo hemos presentado algunos ejemplos de porqué el software libre puede actuar como catalizador para el desarrollo de esta industria.

Hay muchas ventajas que se pueden derivar del uso de modelos y estructuras de desarrollo de software libre en el campo de la robótica, de las que, tanto usuarios como empresas, están comenzando a sacar partido. De hecho, algunas empresas han construido su modelo del negocio alrededor del uso del software del libre, como hemos visto en este trabajo.

Hemos mostrado ejemplos donde el uso del software libre ha sido el factor clave para hacer de un producto la referencia de un mercado completo, tanto en la investigación como en la educación. En ambos casos, el software libre es más fácil de adaptar a las necesidades particulares de usuarios y reutilizarlo en otras situaciones. Las comunidades del software libre han demostrado que herramientas libres pueden llegar a ser más populares incluso que las proporcionadas por los fabricantes originales.

Finalmente hemos presentado un ejemplo donde el uso del software libre ha permitido alcanzar progresos más rápidos y resultados mejores. Compartiendo las soluciones ya conocidas, los desarrolladores nuevos pueden centrarse en mejorar las partes del sistema que no están maduras.

Capítulo 2

Reglamento

2.1. Objetivos

El objetivo del concurso Robocampeones es fomentar la capacidad creativa y de innovación de los estudiantes, en especial en la rama científicotécnica.

El concurso se organiza en dos pruebas de competición entre robots autónomos.

Las pruebas persiguen potenciar el conocimiento y uso de las nuevas tecnologías de forma estimulante y atractiva para los estudiantes.

2.2. Participantes

El concurso va dirigido a estudiantes de secundaria y bachillerato de centros públicos. Los equipos formados en parte o en su totalidad por alumnos de cursos inferiores (primaria p.e.) son bienvenidos. No pueden formar parte alumnos de cursos superiores (universidad, titulados, etc.)

La participación de equipos de centros privados y concertados es también bienvenida pero no podrán optar a la financiación pública si existiese, aunque sí a los premios que aporten los patrocinadores.

Cada centro podrá presentar un máximo de dos grupos de alumnos, siempre que cada uno de ellos esté bajo la responsabilidad de un profesor diferente. Cada grupo de alumnos estará formado por un máximo de cuatro alumnos.

Durante la celebración de las pruebas se realizarán fotografías y grabaciones de vídeo con fines promocionales (prensa, Internet, etc.). Los profesores responsables de cada grupo son los encargados de obtener los permisos de los padres o tutores de los alumnos. La inscripción en el concurso presupone el permiso de los padres o tutores, tanto para la participación como para la grabación.

2.3. Descripción de las pruebas

Las pruebas del concurso en la edición del 2005 son: *El pañuelo Curvo* y *Limpiadores de Latas*. A continuación se da su descripción y reglamento.

2.3.1. Prueba 1: El pañuelo curvo

La primera prueba consiste en una versión simplificada del tradicional juego del pañuelo. En ella dos robots, que parten de extremos opuestos de una línea curva, deben seguir dicha línea para atrapar una lata de refresco que se encuentra sobre la misma, y llevarla a su lugar de origen. Se descalificará a los equipos que aprovechen un posible conocimiento de la pista (o que evidentemente ignoren la línea).

El ganador será el que consiga llevar la lata a su lugar de partida. El robot deberá detenerse una vez alcanzada la posición inicial (la lata deberá cruzar la línea de partida). Si los dos robots quedan bloqueados se considera empate y se repetirá la prueba, igualmente si un robot consigue coger la lata pero no dejarla en la posición de salida se considerará empate. Si se repitiese el empate por segunda vez se repetiría de nuevo, y de producirse un tercer el empate el árbitro decidirá el ganador.

La pista de competición consistirá en una línea de longitud desconocida, de trazado también desconocido (igual para ambos

participantes) de color negro sobre fondo blanco, y 1,5 centímetros de ancho. En los extremos y en el centro de la misma se colocarán líneas perpendiculares de igual anchura y 20 centímetros de largo que marcarán los lugares de partida de los robots y la posición de la lata. Los robots podrán tener cualquier tamaño, pero en la salida todo el robot deberá estar por detrás de la línea de partida.

La lata estará pintada de blanco y lastrada con arena de forma que el peso total de la misma sea de aproximadamente 150gr. La figura 2.1 ilustra una posible configuración de la prueba.

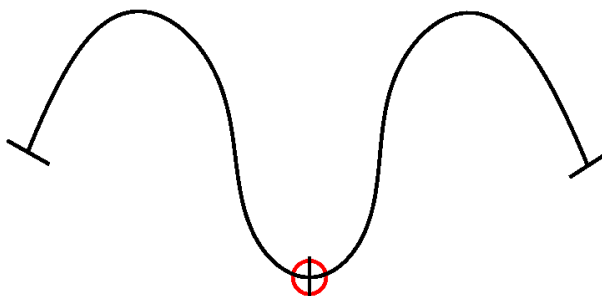


Figura 2.1: Pista ejemplo para la prueba del pañuelo

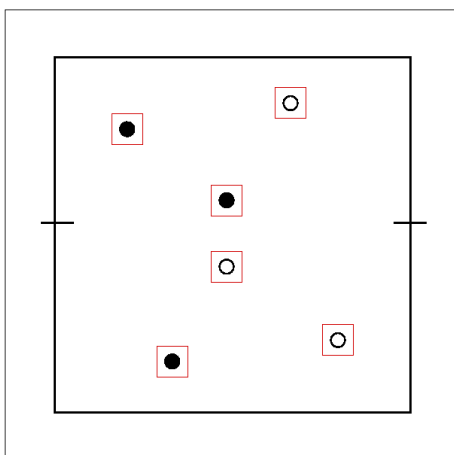
2.3.2. Prueba 2: Limpiadores de latas

La segunda prueba consiste en la expulsión de 3 latas de refresco, forradas de cartulina negra o blanca, de un área rectangular de dimensiones conocidas. En la prueba participarán simultáneamente 2 robots, cada uno de los cuales deberá expulsar las latas de un color (uno las blancas y el otro las negras). Al comienzo de la prueba se sorteará el color de las latas a extraer.

El ganador será el robot que consiga sacar en primer lugar las 3 latas de su color fuera del recinto. Existirá un tiempo límite de 3 minutos, pasado el cual si ninguno de los robots ha sacado las latas ganará el que más latas haya sacado. Si han sacado el mismo número se considerará ganador al que haya sacado antes la última lata expulsada de su color.

La pista de la prueba será un cuadrado de 2x2 metros, dentro del cual se delimita una zona de 1,5x1,5 metros mediante una línea negra de 1,5 centímetros de grosor. Las latas tienen que salir completamente del cuadrado para considerarse expulsada. Los robots pueden salir y entrar del cuadrado cuantas veces quieran, pero no podrán ser manipulados durante la duración de la prueba (p.e. recolocados)

La Figura 2.3.2 muestra un esquema de una posible configuración:



La disposición de las latas será desconocida a priori y puede variar entre la ejecución de las distintas fases. Se garantizará que la colocación de las latas será simétrica para que los equipos compitan en igualdad de condiciones. Los dos robots parten simultáneamente de lugares opuestos de la pista. Los lugares de partida de los robots estarán marcados con las líneas perpendiculares al borde, como muestra la figura 2.3.2.

Un robot no puede mover las latas que tiene que sacar su oponente. Alrededor de cada lata habrá un cuadrado que marcará el máximo desplazamiento que un robot puede hacer de las latas de su oponente. Si un robot saca alguna lata de su oponente de ese cuadrado perderá esa prueba. El cuadrado estará realizado con una línea fina, que se intentará que sea invisible para los sensores.

2.4. Reglamento General

2.4.1. Artículo 1. Organización y arbitraje

En cada prueba existirá un árbitro, miembro del Grupo de Robótica de la Universidad Rey Juan Carlos, que se encargará de garantizar el correcto desarrollo de la misma, que velará por el cumplimiento de las normas, de su interpretación y que resolverá cualquier situación no contemplada en las bases del concurso. Las decisiones de los jueces serán inapelables.

2.4.2. Artículo 2. Participación

Cada centro puede presentar un máximo de 2 grupos de alumnos, cada uno de ellos bajo la responsabilidad de un profesor diferente. Cada grupo de alumnos estará formado por un máximo de 4 alumnos.

Todos los miembros del grupo, alumnos y profesor, deberán estar presentes durante toda la prueba en la que se presente. La no asistencia de un participante a las pruebas en las que está inscrito significará la no consideración como parte del grupo. Cada grupo podrá presentarse a una o varias pruebas con el mismo o con uno diferente para cada prueba.

2.4.3. Artículo 3. Características del robot

Los proyectos se construirán con piezas de LEGO y el controlador de LEGO. No se podrán utilizar piezas no oficiales de LEGO, aunque sí elementos para la decoración del robot o la sujeción de las piezas (cartulinas, gomas elásticas, etc.)

Los robots deberán ser autónomos y tanto el controlador como el resto de elementos serán únicamente los disponibles en la mencionada dotación.

2.4.4. Artículo 4. Preparación

Antes de comenzar la primera prueba, los participantes dispondrán, a la vez, de una hora para probar los robots en el escenario, ajustarlos las condiciones de luz, etc.

Entre cada prueba habrá un periodo de media hora para que los participantes puedan modificar el software y hardware de los robots, fuera de este periodo los robots no se podrán manipular.

En las dos pruebas competitivas en ningún caso se podrá controlar el robot desde el exterior (con un mando a distancia, puerto IR, etc).

2.4.5. Artículo 5. Condiciones de seguridad

Los sistemas construidos no serán peligrosos ni molestos para los participantes, ni otros asistentes al concurso. El jurado de las diferentes pruebas podrá eliminar por este motivo a cualquier participante en el concurso.

Todos los participantes mantendrán un espíritu de convivencia y respeto a los proyectos que se presenten.

2.4.6. Artículo 6. Formato de la competición

El formato de la competición dependerá del número de participantes inscritos:

<i>Participantes</i>	<i>Método</i>
Hasta 5	Todos contra todos (1 a 20 competiciones).
De 6 a 10	Por sorteo se dividen en 2 grupos, dentro de cada grupo se enfrentan todos contra todos y el mejor de cada grupo pasa a la final.
De 11 en adelante	Por sorteo se dividen en grupos que se enfrentan entre si, clasificándose para una ronda final el mejor de cada grupo.

Cuadro 2.1: Formato de la competición

2.5. Fechas de inscripción y condiciones

Se establecen las siguientes fechas y condiciones para poder participar:

La inscripción, o intención de participar, se realizará entre el 10 y el 30 de Abril 2005. Se deberá realizar una pre-inscripción por cada grupo indicando el nombre del profesor responsable y el nombre del instituto así como las pruebas en las que espera participar. Antes del 10 de Mayo de 2005 se entregará una *ficha de participación* que confirmará la participación. Dicha ficha se podrá descargar en la página web de Robocampeones (<http://www.robocampeones.com>) e incluirá la siguiente información y será enviada en formato electrónico por cada robot o sistema que presenta al concurso:

- Nombre del grupo.
- Nombre del instituto al que pertenece.
- Nombres y apellidos de los alumnos participantes y el profesor responsable.
- Fotografía del robot.
- Breve descripción del mismo.

Con la ficha de participación se realizará una publicación resumen del evento, la participación en el concurso implica la aceptación de aparición de los datos de la ficha de participación en dicha publicación.

El concurso se celebrará el día 12 de Mayo de 2005, jueves, de 9:00 a 14:00 conforme a un programa de actividades que se hará público con antelación en la página web del concurso.

El concurso tendrá lugar en la Escuela Superior de Ciencias Experimentales y Tecnología de la Universidad Rey Juan Carlos, sita en el campus de Móstoles. Accesos: Metro-Sur: parada Universidad Rey Juan Carlos, RENFE: estación Móstoles-El Soto, Autobuses: 522 con salida en Príncipe Pío.

La organización pondrá a disposición de los participantes un conjunto de monitores TFT, teclados, ratones, (NO CPUs), mesas

y sillas, pero no puede garantizar que sean suficientes para todos los participantes (dependerá de las inscripciones). Caso de no ser suficientes se arbitrará un mecanismo de turnos para su uso entre los participantes.

Los participantes se encargarán del traslado de dichos materiales al lugar de celebración del concurso así como de su retirada.

Los participantes deberán cuidar y respetar las instalaciones de la sede del concurso.

2.6. Premios

El centro ganador en cada prueba recibirá una dotación de 1000 euro. en productos LEGO donado por PRODEL S.A.

Los alumnos de los equipos ganadores recibirán un kit LEGO para construir sus propios robots.

El centro ganador y el finalista en cada prueba recibirá un trofeo y un diploma acreditativo, así como los componentes del equipo.

Todos los participantes recibirán un diploma por su participación.

Premios sorpresa y merchandising para los participantes y asistentes.

Capítulo 3

Descripción de los Robots participantes

3.1. Gran Capitán

3.1.1. Datos del Instituto

IES Gran Capitán
Pseo. de los Melancólicos, 51
28019 - Madrid
913659190

3.1.2. Participantes

Profesor Responsable: Enrique Suárez y Salvador Martín
Alumno 1: Alberto Benítez Salamanca
Alumno 2: Álvaro Fernández Iñigo
Alumno 3: Daniel Hernández Álvarez
Alumno 4: Víctor Izquierdo López
Alumno 5: Javier Jiménez Victorio
Alumno 6: Tamara Lerena Lara
Alumno 7: Víctor Montero Sánchez

Alumno 8: Fernando Muñoz Gallardo

Alumno 9: Germán Muñoz Rico

Alumno 10: Mario Robledo Pérez

3.1.3. Descripción del robot

Lo primero que hicieron los alumnos para resolver y afrontar la prueba fue descomponerla en problemas más pequeños, y así más fáciles de resolver. Estos fracciones del programa son:

- 1.- Sobrepasar la línea transversal al recorrido
- 2.- Seguir la línea negra con dos sensores de luz
- 3.- Detectar la lata
- 4.- Coger la lata con la pinza
- 5.- Girar y dar la vuelta al robot
- 6.- Seguir la línea negra en el retorno
- 7.- Parar el coche una vez que detecta la línea del final del recorrido.

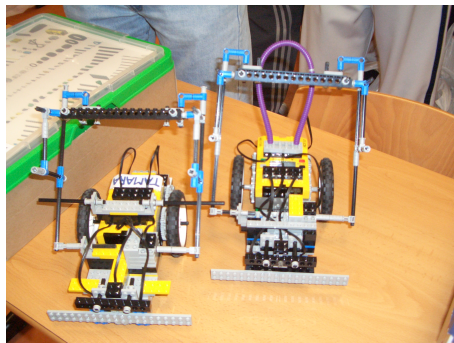


Figura 3.1: Robots Mario y Tamara

3.2. Los Olivos (I)

3.2.1. Datos del Instituto

SIES Los Olivos
C/Olivar, 14
28891 - Velilla de San Antonio
916608987

3.2.2. Participantes

Profesor Responsable: Angélica Berzal Izquierdo y Jesús Ramiro López

Alumno 1: Rubén González Martín
Alumno 2: David Costa Pradillo
Alumno 3: Bartolomé Matejek Sadlik
Alumno 4: Ana Ramona Turcán

3.2.3. Descripción del robot

La principal característica de *Cangrejillo* es su tenaza que permite coger la lata una vez que ésta ha sido detectada. Para esa detección se cuenta con un sensor de contacto en la parte frontal del robot al que se le ha añadido un parachoques para aumentar su sensibilidad. Por otro lado, dispone de otros dos sensores de luz; el primero de ellos permite seguir la línea negra y el segundo se emplea para detener al robot una vez que ha terminado la prueba.

Willy está diseñado para sacar latas de una superficie limitada por líneas de color negro. Para ello consta, por un lado, de un sensor de luz que hace que el robot, una vez que está dentro del cuadro de juego, nunca se salga del mismo. Esto es así porque cuando llega a unos de los lados, da marcha atrás y gira, buscando otra lata. Cuando encuentra una, la cual es detectada por un sensor de contacto situado en el frontal del robot, analiza su color, mediante otro sensor de luz situado igualmente en la parte delantera; y así dependiendo del color que sea, la saca hacia afuera o retrocede.

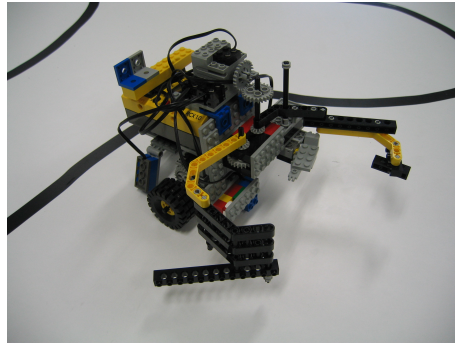


Figura 3.2: Robot Cangrejillo

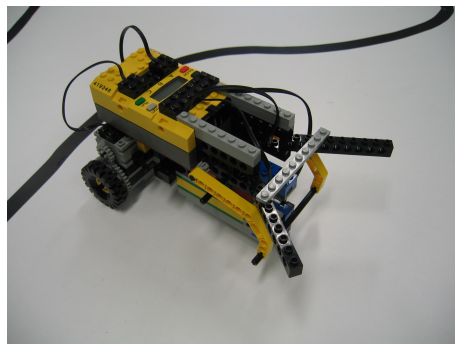


Figura 3.3: Robot Willy

3.3. Los Olivos (II)

3.3.1. Datos del Instituto

SIES Los Olivos
C/Olivar, 14
28891 - Velilla de San Antonio
916608987

3.3.2. Participantes

Profesor Responsable: Angélica Berzal Izquierdo y Jesús Ramiro López

Alumno 1: Daniel Barco Rodríguez

Alumno 2: Pablo Esteban Martín

Alumno 3: Álvaro Morales Flores

Alumno 4: Ángel Belinchón Nahón

3.3.3. Descripción del robot

Hunter está diseñado para realizar la prueba del pañuelo curvo. El robot utiliza un lazo para capturar la lata. Este lazo se activa mediante un motor que se pone en funcionamiento cuando el robot ha detectado la lata con un sensor de contacto colocado en la parte delantera del robot. Para llegar a esa lata es necesario seguir una línea negra, para ello se vale de un sensor de luz. Además dispone de otro sensor de luz que detecta cuando se ha llegado de nuevo al punto de partida y así indicar que el robot ha terminado la prueba.

Toro es un robot diseñado para echar latas fuera de un cuadro gracias a sus *cuernos* que ayudan a detectar la lata, mediante un sensor de contacto situado en la parte frontal del robot. Para saber de qué color es la lata tiene un sensor de luz en el medio de los *cuernos*. En función de ese color hará que el robot sigue hacia delante y expulse la lata o dé marcha atrás. Para que el robot nunca se salga del cuadro, tiene otro sensor de luz que hará que el robot retroceda cuando llegue a un extremo del cuadro.

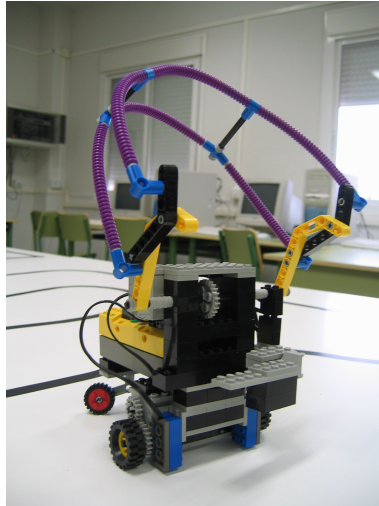


Figura 3.4: Robot Hunter

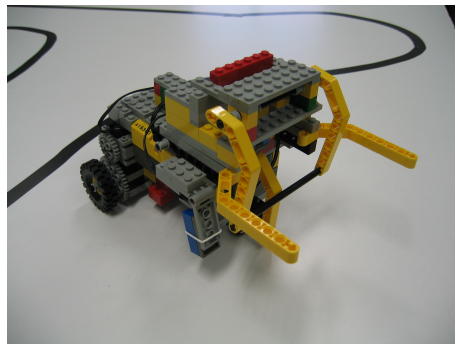


Figura 3.5: Robot Toro

3.4. Palas Atenea

3.4.1. Datos del Instituto

IES Palas Atenea
C/Arrope, s/n
28850 - Torrejón de Ardoz
916772915

3.4.2. Participantes

Profesor Responsable: Julio Sánchez Olías
Alumno 1: Juan Carlos González Rosillo
Alumno 2: Juan Carlos Ramírez Luque
Alumno 3: Ángela Solís Jara
Alumno 4: Mario Torres de Mesa

3.4.3. Descripción del robot

Manolebot se desplaza mediante dos motores, cada uno de ellos actúa sobre dos ruedas motrices mediante engranajes. De esta forma, dispone de cuatro ruedas motrices. El chasis está diseñado para que se pueda cambiar la relación de transmisión del robot sin necesidad de desmontarlo. Para rastrear la línea negra, dispone de dos sensores de luz montados en un soporte que permite el deslizamiento horizontal de los mismos. Así se puede variar la separación entre los sensores, en función del ancho de la línea a seguir. El sistema de agarre de la lata está formado por una pinza, accionada por un tercer motor. Para evitar que la pinza choque contra la lata en caso de que esta esté situada en una curva de la pista, la pinza se abre totalmente hasta que se detecta la lata. Para detectar la lata, se usa un sensor de contacto, accionado mediante un parachoques suficientemente ancho como para detectar la lata cualquiera que sea su posición respecto a la pista.

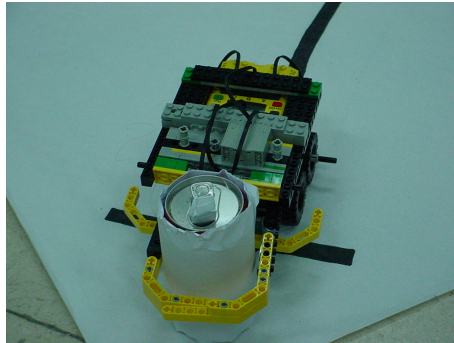


Figura 3.6: Robot Manolebot

3.5. Numancia

3.5.1. Datos del Instituto

IES Numancia
C/Marismas, 14
28038 - Madrid
915015001

3.5.2. Participantes

Profesor Responsable: César Prestel
Alumno 1: Víctor Calvo
Alumno 2: Carlos López
Alumno 3: Mikel García

3.5.3. Descripción del robot

El robot *Campeón* dispone de un lazo activado por un motor, que le permite capturar la lata, y de dos sensores de luz para detectar la presencia de la línea negra que debe seguir, además de un sensor de contacto con un parachoques para detectar la lata cuando choque con ella. Se desplaza mediante ruedas activadas por dos motores.

3.5.4. Descripción del programa

En primer lugar, el programa obligaba al robot a desplazarse en línea recta durante 1,5 segundos. Después, si el sensor de contacto no había detectado la lata, el robot se movía en línea recta mientras los sensores de luz permanecían activados. En caso de encontrarse con la línea negra, el robot debía retroceder durante 0,6 segundos a velocidad 1, y después debía girar en sentido contrario para corregir su trayectoria, durante otros 0,2 segundos. Posteriormente, seguía su marcha normalmente. Esta secuencia se repetía indefinidamente, mientras el detector de contacto no estuviese activado, mediante los saltos correspondientes.

El programa incluía también las instrucciones que el robot debía seguir en caso de detectar la lata. En este caso, el robot debía detenerse y capturar la lata con el lazo, activado por un motor. Después, el robot regresaba marcha atrás, empleando el mismo sistema para corregir su trayectoria que en la primera parte del programa. Esta secuencia se repetía indefinidamente mediante los saltos correspondientes. Finalmente, al detectar ambos sensores la presencia de la raya negra transversal, el robot seguía moviéndose marcha atrás durante 3 segundos, para asegurarse de que la lata pasaba de esta raya, y después el robot se detenía.

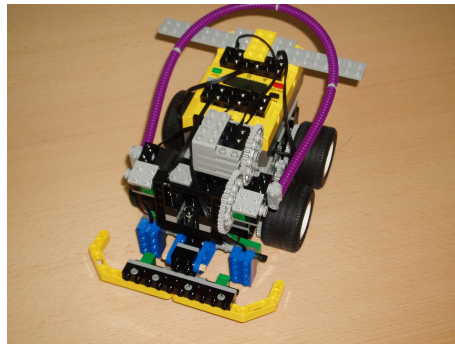


Figura 3.7: Robot Campeón

3.6. Sefarad (I)

3.6.1. Datos del Instituto

IES Sefarad
Cno. del Molino, 3
28943 - Fuenlabrada
914869492

3.6.2. Participantes

Profesor Responsable: José González Santamaría
Alumno 1: Héctor Florín Navarro
Alumno 2: Sergio Jiménez López
Alumno 3: Natalia Vega Piris
Alumno 4: José Manuel Vicente Moreno

3.6.3. Descripción del robot

Toro es el robot diseñado para la prueba de atrapar latas, se trata de un vehículo de exploración y localización de latas en un entorno delimitado.

3.7. Sefarad (II)

3.7.1. Datos del Instituto

IES Sefarad
Cno. del Molino, 3
28943 - Fuenlabrada
914869492

3.7.2. Participantes

Profesor Responsable: Claudia Sandra Villa López
Alumno 1: Fernando García Henández
Alumno 2: Alejandro López Alonso
Alumno 3: Mario Carriches Quevedo
Alumno 4: Christian Salso Pérez

3.7.3. Descripción del robot

Velociraptor es el robot diseñado por el IES. Sefarad para la prueba del pañuelo curvo de Robocampeones 2005.

3.8. Octavio Paz (I)

3.8.1. Datos del Instituto

IES Octavio Paz
C/Beatriz Galindo, 1
28914 - Leganés
916886444

3.8.2. Participantes

Profesor Responsable: Rafael Morales
Alumno 1: Andres Felipe Quintero
Alumno 2: Javier Lombal
Alumno 3: Aitor González
Alumno 4: Rodrigo Castañedo

3.8.3. Descripción del robot

El robot *Respirit* dispone de dos sensores de luz para el rastreo de la línea negra y dos sensores de tacto en paralelo para detectar la lata. Dos ruedas acopladas a dos motores y dos puntos de apoyo aseguran la estabilidad del robot. Para ganar velocidad inicialmente se acopló un sensor de tacto a un tercer sensor de luz para disponer de tres sensores de luz que guiasen al robot y un sensor de tacto en paralelo para la detección de la lata. Sin embargo, las pruebas sobre circuitos con curvas muy cerradas impedían asegurar un recorrido seguro por lo que se optó por una estrategia más conservadora buscando un equilibrio entre velocidad y seguridad. Además se redujo la velocidad del robot con engranajes reductores para aumentar su fiabilidad. Una vez detectada la lata se activa un tercer motor que despliega una cesta para capturarla rápidamente. Un meritorio tercer puesto es un gran triunfo teniendo en cuenta la estrategia conservadora desplegada.

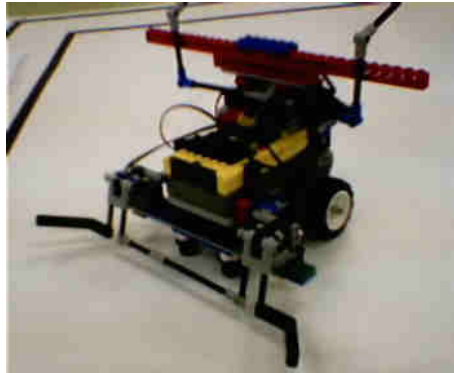


Figura 3.8: Robot Respirit

El robot *Ocassionity* exprime las posibilidades del RCX acoplando cuatro sensores de luz y dos sensores de tacto. El chasis esta construido sobre dos ruedas motrices acopladas con engranajes para reducir ligeramente la velocidad del robot y dos puntos de apoyo. Dos garras permiten rodear la lata y mediante dos sensores de luz en los extremos se detecta el color de las mismas. En caso de golpear una lata con las garras se han acoplado en paralelo a los sensores de luz unos sensores de tacto (en cada sensor de luz) que corrigen la trayectoria del robot para situar la lata dentro de las "garras" evitando tirarlas y detectar a la vez su color. El programa determina que sensor ha sido activado, el sensor de luz o el de tacto, acoplados al mismo puerto (debemos aclarar que un sensor de tacto acoplado a un puerto donde debería haber un sensor de luz marcará un nivel de luz de 100). Los ensayos sobre un rastreo de superficie le hacían muy lento por lo que se optó por un navegación aleatoria. Un ÚNICO programa determina que color deberá evitar el robot y por tanto que latas deberá sacar del recinto.

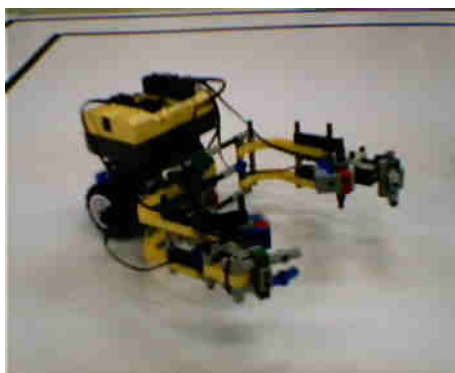


Figura 3.9: Robot Ocassionity

3.9. Octavio Paz (II)

3.9.1. Datos del Instituto

IES Octavio Paz
C/Beatriz Galindo, 1
28914 - Leganés
916886444

3.9.2. Participantes

Profesor Responsable: Gregorio Romo
Alumno 1: Lourdes Asensio
Alumno 2: Javier Peralta
Alumno 3: Carlos Antonio Azañon
Alumno 4: Daniel Beato

3.9.3. Descripción del robot

El robot *Game Over* es muy similar a su hermano gemelo *Respirit*, dispone de dos sensores de luz para el seguimiento de la línea y un sensor de tacto para la detección de la lata. Dos ruedas acopladas a dos motores con engranajes reductores y un punto de apoyo permiten al robot desplazarse sin dificultad. Para capturar la lata se acopló al robot una pinza construida con un tornillo sinfín y dos engranajes que agarraban la lata e impedían que llegará a soltarse.

El robot *No Fufona* dispone de ruedas de oruga para desplazarse a una velocidad aceptable. Incorpora dos sensores de luz: uno trasero para detectar los límites del recinto y uno frontal para detectar el color de la lata. Dos brazos en los extremos colocan a la lata delante del sensor para identificar su color y dos ruedas en estos brazos ayudan a su colocación. Un ÚNICO programa determina que color deberá evitar el robot y por tanto que latas deberá sacar del recinto. Un digno tercer puesto compensa la eliminación en semifinales de su hermano mayor *Ocassionity*.



Figura 3.10: Robot Game Over



Figura 3.11: Robot No Fufona

3.10. Gaspar Melchor de Jovellanos (I)

3.10.1. Datos del Instituto

IES Gaspar Melchor de Jovellanos
C/Móstoles 64, Fuenlabrada
28942
916971565

3.10.2. Participantes

Profesor Responsable: José Alberto Espejo Redondo
Alumno 1: Alberto García San José
Alumno 2: José Carlos Gómez Blasco
Alumno 3: Juan Carlos López de la Torre
Alumno 4: Rubén Montero Vázquez

3.10.3. Descripción del robot

El robot diseñado para la prueba del pañuelo curvo es *Trotalineas*. Para la prueba de cazar latas *Zampalatas* avanza hasta detectar una lata y la saca del tablero.



Figura 3.12: Robot Zampalatas



Figura 3.13: Robot Trotalineas

3.11. Gaspar Melchor de Jovellanos (II)

3.11.1. Datos del Instituto

IES Gaspar Melchor de Jovellanos
C/Móstoles 64, Fuenlabrada
28942
916971565

3.11.2. Participantes

Profesor Responsable: Víctor Gallego Le Forlot
Alumno 1: Patricia Maldonado Salguero
Alumno 2: Juan Israel Menéndez López
Alumno 3: Samuel Silva Calonge
Alumno 4: Carlos Jiménez Cepas

3.11.3. Descripción del robot

Husmeador es el robot diseñado para la prueba de atrapar latas. Es un robot con seis motores, tres sensores de luz y dos de contacto, además tiene una cesta para recoger las latas. Por su parte *La Bestia* es el robot diseñado para la prueba del pañuelo curvo.

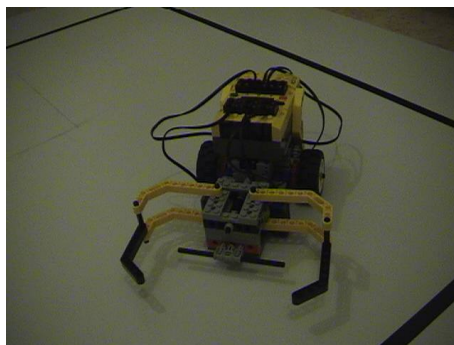


Figura 3.14: Robot La Bestia

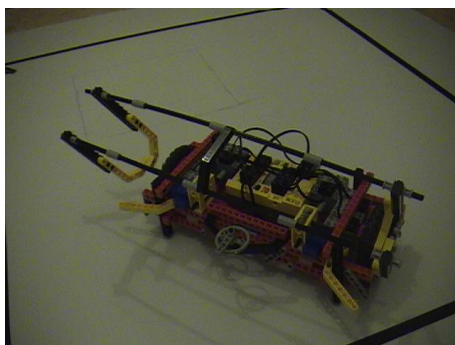


Figura 3.15: Robot Husmeador

3.12. Miguel Hernandez (I)

3.12.1. Datos del Colegio

C.P. Miguel Hernández
C/ Luis de Madrona S/N
28805 Alcalá de Henares
Madrid

3.12.2. Participantes

Profesor Responsable: Eduardo Gallego del Pozo
Alumno 1: Iván Gallego Bravo
Alumno 2: Silvia Santano Guillén
Alumno 3: Nerea de la Riva Iriepa
Alumno 4: Kelsey Jacobs Solar

3.12.3. Descripción del robot

Scorpy es un robot diseñado para la prueba de barredores de latas, el principal objetivo es diseñar y construir un robot que cumpla el reglamento de la prueba al 100 %, así pues usa un único programa para sacar latas blancas o negras (se seleccionan mediante un pulsador que se lee al comienzo del programa), se para cuando saca las tres latas del color especificado.

Las latas las transporta levantándolas mediante un mecanismo que se comienza a levantar cuando los dos brazos encuentran algo que sujetar. Para determinar si una lata es del color seleccionado, la coge y se la acerca para realizar la prueba con más seguridad.

Una vez levantada la lata es capaz de sacarla fuera del perímetro señalado esquivando, si fuera necesario, otras latas con los bumpers delanteros.

No-Se es un robot diseñado para la prueba del pañuelo curvo. Esencialmente el robot es un sigue-lineas que utiliza un único motor para avanzar y una pequeña rueda delantera que controla con otro

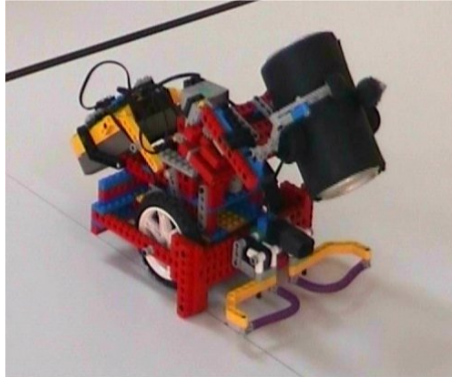


Figura 3.16: Robot Scorpy

motor. Para facilitar el avance dispone de un diferencial que permite que cada una de las dos ruedas motrices gire a distinta velocidad, si es necesario (en una curva, por ejemplo). Dispone también de un sistema de engranajes intercambiables para modificar la velocidad del robot.

La detección de la línea la realiza mediante un único sensor siendo incluso capaz de determinar cuando llega a la meta para depositar la lata y detenerse.

La lata la localiza mediante un sensor de contacto muy sensible. Para trasportarla se utiliza una pequeña catapulta que se dispara eléctricamente pero que actúa con la fuerza de dos gomas elásticas.

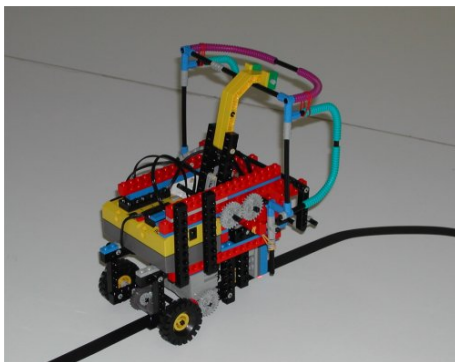


Figura 3.17: Robot No-Se

3.13. Miguel Hernandez (II)

3.13.1. Datos del Instituto

C.P. Miguel Hernández
C/ Luis de Madrona S/N
28805 Alcalá de Henares
Madrid

3.13.2. Participantes

Profesor Responsable: Eduardo Gallego del Pozo
Alumno 1: Jessica Rodriguez Doñoro
Alumno 2: Victoria Jabal Uriel
Alumno 3: Rocío Liebana Fernandez
Alumno 4: Concepción Arroyo Herranz

3.13.3. Descripción del robot

Whatapanashi es un robot diseñado para la prueba de barredores de latas, el principal objetivo fue diseñar y construir un robot que cumpliera el reglamento de la prueba al 100%, así pues usaba un único programa para sacar latas blancas o negras (se seleccionan mediante un pulsador que se lee al comienzo del programa), se paraba cuando sacaba las tres latas del color especificado.

Las latas las transporta arrastrandolas mediante un sistema de garras que se encuentra en su parte trasera. Una vez que detecta la presencia de una lata, el robot gira 180 grados para cogerla. Para determinar si una lata es del color seleccionado, se la acerca para realizar la prueba con más seguridad. En caso de no coincidir, suelta la lata y continua en busca de otra.

Dado que arrastra la lata con su parte trasera, le resulta muy fácil esquivar otras latas con los bumpers delanteros. Cuando encuentra la línea del perímetro gira para depositar la lata fuera de él.

SVC-23 es un robot diseñado para la prueba del pañuelo curvo.

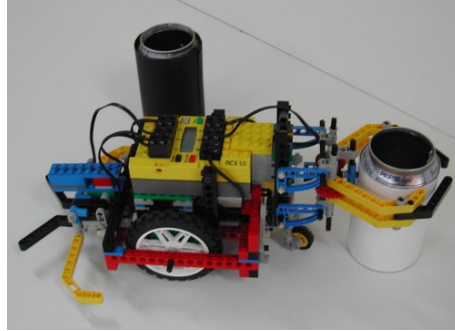


Figura 3.18: Robot Whatapanashi

Esencialmente el robot es un siguelíneas con dos rudas motrices con sendos motores.

La detección de la línea la realiza mediante dos sensores siendo incluso capaz de determinar cuando llega a la meta para depositar la lata y detenerse.

La lata la localiza mediante un sensor de contacto muy sensible. Para trasportarla se utiliza una pequeña catapulta que se dispara eléctricamente pero que actúa con la fuerza de dos gomas elásticas.



Figura 3.19: Robot SVC-23

3.14. Virgen de la Paz

3.14.1. Datos del Instituto

Centro: IES Virgen de la Paz
C/ Francisco Chico Mendes, 4
28100 - Alcobendas (Madrid)
916619004

3.14.2. Participantes

Profesor Responsable: Ángel Lizama Germes
Alumno 1: Ángel Yanguas
Alumno 2: Verónica Estevez
Alumno 3: Óscar Martínez

3.14.3. Descripción del robot

El robot *Atrapalatas*, diseñado para la prueba del pañuelo curvo, lleva dos ruedas motrices traseras acopladas directamente a sendos motores (salidas A y C del RCX). El vehículo avanza por la línea negra controlado por dos sensores de luz situados en la zona central del robot, que le indican cuando girar para no salirse de la línea negra. Cuando choca con la lata se acciona un sensor de contacto y además se libera de forma mecánica una estructura rectangular, sujeta por un simple enganche, que es volteada por encima de la lata, atrapándola. El sensor de contacto da la orden de invertir el sentido de giro de los motores, y el vehículo comienza a retroceder arrastrando la lata. De nuevo los sensores de luz evitan que pierda la línea negra.

En cuanto al programa éste pone en marcha los motores A y C y verifica una condición de luz del sensor 1. Si el nivel de luz es superior de 45 no hace nada, en caso contrario provoca un giro invirtiendo el sentido de giro del motor A. A continuación verifica un condicional de luz del sensor 3. Si el nivel de luz es superior de 45 no hace nada, en caso contrario provoca un giro invirtiendo el sentido

de giro del motor C. Seguidamente verifica el sensor de contacto 2. Si no está apretado regresa al inicio del programa, en caso contrario pasa a la segunda fase del programa en la que el avance es a la inversa.

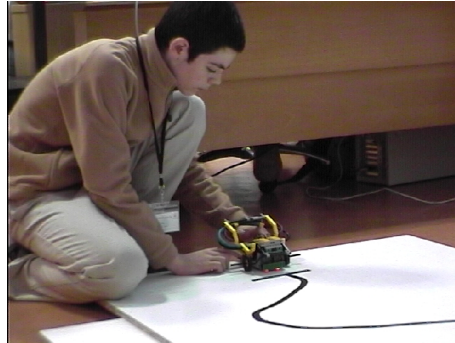


Figura 3.20: Robot atrapalatas

Torete2 es el robot usado para la prueba de barredores de latas. El robot lleva dos ruedas motrices traseras acopladas directamente a sendos motores (salidas A y C del RCX). Un sensor de luz (entrada 2) en su parte central, mira hacia el suelo e impide que salga del recinto de juego. Otro sensor de contacto (entrada 1) situado frontalmente, detecta cuando ha tocado una lata. Además un sensor de luz (entrada 3) frontal verificará si es una lata blanca o negra. Dos brazos a modo de cuernos amplían la franja barrida por el robot y facilitan su contacto con las latas.

3.14.4. Descripción del programa

El programa es muy sencillo. El robot avanza y seguidamente verifica si el nivel de luz en el sensor 2 es mayor de 45 en cuyo caso no hace nada, o menor en cuyo caso retrocede 1 segundo pero con un motor a plena potencia y el otro a nivel 2 y vuelve a avanzar. Seguidamente verifica el sensor de contacto. Si no está apretado no hace nada, y si lo está verifica el sensor de luz frontal (entrada 3). Si la luz es mayor de 45 no hace nada (es decir sigue avanzando en caso de tener que sacar las latas blancas del terreno de juego),

pero si es menor de 45 retrocede 1 segundo pero con un motor a plena potencia y el otro a nivel 2 y vuelve a avanzar (para salvar la lata). Finalmente regresa de nuevo al inicio del programa. En caso de tener que sacar las latas negras tiene preparado otro programa en el que la entrada 3 da las instrucciones inversas.



Figura 3.21: Robot Torete2

3.15. Juan Carlos I

3.15.1. Datos del Instituto

IES. Juan Carlos I
Av San Francisco s/n
Ciempozuelos 28350

3.15.2. Participantes

Profesor Responsable: Carlos Lamparero
Alumno 1: Emilio Manzanero
Alumno 2: Carmen Ji
Alumno 3: Carlos Cuadrado
Alumno 4: José Luis Rodríguez

3.15.3. Descripción del robot

Carpanta es el robot diseñado por el equipo del IES. Juan Carlos I para la prueba del pañuelo curvo.

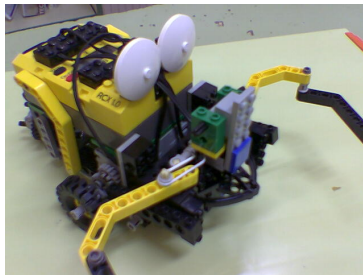


Figura 3.22: Robot Carpanta

3.16. Isabel la Católica

3.16.1. Datos del Instituto

I.E.S. Isabel la Católica
Alfonso XII, 3 y 5
28014 Madrid
Tno: 91-5277878

3.16.2. Participantes

Profesor Responsable: Manuel Contreras Porta
Alumno 1: Víctor Gómez
Alumno 2: Rebeca Cuesta
Alumno 3: Tomás González
Alumno 4: Alberto Díez

3.16.3. Descripción del robot

El montaje conmemora el episodio del Quijote donde éste se lanza sobre los molinos pensando que eran gigantes. Los robots se han montado utilizando RCX de Lego.

El funcionamiento es el siguiente: D. Quijote comienza a moverse en línea recta hacia un molino. Al chocarse, acciona un sensor de contacto, se para y comienza a girar $180\frac{1}{4}$ la parte superior, donde va montado un dibujo de D. Quijote con una lanza. En este momento avanza en sentido contrario hasta chocarse con otro molino. Otra vez vuelve a girar la parte superior $180\frac{1}{4}$, quedándose preparado para realizar un recorrido circular alrededor de un paisaje manchego. En el momento de comenzar el recorrido, envía una señal al RCX que representa a Sancho (con una silueta en la parte superior), que hace que se ponga en marcha. En estos momentos ambos personajes comienzan un recorrido circular, donde D. Quijote va delante de Sancho. Al final, como D. Quijote avanza mas rápido que Sancho, lo alcanza y se choca con él. En este momento D. Quijote se para. Sancho continúa el recorrido hasta que alcanza a D. Quijote, que

está parado, y se choca con él. Entonces se para y comienza a sonar una melodía grabada en su RCX. Esta melodía está sacada de la serie de dibujos animados que se emitió hace años.

Los recorridos realizados por D. Quijote y Sancho se basan en la utilización de un detector de luz que guía los robots por una línea negra sobre una pista de fondo blanco

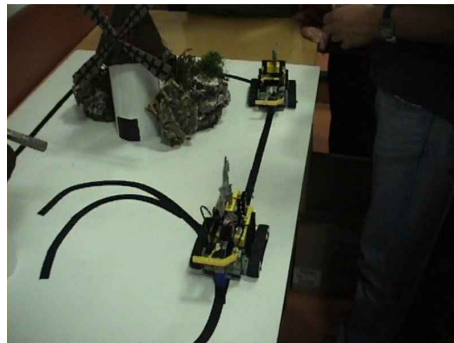


Figura 3.23: Montaje de los robots Don Quijote y Sancho Panza

3.17. Carmen Martín Gaité (I)

3.17.1. Datos del Instituto

SIES Carmen Martín Gaité Sección Arroyomolinos
Miguel de Cervantes, 104
Arroyomolinos

3.17.2. Participantes

Profesor Responsable: Alejandro Arévalo Viñuales
Alumno 1: Nadia Abdallah Kassab
Alumno 2: David Hernández Aguado
Alumno 3: Eric Magro Viforcós
Alumno 4: Cristina Nieto García

3.17.3. Descripción del robot

Sombra y *Arroyo1* son los robots con los que participó uno de los equipos del SIES Carmen Martín Gaité en Robocampeones 2005.

3.18. Carmen Martín Gaité (II)

3.18.1. Datos del Instituto

SIES Carmen Martín Gaité Sección Arroyomolinos
Miguel de Cervantes, 104
Arroyomolinos

3.18.2. Participantes

Profesor Responsable: Maribel López Ruiz
Alumno 1: Daniel Muñoz Fernández
Alumno 2: Pablo Hidalgo Fernández
Alumno 3: Julian López Espinosa
Alumno 4: Estefan Martínez Jereskes

3.18.3. Descripción del robot

Pincitas es el robot con el que participó el SIES Carmen Martín Gaité en Robocampeones 2005.

3.19. Antonio de Nebrija (I)

3.19.1. Datos del Instituto

IES Antonio de Nebrija
Avd. de la Omu, 81
916466443

3.19.2. Participantes

Profesor Responsable: Roberto Linares
Alumno 1: Elena Corrochano García
Alumno 2: Paola Espasa Borghesio
Alumno 3: Laura López Fernández
Alumno 4: Joseph Mensah Lozano

3.19.3. Descripción del robot

El Muñeco y *El Bixo* son los robots con los que participó uno de los equipos del IES Antonio de Nebrija en Robocampeones 2005.

3.20. Antonio de Nebrija (II)

3.20.1. Datos del Instituto

IES Antonio de Nebrija
Avd. de la Omu, 81
916466443

3.20.2. Participantes

Profesor Responsable: Agustín Vaquero
Alumno 1: Iván Gómez González
Alumno 2: Alex Pajares SánchezCid
Alumno 3: Javier García Bustos
Alumno 4: Christian Flores Carro

3.20.3. Descripción del robot

El robot *Robot5000* incorpora tres motores, dos para la tracción y uno más para controlar la bajado del lazo al atrapar la lata. Además va provisto de cuatro sensores: dos de luz y dos de contacto.

3.21. Enrique Tierno Galvan (I)

3.21.1. Datos del Instituto

IES Enrique Tierno Galvan
Avd. Juan Carlos I, s/n
916981111

3.21.2. Participantes

Profesor Responsable: Rafael Antonio Gil
Alumno 1: Esther Sánchez Martínez
Alumno 2: Daniel Fernández Yuste
Alumno 3: Irene Barrio San Pedro
Alumno 4: Zahira Marín Moral

3.21.3. Descripción del robot

El robot *Es-pañolator* fue diseñado para participar en la prueba del pañuelo curvo de Robocampeones 2005. Mientras que *BarreFerrary* fue el robot para la prueba del barredor de latas.

3.22. Enrique Tierno Galvan (II)

3.22.1. Datos del Instituto

IES Enrique Tierno Galvan
Avd. Juan Carlos I, s/n
916981111

3.22.2. Participantes

Profesor Responsable: David Díez López
Alumno 1: María Cristina Fernández Redondo
Alumno 2: Ainhoa Ventosinos Núñez
Alumno 3: Raúl Alcántara Campos
Alumno 4: Laura Sancho Serrano

3.22.3. Descripción del robot

Alonso Barredor fue el robot con el que participaron en la prueba del barredor de latas y *Pañi Pedrosa* el usado en la del pañuelo curvo.

3.23. Luis Vives

3.23.1. Datos del Instituto

IES Luis Vives
Pso. de la Ermita, 15
916807712

3.23.2. Participantes

Profesor Responsable: Henar Lastres García
Alumno 1: Cristina Riesco
Alumno 2: Juan Luis Vélez
Alumno 3: Belén Campo
Alumno 4: Isabel Martín

3.23.3. Descripción del robot

Los robots *NI FU* y *NI FA* fueron los diseñados para participar en las pruebas de Robocampeones 2005.

Bibliografía

- [Batavia00] **Parag Batavia and Illah Nourbakhsh** *Path Planning for the Cye Robot*. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2000, October 2000, Vol. 1, pages 15-20.
- [Baum99] **Dave Baum**. *Dave Baum's Definitive Guide to LEGO Mindstorms*. Apress, USA, 1999.
- [Bruyninckx01] **Herman Bruyninckx**
Open robot control software: the OROCOS project. Proceedings of the International Conference on Robotics and Automation, 2001. Vol 3. Pages 2523- 2528.
- [Gerkey03] **Brian Gerkey, Richard T. Vaughan and Andrew Howard** *The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems* Proceedings of the 11th International Conference on Advanced Robotics, pages 317-323, Coimbra, Portugal, June 2003 (ICAR'03).
- [Barahona00] **Jesús M. González-Barahona, Pedro de las-Heras-Quirós, José Centeno-González, Vicente Matellán-Olivera, and Francisco J. Ballesteros**. *Libre software in CS practice teaching (The experience at Carlos III University)*. IEEE Software, Vol. 17, No. 3, pp. 76-80, May/June 2000.

- [Hecker] **Frank Hecker.** *Setting Up Shop: The Business of Open-Source Software.*
<http://http://www.openresources.com/documents/setting-up-shop>
- [Kitano97] **Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda and Eiichi Osawa**
RoboCup: The Robot World Cup Initiative. Proceedings of the first international conference on Autonomous agents table of contents, 1997. Marina del Rey, California, United States. Pages 340-347.
- [Knudsen99] **Jonathan B. Knudsen.** *The Unofficial Guide to LEGO MINDSTORMS[tm] Robots.* O'Reilly, 1st edition edition, 1999.
- [Kumar98] **Deepak Kumar and Lisa Meeden.** *A Robot Laboratory for Teaching Artificial Intelligence* Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE-98), February 1998. Pages 341-344.
- [Libre00] **Working group on Libre Software.** *Free Software / Open Source: Information Society Opportunities for Europe?.*
<http://eu.conecta.it>. 2000
- [Meyer03] **Tony Meyer.** *Building Cost-effective Research Platforms: Utilising Free — Open-source Software in Research Projects.* Res. Lett. Inf. Math. Sci. (2003) 4, 91-99.
<http://iims.massey.ac.nz/research/letters/volume4/10meyer.pdf>.
- [Lego] **Markus L. Noga.** *Open-source embedded operating system for the LEGO Mindstorms.* <http://www.noga.de/lego0S/>.
- [Ohara03] **Keith O'Hara and Jennifer S.Kay.** *Investigating Open Source Software and Educational Robotics.* The Journal of Computing Sciences in Colleges, Volume 18 , Issue 3 (February 2003), Pages 8-16.
<http://elvis.rowan.edu/~kay/papers/OSSEduRob.pdf>.

- [Raymond98] **Eric S. Raymond.** *The Cathedral and the Bazaar*, 1998, Available at <http://catb.org/~esr/writings/cathedral-bazaar/>.
- [Stallman98] **Richard M. Stallman.** *Why Software Should Not Have Owners.* 1998. Available at <http://www.gnu.org/philosophy/why-free.html>.
- [Stallman99] **Richard M. Stallman.** *The GNU Manifesto.* <http://www.gnu.org/gnu/manifesto.html>
- [UNECE00] **United Nations Economic Commission for Europe** *The Boom in Robot Investment Continues - 900,000 Industrial Robots by 2003.* Press Release ECE/STAT/00/10. Geneva, 17 October 2000. <http://www.unece.org/press/00stat10e.htm>